



Manuel Felipe Arcila Gago

Modelo de Juntas Soldadas por FSW utilizando Métodos de Aprendizagem de Máquina através de Dados Experimentais

65/2013

**CAMPINAS
2013**



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

Manuel Felipe Arcila Gago

Modelo de Juntas Soldadas por FSW utilizando Métodos de Aprendizagem de Máquina através de Dados Experimentais

Orientador: Prof. Dr. Janito Vaqueiro Ferreira

Dissertação de Mestrado apresentada à Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas para obtenção do título de Mestre em Engenharia Mecânica, na Área de Mecânica dos Sólidos e Projeto Mecânico.

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL
DA DISSERTAÇÃO DEFENDIDA PELO(A) ALUNO(A)
Manuel Felipe Arcila Gago, E ORIENTADO(A)
PELO(A) PROF(A). DR(A) Janito Vaqueiro Ferreira

Janito Vaqueiro Ferreira
ASSINATURA DO(A) ORIENTADOR(A)

CAMPINAS
2013

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

Ar26m Arcila Gago, Manuel Felipe, 1987-
Modelo de juntas soldadas por FSW utilizando métodos de aprendizagem de máquina através de dados experimentais / Manuel Felipe Arcila Gago. – Campinas, SP : [s.n.], 2013.

Orientador: Janito Vaqueiro Ferreira.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Soldadura por fricção. 2. Análise de variância. 3. Redes neurais (Computação). 4. Máquina de vetores de suporte. 5. Método de Monte Carlo. I. Ferreira, Janito Vaqueiro, 1961-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Welded joint model by FSW using machine learning methods through experimental data

Palavras-chave em inglês:

Friction stir welding

Analysis of variance

Neural network

Support vector machine

Monte Carlo method

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Titulação: Mestre em Engenharia Mecânica

Banca examinadora:

Janito Vaqueiro Ferreira [Orientador]

Alberto Luiz Serpa

Vicente Lopes Júnior

Data de defesa: 30-07-2013

Programa de Pós-Graduação: Engenharia Mecânica

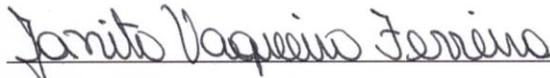
**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE MECÂNICA COMPUTACIONAL**

DISSERTAÇÃO DE MESTRADO ACADÊMICO

**Modelo de Juntas Soldadas por FSW utilizando
Métodos de Aprendizagem de Máquina através
de Dados Experimentais**

Autor: Manuel Felipe Arcila Gago
Orientador: Prof. Dr. Janito Vaqueiro Ferreira

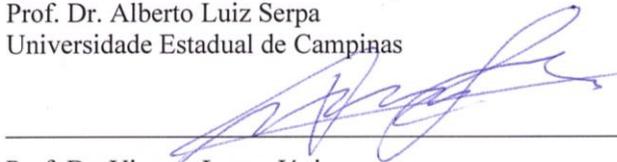
A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:



Prof. Dr. Janito Vaqueiro Ferreira
Universidade Estadual de Campinas



Prof. Dr. Alberto Luiz Serpa
Universidade Estadual de Campinas



Prof. Dr. Vicente Lopes Júnior
Universidade Estadual Paulista

Campinas, 30 de Julho de 2013

Dedico este trabalho aos meus pais, Edgar e Maria, por ter-me guiado pelo caminho do sucesso e a meu irmão pelo apoio.

Agradecimentos

Quero começar expressando meus profundos agradecimentos à Universidade Estadual de Campinas por me permitir realizar meus estudos de mestrado no Brasil e a empresa Embraer pelo financiamento do projeto. Durante o tempo de estudo, tive a oportunidade de adquirir conhecimentos nas áreas pessoal e profissional.

Eu particularmente quero agradecer ao meu orientador Janito Vaqueiro e ao professor Renato Pavanello, tendo aberto as portas do seu grupo de pesquisa, além de tem me dado não só o conhecimento, se não também a sua confiança e apoio incondicional para chegar a esta conquista.

Por outro lado, mais do que um agradecimento dedico esta dissertação a minha grande e especial família, pessoas certas com quem Deus tem me abençoado.

Aos meus pais Maria e Edgar, sem eles não teria conseguido ser a pessoa que eu sou "Obrigado pelo apoio que sempre me deram, respeitando as minhas decisões, não importa se elas estão erradas". A meu irmão Sebastian por estar sempre comigo em tudo momento e por sua constante ajuda e apoio.

A Jenny, que apesar da distância sempre esteve ao meu lado e muitas vezes era a minha fonte de inspiração e alegria. "Obrigado por sua paciência e pelo amor que você me oferecer."

Aqui termina um longo Caminho, valeu a pena. Acabei de fazer um dos meus sonhos nesta vida. Finalmente, graças à minha lista longa e valiosa de amigos, sempre estarão no meu coração por tudo que vocês têm me dado neste momento, pelo apoio e incentivo nos momentos difíceis e as alegrias que eu fiz ao vivo. "Obrigado a todos os amigos".

*“As pessoas que vencem neste mundo são as que procuram as
circunstâncias de que precisam e, quando
não as encontram, as criam”*

Bernard Shaw

Resumo

Gago, M. F. A. Modelo de Juntas Soldadas por FSW utilizando Métodos de Aprendizagem de Máquina através de Dados Experimentais. Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2013.164p. Dissertação (Mestrado).

A variedade de materiais no setor aeronáutico para redução de peso e custo tem se proliferado a um grau intensivo, onde tem sido revisadas diferentes pesquisas para encontrar outros tipos de materiais de fácil maneabilidade para construção de peças que satisfazem as restrições impostas. Assim, existe uma procura constante de soluções para facilitar a produção, e ao mesmo tempo aumentar a segurança das aeronaves levando em consideração pontos importantes como a fadiga e ruptura do material. Um material frequentemente utilizado que atende a estes requisitos devido a suas propriedades de densidade e resistência é o alumínio, e é neste ambiente que existe um processo de manufatura utilizado para a soldagem conhecido como “*Friction Stir Welding*” (FSW). No presente momento, estudos para criação de modelos que representem características mecânicas utilizadas em projetos em função de parâmetros do processo tem sido pesquisados. Embora este processo seja de difícil modelagem devidos as suas complexidades, tem sido estudado e utilizado diferentes algoritmos que possibilitem o melhoramento da representação do modelo, tais como os relacionados com máquinas de aprendizagem (ML) e suas diferentes otimizações. Neste contexto, a presente pesquisa tem seu foco na obtenção de um modelo baseado no algoritmo de aprendizagem de Máquina de Vetores de Suporte (SVM), e também com outros algoritmos tais como Regressão Polinomial (RP) e Rede Neural Artificial (RNA), buscando encontrar modelos que representem o processo de soldagem por FSW através das propriedades mecânicas obtidas pelos ensaios de tração e por análise de variância (ANOVA), entendendo suas vantagens e, posteriormente, recomendar quais dos algoritmos de aprendizagem tem maior benefício.

Palavras Chave: *Friction Stir Welding, Análise de Variância, Rede Neural, Máquina de Vetores de Suporte, Método de Monte Carlo.*

Abstract

Gago, M. F. A. Welded Joints Model by FSW using Machine Learning Methods through Experimental Data. Campinas: Department of Mechanical Engineering, University of Campinas, 2013. 164 p. Thesis (Master).

In the aerospace industry to reduce weight and cost, a great quantity of materials has been used, which has generated research to find types of materials, that have been better maneuverability and to guarantee the properties required to development of pieces for the industry. Thus, the studies look for optimize between production easiness and increase the aircraft safety, taking into consideration important issues such as fatigue and fracture of the materials. One of the most common approach used is aluminum by their mechanical properties (density and strength), although it has many problems to be welding with the traditional methods.

Currently, the Friction Stir Welding (FSW) process is used in the industry, as well in the academy. However, the FSW is difficult to model by the complexities in the physical phenomenal occurred during the weld process, as result, has been studied and used different algorithms that allow enhance the model representation. The Machine Learning (ML) is a methodology studied to obtain the model optimized. In this context, the present research focus by to obtain a model-based in learning algorithm using Support Vector Machine (SVM). Although comparisons were made with other algorithms such as Polynomial Regression (PR) and Artificial Neural Network (ANN), searching to find models that represent the FSW process weld using the mechanical properties obtained by tensile tests and analysis of variance (ANOVA). Finally, conclusions to understand the advantages learning algorithms are presented.

Key Words: *Friction Stir Welding, Analysis of Variance, Neural Network, Support Vector Machine, Monte Carlo Method.*

Lista de Figuras

Figura 1.1 Arquitetura da Proposta.	5
Figura 2.1 Sistema Dinâmico com Entrada e Saída.	8
Figura 2.2 Sistema de Identificação.	9
Figura 2.3 Esquema do Processo de Soldagem. Adaptado de (KUMAR AND KAILAS, 2008).	11
Figura 2.4 Estrutura Aprendizagem de Máquina.	12
Figura 2.5 Aprendizagem de Máquina - Caixa Preta.	14
Figura 2.6 Exemplo de aprendizagem Supervisionado Regressão.	16
Figura 2.7 Exemplo de Aprendizagem Supervisionado de Classificação.	16
Figura 2.8 Exemplo de Aprendizagem não Supervisionado.	17
Figura 2.9 Curvas de Aprendizagem de Desambiguação para Conjunto de Confusão.	19
Figura 2.10 Neurônio em Comparação com Modelo do RNA. (a) Neurônio Real Adaptado de [FEUERSTEIN et al.,2013]. (b) Neurônio Matemático.	20
Figura 2.11 Comunicação dos Neurônios no Cérebro. (NATIONAL INTITUTE ON AGING, 2008).	21
Figura 3.1 Procura do Mínimo da Função: (a) Entradas Escalonadas; (b) Entradas sem Escalar.	25
Figura 3.2 Função de Custo da Regressão Logística.	28
Figura 3.3 Função de Custo do (SVM) versus Regressão Logística.	28
Figura 3.4 Exemplo de uma Amostra de Treinamento pelos Parâmetros θ	31
Figura 3.5 Exemplos de Classificação, Comportamento da Margem.	32
Figura 3.6 Exemplo de Mapeamento do SVM, foi Convertido para um Problema Linearmente Separável.	34
Figura 3.7 Similaridade, Kernel Gaussiano para uma Amostra.	35
Figura 3.8 Análise da Gaussiana com Diferentes Parâmetros de σ^2 no Ponto [0 0].	36
Figura 3.9 Bias ou Variância com Respeito aos Modelos.	42
Figura 3.10 Gráfico de Curvas de Aprendizagem.	43
Figura 3.11 Curva de Aprendizagem. (a) Apresenta Alta Bias (b) Apresenta Alta Variância.	44
Figura 3.12 Algoritmo Gráfico de Roleta.	48
Figura 4.1 Arquitetura das Etapas do Experimento.	51
Figura 4.2 Esquema da Criação do Modelo da Solda FSW.	52
Figura 4.3 Esboço do Corpo de Prova Utilizado.	53
Figura 4.4 Curva Características de Ensaio de Tração.	55
Figura 4.5 Esquema de Algoritmo de Aprendizagem.	56
Figura 4.6 Esquema do Modelo de Aprendizagem dos Métodos.	57
Figura 4.7 Transformação do Kernel do Método SVMr.	61
Figura 4.8 Arquitetura do Método SVM para Regressão.	61
Figura 4.9 Variável de Perda ξ (Margem).	63
Figura 4.10 Esquema do Filtro de Partículas para SVMr: (a) Ponto Inicial Aplicando Covariância; (b) Partículas que Sobrevieram por seus Pesos; (c) Covariância de cada Partícula Sobrevivente; (d) Partículas que Sobrevieram na Seguinte Iteração; (e) Ponto Médio do Valor $[C,\sigma,\epsilon]$; (f) Convergência dos Valores $[C,\sigma,\epsilon]$	69

Figura 5.1 Efeitos dos Fatores sobre os Parâmetros de Resposta.....	75
Figura 5.2 Superfície de Resposta de Alongamento de Escoamento.	78
Figura 5.3 Superfícies de Respostas que Apresentaram Fator de Curvatura: (a) T.U. Fatores Significativos; (b) T.U. Relação ao Fator não Significativo; (c) T.E. Fatores Significativos; (d) T.E. Relação ao fator não Significativo; (e) A.F. Fatores Significativos; (f) A.F. Relação ao Fator não Significativo.	79
Figura 5.4 SVMr - Curvas de Aprendizagem MSE – Sample: (a) Tensão de Escoamento (b) Tensão Última (c) Alongamento.....	83
Figura 5.5 Regressão por o Método de SVM para a Resposta de Tensão de Escoamento.	85
Figura 5.6 Superfície de Resposta da Tensão de Escoamento por o Algoritmo SVMr; (a) VR-VT; (b) VR-FA; (c) VT-FA.	86
Figura 5.7 Evolução do Erro de Teste do Sistema SVMr com Filtro de Partículas com Diferentes Números de Partículas em cada Iteração.....	88
Figura 5.8 Comportamento das Partículas no Algoritmo. Diferentes Números de Partículas.....	89
Figura 5.9 Comportamento do Erro em cada Iteração do Algoritmo SVMr com Filtro de Partículas.....	91
Figura 5.10 Comportamento dos Parâmetros dos Estados Ocultos do Filtro de Partículas; (a) Regularização - C; (b) “Kernel” – σ^2 ; (C) Restrições da Margem - ϵ	92
Figura 5.11 Superfície de Resposta da Tensão de Escoamento por o Algoritmo SVMr-FP.....	94
Figura 5.12 Comparação dos Resultados dos Métodos SVMr+PF; SVMr; Rede Neural; e Regressão Polinomial dos Dados de treinamento.....	96
Figura B.1 Processo da Avaliação da Hipóteses.	108
Figura C.1 Espécimen para Ensaios de Tração.	110
Figura C.2 Curva Características de Ensaios de Tração Padrão.	111
Figura C.3 Ponto de Tensão de Escoamento na gráfica de Tensão-Deformação.....	112
Figura C.4 Corpos de prova com solda FSW da matriz 2 condição 1; sensor MTS no teste; e equipamento MTS para os ensaios de tração.....	113
Figura D.1 Função Sigmoide.	114
Figura D.2 Arquitetura da Rede Neural para Classificação.	116
Figura D.3 Gráfica de “Forward Propagation”.	117
Figura D.4 Gráfica de “Back Propagation”.....	119
Figura E.1 Divergência ao Mínimo da Função (Mínimo Local e Global).	120
Figura E.2 (a)Derivada Encontrando o Mínimo com Atualização Unidimensional (b) Mínimo Local da Função.....	122
Figura G.1 Arquitetura da RNA para o Modelo FSW.	127
Figura i.1 Regressão Polinomial - Evolução dos Erros dos Modelos MSE – Iterações: (a) Treinamento da Tensão de Escoamento (b) Teste da Tensão de Escoamento.	134
Figura ii.1 Rede Neural - Evolução dos erros dos Modelos MSE – Iterações: (a) Treinamento da Tensão de Escoamento (b) Teste da Tensão de Escoamento.....	138
Figura iii.1 Regressão Polinomial - Curvas de Aprendizagem MSE - Sample: (a) Tensão de Escoamento (b) Tensão Última (c) Alongamento.	139
Figura iii.2 Superfície de Resposta da Tensão de Escoamento por o Algoritmo Regressão Polinomial; (a) VR-VT; (b) VR-FA; (c) VT-FA.....	140

Figura iv.1 Rede Neural - Curvas de Aprendizagem MSE - Sample: (a) Tensão de Escoamento (b) Tensão Última (c) Alongamento.....	142
Figura iv.2 Superfície de Resposta da Tensão de Escoamento por o Algoritmo RNA; (a) VR-VT; (b) VR-FA; (c) VT-FA;	143

Lista de Tabelas

Tabela 2.1 Exemplos de Aprendizagem de Máquina (NG, 2012).	13
Tabela 2.2 Breve Historia das Redes Neurais. Baseado em (EBERHART, 1990) e (EBERHART et al., 2007).	22
Tabela 2.3 Breve História de SVM. Baseado em (BOSER, 1992; VAPNIK, 1999; SHAWE-TAYLOR , CRISTIANINI ,2000).	23
Tabela 3.1 Parâmetros para Escalonar Características.	26
Tabela 3.2 Funções Kernel.	34
Tabela 3.3 Comparação entre SVM e RNA.	38
Tabela 4.1 Fatorial Completo de dois Níveis.	53
Tabela 4.2 Níveis das Variáveis de Entrada para os Ensaios.	54
Tabela 5.1 Níveis das Variáveis de Entrada para os Ensaios – Matriz 1.	71
Tabela 5.2 Níveis das Variáveis de Entrada para os Ensaios – Matriz 2.	71
Tabela 5.3 Níveis das Variáveis de Entrada para os Ensaios – Matriz 3.	72
Tabela 5.4 Fatorial Completo de dois Níveis.	72
Tabela 5.5 Valores Reais das Variáveis e Parâmetros Mecânicos de cada Experimento do Planejamento 23.	74
Tabela 5.6 ANOVA para as Respostas Tensão Última, Tensão de Escoamento, Módulo de Elasticidade e Alongamento da Falha.	76
Tabela 5.7 Preditividade e Significância do Modelo Matemático do Alongamento de Escoamento.	77
Tabela 5.8 Parâmetros e (MSE) Finais Obtidos por o Método Validação Cruzada para SVMr.	81
Tabela 5.9 Erros (MSE) e Coeficiente de Determinação (R^2) de Treinamento-Validação e Teste para os Modelos Finais da SVMr.	82
Tabela 5.10 A Porcentagem do Erro (MAPE) de Treinamento e Validação para SVMr.	84
Tabela 5.11 Resultados de Treinamento-Validação para o SVMr com Filtro de Partículas com Diferentes Números de Partículas.	87
Tabela 5.12 Resultados do Erro e Coeficiente de Determinação de Treinamento-Validação e Teste para o SVMr com Filtro de Partículas com Diferentes Números de Partículas.	88
Tabela 5.13 A Porcentagem do Erro (MAPE) de Treinamento e Validação para SVMr.	93
Tabela 5.14 Comparação dos Resultados dos Métodos SVMr+PF; SVMr; Rede Neural e Regressão Polinomial.	95
Tabela 5.15 Comparação dos Erros Porcentual dos Métodos SVMr+PF; SVMr; Rede Neural e Regressão Polinomial.	95
Tabela i.1 Modelos de Regressão Polinomial	132
Tabela i.2 Resultados de Treinamento-Validação para os Modelos Desenvolvidos de Regressão Polinomial.	133
Tabela i.3 Parâmetros e (MSE) finais obtidos por o método validação cruzada para Regressão Polinomial.	133
Tabela i.4 Erros (MSE) e Coeficiente de determinação (R^2) de Treinamento-Validão e Teste para os Modelos Finais da Regressão Polinomial.	133
Tabela i.5 Regressão Polinomial - Treinamento e Teste dos Modelos Selecionados pelas Iterações.	134

Tabela ii.1 Modelos da RNA.....	136
Tabela ii.2 Resultados de Treinamento-Validação para os Modelos Desenvolvidos de Redes Neurais.	136
Tabela ii.3 Parâmetros e (MSE) Finais Obtidos por o Método Validação Cruzada para Rede Neural.	137
Tabela ii.4 Erros (MSE) e Coeficiente de Determinação (R^2) de Treinamento-Validação e Teste para os Modelos Finais da Rede Neural.	137
Tabela ii.5 Rede Neural -Treinamento e Teste dos Modelos Selecionados pelas Iterações.	138
Tabela iii.1 A Porcentagem do Erro (MAPE) de Treinamento e Validação para Regressão Polinomial ..	140
Tabela iii.2 Resultados de Treinamento e Teste para os Modelos Desenvolvidos de Regressão Polinomial.	141
Tabela iv.1 A porcentagem do Erro (MAPE) de Treinamento e Validação para Rede Neural.....	143
Tabela iv.2 Resultados de Treinamento e Teste para os Modelos Desenvolvidos de Rede Neural.	144

Lista de Abreviaturas e Siglas

Abreviações

AE – Alongamento de Escoamento

AF – Alongamento de Falha (Ruptura)

ANOVA – Análise de Variância

BFGS – Broyden, Fletcher, Goldfarb, Shanno (Algoritmo de Quase-Newton)

FA – Força Axial

FP – Filtro de Partículas

FSW – Friction Stir Welding

GA – Algoritmo Genético

K – Kernel

MAPE – Porcentagem Absoluto do Erro

ME – Módulo de Elasticidade

ML – Machine Learning (Aprendizagem de Máquina)

MSE – Mean Square Error (Erro Quadrático Médio)

PSO – Otimização de Enxame de Partículas

QP – Quadratic Programming (Programação Quadrática)

RL – Regressão Logística

RNA – Rede Neural Artificial

RP – Regressão Polinomial

SV – Support Vector (Vetores de Suporte)

SVM – Support Vector Machine (Máquina de Vetores de Suporte)

SVMr – Support Vector Machine Regression (Máquina de Vetores de Suporte para Regressão)

TE – Tensão de Escoamento

TF – Tensão de Falha (Ruptura)

TU – Tensão Última

VR – Velocidade de Rotação

VT – Velocidade de Translação

Siglas

b – Termo de Bias

C – Termo de Penalização do SVM

Cost – Função para Classificação do SVM

d_k – Ruído da Função dos Vectors de Medições

f – Características Transformadas

$f(x)$ – Função do Modelo

f_k – Estados não Lineares do FP

h_k – Função de Medida do FP

h_θ – Hipóteses do Modelo

$J(\theta)$ – Função de Custo

l – “*Landmarks*” para Utilização do Kernel

L – Função de Lagrange

m – Número de Amostras

M – Matriz do “*Kernel*”

n – Número de Características

n_p – Número de Partículas

p – Função de Probabilidade do FP

q – Função de Densidade do FP

R^2 – Coeficiente de Determinação

s – Desvio Padrão dos Dados

v_k – Ruído da Função dos Vectors de Estado

w – Pesos do FP

x_d – Entradas Desejada

x_k – Vetor de Estado do FP

y_p – Saída Predita

z – Combinação Linear

z_k – Vetor de Medições do FP

λ – Termo de Regularização do RNA, RL e RP

μ – Média dos Dados

θ – Parâmetros da Função

α – Multiplicador de Lagrange

α_a – Taxa de Aprendizagem

Φ – Função Kernel

σ^2 – Fator do Kernel

ξ – Dados Errados na Classificação ou Regressão

$\frac{\partial}{\partial \theta_j}$ – Gradiente

ε – Fator da Margem SVMr

Sumário

<i>Capítulo 1 . INTRODUÇÃO</i>	1
1.1 Introdução	1
1.2 Motivação e Contextualização	2
1.3 Proposta do Trabalho	3
1.3.1 Objetivo Geral	3
1.4.2 Objetivos Específicos	4
1.4 Arquitetura	4
1.6 Organização da Dissertação	6
<i>Capítulo 2 . REVISÃO BIBLIOGRÁFICA</i>	8
2.1 Identificação de Sistemas.....	8
2.2 Friction Stir Welding	10
2.3 Aprendizagem de Máquina	12
2.3.1 Tipos de Aprendizagem de Máquina.....	15
2.3.2 Histórico Evolutivo da Aprendizagem de Máquina	17
2.3.3 Histórico Evolutivo dos Algoritmos de Aprendizagem de Máquina.....	19
<i>Capítulo 3 . FUNDAMENTOS TEÓRICOS</i>	25
3.1 Fundamentos para Realizar Aprendizagem de Máquina	25
3.1.1 Escalonando Características	25
3.1.2 Desenvolvimento dos Parâmetros	26
3.2 Algoritmo de Máquina de Vetores de Suporte	27
3.2.1 Definição Matemática dos Vetores de Suporte	27
3.2.2 Margem da Fronteira	29
3.2.3 Caso não Linear: Kernels	34
3.3 Diagnóstico de Aprendizagem de Máquina	39
3.3.1 Avaliar a Hipótese	39
3.3.2 Seleção do Modelo	40
3.3.3 Diagnóstico Bias Versus Variância.....	41
3.3.4 Curvas de Aprendizagem.....	42
3.4 Filtro de Partículas.....	44

3.4.1	Características do Filtro de Partículas	44
3.4.2	Matemática do Filtro de Partículas	45
3.4.3	Funcionamento do Filtro de Partículas	46
3.4.4	Procedimento do Filtro de Partículas	47
<i>Capítulo 4 . METODOLOGIA</i>		50
4.1	Realização de Experimentos	50
4.1.1	Análise Preliminar	51
4.1.2	Análise de Dados	52
4.1.3	Análise de Algoritmos.....	55
4.2	Máquina de Vetores de Suporte Regressão.....	58
4.2.1	Formulação Preliminar	58
4.2.2	Caso Não Linear: Kernel	62
4.2.3	Hiperplano de Regressão	63
4.2.4	Esquema da Máquina de Vetores de Suporte.....	64
4.3	Filtro de Partículas.....	65
4.3.1	Modelo de espaço de estados de SVMr.....	65
4.3.2	Implementação do Filtro de Partículas para o Treinamento de SVMr	66
4.3.3	Esquema do Filtro de Partículas em SVMr	68
<i>Capítulo 5 . RESULTADOS</i>		70
5.1	Planejamento Fatorial Completo 2 ³	70
5.1.1	Planejamento de Experimentos e Matriz de Testes.....	70
5.1.2	Análise Estatística.....	73
5.2	Máquina de Vetores de Suporte para Regressão	80
5.2.1	Seleção dos Modelos do SVMr.....	80
5.2.2	Evolução dos Modelos Finais do SVMr.....	81
5.2.3	Curvas de Aprendizagem do SVMr.....	82
5.2.4	Análise de Predição das Respostas da SVMr.....	84
5.2.5	Superfície de Resposta da SVMr	85
5.3	SVMr ajustando os parâmetros com o Filtro de Partículas.....	87
5.3.1	Comportamento do Erro e Parâmetros do Modelo.....	90
5.3.2	Análise de Predição das Respostas da SVMr.....	93

5.3.3	Superfície de Resposta do SVMr - FP	93
5.4	Comparação Dos Resultados.....	94
<i>Capítulo 6 . CONCLUSÕES E RECOMENDAÇÕES DE TRABALHOS FUTUROS.....</i>		<i>97</i>
6.1	Conclusões.....	97
6.2	Recomendações de Trabalhos Futuros	99
REFERÊNCIAS BIBLIOGRÁFICAS		101
APÊNDICE A.	– Metodologia para a Avaliação do Modelo	106
APÊNDICE B.	– Análise Estatística	108
APÊNDICE C.	– Curva de Tensão-Deformação	110
APÊNDICE D.	– Algoritmos de Classificação	114
APÊNDICE E.	– Otimização	120
APÊNDICE F.	– Regressão Polinomial	124
APÊNDICE G.	– Rede Neural.....	127
ANEXO i.	– Regressão Polinomial Arquitetura.....	132
ANEXO ii.	– Rede Neural Arquitetura	135
ANEXO iii.	– Resultados de Regressão Polinomial.....	139
ANEXO iv.	– Resultados de Rede Neural.	142

Capítulo 1 .

INTRODUÇÃO

1.1 Introdução

Um dos principais focos de estudo no sector aeronáutico está baseado na redução de peso. Isto pode significar uma diminuição de custos de produção e melhora do produto que se encontram relacionados à redução de uso de combustíveis e maiores facilidades de produção. Um material utilizado que atende estes requisitos devido as suas propriedades de densidade e resistência é o alumínio. Apesar desta vantagem, o processo de soldagem para este material não apresenta facilidade de uso pelos métodos convencionais em relação a seu baixo ponto de fusão, gerando assim, uma necessidade de achar um processo mais eficiente de soldagem. Moreira (2008) mostra em sua pesquisa alguns dos diferentes métodos da soldagem do alumínio, a solda à laser “*Laser Beam Welding*” e a solda por atrito “*Friction Stir Welding*” (FSW).

O processo de FSW é um método para soldagem que foi desenvolvido pelo Instituto Tecnológico de Soldagem (TWI) em 1991. O FSW é um processo que consiste no movimento de uma ferramenta associada com movimentos de translação e de rotação pré-estabelecidos, sendo a mesma pressionada contra a linha de união entre duas peças, ou seja, um processo em estado sólido, com baixo calor de entrada comparado a outros processos de soldagem. O aquecimento local causado pelo atrito entre a ferramenta e a peça causa um amolecimento e mistura dos metais, o que é suficiente para que os materiais se unam (SHIVARAJ, et al, 2010).

Este projeto tem por objetivo um estudo de algoritmos considerados adequados para a obtenção de um modelo de propriedades mecânicas em ensaios de tração, que relacione os parâmetros de entrada com os de saída do processo na soldagem FSW.

Um procedimento bastante conhecido baseia-se em análise estatísticas nos dados obtidos, no qual é possível observar quando os fatores de entrada são significativos nas respostas, assim como se os modelos são preditivos. Para os modelos onde se tem um comportamento não linear é

possível a utilização de métodos baseados em aprendizagem de máquina (ML - “*Machine Learning*”) como a Regressão Polinomial (RP), as Redes Neurais Artificiais (RNA), o método “*Support Vector Machines Regression*” (SVMr), e as Árvores de Regressão os quais obtêm um modelo que relaciona entrada/saída.

1.2 Motivação e Contextualização

Na área da aeronáutica tem sido pesquisado diferentes formas para união de peças na solução de problemas de resistência e peso. Assim, no processo da solda FSW, devido as complexidades intrínsecas, torna-se difícil a construção de modelos matemáticos não paramétricos que descrevem o seu comportamento.

É neste cenário que se insere a síntese de ferramentas computacionais para treinamento supervisionado e não-supervisionado, a partir de dados amostrados. Esta exploração consistente na capacidade de aproximação universal, produz o que se convencionou chamar de maximização da capacidade de generalização, ou seja, obtenção de desempenho ótimo junto a dados não observados durante o processo de treinamento. A teoria de regularização (VAPNIK, 1999), assim como conceitos de teoria de informação (CORTES; VAPNIK, 1995) e de teoria de aprendizado estatístico (IZENMAN, 2008) fornecem subsídios para operar de forma eficaz com modelagem não paramétrica em aprendizado de máquina do processo FSW.

A Regressão Polinomial tem limitações pela quantidade de interações dos termos utilizados em sua aproximação, além de seu custo computacional se seu polinômio é muito elevado. As RNA são os métodos mais utilizados, já que podem aproximar qualquer função não linear. Embora, o método RNA tem alguns problemas como é a presença de mínimos locais e a escolha da estrutura de rede. Neste sentido, o método da máquina de vetor de suporte (SVM) tem triunfado nos últimos anos de aplicações como se mostra em (XUNKAI et al. 2004).

Máquina de vetor de suporte para regressão (SVMr) tem uma função, no âmbito da teoria do aprendizado estatístico e da teoria da regularização clássica para aproximação de funções (VAPNIK et al. 1997, EVGENIOU et al. 2000).

Têm sido realizadas algumas pesquisas sobre as características deste processo de soldagem (FSW), onde por meio de análise estatística e Máquina de Vetores de Suporte, obtiveram-se os efeitos entre os parâmetros de soldagem FSW em alumínio e suas propriedades mecânicas e criaram-se modelos.

Os efeitos dos parâmetros de processo na soldagem FSW, tais como a rotação da ferramenta e velocidade de soldagem, sobre as propriedades mecânicas de juntas unidas por FSW foi investigado por Rajamanickam et al. (2009) e Sarsilmaz, Çaydas (2009) usando planejamentos de experimentos fatoriais e análise de variância (ANOVA).

Os trabalhos de Yousif et al.(2008) e Okuyucu et al. (2007) empregam redes neurais para criar um modelo em que as entradas são as velocidades de rotação e de translação na soldagem, e tem com saída a tensão de escoamento, tensão de ruptura e alongamento. Estes trabalhos mencionam que o modelo de redes neurais apresentaram um comportamento considerado adequado.

Shuangsheng et al. (2012) empregaram regressão com vetores de suporte para criar um modelo de juntas soldadas. Foi desenvolvido baseado em diferentes funções “*kernel*”, no qual os parâmetros são a velocidade de soldagem e o fluxo de argon.

Na literatura, os principais aspectos para a escolha do SVM estão relacionados com a sua não apresentação de problemas com os mínimos locais e com a sua característica de obtenção dos vetores através dos dados que apresentam maior contribuição que são chamados de vetores de suporte (SV - “*Support Vectors*”). O método apresenta outra vantagem que é a de possuir poucos parâmetros para se ajustar (CORTES; VAPNIK, 1995).

1.3 Proposta do Trabalho

1.3.1 Objetivo Geral

O objetivo do trabalho é prever os melhores resultados do comportamento mecânico de peças em corpos de prova, onde é aplicada a união por solda FSW utilizando métodos de aprendizagem de máquina para gerar modelos da curva (Tensão-Deformação). Os modelos devem apresentar comportamentos significativo e preditivo, assim, um investimento em função

da robustez dos algoritmos utilizados é necessário a fim de projetar um modelo adequado do sistema. A ideia que está por trás de obter um modelo robusto do processo FSW é encontrar a função com uma aproximação adequada para representar o sistema através das entradas e saídas desejadas, com o objetivo de ter uma segurança do valor predito sobre a curva de (Tensão-Deformação).

1.4.2 Objetivos Específicos

- Preparar e caracterizar os ensaios de tração dos corpos de prova onde foram aplicadas a união por solda FSW, para obtenção dos dados experimentais.
- Desenvolver análise estatística, mediante um planejamento fatorial e análise de variância (ANOVA), com o fim de determinar significância e fator de curvatura (não linearidade) das entradas.
- Desenvolver modelos não lineares do sistema com as respostas que apresentam curvatura na análise estatística a partir de dados experimentais de entrada e saída, obtidas do ensaio de tração de corpo de provas confeccionadas pelo processo de soldagem FSW.
- Validar os modelos feitos pelos métodos de aprendizagem de máquina dos sistemas não lineares.
- Ajustar os parâmetros do algoritmo SVM por meio do filtro de estimação, filtro de partículas (PF), obtendo os estados ocultos do SVM.
- Comparar os modelos realizados com SVM com outros métodos como regressão polinomial e RNA.

1.4 Arquitetura

Na Figura 1.1, mostra-se a arquitetura da proposta. Nela estão contidos um conjunto de procedimentos para a análise das propriedades mecânicas dos corpos de prova e um conjunto de

métodos para o desenvolvimento de modelos não lineares. Uma forma do desenvolvimento de predição, é usando aprendizagem de máquina o qual obtém um modelo por meio dos dados experimentais, já que não se tem o modelo matemático do processo.

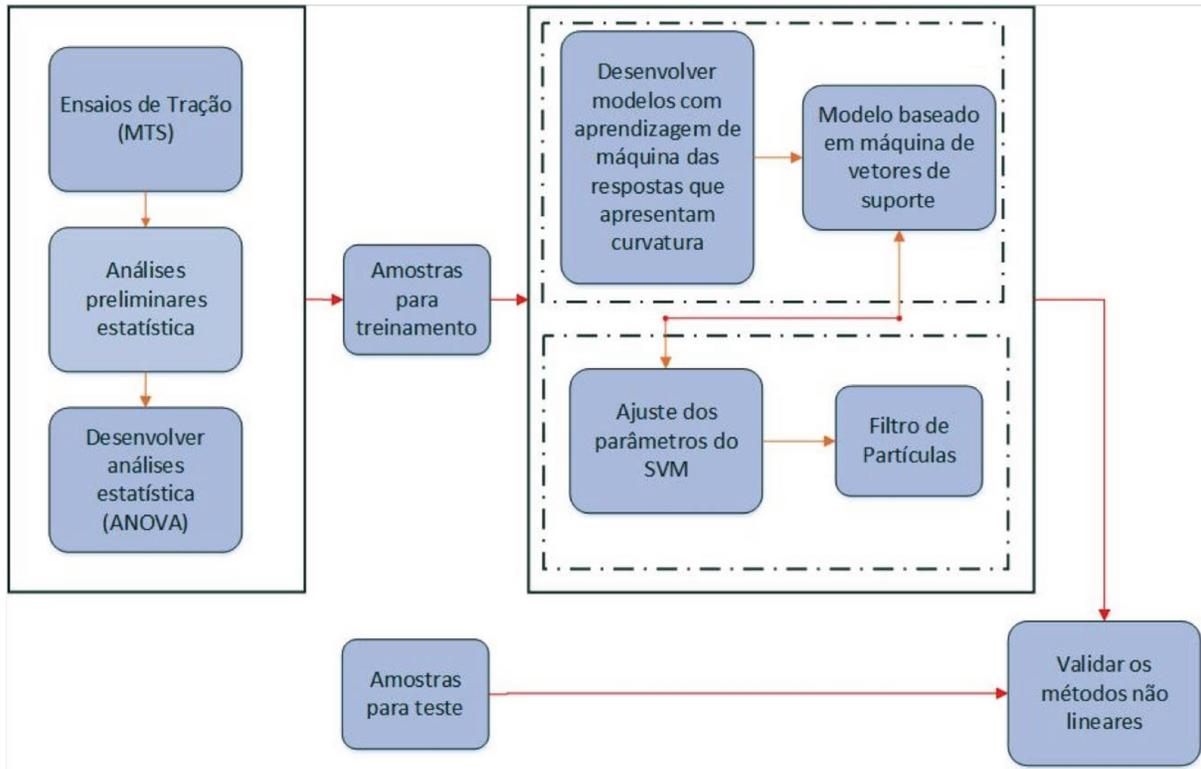


Figura 1.1 Arquitetura da Proposta.

No processo de desenvolvimento de análise estatística é possível observar o comportamento de cada parâmetro do processo sobre as respostas obtidas pelos ensaios de tração, no qual será possível observar se na faixa estudada estes parâmetros tem significância e se apresentam fator de curvatura que indicam modelos não lineares. Jayaraman et al. (2009) realizaram uma investigação experimental da velocidade de rotação da ferramenta, da velocidade axial de soldagem e da força axial através de um planejamento experimental do tipo Taguchi com o objetivo de maximizar a tensão limite de ruptura do material soldado. Um modelo matemático foi feito com base em regressão não linear dos dados obtidos usando o aplicativo computacional MINITAB.

No processo de desenvolvimento de aprendizagem de máquina é possível determinar modelos para sistemas não lineares (respostas que apresentaram curvatura) com algoritmos como o SVMr, RP e RNA (APÊNDICE F e APÊNDICE G), desta forma observar na faixa estudada que os modelos sejam preditivos mediante o coeficiente de determinação (R^2).

Embora para o SVMr precisa ajuste de alguns parâmetros, no qual será utilizado o método de validação cruzada, mas, este método tem um custo computacional elevado e não garante os parâmetros ótimos, e assim será feito o algoritmo de filtro de partículas (FP) para atingir dito problema.

Por último realizaram-se outros métodos de aprendizagem como RP e RNA para realizar um comparativo com o método SVMr, onde vão ser analisados o erro de predição (MSE), o erro percentual (MAPE), e coeficiente de determinação (R^2).

1.6 Organização da Dissertação

Esse trabalho está estruturado em seis capítulos.

No *Capítulo 1*, introdução, apresenta a justificativa do trabalho, definição do problema, os objetivos e a metodologia da pesquisa.

No *Capítulo 2* apresenta-se uma revisão bibliográfica dos principais conceitos a serem tratados nesta pesquisa. Abordam-se conceitos de “*Frition Stir Welding*”, Aprendizagem de Máquina e máquina de vetores de suporte.

No *Capítulo 3* apresenta-se os fundamentos teóricos e a modelagem matemática dos modelos utilizados, fazendo-se uma análise de benefícios ou vantagens obtidas a partir da implantação dos conceitos de otimização e propriedades dos algoritmos de aprendizagem.

No *Capítulo 4* apresenta-se a proposta de aplicação para a realização dos modelos, onde encontram-se as fases que compõem o método de Máquina de Vetores de Suporte de regressão. Além disso é mostrado uma melhora feita com o filtro de partículas para o algoritmo SVMr.

No *Capítulo 5* é mostrado um estudo do processo de FSW, onde é aplicada a proposta do capítulo anterior, visando obter a análise das informações coletadas e gerar modelos das respostas identificadas pelos ensaios de tração.

No *Capítulo 6* apresentam-se as conclusões da pesquisa, assim como sugestões para trabalhos futuros.

Capítulo 2 .

REVISÃO BIBLIOGRÁFICA

Neste capítulo apresentam-se os conceitos básicos referentes à identificação de sistemas aplicados ao processo de FSW, incluindo um estudo sobre o estado da arte relacionado com o tema de algoritmos de aprendizagem de máquina e sua respectiva evolução até a atualidade, mostrando as diferentes vantagens assim como os principais inconvenientes de cada um dos métodos. Este estudo inclui uma recompilação dos principais métodos aplicados por outros pesquisadores para tratar deste problema.

2.1 Identificação de Sistemas

Entende-se por identificação de sistemas, a determinação de um modelo que represente o comportamento de um sistema em estudo. O modelo pode ser linear, não linear, estático, dinâmico, determinístico, estocástico, dependendo do tipo do processo e aplicação que se considere. O modelo pode ser obtido através de duas formas: da análise físico-matemático “*Model Building*” conhecido como uma modelagem de caixa branca, que se obtém através da descrição do comportamento; ou da identificação experimental também conhecida como modelagem de caixa preta que consiste em construir um modelo do sistema através de medições (NORGAARD et al. 2003). Na Figura 2.1 mostra-se a representação do sistema dinâmico através de entradas e saídas.

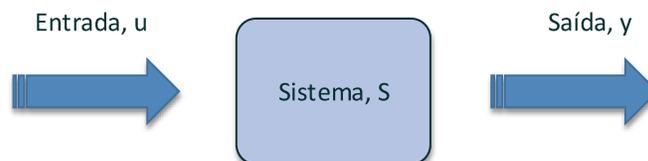


Figura 2.1 Sistema Dinâmico com Entrada e Saída.

Na maioria dos trabalhos disponíveis na literatura existente, encontram-se mais informações sobre identificação dos sistemas através de técnicas de modelagem lineares que não

lineares; as razões mais importantes para este fato são que os modelos lineares são mais simples de serem obtidos e que eles podem ser utilizados em diferentes pontos de operação do sistema.

A identificação experimental de um sistema tem como objetivo a construção de modelos, onde, a partir das medidas realizadas, obtém-se uma função que representa o sistema, função esta descrita por termos com número finito de parâmetros. Na Figura 2.2 mostram-se as etapas que se apresentam na identificação de sistemas (AGUIRRE, 2000) e (LJUNG, 1999).

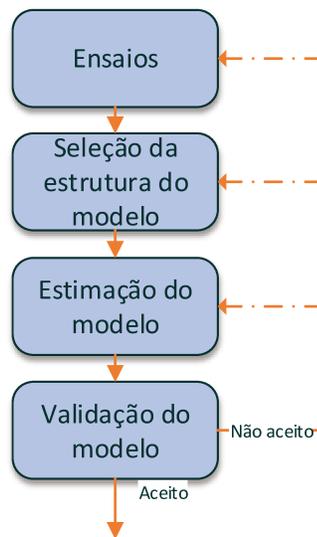


Figura 2.2 Sistema de Identificação.

Na etapa de ensaios experimentais, obtém-se os dados necessários que descrevem como é o comportamento do sistema. Para a etapa de seleção do modelo, consideram-se estruturas apropriadas como modelos de entrada/saída ou modelos do espaço de estado (ALMA et al., 2009), (LJUNG, 1999). Na etapa de estimação do modelo se constrói um modelo com o melhor desempenho de acordo com o tipo de critério de estimação adotado, e no caso dos métodos de aprendizagem de máquina (ML), o modelo estimado é obtido através de técnicas de treinamento. Depois de estimado, deve-se avaliar o modelo através dos requisitos especificados.

No caso dos métodos de aprendizagem de máquina (por exemplo, Máquina de Vetores de Suporte - SVMr), como as não linearidades presentes e a complexidade intrínseca do problema não são conhecidas, os algoritmos de otimização e as ferramentas estatísticas utilizadas para seleção de modelos podem induzir a modelos com baixa capacidade de generalização, ou seja

pouco representativos. Por isso como estes métodos ainda não são dotados de algoritmos de treinamento robustos desta maximização da capacidade de generalização de uma forma sistemática, esta etapa de validação do modelo se torna um passo muito importante.

2.2 Friction Stir Welding

Recentemente, devido ao peso, o alumínio tem sido considerado um material estrutural de “*Energy-Saving*” em aplicações avançadas. Além disso, o alumínio é um recurso facilmente reutilizável porque pode ser reciclado, por conseguinte é um material metálico amigável ao meio ambiente.

Um processo de solda que satisfaz o comportamento aceitável na resistência do material (Alumínio) nas propriedades mecânicas tem sido o “*Friction Stir Welding*” (FSW). O FSW foi desenvolvido inicialmente para ligas de alumínio, pelo Instituto de Soldadura (TWI) do Reino Unido em 1991 (DAWES et al., 1996).

O FSW é um processo de união de estado sólido. O processo FSW ocorre a uma temperatura inferior ao ponto de fusão do material da peça de trabalho e tem várias vantagens sobre o processo de solda por fusão. Algumas das vantagens do processo FSW são indicadas na lista seguinte (YOUSIF et al., 2008):

- FSW requer o mínimo material consumível.
- FSW produz microestruturas desejáveis na solda e nas zonas afetadas pelo calor.
- FSW é ambientalmente "amigável" (sem fumaça, ruídos, ou faíscas)
- FSW pode ser bem-sucedido na junção de materiais que não são unidos por métodos de soldagem por fusão.
- FSW produz menos distorção do que as técnicas de soldagem por fusão.

Em FSW utiliza-se uma ferramenta considerada não consumível composta principalmente por um cilindro e um pino concêntrico no extremo inferior, que é inserida entre os materiais que se deseja unir, aplicando uma força axial ao mesmo tempo que coloca-se um movimento de

rotação e translação. Na Figura 2.3 observa-se um esquema do processo FSW e suas principais variáveis.

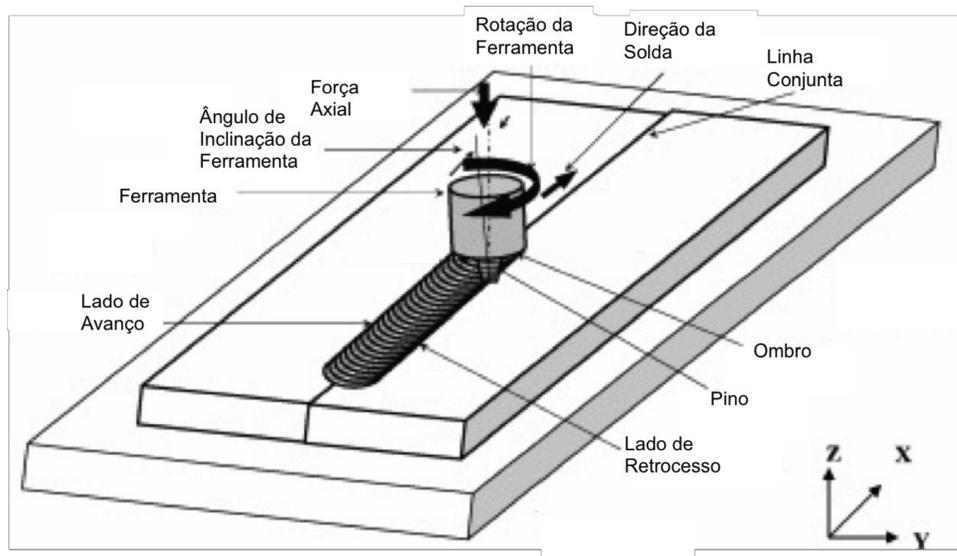


Figura 2.3 Esquema do Processo de Soldagem. Adaptado de (KUMAR AND KAILAS, 2008).

Procede-se à inserção da ferramenta até se ter o contato com a superfície do material a soldar, aplicando uma força axial controlada pelo movimento vertical. O contato provocado, gera calor devido a fricção elevando a temperatura do sistema. A rotação e o avanço da ferramenta, tem como resultado a conformação de fluxo plástico do material, que com ajuda da geometria da ferramenta, é agitado e misturado até se obter uma junção metalúrgica em fase sólida (KUMAR, KAILAS, 2008).

Em Record et al. (2007) observa-se uma investigação de nove parâmetros do processo FSW e é feita do ponto de vista experimental, usando técnicas baseadas em planejamento de experimentos. Estes autores concluem que os principais fatores de influência são a rotação da ferramenta, a velocidade de soldagem e a profundidade da solda. Desta forma, estas foram as variáveis adotadas para o desenvolvimento deste trabalho.

O trabalho de Lombard et al. (2008) investiga o efeito da rotação da ferramenta e da velocidade axial de soldagem na tensão limite de ruptura do material. Estes autores concluem que a velocidade de rotação da ferramenta é o fator que mais afeta a tensão de ruptura.

Os artigos Elangovan et al. (2008) e Elangovan et al.(2009) apresentam o efeito da rotação da ferramenta, da velocidade de soldagem, da força axial e do tipo de ferramenta na tensão limite de ruptura da junta soldada, também, através de planejamento experimental, análise da superfície de resposta e análise de variância, concluindo sobre o emprego de um conjunto de parâmetros mais adequado para o processo FSW.

2.3 Aprendizagem de Máquina

A aprendizagem de máquina “*Machine Learning*” (ML) é uma das tecnologias mais utilizadas na atualidade, uma programação por computador que busca otimizar o desempenho utilizando dados de amostras ou experiências passadas. O modelo pode ser preditivo para fazer predições, ou descritivo para ganhar conhecimento dos dados, ou inclusive preditivo e descritivo a mesmo tempo. A Figura 2.4 mostra as diferentes etapas para a obtenção do modelo, por meio de algoritmos de aprendizagem (ETHEM, 2010); (ABU-MOSTAFA,2012) e (NG, 2012).

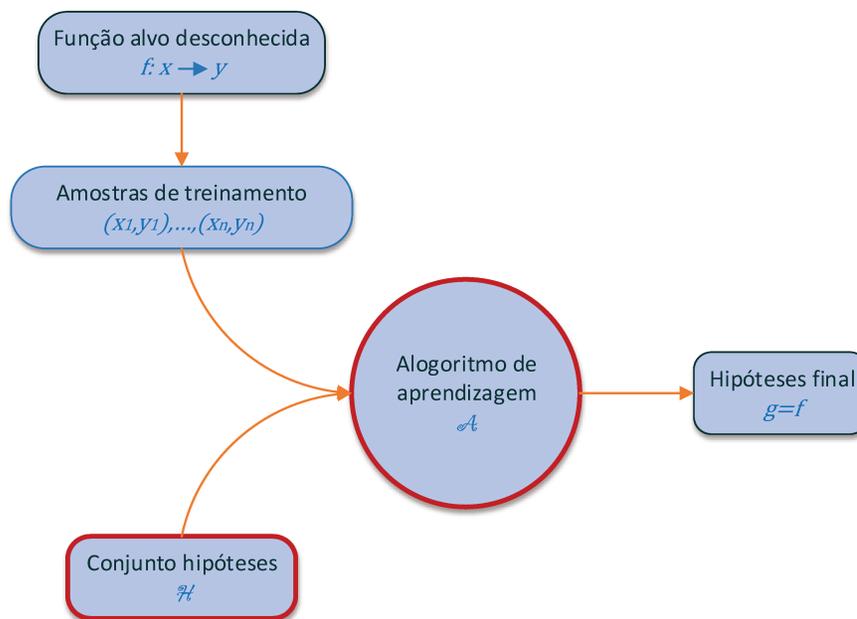


Figura 2.4 Estrutura Aprendizagem de Máquina.

Os algoritmos de aprendizagem estão baseados numa propriedade humana essencial, a capacidade de aprendizagem através de observações, que pode ser representada por sequencias de instruções implementadas em algoritmos que serão executados nos computadores utilizando as

observações como treinamento, resultando em um modelo que possibilita realizar previsões do comportamento. Portanto, dado uma quantidade finita de dados observados para o treinamento, procura-se derivar uma relação para um domínio infinito (NG, 2012).

Tabela 2.1 Exemplos de Aprendizagem de Máquina (NG, 2012).

	DESCRIÇÃO	EXEMPLOS
MINERAÇÃO DE DADOS	Com o efeito do crescimento da web e da automatização, as empresas estão coletando constantemente dados dos diferentes problemas na indústria.	<ul style="list-style-type: none"> • Fluxo de cliques na web para entender ao usuário. • Registros médicos onde é possível transformar em conhecimento, assim entender as doenças. • Campo da engenharia para ter uma solução alguns problemas (MITCHEL, 2006). • Algoritmo probabilístico que mede o cluster de onde se encontra o individuo tem o não o gene (LEE et al., 2006).
APLICAÇÕES COMPLEXAS QUE SÃO DIFÍCEIS DE SE IMAGINAR COMO PROGRAMAR À MÃO	São algoritmos que não podem ser diretamente aplicadas na solução de problemas gerais, já que não se tem os seus modelos, para isto o computador tem que aprender por ele mesmo como resolver cada problema.	<ul style="list-style-type: none"> • Helicópteros autônomos (ABBEEL et al., 2007). • Processamento de linguagem natural (NLP). • Visão computacional (MITCHELL, 1996).
PROGRAMAS AUTO-CUSTOMIZÁVEIS	São aplicativos para recomendações personalizadas, já que se tem milhões de usuários e não tem milhões de programas diferentes para cada um.	Empresas como Amazon, Netflix ou iTunes, recomendam os filmes, produtos e musica.

A aprendizagem de máquina usa a teoria estatística para a construção de modelos matemáticos, onde o objetivo principal é fazer dedução a partir das amostras. O papel da ciência da computação é baseado primeiramente na formação do modelo. Precisa-se de algoritmos eficientes para resolver problemas de otimização, bem como para armazenar e processar a enorme quantidade de dados que geralmente se têm. Em segundo lugar, uma vez que o

modelo foi treinado, sua representação e solução para dedução precisa ser eficiente. Em certas aplicações, a eficiência do algoritmo de aprendizagem ou dedução pode ser tão importante como a sua precisão preditiva.

A aprendizagem de máquina foi desenvolvida como uma nova capacidade para os computadores. Na atualidade encontra-se em muitos segmentos da indústria e ciência básica, sendo utilizada por grandes empresas como o Google ou o Microsoft (Bing) como um mecanismo de pesquisa na web, ou o Facebook e a Apple no aplicativo de fotos (NG, 2012).

Teve seu crescimento no campo da inteligência artificial (IA), já que os pesquisadores acreditam que a melhor maneira para chegar à inteligência humana é através de algoritmos de aprendizagem que tentam imitar a forma como o cérebro humano aprende. Porém foi desenvolvido em diferentes campos da indústria com a disponibilidade da biblioteca de dados. Alguns exemplos são mostrados na Tabela 2.1 (ABU-MOSTAFA,2012) e (NG, 2012).

Pode-se ver no esquema da Figura 2.5 o processo de ML, obtendo uma predição pelos dados recolhidos.



Figura 2.5 Aprendizagem de Máquina - Caixa Preta.

A ML é frequentemente utilizada tanto para fins descritivos como preditivos (IZENMAN, 2008). Entre as atividades de análise estão a busca descritiva de relações, estruturas, tendências, grupos e padrões não evidentes nos dados brutos, bem como a identificação de observações atípicas particulares “*outliers*”.

A aprendizagem de máquina está dividida em dois problemas: a classificação e a regressão. Os problemas da classificação ocorrem quando as saídas estão definidas pelas classes. O objetivo portanto é separar as amostras em diferentes classes, e para isto deve-se obter os parâmetros necessários da função para a separação das classes. Por outro lado, os problemas da

regressão têm como objetivo prever a saída mediante uma respectiva entrada no modelo, por isso, a saída do modelo tem valor contínuo (ABU-MOSTAFA,2012) e (NG, 2012).

2.3.1 Tipos de Aprendizagem de Máquina

Os tipos principais de ML são aprendizado supervisionado no qual a ideia é ensinar ao computador como realizar uma tarefa e aprendizado não supervisionado no qual a ideia é permitir que o computador aprenda por si mesmo (ETHEM, 2010). Outros tipos que não são tão utilizados são aprendizagem por reforço e sistemas de recomendação.

1. **Aprendizado Supervisionado:** Em aprendizado supervisionado, têm-se dados tanto de entrada como de saída. Além disso, existem problemas de regressão ou classificação. As distintas técnicas utilizadas nesta aprendizagem podem ser classificadas de acordo com a sua abordagem, seja linear ou não linear, paramétrica ou não paramétrica (ETHEM, 2010) e (ABU-MOSTAFA,2012). Segundo eles, em geral, as variáveis de entrada são conhecidas também como variáveis explicativas, independentes ou preditas, e as variáveis de saída são conhecidas como variáveis de resposta ou dependentes.
 - **Algoritmos de regressão:** Nestes problemas, o algoritmo tenta localizar uma função das variáveis de entrada para aproximar as saídas conhecidas. Como exemplo de regressão, a ideia é prever uma saída y_p em função de x_d , tem-se um conjunto de dados que são mostrados na Figura 2.6, onde a saída se encontra no eixo y e a entrada no eixo x . O objetivo é prever a saída desejada com respeito a uma entrada determinada. Refere-se ao fato que nós damos ao algoritmo um conjunto de dados nos quais as “respostas certas” foram obtidas.

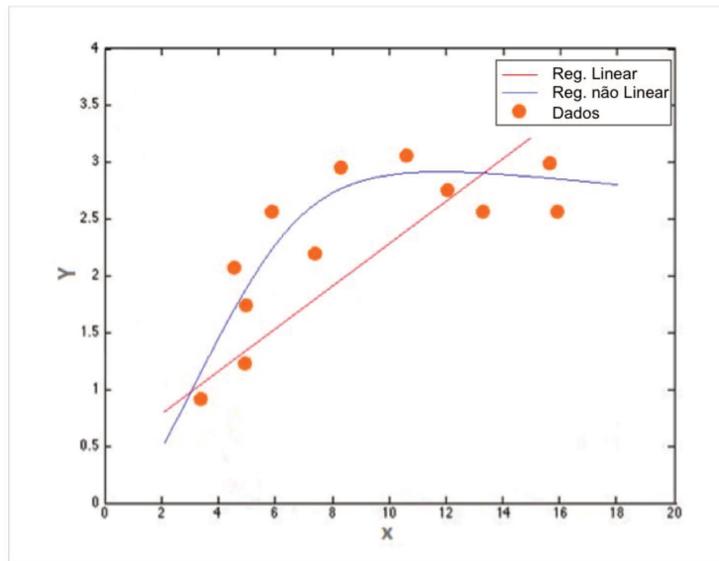


Figura 2.6 Exemplo de aprendizagem Supervisionado Regressão.

- Algoritmos de classificação: Como exemplo de classificação, a ideia é prever que tipo de classe. O eixo y e x são características do modelo. Os dados são mostrados na Figura 2.7. Este exemplo é um problema que apresenta duas classes, embora é possível aplicar o algoritmo para um número infinito de classes (ABU-MOSTAFA,2012).

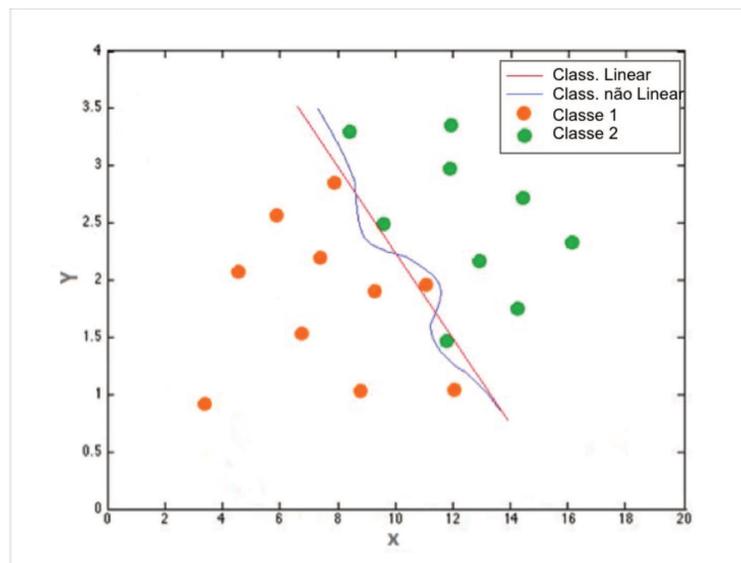


Figura 2.7 Exemplo de Aprendizagem Supervisionado de Classificação.

2. **Aprendizado Não Supervisionado:** Em aprendizado não supervisionado, os dados não são etiquetados, já que não se conhece sua saída. O que se quer com este algoritmo é encontrar algum padrão nos dados (ETHEM, 2010) e (NG, 2012).

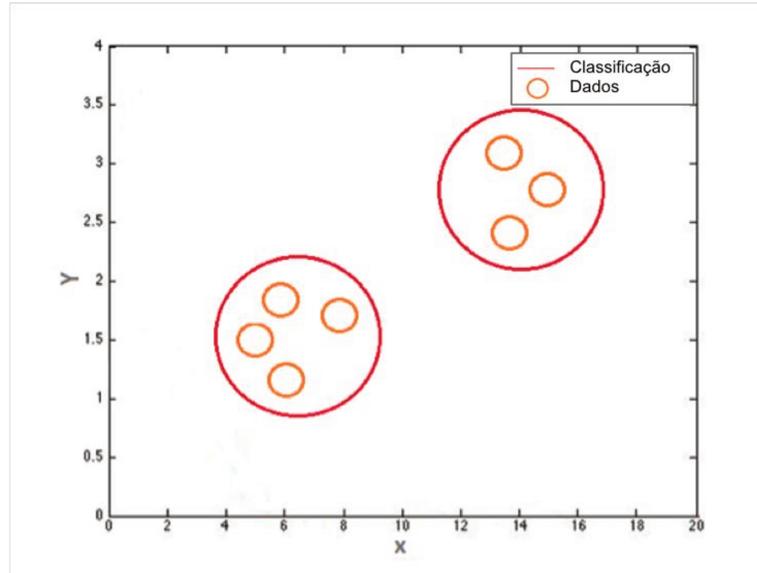


Figura 2.8 Exemplo de Aprendizagem não Supervisionado.

Tem-se uma estrutura para o espaço de entrada de tal forma que certos padrões se obtêm com mais frequência que outros, e se deseja ver o que geralmente acontece. O objetivo do algoritmo é encontrar clusters, onde cada cluster está conformado por um grupo de dados que são diferentes ao restante. Na Figura 2.8 podem-se observar um exemplo com dois clusters.

3. **Aprendizagem de reforço:** O algoritmo aprende observando o mundo ao seu redor. Sua informação é dada por retroalimentação “*feedback*” que se obtém do mundo exterior, em resposta as suas ações. Portanto, o sistema aprende baseado nas práticas de tentativa e erro.

2.3.2 Histórico Evolutivo da Aprendizagem de Máquina

Mesmo entre os pesquisadores de aprendizagem de máquina não existe uma definição bem acertada sobre o que é o ML. Algumas definições são:

- Arthur Samuel (1959) “*Machine Learning: É o campo de estudo que dos computadores as habilidades de aprender algo sobre o que não tem sido*

explicitamente programado” (ARTHUR, 1954; RETRIEVED 2011). Foi pioneiro no campo da aprendizagem de máquina trabalhando com jogo de damas, onde o computador aprendeu jogar melhor que ele.

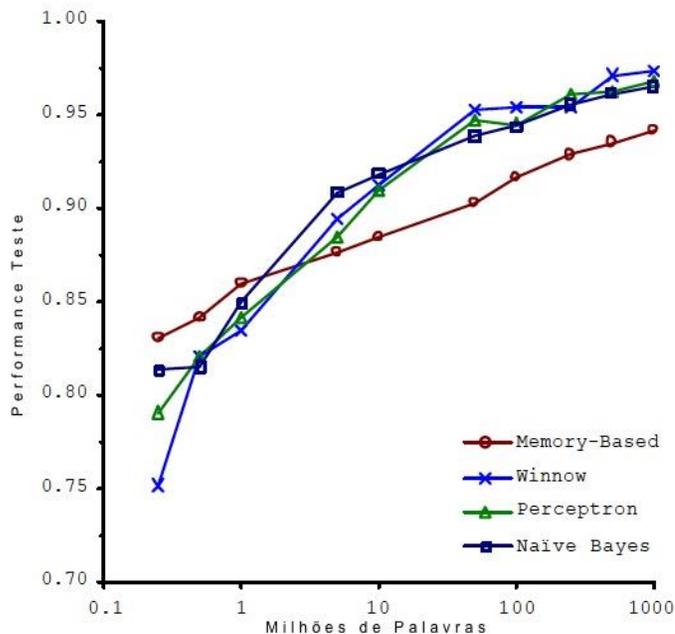
- Tom Mitchell (1998) *“Machine Learning: Um problema bem definido de aprendizagem é definido como um programa de computador aprende a partir de uma experiência E com respeito a alguma classe de tarefas T e medida de performance P, “SE” sua performance nas tarefas T, medida por P, “MELHORA” com a experiência E”* (MITCHELL, 2006). Por exemplo, no caso de jogo de damas (ARTHUR, 1954; RETRIEVED, 2011). Tarefa T: jogar damas. Medida de performance P: Percentagem de jogos ganhos contra oponente. Experiência de treinamento E: jogando contra si mesmo.

Havia uma percepção de que a ML funcionava só para problemas descritos anteriormente na Tabela 2.1 (pesquisa na web, marcações em fotos ou Antispam de e-mail). Porém, a ML foi desenvolvida como uma nova capacidade para os computadores e hoje faz parte de muitos segmentos da indústria e ciência básica. Na atualidade os algoritmos de aprendizagem estão em campos como aviação, biologia computacional, sistemas de computadores, robôs autônomos e no campo da engenharia (ETHEM, 2010) e (NG, 2012).

Com muitos algoritmos de aprendizagem de máquina, é difícil dizer qual deles apresenta uma melhor performance. Isto ainda se torna mais difícil quando o número de dados de aprendizagem é muito grande, pois todos os algoritmos apresentam uma performance similar segundo o estudo realizado por (BANKO, BRILL, 2001).

Os pesquisadores Banko e Brill (2001) estudaram o efeito de diferentes algoritmos com respeito a quantidade de dados. O problema escolhido foi de classificação de palavras. Utilizaram algoritmos como Perceptron, Winnow, Memory-based e Naive Bayes, algoritmos considerados parte do estudo da arte em 2001. Eles utilizaram diferentes tamanhos de conjuntos de treinamentos aplicados nos algoritmos como se mostra na Figura 2.9.

Sistema de Aprendizagem com Performance Elevada



*E.g. Classificação entre palavras confusas. {to, two, too}
For breakfast I ate two eggs.*

Figura 2.9 Curvas de Aprendizagem de Desambiguação para Conjunto de Confusão.
Adaptado de (BANKO AND BRILL, 2001).

Eles chegaram à conclusão, que realizando o treinamento em todos os métodos analisados com um número grande de amostras, os métodos atingem uma performance similar.

2.3.3 Histórico Evolutivo dos Algoritmos de Aprendizagem de Máquina

No começo dos algoritmos de ML, a ideia era tentar construir algoritmos que realizavam tarefas sem precisar de informações do sistema. Assim, o principal algoritmo de aprendizagem naquela época foi o de Regressão Logística (RL) (APÊNDICE D), onde aplicando polinômios era possível deduzir as saídas do sistema para uma determinada entrada. Na procura para obter um algoritmo mais complexo em problemas não lineares foram desenvolvidas as Redes Neurais Artificiais “*Artificial Neural Network*” (RNA), na qual foi originalmente motivado com a meta de se obter uma máquina que pudesse imitar o cérebro humano.

A RNA foi desenvolvida simulando as redes neurais do cérebro. Na Figura 2.10 (a) mostra-se um neurônio normal, indicando onde se encontra o seu núcleo. Além disso, são mostradas entradas ou conexões chamadas dendrites que recebem informações de outros neurônios, e a única saída chamada de “Axon”, a qual envia um sinal ou mensagem a outros neurônios (NG, 2012).

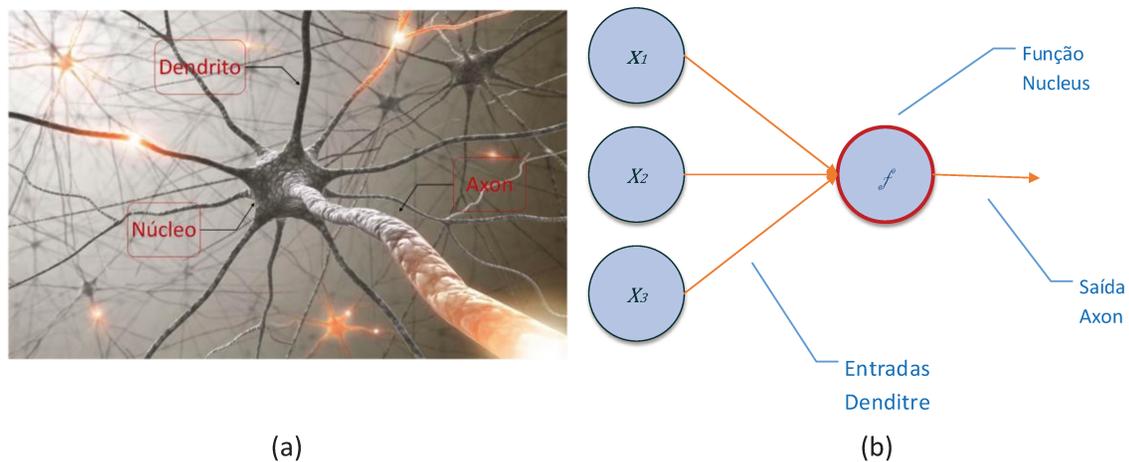


Figura 2.10 Neurônio em Comparação com Modelo do RNA. (a) Neurônio Real Adaptado de [FEUERSTEIN et al.,2013]. (b) Neurônio Matemático.

Como comparação também é mostrado na Figura 2.10 (b) o modelo simples de rede neural, o círculo vermelho representa o corpo do neurônio, as entradas x representam os dendritos do neurônio, e por último a saída $h_{\theta}(x)$ representa o fio “Axon”. Na Figura 2.11 mostra-se um grupo de neurônios, onde a comunicação entre eles é por meio de um pulso de eletricidade chamado “Spike”. O neurônio envia a mensagem pelo “Axon”, o qual é recebido por algum dendrite de outro neurônio (NATIONAL INTITUTE ON AGING, 2008) e (NG, 2012).

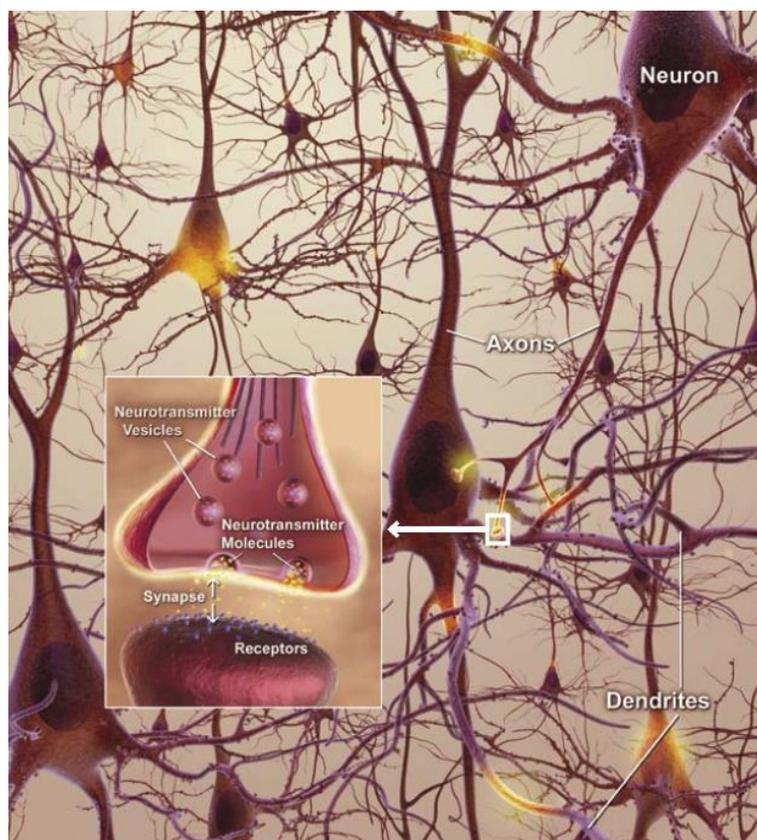


Figura 2.11 Comunicação dos Neurônios no Cérebro. (NATIONAL INTITUTE ON AGING, 2008).

No começo do 1960s pesquisou-se a primeira geração de redes neurais (Perceptron). Porém, o método só foi muito utilizado na década de 80, embora a popularidade diminuiu ao final da década de 90. Embora teve uma ressurreição com o surgimento dos computadores, devido ao fato de que as redes neurais demandam um grande custo computacional, e portanto na atualidade os computadores permitem desenvolver as redes neurais eficientemente, ajudando na utilização de muitas aplicações.

As redes neurais são atualmente uma ideia antiga que havia caído em desuso por um tempo. Porém hoje em dia é a técnica de estado de arte para muitos diferentes problemas de aprendizagem de máquina. Na Tabela 2.2 se mostra uma breve história da evolução deste método. Uma das aplicações mais comuns das redes neurais, tem sido na identificação da escrita a mão, com o fim de obter entendimento de qualquer figura desenhada seja número ou letra, por meio de um algoritmo do computador sem a intervenção humana.

Tabela 2.2 Breve Historia das Redes Neurais. Baseado em (EBERHART, 1990) e (EBERHART et al., 2007).

ANO	AUTORES	CONTRIBUÇÃO
1943	W. McCulloch e W.Pitts	Neurônio Formal.
1949	Donald Hebb	Aprendizagem Hebbiano.
1951 e 56	J. von Neuman	Atomata e Neurônio Computacional Redundante.
1957 e 62	F. Rosenblatt	Regra Delta e Perceptron.
1962	B. Widrow e M. Hoff	ADALINE e MADALINE
1963	Winograd e Cowan	Processo Distribuído dos Parâmetros.
1969	M. Minsky e S. Papert	Crítica ao Perceptron
1971	T. Kohonen	Memória Associativa
1977	J. Anderson	Autoassociação
1977 e 79	K. Fukushima	Cognitron e Nocognitron
1981	J.J. Hopfield	Redes Hopfield
1982	T. Kohonen	Self Organizing Map (SOM)
1986	Rumelhart, Hinton e Williams	Regra Delta Generalizada, Backpropagation

Foram desenvolvidas aplicações de reconhecimento online de palavras manuscritas em estilos mistos com restrições, onde as palavras são representadas com imagens de baixa resolução e cada pixel foi inserido como uma entrada da rede, a qual apresenta informações da curvatura da linha e da direção da trajetória (LeCUN AND BENGIO, 1994). Na classificação de números reais de endereços de casas foi desenvolvida uma rede neural de convolução (SERMANET et al., 2012).

A máquina de vetores de suporte (SVM), está sendo utilizada tanto na indústria, como na academia. Em comparação aos algoritmos de Regressão Polinomial e Rede Neural, é mais potente para aprendizagem de funções complexas não lineares (RYCHETSKY,2001).

O SVM é um método de aprendizagem introduzido por Vapnik, Golowich Smola (1997). Na Tabela 2.3 se mostra uma breve história da evolução deste método. A máquina propõe realizar

uma projeção geralmente não linear dos dados de entrada a um espaço de características de alta dimensão. Neste espaço projetado toma-se uma superfície de decisão (um hiperplano) que maximiza a distância das duas classes ao hiperplano (VAPNIK et al, 1997).

Tabela 2.3 Breve História de SVM. Baseado em (BOSER, 1992; VAPNIK, 1999; SHAWE-TAYLOR , CRISTIANINI ,2000).

ANTES DO ALGORITMO SVM		
ANO	AUTORES	CONTRIBUIÇÃO
1936	R.A. Fisher	Reconhecimento de Padrões.
1950	Aronszajn	Teoria de Reprodução de Kernels.
1964	Vapnik e Chervonenkis	Generalized Portrait Algorithm
1973	Duda e Hart	Margem de Hiperplano no Espaço de Entrada
1974	Vapnik e Chervonenkis	Statistical Learning Theory
1990	M. Minsky e S. Papert	Uso dos Kernels
1992	Boser, Guyon e Vapnik	Classificação de Margem Ótima
COMEÇO DO ALGORITMO SVM		
1995	Corte e Vapnik	Soft Margem Classificação (SVM)
1995	Vapnik	Soft Margem Regressão (SVM)
1998	Bartlett; Shawe-Taylo et al.	Generalisation of Hard Margin (SVM)
2000	Shave-Taylor e Cristianini	Generalisation of Soft Margem, Regression Case (SVM)

Existem pesquisas relacionadas às aplicações na área da engenharia, especialmente RNA e SVM, tais como previsão de processos de fabricação, monitoramento e controle. Para este trabalho os métodos são utilizados para predição de variáveis no processo (BEZAZI et al, 2006).

O SVM apresenta vantagem com relação a outros métodos de aprendizagem já que não precisa escolher a arquitetura do modelo e não apresenta problemas com os mínimos locais. O maior problema do método SVM é relativo a fase do treinamento, que exige um grande custo computacional para conseguir a identificação de sistemas não lineares em tempo real “*on-line*”. Na literatura pode-se encontrar algumas técnicas para esta classe de aplicação, como por

exemplo, os pesquisadores Collobert e Bengio (2001), mostraram que é possível descompor o problema em subproblemas mais simples, ou o pesquisador Martin (2002), criou um algoritmo recursivo para o treinamento, encontrando condições apropriadas para a atualização dos coeficientes. Para o presente projeto se realizará a identificação do sistema em modo “*off-line*”.

Capítulo 3 .

FUNDAMENTOS TEÓRICOS

Neste capítulo é descrito com maior profundidade os conceitos teóricos relacionados com os diferentes métodos de realização de modelos desenvolvidos nesta dissertação. Na primeira parte se detalha os fundamentos para realizar aprendizagem de máquina. Seguidamente será descrito o algoritmo da máquina de vetores de suporte. Finalmente, também detalham-se outras técnicas que também foram necessárias como a regulação para generalizar o modelo, e a escolha do modelo analisada pelas curvas de aprendizagem.

3.1 Fundamentos para Realizar Aprendizagem de Máquina

3.1.1 Escalonando Características

A ideia principal é que as características ou entradas estejam em uma escala similar.

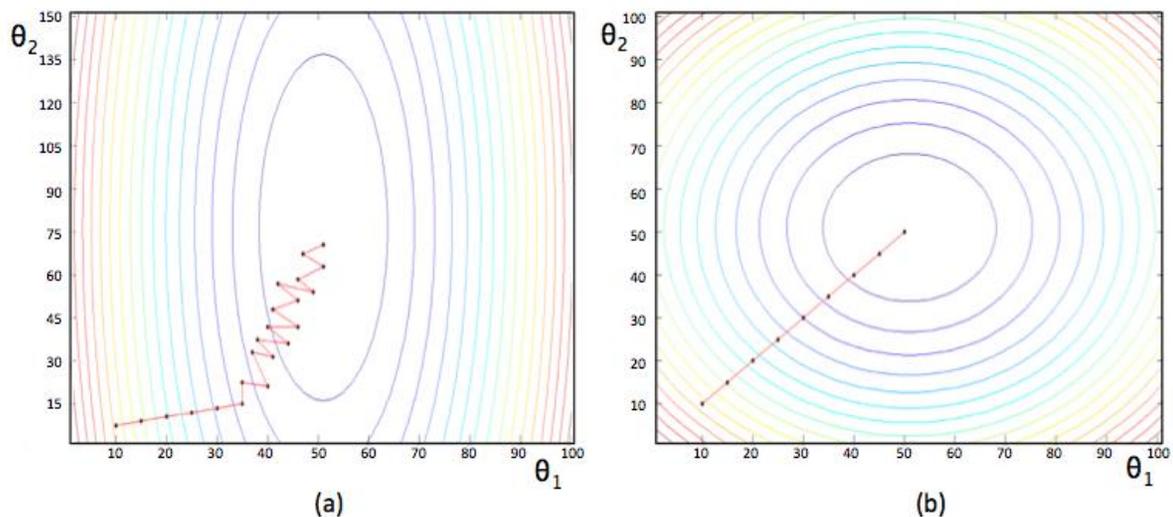


Figura 3.1 Procura do Mínimo da Função: (a) Entradas Escalonadas; (b) Entradas sem Escalar.

Como resultado tem-se um algoritmo de otimização mais rápido. Portanto, se existe uma característica x_1 com variação grande e x_2 com variação muito menor, a função de custo fica com eclipses alongados, já que há muita diferença entre os eixos. Por conseguinte o algoritmo demora em encontrar o mínimo (Refere-se Figura 3.1). Dessa forma, se escalamos os dados, o algoritmo de otimização pode convergir mais rápido (CHERKASSKY, 2007).

O escalonamento das características variam de $-1 \leq x_j \leq 1$. A equação utilizada para o escalonamento é mostrada na Equação (3.1) (NG, 2012),

Tabela 3.1 Parâmetros para Escalonar Características.

Parâmetro	Descrição
μ	<i>média dos dados.</i>
s	<i>desvio padrão dos dados.</i>

$$x_j = \frac{(x_j - \mu)}{s}. \quad (3.1)$$

3.1.2 Desenvolvimento dos Parâmetros

Na maioria dos problemas matemáticos com uma quantidade consideravelmente grande de dados, trabalha-se com vetores, e como resultado tem-se um desempenho melhor e facilidade na utilização de operações.

Os parâmetros θ estão comprimidos em um único vetor, o qual ajuda com que as operações sejam mais rápidas. Embora, para alguns casos como por exemplo, encontrar os gradientes ou a função de custo na RNA não é tão fácil com um único vetor, recomenda-se separar a matriz dos parâmetros θ para realização de operações, neste caso em vetores por cada camada que conformam a rede. Posteriormente a obtenção dos gradientes e da função de custo, comprime-se os parâmetros θ em um único vetor para utilizar o algoritmo de otimização (NG, 2012).

3.2 Algoritmo de Máquina de Vetores de Suporte

As SVM foram desenvolvidas usando ferramentas estatísticas de otimização e das máquinas de aprendizagem. Para o caso de classificação de variáveis discretas, a estrutura das SVM só depende de um parâmetro de penalização C na função de custo e dos parâmetros que tem na função do “kernel”; para o caso de estimação de variáveis contínua (regressão) tem-se adicionalmente um parâmetro de controle que define a margem de aceitação da estimação (ABU-MOSTAFA,2012).

3.2.1 Definição Matemática dos Vetores de Suporte

A partir da função de custo de regressão logística na Equação (3.2) (APÊNDICE D), é possível definir uma nova função de custo para o SVM, como será descrito a seguir (NG, 2012),

$$J(\theta) = -y \log(h_{\theta}(x)) - (1 - y) \log(1 - h_{\theta}(x))$$
$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}, \quad (3.2)$$

portanto, quando o valor de saída y é igual a 1, a função de custo simplifica-se conforme mostrado na Equação (3.3),

$$J(\theta) = -\log\left(\frac{1}{1 + e^{-\theta^T x}}\right), \quad (3.3)$$

e da mesma forma, quando o valor de saída y é igual a 0, a função de custo simplifica-se conforme mostrado na Equação (3.4),

$$J(\theta) = -\log\left(1 - \frac{1}{1 + e^{-\theta^T x}}\right). \quad (3.4)$$

A Figura 3.2 mostra na forma gráfica a função da Equação (3.3) e a da Equação (3.4) com respeito à função $z = \theta^T x$, onde é possível observar que a primeira figura quando z tem um valor grande, a função custo $J(\theta)$ vai ter um valor pequeno, demonstrando uma contribuição pequena da função de custo para valores grandes de z . De forma contrária, pode-se observar na segunda

figura que quando agora z tem um valor pequeno, a $J(\theta)$ vai ter um valor pequeno, demonstrando uma contribuição pequena da função de custo para valores pequenos de z .

O algoritmo SVM, tem uma modificação na hipótese $h_{\theta}(x)$ e a função de custo $J(\theta)$ da Equação (3.2) (refere-se APÊNDICE D). Quando $y = 1$, a função de custo a partir do $z \geq 1$ vai ter um valor zero.

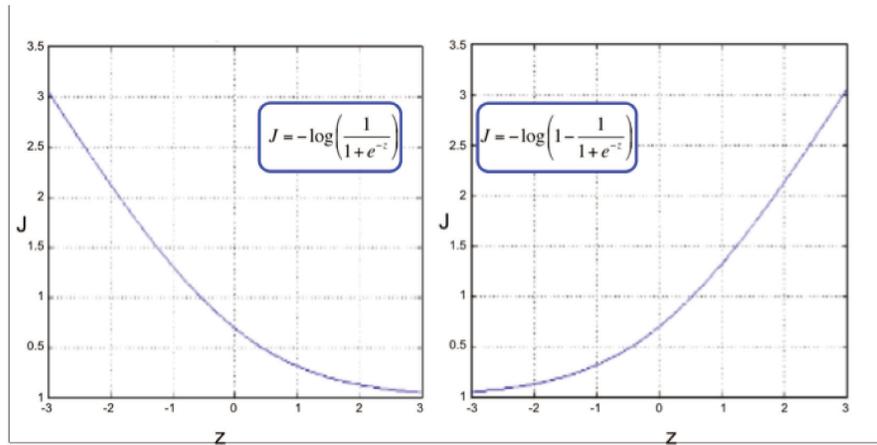


Figura 3.2 Função de Custo da Regressão Logística.

Semelhantemente, quando $y = 0$, a função de custo a partir $z \leq -1$ vai ter um valor de zero, ou seja, o valor da inclinação nessas faixas vai ser zero, sendo representado por dois segmentos de reta, como mostrado na Figura 3.3. Esta modificação acarreta uma vantagem computacional no tempo de solução do problema de otimização (NG, 2012).

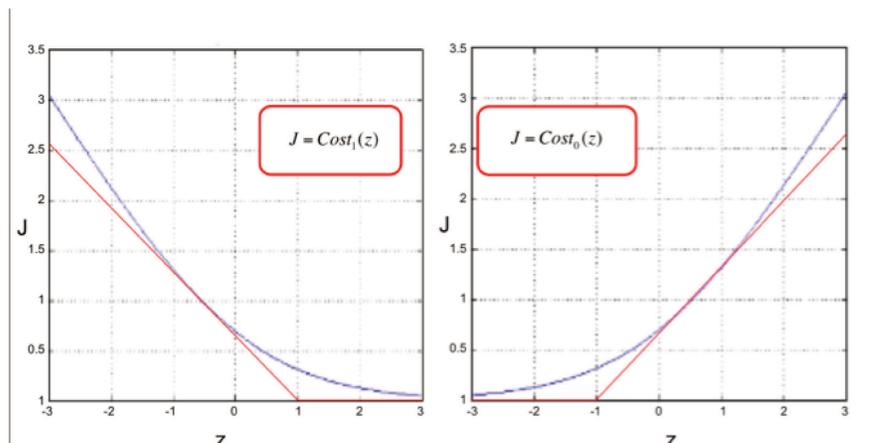


Figura 3.3 Função de Custo do (SVM) versus Regressão Logística.

As novas funções desenhadas na Figura 3.3 vão ser chamadas $Cost_1(z)$ e $Cost_0(z)$, respectivamente. Substituindo na Equação (3.2) a funções de $Cost_1(z)$ e $Cost_0(z)$ obtemos,

$$J(\theta) = y Cost_1(\theta^T x) + (1 - y)Cost_0(\theta^T x), \quad (3.5)$$

e realizando a minimização da função da Equação (3.5) e colocando a regularização (penalização dos parâmetros que não ajudam à função), tem-se,

$$\underbrace{\text{minimize}}_{\theta_0, \theta_1, \dots, \theta_n} \frac{1}{m} \sum_{i=1}^m [y^{(i)} Cost_1(\theta^T x^{(i)}) + (1 - y^{(i)})Cost_0(\theta^T x^{(i)})] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2. \quad (3.6)$$

Observa-se que é possível o cancelamento do termo $1/m$ por consequência de ser uma constante que não muda o resultado da minimização obtendo os mesmos parâmetros θ . Além disso, o termo λ não existe no algoritmo SVM, já que é substituído pelo termo C que multiplica a função de custo sem o termo de regularização. Portanto, C realiza a mesma função de λ , ou seja, realizando uma penalização à função para todos os dados de validação obtendo assim um melhor desempenho nos dados de teste. Se C fosse muito maior, a fronteira de decisão teria a informação de todos os dados, podendo acontecer problemas de sobre ajuste. Pelo contrário, se fosse muito pequeno, poderia ter problemas de baixo ajuste,

$$\underbrace{\text{minimize}}_{\theta_0, \theta_1, \dots, \theta_n} C \sum_{i=1}^m [y^{(i)} Cost_1(\theta^T x^{(i)}) + (1 - y^{(i)})Cost_0(\theta^T x^{(i)})] + \frac{1}{2} \sum_{j=1}^n \theta_j^2. \quad (3.7)$$

3.2.2 Margem da Fronteira

A ideia principal de uma máquina de vetores de suporte é construir um hiperplano como superfície de decisão (Margem da Fronteira), de forma que a margem de separação entre os exemplos seja máxima. Para isto realiza-se mudanças na Equação (3.2) (nas funções de custo e hipótese) para o SVM, conforme será descrito a seguir.

Como é possível observar em outros algoritmos como regressão logística ou rede neural (APÊNDICE D), quando é predito um valor na Equação (3.2), tem-se que escolher $y = 1$ se $\theta^T x \geq 0$, ou $y = 0$ se $\theta^T x \leq 0$. Por outro lado se forem utilizadas funções $Cost_1$ e $Cost_0$, no algoritmo SVM, apresenta mais certeza na predição, atribuindo $y = 1$ se $\theta^T x \geq 1$, ou $y = 0$ se $\theta^T x \leq -1$.

Quando se tem o problema de otimização (APÊNDICE E), minimiza-se a função de custo com respeito ao θ o qual acarreta no cancelamento do termo da primeira somatória da Equação (3.7). Observa-se que isto acontece por causa da forma das funções de Cost (Figura 3.3), já que as funções Cost possuem um valor de zero no intervalo de -1 a 1, e como resultado, obteve-se a melhor fronteira de decisão que separam os dados (ABU-MOSTAFA,2012).

Substituindo os valores das funções Cost na Equação (3.5), chega-se a função de minimização da Equação (3.5),

$$\begin{aligned} & \underset{\theta_0, \theta_1, \dots, \theta_n}{\text{minimize}} C(0) + \frac{1}{2} \sum_{i=1}^n \theta_j^2 \\ & \text{restrições} \begin{cases} \theta^T x^{(i)} \geq 1 & \text{se } y^{(i)} = 1 \\ \theta^T x^{(i)} \leq -1 & \text{se } y^{(i)} = 0 \end{cases} \end{aligned} \quad (3.8)$$

onde, pode-se visualizar a somatória da Equação (3.8) que adotou-se como termo $\theta_0 = 0$ e características $n = 2$, como restrições na função. Com estes parâmetros e resolvendo a Equação (3.8) é possível obter a norma euclidiana de θ , assim, obtém-se,

$$\begin{aligned} & \underset{\theta_0, \theta_1, \dots, \theta_n}{\text{minimize}} \frac{1}{2} \sum_{i=1}^n \theta_j^2 = \frac{1}{2} (\theta_1^2 + \theta_2^2) = \frac{1}{2} \left(\sqrt{\theta_1^2 + \theta_2^2} \right)^2 = \frac{1}{2} \|\theta\|^2 \\ & \text{restrições} \begin{cases} \theta^T x^{(i)} \geq 1 & \text{se } y^{(i)} = 1 \\ \theta^T x^{(i)} \leq -1 & \text{se } y^{(i)} = 0 \\ \theta_0 = 0 \end{cases} \end{aligned} \quad (3.9)$$

e reescrevendo a Equação (3.9) em termos de projeção e norma de vetores, como mostrado na Figura 3.4, chega-se a Equação (3.10).

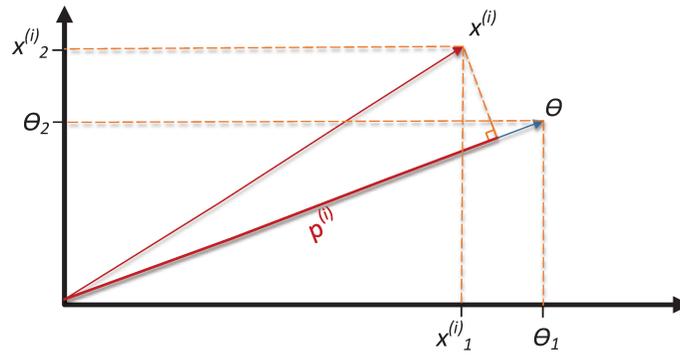


Figura 3.4 Exemplo de uma Amostra de Treinamento pelos Parâmetros θ .

$$\begin{aligned}
 \theta^T x^{(i)} &= p \|\theta\| = \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} \\
 \underset{\theta_0, \theta_1, \dots, \theta_n}{\text{minimize}} & \frac{1}{2} \sum_{i=1}^n \theta_j^2 = \frac{1}{2} \|\theta\|^2 \\
 \text{restrições} & \begin{cases} p^{(i)} \|\theta\| \geq 1 & \text{se } y^{(i)} = 1 \\ p^{(i)} \|\theta\| \leq -1 & \text{se } y^{(i)} = 0 \\ \theta_0 = 0 \end{cases}
 \end{aligned} \tag{3.10}$$

Obtido a Equação (3.10) em termos de projeção e norma dos vetores, é possível visualizar uma explicação gráfica da fronteira de decisão do algoritmo SVM. Na Figura 3.5, observa-se um exemplo de amostras de treinamentos separados pelo algoritmo SVM (NG, 2012).

Para um grupo de dados de treinamento de tamanho n composto por pares atributo-etiqueta, ou seja, característica do dado e característica da classe, (x_i, y_i) com $1 \leq i \leq n$, onde $x_i \in R$ e $y_i \in \{-1, 1\}$, se deseja obter a equação do hiperplano que separe o grupo de treinamento, sendo que aqueles pontos com igual etiqueta encontram-se a um mesmo lado do hiperplano. O termo dado b é chamado de bias, ou seja, θ_0 (NG, 2012).

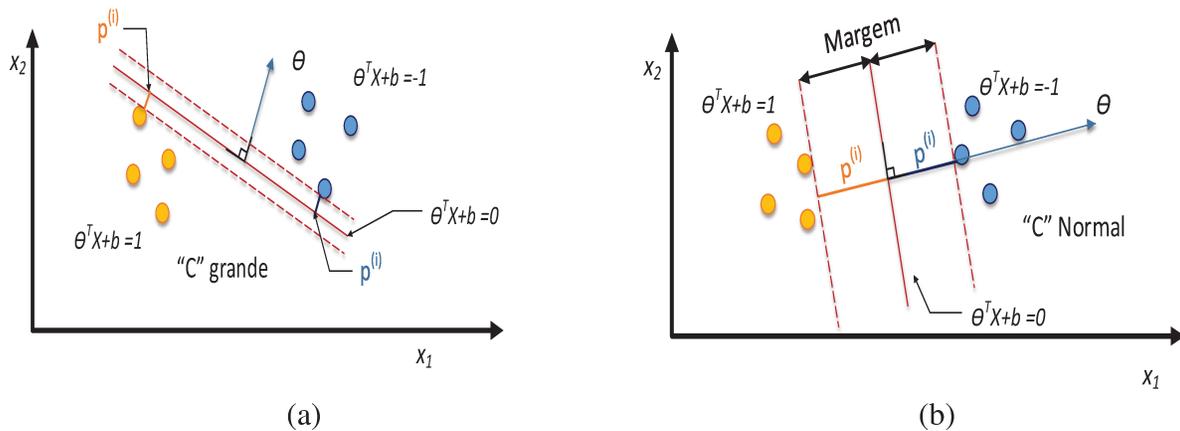


Figura 3.5 Exemplos de Classificação, Comportamento da Margem.

Se o SVM escolhe a fronteira de decisão mostrada na Figura 3.5 (a), percebe-se que não é muito boa, já que não é um algoritmo generalizado. O vetor θ vai estar 90 graus da fronteira de decisão. Observa-se que qualquer amostra $x^{(i)}$, a $p^{(i)}$ do vetor é muito pequena, por conseguinte $\|\theta\|$ tem que ser muito grande para cumprir as condições das restrições da Equação (3.10). Pelo contrário, se o SVM escolhe a fronteira de decisão mostrada na Figura 3.5 (b), pode-se observar que a $p^{(i)}$ das amostras são maiores, ajudando para que $\|\theta\|$ seja menor, contribuindo para com o objetivo da otimização.

O problema global equivale a calcular da Equação (3.10) encontrando um hiperplano $f(x^{(i)}) = \langle \theta, x^{(i)} \rangle + b$ definido por $\theta \in \mathfrak{R}^n$ e $\theta_0 = b \in \mathfrak{R}$, conforme escrito na Equação (3.11),

$$\underbrace{\text{minimize}}_{\theta, b} \frac{1}{2} \|\theta\|^2 \quad . \quad (3.11)$$

$$y^{(i)} f(x^{(i)}) \geq 1 \text{ Com } i (1, n)$$

Este problema pode ser resolvido utilizando os multiplicadores de Lagrange $\alpha^{(i)}$. Esta formulação de Lagrange permite resolver um problema de otimização, com uma série de restrições como é o caso do SVM (Equação (3.11)), mediante a introdução de novas variáveis, $\alpha^{(i)}$. O valor de cada um dos θ é obtido pelos multiplicadores de Lagrange. A ideia é subtrair da função de custo dada pela Equação (3.11) pelo produto do multiplicador de Lagrange pelas restrições dadas também pela Equação (3.11), obtendo como resultado a Equação (3.12),

$$\begin{aligned}
L &= \frac{1}{2} \|\theta\|^2 - \sum_{i=1}^n \alpha^{(i)} (y^{(i)} f(x^{(i)}) - 1) \\
&= \frac{1}{2} \|\theta\|^2 - \sum_{i=1}^n \alpha^{(i)} y^{(i)} f(x^{(i)}) - \sum_{i=1}^n \alpha^{(i)},
\end{aligned} \tag{3.12}$$

onde, o termo da somatória subtraída são os dados com classificação errada (seção 3.2.4). Obtido L , é necessário minimizá-lo com respeito as variáveis θ e b , além disso, simultaneamente deve-se igualar as derivadas de L a zero, com respeito as variáveis θ e b , ou seja os gradientes, conforme mostrado na Equação (3.13),

$$\frac{\partial}{\partial \theta} L(\theta, b, \alpha) = 0 \quad \left\{ \begin{array}{l} \theta = \sum_{i=1}^n \alpha^{(i)} y^{(i)} x^{(i)} \\ \sum_{i=1}^n \alpha^{(i)} y^{(i)} = 0 \end{array} \right. , \tag{3.13}$$

e substituindo a Equação (3.13) na Equação (3.12), obtém-se,

$$\begin{aligned}
&\underset{\alpha}{\text{minimize}} \frac{1}{2} \sum_{i,j} \alpha^{(i)} \alpha^{(j)} y^{(i)} y^{(j)} \langle x^{(i)}, x^{(j)} \rangle - \sum_{i=1}^n \alpha^{(i)} \\
&\text{sendo } \sum_{i=1}^n \alpha^{(i)} y^{(i)} = 0, \\
&\alpha^{(i)} \geq 0 \quad \forall \text{ os } i
\end{aligned} \tag{3.14}$$

assim, a Equação (3.14) obtida é um problema de otimização quadrática (APÊNDICE E). Finalmente o hiperplano de decisão obtido pode ser escrito,

$$f(x) = \sum_{i=1}^n \alpha^{(i)} y^{(i)} \langle x^{(i)}, x \rangle + b, \tag{3.15}$$

onde b é o multiplicador de Lagrange associado à restrição $\sum_{i=1}^n \alpha^{(i)} y^{(i)} = 0$ do problema dual. O SVM obtém a linha com maior distância de qualquer um dos pontos. A distância obtida é chamada de margem, a qual tenta separar os dados o máximo possível.

3.2.3 Caso não Linear: Kernels

Os “*kernels*” são técnicas utilizadas para desenvolver funções não lineares no SVM. A ideia principal é transformar a hipótese da fronteira de decisão da Equação (3.15), em novas características f_i . Existem diferentes tipos de “*kernel*” (Tabela 3.2), embora neste trabalho adotou-se a utilização unicamente do “*kernel*” gaussiano. No “*kernel*”, os pontos de aprendizagem são projetados em um novo espaço graças a função $\Phi(x)$, que quase sempre torna o problema linearmente separável.

Tabela 3.2 Funções Kernel.

Tipos Kernel	Função de kernel	Descrição
Polinomial	$(x^T y + 1)^p$	O parâmetro é p .
RBF Gaussiana	$e^{-\left(\frac{ x-x_i ^2}{2\sigma^2}\right)}$	O parâmetro é σ
Perceptron de 2 camadas	$\tanh(\beta_0 x^T x_i + \beta_1)$	Os parâmetros são β_0 e β_1

Segundo Cortes (1995), os vetores de entrada $\langle x^{(i)}, x \rangle$ (produto interno) aparecem como produto interno no espaço de características (um dos motivos pela mudança para formulação de Lagrange).

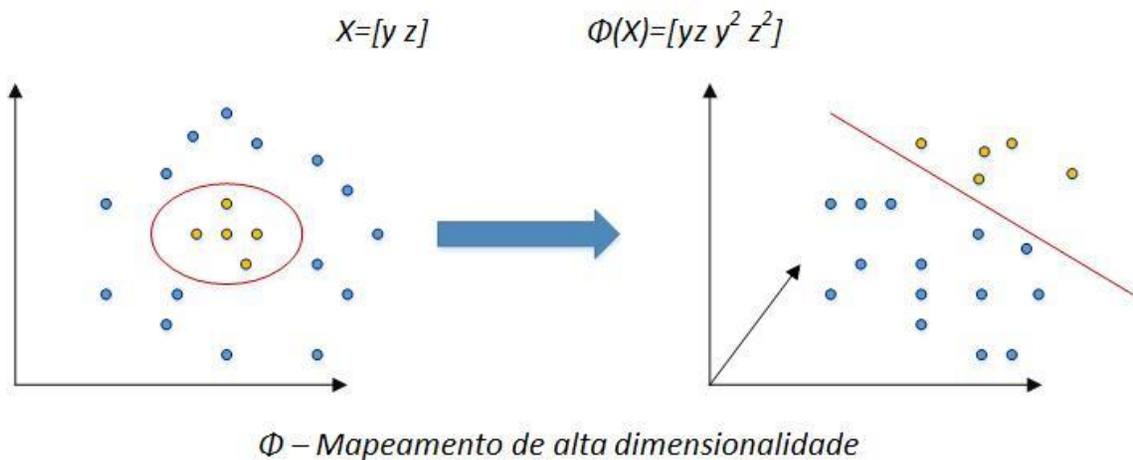


Figura 3.6 Exemplo de Mapeamento do SVM, foi Convertido para um Problema Linearmente Separável.

Por conseguinte, é possível definir uma função “kernel”, a qual permite calcular o produto interno dos vetores sem necessidade de conhecer explicitamente o vetor mapeado no espaço de características finais. Na Figura 3.6 mostra-se um exemplo do funcionamento geral de um “kernel”.

Quanto maior for a dimensão do espaço de descrição, o “kernel” vai apresentar uma maior probabilidade de obter um hiperplano. Ao transformar o espaço de entrada em um espaço superior (teoricamente de dimensão infinita), é possível obter uma separação das classes mediante hiperplanos. As novas características f_i vão depender da similaridade dos pontos de referência “landmarks” $l^{(i)}$, dado uma entrada x conforme pode ser visto na Equação (3.16) (NG, 2012),

$$f_i = \text{similaridade}(x, l^{(i)}) = \exp\left(-\frac{\|x - l^{(i)}\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{j=1}^n (x_j - l_j^{(i)})^2}{2\sigma^2}\right), \quad (3.16)$$

similaridade = kernel Gaussiano

portanto, se $x \approx l^{(i)}$, ou seja a distância é muito perto de zero, f_i seria 1. Pelo contrário, se x está muito longe de $l^{(i)}$, f_i seria 0, como se mostra na Equação (3.17).

$$f_i \approx \exp\left(-\frac{(\text{valor pequeno})^2}{2\sigma^2}\right) \approx 1, \quad f_i \approx \exp\left(-\frac{(\text{valor grande})^2}{2\sigma^2}\right) \approx 0. \quad (3.17)$$

Na Figura 3.7 se mostra a Equação (3.17), para uma amostra.

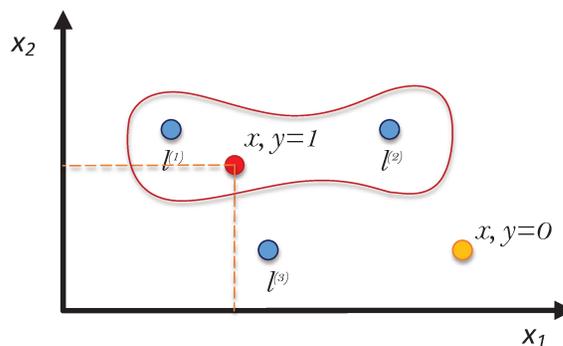


Figura 3.7 Similaridade, Kernel Gaussiano para uma Amostra.

O parâmetro σ^2 da gaussiana é o limiar para a transformação das novas características. Assim, quanto menor o σ^2 , a área plana da gaussiana será maior, portanto terá maior probabilidade para que f_i seja zero, como mostrado na

Figura 3.8, $x = [x_1 \ x_2]$ (NG, 2012).

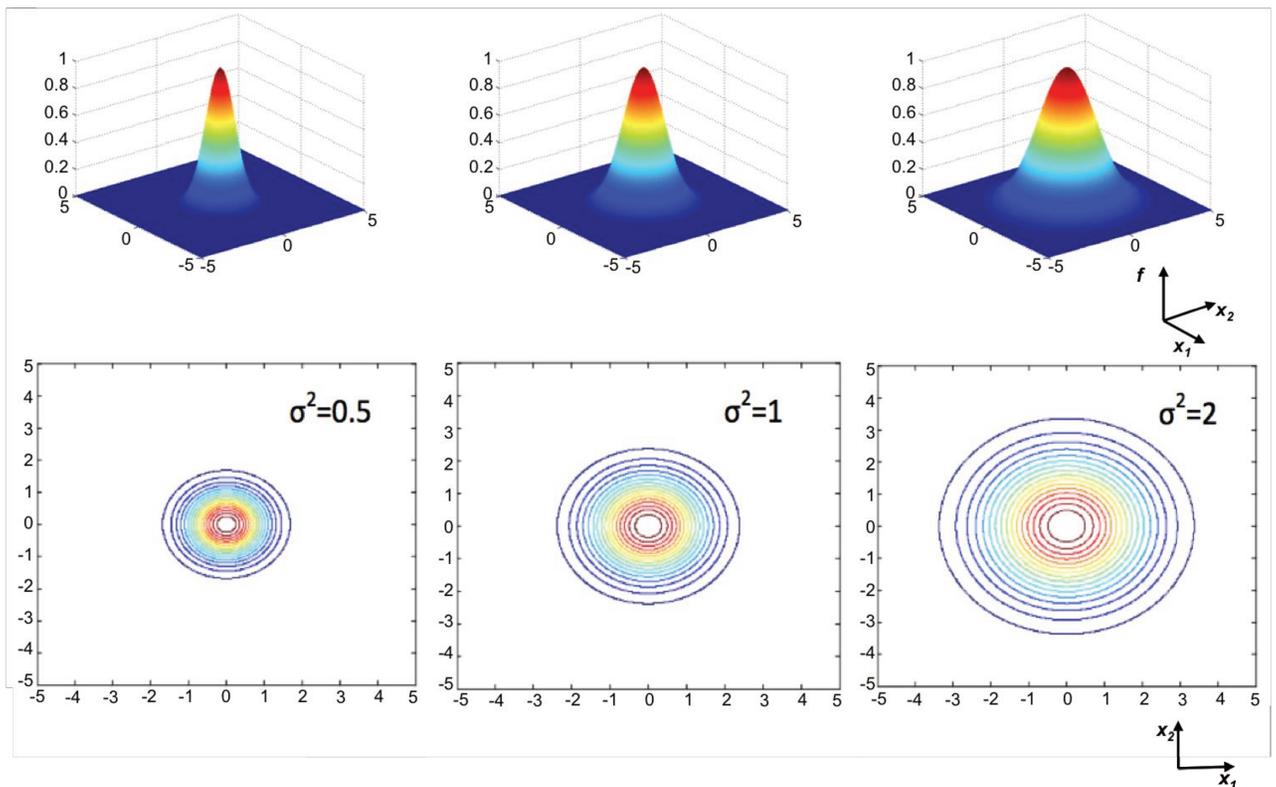


Figura 3.8 Análise da Gaussiana com Diferentes Parâmetros de σ^2 no Ponto $[0 \ 0]$.

Na prática, os $l^{(i)}$ são exatamente iguais os dados de treinamento do algoritmo SVM (NG, 2012). Quer dizer, $l^{(i)} = x_{train}^{(i)}$, portanto é medido quão perto cada amostra está com respeito a todo o conjunto de treinamento. Desta forma, tendo as características originais, transforma-se nas novas características f_i , explicadas na Equação (3.16), sendo $f_0 = 1$.

Substituindo a Equação (3.16) (as novas características f_i) na Equação (3.7), com $n = m$, já que os θ são do mesmo tamanho que as características, tem-se,

$$\begin{aligned} \underset{\theta_0, \theta_1, \dots, \theta_n}{\text{minimize}} \quad & C \sum_{i=1}^m [y^{(i)} \text{Cost}_1(\theta^T f^{(i)}) + (1 - y^{(i)}) \text{Cost}_0(\theta^T f^{(i)})] \\ & + \frac{1}{2} \sum_{j=1}^m \theta_j^2, \end{aligned} \quad (3.18)$$

onde, na implementação do SVM, a segunda somatória da Equação (3.18), pode ser substituída por $\theta^T \theta$ ou $\theta^T M \theta$, onde M é uma matriz definida dependendo do “kernel” utilizado. Desta forma, permite executar o algoritmo SVM com maior eficiência, já que é possível escalonar o conjunto de dados de treinamento e reduzir o custo computacional.

Assim, no caso não-linear obtém-se facilmente as novas características substituindo na Equação , $\langle x^{(i)}, x \rangle$ por $K(x^{(i)}, x^{(j)}) = \langle \Phi(x^{(i)}, x^{(j)}) \rangle$, onde K é a função chamada de “kernel”. Por conseguinte, podem-se realizar todos os cálculos utilizando K obtendo a Equação (3.19) (ABU-MOSTAFA, 2012).

$$f(x) = \sum_{i=1}^n \alpha^{(i)} y^{(i)} K(x^{(i)}, x^{(j)}) + b. \quad (3.19)$$

As funções de “kernel” aceitáveis devem cumprir a condição de Mercer (SCHÖLKOPF, et al., 1998). O cumprimento da condição de Mercer garante que o problema quadrático tenha solução. Os parâmetros do algoritmo SVM, que precisam ser escolhidos são C e o σ^2 , no caso de se utilizar o “kernel” gaussiano.

- C grande : baixa bias e alta variância.
- C pequeno : alta bias e baixa variância.
- σ^2 grande : alta bias e baixa variância.
- σ^2 pequeno : baixa bias e alta variância.

3.2.4 Caso não Separáveis

Quando o conjunto de aprendizagem não é separável, é necessário introduzir uma variável de relaxação na definição da restrição. A cada dado é associado uma nova variável $\xi^{(i)}$ o

qual nos indicará se o dado está no lugar certo ou errado do hiperplano. Este termo está relacionado com a função de custo.

Um dado $x^{(i)}$ está bem classificado se $\xi^{(i)} = 0$. Se $\xi^{(i)} \neq 0$, ou seja, se $x^{(i)}$ for classificado errado, então $\xi^{(i)} \geq 1$; assim indicando até que ponto está errado. O problema é a procura do hiperplano que implica numa margem maior e um número menor de erro de dados classificados, obtendo como função de custo a Equação (3.20),

$$\underbrace{\text{minimize}}_{\theta, b, \xi} \frac{1}{2} \|\theta\|^2 + C \sum_{i=1}^n (\xi^{(i)})^2, \quad (3.20)$$

$$y^{(i)} f(x^{(i)}) \geq 1 - \xi^{(i)} \text{ Com } i (1, n)$$

onde $\xi^{(i)}$ é a variável de relaxação de uma dificuldade e C a variável de regularização, ou seja, penalização da relaxação.

3.2.5 Comparação entre Máquina de Suporte de Vetores e Rede Neural.

A Tabela 3.3 apresenta uma comparação qualitativa entre os métodos baseados em SVM e RNA. A transformação de características é também conhecido como transformação em espaços de dimensão.

Tabela 3.3 Comparação entre SVM e RNA.

Máquina de Vetores de Suporte	Rede Neurais
As funções "kernel" transformam os dados em espaços de alta dimensionalidade.	As camadas transformam os dados em espaços de qualquer dimensão.
Tem somente um mínimo global.	Apresenta mínimos locais.
Precisa dos parâmetros custo e "kernel".	Precisa da estrutura da rede (camadas e nós).

3.3 Diagnóstico de Aprendizagem de Máquina

Nos problemas de aprendizagem de máquina, é possível realizar procedimentos para melhorar a performance do algoritmo, tendo diferentes soluções (NG, 2012):

- Maiores amostras de treinamento: Com maiores amostras, maior informação do sistema, porém alguns problemas apresentam dificuldades na obtenção de dados para o modelo. Ajuda nos problemas de alta variância.
- Conjunto menor de características: Ajuda a evitar problemas de sobre ajuste. Ou seja, ajuda nos problemas de alta variância.
- Adicionar características: Com mais entradas, o modelo é mais completo, embora pode-se gastar muito tempo obtendo a informação do sistema. Ajuda nos problemas de alta bias.
- Diminuir ou incrementar regularização: Ajuda nos problemas de sobre ajuste ou baixo ajuste. Diminuindo a regularização ajuda nos problemas de altas bias. Incrementando a regularização ajuda nos problemas de alta variância.

Porém, saber qual dos passos anteriores tem-se que realizar é difícil, e por isso utiliza-se como ajuda uma técnica simples chamada, diagnóstico de aprendizagem de máquina. A técnica ajuda a orientar na solução dos problemas do algoritmo, sem perder tempo realizando tarefas que não ajudariam ao problema (ABU-MOSTAFA,2012).

3.3.1 Avaliar a Hipótese

Pode-se pensar que a obtenção de um valor muito baixo de erro de treinamento significa que o algoritmo teve uma boa performance. Porém, pode acontecer que o algoritmo tenha tido problemas de sobre ajuste, portanto falhará ao generalizar novos exemplos. Mas como saber se esta apresentando problemas de sobre ajuste?.

É possível saber, colocando na forma gráfica a função $h_{\theta}(x)$ e observar se apresentou um “*overfit*” (sobreajuste). Entretanto no caso de múltiplas características, torna-se difícil a sua visualização. Por conseguinte é necessário de outra forma avaliar as hipóteses (NG, 2012).

Para garantir a avaliação das hipóteses, é possível dividir o conjunto de dados em duas partes. A primeira parte vai ser um conjunto de treinamento com 70% dos dados, e a segunda opção vai ser o conjunto de teste com 30%. Para evitar qualquer tipo de ordenamento, escolhem-se os dados aleatoriamente, conforme a notação a seguir.

$$\begin{aligned}(x^i, y^i) &\rightarrow \text{Dados treinamento } 70\% \\ (x_{test}^i, y_{test}^i) &\rightarrow \text{Dados teste } 30\%\end{aligned}$$

A principal tarefa é aprender os parâmetros do vetor θ com o conjunto de treinamento, minimizando o erro de treinamento, $J(\theta)$. Posteriormente deve-se computar o erro de teste conforme mostrado na Equação (3.21),

$$J_{test}(\theta) = \frac{1}{2 m_{test}} \sum_{i=1}^{m_{test}} (h_{\theta}(x_{test}^{(i)}) - y_{test}^{(i)})^2. \quad (3.21)$$

3.3.2 Seleção do Modelo

Para escolher o grau do polinômio em regressão logística; ou a estrutura na rede neural (Refere-se APÊNDICE D); ou inclusive para se escolher os parâmetros de regularização λ (Regressão Logística e Rede Neural, Refere-se APÊNDICE D), C e σ^2 (Máquina de Vetores de Suporte), precisa-se dividir os dados em três partes denominados de treinamento, validação e teste.

Se é utilizado somente o erro de treinamento para se escolher o melhor modelo, tem-se problemas de generalização, já que o modelo foi generalizado com estes dados de treinamento, não para novos dados. Em virtude disso, precisa-se de dados do outro grupo de validação.

Portanto a divisão dos dados adotada ficou da seguinte forma, 60% de treinamento, 20% de validação e 20% de teste, e ainda colocando aleatoriamente as amostras nos grupos para evitar uma generalização errada, conforme mostra a notação seguinte.

$$\begin{aligned}(x^i, y^i) &\rightarrow \text{Dados treinamento } 60\% \\ (x_{val}^i, y_{val}^i) &\rightarrow \text{Dados validação } 20\% \\ (x_{test}^i, y_{test}^i) &\rightarrow \text{Dados teste } 20\%\end{aligned}$$

Os erros são dados pela Equação (3.22),

$$\begin{aligned} \text{erro treinamento } J_{\text{train}}(\theta) &= \frac{1}{2 m_{\text{train}}} \sum_{i=1}^{m_{\text{train}}} (h_{\theta}(x_{\text{train}}^{(i)}) - y_{\text{train}}^{(i)})^2 \\ \text{erro validação } J_{\text{val}}(\theta) &= \frac{1}{2 m_{\text{val}}} \sum_{i=1}^{m_{\text{val}}} (h_{\theta}(x_{\text{val}}^{(i)}) - y_{\text{val}}^{(i)})^2 \\ \text{erro test } J_{\text{test}}(\theta) &= \frac{1}{2 m_{\text{test}}} \sum_{i=1}^{m_{\text{test}}} (h_{\theta}(x_{\text{test}}^{(i)}) - y_{\text{test}}^{(i)})^2 \end{aligned} \quad (3.22)$$

Para a seleção do modelo utiliza-se o conjunto de validação, ou seja, realiza-se vários modelos e escolhe-se o modelo com o erro mais baixo na validação. Isto é conhecido como o método de validação cruzada.

3.3.3 Diagnóstico Bias Versus Variância

Nos algoritmos de aprendizagem é necessário saber se existe problemas de bias (baixo ajuste) ou de variância (sobre ajuste), ou inclusive se apresenta os dois, para escolher o caminho para se corrigir ou tratar de melhorar o algoritmo.

No desenvolvimento do algoritmo pode-se ter problemas de sobre ajuste ou baixo ajuste. A melhor forma de solucionar esses problemas é ter hipóteses em um nível intermédio de complexidade, já que se obtém o menor erro generalizado (NG, 2012).

Na Figura 3.9 mostra-se os erros de validação e treinamento com respeito aos modelos feitos do algoritmo. Quer dizer, se for regressão polinomial (APÊNDICE F) seria o grau polinomial, se for Rede Neural (APÊNDICE G) seria a estrutura da rede e se for o SVM seriam os parâmetros de penalização e fator gaussiana.

Se o modelo for muito simples, o algoritmo pode ter baixo ajuste. Porém, se for muito complexo pode ter sobre ajuste, e por conseguinte, pode ter um erro de generalização elevado.

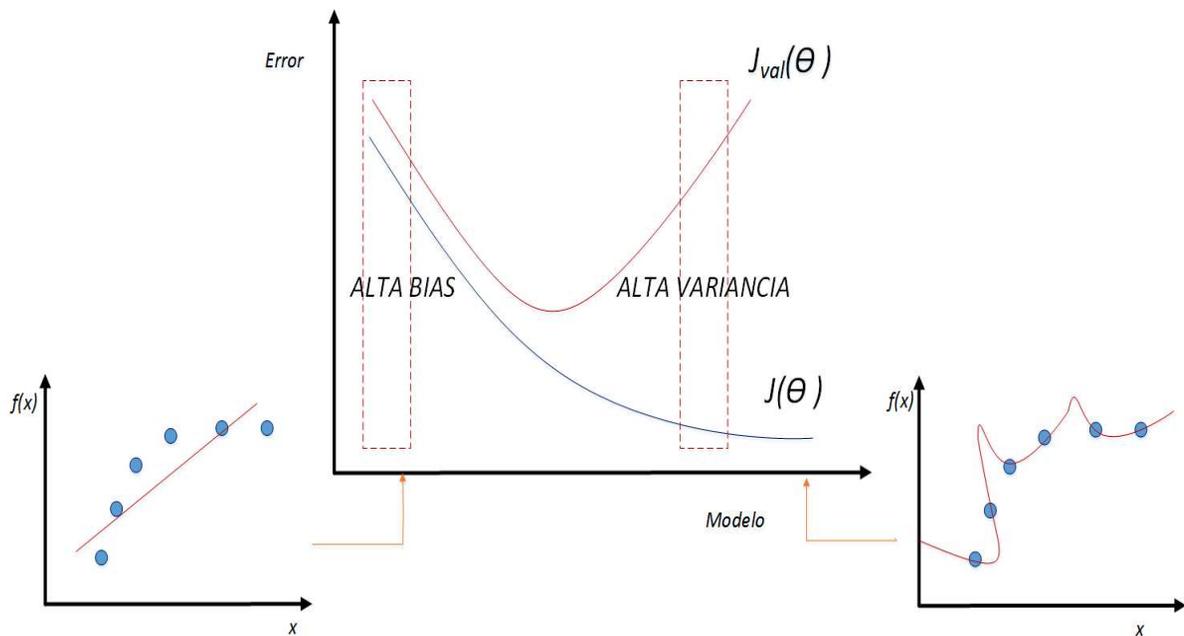


Figura 3.9 Bias ou Variância com Respeito aos Modelos.

A Figura 3.9 ajuda no entendimento das noções de bias e de variância. Para descobrir se o algoritmo apresenta alta bias ou alta variância, onde ambos apresentam erro de validação cruzada alta, vai depender do erro de treinamento, conforme mostrado na notação abaixo.

$$\begin{aligned} \text{Bias} &\rightarrow J_{\text{train}}(\theta) \approx J_{\text{val}}(\theta) \\ \text{Variância} &\rightarrow J_{\text{train}}(\theta) \ll J_{\text{val}}(\theta) \end{aligned}$$

3.3.4 Curvas de Aprendizagem

As curvas de aprendizagem se mostram como uma ferramenta importante para se saber quando o algoritmo está trabalhando corretamente, diagnosticando problemas de bias ou variância. O objetivo portanto é colocar na forma gráfica (Refere-se Figura 3.10) os erros de treinamento e validação em função ao número de amostras, ou seja m , e quando o conjunto de dados é muito grande recomenda-se escolher 10 ou 20 amostras para observar o seu comportamento (ETHEM, 2010) e (NG, 2012).

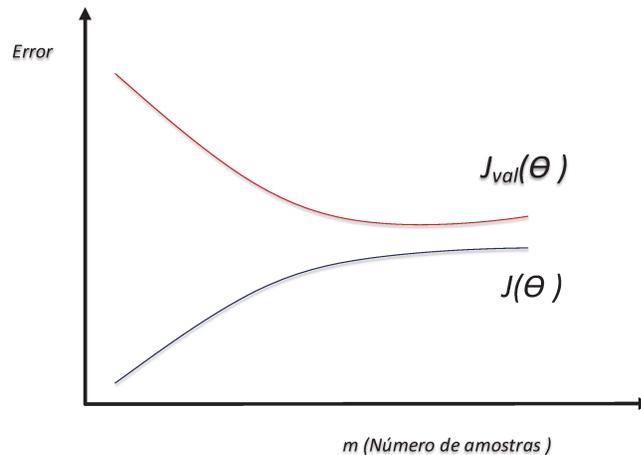


Figura 3.10 Gráfico de Curvas de Aprendizagem.

Quanto menor for o número de amostras, o erro de treinamento vai ser pequeno e o erro de validação elevado. De outro modo, quanto maior for o número de amostras, o valor dos erros passam a ser parecidos. Com ajuda da curva de aprendizagem é possível determinar se o algoritmo apresenta alta bias ou alta covariância (Refere-se Figura 3.11).

- Alta bias: Se a curva tem um comportamento como mostra-se na Figura 3.11 (a), significa que apresenta um problema de baixo ajuste. Se o algoritmo apresenta o problema de alta bias, o erro tanto de treinamento como de validação vão ser elevados. É importante notar que a obtenção de um número maior de amostras não vai ajudar muito, já que vai apresentar a mesma resposta de baixo ajuste.
- Alta variância: Se a curva tem um comportamento como mostrado na Figura 3.11 (b), significa que apresenta um problema de sobre ajuste. Se o algoritmo apresenta o problema de alta variância, os erros vão ser muito diferentes em no gráfico. Portanto neste caso, se são obtidas mais amostras, ajuda na obtenção do modelo.

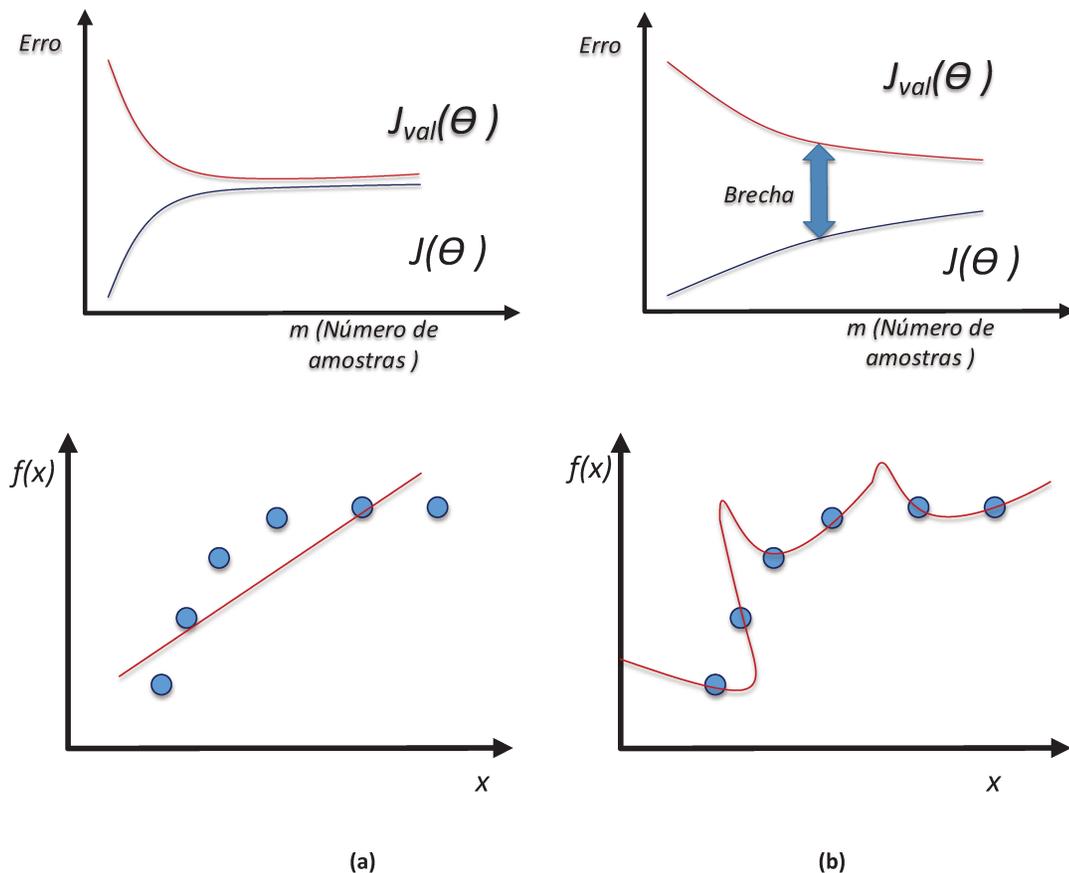


Figura 3.11 Curva de Aprendizagem. (a) Apresenta Alta Biais (b) Apresenta Alta Variância.

3.4 Filtro de Partículas

O filtro de Partículas, em estatística, é conhecido como Método Recursivo de Monte Carlo. É um método sofisticado de estimação baseado em simulação dos estados que se deseja saber. Cada partícula é uma simulação completa das variáveis que se pode obter com sensores reais.

3.4.1 Características do Filtro de Partículas

- As partículas operam em um ambiente simulado para obter medidas contínuas.
- Cada partícula é uma hipótese.
- O conjunto de muitas partículas forma um sistema multimodal de modo a cobrir de maneira uniforme uma grande parte do universo de discurso em que se está operando.

3.4.2 Matemática do Filtro de Partículas

Os filtros de partículas são eficazes para a resolução de problemas de estimação de espaço de estado quando as equações são não lineares e a densidade posterior (probabilidade dos estados ocultos) é não gaussiana. O filtro de partículas calcula a probabilidade posterior do estado do sistema sequencialmente (DOUCET, TADIC, 2003) e (THRUN, 2013). Depois, a probabilidade posterior é utilizada para se calcular os parâmetros desconhecidos do sistema. Considere o seguinte modelo de espaço de estado com estados não lineares e uma função de medida, f_k e h_k , respectivamente,

$$\begin{aligned} x_k &= f_k(x_{k-1}, v_{k-1}) \\ z_k &= h_k(x_k, n_k) \end{aligned} \quad (3.23)$$

sendo k o index, x_k o vetor de estados e z_k o vetor de medições. Os parâmetros v_k e n_k são variáveis independentes e tem ruído identicamente distribuído para o processo e medição do sistema, respectivamente (LIANGCHENG, HUIZHONG, 2009). O objetivo da estimação de estados é calcular sequencialmente o vetor de estados x_k dadas as medições z_k .

O objetivo da estimação de estado é achar a função de probabilidade $p(x_k|z_{1:k})$ do estado x_k dada a medições sequencial $z_{1:k-1}$, ($z_{1:k-1} = [z_1, z_2, \dots, z_{k-1}]$).

A função de probabilidade pode ser obtida sequencialmente através de predição e atualização, como se mostra na seguinte equação (THRUN, 2013) e (LIANGCHENG, HUIZHONG, 2009),

$$\begin{aligned} p(x_k|z_{1:k-1}) &= \int p(x_k|x_{k-1}) p(x_{k-1}|z_{1:k-1}) dx_{k-1} \\ p(x_k|z_{1:k}) &= \frac{p(x_k|z_k) p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \end{aligned} \quad (3.24)$$

A ideia básica do filtro de partículas é aproximar $p(x_k|z_{1:k})$ usando ajuste de amostras randômicas, quer dizer, partículas x_k^i associadas ao parâmetro pesos w_k^i (THRUN, 2013), conforme mostrado na Equação (3.25),

$$p(x_k|z_{1:k}) \cong \sum_{i=1}^N w_k^i \delta(x_k - x_k^i), \quad (3.25)$$

sendo, $\sum_{i=1}^N w_k^i = 1$

sendo $\delta(x)$ uma função indicadora com valor igual a 1 se $x=0$, e igual a 0 se $x=1$. Assim, o passo chave passa a ser a geração de amostras aleatórias de $p(x_k|z_{1:k})$. A reamostragem é usada para se obter as partículas e os pesos associados. Na amostragem deve-se definir a densidade $q(x_k|z_{1:k})$ na qual as amostras x_k^i podem ser obtidas (SMITH, GELFAND, 1992). Os pesos são definidos pela Equação (3.26),

$$w_k^i \propto \frac{p(x_k|z_{1:k})}{q(x_k|z_{1:k})}. \quad (3.26)$$

Para estimação sequencial do problema, no ponto k as partículas que aproximam-se $p(x_k|z_{1:k})$ (probabilidade posterior) serão colocadas na função de estados e atualizado com uma nova medição z_k para aproximar $p(x_k|z_{1:k})$ (PITT et al, 1999) e (LIANGCHENG, HUIZHONG, 2009). Se a densidade só depende da medição z_k e o estado anterior x_{k-1} os pesos podem ser atualizados pela equação a seguir,

$$w_k^i \propto w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)}. \quad (3.27)$$

Com essas partículas e os pesos associados, a estimação do vetor de estados \hat{x}_k é o valor médio de $p(x_k|z_{1:k})$ e é calculado pela Equação (3.28),

$$\hat{x}_k = \sum_{i=1}^N w_k^i x_k^i. \quad (3.28)$$

3.4.3 Funcionamento do Filtro de Partículas

Tendo que o sistema não conhece qual é o valor das variáveis a serem estimadas, ele supõe que pode ser de qualquer valor dada uma covariância. Isto leva a uma distribuição

uniforme onde todos os valores possuem a mesma probabilidade de serem o valor verdadeiro. Então, o filtro de partículas espalhará de maneira uniforme uma quantidade n_p de partículas pela covariância disponibilizada. Cada partícula será uma simulação do sistema medido dos mesmos sensores que o sistema real (THRUN, 2013).

O sistema realiza uma medida de sensores. Todas as partículas realizarão o mesmo procedimento. Compara-se as medidas dos sensores das partículas com as medidas do sensor real. Desta forma, guarda-se as similares, e as demais são descartadas. Como se terminou com menos partículas, são reproduzidas as partículas faltantes até obter novamente as n_p partículas. As partículas que mais se repetirão serão as que possuem medidas mais próximas das medidas reais. A quantidade de partículas são geradas usando o métodos de Monte Carlo (DOUCET et al. 2000), para aproximar a probabilidade posterior dos estados. Da mesma forma, as partículas concentram-se de forma adaptativa nas regiões de alta probabilidade.

Movimenta o sistema e todas as partículas pelo ambiente de maneira igual e são repetidos os passos anteriores de obtenção e comparação de medidas, na qual faz-se uma reprodução das melhores partículas. Assim, repetindo estes passos novamente, as partículas tenderão a ficar todas no mesmo valor, e a média de todas as partículas fornecerá uma boa aproximação do valor dos estados do sistema que se deseja achar (THRUN, 2013).

3.4.4 Procedimento do Filtro de Partículas

1. Movimento do sistema: rodar o sistema k iterações.
2. Sensoriamento: Medição dos sensores.
3. Comparação: Obtenha o peso de importância w de cada partícula como se mostra no Algoritmo 3.1. As partículas que sobrevivem são aleatórias, embora a probabilidade de sobrevivência será proporcional aos pesos, ou seja, uma partícula com peso maior vai sobreviver a uma probabilidade maior do que uma partícula com um peso pequeno. Escolhe-se as partículas com alta probabilidade, faz-se com que as partículas se agrupam em torno de regiões com alta probabilidade posterior. (THRUN, 2013).

Algoritmo 3.1 Algoritmo achar os pesos das partículas .

Entrada: Sensor partículas e sensor real, α dos sensores.

Saída: Pesos w .

Auxiliar: Número de Partículas n e sensores l .

$w = [1 \ 1 \ \dots \ 1_n]$

for $i := 1$ **to** n **do** %partículas

for $j := 1$ **to** l **do** %sensores

$$gauss = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}(\text{sensor}_{part.}(i,j) - \text{sensor}_{real}(j))^2\right)$$

$$w(i) = gauss * w(i)$$

$$w = \frac{w}{\sum_{i=1}^n w(i)} \text{ \%normalização}$$

4. Reamostragem: O Algoritmo 3.2 (roleta) é utilizado para emular uma roleta aleatória que escolha uma das n_p partículas utilizando as chances dos pesos, como se mostra na Figura 3.12 (THRUN, 2013) e (PITT et al, 1999).

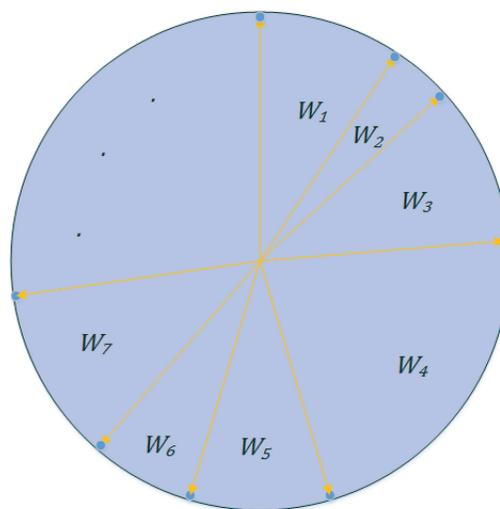


Figura 3.12 Algoritmo Gráfico de Roleta.

A reamostragem precisa de todas as partículas e seus pesos normalizados, e assim, substituindo com novas partículas dependendo da probabilidade das partículas anteriores escolhidas aleatoriamente (THRUN, 2013) e (LIANGCHENG, HUIZHONG, 2009).

Algoritmo 3.2 Algoritmo roleta.

Entrada: Pesos w .

Saída: Partículas_{remost}.

Auxiliar: Número de Partículas n .

$\beta = 0$.

$w_{max} = \max(w)$.

$index = \text{reound}(rand * n)$.

for $i := 1$ **to** n **do** %partículas

$\beta = \beta + 2w_{max}rand$

while $\beta > w(index)$ **do**

$\beta = \beta - w(index)$

$(index + 1) \bmod n$

Partículas_{remost}. $(i) = \text{partículas}(index) + \sigma_{reamostragem}randn$

Capítulo 4 .

METODOLOGIA

Muitas pesquisas apresentam resultados de programas de modelamento de solda por fricção. Porém, esses modelos são muito difíceis de obter, já que são necessários vários fatores para serem considerados quando modela-se o sistema FSW.

Esta pesquisa analisou diferentes propriedades de ensaio de tração, embora a ideia principal fosse somente obter uma boa predição da tensão de escoamento, sendo uma resposta importante na Curva Tensão-Deformação (APÊNDICE C) das propriedades mecânicas do sistema FSW. Isto deve-se à consideração de que a tensão de escoamento é a tensão máxima que o material suporta ainda no regime elástico de deformação, portanto se a tensão de escoamento é superada começa a sofrer deformação plástica, quer dizer, deformação permanente.

Os passos principais para se obter um modelo representativo das propriedades mecânicas da soldagem por FSW são: obtenção dos dados (experiências anteriores); escolha da hipóteses (o que é possível assumir do problema); representação do modelo; estimação do modelo; avaliação do modelo (desempenho do modelo); e em alguns casos escolha do modelo (se for possível melhorar).

4.1 Realização de Experimentos

Realizaram-se experimentos de preparação de testes, no qual foram mudadas as variáveis de entradas como as velocidades de rotação, de translação, e a força axial da ferramenta onde foi possível observar e identificar as mudanças da resposta de saída. Estes valores foram dados pela empresa EMBRAER. Com modelos estatísticos determinaram-se se alguns fatores influíam nas variáveis de interesse (propriedades mecânicas).

Foram realizadas replicas dos diferentes testes, nos quais são obtidas uma estimação do erro experimental, assim como, calcular uma resposta mais precisa. Na Figura 4.1 mostram-se as etapas para a realização do experimento.

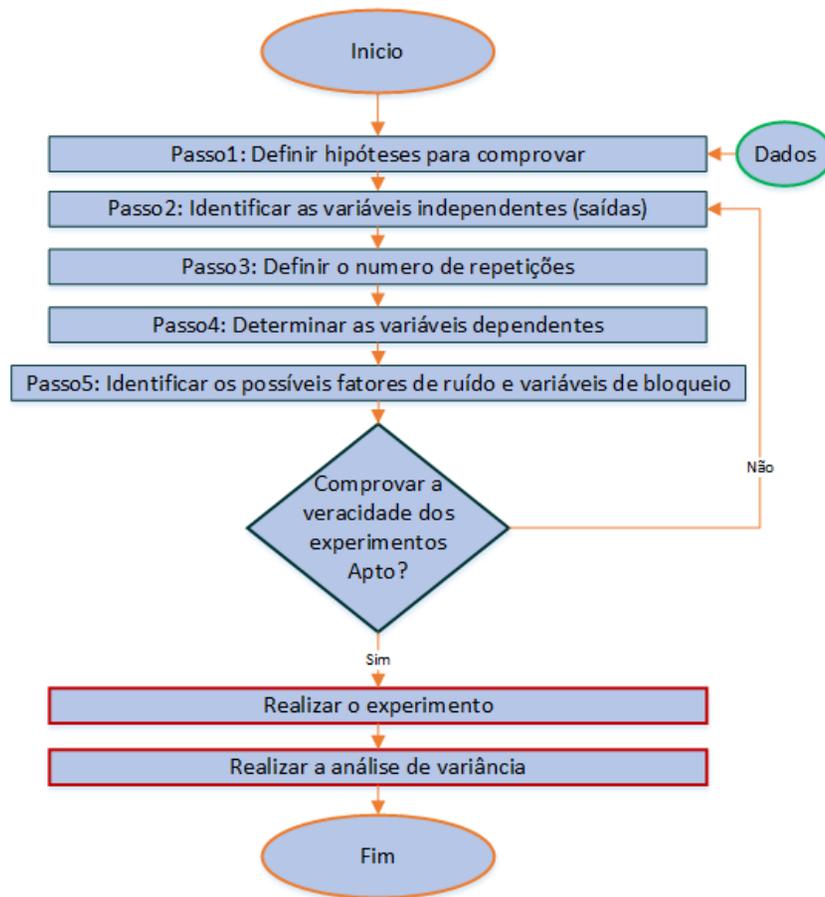


Figura 4.1 Arquitetura das Etapas do Experimento.

4.1.1 Análise Preliminar

Realizaram-se corpos de prova com a norma ASTM E8M, para obter os resultados dos testes de tração para os dados da saída (parâmetros da curva Tensão-Deformação) (APÊNDICE C) em função dos dados de entrada feitos na soldagem FSW (velocidade de rotação, velocidade de translação e força axial) (ASTM, 2012). Realizou-se um análise de variância (ANOVA) (APÊNDICE B) com o intuito de se determinar as diferentes respostas obtidas pelos ensaios de tração, identificando assim, as respostas que apresentam comportamento não linear.

Depois que foram obtidas as respostas não lineares, deve-se realizar: primeiro, a criação de uma tabela com os dados (curva σ - ϵ , “tensão-deformação”); segundo, obter um treinamento pelos dados; terceiro, escolher uma função de hipóteses; quarto, criar o algoritmo de

aprendizagem; quinto, validar o algoritmo com novos dados para observar o desempenho do modelo. Pode-se observar um esquema das tarefas realizadas na Figura 4.2.

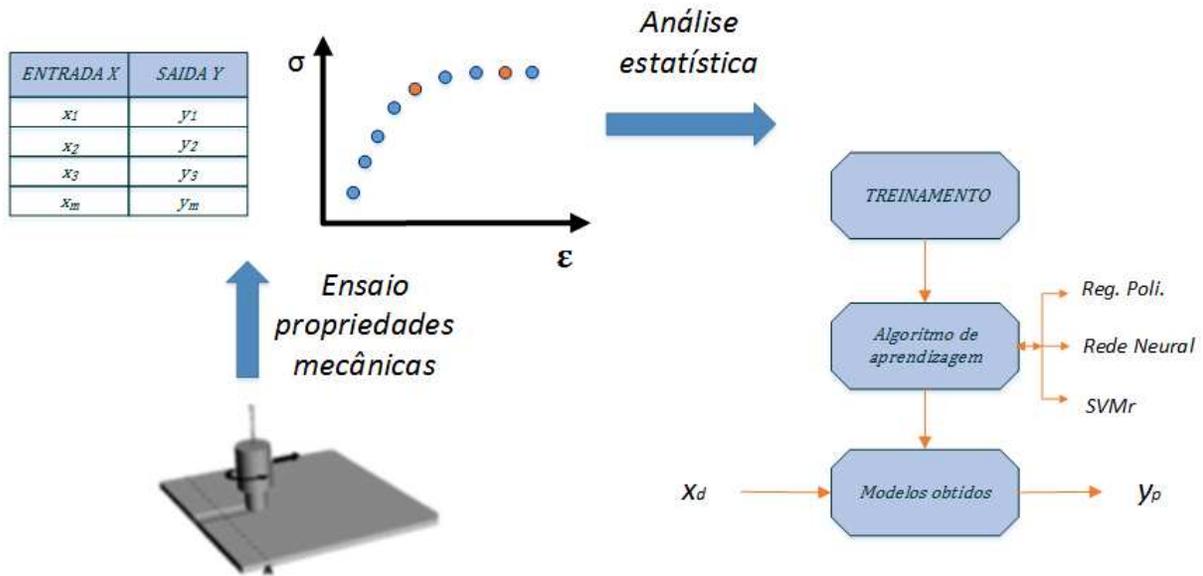


Figura 4.2 Esquema da Criação do Modelo da Solda FSW.

Os algoritmos de aprendizagem são métodos caixa preta baseados nos dados, e através deles pode-se evitar o problema de se entender a estrutura inerente aos processos que tem lugar no sistema modelado (ETHEM, 2010). O modelo SVMr como também as redes neurais são métodos caixa preta.

O problema de regressão dispõe de uma série de dados de treinamento que representam as entradas e as saídas de um sistema linear ou não linear (obtidos pelos ensaios de tração). O objetivo da regressão é descobrir a relação funcional entre as entradas e as saídas do sistema, para poder assim prever as saídas do sistema quando se ingressa com um novo dado de entrada (Aprendizagem de Máquina).

4.1.2 Análise de Dados

Para os dados de entrada especificados no desenvolvimento do modelo realizaram-se testes de tração para se obter as saídas necessárias e poder encontrar os parâmetros ótimos do modelo. Assim realizou-se um fatorial completo de dois níveis para se determinar diferentes

aspectos como significância dos fatores, ou predição do modelo, como mostrado na Tabela 4.1; velocidade de rotação, velocidade de translação e força axial.

Tabela 4.1 Fatorial Completo de dois Níveis.

Experimentos	Fatores		
	VR (RPM)	VT (mm/min)	FA (kN)
1	-1	-1	-1
2	1	-1	-1
3	-1	1	-1
4	1	1	-1
5	-1	-1	1
6	1	-1	1
7	-1	1	1
8	1	1	1

Todos os procedimentos do ensaio foram iguais em todos os corpos de prova, igualmente estão de acordo com a norma ASTM E8M (ASTM, 2012). É importante mencionar que a velocidade de tracionamento utilizada para a realização dos ensaios foi de 2mm/min. Na Figura 4.3 mostra-se o plano do corpo de prova utilizado nos testes.

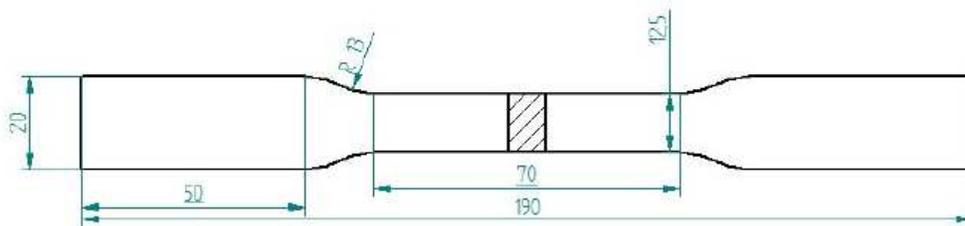


Figura 4.3 Esboço do Corpo de Prova Utilizado.

Para uma maior confiabilidade nos resultados obtidos, determinou-se que cada ensaio fosse repetido dez vezes, tendo como objetivo diminuir o erro do resultado experimental. Assim para cada combinação demonstrada nas matrizes de testes, foram confeccionados dez corpos de prova.

Resumindo, tem-se um conjunto de dados de 25 amostras, dado que realizaram-se 3 faixas de estudo dadas pelos ensaios de tração feitas no laboratório para a coleta do conjunto de dados, as quais serão analisadas posteriormente quanto a sua significância. Tem-se 3 variáveis de entradas como foi dito anteriormente e 6 variáveis de saída da curva Tensão-Deformação (tensão última; tensão de escoamento; módulo de elasticidade; alongamento da falha; alongamento do escoamento; tensão da falha) (APÊNDICE C). Sendo a solda FSW um processo altamente complexo, a realização do modelo de seu comportamento é uma tarefa muito difícil. Para isto os dados serão analisados estatisticamente (APÊNDICE B) e os que apresentarem curvatura serão modelados por diferentes algoritmos de aprendizagem (APÊNDICE F e APÊNDICE G).

A Tabela 4.2 mostra valores em porcentagem da velocidade média adotada nos ensaios do FSW. O conjunto de dados de 25 ensaios foram realizados e cada um contém uma entrada de vector de 3 características, 6 valores de saídas. Na Figura 4.4 mostram-se as 6 respostas obtidas pelas curva Tensão-Deformação (APÊNDICE C) dos ensaios de tração, sendo as saídas: Módulo de Elasticidade (Fator de proporcionalidade); Tensão e Alongamento de Escoamento; Tensão e Alongamento Última; e Alongamento de Ruptura.

Tabela 4.2 Níveis das Variáveis de Entrada para os Ensaios.

FATOR	NOTAÇÃO	UNIDADE	NIVEIS (%)						
			-3	-2	-1	0	1	2	3
Velocidade de Rotação	V.R.	RPM	82.3%	88.2%	94.1%	100%	105.8%	111.7%	117.6%
Velocidade de Translação	V.T.	mm/min	50%	75%	90%	100%	110%	125%	150%
Força Axial	F.A.	kN	87.2%	91.4%	95.7%	100%	104.2%	108.5%	112.7%

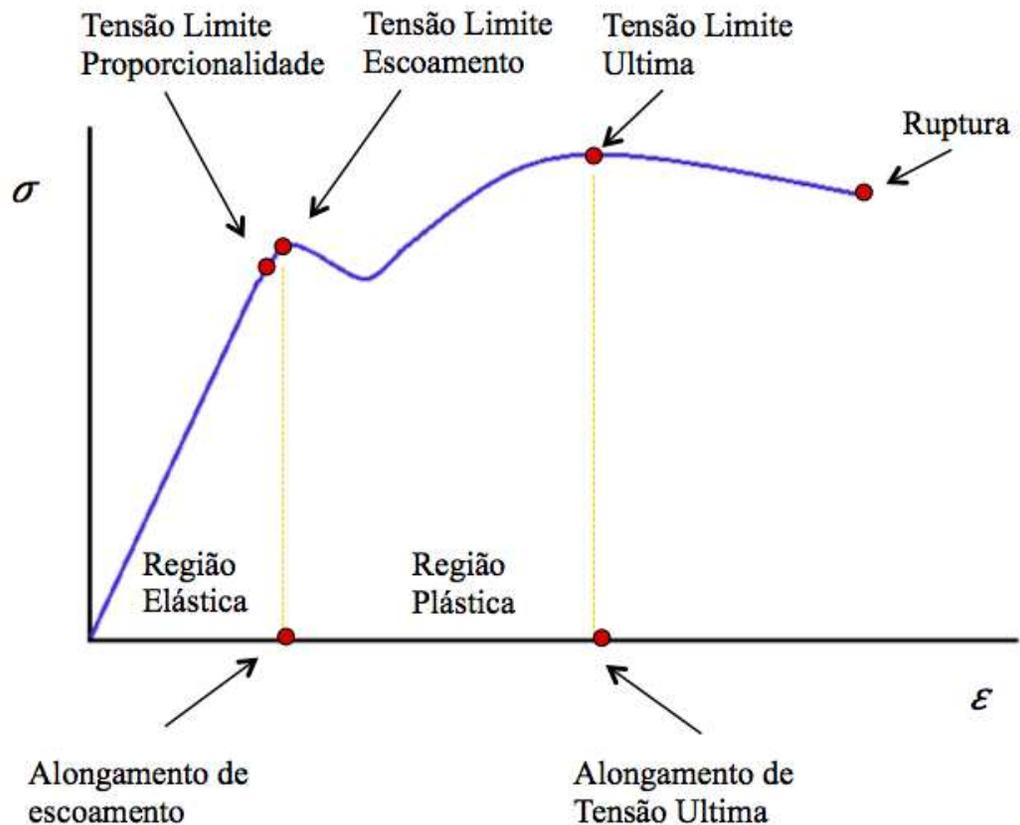


Figura 4.4 Curva Características de Ensaio de Tração.

4.1.3 Análise de Algoritmos

Utilizou-se diferentes algoritmos onde foi escolhido um conjunto de dados para se determinar as propriedades da solda FSW em alumínio. A solda FSW é um processo de alta complexidade. As propriedades mecânicas do FSW em alumínio têm diferentes características que influem não só na tensão de escoamento, mas também na tensão última e alongamento, características necessárias para se obter o gráfico Tensão-Deformação, o quais apresentam não linearidade (YOUSIF et al. 2008; OKUYUCU et al. 2007).

Nestes métodos tem-se dados de entrada que denominamos $x(i)$ e dados de saída que denominamos $y(i)$, portanto, o par (x, y) representa uma amostra de treinamento e m é o número de treinamento das amostras, conforme mostrado na notação seguinte.

$m =$ Numero de mostras de treinamento.
 $x =$ "entrada" variável / características.
 $y =$ "saída" variável.

Tendo as amostras obtidas pelos ensaios de tração (dados de treinamento), estes serão alimentados aos algoritmos de aprendizagem. O trabalho do algoritmo de aprendizagem é retornar uma função que, por convenção, é geralmente representada por uma função h , o qual representa as hipóteses, ou seja, h é uma função que recebe a/as entrada/s e tenta retornar o valor estimado de y para uma saída correspondente. Ou seja, h é uma função que mapeia de x para y , como mostrdo na Figura 4.5.

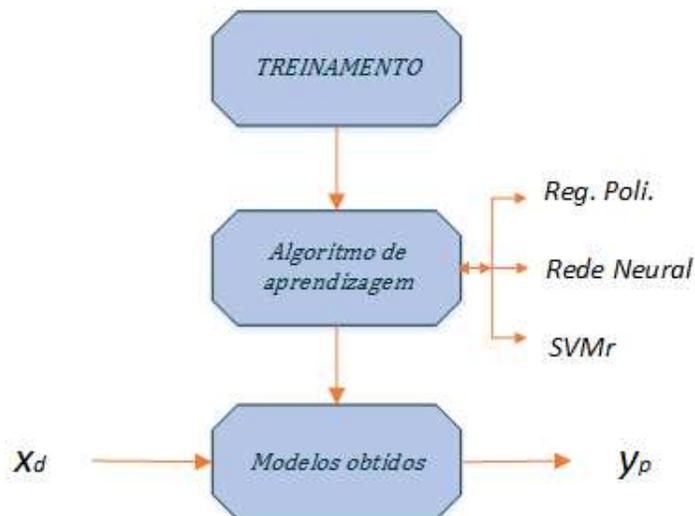


Figura 4.5 Esquema de Algoritmo de Aprendizagem.

Na Figura 4.6 é mostrado como para cada modelo de aprendizagem tem-se seu respectivo algoritmo para encontrar os parâmetros (NG, 2012). Por exemplo, para regressão logística (APÊNDICE D) somente precisa ter o método de Gradiente (otimização) (APÊNDICE E), para rede neural os métodos utilizados são “Backpropagation” (APÊNDICE G) e o método de gradiente (APÊNDICE E); e para o SVM são utilizados são algum tipo de “kernel” e “Quadratic Programming” (otimização dual) (SCHÖLKOPF, et al., 1998).

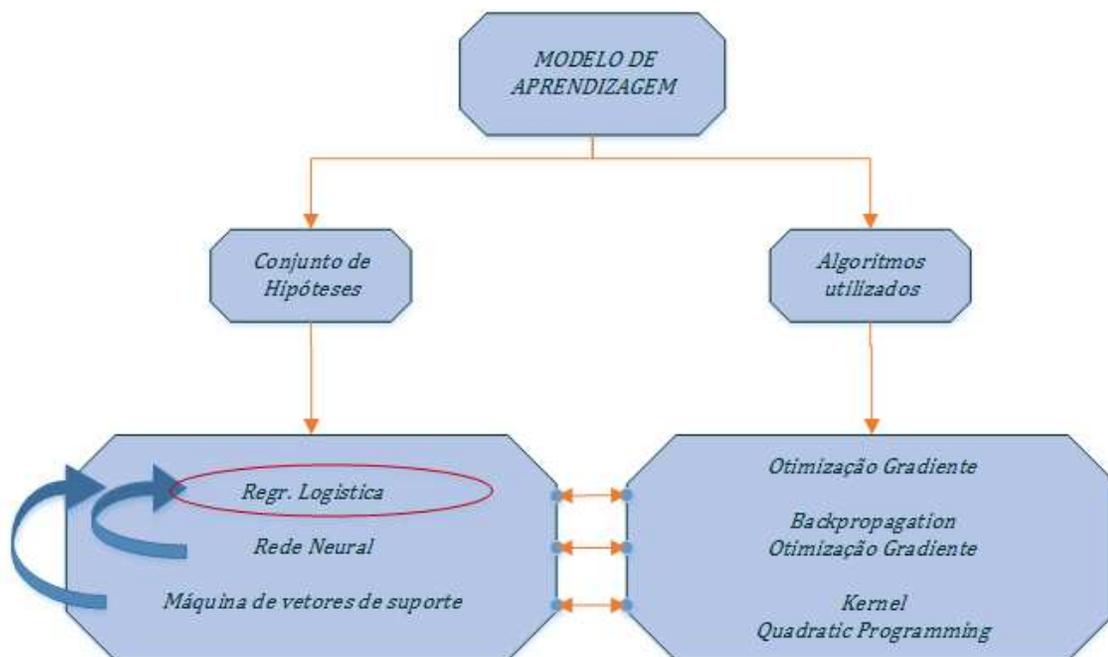


Figura 4.6 Esquema do Modelo de Aprendizagem dos Métodos.

O grande problema para se trabalhar com Regressão Polinomial (APÊNDICE F), em aplicações não lineares é ter uma quantidade alta de características ou entradas, acarretando que as hipóteses da função ou a hipótese polinomial vai ser muito grande. Além disso, a ordem de características cresce com a ordem do grau de polinômio, e a função polinomial vai ter muito mais termos, resultando com problema de sobre ajuste e custo computacional muito elevado. Além disso, para funções não lineares, somente existe aproximação por meio de polinômio, e estes problemas são solucionados tanto pela RNA (APÊNDICE G), como por SVM (ABU-MOSTAFA,2012).

Uma vantagem significativa do SVM é que enquanto as RNA podem apresentar vários mínimos locais, a solução para um SVM é global e única (BURGES, 1998). Além disso, o SVM tem vantagens quanto a sua interpretação geométrica simples e se obtiver uma solução dispersa, ao contrário da RNA (APÊNDICE G), a complexidade computacional de SVM não depende da dimensionalidade do espaço de entrada (VAPTNIK et al. 1997). Também, a RNA utiliza minimização do risco empírico, enquanto (SVM) utiliza minimização do risco estrutural, ou seja, que a função de custo do SVM funciona com a maximização da margem entre dados e os outros dois algoritmos são medidas pela diferença entre a saída predita com a real (APÊNDICE D). Em

contraste com SVM, a RNA seleciona automaticamente seu tamanho de modelo “selecionando os vetores de suporte” (RYCHETSKY, 2001).

Além das vantagens que apresenta o SVM em comparação com os outros métodos, como foi mencionado anteriormente nesta pesquisa, encontram-se os “*kernels*” que ajudam ao algoritmo SVM, ajudando para que a função tenha um mínimo global para posteriormente resolver com o método de programação quadrática (SHAWE-TAYLOR, CRISTIANINI, 2000). Embora, o algoritmo apresente algumas desvantagens que hoje em dia estão em estudo, como por exemplo, a escolha do “*kernel*” e seu parâmetro (BURGES, 1998), o mais utilizado na atualidade que apresenta bons resultados é o “*kernel*” gaussiano, o qual foi utilizado nesta pesquisa.

4.2 Máquina de Vetores de Suporte Regressão

A ideia básica para criar um comportamento de estimação no processo FSW é em forma de regressão. Para isto foi utilizado o método de Máquina de Vetores de Suporte (SVM) como regressão e não como classificação denominada na literatura como “*Support Vector Machine*”, explicado na Seção 3.2.

4.2.1 Formulação Preliminar

Para o propósito da regressão, estuda-se a dependência funcional da variável de saída $y \in \mathfrak{R}$ sobre uma variável de entrada n -dimensional.

O SVM não só tem sido aplicada ao problema de classificação de variáveis, como também tem sido aplicado satisfatoriamente ao problema de regressão ou de estimação de variáveis com o nome de SVMr, por meio da introdução de uma função de perda como se mostra na Equação (4.1),

$$\begin{aligned} f(x, \theta) &= \langle \theta, x \rangle + b \\ f(x, \theta) &= y^{(i)} \quad y^{(i)} \in \mathfrak{R} \quad \forall i \end{aligned} \quad (4.1)$$

A formulação do SVMr pode ser abordada da seguinte maneira: Considera-se um conjunto de N dados da forma $(x^{(i)}, y^{(i)})$ com $1 \leq i \leq N$, onde $x^{(i)} \in \mathfrak{R}$ e $y^{(i)} \in \mathfrak{R}$; deseja-se obter uma função linear ou não linear que produza o mínimo erro da estimação para os N dados.

Para o sistema baseado em regressão, $y^{(i)}$ será considerado como uma função dependente de $x^{(i)}$, assim, a função vai poder ser qualquer valor real. Neste caso definiu-se um erro de aproximação (função de perda) definido na Equação (4.2) em lugar da margem utilizada na classificação, através de uma nova variável de sensibilidade chamada ε ,

$$e(x, y, f) = |y - f(x, \theta)|_\varepsilon = \begin{cases} 0 & |y - f(x, \theta)| \leq \varepsilon \\ |y - f(x, \theta)| - \varepsilon & \text{contrario} \end{cases} \quad (4.2)$$

Os objetivos serão os mesmos que foram utilizados na classificação (Seção 3.2), primeiramente o de maximizar a margem e segundo o de minimizar a função de custo, ou seja o erro. O parâmetro C (regularização) neste caso indicará a qualidade da aproximação da função aos dados.

Assim, a função de perda é igual a zero se a diferença entre o valor atual e o predito de y é menor que ε . Como na classificação, o método procura minimizar o risco empírico (diferença entre o real e o predito) e os parâmetros θ simultaneamente, portanto o plano para regressão é construído minimizando a seguinte equação,

$$\frac{1}{2} \|\theta\|^2 + C \sum_i |y^{(i)} - f(x^{(i)}, \theta)|_\varepsilon \equiv \frac{1}{2} \|\theta\|^2 + C \sum_i (\xi^i + \xi^{i*}) \quad (4.3)$$

$$\text{Restrições} \begin{cases} y - \theta^T x \leq \varepsilon + \xi \\ \theta^T x - y \leq \varepsilon + \xi^* \\ \varepsilon, \xi, \xi^* \geq 0 \end{cases} .$$

Desta forma as amostras que estão a uma distância do plano menor a ε não serão penalizados pelo sistema (corresponde a segunda somatória da Equação (4.3) explicada na Seção 3.2), onde as variáveis ξ e ξ^* da segunda somatória da direita são, respetivamente, variáveis “*slack*” para medidas por acima e por baixo da zona de sensibilidade ε . O problema de otimização pode portanto ser obtido através da minimização do Lagrangiano dual,

$$\begin{aligned}
L_{\alpha, \alpha^*} = & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha^{(i)} - \alpha^{(i)*}) (\alpha^{(j)} - \alpha^{(j)*}) x^{(i)T} x^{(j)} - \varepsilon \sum_{i=1}^N (\alpha^{(i)} - \alpha^{(i)*}) \\
& + \sum_{i=1}^N y_i (\alpha^{(i)} - \alpha^{(i)*}),
\end{aligned} \tag{4.4}$$

e satisfazendo algumas condições que são mostradas a seguir,

$$\begin{aligned}
\sum_{i=1}^N (\alpha^{(i)} - \alpha^{(i)*}) = 0, \\
\alpha, \alpha^* \in [0, C]
\end{aligned} \tag{4.5}$$

e mediante o cálculo dos multiplicadores de Lagrange $\alpha^{(i)}$ e $\alpha^{(i)*}$ encontra-se o vetor de peso ótimo do hiperplano de regressão, conforme mostrado na Equação (4.6),

$$V_p = \sum_{i=1}^N (\alpha^{(i)} - \alpha^{(i)*}) x^{(i)}, \tag{4.6}$$

onde o valor de bias b ou θ_0 pode calculado através dos vetores de treinamento que satisfazem a Equação (4.7). Pelo tanto o hiperplano ótimo para regressão é a Equação (4.8),

$$\begin{aligned}
b = y - \theta_{1...n}^T x \mp \varepsilon \\
\alpha, \alpha^* \in [0, C]
\end{aligned} \tag{4.7}$$

$$f(x, \theta) = \sum_{i=1}^N (\alpha^{(i)} - \alpha^{(i)*}) x^{(i)T} x + b. \tag{4.8}$$

Uma ideia chave para um bom funcionamento do algoritmo é que as saídas correspondentes às entradas tenham resposta lineares. Para isso, faz-se uso das noções de “*kernels*” (núcleos) K . Um “*kernel*” representa uma medida de similaridade entre dois dados de entrada. É uma função simétrica que tem como entrada dois dados e cria um número real que mede a similaridade dos dados (Seção 3.2).

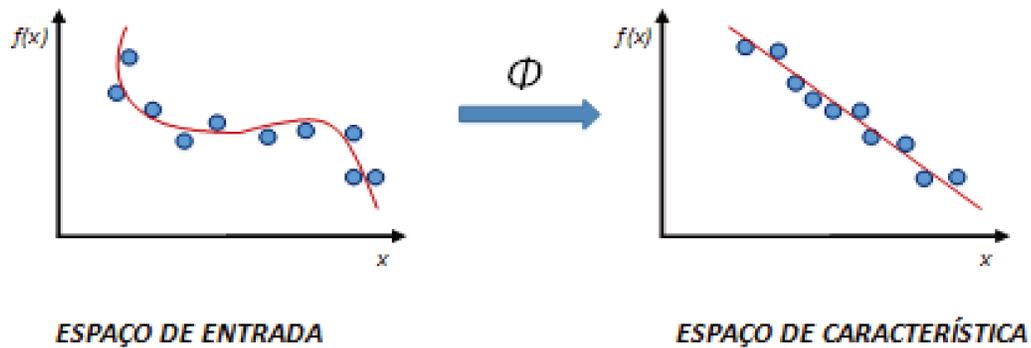


Figura 4.7 Transformação do Kernel do Método SVMr.

A ideia básica dos “*kernels*” é um mapeamento “ φ ” que transforma os pontos de entrada (pontos azuis) a um espaço de características de alta dimensão, onde correspondem a um modelo aproximadamente linear como se mostra na Figura 4.7. Este modelo no espaço de características corresponde a um modelo não linear no espaço de entrada.

Na Figura 4.8 mostra-se a arquitetura do modelo (SVMr), no qual o “*kernel*” está dado por $K(x_i, x)$ e o θ são os parâmetros achados para predição (Explicado na seguinte Seção).

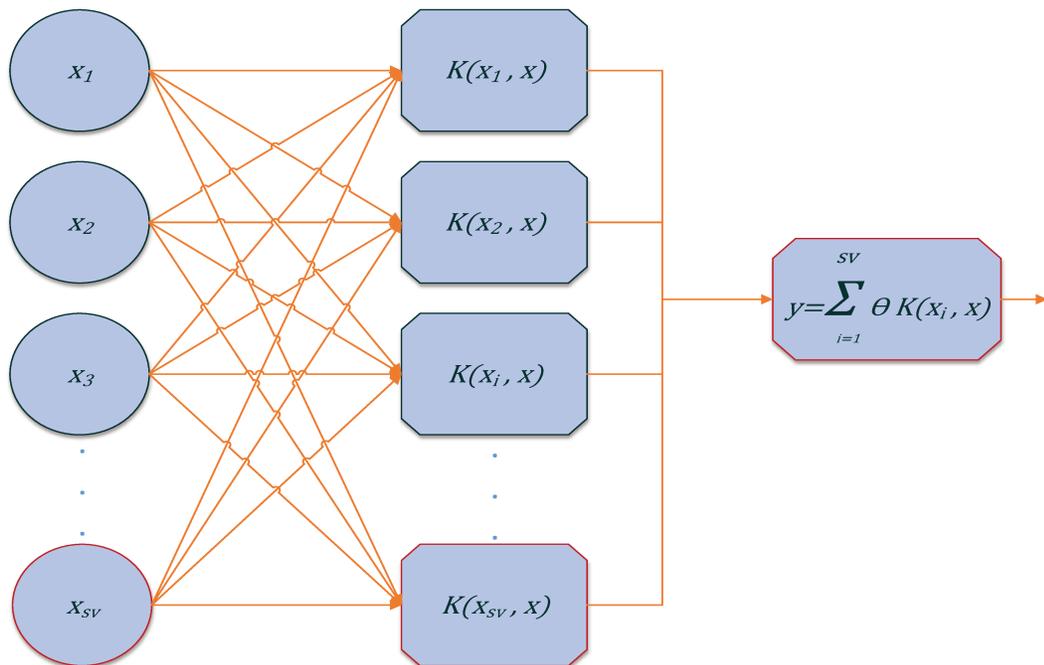


Figura 4.8 Arquitetura do Método SVM para Regressão.

4.2.2 Caso Não Linear: Kernel

Na maioria das aplicações dos problemas de regressão, a estimação de um conjunto de dados por meio de uma formulação linear não é apropriada devido à estrutura dos mesmos. A ideia do método SVMr não linear é transformar o espaço de entrada num espaço de representação de dimensão maior, onde os dados são lineares, através de uma função não linear (Seção 3.2). Denomina-se esta função como $\Phi(\cdot)$ que transforma cada dado do problema de não linear para linear, o qual foi substituído por $K(x^{(i)}, x^{(j)}) = \langle \Phi(x^{(i)})^T, \Phi(x^{(j)}) \rangle$ na Equação (4.4), mostrada na Equação (4.9),

$$L = \frac{1}{2} \sum_{i,j=1}^N (\alpha^{(i)} - \alpha^{(i)*}) (\alpha^{(j)} - \alpha^{(j)*}) K(x^{(i)}, x^{(j)}) - \varepsilon \sum_{i=1}^N (\alpha_i + \hat{\alpha}_i) + \sum_{i=1}^N y_i (\alpha_i + \hat{\alpha}_i), \quad (4.9)$$

onde foram satisfeitas algumas condições que são mostradas na Equação (4.10),

$$\sum_{i=1}^N (\alpha_i - \hat{\alpha}_i) = 0, \quad (4.10)$$

$$\alpha, \hat{\alpha} \in [0, C]$$

O produto interno $\langle \Phi(x_i), \Phi(x_j) \rangle$ é dado pelas funções de “kernel” e sempre se evita o cálculo explícito da função de transformação $\Phi(\cdot)$. É por esta razão que para os SVMr não lineares, o problema de otimização deve ser resolvido por meio da formulação dual (otimização quadrática), já que a formulação primal “encontrar o mínimo da função” requer uma função de transformação explícita, ainda que, esta transformação seja muito difícil de se obter.

No processo de treinamento, estes parâmetros deverão ser fixados pelo usuário. O desempenho do SVMr está relacionado ao valor dos parâmetros de penalização C para correção e dos parâmetros do “kernel” que é a função do núcleo. Para achar estes dados existem métodos de implementação como a “Grid – Search” e a “Cross – Validation” (Validação cruzada) (Seção 3.3) (LIN et al., 2004).

4.2.3 Hiperplano de Regressão

No SVMr o objetivo principal é encontrar uma função $f(x)$ que tenha como máximo um desvio ε da saída y_i para todos os dados do treinamento. Ou seja, tem que se preocupar com os dados que estejam dentro do intervalo a $[\varepsilon, -\varepsilon]$. Utilizando a função “kernel” $K(x^{(i)}, x^{(j)}) = \Phi(x^{(i)})^T \Phi(x^{(j)})$ e substituindo na Equação (4.8), tem-se,

$$f(x, \theta) = V_p^T \Phi(x) + b = \sum_{i=1}^N (\alpha^{(i)} - \alpha^{(i)*}) K(x^{(i)}, x^{(j)}) + b, \quad (4.11)$$

onde o termo de bias calcula-se pela Equação (4.12),

$$b = y - \sum_{j=1}^{\langle Sup|Inf \rangle} (\alpha^{(j)} - \alpha^{(j)*}) K(x^{(j)}, x^{(j)}) \mp \varepsilon, \quad (4.12)$$

$$\alpha, \alpha^* \in [0, C]$$

sendo, Sup e Inf os números de vetores livres superiores e inferiores, respectivamente.

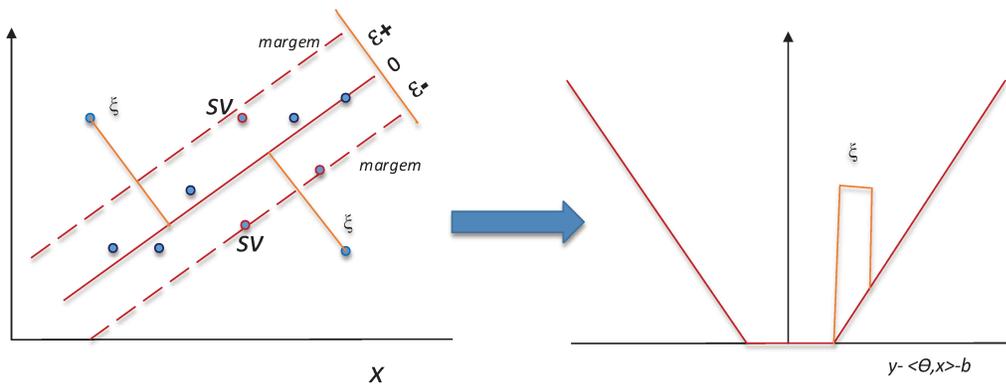


Figura 4.9 Variável de Perda ξ (Margem).

A extensão do SVM para o tratamento de problemas de regressão, é obtida pela introdução de uma função de perda. A ideia é que a maioria das amostras estejam dentro do hiperplano, como pode ser visto na Figura 4.9. Os SV são os vetores de suporte.

4.2.4 Esquema da Máquina de Vetores de Suporte

Na pesquisa foi desenvolvido um modelo de SVMr. Utilizou-se a função “*kernel*” gaussiana para o mapeamento complexo de entrada – saída. O processo de aprendizagem é usado para se determinar os parâmetros das entradas. A estratégia básica para o desenvolvimento do modelo baseado no SVMr do comportamento da FSW é treinar com os resultados das séries de experimentos realizados nos corpos de prova de tração. Se os resultados dos experimentos contém informação relevante sobre o comportamento da solda FSW, então o SVMr treinado deveria conter suficiente informação sobre o comportamento da solda FSW para caracterizar o modelo. Assim, o SVMr treinado não só deveria reproduzir os resultados experimentais com a que foi treinada, mas também através da capacidade generalizada ele deveria aproximar os resultados dos experimentos. O procedimento feito foi:

- Pré-processamento das entradas de solda FSW, ou seja, obter a média e o desvio padrão de velocidade de rotação, velocidade de translação e força axial; e obter as saídas da Tensão-Deformação por meio dos ensaios tração.
- Utilizar a função “*kernel*”. Foi utilizado a função gaussiana.
- Obter os vetores de suporte, diferença de multiplicadores de Lagrange.
- Ajustar os limites $\alpha, \alpha^* \in [0, C]$.
- Utilizou-se o toolbox QP “*Quadratic Programming*” de Matlab para o problema de otimização, para resolver a solução global.

Os valores dos parâmetros necessários para a projeção das entradas estão presentes no algoritmo SVMr, depois de se executar o treinamento. Por meio do “*kernel*” utilizado é possível mapear as entradas a predizer e achar sua respectiva saída.

Para obter os melhores valores das variáveis do SVMr (o σ do “*kernel*”, C da regularização e o ε da margem), foi utilizado a validação cruzada (Seção 3.3), sendo um método computacionalmente elevado dado pelas combinações das variáveis. Por conseguinte, para o desenvolvimento do modelo foi utilizado o conjunto de ferramentas, “*toolboxes*”, de distribuição gratuita executados no Matlab (LIBSVM) (CHANG et al, 2011).

Para o teste de precisão do SVMr treinada, foi adoptado o coeficiente de determinação R^2 e o erro MSE (APÊNDICE A).

4.3 Filtro de Partículas

O método de validação cruzada é computacionalmente muito exigente porque requer que o problema SVMr seja resolvido repetidas vezes, realizando o algoritmo por cada ponto dado até chegar a um limite definido pelo usuário.

O processo de formação da SVMr tem alguns parâmetros de ajuste como o parâmetro de penalização C , o valor σ do “kernel” gaussiano e o limite de sensibilidade da margem ϵ , os quais são valores que devem ser determinados (Seção 4.2). A estimação deste parâmetros afeta diretamente o rendimento do SVMr (USTÜN AND MELSEN, 2005).

Muitos métodos de otimização estão sendo estudados para se determinar os parâmetros de ajuste como o método de otimização de enxame de partículas (PSO) e o algoritmo genético (GA), porém, estes métodos não apresentam ainda resultados satisfatórios, devido a sua complexidade e a sua capacidade de busca global limitada (LIANGCHENG, HUIZHONG, 2009).

O filtro de partículas foi utilizado para se obter uma estimação destes parâmetros de ajuste, dado que o SVMr pode ser considerado como um modelo de espaço de estado geral, ajustando os parâmetros expostos anteriormente, definindo-os como estados. Algumas pesquisas estão sendo realizadas para estimação destes parâmetros com filtro de partículas para diferentes problemas (GENCAGA et al., 2006).

4.3.1 Modelo de espaço de estados de SVMr

A determinação do parâmetro de regularização C , do limite de sensibilidade da função de perda ϵ e da largura da distribuição gaussiana do “kernel” σ foram obtidas neste trabalho de pesquisa. Os três parâmetros tem efeito direto sobre o desempenho do treinamento do SVMr. Para se aplicar o filtro de partículas para estimação desses parâmetros, adotou-se por uma representação de espaço de estados da SVMr, dada na seguinte Equação,

$$\begin{aligned} p_k &= p_{k-1} + d_{k-1} \\ z_k &= h_k(x_k, p_k) + v_k' \end{aligned} \quad (4.13)$$

onde k é o index, x é a entrada do SVMr (velocidade de rotação, velocidade de translação e força axial - para treinamento e validação) e as saídas (Propriedades Mecânica-treinamento e validação). h_k representa o modelo SVMr. p é o vetor de parâmetros definido como $p = [C, \varepsilon, \sigma]$ no qual é considerado como um estado do sistema e precisa ser estimado. d_{k-1} e v_k são os ruídos do processo do sistema e da medição, sendo uma distribuição gaussiana com covariância R e Q, respetivamente (LIANGCHENG, HUIZHONG, 2009).

4.3.2 Implementação do Filtro de Partículas para o Treinamento de SVMr

O modelo SVMr é uma classe de método realizado pelos dados que requerem informação de entrada e saída do sistema. Assim, para aplicar o filtro de partículas para treinar o SVMr precisa-se um modelo inicial do SVMr h_0 e parâmetros iniciais p_0 . Este valor inicial foi obtido pelo método de validação cruzada (Seção 3.3). Para a implementação da densidade foi utilizado a distribuição a priori $p(z_k | p_{k-1}^i)$ (probabilidade que expressa a incerteza de um valor antes de qualquer medida) tendo uma forma simples para a atualização dos pesos $w_k^i \propto w_{k-1}^i p(z_k | p_k^i)$ (THRUN, 2013) e (PITT et al, 1999).

Da mesma forma, é importante decidir o número apropriado de partículas. Normalmente, é determinado pelas dimensões do estado, da linearidade do sistema e as propriedades dos parâmetros desconhecido. Em geral, mais partículas podem melhorar o seu desempenho (THRUN, 2013). Na prática, a decisão do número de partículas é realizada através de experimentos.

Os passos para o processo de treinamento de SVMr baseado em filtro de partículas são os seguintes:

1. Decidiu-se pelo número de partículas N empiricamente (100 200 500 e 1000). Colocou-se uma covariância no vector de estados p_k ($C = 0.6$, $\varepsilon = 0.03$ e $\sigma = 0.3$) e $\sigma_d = 0.4$ sendo o ruído do sensor.

2. Pré-processamento das entradas x e as saídas y , com 24 amostras recolhidas pelos ensaios de tração dos corpos de prova.
3. Ajustar o valor inicial dos parâmetros do vector p_o . Dada a quantidade tão pequena de dados, foi utilizado o método “*Leave-one-out*”, obtendo uma predição do erro de validação e treinamento. Foram definidos 19 dados para o treinamento-validação, os outro 5 dados restantes foram utilizados para o teste do algoritmo.

$$h_k(x, p_k) = svmr(x, p_k) \quad (4.14)$$

obtido p_k com validação cruzada

4. Multiplica-se uma distribuição uniforme aleatória à covariância do passo 1. Assim, somando a covariância para cada partícula, e gerando novas partículas (LIANGCHENG, HUIZHONG, 2009).

$$Part_i = Part_i + Cov \quad (4.15)$$

*Cov * randn*

5. Os sensores do sistema são as predições em relação à saída observada, tendo tanto sensores como dados disponíveis, ou seja, os sensores foram os erros de predição de cada amostra. Assim, as medições dos sensores são os erros de predição tanto dos dados de treinamento como dos dados de validação, porque ao se realizar somente com os dados de treinamento é possível que se generalize somente para os dados de treinamento. Foi utilizado a fusão de sensores com ajuda do erro quadrático médio (MSE), tanto para treinamento como para validação, tendo como resultado dois sensores definitivos MSE_{train} e MSE_{val} .
6. Para o grupo dos dados, calcular os pesos das partículas w_k . Para isto, é necessário obter $p(z_k | p_k^i)$ dada pela Equação (THRUN, 2013),

$$p(z_k | p_k^i) = \frac{1}{\sqrt{2\pi}\sigma_d} \exp\left(-0.5 * \left(\frac{z}{\sigma_d}\right)^2\right). \quad (4.16)$$

$$w_k^i \propto w_{k-1}^i p(z_k | p_k^i)$$

7. Gerar novas partículas p_k usando re-amostragem. Foi utilizado o algoritmo da roleta.

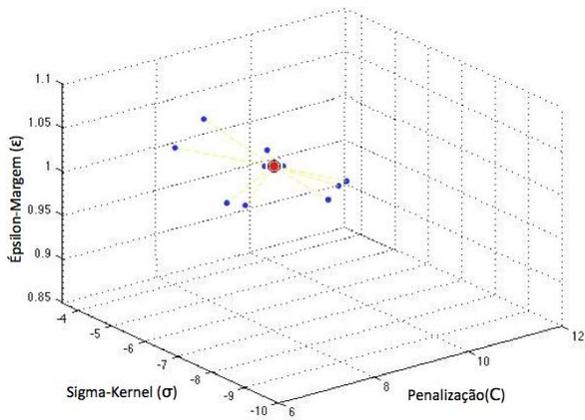
8. Treinar o modelo SVMr com o vetor p_k atualizado, sendo a atualização a média do valor das partículas.
9. Incrementar k e voltar ao passo 4.

As iterações k são até a convergência do erro, o p_k obtido da última vez que foi feito o algoritmo é o vetor ótimo de parâmetros e h_k é o modelo final do SVMr.

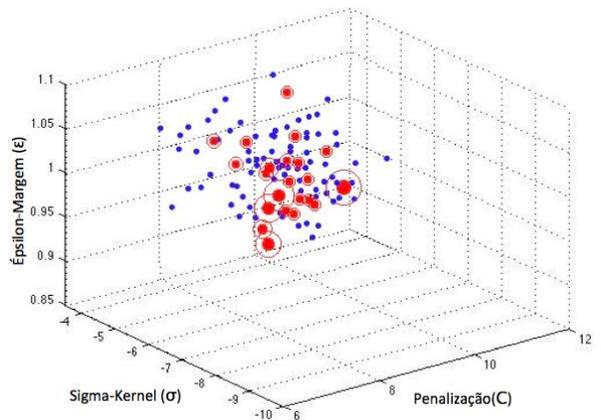
4.3.3 Esquema do Filtro de Partículas em SVMr

Na pesquisa foi desenvolvido o modelo de filtro de partículas para o sistema SVMr da solda FSW das propriedades mecânicas de tração da solda FSW. No FP utilizou-se função gaussiana para a função de distribuição dos sensores. O processo de FP é usado para determinar os estados ocultos do SVMr (C , ε e σ). A estratégia básica para o desenvolvimento do modelo baseado no SVMr com FP do comportamento da FSW é ajustar os parâmetros que afetam implicitamente o algoritmo com os resultados das séries de experimentos sobre as amostras. Desta forma, foi possível achar os valores ótimos destes parâmetros, com uma distribuição aleatória.

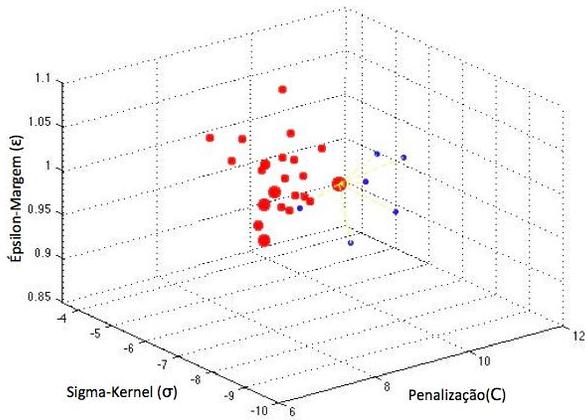
Na Figura 4.10 mostra-se as diferentes etapas do filtro de partículas para cada iteração. Por meio da validação cruzada foi obtido o ponto inicial como é mostrado na Figura 4.10 (a), onde foi aplicada uma covariância para cada valor que é composta cada partícula (LIANGCHENG, HUIZHONG, 2009). Depois aplicou-se $p(z_k | p_k^i)$ para obter os pesos das partículas sendo escolhidas somente as partículas com maior probabilidade de comportar-se como o modelo real, como é mostrada na Figura 4.10 (b). Após obter as partículas com maior probabilidade, deve-se gerar partículas novas, obtidas por meio de uma covariância dos parâmetros dos estado ocultos, e assim, aplicar o passo anterior da Figura 4.10 (b), como é mostrado na Figura 4.10 (c) e na Figura 4.10 (d). Por último se realiza uma média das partículas que sobreviveram, dado que a somatória dos pesos de todas as partículas é 1, foi possível obter um dado estimado dos parâmetros, e assim por diante foi feito para todas as iterações, como são mostradas na Figura 4.10 (e) e na Figura 4.10 (f).



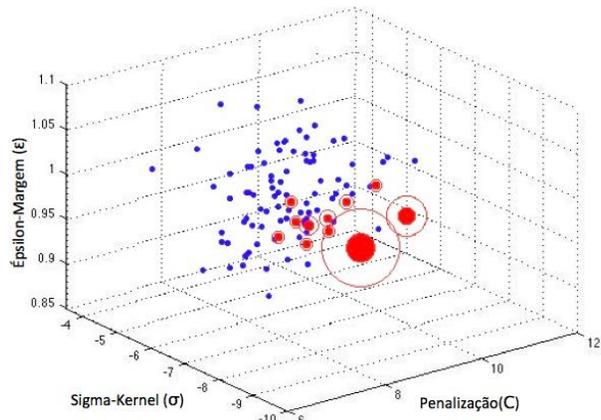
(a)



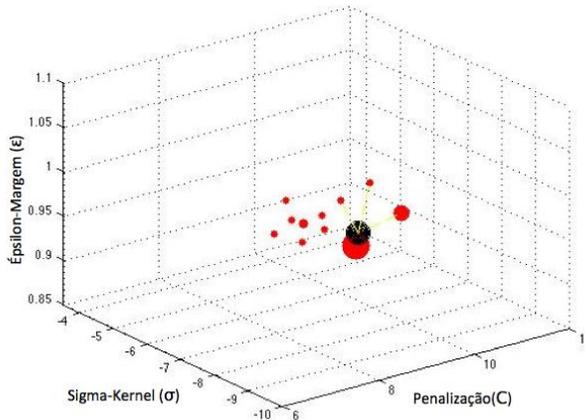
(b)



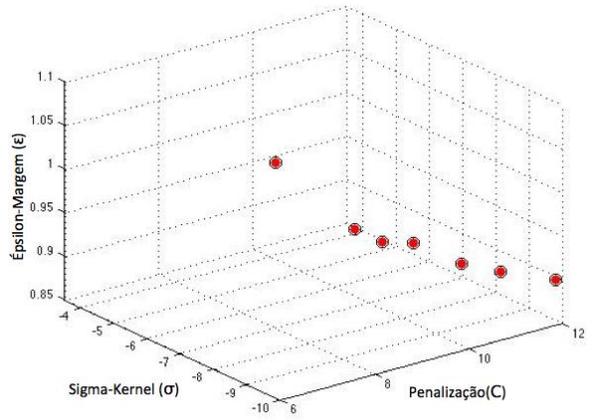
(c)



(d)



(e)



(f)

Figura 4.10 Esquema do Filtro de Partículas para SVMr: (a) Ponto Inicial Aplicando Covariância; (b) Partículas que Sobrevieram por seus Pesos; (c) Covariância de cada Partícula Sobrevivente; (d) Partículas que Sobrevieram na Seguinte Iteração; (e) Ponto Médio do Valor $[C, \sigma, \epsilon]$; (f) Convergência dos Valores $[C, \sigma, \epsilon]$.

Capítulo 5 .

RESULTADOS

Neste capítulo serão aplicados os diferentes algoritmos para obtenção do modelo de predição dos parâmetros de FSW mencionado nos capítulos anteriores, analisando-se os métodos que melhor representam o comportamento do modelo de solda por FSW. Para isto realizou-se inicialmente os ensaios de tração para se obter as saídas desejadas como as propriedades mecânicas (APÊNDICE C), obtendo um modelo para cada saída, eliminando os que não cumprem alguns critérios e com isso assegurando o comportamento do modelo (APÊNDICE B).

5.1 Planejamento Fatorial Completo 2³

As técnicas de planejamento de experimentos e análise estatística dos resultados deste trabalho podem ser encontradas em Montgomery (2009), que é uma publicação clássica nesta área (APÊNDICE B).

As técnicas baseadas em planejamento de experimentos são úteis no estudo de problemas para se determinar as variáveis que exercem maior influência no desempenho de um determinado processo. Assim, foi possível avaliar e comparar os diferentes modelos e selecionar os parâmetros do processo que exercem maior influência (APÊNDICE B).

5.1.1 Planejamento de Experimentos e Matriz de Testes

As técnicas de planejamento de experimentos requerem a realização de experimentos e testes nas situações de interesse (APÊNDICE C). Particularmente no caso do processo FSW, os efeitos causados pelos parâmetros do processo nas propriedades da peça soldada caracterizam um problema desta categoria, onde a realização de um planejamento de experimentos se justifica no sentido de ter resultados baseados em dados experimentais e que contribuem para o desenvolvimento de modelos de previsão que evitam a necessidade de testes futuros.

O planejamento fatorial completo começa com uma série inicial de experimentos, com o objetivo de definir as variáveis de entradas envolvidas no experimento e os níveis importantes, ou

seja, a faixa de variação das variáveis de entradas selecionadas. Realizou-se os experimentos com várias amostras para se obter uma estimativa do erro experimental (a estimativa desse erro é básica para verificar se as diferenças observadas nos dados são estatisticamente diferentes) e aumentar a confiabilidade dos resultados (a média da amostra é usada para estimar o efeito de um fator no experimento).

Foi utilizado um planejamento fatorial completo 2^3 para estudar os efeitos dos seguintes parâmetro nas propriedades mecânicas da solda obtida pelo processo FSW nos ensaios de tração: velocidade da rotação da ferramenta (VR), velocidade de translação da ferramenta (VT) e força axial da ferramenta sobre a junta soldada (FA). Os valores dos parâmetros das entradas de referência foram dados pela empresa Embraer [VTr, VRr e FAr] (100%). As matrizes do planejamento foram realizadas com 5 repetições para a matriz 1 (Tabela 5.1) e 10 repetições para as matrizes 2 (Tabela 5.2) e 3 (Tabela 5.3) em cada ponto, incluindo o ponto central na matriz 3 para observar o fator de curvatura, totalizou-se 40 ensaios para matriz 1, 80 ensaios para a matriz 2 e 90 ensaios para matriz 3. Os ensaios estão de acordo com a norma ASTM E8M (ASTM, 2012).

Tabela 5.1 Níveis das Variáveis de Entrada para os Ensaios – Matriz 1.

Níveis	VR (RPM)	VT (mm/min)	FA (kN)
-1	94.1%	90%	95.7%
1	105.8%	110%	104.2%

Tabela 5.2 Níveis das Variáveis de Entrada para os Ensaios – Matriz 2.

Níveis	VR (RPM)	VT (mm/min)	FA (kN)
-1	88.2	75%	91.4%
1	111.7%	125%	108.5%

As análises preliminares realizaram-se com a faixa de estudo mostrada na Tabela 5.1 e com sentido de explorar uma faixa maior de influência dos parâmetros de entrada, foi também estudada a faixa mostrada na Tabela 5.2. Por meio de análise Estatística foi determinado que para

ambos casos, as variáveis de entrada não apresentaram efeitos significativos sobre as respostas (APÊNDICE B).

Finalmente, foi selecionada uma faixa de estudo (Tabela 5.3), na qual as variáveis de entrada apresentam efeitos significativos nas respostas. O nível inferior -1 de cada variável representa um valor menor da linha base, o ponto central 0 é a linha base e o nível superior 1 corresponde a valores maiores da linha base.

Tabela 5.3 Níveis das Variáveis de Entrada para os Ensaio – Matriz 3.

Níveis	VR (RPM)	VT (mm/min)	FA (kN)
-1	82.3%	50%	87.2%
0	100%	100%	100%
1	117.6%	150%	112,7%

A Tabela 5.4 apresenta as combinações dos testes do planejamento fatorial completo considerando os níveis -1 e +1. Então tomando como exemplo, o caso do experimento 1, onde é lido VR – -1/ VT – -1/ FA – -1 entende-se de acordo com a tabela dos níveis de interesse , VR – 82.3% [rpm] / VT – 50% [mm/min] / FA – 87.2 [kN].

Tabela 5.4 Fatorial Completo de dois Níveis.

Experimentos	Fatores		
	VR (RPM)	VT (mm/min)	FA (kN)
1	-1	-1	-1
2	1	-1	-1
3	-1	1	-1
4	1	1	-1
5	-1	-1	1
6	1	-1	1
7	-1	1	1
8	1	1	1
9	0	0	0

Para realizar a análise estatística, levaram-se em conta os resultados obtidos pelos gráficos de Paretos (se as entradas influem nas respostas, p-value); análise de variância (ANOVA) (APÊNDICE B) (se os fatores do modelo tem uma relação entre os valores observados, R^2) e teste de Fisher (se o modelo esta bem ajustado, F) (MONTGOMERY, 2009).

5.1.2 Análise Estatística

O software comercial STATISTICA 7.0 foi utilizado para a análise estatística (APÊNDICE B) dos resultados. Na solda por FSW foram determinados os efeitos da velocidade de rotação da ferramenta, velocidade de translação da ferramenta e força axial da ferramenta definidas a partir do planejamento experimental 2^3 , sobre as propriedades mecânicas por meio dos ensaios de tração em corpos de provas.

A Tabela 5.5 apresenta as saídas das propriedades mecânicas ao longo dos 9 experimentos, com seus respectivas médias e desvios padrão obtidas através de repetições dos ensaios de tração, incluindo a linha base ou ponto central da matriz 3 (Tabela 5.3).

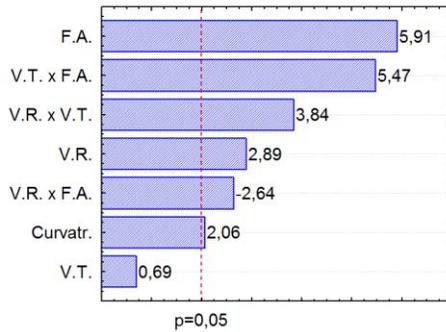
Foram variados os parâmetros (VR, VT e FA) do processo de solda por FSW para o estudo das propriedades mecânicas (Tabela 1.5). Os valores de tensão de escoamento (TE), tensão última (TU) e tensão de falha (TF) são dados em pascais, as respostas de alongamento são dadas pela porcentagem do aumento do comprimento do corpo de prova. A Figura 5.1 apresenta gráficos de Paretos com os efeitos exercidos pela velocidade de rotação, velocidade de translação, força axial e a interação dos mesmos sobre cada um dos parâmetros de resposta dos ensaios de tração (APÊNDICE B).

Conforme mostrado na Figura 5.1, onde foi utilizado o nível de confiança de 95% ($p \leq 0,05$), foi observado que a velocidade de translação, dentro do intervalo estudado (50% a 150% do valor de referencia VTr), não apresentou efeito significativo sobre a tensão última e tensão de falha. O fator da velocidade de rotação dentro do intervalo estudado (82.3% a 117.6% do valor de referencia VRr), não apresentou efeitos significativos em tensão de escoamento e alongamento de falha, e o fator da força axial dentro do intervalo estudado (87.2% a 112.7% do valor de referencia FAr) não apresentou efeito significativo em alongamento de escoamento.

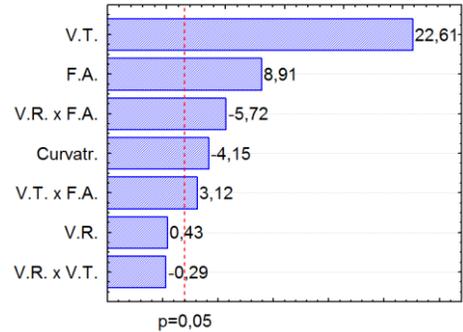
Tabela 5.5 Valores Reais das Variáveis e Parâmetros Mecânicos de cada Experimento do Planejamento 23.

Experimento	Variáveis			Respostas					
	VR (RPM)	VT (mm/min)	FA (kN)	T.U. (MPa)	T.E. 0.2% (MPa)	M.E. (GPa)	A.F. (%)	A.E. 0.2% (%)	T.F. (Mpa)
1	82.3%	50%	87.2%	460,5 ± 5,19	333,2 ± 4,96	71,2 ± 1,69	14,5 ± 0,45	0,67 ± 0,01	452,6 ± 7,11
2	117.6%	50%	87.2%	458,4 ± 11,89	338,2 ± 2,66	72,2 ± 2,86	9,41 ± 2,90	0,67 ± 0,02	456,1 ± 10,05
3	82.3%	150%	87.2%	422,7 ± 5,21	347,4 ± 1,84	68,7 ± 0,82	4,02 ± 0,46	0,72 ± 0,01	421,9 ± 5,26
4	117.6%	150%	87.2%	463,1 ± 13,49	350,9 ± 2,64	70,3 ± 1,42	7,88 ± 1,70	0,71 ± 0,01	459,7 ± 12,78
5	82.3%	50%	112,7%	463,2 ± 13,91	341,7 ± 2,91	72,4 ± 2,72	10,5 ± 2,82	0,68 ± 0,02	460,8 ± 13,76
6	117.6%	50%	112,7%	471,1 ± 5,51	337,7 ± 3,09	73,9 ± 2,47	11,6 ± 4,36	0,66 ± 0,02	468,5 ± 4,64
7	82.3%	150%	112,7%	479,2 ± 2,78	359,6 ± 3,28	70,4 ± 1,74	10,9 ± 1,17	0,72 ± 0,02	475,6 ± 2,58
8	117.6%	150%	112,7%	485,4 ± 2,95	355,8 ± 3,76	70,7 ± 0,95	11,21 ± 1,34	0,72 ± 0,01	483,7 ± 3,08
9	100%	100%	100%	471,6 ± 2,32	341,2 ± 0,79	69,2 ± 1,40	14,40 ± 2,07	0,70 ± 0,01	466,7 ± 3,26

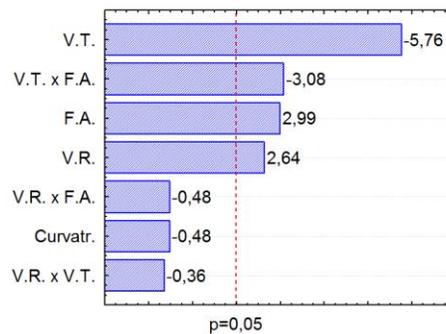
(T.U.) Tensão última; (T.E.) Tensão de escoamento; (M.E.) Módulo de elasticidade; (A.F.) Alongamento da falha; (A.E.) Alongamento do escoamento; (T.F.) Tensão da falha.



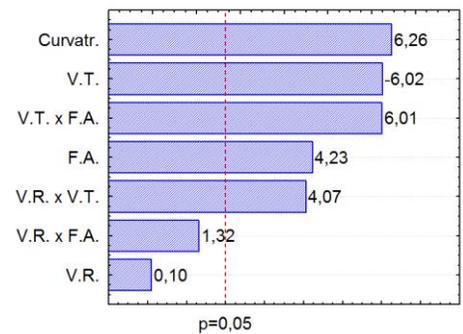
Tensão última (T.U.)



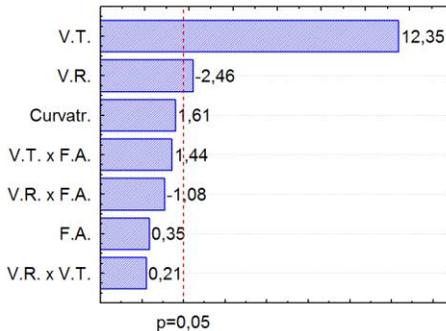
Tensão de escoamento 0.2% (T.E.)



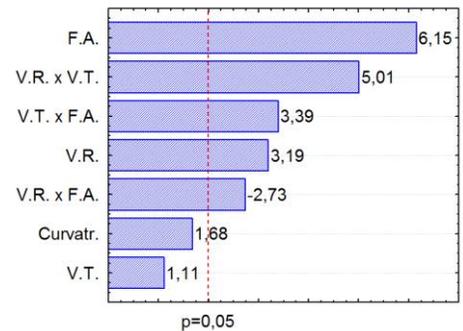
Módulo de elasticidade (M.E.)



Alongamento da falha (A.F.)



Alongamento de escoamento 0.2% (A.E.)



Tensão da falha (T.F.)

(V.R.) Velocidade de rotação; (V.T.) Velocidade de translação; (F.A.) Força Axial.

Figura 5.1 Efeitos dos Fatores sobre os Parâmetros de Resposta.

Conforme mostrado na Figura 5.1, a tensão última, a tensão de escoamento e o alongamento de falha apresentaram um fator de curvatura significativo, entretanto eles não são

bem representados por modelos lineares. Portanto, foram desenvolvidas por métodos não lineares em resultados posteriores (ANEXO i e ANEXO ii).

Pela análise de variância (ANOVA) (APÊNDICE B), para o módulo de elasticidade e a tensão da falha apresentada na Tabela 5.6, obteve-se R^2 os valores de 31% e 48%, respectivamente, valores não aceitáveis para este processo, considerando-se assim, que estes modelos matemáticos não serão preditivos.

Tabela 5.6 ANOVA para as Respostas Tensão Última, Tensão de Escoamento, Módulo de Elasticidade e Alongamento da Falha.

Respostas	Fonte de Variação	SQ	GL	Quadrado médio	F_{cal}	F_{tab}	R^2
Alongamento de escoamento (A.E.)	Regressão	0,048	6	0,008	26,55	2,21	0,69
	Resíduos	0,02	83	0,0003			
	Total	0,073	89				
Módulo de elasticidade (M.E.)	Regressão	172,7	6	28,783	7,51	2,21	0,31
	Resíduos	318,289	83	3,8348			
	Total	490,989	89				
Tensão da falha (T.F.)	Regressão	22629,41	6	3771,57	15,02	2,21	0,48
	Resíduos	20841,16	83	251,098			
	Total	43470,56	89				

Pela análise ANOVA para alongamento de escoamento (AE) (Tabela 5.6), não foram significativos os termos lineares da força axial (FA) e das interações entre os três fatores (VTxFA) 1.44, (VRxFA) -1.08 e (VRxVT) 0.21 (Figura 5.1), tendo sido incorporados aos resíduos mostrados no item resíduos da Tabela 5.6 (APÊNDICE B). Para o caso do teste de Fisher, o F_{calc} foi maior que o F_{tab} , considerando-se assim, o modelo matemático significativo (MONTGOMERY, 2009) e a porcentagem de variação (R^2) de 0,69, valor aceitável para este processo (APÊNDICE A), pode-se portanto concluir que o modelo de (AE) se ajusta bem aos dados experimentais. A Tabela 5.7 apresenta um resumo da análise ANOVA para alongamento de escoamento.

Tabela 5.7 Preditividade e Significância do Modelo Matemático do Alongamento de Escoamento.

Respostas	Efeito significativo	R ²	Modelo preditivo	F _{cal} >F _{tab}	Modelo significativo
Alongamento de escoamento (A.E.)	Velocidades de rotação e translação	0,66	Sim	12.01	Sim

A Equação (5.1) apresenta o modelo matemático construído para o alongamento de escoamento, obtido pelo software comercial STATISTICA 7.0,

$$[A.E.] = 0,69 - 0,004737 * [V.R.] + 0,023787 * [V.T.], \quad (5.1)$$

conforme as notações seguintes,

[A.E.] = Alongamento de escoamento 0.2% (%),

[V.R.] = Velocidade de rotação da ferramenta (RPM),

[V.T.] = Velocidade de translação da ferramenta (mm/min).

Uma forma de análise é através de ferramentas gráficas como as superfícies de resposta. Contudo, para isso a entrada é limitada a dois parâmetros, caracterizando dois eixos de entrada, e a resposta caracteriza o terceiro eixo, obtendo-se assim uma representação tridimensional.

A superfície de resposta (Figura 5.2) gerada pelo modelo descreve o efeito das velocidades de rotação e translação (com a linha base no ponto central) sobre o parâmetro da solda FSW (Alongamento de escoamento). O aumento da velocidade de translação (VT) resultou em incremento no alongamento de escoamento, pelo contrário, o aumento da velocidade de rotação (VR) resultou em sua diminuição.

Por meio da análise estatística foi possível observar que a resposta de alongamento de escoamento (nas faixas dos parâmetros estudadas - matriz 3, Tabela 5.3) foi a única que conseguiu apresentar um modelo linear significativo e preditivo (Tabela 5.7), no qual, somente as velocidades apresentaram efeito significativo sobre a resposta. As respostas de módulo de

elasticidade e tensão de falha não representam um modelo linear preditivo, nem apresentaram fator de curvatura na faixa estudada.

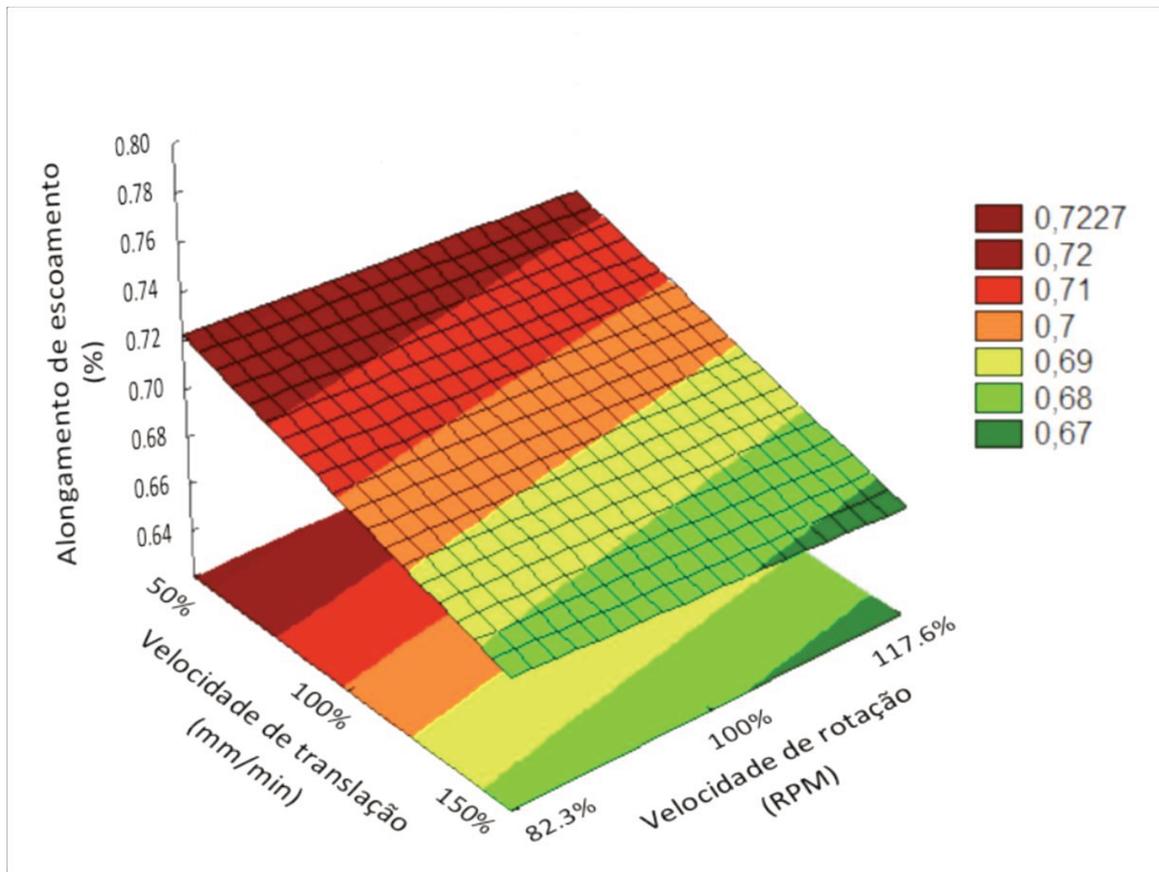
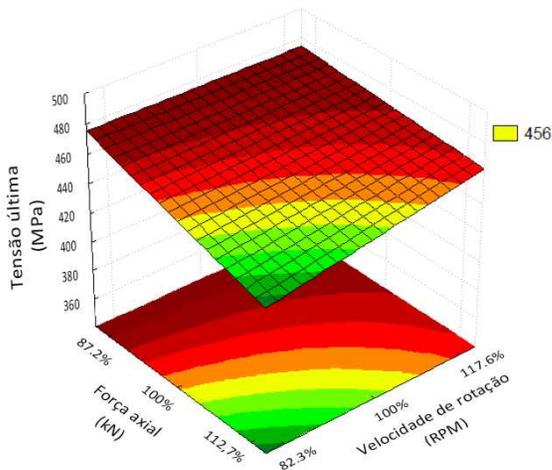
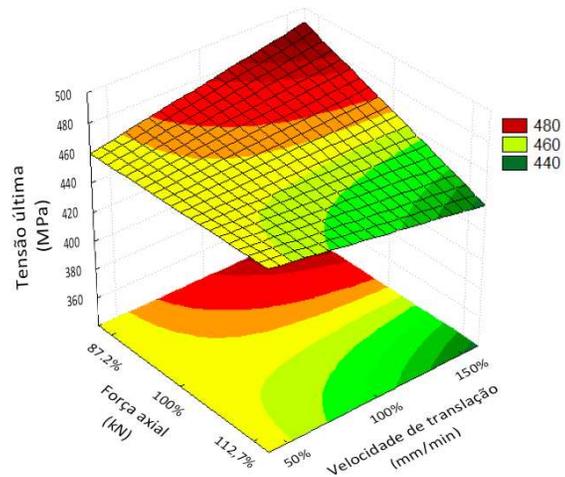


Figura 5.2 Superfície de Resposta de Alongamento de Escoamento.

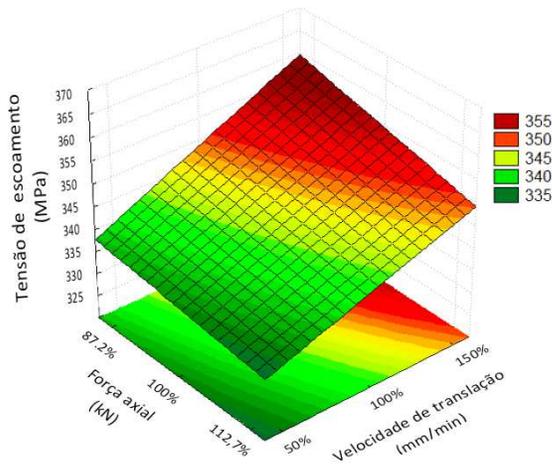
Finalmente, as respostas tensão de escoamento, tensão última e alongamentos de falha apresentam o fator de curvatura no qual indica que pode ser representadas com modelo não lineares (ANEXO i e ANEXO ii). Na Figura 5.3 são mostrados os resultados nos modelos um polinômio de grau 2, os quais apresentaram o fator de curvatura.



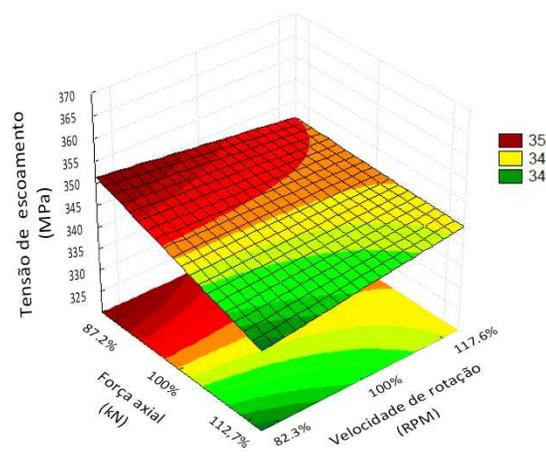
(a) Tensão última (F.A. - V.R.)



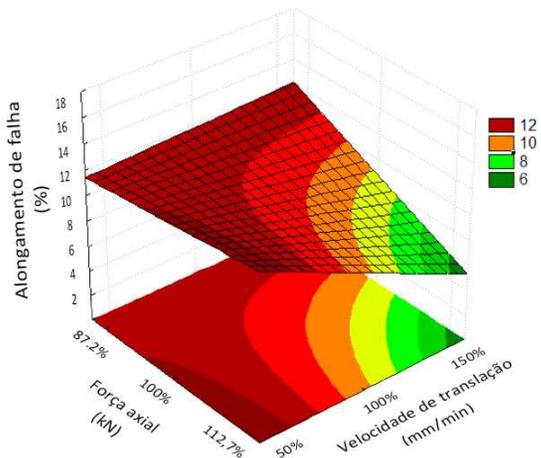
(b) Tensão última (F.A. - V.T.)



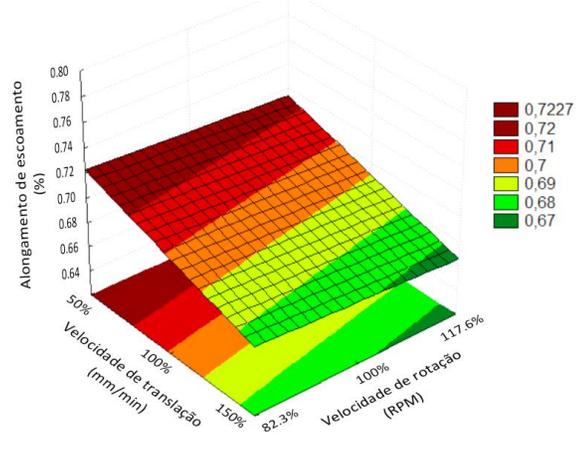
(c) Tensão de escoamento (F.A. - V.T.)



(d) Tensão de escoamento (F.A. - V.R.)



(e) Alongamento de falha (F.A. - V.T.)



(f) Alongamento de falha (V.T. - V.R.)

Figura 5.3 Superfícies de Respostas que Apresentaram Fator de Curvatura: (a) T.U. Fatores Significativos; (b) T.U. Relação ao Fator não Significativo; (c) T.E. Fatores Significativos; (d) T.E. Relação ao fator não Significativo; (e) A.F. Fatores Significativos; (f) A.F. Relação ao Fator não Significativo.

Na Figura 5.3 (a) pode observar que a tensão última (TU) diminui quando os fatores significativos da faixa estudada são baixos, força axial (87.2% a 112.7% do valor de referência FAR) e velocidade de rotação (82.3% a 117.6% do valor de referencia VRr). O fator da velocidade de translação com valores altos influi no resultado através da sua interação com a força axial obtendo tensão maior quando (FA) aumenta e tensão menor quando (FA) diminui, conforme mostra a Figura 5.3 (b).

Na tensão de escoamento foi possível observar que o fator significativo de velocidade de translação (50% a 150% do valor de referencia VTr) tem influência na resposta, aumentando a (VT) aumenta a tensão (TE), pelo contrário (TE) diminui com o fator (FA) que tem uma pequena significância na faixa estudada Figura 5.3 (c). Também foi possível observar que a resposta (TE) aumenta em valores altos de (FA) e valores baixos de (VR) conforme mostra a Figura 5.3 (d).

Finalmente, na Figura 5.3 (e) observou-se que quanto menor valor de força axial (F.A) e maior valor de velocidade de translação (VT) (fatores significativos da resposta de alongamento de falha (AF)) o alongamento de falha vai ter uma tendência de descida. Eventualmente foi colocado na Figura 5.3 (f) o alongamento de falha com respeito às velocidades de rotação e translação, obtendo uma diminuição da resposta (AF) quando a (VT) tem um crescimento e a (VR) tem valores baixos.

Como resultado, as respostas (tensão de escoamento, tensão última e alongamento de falha) foram feitas com métodos de modelos não lineares mostrados em capítulos anteriores e (ANEXO i e ANEXO ii), já que foi possível observar curvaturas na Figura 5.3.

5.2 Máquina de Vetores de Suporte para Regressão

5.2.1 Seleção dos Modelos do SVMr

A vantagem do SVMr é que não precisa escolher o modelo, já que o método trabalha com as amostras dadas e encontra o hiperplano que satisfaz a função (Seção 3.2 e 4.2). Isto deve-se ao fato de trabalhar com todas as amostras realizando um mapeamento de características aumentando a dimensionalidade (transformação de características). Porém, é necessário encontrar parâmetros para um bom funcionamento do SVMr, os quais são: C da regularização, σ^2 valor do

tamanho da gaussiana do “kernel” e ϵ sensibilidade da margem. Dado que o SVMr converte o modelo em uma função convexa graças aos vectores de suporte e o “kernel”, quer dizer, utiliza o risco estrutural minimizando a distância dos vetores e maximizando a margem (Seção 3.2 e 4.2) (SCHÖLKOPF, et al., 1998).

Este método SVMr foi aplicado nas saídas que apresentaram curvatura na análise Estatística (APÊNDICE B), sendo tensão de escoamento (TE), tensão última (TU) e alongamento (AF). Também foram comparadas com os métodos de regressão polinomial (RP) (ANEXO i) e de redes neurais (RNA) (ANEXO ii). Para os treinamento dos parâmetros utilizou-se o método de deixar um fora “leave-one-out” de validação cruzada (Seção 3.3), realizando N cálculos de erro, um para cada dado.

5.2.2 Evolução dos Modelos Finais do SVMr

Como foi explicado anteriormente, tem-se o modelo, onde foi utilizado o conjunto de treinamento mais o conjunto de validação, quer dizer que treinou-se com os 19 dados de entrada (Seção 3.3). Para o teste foram utilizados os 5 dados de entrada restantes. Os resultados obtidos da (TE), (TU) e (AF) são mostrados na Tabela 5.8 e Tabela 5.9 .

Tabela 5.8 Parâmetros e (MSE) Finais Obtidos por o Método Validação Cruzada para SVMr.

	C	σ^2	ϵ	Treinamento (MSE)	Validação (MSE)
Tensão Escoamento	64	0.0312	2	1.4719	4.7220
Tensão Última	64	0.25	0	12.7534	60.3135
Alongamento	8	0.5	4	3.9618	5.1610

Na tensão de escoamento, os valores de (R^2) 0.9002 e 0.9020, e os valores de (MSE) 1.4809 (MPa) e 1.9029 (MPa) do treinamento e validação respectivamente, indicam que o modelo do SVMr foi totalmente treinado para reconhecer qualquer saída dentro do conjunto de dados disponíveis. Os erros estão no faixa factível e os valores dos coeficientes de determinação estão muito perto de um para obtenção de uma boa predição (APÊNDICE A).

Na tensão última, os valores de (R^2) 0.8575 e 0, e os valores de (MSE) 13.4296 (MPa) e 54.2572 (MPa) do treinamento e validação respectivamente, indicam que o modelo selecionado do SVMr foi totalmente treinado, mas não reconhece muito bem a saída dentro do conjunto de dados disponíveis. Isto deve-se, pelos valores dos erros que são elevados, e o valor do coeficiente de determinação da validação é zero, obtendo uma predição não tão boa para o algoritmo (APÊNDICE A).

No alongamento, os valores de (R^2) 0.3998 e 0.2977, e os valores de (MSE) 3.9102% e 1.5084% do treinamento e validação respectivamente, indicam que o modelo selecionado da SVMr foi totalmente treinado. Os valores dos erros são muito elevados sendo que o alongamento máximo tinha sido de 16%, o que pode indicar que não se tem uma boa predição, e os valores dos coeficientes de determinação perto do zero, por conseguinte a regressão tem pouca correlação (APÊNDICE A).

Tabela 5.9 Erros (MSE) e Coeficiente de Determinação (R^2) de Treinamento-Validação e Teste para os Modelos Finais da SVMr.

	C	σ^2	ϵ	Treina.-Val. (MSE)	Teste (MSE)	Treina.-Val. (R^2)	Teste (R^2)
Tensão Escoamento	64	0.0312	2	1.4809	1.9029	0.9002	0.9020
Tensão Última	64	0.25	0	13.4296	54.2572	0.8575	0
Alongamento	8	0.5	4	3.9102	1.5084	0.3998	0.2977

A tensão de escoamento foi a única resposta que obteve um coeficiente de determinação (R^2) maior ao 70% tanto no treinamento-validação como nos dados de teste.

5.2.3 Curvas de Aprendizagem do SVMr

Na Figura 5.4 mostra-se as curvas de aprendizagem. Em todas as saídas do algoritmo apresentam alta variância, o que significa que tem uma grande diferença entre os erros de treinamento e validação, e por conseguinte uma boa solução é obter maior conjunto de dados para o treinamento do algoritmo (Seção 3.3.4).

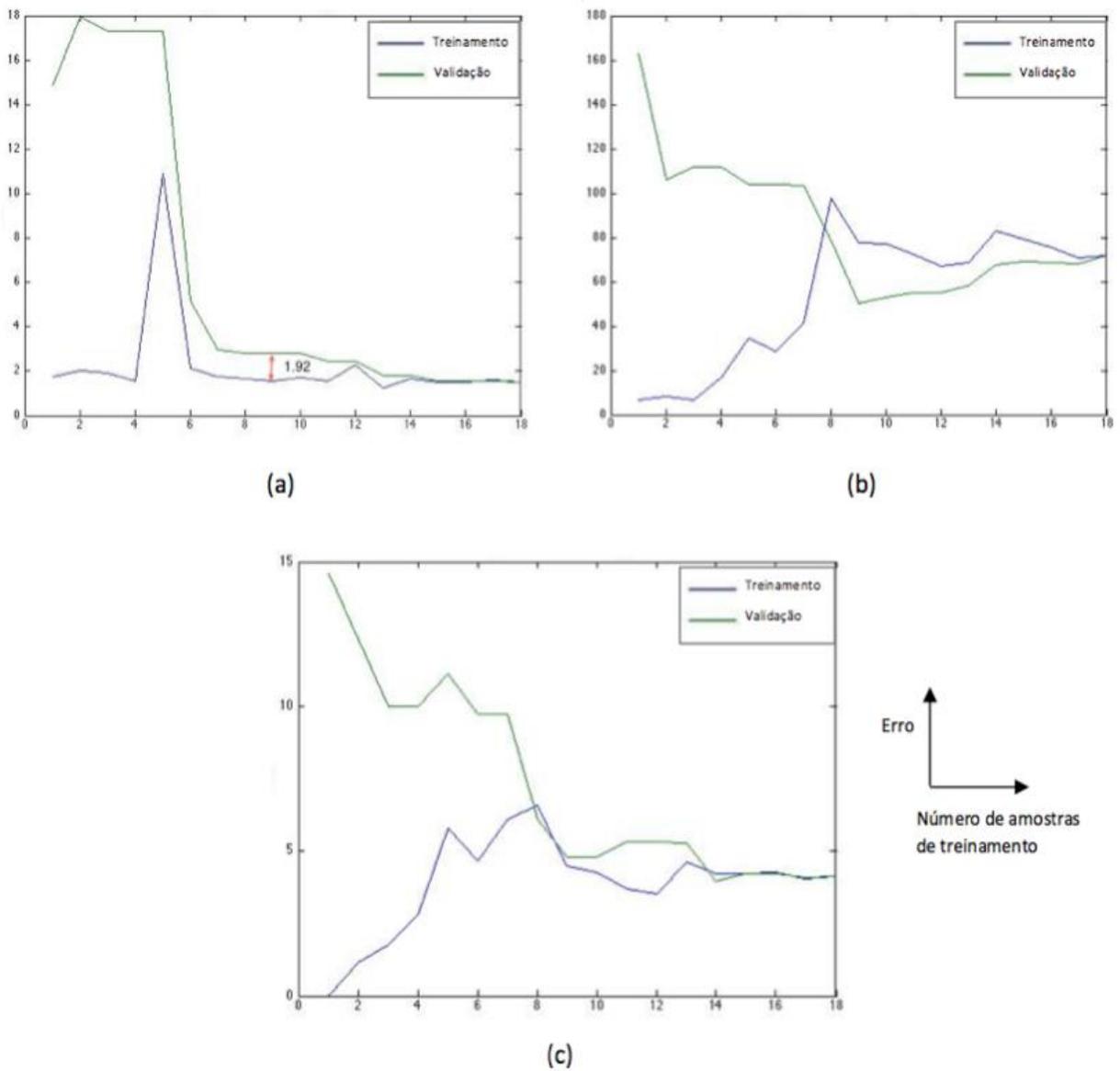


Figura 5.4 SVMr - Curvas de Aprendizagem MSE – Sample: (a) Tensão de Escoamento (b) Tensão Última (c) Alongamento.

Na Figura 5.4 (a) é possível observar que a diferença entre o erro de treinamento e de validação é de 1.92 (MPa), melhor que os outros métodos (ANEXO iii e ANEXO iv), sendo uma medida pequena em relação a média da tensão de escoamento de 343 (MPa). Por conseguinte, foi possível concluir que o algoritmo tem um bom comportamento, embora, realizar um treinamento com uma maior quantidade de dados ajudaria ao algoritmo.

Para a tensão última é possível observar que não pode-se medir a diferença entre o erro de treinamento e validação, sendo uma medida considerável em relação a média da resposta de tensão de escoamento (TE) de 459(MPa) como se mostra na Figura 5.4 (b). A explicação foi que o erro de treinamento superou o erro de validação, o que significa que não tem boa generalização.

A resposta de alongamento não apresentou uma diferença tendo uma média de 10.64%. Por conseguinte, foi possível concluir que o algoritmo tem um alta bias. Para tentar solucionar este problema deve-se realizar um treinamento com uma maior quantidade de dados, dado que o SVMr aumenta a dimensionalidade dependo da quantidade dos dados.

5.2.4 Análise de Predição das Respostas da SVMr.

Foi utilizado a porcentagem de erro da média absoluta (MAPE) para as respostas finais de tensão de escoamento (TE), tensão última (TU) e alongamento (AF) (APÊNDICE A).

Na Tabela 5.10 mostra-se o erro dos modelos feitos pelo método de SVMr, tendo a tensão de escoamento um erro de 0.4112% e 0.4211% para o treinamento e teste, respectivamente, valores consideráveis bons para o modelo.

Para (TU) foi observado que o erro é de 0.4313% e 1.5178%, para treinamento e teste, respectivamente, embora sejam considerados erros bons para ter uma confiabilidade no modelo, o coeficiente de determinação (R^2) é menor ao 70%. Também observou-se que o alongamento (AF) foi que teve maior erro de todos os modelos, com um erro de 25.3566% para treinamento e 16.1198%, mostrando uma predição baixa.

Tabela 5.10 A Porcentagem do Erro (MAPE) de Treinamento e Validação para SVMr.

	C	σ^2	ϵ	Treinamento (MAPE) %	Teste (MAPE) %
Tensão Escoamento	64	0.0312	2	0.4112	0.4211
Tensão Última	64	0.25	0	0.4313	1.5178
Alongamento	8	0.5	4	25.3566	16.1198

Sendo a tensão de escoamento foi a única resposta que apresentou mais de 70% de coeficiente de determinação (Tabela 5.9), e com erros muito pequenos (Tabela 5.10) conforme mostra a Figura 5.5.

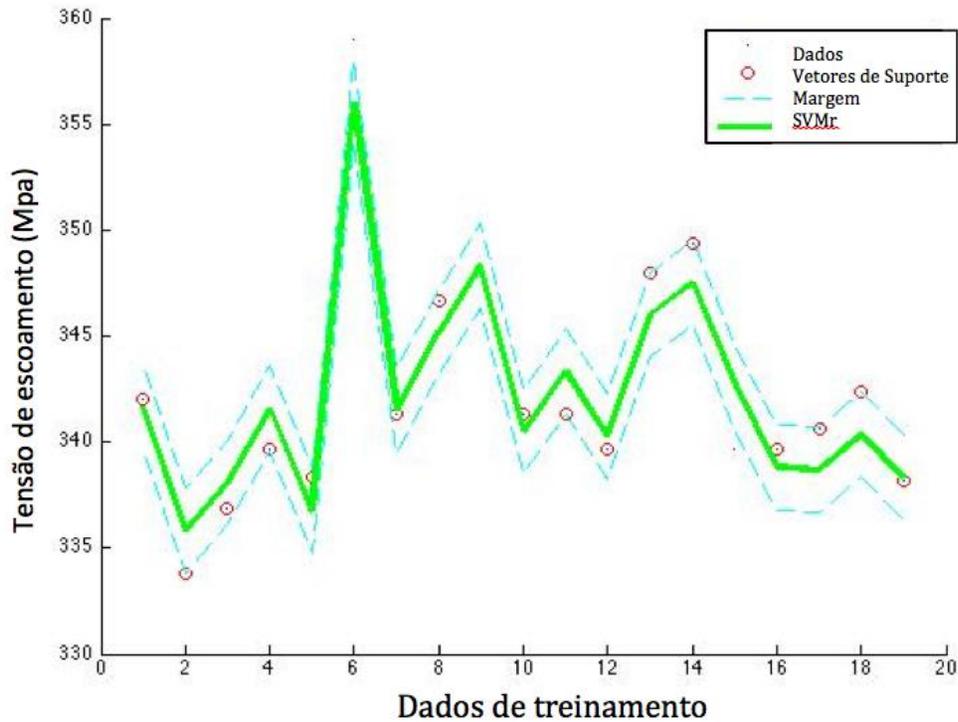


Figura 5.5 Regressão por o Método de SVM para a Resposta de Tensão de Escoamento.

Na Figura 5.5 é possível observar a margem achada da resposta de tensão de escoamento dos dados de treinamento (linha verde) e o parâmetro ϵ que mostra a sensibilidade da generalização do algoritmo (linha azuis). O valor utilizado para o parâmetro σ^2 do "kernel" e para o valor de penalização C ajudaram a encontrar os vetores de suporte (círculos vermelhos) explicados na seção 4.2.

5.2.5 Superfície de Resposta da SVMr

O resultado de Tensão de Escoamento foi analisado através de ferramentas gráficas de superfícies de resposta como foi feito na Regressão Polinomial (ANEXO iii) e nas Redes Neurais (ANEXO iv). Contudo, a entrada é limitada a dois parâmetros, caracterizando dois eixos de entrada, e a resposta caracteriza o terceiro eixo, obtendo-se assim uma representação tridimensional.

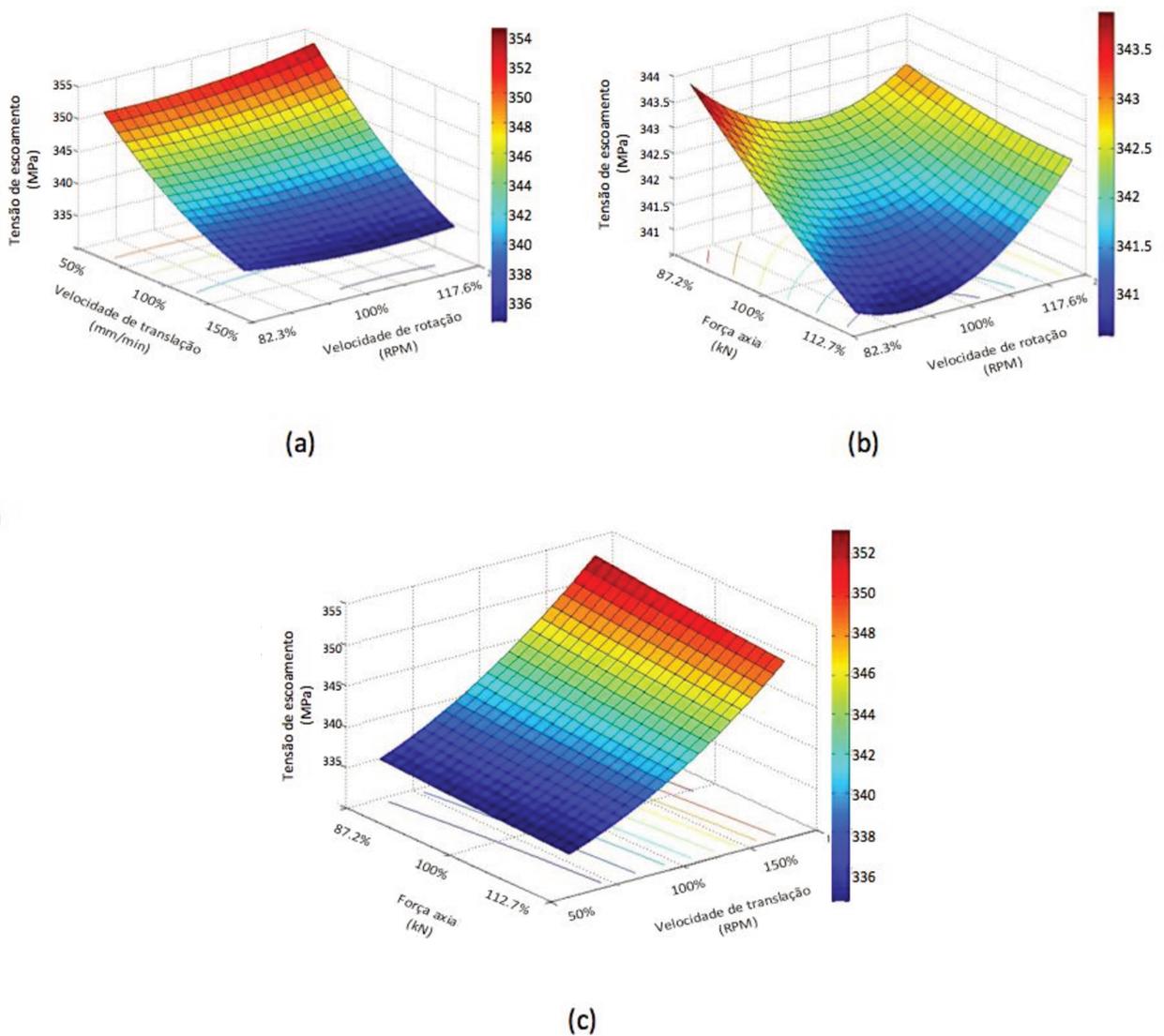


Figura 5.6 Superfície de Resposta da Tensão de Escoamento por o Algoritmo SVMr; (a) VR-VT; (b) VR-FA; (c) VT-FA.

A superfície de resposta Figura 5.6 gerada pelo modelo descreve o efeito das velocidades de rotação e translação e a força axial sobre o parâmetro de tensão de escoamento da solda por FSW. Na Figura 5.6 (a), apresentou-se o mesmo comportamento que os outros métodos (ANEXO iii e ANEXO iv), ou seja, aumento da velocidade de translação resultou em incremento na tensão de escoamento, pelo contrário, o aumento da velocidade de rotação teve pouca influência.

Na Figura 5.6 (b), observou-se que o comportamento foi similar ao apresentado em rede neural, onde o incremento de velocidade de rotação e força axial, aumenta a tensão de escoamento. Pelo contrário, o decrescimento dos dois fatores diminui a tensão de escoamento, sendo que as entrada (FA) e (VR) apresentaram pouca influência no resultado de tensão de escoamento.

Na Figura 5.6 (c) apresentou o mesmo comportamento que os métodos anteriores, ou seja, o incremento de velocidade de rotação aumenta a tensão de escoamento e o fator de força axial não tem grande influência no resultado.

5.3 SVMr ajustando os parâmetros com o Filtro de Partículas

Os resultados obtidos pelo filtro de partículas para o ajuste dos parâmetros no modelo SVMr é mostrado na Tabela 5.11 e Tabela 5.12 (referência Figura 5.7 e Figura 5.8).

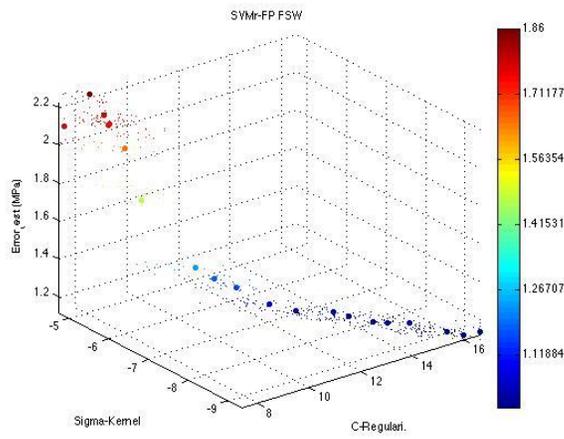
O MSE de treinamento e o MSE de validação foram os sensores utilizados para filtro de partículas, otimizando os parâmetros (C , σ^2 , ϵ). Na Tabela 5.11 mostra-se que o valor do erro melhora com respeito ao SVMr e o método de validação cruzada. Os erros tanto de treinamento como validação convergiram ao mesmo ponto, sem importar a quantidade das partículas, o que significa que somente com 100 partículas consegue-se um resultado similar ao algoritmo de 1000 partículas.

Tabela 5.11 Resultados de Treinamento-Validação para o SVMr com Filtro de Partículas com Diferentes Números de Partículas.

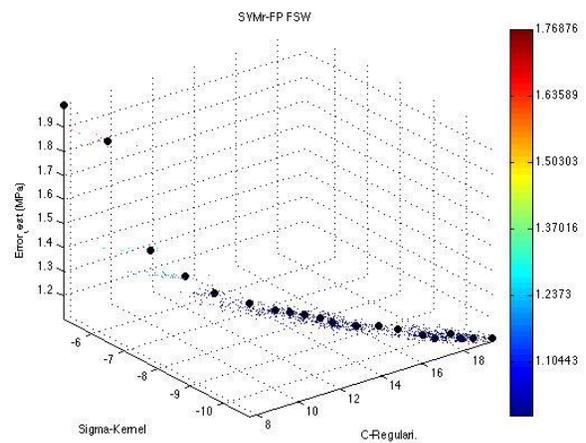
No. Partículas	2^C	2^{σ^2}	2^ϵ	Treinamento (MSE)	Validação (MSE)
100	16.6	-9.4	0.85	1.2999	3.2304
200	19.4	-10.8	0.85	1.2888	3.1167
500	20.5	-11.38	0.85	1.2866	3.2440
1000	20.3	-11.22	0.86	1.2849	3.2267

Tabela 5.12 Resultados do Erro e Coeficiente de Determinação de Treinamento-Validação e Teste para o SVMr com Filtro de Partículas com Diferentes Números de Partículas.

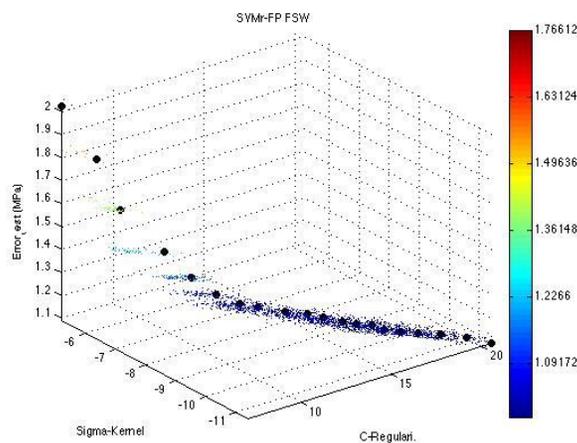
No. Partículas	2^C	2^{σ^2}	2^ϵ	Treina.-Val. (MSE)	Teste (MSE)	Treina.-Val. (R^2)	Teste (R^2)
100	16.6	-9.4	0.86	1.2976	1.1369	0.9132	0.9414
200	19.4	-10.8	0.86	1.2942	1.1193	0.9128	0.9423
500	20.5	-11.38	0.85	1.2924	1.0917	0.9129	0.9438
1000	20.3	-11.22	0.86	1.2901	1.0952	0.9130	0.9436



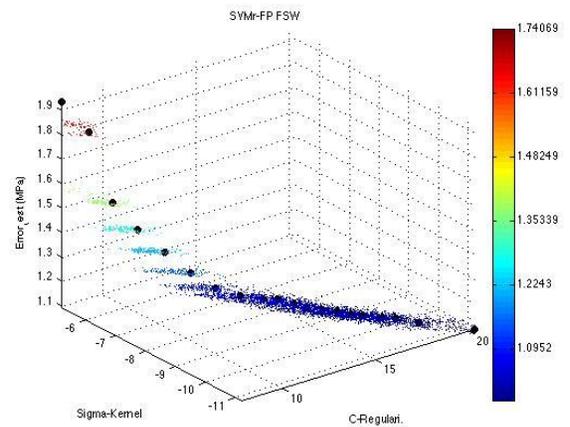
a. 100 Partículas.



b. 200 Partículas.



c. 500 Partículas.



d. 1000 Partículas.

Figura 5.7 Evolução do Erro de Teste do Sistema SVMr com Filtro de Partículas com Diferentes Números de Partículas em cada Iteração.

Na Tabela 5.12 mostra-se o erro das entradas de teste e o erro de treinamento-validação, sendo que o modelo final foi construído a partir destes dados. Também é mostrado o valor do coeficiente de determinação R^2 (APÊNDICE A). Observou-se que tanto os erros de treinamento-validação e teste, como os coeficientes de determinação (Tabela 5.12) têm resultados similares no algoritmo com diferentes número de partículas, obtendo uma boa predição com o mínimo de partículas de 100.

Embora, seja possível observar que o gráfico da Figura 5.7 (a) obtido para o algoritmo com o número de 100 partículas, apresentou um crescimento no início, além disso, a convergência foi similar que o algoritmo com maior quantidade de partículas.

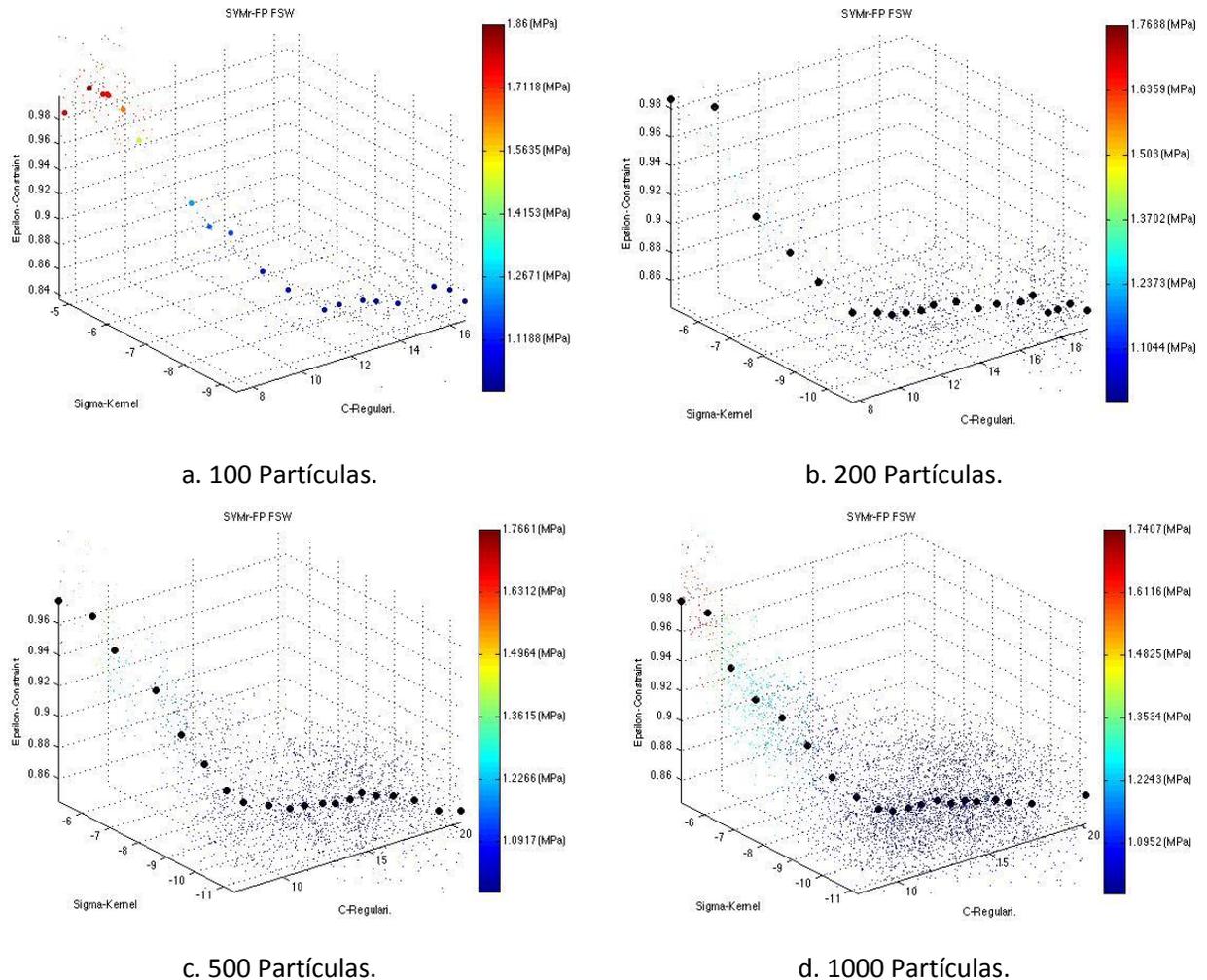


Figura 5.8 Comportamento das Partículas no Algoritmo. Diferentes Números de Partículas.

Na Figura 5.8 mostra-se a evolução do algoritmo através das iterações do filtro de partículas e pode-se observar os parâmetros que foram otimizados para o sistema, sendo C regularização no eixo y, no eixo x observa o σ^2 variável do “kernel” e por último no eixo z está a variável das restrições ε .

A cor significa o erro do teste em (Mpa), onde tem o comportamento similar em qualquer quantidade de partícula. Porém, pode-se observar que quanto maior a quantidade de partículas maior o campo de procura dos melhores parâmetros, sendo o algoritmo com 1000 partículas com maior campo de variação aleatória na faixa estudada. Frequentemente, trabalhar com mais partículas significa melhorar o algoritmo, embora, o custo computacional seja aumentado proporcionalmente.

5.3.1 Comportamento do Erro e Parâmetros do Modelo

Na Figura 5.9 pode-se observar o comportamento do erro das entradas de teste (MSE) em cada iteração. O algoritmo com 100 partículas teve um comportamento diferente, tendo problemas para encontrar os valores ótimos no início, embora seja possível observar que no final das iterações convergiu basicamente para o mesmo ponto que os outros algoritmos. Isto indica que o algoritmo precisa de mais partículas para melhorar a convergência do erro. Depois de 200 partículas o algoritmo apresentou um comportamento semelhante com uma boa resposta.

O valor do número de partículas escolhido foi 200, porque seu comportamento a partir desse ponto é similar, além disso, o custo computacional foi menor que o algoritmo com 500 e 1000 partículas.

Na Figura 5.10 mostra-se o comportamento dos parâmetros que foram otimizados no modelo SVMr. A ideia fundamental foi achar os melhores parâmetros para convergência do erro em cada modelo com diferentes números de partículas.

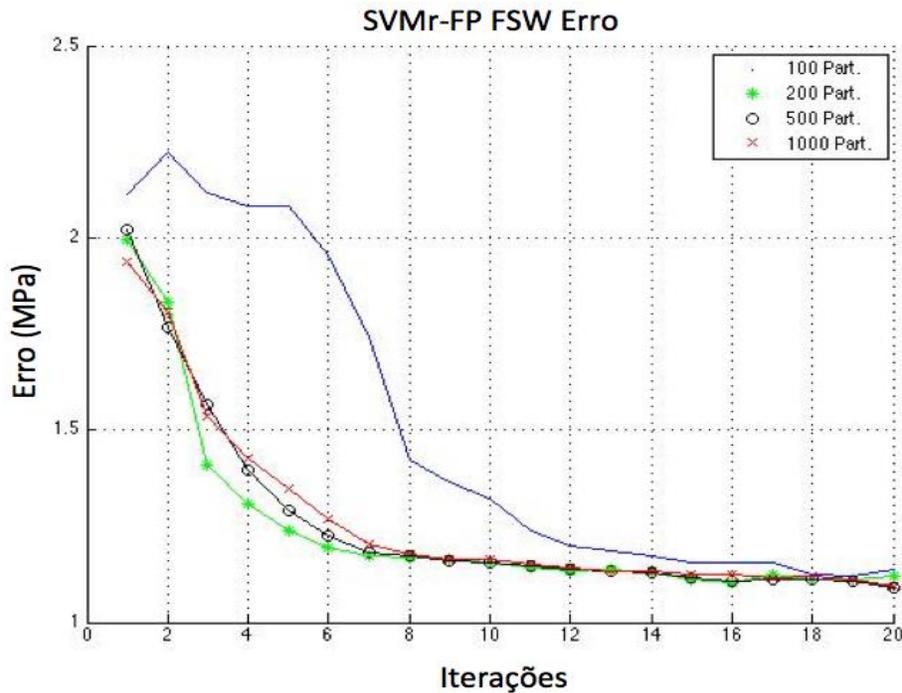


Figura 5.9 Comportamento do Erro em cada Iteração do Algoritmo SVMr com Filtro de Partículas.

Pode-se observar na Figura 5.10 (a) o parâmetro de regularização C (Seção 3.2), vai crescendo com respeito as iterações. Isto deve-se ao fato que ao se generalizar, no modelo penaliza-se todos os dados, já que o algoritmo utilizado para regressão foi ϵ -svm, onde o parâmetro C tendeu ao infinito. Embora, sendo o infinito o C a melhor resposta, ele funciona somente na teoria, além disso quando o valor é muito grande o custo computacional é muito elevado. Para poder enfrentar este problema a solução foi obter o dado quando a convergência tenha uma mudança com significância, colocando um valor limite de convergência para parar as iterações, quer dizer, quando a mudança do erro for menor que 0.001 o algoritmo termina.

Na Figura 5.10 (b) mostra-se o comportamento do coeficiente do “kernel” σ^2 , no qual vai decrescendo pelas iterações. É possível concluir que quanto menor for o valor do σ^2 melhor se assemelha aos dados de treinamento, dando uma diferença maior entre eles, já que o parâmetro σ^2 é o tamanho da gaussiana utilizada (Seção 3.2).

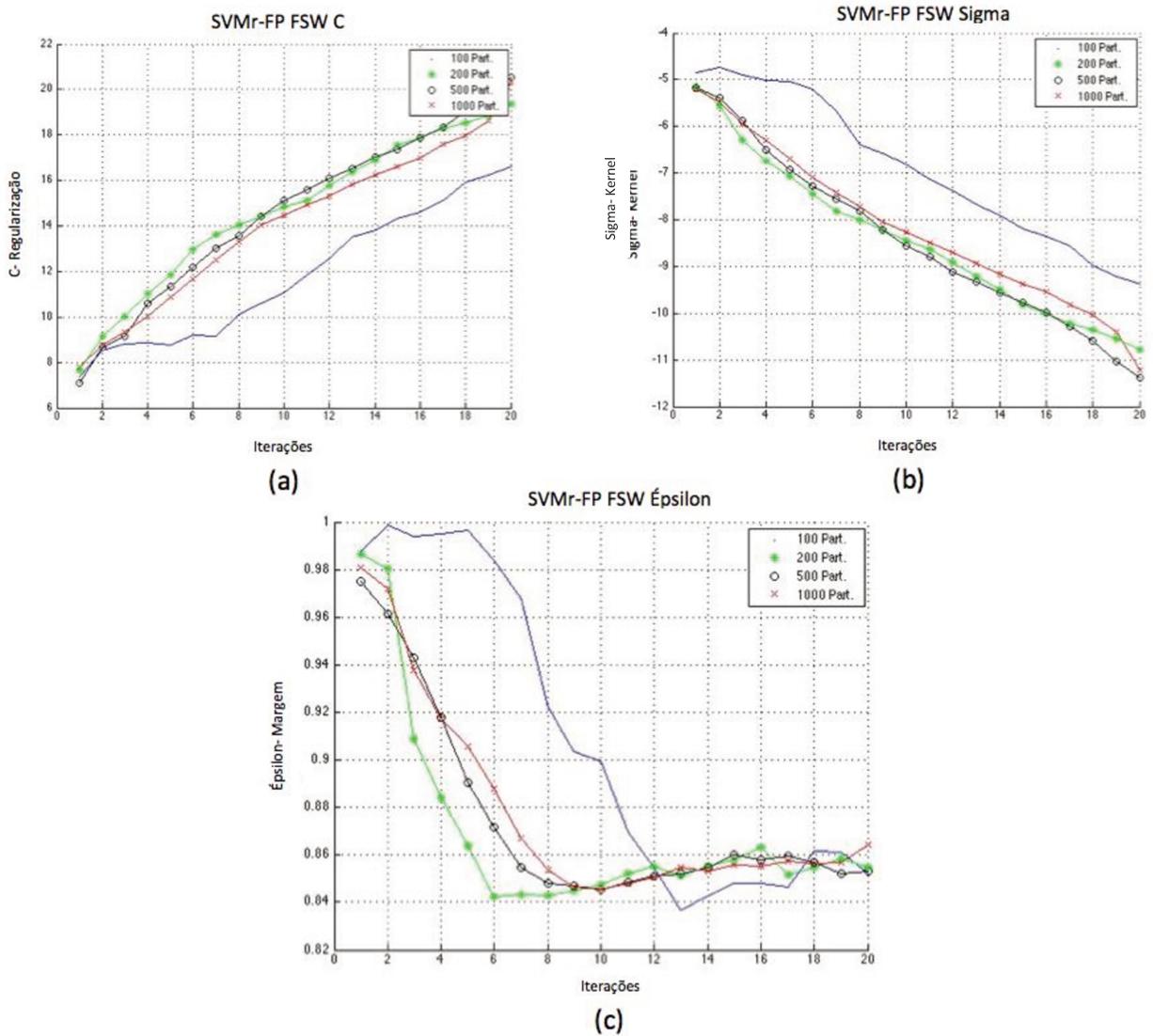


Figura 5.10 Comportamento dos Parâmetros dos Estados Ocultos do Filtro de Partículas; (a) Regularização - C; (b) “Kernel” – σ^2 ; (c) Restrições da Margem - ϵ .

Na Figura 5.10 (c) mostra-se o comportamento do ϵ , a qual é a restrição que foi colocada para a margem (Seção 4.2). Pode-se observar que foi estável a partir da sexta-oitava iteração, sem incluir o filtro de partículas de 100. Isto deve-se ao fator de ter sido encontrado um valor estável obtendo o melhor valor para o modelo, sendo o valor achado 0.86.

5.3.2 Análise de Predição das Respostas da SVMr

Foi utilizado a porcentagem de erro da média absoluta (MAPE) (APÊNDICE A) para a respostas de tensão de escoamento (TE) aplicando o filtro de partículas para estimação dos parâmetros com 200 partículas.

Na Tabela 5.13 mostra-se o erro do modelo feitos (SVMr-FP), tendo a tensão de escoamento um erro de 0.4058% e 0.3202% para o treinamento e teste, respectivamente, valores consideráveis bons para o modelo.

Tabela 5.13 A Porcentagem do Erro (MAPE) de Treinamento e Validação para SVMr.

	2^C	2^{σ^2}	2^ϵ	Treinamento (MAPE) %	Teste (MAPE) %
Escoamento (200 Partículas)	19.4	-10.8	0.85	0.4058	0.3202

5.3.3 Superfície de Resposta do SVMr - FP

O resultado da tensão de escoamento foi analisado através das ferramentas gráficas de superfícies de resposta como foi feito com os outros métodos (ANEXO iii e ANEXO iv). Contudo, a entrada foi limitada a dois parâmetros (VT e VR), caracterizando dois eixos de entrada x e y , e a resposta caracteriza ao terceiro eixo z , obtendo-se assim uma representação tridimensional.

A superfície de resposta Figura 5.11 gerada pelo modelo descreve o efeito das velocidades de rotação e translação sobre o parâmetro de tensão de escoamento da solda SWF. Na Figura 5.11, é possível observar o mesmo comportamento encontrado nos outros métodos (ANEXO iii e ANEXO iv), ou seja, o aumento da velocidade de translação resultou em incremento na tensão de escoamento, pelo contrário, o aumento da velocidade de rotação não teve influência.

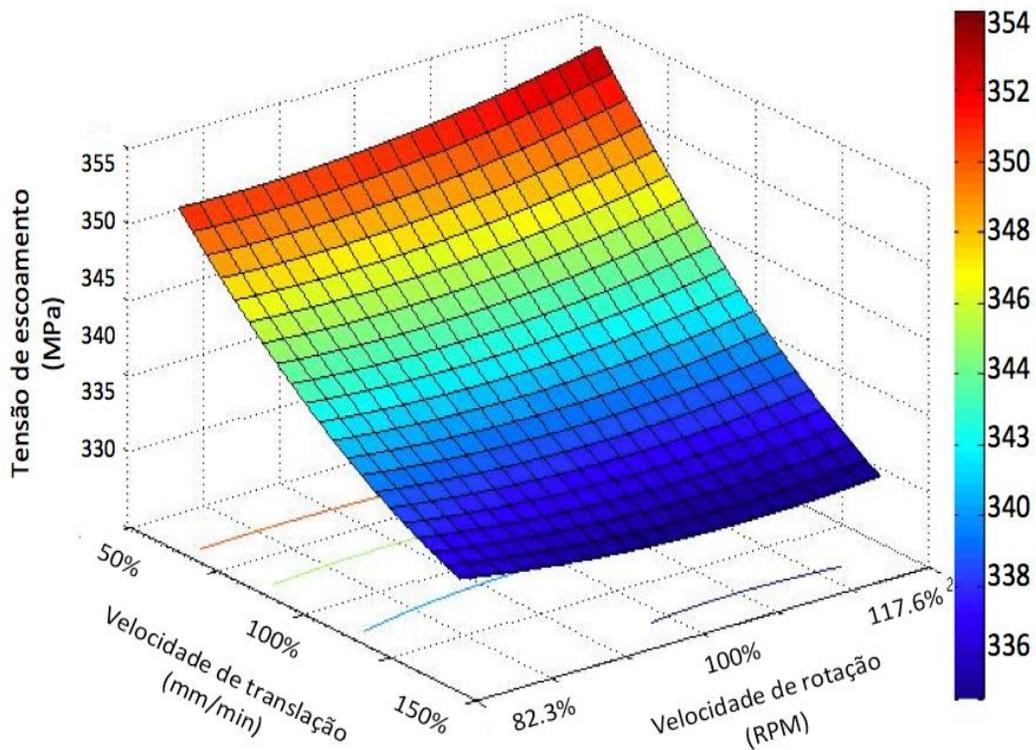


Figura 5.11 Superfície de Resposta da Tensão de Escoamento por o Algoritmo SVMr-FP.

5.4 Comparação Dos Resultados

Na Tabela 5.14 mostra-se uma comparação dos algoritmos feitos. Na regressão polinomial foi feito o modelo 2 (grau 2) (ANEXO i) e na rede neural foi feito o modelo 8 (2 camadas ocultas; LogSig Função de ativação; 20 neurônios) (ANEXO ii). Pode-se observar que o SVMr com filtro de partículas (200 partículas) tem o valor do erro de teste 1.1193 menor que os outros métodos (ANEXO iii e ANEXO iv), o que significa que generaliza muito melhor que os demais, embora a rede neural apresentou o menor erro de treinamento-validação com 0.3337.

No valor de coeficiente de determinação do treinamento, a rede neural (ANEXO iv) conseguiu o melhor resultado, porém, o (R^2) do teste foi o pior dos modelos. Em relação ao treinamento e teste o melhor modelo foi o SVMr com o FP.

Tabela 5.14 Comparação dos Resultados dos Métodos SVMr+PF; SVMr; Rede Neural e Regressão Polinomial.

Algoritmo	Resposta	Parâmetros	Treina.-Val. (MSE)	Teste (MSE)	Treina.-Val. (R ²)	Teste (R ²)
SVM reg + FP (200)	Tensão Escoamento	$2^C = 16.6$ $2^{\sigma^2} = -9.4$ $2^\varepsilon = 0.86$	1.2942	1.1193	0.9128	0.9423
SVM reg.	Tensão Escoamento	$C = 64$ $\sigma^2 = 0.0312$ $\varepsilon = 2$	1.4809	1.9029	0.9002	0.9020
RNA	Tensão Escoamento	$Modelo = 8$ $(\lambda) = 0.3$	0.3337	2.4282	0.9691	0.7809
Reg. Polinomial	Tensão Escoamento	$Modelo = 2$ $(\lambda) = 0.3$	1.0296	2.2639	0.9306	0.8834

Na Tabela 5.15 mostra-se uma comparação dos erros percentuais dos métodos utilizados. Pode-se observar que o erro mais baixo no teste do modelo (TE) foi o SVMr com filtro de partículas (200) com 0.3202%, embora, o erro mais baixo de treinamento foi obtido pela rede neural com 0.2447%.

Tabela 5.15 Comparação dos Erros Porcentual dos Métodos SVMr+PF; SVMr; Rede Neural e Regressão Polinomial.

Algoritmo	Resposta	Parâmetros	Treinamento (MAPE) %	Teste (MAPE) %
SVM reg + FP (200)	Tensão Escoamento	$2^C = 16.6$ $2^{\sigma^2} = -9.4$ $2^\varepsilon = 0.86$	0.4058	0.3202
SVM reg.	Tensão Escoamento	$C = 64$ $\sigma^2 = 0.0312$ $\varepsilon = 2$	0.4112	0.4211
RNA	Tensão Escoamento	$Modelo = 8$ $(\lambda) = 0.3$	0.2447	0.4644
Reg. Polinomial	Tensão Escoamento	$Modelo = 2$ $(\lambda) = 0.3$	0.3631	0.5340

Na Figura 5.12 mostra a regressão dos dados de treinamento de tensão de escoamento com todos os métodos. Devido à quantidade dos dados observa-se que os algoritmos tiveram resultados similares.

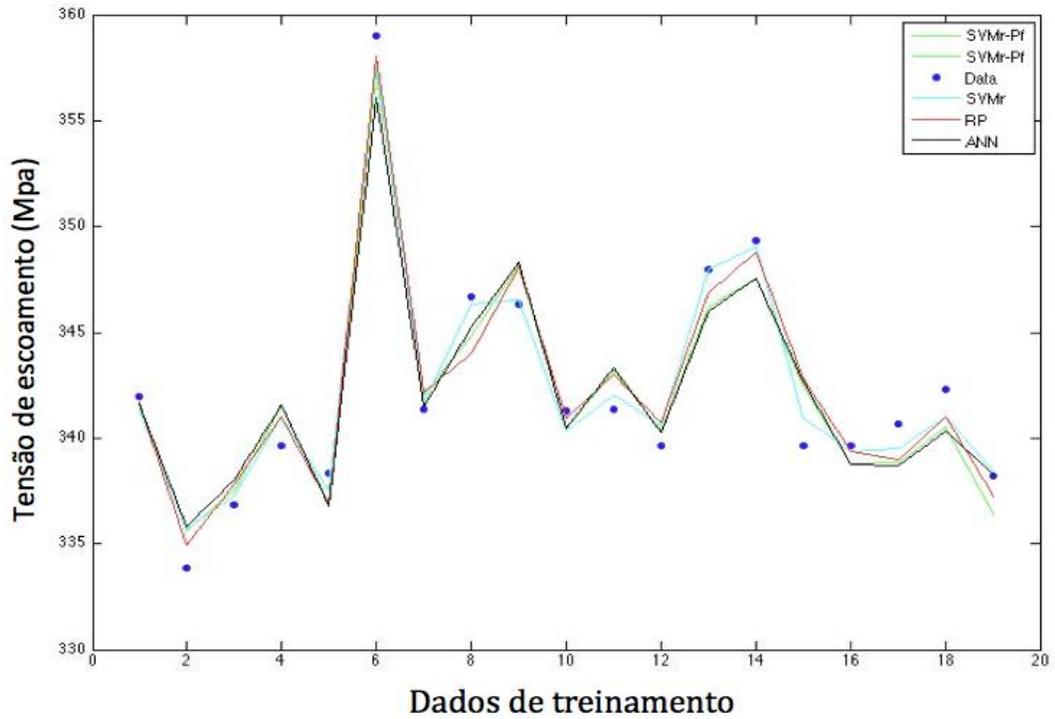


Figura 5.12 Comparação dos Resultados dos Métodos SVMr+PF; SVMr; Rede Neural; e Regressão Polinomial dos Dados de treinamento.

Capítulo 6 .

CONCLUSÕES E RECOMENDAÇÕES DE TRABALHOS FUTUROS

6.1 Conclusões

Devido à ampla variação possível das características do processo por FSW, é geralmente difícil de fornecer um modelo preciso que seja adequado para prever as propriedades mecânicas. Este trabalho teve por finalidade demonstrar as possibilidades de adaptação de diferentes métodos para modelar as tensões resultantes do processo devido as variações de suas características. O método apresentado facilita o desenvolvimento de modelos mais precisos do comportamento resultante do processos por FSW. A metodologia proposta fornece uma diretriz para modelar comportamentos de modelos complexos usando apenas uma quantidade limitada de dados experimentais.

O objetivo deste trabalho foi o de estudar possibilidades do uso de algoritmos de aprendizagem de máquina para o cálculo das propriedades mecânicas de chapas de Alumínio soldadas utilizando o método FSW. Os resultados mostraram que os algoritmos podem ser utilizados como uma forma alternativa nestes sistemas. Para o modelo Regressão Polinomial e Rede Neural proposto foi otimizado por BFGS (Quase Newton) mostrando melhor desempenho que outros métodos e para o SVMr foi utilizado programação quadrática (QP) “toolbox do Matlab” que realiza a otimização por meio de componentes de Lagrange. Para achar os parâmetros ótimos necessários no algoritmo foi desenvolvido um filtro de partículas obtendo melhores resultados que com o método de validação cruzada. Para isto foi necessário obter o modelo de SVM em espaço de estado com respeito a seus parâmetros como estados do sistema.

Este estudo levou às seguintes conclusões:

1. Realizou-se uma análise Estatística cujo resultado foi que a única resposta com significância e modelo preditivo linear é a do alongamento de escoamento, obtendo um $R^2(0.69)$, muito perto ao nível razoável para se determinar quando um modelo é

preditivo. Além disso, demonstrou-se também que tanto a tensão de escoamento, como tensão última e o alongamento na curva tensão-deformação, tem fator curvatura (fator não linear), por conseguinte foram feitos diferentes algoritmos de aprendizagem de máquina para os modelos dessas respostas.

2. Algumas variáveis de interesse resultantes dos diferentes parâmetros adotados no processo FSW podem ser calculadas utilizando os modelos construídos com esta metodologia. Isso torna conveniente e fácil de usar esses modelos para experimentos numéricos, assim, poder avaliar por exemplo os efeitos de cada variável nas tensões resultantes na solda por FSW.
3. As respostas de tensão última e alongamento não tiveram bons modelos em todos os algoritmos utilizados, apresentando um coeficiente de determinação muito menor do que 70%, inclusive chegando até 0, ou seja, sem relação entre entradas e saídas das respostas.
4. O coeficiente de determinação $R^2(0.78)$ e $R^2(0.96)$ (teste e treinamento-validação) da tensão de escoamento da rede neural (RNA) foi diferente aos valores obtidos de $R^2(0.88)$ e $R^2(0.93)$ da regressão polinomial. Portanto, o modelo da tensão de escoamento resultante do processo FSW baseado em RNA (tendo em conta que depende dos dados aleatórios) foi melhor ajustado conforme mostra o resultado do coeficiente de determinação quando utilizamos os dados treinados do que o modelo ajustado baseado em regressão polinomial, embora, o modelo ajustado pelo algoritmo de regressão polinomial, tenha tido melhor desempenho com os dados de teste, ou seja, mostrando que sua generalização é melhor. O algoritmo SVMr teve um coeficiente de determinação superando os 90%, o que significa que tanto no treinamento como no teste obteve bons resultados. O algoritmo SVMr com filtro de partículas teve uma pequena melhora com respeito ao SVMr, chegando até 94% de correlação entre entradas e saídas, obtendo melhores resultados que os outros métodos.
5. Com o filtro de partículas foi possível estimar os parâmetros ótimos do SVMr para o treinamento, ajudando para que o modelo obtenha a melhor representação do sistema. Além disso, colocando uma covariância nas partículas e utilizando a probabilidade

estatística randômica, conseguiu-se obter os estados ocultos, sendo sua velocidade de processamento mais rápida do que com o método de validação cruzada.

6. A velocidade de translação (VT) foi o fator que mais influenciou nas respostas de alongamento (Linear) e Tensão (Não Linear) de escoamento. A quantidade de dados para obtenção dos modelos foram poucos, sendo utilizados os 24 dados obtidos das matrizes. Assim, através das análises estatísticas realizadas (curvas de aprendizagem) concluiu-se que com a aplicação de uma quantidade maior de dados se espera um melhoramento nos modelos.

6.2 Recomendações de Trabalhos Futuros

Como perspectiva futura, pretende-se obter um modelo das propriedades mecânicas (curva de Tensão-Deformação) mais completo do processo da solda por FSW incluindo mais entradas e saídas, já que neste trabalho foram utilizadas as entradas da velocidade de rotação, velocidade de translação e força axial, com o objetivo de avaliar o desempenho dos métodos de aprendizagem de máquina. Assim um próximo estágio envolveria uma mudança nas saídas adicionando por exemplo medições de temperatura máxima e taxa de aquecimento da solda como parâmetros de saídas a partir das condições da solda. Também seria interessante obter modelos de ensaios de fadiga para obter a maior informação do modelo de FSW. Além disso, levar em consideração as microestruturas da solda, na qual tem efeitos significativos nas propriedades tanto estruturais como mecânicas com diferentes parâmetros de FSW.

Um dos principais problemas do SVM é a sua limitação prática de obtenção de seus parâmetros, dado que a qualidade dos modelos depende fortemente de um ambiente adequado de hiperplanos, ou seja dos parâmetros SVM (C e ϵ) e os parâmetros do “kernel” (σ^2). Por conseguinte, é necessário um abordagem automatizada, confiável e relativamente rápida para determinação dos valores destes parâmetros que conduzem ao menor erro generalizado. Isto, por exemplo, pode ser resolvido com o uso de algoritmos como PSO (“*Particle Swarm Optimization*”) ou GA (Algoritmo Genético), embora estes algoritmos também tenham que adotar uma função objetiva baseada em validação cruzada.

No que se refere ao filtro de partículas de estimação de estados ocultos dos parâmetros de SVMr, ele apresenta um problema de custo computacional alto, dado a necessidade de uma quantidade grande de partículas para se obter uma boa predição, embora obteve-se uma melhora aceitável utilizando-se a validação cruzada. Caberia realizar outro algoritmo estatístico como o filtro de Kalman para obtenção dos parâmetros necessários do SVMr, cujos algoritmos podem obter estes parâmetros mais rápidos dos que os obtidos com o filtro de partículas, embora ainda seja necessário a utilização do EKF (“*Extend Kalman Filter*”) uma vez que o modelo é não linear.

Outro problema apresentado foi relativo a quantidade de dados para obtenção do modelo, onde as curvas de aprendizagem obtidas pelos algoritmos de aprendizagem mostraram que com aumento das amostras poderia ter um melhoramento nos desempenhos dos algoritmos. Assim em uma próxima etapa o modelo estaria relacionado com a integração de uma maior quantidade de dados obtidos pelos ensaios de tração para os diferentes tipos de respostas (curvas de Tensão-Deformação) do sistema do processo da solda por FSW.

REFERÊNCIAS BIBLIOGRÁFICAS

ABBEEL, P.; COATES, A.; OUIGLEY, M.; NG, A. **An Application of Reinforcement Learning to Aerobatic Helicopter Flight**. In NIPS 19, 2007.

ABU-MOSTAFA, Y. **Learning from data**. Machine learning Course, California Institute of Technology, Caltech, 2012.

AGUIRRE, L. A. **Introdução à identificação de Sistemas: Técnicas Lineares e não Lineares Aplicadas a Sistemas Reais**, Editora UFMG, 2000.

ALMA, Y.; ALANIS, E. N.; SÁNCHEZ, A. G.; LOUKIANOV; PÉREZ, M. A. **Real-time Discrete Nonlinear Identification via Recurrent High Order Neural Networks**, Mexico, 2009.

ARTHUR, S. **Some Studies in Machine Learning Using the Game of Checkers**. IBM Journal 3(3):210-229, Mar. 1959, Retrieved Oct. 2011.

ASTM, International. **Standard Test Methods for Tension Testing of Metallic Materials. Designation:E8/E8M-11**. Highway and Transportation Officials Standard, AASHTO No.: T68, An American National Standard, acessado em: Julio de 2012.

BANKO, M. AND BRILL, E. **Scaling to very very Large Corporal for Natural Language Disambiguation**. In Proceedings of ACL,2001.

BEZAZI, A.; PIERCE, G. S.; WORDEN, K.; HARKATI, E. H. **Fatigue life prediction of sándwich composite materials under flexural tests using a Bayesian trained artificial neural network**. International Journal of Fatigue, Volume 29, Pages 738-747, 2007.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. **A training algorithm for optimal margin classifiers**. In: COLT '92: Proceedings of the Fifth Annual Workshop on Computational Learning Theory. New York, NY, USA: ACM Press, pp. 144–152.,1992.

BURGES, C. J. **A Tutorial on Support Vector Machines for Pattern**. Data Mining and Knowledge Discovery 2. Kluwer Academic Publishers, Boston. Manufactured in The Netherlands., 121-167, 1998.

CHANG, C-C. AND LIN, C-J. **LIBSVM** - A library for support vector machine, 2011.

CHERKASSKY, V. AND MULIER, F. **Learning from data Concepts, Theory, and Methods**. Second Edition, 2007.

COLLOBERT, R. AND BENGIO, S. **SVMtorch: support vector machines for largescale regression problems**. Machine Learning Res, 2001.

COMBETTES, P. L. AND PESQUET J. C. Chapter: Proximal Splitting Methods in Signal Processing, **Fixed-Point Algorithms for Inverse Problems in Science and Engineering**. (BAUSCHKE, H.H., BURACHIK, R.S., COMBETTES, P.L., ELSER, V., LUKE, D. R. AND WOLKOWICZ H., Editors), Springer, New York, 2011.

CORTES, C. AND VAPNIK, V. **Support-vector networks**. Machine Learning. 1995.

DAWES, C.J. AND THOMAS, W.M. **Friction stir process welds aluminum alloys**. The process produces low-distortion, high-quality, low-cost welds on aluminum. Welding Journal, Volume 75, Issue 3, Pages 41-45, 1996.

DOUCET, A.; GODSILL, S.; ANDRIEU, C. **On sequential Monte Carlo sampling methods for Bayesian filtering**”, Statist. Comput. vol. 10, pp. 197–208, ,2000.

DOUCET, A. AND TADIC, V. **Parameter estimation in general state-space models using particle methods**. Annals of the Institute of Statistical Mathematics, vol. 55, pp. 409-422,2003,

EBERHART, R. **Computational Intelligence Concepts to Implementations**. Imprint, Morgan Kaufmann, ago. 2007.

EBERHART, R.C. AND DOBBINS, R.W. **Early neural network development history: the age of Camelot**. Engineering in Medicine and Biology Magazine, IEEE , vol.9, no.3, pp.15,18, Sept. 1990.

ELANGO VAN, K.; BALASUBRAMANIAN, BABU, S. **Developing an Empirical Relationship to Predict Tensile Strength of Friction Stir Welded AA2219 Aluminum Alloy**. Journal of Materials Engineering and Performance, 17(6), 820-830, 2008.

ELANGO VAN, K.; BALASUBRAMANIAN B., V.; BABU, S. **Predicting tensile strength of friction stir welded AA6061 aluminum alloy joints by a mathematical model**. Materials and Design, 30, 188–193, 2009.

ETHEM A. **Introduction to Machine Learning**. The MIT Press Cambridge, Massachusetts London, Second Edition, England 2010.

EVGENIOU, T.; PONTIL, M.; POGGIO, M. **Regularization Networks and Support Vector Machines**. Advances in Computational Mathematics, 2000.

FEUERSTEIN, R.; FALIK, L. H.; & FEUERSTEIN, R. S. **The cognitive elements of neural plasticity**. Retrieved month 09, 2013, from <http://www.neuropsychotherapist.com/cognitive-elements-neural-plasticity> doi: 10.12744/tnpt.19.03.2013.01.

GENCAGA, D.; KURUOGLU, E.E.; ERTÜZÜN, A. **Estimation of time-varying autoregressive symmetric alpha stable processes by particle filters**. Technical Report No. 2006-TR-04, ISTI-CNR, 2006.

IZENMAN A.J. **Modern Multivariate Statistical Techniques**. Springer Texts in Statistics. 2008.

JAYARAMAN, M.; SIVASUBRAMANIAN, R.; BALASUBRAMANIAN, V. ; LAKSHMINARAYANAN, A. K. **Optimization of process parameters for friction stir welding of cast aluminum alloy A319 by Taguchi method**. Journal of Scientific and Industrial Research, Vol. 68, 36–43, 2009.

KUMAR, K. AND KAILAS, S.V. **The role of friction stir welding tool on material flow and weld formation**. Materials Science and Engineering: A, Volume 485, Issues 1–2, Pages 367-374, 25 June 2008. (<http://www.sciencedirect.com/science/article/pii/S0921509307015419>)

LEE, S.; PE'ER, D.; DUDLEY, A. M.; CHURCH, G. M.; KOLLER, D. **Identifying Regulatory Mechanisms Using Individual Variation Reveals Key Role for Chormatin Modification**. Proceedings of the National Academy of Sciences (PNAS), Sep. 2006.

LeCUN, Y. AND BENGIO, Y. **Word-Level Training of a Handritten Word Recognizer Based on Convolutional Neural Network**. IEEE Proc. Of the International Conference on Pattern Recognition, II:88-92, Jerusalem, Oct. 1994.

LIANGCHENG, C. AND HUIZHONG, Y. **A Learning Method of Support Vector Machine Based on Particle Filters**. School of Communication & Control Engineering, Fifth International Conference on Natural Computation, Tianjian, 2009.

LOMBARD, H.; HATTINGH, D.G.; STEUWER, A.; JAMES, M.N. **Optimizing FSW process parameters to minimize defects and maximize fatigue life in 5083-H32 aluminum alloy**. Engineering Fracture Mechanics, 75, 341–354, 2008.

LIN. C.; CHANG, C.; HSU, C. **A practical guide to support vector clasification**. National Taiwan University. 2004.

LJUNG, L. **System Identification: Theory for the User**, Prentice Hall, 2nd Ed., 1999.

MARTIN, M. **On-line support vector machines for function approximation**. (Tech. Rep. LSI-02-11-R).Catalunya, Spain: Software Department, Universitat Politecnica de Catalunya, 2002.

MENARD, S. **Applied Logistic Regression Analysis**. Second Edition, Sage Publications, 2001.

MITCHELL, T.M. **The Discipline of Machine Learning**. Machine Learning Department Technical Report CMU-ML-06-108, Carnegie Mellon University, Jul. 2006.

MITCHELL, T. M. **Machine Learning**. McGraw Hill, 1996.

MONTGOMERY, D.C. **Design and Analysis of Experiments**. John Wiley & Sons Inc. 7th Edition, 2009.

- MOREIRA, P. M. **Lightweight Stiffened Panels: Mechanical Characterization of Emerging Fabrication Technologies**. Tese de Doutorado, Universidade do Porto, 2008.
- NATIONAL INTITUTE ON AGING, NATIONAL INTITUTE OF HEALTH. **Alzheimer's Disease Eunnveling the Mystery**. NIH Publication Number: 08-3782, Sep. 2008.
- NG, A. **Introduction to Machine Learning**. Machine learning Course, Coursera based into Standford University, 2012.
- NORGAARD, M.,; RAVN, O.; POULSEN, N. K.; HANSEN, L. K. **Neural Networks for Modelling and Control of Dynamic Systems**. Springer-Verlag, London, 2003.
- OKUYUCU, H.; KURT, A.; ARCAKLIOGLU, E. **Artificial neural network application to the friction stir welding of aluminum plates**. Materials and Design, 28, 78-84, 2007.
- PITT, M. AND SHEPHARD, N. **Filtering via simulation: auxiliary particle Filters**. Journal of American Statistical Association, vol. 94, pp. 590-599, 1999.
- RAJAMANICKAM, N.; BALUSAMY, V.; MADHUSUDHANNA REDDY, G.; NATARAJAN, K. **Effect of process parameters on thermal history and mechanical properties of friction stir welds**. Materials & Design, v. 30, n. 7, p. 2726-2731, 2009.
- RECORD, J. H.; COVINGTON, J. L.; NELSON, T. W.; SORENSEN, C.D.; WEBB, B.W. **A Look at the statistical identification of critical process parameters in friction stir welding**. Welding Journal, 97-103,2007.
- RYCHETSKY, M. **Algorithms and Architectures for Machine Learning Based on Regularized Neural Networks and Support Vector Approaches (Berichte Aus Der Informatik)**. Shaker Verlag GmbH, Germany, Dec. 17, 2001.
- SAHA, G. **Applying Logistic Regression Model to the Examination Results Data**. Journal of Reliability and Statical Studies (ISSN), Kamalpur, Tripura, 2011.
- SARSILMAZ, F.; ÇAYDAS, U. **Statistical analysis on mechanical properties of friction-stir-welded AA 1050/AA 5083 couples**. Int. J. Adv. Manuf. Technol., 43, 248–255, 2009.
- SERMANET, P.; CHINTALA S.; LeCUN Y. **Convolutional Neural Networks Applied to House Numbers Digit Classification**. International Conference on Pattern Recognition (ICPR), 2012.
- SHAWE-TAYLOR, J. AND CRISTIANINI N. **Margin distribution and soft margin**. In: Alexander J. SMOLA, et al., eds. Advances in Large Margin Classifiers. The MIT Press, pp. 349–358,2000.

SHIVARAJ, M.K.; DINAKARAN, V.; MAHADEVAN, V. **Frictional Stir Welding on Aluminum Alloys AA2024-T4 and AA7075-T6**. Institute of Electrical and Electronics Engineers IEEE, 2010.

SHUANGSHENG, G.; XINGWEI, T.; SHUDE, J.; ZHITAO, Y. **Prediction of Mechanical Properties of Welded Joints Based on Support Vector Regression**. International Workshop on Information and Electronics Engineering, Volume 29, Pages 1471-1475,2012.

SCHÖLKOPF, B.; BURGUES, J.C.; SMOLA, A. **Advances in Kernel Methods. Support Vector Learning**. Massachusetts Institute of Technology, 1998.

SMITH, A.F.M.; GELFAND, A.E. **Bayesian statistics without tears: a sampling-resampling perspective**. American Statistician, vol. 46, pp. 84-88, 1992.

SNYMAN, J. A. **Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms**. Springer Publishing, 2005.

THRUN, S. **Artificial Intelligence for Robotic**. Computer Science Course, Udacity, 2013.

USTÜN, B. AND MELSSEN, W. J. **Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization**. Analytical Chimica Acta, vol. 544, pp. 292-305, 2005.

VAPNIK, V.; GOLOWICH, S.; SMOLA, A. **Support vector method for function approximation regression, estimation, and signal processing**. In M. Mozer, M. Jordan, and T. Petsche, editors, Advances in Neural Information Processing Systems 9, Cambridge, MA, 1997.

VAPNIK, V. **The Nature of Statistical Learning Theory**. 2nd edition, Springer, 1999.

XUNKAI, W.; YINGHONG, L.; CHEN, W.; JIANMING, L. **A Novel Identification Model of Aeroengine Based on Support Vector Machines**. Proceedings of the 5th World Congress on Intelligent Control and Automation, 2004.

YOUSIF, Y. K.; DAWS, K. M.; KAZEM, B. I. **Prediction of Friction Stir Welding Characteristic Using Neural Network**. Jordan Journal of Mechanical and Industrial Engineering, Vol. 2, No. 3, 151–155, 2008.

APÊNDICE A. – Metodologia para a Avaliação do Modelo

Na atualidade não tem sido desenvolvido nenhum protocolo de avaliação para os modelos, sendo uma tarefa pendente e necessária. A maioria das pesquisas se destacam por uma avaliação dos resultados qualitativos. Já nas pesquisas onde qualificam a análise, optam-se por utilizar estatística simples como erro quadrático médio (MSE), erro absoluto médio (MAE) ou coeficiente de determinação (R^2).

Para está avaliação foi realizado erro quadrático médio (MSE), coeficiente de determinação (R^2) e porcentagem absoluto médio (MAPE).

A precisão é dada pela média dos valores preditos e os valores observados. Por valores observados têm-se aqueles obtidos pelo ensaio de tração.

Para o cálculo de precisão é utilizado o erro médio quadrático, definido como,

$$MSE = \sum_{i=1}^n \frac{(Y_i - Y_{iobs})^2}{n}, \quad (A.1)$$

onde, Y_i é o valor predito pelo algoritmo de aprendizagem na entrada i ; Y_{iobs} é o valor observado da entrada i e n é o número de valores analisados. Dessa forma, foram feitos os três algoritmos: RP, RNA e SVM; e assim, analisou-se o performance dos algoritmos de aprendizagem.

A porcentagem absoluta do erro (MAPE) é uma medição do desempenho para a construção de ajustes de valores de séries em estatística (YOUSIF et al., 2008), definido como,

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{Y_{iobs} - Y_i}{Y_{iobs}} \right|. \quad (A.2)$$

O valor absoluto neste cálculo é somado para cada ponto observado e é dividido pelo número de pontos analisados n . Quando multiplicado por 100 torna-se um erro percentual, conhecido como (MPE). Utilizando esta formulação estatística comparou-se os algoritmos feitos para a análise do FSW.

O coeficiente de determinação (R^2) é uma medida para ajustar os modelos estatísticos generalizados em relação aos valores observados. O modelo varia entre 0 e 1, indicando os valores observados em valores de porcentagem. Quanto mais perto de 1 é o valor, mais explicativo é o modelo, definido como,

$$R^2 = \frac{\sum_{i=1}^n (Y_i - Y_{media})^2}{\sum_{i=1}^n (Y_{iobs} - Y_{media})^2} = 1 - \frac{\sum_{i=1}^n (Y_i - Y_{iobs})^2}{\sum_{i=1}^n (Y_{iobs} - Y_{media})^2}, \quad (A.3)$$

onde Y_i é o valor prognosticado pelo algoritmo de aprendizagem na entrada i ; Y_{media} a média dos valores, Y_{iobs} o valor observado da entrada i e n o número de valores analisados. A somatória do dividendo é a soma dos quadrados das diferenças entre a média e cada valor observado. As outras somatórias são a soma dos quadrados explicada e dos resíduos, respectivamente. A primeira indica a diferença entre a média das observações e o valor estimado para cada observação, a segunda calcula a parte não explicativa do modelo.

Em resumo, R^2 é a predição percentual do comportamento da saída que pode ser explicada pela entrada.

- $R^2 < 30\%$ é considerado que o modelo não tem correlação e a explicação de seu comportamento é aleatória.
- R^2 (30% – 49.99%) são considerados um relacionamento leve.
- R^2 (50% – 69.99%) são considerados um relacionamento moderado.
- R^2 (70% – 100%) são considerados um relacionamento forte.

APÊNDICE B. - Análise Estatística

Com uma análise estatística é possível verificar as hipóteses entre incerteza ou variabilidade. Desta forma, compara-se as previsões com os dados observados, se as comparações estão na margem admissível, o modelo é explicativo. Caso contrário, o modelo deve ser rejeitado por um novo modelo, em geral mais complexo, que explique melhor as observações (IZENMAN, 2008).

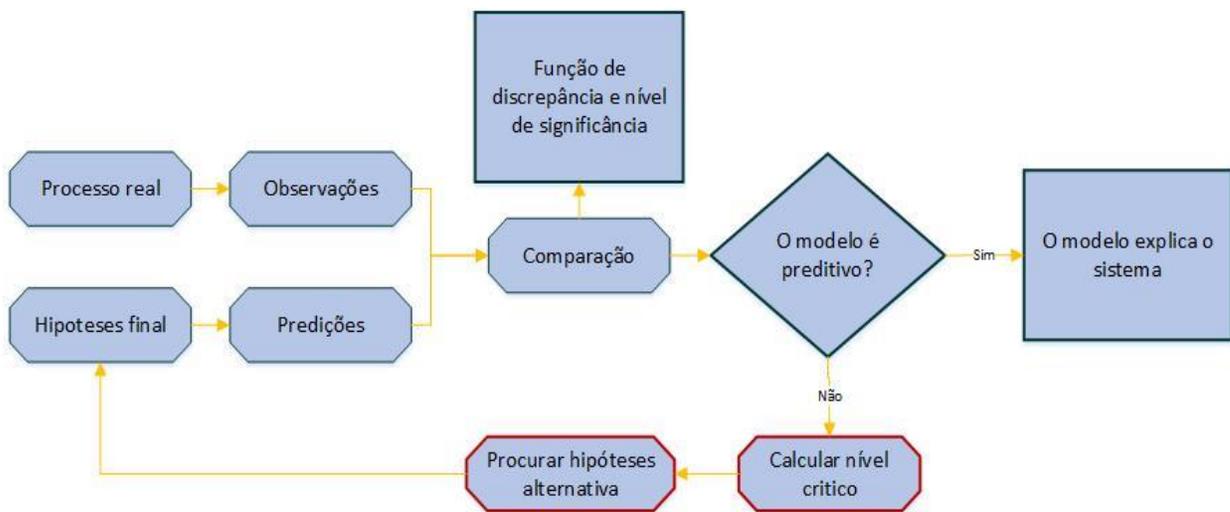


Figura B.1 Processo da Avaliação da Hipóteses.

Dessa maneira, realizou-se os Paretos (Seção 5.1.2) para determinar as significâncias estatísticas por cada fator até sua segunda interação. Esta análise é obtida dividindo o valor estimado do coeficiente da variável independente pelo desvio ou erro padrão deste coeficiente. A probabilidade total da curva de distribuição do parâmetro estimado tem 100%. Esta probabilidade se divide entre o nível de confiança e de significância. A significância estatística está nas extremidades da curva e no nível de confiança do interior da distribuição. Para as saídas obtidas tomou-se um nível de confiança de 95%, portanto, o nível de significância estatística que foi selecionado é de 5%.

Em seguida, a ANOVA calcula a medida da variabilidade total da saída (TSS), a qual é a soma da diferença entre a valor real e a média ao quadrado. Logo, obtém-se a medida da

variabilidade não explicada pela variável repressora (RSS), a qual é a soma dos resíduos ao quadrado. E finalmente, uma medida de quanto contribuem as variáveis de entrada para explicar a variabilidade da saída (variabilidade explicada pelo modelo de regressão) (RegSS); conforme mostradas na Equação (B.1) (IZENMAN, 2008),

$$\begin{aligned}
 TSS &= \sum_{i=1}^n (y^{(i)} - \bar{y})^2 \\
 RSS &= \sum_{i=1}^n (y^{(i)} - \widehat{y}^{(i)})^2, \\
 RegSS &= \sum_{i=1}^n (\widehat{y}^{(i)} - \bar{y})^2
 \end{aligned}
 \tag{B.1}$$

onde as somatórias de quadrados correspondentes as três fontes de variação que foram descritas acima apresentam-se em uma tabela, denominada ANOVA.

Tabela B.1 Análise de Variância (ANOVA)

Fonte de Variação	SS	d.f	MS
Modelo	RegSS	Np-1	ReggSS/ (Np-1)
Residual	RSS	n-Np	RSS/(n-Np)
Total	TSS	n-1	

A coluna *d.f* indica os graus de liberdade de cada *SS*. O modelo tem *d.f* = *Número de parâmetros no modelo* – 1, ou seja, o número de co-variáveis no modelo; a soma de quadrados residual tem *n – graus de liberdade* e a soma de quadrados total tem *n – 1*. A coluna MS (Mean Square) obtém os coeficientes entre o SS e seus respectivos graus de liberdade. O teste de F (Fisher) (MONTGOMERY, 2009) é dado pela relação entre MS da regressão e MS do residual, encontrando uma significância do modelo, tem-se:

$$F = \frac{MSRegg}{MSR}
 \tag{B.2}$$

APÊNDICE C. - Curva de Tensão-Deformação

C.1 Teoria de Curva de Tensão-Deformação

O diagrama de Tensão-Deformação como se mostra na Figura C.2 pode ser obtido por meio de ensaios de tração, os quais expressam a relação entre uma força aplicada ao material e a deformação do material devido a carga aplicada.

Os ensaios de tração foram feitos na máquina para testes de tração MTS, a qual tem um controlador proporcional uniformemente crescente que aplica força de tensão ao espécimen. As extremidades das amostras são seguradas e fixadas na MTS, e um sensor de deformação L_0 (comprimento calibrado entre as marcas do frente da amostra) é continuamente medida até a ruptura. Na Figura C.1 observa-se o espécimen utilizado para os ensaios de tração.

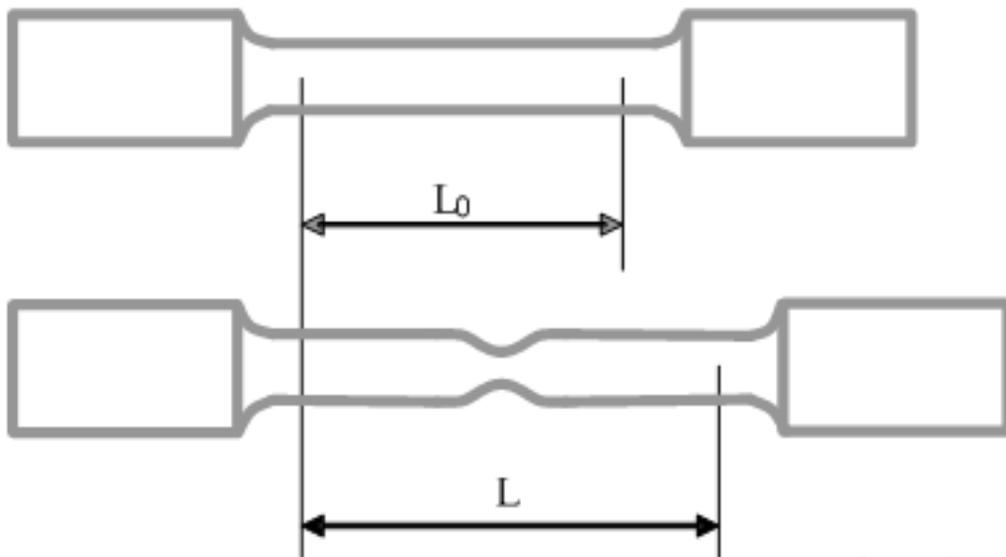


Figura C.1 Espécimen para Ensaios de Tração.

A deformação da amostra é a razão entre o deslocamento e o comprimento original, conforme mostra na Equação (C.1)

$$\delta = \frac{(L - L_0)}{L_0}. \quad (C.1)$$

A tensão é a razão entre as cargas de tração F aplicada à amostra a sua área de seção transversal original S_0 , conforme mostra na Equação (C.2),

$$\sigma = \frac{F}{S_0}. \quad (C.2)$$

A linha reta inicial da curva característica (Figura C.2) tem uma relação proporcional entre a tensão e a deformação, ou seja, uma relação linear. Este valor deverá ser menor que o ponto de limite de proporcionalidade, baseando-se na lei de Hook,

$$\begin{aligned} \sigma_p &= \frac{F_p}{S_0}, \\ \sigma &= E * \delta \end{aligned} \quad (C.3)$$

onde E é uma constante, conhecida como módulo de Young. Este valor é determinado principalmente pela natureza do material, praticamente insensível ao tratamento térmico e obtido por sua composição. O valor do módulo de Young determina a rigidez de um corpo de uma deformação elástica causada por uma força aplicada.

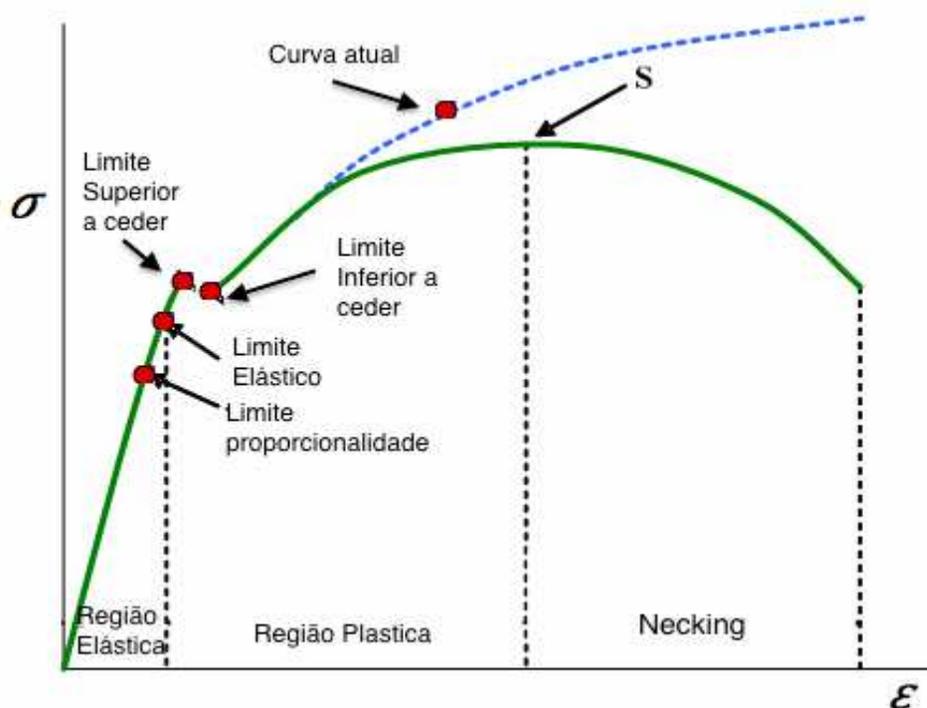


Figura C.2 Curva Características de Ensaios de Tração Padrão.

A curva até o ponto de deformação elástica (Figura C.2) define a fase elástica do material. A remoção de carga em qualquer ponto da curva nesta fase retorna o seu valor inicial. O ponto onde a tensão provoca deformação súbita, sem qualquer aumento na força é chamado de limite a ceder “*Yield Limit*”.

Se a carga atinge o limite de escoamento da amostra, esta sofre uma deformação plástica, ou seja, não retornará ao seu comprimento original após a remoção da carga.

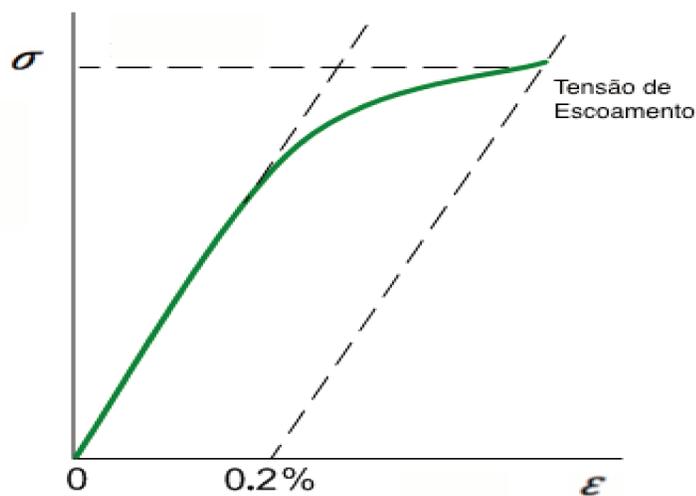


Figura C.3 Ponto de Tensão de Escoamento na gráfica de Tensão-Deformação.

A maioria dos materiais não têm definido o limite de escoamento, portanto, o que corresponde a uma tensão definitiva (0.2%) é o limite utilizado. Esta tensão é chamada de tensão de escoamento conforme mostrado na Figura C.3.

Com o aumento de carga, a amostra continua a sofrer deformação plástica e posteriormente se obtém uma diminuição na seção transversal devido ao “*necking*”, como mostrado na Figura C.2. Neste ponto, a tensão obtém o seu valor máximo (tensão última); em seguida, sua ruptura na amostra.

C.2 Obtenção da curva Tensão-Deformação

Os ensaios de tração foram obtidos de acordo com a norma ASTM E8M, para os experimentos de tração de materiais mecânicos, como mostra-se na Figura C.4 (ASTM, 2012).

A “*MTS Landmark*” hidráulica foi utilizada para ensaios de tração com o objetivo de obter a Curva Tensão-Deformação. Além disso, foram utilizados softwares para testes, os quais geram os relatórios dos experimentos; (MTS TestSuite™ Software e TestWorks® Software)



Figura C.4 Corpos de prova com solda FSW da matriz 2 condição 1; sensor MTS no teste; e equipamento MTS para os ensaios de tração.

O sensor utilizado foi a célula de medição axial (MTS 634.11e-24). Este tem um comprimento $L = 25\text{mm}$ com movimentação de tração de $+5\text{mm}$ e compressão -2.5mm . A faixa de tensão é $+20\%$ a -10% , mostrado na Figura C.4.

APÊNDICE D. - Algoritmos de Classificação

D.1 Regressão Logística

Este algoritmo funciona com uma função logística, ou seja, uma função sigmoide. Substituindo a função polinomial (NG, 2012),

$$\begin{aligned}h_{\theta}(x) &= g(z) \\g(z) &= \frac{1}{1 + e^{-z}} . \\h_{\theta}(x) &= \frac{1}{1 + e^{-\theta^T x}}\end{aligned}\tag{D.1}$$

A função sigmoide é definida no intervalo de $0 \leq h_{\theta}(x) \leq 1$. Mostrado na Figura D.1. Pode-se observar que dependendo a saída da $h_{\theta}(x)$, o valor da sigmoide será alterado.

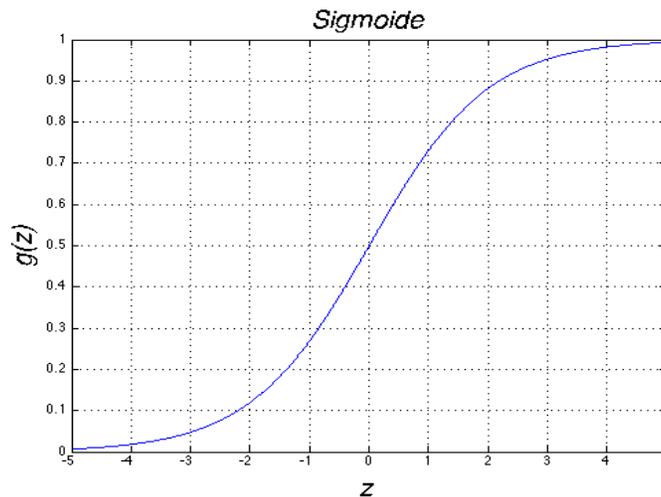


Figura D.1 Função Sigmoide.

Para regressão logística não é possível determinar a função de custo da regressão polinomial. Já que a função sigmoide é não linear. Como resultado deve-se mudar a função $J(\theta)$. A Equação (D.2) funciona de forma que $-\log(h_{\theta}(x))$ tende ao ∞ . Por consequência $-\log(1 - h_{\theta}(x))$ tende ao ∞ ,

$$J(\theta) = \begin{cases} -\log(h_{\theta}(x)) & \text{se } y = 1 \\ -\log(1 - h_{\theta}(x)) & \text{se } y = 0 \end{cases}\tag{D.2}$$

e simplificando a Equação (D.2) (SAHA, 2011) e substituindo a hipótese, tem-se a (D.3) (NG, 2012).

$$J(\theta) = -y \log(h_\theta(x)) - (1 - y) \log(1 - h_\theta(x))$$
$$J(\theta) = -y \log\left(\frac{1}{1 + e^{-\theta^T x}}\right) - (1 - y) \log\left(1 - \frac{1}{1 + e^{-\theta^T x}}\right). \quad (D.3)$$

Esta função de custo foi escolhida para este tipo de problema, como resultado da derivação da estatística usando o princípio de máxima verossimilhança (MLE). Este é um método para estimar os parâmetros de um modelo e também tem uma propriedade que acaba sendo convexa (MENARD, 2001).

A otimização deste método é possível realizar com o algoritmo BFGS (APÊNDICE E).

D.2 Rede Neural

As redes neurais artificiais estão compostas pela interação massiva em paralelo por elementos mais simples com organização hierárquica. Com a missão de relacionar os objetos do mundo real da mesma forma que o sistema nervoso biológico (ETHEM, 2010); (ABU-MOSTAFA, 2012); e (NG, 2012).

O funcionamento da rede neural tem um processo muito simples. Cada neurônio recebe entradas de outros neurônios para se comunicar. O efeito de cada linha de entrada sobre o neurônio é controlado por parâmetros ou pesos, os quais adaptam-se para que a rede aprenda e desempenhe uma tarefa.

A primeira camada é chamada de camada de entrada, a qual tem as características ou entradas da função. A camada última é conhecida como camada de saída, a qual tem-se a hipótese final. As camadas que se encontram entre as camadas de entrada e saída são chamadas de camadas ocultas (ABU-MOSTAFA, 2012). Na representação do modelo das redes neurais, tem-se outro termo denominado “unidade de bias” nas entradas de cada camada, com um valor de 1. A arquitetura da RNA é mostrada na Figura D.2.

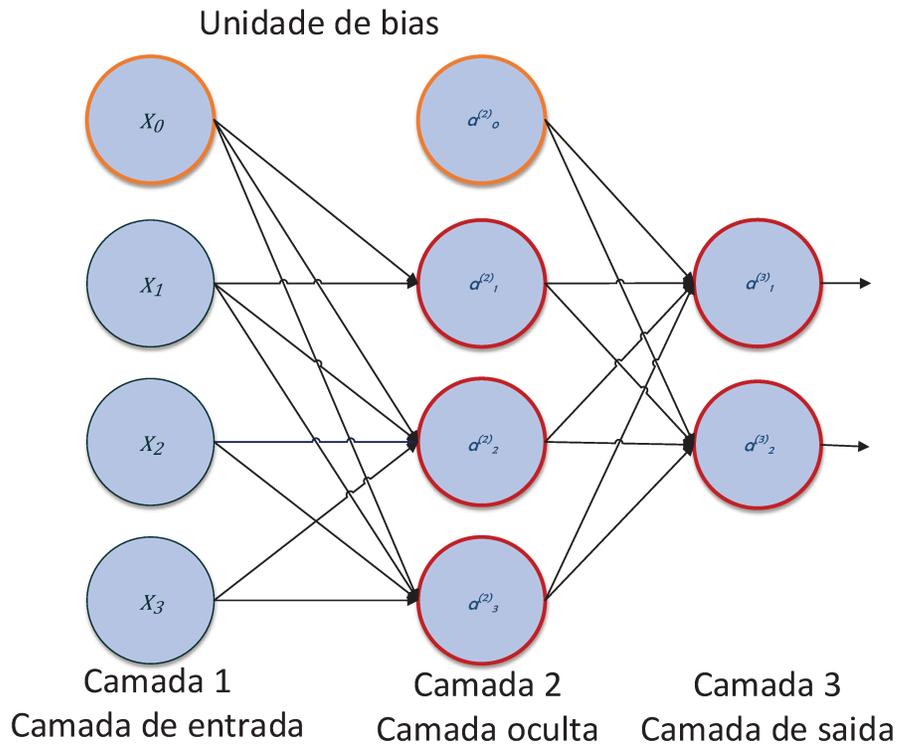


Figura D.2 Arquitetura da Rede Neural para Classificação.

A rede neural é parametrizada por as matrizes dos pesos controlando a função de mapeamento desde uma camada para a seguinte. O valor computado por cada neurônio da camada oculta é feita pela função de ativação, aplicado a um tipo de combinação linear das entradas do neurônio e multiplicado pelos parâmetros θ das conexões (NG, 2012).

- **Forward Propagation**

O processo de computação para encontrar a hipótese, é conhecido como “*Forward Propagation*”, o nome é dado porque se começa nas entradas e calcula as ativações das camadas ocultas até chegar ao final da camada de saída, mostrada na Equação (D.4),

$$\begin{aligned} z^{(i)} &= \theta^{(i-1)} a^{(i-1)} \\ a^{(i)} &= g(z^{(i)}) \end{aligned} \quad (D.4)$$

A vantagem das características é que aprendem por si mesmas como funções de entradas. Concretamente, a função de mapeamento desde a camada de entrada para a camada oculta são ajustados por parâmetros $\theta^{(1)}$ (NG, 2012). A rede neural pelo contrário de estar limitada

alimentando as características x (Entradas) como a Regressão Polinomial, conforme o APÊNDICE F, tem a oportunidade de aprender suas próprias características $a^{(i)}$. Dado que o algoritmo depende dos parâmetros θ escolhidos para aprender uma função de características complexas, por conseguinte é necessário à utilização das camadas ocultas, como é mostrado na Figura D.3.

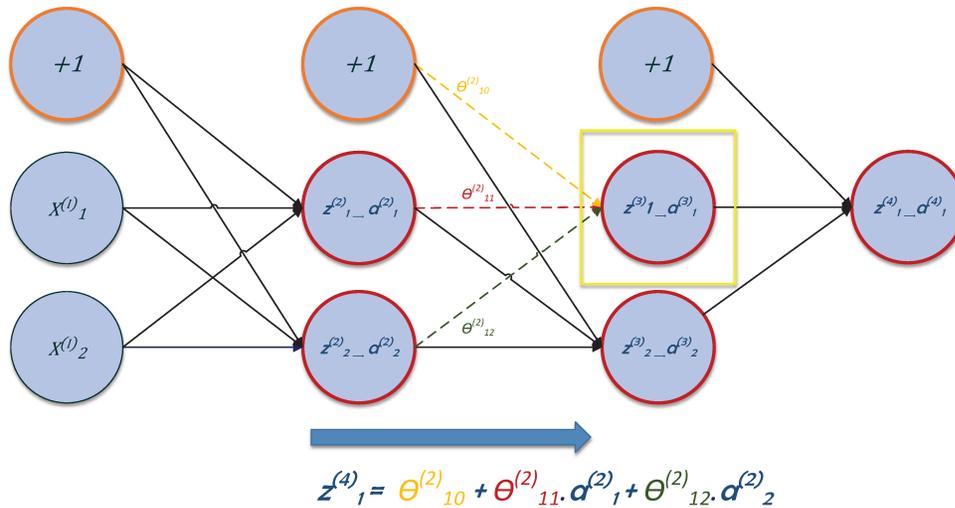


Figura D.3 Gráfica de “Forward Propagation”.

A rede neural pode ter diferentes arquiteturas, o termo de arquitetura refere-se aos diferentes neurônios conectados entre si. A primeira camada tem as características do modelo, na seguinte camada, as características são convertidas em características mais complexas, e como resultado pode-se obter uma hipótese não linear (ABU-MOSTAFA,2012).

- **Back Propagation**

O “*Back propagation*” é um algoritmo para obter o gradiente da função. Primeiramente, obtém-se o erro derivado δ entre o dado real e o estimado, conforme mostrado na Equação (D.5)(NG, 2012).

$$\begin{aligned}
E &= \frac{1}{2} \sum_j^m (y_j - a_j)^2 \\
\frac{\partial E}{\partial y_j} &= -\frac{2}{2} (a_j^{(4)} - y_j) \\
\delta_j^{(4)} &= a_j^{(4)} - y_j
\end{aligned} \tag{D.5}$$

Após obter o primeiro δ , se utiliza-o para calcular os δ posteriores das camadas da rede, por conseguinte, multiplica-se os δ obtidos pelos θ das conexões pela derivada da função de ativação sigmoide $g'(z)$, avaliados na combinação linear, mostrado na Equação (D.6),

$$\frac{\partial y_j}{\partial z_j} = g'(z) = a(1 - a). \tag{D.6}$$

O procedimento é feito até chegar à primeira camada. Em resumo, o δ soma os erros anteriores utilizados para corrigir os parâmetros θ , ou seja, os pesos da rede, conforme descrito na Equação (D.7),

$$\frac{\partial E}{\partial z_j} = \frac{dy_j}{dz_j} \frac{\partial E}{\partial y_j} = \delta_j = \theta^T \delta_{j+1} .* g'(z_j). \tag{D.7}$$

O procedimento é realizado para todas as amostras do treinamento, ou seja, m vezes. O resultado do gradiente será somado.

Formalmente, o erro de custo de cada ativação do neurônio é dado pela Equação D.8. Sendo l o número da camada e j o número do neurônio. A função $cost(i)$ é dada dependendo da classe do algoritmo para classificação da função de custo, a qual foi descrita anteriormente (NG, 2012), conforme a Equação (D.8),

$$\begin{aligned}
\delta_j^l &= \frac{\partial y}{\partial z_j^{(l)}} cost(i) \\
cost(i) &= -y \log(h_\theta(x)) - (1 - y) \log(1 - h_\theta(x))
\end{aligned} \tag{D.8}$$

A otimização deste método é possível realizar com o algoritmo BFGS (APÊNDICE E).

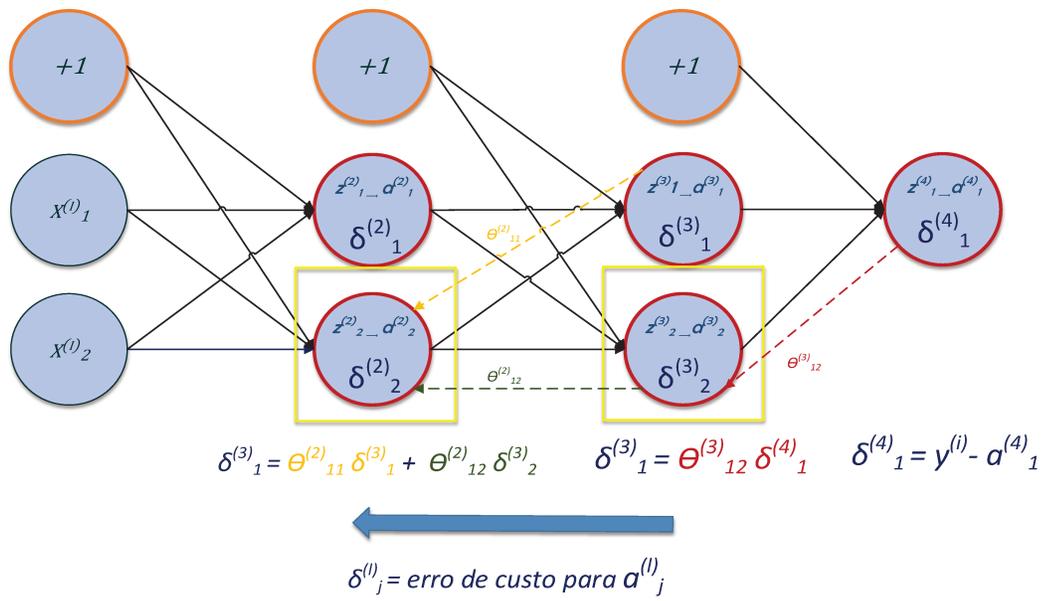


Figura D.4 Gráfica de “Back Propagation”.

APÊNDICE E. - Otimização

Na atualidade é bem determinada a importância da otimização como um dos princípios básicos de análises dos problemas complexos de decisão. É muito importante para os problemas de Aprendizagem de Máquina calcular os parâmetros necessários minimizando a função de custo.

Existem diferentes métodos de otimização para minimização de uma função. Os métodos mais comuns são os métodos de gradiente, os quais são muito importantes na ML. Um problema dos métodos de gradiente é que quando se encontra o mínimo da função não se tem certeza que este mínimo é global, já que o algoritmo poderá convergir ao mínimo local, como é mostrado na Figura E.1 (ABU-MOSTAFA,2012).

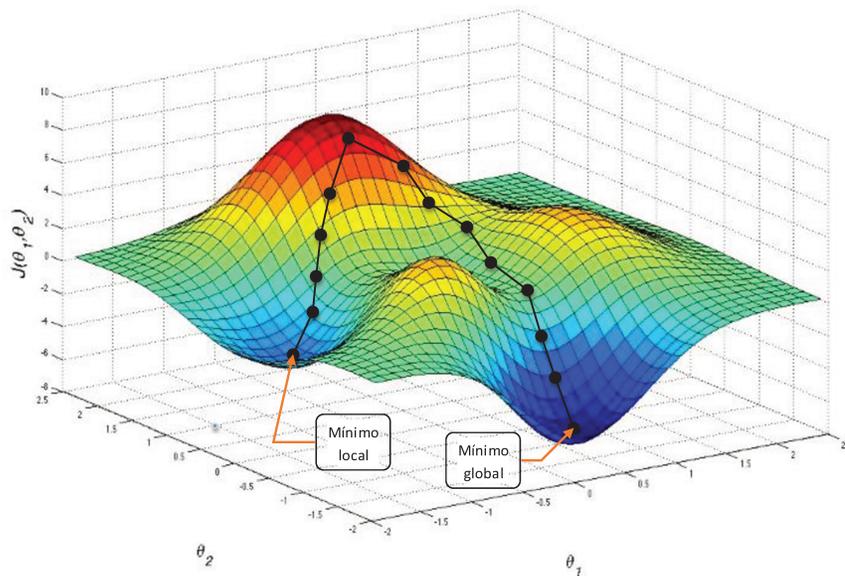


Figura E.1 Divergência ao Mínimo da Função (Mínimo Local e Global).

No uso dos algoritmos gradiente é utilizado a função de custo $J(\theta)$ e o gradiente $\frac{\partial}{\partial \theta_j} J(\theta)$. Além disso, existem outros algoritmos como Newton que precisa de um dado extra, a hessiana $\frac{\partial^2}{\partial \theta_j^2} J(\theta)$.

E.1 Gradiente Descendente Batch

O gradiente descendente é o algoritmo mais geral de otimização. Inclusive é utilizado em vários métodos de aprendizagem de máquina (“*Machine Learning*”-ML). Uma extensão do gradiente descendente é colocar restrições incluindo uma projeção. Este método é um caso específico do algoritmo “*forward-backward*” (COMBETTES, PESQUET, 2011).

Primeiramente, deve-se iniciar em algum ponto da função com os parâmetros θ . Posteriormente, armazena-se o valor da mudança dos θ até convergir ao mínimo (SNYMAN, 2005). Assim, para encontrar o mínimo da função, deve-se achar a derivada parcial de $J(\theta)$, também conhecida como o gradiente mostrado na Equação (E.1),

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1, \dots, \theta_n). \quad (\text{E.1})$$

A ideia principal do gradiente (derivada parcial de função de custo) é calcular o declive, ou seja, a tangente da função no ponto dado. Dessa forma, precisa-se atualizar cada parâmetro, quando são obtidos os novos θ . Para obter a mudança em cada iteração, aplicamos um parâmetro chamado taxa de aprendizagem “*Learning Rate*” α na Equação (E.2),

$$\alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1, \dots, \theta_n). \quad (\text{E.2})$$

O α controla o quanto será o tamanho do passo para o seguinte valor dos parâmetros θ , sendo assim, um valor positivo. Se a derivada parcial ou declive for positiva, deve-se restar α vezes a derivada. Pelo contrário, se for negativo, deve-se somar α vezes a derivada para estar mais próximo do mínimo (SNYMAN, 2005). A Figura E.2 exibe o gráfico do resultado da atualização unidimensional.

Quando o algoritmo de otimização encontra o mínimo, seja global ou local, a derivada é zero, ou seja, o declive da tangente no ponto é zero, conforme mostrado na Figura E.2. O que origina que o parâmetro α se cancela.

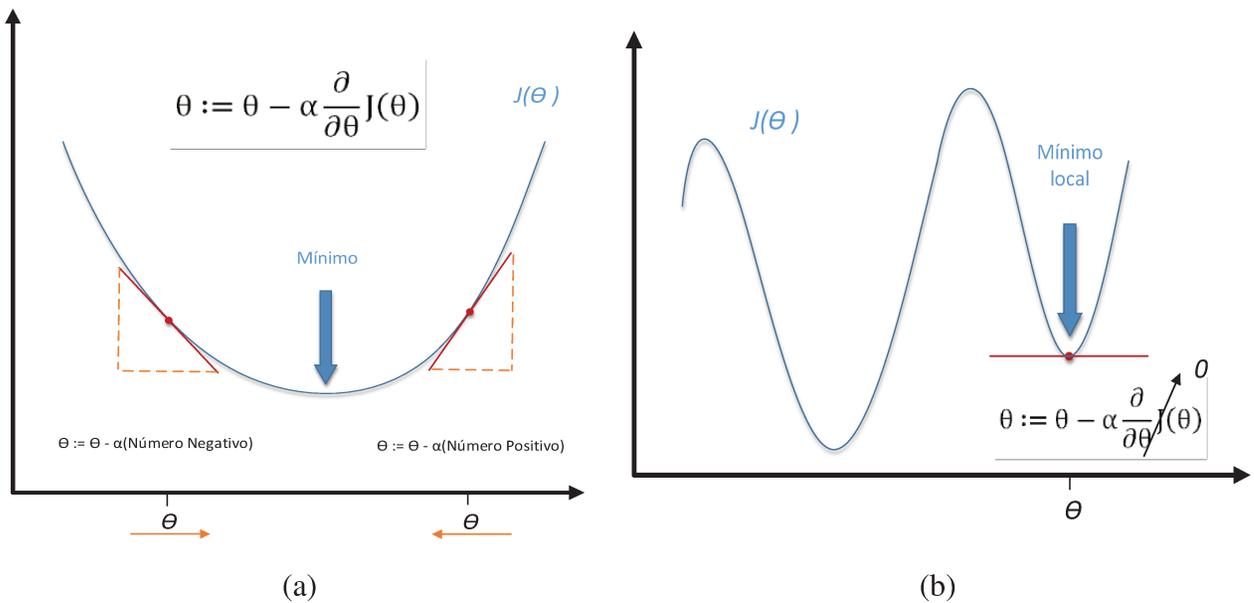


Figura E.2 (a) Derivada Encontrando o Mínimo com Atualização Unidimensional (b) Mínimo Local da Função.

Quando se tem múltiplos parâmetros θ , a atualização deve ser simultânea, já que todos os θ são utilizados para encontrar cada parâmetro da função (NG, 2012). O algoritmo gradiente descendente é exibido a seguir:

Algoritmo E.1 Algoritmo Gradiente Descendente Batch

Entrada: Parâmetros θ , Função de Custo $J(\theta)$ e Gradiente $\frac{\partial}{\partial \theta_j} J(\theta)$

Saída: Atualização dos Parâmetros θ .

```

for  $i := 1$  to Convergir do
  for  $j := 0$  to  $n$  do
     $temp_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1, \dots, \theta_n)$ 
  for  $j := 0$  to  $n$  do
     $\theta_n := temp_n$ 

```

E.2 BFGS

Os métodos Quase-Newton consistem em construir uma estimativa da Hessiana baseado em informação sobre o valor da função e do gradiente. Sendo o método mais utilizado é o Broyden, Fletcher, Goldfarb, Shanno ou BFGS (HALL,2013) e (NG, 2012).

Algoritmo E.2 Algoritmo Gradiente Descendente Mini-Batch

Entrada: Parâmetros θ , Função de Custo $Cost$ e Gradiente $\frac{\partial}{\partial \theta_j} Cost$

Saída: Atualização dos Parâmetros θ .

Auxiliar: Número de amostras m .

for $k := 1$ **to** Convergir **do**

$$d^k := -B_K \nabla f(x^k)$$

$$\alpha = \operatorname{argmin}_\alpha f(x^k + \alpha d^k)$$

$$x^{k+1} \leftarrow x^k + \alpha d^k$$

$$s_k = \alpha d^k = x^{k+1} - x^k$$

$$y_k = \nabla f(x^{k+1}) - \nabla f(x^k)$$

$$H^{k+1} \leftarrow \left(I - \frac{y_k s_k^T}{y_k^T s_k} \right)^T H^k \left(I - \frac{y_k s_k^T}{y_k^T s_k} \right) + \frac{s_k s_k^T}{y_k^T s_k}$$

Os métodos Quase-Newton utilizam a busca linear, onde se encontra a direção de descenso da função de custo. Dessa forma, busca-se o valor pela qual multiplicar o vetor da direção d para obter um decrescimento α , assim, satisfazer as condições Wolfe (HALL,2013).

1. Tomar $\alpha = 1$.
2. Verificar as condições de Wolfe.
3. Se não satisfazem as condições, define-se $\alpha = (1/2)\alpha$, atribui-se $\alpha_k = \alpha$ e volta ao passo 1, caso contrário $x^{k+1} \leftarrow x^k + \alpha d^k$.

APÊNDICE F. - Regressão Polinomial

Regressão linear é uma aprendizagem supervisionada (ABU-MOSTAFA,2012). Embora, pode converter-se em uma regressão polinomial, a qual permite ser utilizada em funções não lineares (NG, 2012).

F.1 Representação do Modelo

Na Equação (F.1) é mostrada a hipótese da regressão polinomial,

$$\begin{aligned} h_{\theta}(x) &= \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_n x^P \\ \theta &= \text{Parâmetros obtidos do Modelo} \\ n &= \text{Número de características do polinômio} \\ P &= \text{Grau do Polinômio} \end{aligned} \quad (\text{F.1})$$

Por conveniência de notação definiu $x_0 = 1$, já que deve-se colocar uma característica adicional, chamada de bias e seu parâmetro é θ_0 , na equação a seguir,

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}, \quad h_{\theta}(x) = \theta^T x. \quad (\text{F.2})$$

F.2 Função de Custo

A função de custo ajuda a descobrir como encaixar os parâmetros θ ao melhor ajuste aos dados. Na regressão temos um conjunto de treinamento $(x(i), y(i))$ de tamanho m e a hipótese h . A hipótese h é definida pelo conjunto $[\theta_0, \theta_1, \dots, \theta_n]$, os quais são os pesos ou parâmetros do modelo.

Como resultado tem-se um problema de minimização (Refere-se APÊNDICE E), ou seja, a diferença entre $h(x)$ e y deveria ser o menor possível, minimizando o quadrado de diferença entre a hipóteses e a saída, conforme a Equação (F.3),

$$\underbrace{\underset{\theta_0, \theta_1, \dots, \theta_n}{\text{minimize}} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2}_{\underset{\theta_0, \theta_1, \dots, \theta_n}{\text{minimize}} J(\theta_0, \theta_1, \dots, \theta_n)} \quad . \quad (\text{F.3})$$

Existem outras funções de perda ou custo para problemas de regressão, porém o erro quadrático é a mais utilizado em métodos como Regressão Polinomial e Rede Neural. Além disso, implementou-se regularização no algoritmo de regressão polinomial, dada por pela Equação (F.3), sendo, λ o fator de regularização (NG, 2012), conforme mostrado na Equação (F.4),

$$\underbrace{\underset{\theta_0, \theta_1, \dots, \theta_n}{\text{minimize}} \frac{1}{2m} \left[\sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^n \theta_j^2 \right]}_{\underset{\theta_0, \theta_1, \dots, \theta_n}{\text{minimize}} J(\theta_0, \theta_1, \dots, \theta_n)} \quad (\text{F.4})$$

F.3 Regressão Polinomial (Gradiente)

Tem-se que minimizar a função de custo $J(\theta)$. É necessário realizar a derivada parcial da $J(\theta)$ para obter o gradiente,

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2. \quad (\text{F.5})$$

Encontramos a derivada parcial, ou seja, a derivada com respeito a cada termo θ . Com a derivada externa se cancela o dois que acompanha a m . Na derivada interna só o θ_j que está acompanhado de $x^{(i)}$, fica multiplicado por $x^{(i)}$, como,

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})x^{(i)}. \quad (\text{F.6})$$

Completado a derivada de $J(\theta)$ com respeito aos parâmetros θ . Seguido a isto, deve ser utilizado o algoritmo de otimização baseado em gradiente (refere-se APÊNDICE E), para atualizar os θ , e poder achar o mínimo da função da hipóteses (ABU-MOSTAFA, 2012), dada por

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})x^{(i)}. \quad (\text{F.7})$$

É necessário realizar a atualização simultânea, já que para o cálculo de cada parâmetro θ_j , são necessário todos os parâmetros θ restantes. O parâmetro de bias θ_0 , não se aplica regularização. Como se mostra no Algoritmo F.1.

Algoritmo F.1 Gradiente para regressão polinomial.

Entrada: Parâmetros θ , Função de Custo $J(\theta)$ e regularização R .

Saída: Gradiente $\frac{\partial}{\partial \theta_j} J(\theta)$

for $i := 1$ **to** $\text{Size}(\theta)$ **do**

if j **more** 2 **do**

$\frac{\partial}{\partial \theta_j} R =$ *Derivada da regularização.*

$\frac{\partial}{\partial \theta_j} J(\theta) :=$ *Derivada da função de custo % sin regularização*

else

$\frac{\partial}{\partial \theta_j} J(\theta) :=$ *Derivada da função de custo de θ_0*

APÊNDICE G. – Rede Neural

A resposta de precisar de outro algoritmo de aprendizagem tendo regressão polinomial (APÊNDICE F), é que na maioria dos problemas são necessárias aprender hipóteses complexas não lineares. O algoritmo está baseado no caso de classificação de RNA (APÊNDICE D).

G.1 Representação do Modelo

Tendo que as Redes Neurais são um algoritmo para funções não lineares, tem-se a função de ativação (nesta pesquisa foi utilizada sigmoide e tangente) $g(z)$. Os parâmetros θ , valores das conexões entre unidades são chamados de pesos (ABU-MOSTAFA,2012).

A primeira camada está conformada pelos parâmetros de entrada. A camada última é conhecida como camada de saída, na qual tem-se a hipótese final. As camadas que se encontram entre as camadas de entrada e saída são chamadas de camadas ocultas, são as responsáveis de transformar as entradas em características mais úteis.

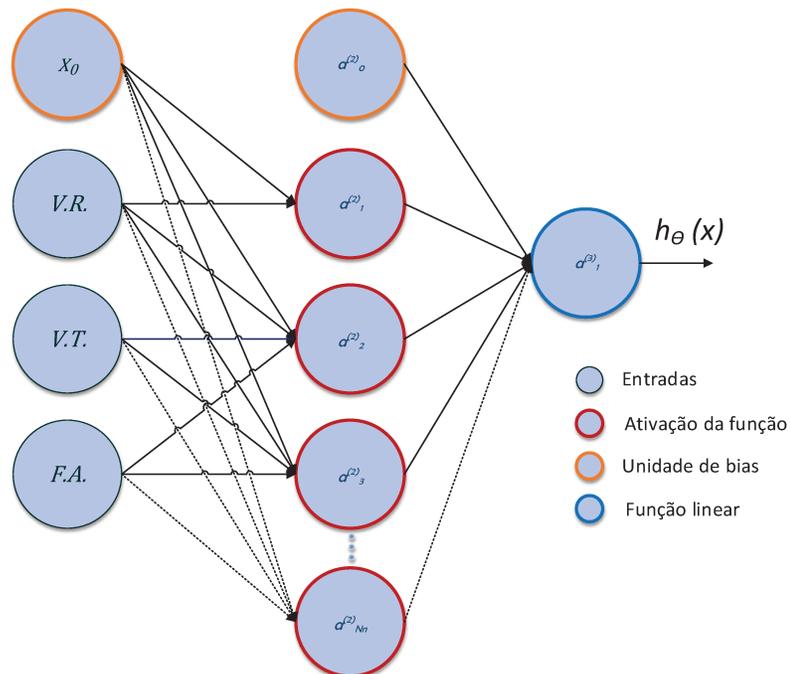


Figura G.1 Arquitetura da RNA para o Modelo FSW.

O valor computado pelo neurônio da camada de saída é uma combinação linear, como resultado a obter uma saída contínua. Isto se deve a que o problema desta pesquisa é regressão e não classificação. O procedimento se realiza até achar o mínimo erro da hipótese.

G.2 Função de Custo

A função de custo utilizada é erro médio quadrático, anteriormente explicado no algoritmo de regressão polinomial (Refere-se APÊNDICE F) (NG, 2012), mostrada na Equação (G.1)

$$J(\theta) = \frac{1}{2m} \left[\sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^l \sum_{i=1}^n \theta_{ij}^2 \right]. \quad (\text{G.1})$$

G.3 Forward Propagation

Na Equação (G.2), tem uma combinação linear dos valores de entradas vezes os parâmetros θ , sendo definida como z . Aplicada à função de ativação nesta pesquisa foi utilizada tanto sigmoide como tangente, onde z vai ser definida pelos parâmetros da rede e os valores da camada anterior,

$$\begin{aligned} a_i^{(j)} &= g(z_i^{(j)}) \\ z_i^{(j)} &= \theta_i^{(j-1)} a_i^{(j-1)}, \end{aligned} \quad (\text{G.2})$$

assim, para a propagação da camada oculta e a camada de saída, tem-se que colocar a unidade de bias na camada oculta como foi colocado na camada de entrada, com o valor de $a_0^{(2)} = 1$. Sendo o problema de regressão a última camada, tem função de linear.

$$\begin{aligned} a^{(3)} &= z^{(3)} \\ h_{\theta}(x) &= a^{(3)}. \end{aligned} \quad (\text{G.3})$$

A rede neural pode ter diferentes arquiteturas, o termo de arquitetura pode referir-se aos diferentes neurônios conectados entre eles. Na primeira camada tem-se as características do

modelo, na seguinte camada, as características são convertidas em características mais complexas, como resultado pode-se obter uma hipótese não linear.

Algoritmo G.1 Forward Propagation

Entrada: Entradas x e Saídas y .

Saída: Função de custo $J(\theta)$

Auxiliar: Número de amostras m .

```

 $\Delta_{ij}^{(l)} = 0$ ; % ajustar para todas  $l, j, i$ , o erro
for  $i := 1$  to  $m$  do
    ajustar  $a^{(1)} = x$ 
    obter os  $\theta_{ij}^{(l)}$ 
     $z_i^{(l)} = \theta_i^{(l-1)} a_i^{(l-1)}$  %somando o termo bias  $a_i^{(0)}$ 
     $a_i^{(l)} = g(z_i^{(l)})$  %Utilizando alguma função de ativação excepto na última
    camada
     $a_i^{(Lend)} = h_{\theta}(x) = z_i^{(Lend)}$ 
erro =  $h_{\theta}(x) - y$ 
 $J(\theta) := \frac{1}{2m} \sum (\text{erro}^2) + \lambda \sum \sum \theta_{ij}^{(l)}$  %o que o mesmo que o erro quadrático medio

```

G.3 Back Propagation

O “Back propagation” é um algoritmo para poder minimizar a $J(\theta)$, o qual ajuda obter o gradiente da função (Refere-se APÊNDICE D). Por conseguinte, para obter os parâmetros da função precisa ter tanto $J(\theta)$ obtido por “Forward Propagation” como $\frac{\partial}{\partial \theta_{ij}} J(\theta)$ obtido por “Back propagation” (Refere-se APÊNDICE E).

Realizando a derivada do erro com respeito à saída explicado no APÊNDICE D. Sendo l o número da última camada, a Equação (G.4) fica,

$$\delta_j^{(l)} = a_j^{(l)} - y_j, \quad (\text{G.4})$$

onde, se obtém os δ de cada neurônio e multiplica-se pelos θ das conexões vezes a derivada da função de ativação sigmoide $g'(z)$ avaliados na combinação linear, conforme mostrado na Equação (G.6).

$$\begin{aligned} \delta_j &= \theta^T \delta_{j+1} .* g'(z_j) \\ g'(z) &= a(1 - a) \end{aligned} \quad (\text{G.5})$$

O procedimento tem-se que realizar por todas as amostras do treinamento, quer dizer, m vezes. O resultado do gradiente vai sendo somado. Achando o primer delta (δ) da última camada, da mesma forma, obtém-se os deltas (δ_j) das camadas anteriores. Como ideia geral, o δ soma os erros anteriores, a qual fez uma diferença no erro geral, e assim corrigir os seguinte parâmetros θ , ou seja, os pesos da rede (ABU-MOSTAFA,2012).

Formalmente o erro de custo de cada ativação do neurônio é dado pela Equação (G.6). Sendo l o número da camada, e j o número do neurônio. A função $cost(i)$ é dada dependendo da classe do algoritmo para classificação a função de custo foi explicada acima,

$$\begin{aligned} \delta_j^l &= \frac{\partial y}{\partial z_j^{(l)}} cost(i) \\ cost(i) &= \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \end{aligned} \quad (\text{G.6})$$

Simplificando o processo tem-se: 1. os dados de entrada apresentados para a rede são propagados através da rede até chegar à última camada. Este processo produz uma predição da saída mostrado no Algoritmo G.1 ; 2. a saída predita é obtida da saída atual e um valor de erro é calculado; 3. a rede utiliza o Algoritmo G.2 para obtenção dos pesos, começando pela última camada até chegar à primeira, realizando propagação para atrás; 4. quando a propagação para

atrás foi feita, começa-se de novo a propagação de frente, realizando um ciclo até a minimização do erro predito e atual.

Algoritmo G.2 Back Propagation

Entrada: Entradas x e Saídas y .

Saída: Gradiente $\frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta)$

Auxiliar: Número de amostras m .

$\Delta_{ij}^{(l)} = 0$; % ajustar para todas l, j, i , o erro

for $i := 1$ **to** m **do**

ajustar $a^{(1)} = x^i$

Desenvolver Forward Propagation para computar $a^{(l)}$

Utilizando $y^{(i)}$, computar $\delta^{(Lend)} = a^{(Lend)} - y^{(i)}$

Computar $\delta^{(Lend-1)}, \delta^{(Lend-2)}, \dots, \delta^{(2)}$

$\Delta_{ij}^{(l)} := \Delta_{ij}^{(l)} + a_{ij}^{(l)} \delta_{ij}^{(l+1)}$

$\frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta) := \frac{1}{m} \Delta_{ij}^{(l)} + \lambda \theta_{ij}^{(l)}$ %sem incluir o termo de bias $\theta_{i0}^{(l)}$

ANEXO i. – Regressão Polinomial Arquitetura

i.1 Arquitetura

Realizou-se o algoritmo Regressão Polinomial (Refere-se APÊNDICE F) para determinar as saídas do comportamento da solda FSW em corpos de prova. Como resultado tem-se a informação das entradas e saídas dadas pelos fatores da solda (FSW – Velocidade de translação; velocidade de rotação e força axial) e suas respostas dadas pelos ensaios de tração, respetivamente. Na pesquisa foram desenvolvidos 5 modelos de regressão polinomial, com diferentes ordens de polinômio, mostrados na Tabela i.1. Por exemplo, para o modelo 2, tem-se um polinômio de ordem 2.

Por conseguinte, as entradas $x = [x_1 \ x_2 \ x_3]$, são transformadas em novas características $[x_1 \ x_2 \ \dots \ x_f \ x_1x_2 \ x_1x_3 \ \dots \ x_fx_f \ x_1x_2^2 \ x_1x_3^2 \ \dots \ x_f^{Nm}x_f^{Nm}]$. Sendo x_1, x_2 e x_3 ; velocidade de translação, velocidade de rotação e força axial, respetivamente.

Tabela i.1 Modelos de Regressão Polinomial

Polinomial	Modelos de Regressão Polinomial				
	Modelo 1	Modelo 2	Modelo 3	Modelo 4	Modelo 5
Polinômio	1	2	3	4	5

O algoritmo utilizado de otimização foi BFGS (Refere-se APÊNDICE E). As opções utilizadas são: máximo número de iterações de 1000, máxima funções avaliadas de 1000, Inicialização do passo inicial foi quadrática, comprimento do passo foi com interpolação e extrapolação cúbica e a tolerância de otimização e progresso foram $1e^{-15}$.

i.2 Seleção dos Modelos da Regressão Polinomial

Para cada modelo realizou-se diferentes parâmetros de regulação (λ) obtendo o menor erro quadrático médio (MSE). Se realizaram 5 diferente modelos mostrados na Tabela i.2, no qual cada modelo tem um grau de polinômio, sendo modelo 1 com grau polinomial 1, modelo 2 com grau polinomial 2 e assim para frente.

Tabela i.2 Resultados de Treinamento-Validação para os Modelos Desenvolvidos de Regressão Polinomial.

Modelo	Tensão Escoamento			Tensão Última			Alongamento		
	(λ)	Treinamento (MSE)	Validação (MSE)	(λ)	Treinamento (MSE)	Validação (MSE)	(λ)	Treinamento (MSE)	Validação (MSE)
Modelo 1	0.3	2.6864	5.2939	10	57.3221	91.4111	30	5.7837	7.2643
Modelo 2	0.3	0.9629	7.3943	30	44.9536	99.8508	30	3.5432	6.0635
Modelo 3	10	1.2194	7.3622	10	17.3047	59.6310	30	4.5085	7.3546
Modelo 4	10	0.9940	13.7643	30	20.6128	99.7724	30	3.4654	10.1085
Modelo 5	30	1.5015	13.2779	30	20.0379	96.2976	30	3.1139	13.9484

i.3 Evolução dos Modelos Finais do Regressão Polinomial

Os resultados obtidos dos modelos selecionados (TE), (TU) e (AF) são mostrados na Tabela i.3 e Tabela i.4. A otimização para o algoritmo foi utilizado BFGS (Quase-Newton), no qual colocou-se diferentes números de iterações para observar o comportamento do algoritmo (Refere-se APÊNDICE E) na Tabela i.5 e Figura i.1.

Tabela i.3 Parâmetros e (MSE) finais obtidos por o método validação cruzada para Regressão Polinomial.

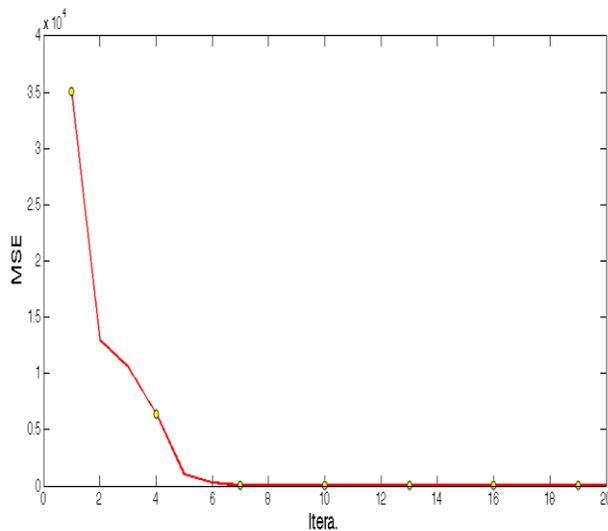
	Modelo	(λ)	Treinamento (MSE)	Validação (MSE)
Tensão Escoamento	2	0.3	0.9629	7.3943
Tensão Última	3	10	17.3047	59.6310
Alongamento	2	30	3.5432	6.0635

Tabela i.4 Erros (MSE) e Coeficiente de determinação (R^2) de Treinamento-Validão e Teste para os Modelos Finais da Regressão Polinomial.

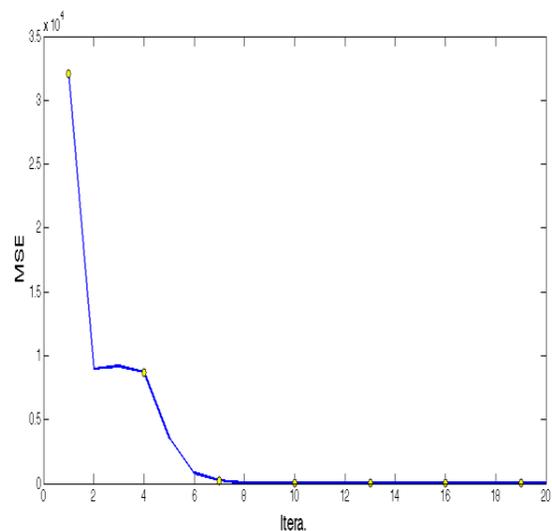
	Modelo	(λ)	Treina.-Val. (MSE)	Teste (MSE)	Treina.-Val. (R^2)	Teste (R^2)
Tensão Escoamento	2	0.3	1.0296	2.2639	0.9306	0.8834
Tensão Última	3	10	17.6353	45.7775	0.8126	0.0211
Alongamento	2	30	4.5582	1.5707	0.3003	0

Tabela i.5 Regressão Polinomial - Treinamento e Teste dos Modelos Seleccionados pelas Iterações.

Iterações	Tensão Escoamento			
	Treinamento (R ²)	Treinamento (MSE)	Teste (R ²)	Teste (MSE)
1	0	3504.5	0	3205.3
4	0	643.39	0	870.25
7	0	31.7192	0	246.388
10	0.7566	3.6105	0	29.6011
13	0.8165	2.7213	0.3561	12.4914
16	0.9305	1.0316	0.8813	2.3045
19	0.9306	1.0296	0.8834	2.2630



(a)



(b)

Figura i.1 Regressão Polinomial - Evolução dos Erros dos Modelos MSE – Iterações: (a) Treinamento da Tensão de Escoamento (b) Teste da Tensão de Escoamento.

ANEXO ii. – Rede Neural Arquitetura

ii.1 Arquitetura

Realizou-se o algoritmo RNA para determinar as saídas do comportamento da solda FSW em corpos de prova. A primeira camada está conformada pelos parâmetros de entrada da solda dos corpos de prova. (VT, VR e FA) (Refere-se APÊNDICE G).

Na pesquisa foram desenvolvidos 8 modelos de RNA, com diferentes estruturas da rede mostrados na Tabela ii.1. O procedimento feito foi:

1. Arquitetura:

- Número de unidades de entradas: Dimensão das características x . Os fatores estudados foram a velocidade de translação; velocidade de rotação e força axial.
- Número de unidades de saída: Dimensão das funções de regressão. As respostas foram as que apresentaram o fator de curvatura (não linearidade) por meio de análises estatístico (Seção 5.1.2).
- Número de unidades ocultas: Nesta pesquisa foram desenvolvidos modelos desde 5 neurônios até 20 neurônios por camada, também foram desenvolvidos tanto uma camada como duas e realizaram-se com diferentes combinações de função de ativação (sigmoide e tangente).

2. Treinamento: São necessários realizar alguns procedimentos para garantir o bom funcionamento da RNA.

- Randomizar os parâmetros iniciais, muito perto do zero.
- Desenvolver “*Forward Propagation*” (Refere-se APÊNDICE G).
- Desenvolver “*Back Propagation*” (Refere-se APÊNDICE G).
- Desenvolver ou utilizar métodos de otimização (BFGS) (Refere-se APÊNDICE E)

Para o teste de precisão da rede treinada, foi adoptado o coeficiente de determinação R^2 . Similarmente como foi feito no algoritmo regressão polinomial (Refere-se APÊNDICE A).

Tabela ii.1 Modelos da RNA.

Camadas	Número de Neurônios							
	Modelo 1	Modelo 2	Modelo 3	Modelo 4	Modelo 5	Modelo 6	Modelo 7	Modelo 8
Entrada	3	3	3	3	3	3	3	3
Oculto 1	5	10	10	20	10	5	20	20
Oculto 2	-	-	-	-	10	10	15	20
Saída	1	1	1	1	1	1	1	1
Funções dos Neurônios								
Oculto 1	Log-Sig	Tan-Sig	Log-Sig	Log-Sig	Log-Sig	Tan-Sig	Log-Sig	Log-Sig
Oculto 2	-	-	-	-	Log-Sig	Tan-Sig	Log-Sig	Log-Sig
Saída	Purelin	Purelin	Purelin	Purelin	Purelin	Purelin	Purelin	Purelin

ii.2 Seleção dos Modelos da Rede Neural

Por cada modelo realizou-se diferentes parâmetros de regulação (λ) obtendo o menor erro quadrático médio (MSE). Se realizaram 8 diferente modelos, no qual cada modelo tem uma estrutura da rede diferente. Na Tabela ii.2 são mostrados os melhores parâmetros.

Tabela ii.2 Resultados de Treinamento-Validação para os Modelos Desenvolvidos de Redes Neurais.

Modelo	Tensão Escoamento			Tensão Última			Alongamento		
	(λ)	Treinamento (MSE)	Validação (MSE)	(λ)	Treinamento (MSE)	Validação (MSE)	(λ)	Treinamento (MSE)	Validação (MSE)
Modelo 1	0.03	0.4662	6.6430	1	41.2142	74.1655	10	6.4946	7.2587
Modelo 2	0.03	0.4321	6.3501	3	33.0787	73.2362	30	6.4946	7.2587
Modelo 3	0.03	0.4133	6.2641	0.3	18.4474	70.7196	10	6.4946	7.2587
Modelo 4	0.03	0.2682	5.8287	0.03	11.2380	68.6136	10	6.4946	7.2587
Modelo 5	0.1	2.2676	8.7882	3	48.3821	82.7441	30	6.4946	7.2587
Modelo 6	0.3	0.8209	5.8191	1	12.7083	70.2363	10	6.4946	7.2587
Modelo 7	0.3	0.5573	5.8089	1	9.1797	69.6792	10	6.4946	7.2587
Modelo 8	0.3	0.4347	5.4122	1	8.9358	68.4075	10	6.4946	7.2587

ii.3 Evolução dos Modelos Finais da Rede Neural

Os resultados obtidos dos modelos selecionados da (TE), (TU) e (AF) são mostrados na Tabela ii.3 e Tabela ii.4. A otimização do treinamento foi utilizado Quase-Newton, no qual colocou-se diferentes números de iterações para observar o comportamento do algoritmo (Refere-se APÊNDICE E). Pode-se observar que encontramos a diminuição do erro e o aumento de coeficiente de determinação quando o algoritmo encontrou os melhores parâmetros dos modelos na Tabela ii.5 e Figura ii.1.

Tabela ii.3 Parâmetros e (MSE) Finais Obtidos por o Método Validação Cruzada para Rede Neural.

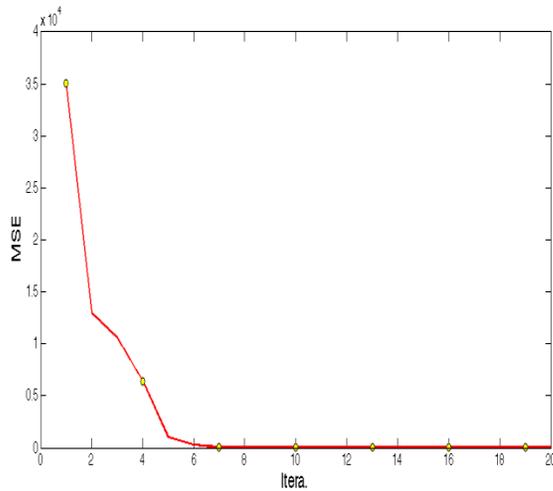
	Modelo	(λ)	Treinamento (MSE)	Validação (MSE)
Tensão Escoamento	8	0.3	0.4347	5.4122
Tensão Última	8	1	8.9358	68.4075
Alongamento	-	-	6.4946	7.2587

Tabela ii.4 Erros (MSE) e Coeficiente de Determinação (R^2) de Treinamento-Validação e Teste para os Modelos Finais da Rede Neural.

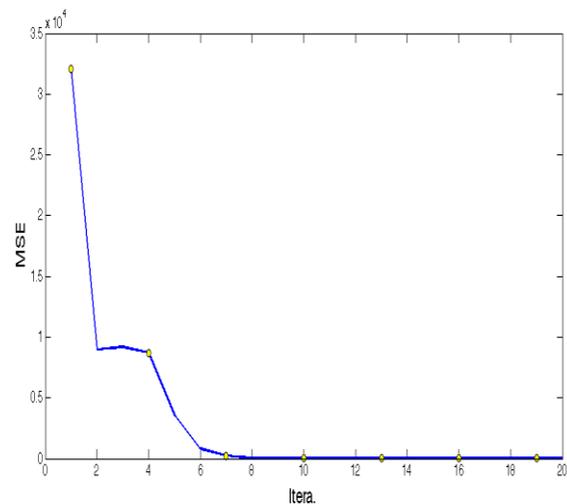
	Modelo	(λ)	Treina.-Val. (MSE)	Teste (MSE)	Treina.-Val. (R^2)	Teste (R^2)
Tensão Escoamento	8	0.3	0.3337	2.4282	0.9691	0.7809
Tensão Última	8	3	8.4045	44.2259	0.9107	0.366
Alongamento	-	-	6.5147	2.6906	0	0

Tabela ii.5 Rede Neural -Treinamento e Teste dos Modelos Selecionados pelas Iterações.

Iterações	Tensão Escoamento			
	Treinamento (R ²)	Treinamento (MSE)	Teste (R ²)	Teste (MSE)
51	0.6524	5.1559	0.6288	7.2054
151	0.9649	0.5208	0.6334	7.1151
251	0.9677	0.4788	0.8018	3.8474
351	0.9731	0.3994	0.8342	3.2203
451	0.9731	0.3988	0.8295	3.5030
551	0.9712	0.4988	0.8362	3.1799
651	0.9705	0.3998	0.7729	3.0896
751	0.9708	0.3988	0.8163	2.9640



(a)



(b)

Figura ii.1 Rede Neural - Evolução dos erros dos Modelos MSE – Iterações: (a) Treinamento da Tensão de Escoamento (b) Teste da Tensão de Escoamento.

ANEXO iii. - Resultados de Regressão Polinomial

iii.1 Curvas de Aprendizagem de Regressão Polinomial

Na Figura iii.1 mostra-se as curvas de aprendizagem (Seção 3.3.4).

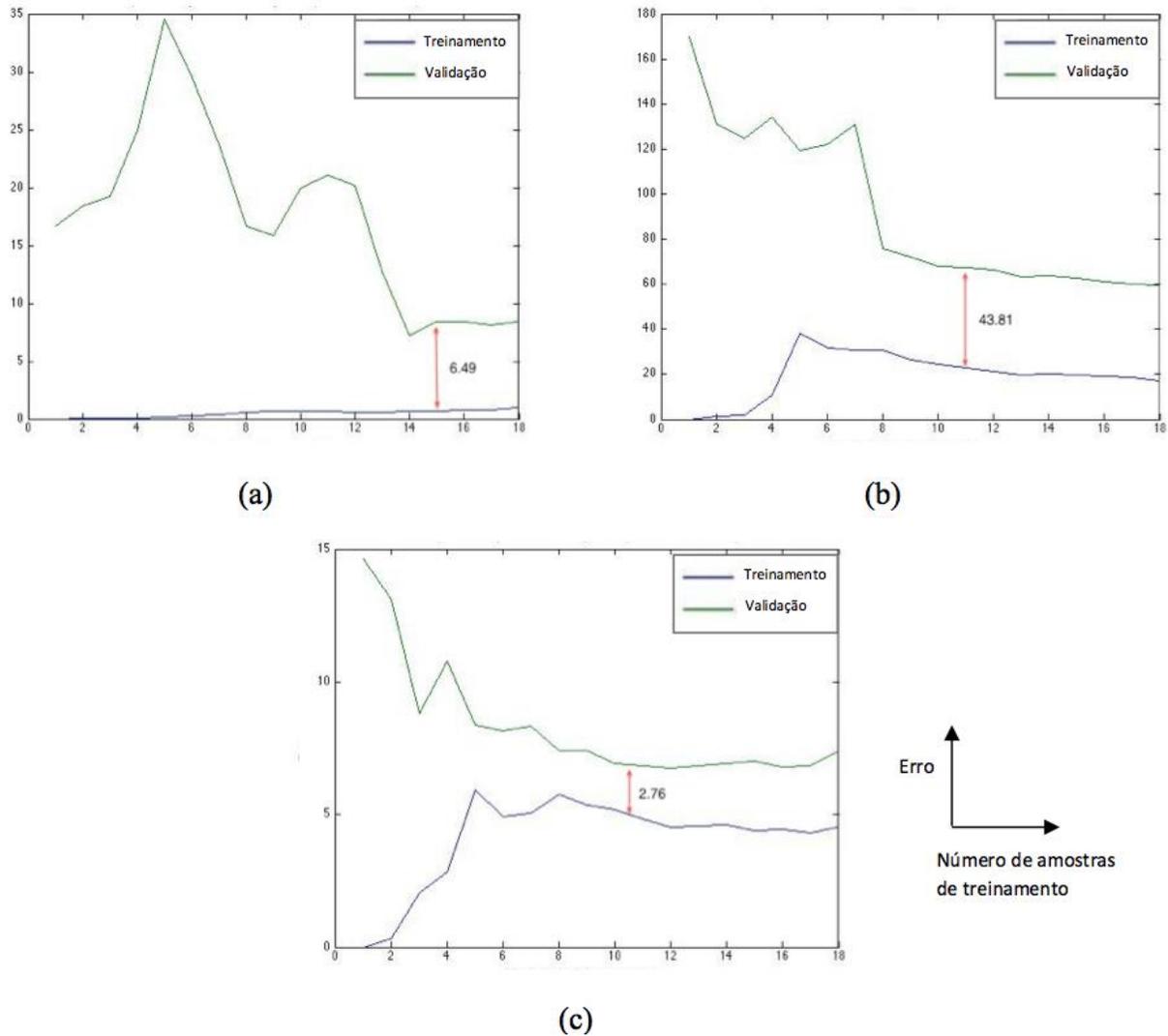


Figura iii.1 Regressão Polinomial - Curvas de Aprendizagem MSE - Sample: (a) Tensão de Escoamento (b) Tensão Última (c) Alongamento.

iii.2 Análise de Predição das Respostas da Regressão Polinomial

Foi utilizada a porcentagem de erro da média absoluta (MAPE) mostrada na Tabela iii.1 para as respostas finais de tensão de escoamento (TE), tensão última (TU) e alongamento (AF) (Refere-se APÊNDICE A) e as respostas de superfície Figura iii.2.

Tabela iii.1 A Porcentagem do Erro (MAPE) de Treinamento e Validação para Regressão Polinomial

	Modelo	(λ)	Treinamento (MAPE) %	Teste (MAPE) %
Tensão Escoamento	2	0.3	0.3631	0.5340
Tensão Última	3	10	0.9305	1.6163
Alongamento	2	30	21.9016	18.7460

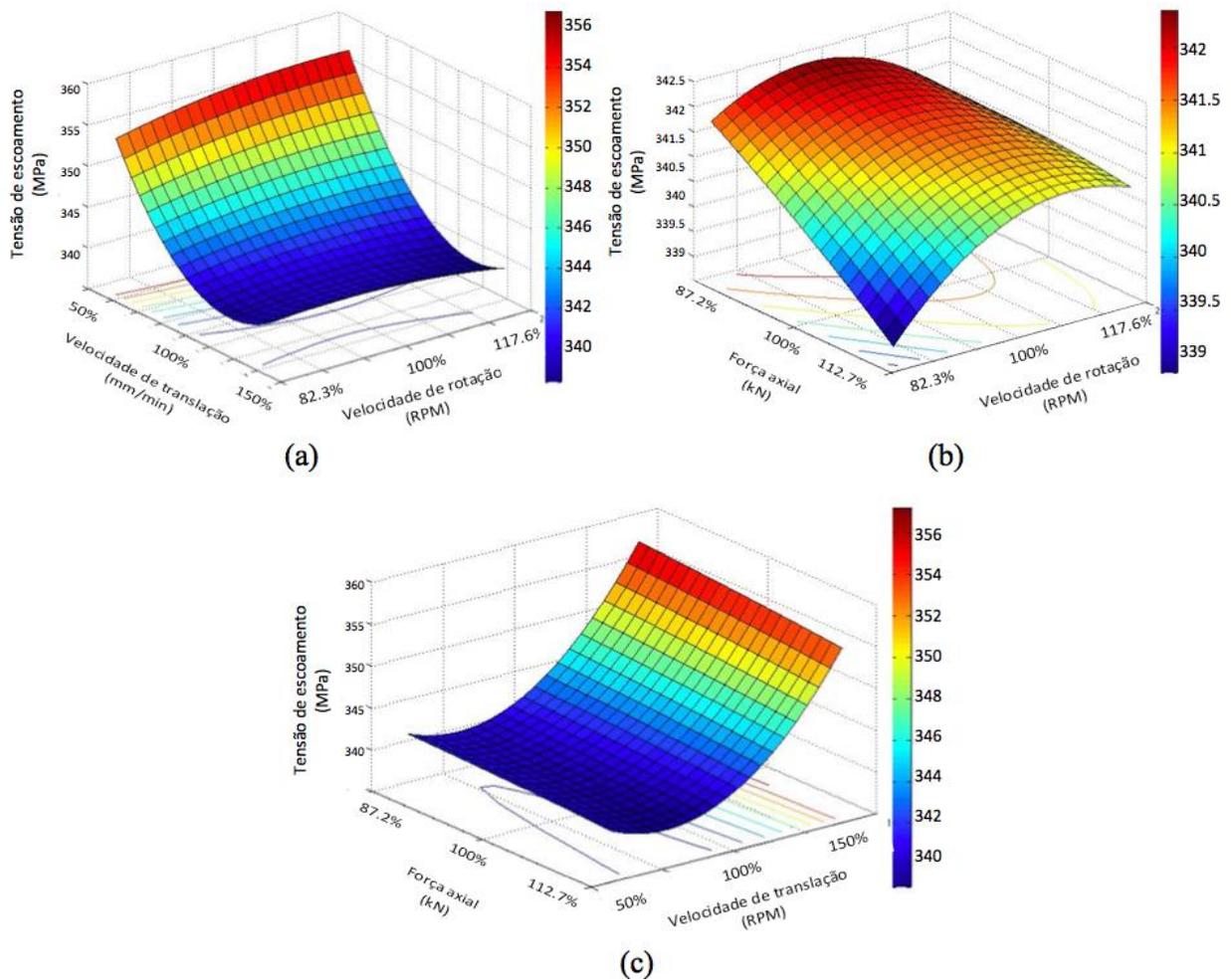


Figura iii.2 Superfície de Resposta da Tensão de Escoamento por o Algoritmo Regressão Polinomial; (a) VR-VT; (b) VR-FA; (c) VT-FA.

iii.3 Todos os Modelos da Regressão Polinomial

Tabela iii.2 Resultados de Treinamento e Teste para os Modelos Desenvolvidos de Regressão Polinomial.

Modelo	No. Dados Treinamento	No. Dados Teste	Tensão Escoamento		Tensão Última		Alongamento	
			Treinamento (MSE)	Test (MSE)	Treinamento (MSE)	Test (MSE)	Treinamento (MSE)	Test (MSE)
Modelo 1	19	5	2.7412	3.2289	57.9055	57.5837	5.8127	2.5707
Modelo 2	19	5	1.0296	2.2639	45.4390	48.6961	4.5582	1.5707
Modelo 3	19	5	1.2359	2.6311	17.6353	45.7775	3.5637	2.8297
Modelo 4	19	5	1.0060	11.5929	20.9458	36.3649	3.5345	3.7510
Modelo 5	19	5	1.5223	29.4859	20.5333	59.8967	3.1772	3.8469
			Treinamento (R^2)	Test (R^2)	Treinamento (R^2)	Test (R^2)	Treinamento (R^2)	Test (R^2)
Modelo 1	19	5	0.8152	0.8336	0.3848	0	0.1078	0
Modelo 2	19	5	0.9306	0.8834	0.5173	0	0.3003	0
Modelo 3	19	5	0.9167	0.8644	0.8126	0.0211	0.4530	0
Modelo 4	19	5	0.9322	0.4027	0.7775	0.2224	0.4575	0
Modelo 5	19	5	0.8974	0	0.7819	0	0.5123	0

ANEXO iv. - Resultados de Rede Neural.

iv.1 Curvas de Aprendizagem de Neural

Na Figura iv.1 mostra-se as curvas de aprendizagem (Seção 3.3.4).

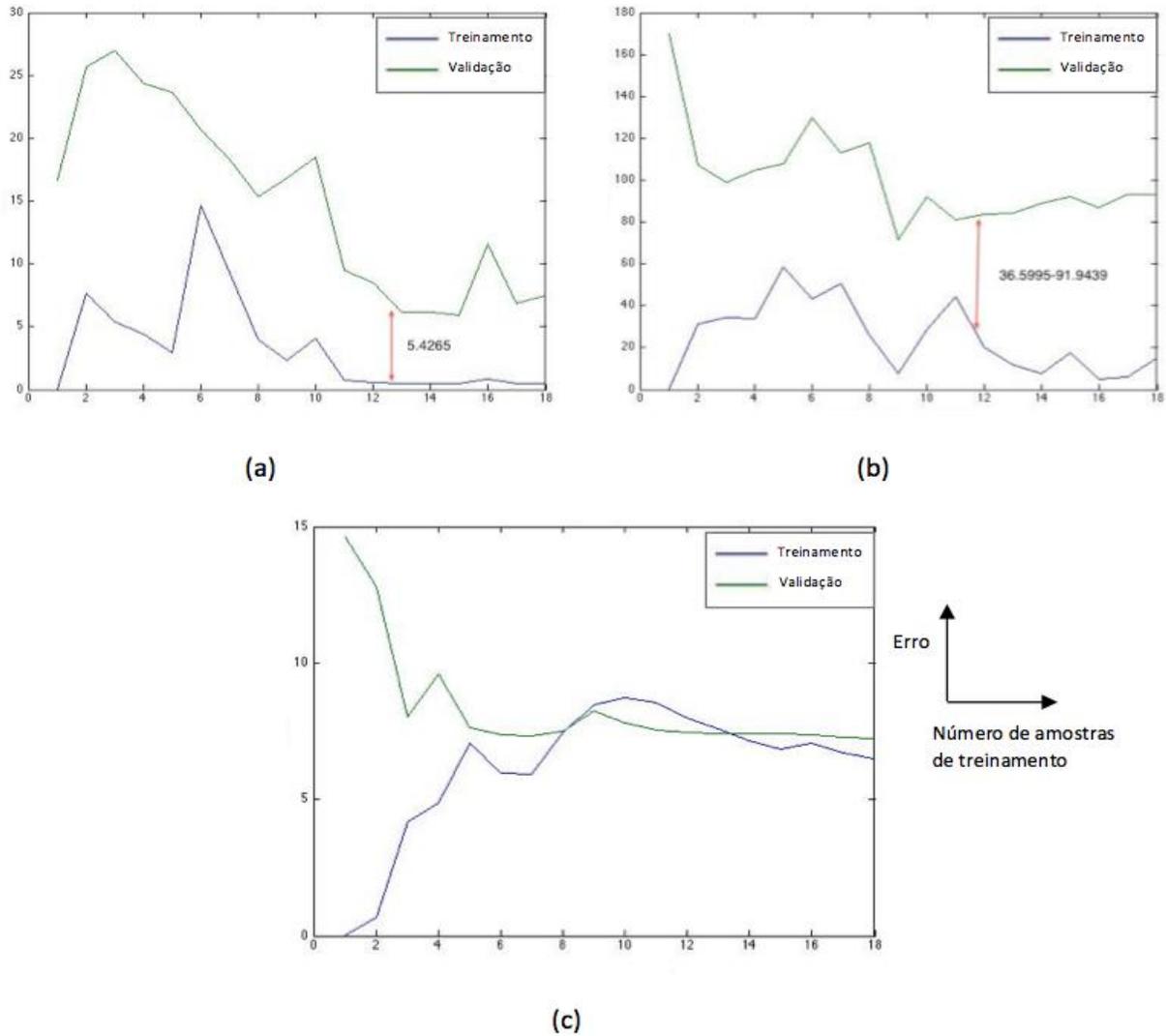


Figura iv.1 Rede Neural - Curvas de Aprendizagem MSE - Sample: (a) Tensão de Escoamento (b) Tensão Última (c) Alongamento.

iv.2 Análise de Predição das Respostas da Rede Neural

Foi utilizada a porcentagem de erro da média absoluta (MAPE) mostrada na Tabela iv.1 para as respostas finais de tensão de escoamento (TE), tensão última (TU) e alongamento (AF) (Refere-se APÊNDICE A) e as respostas de superfície Figura iv.2.

Tabela iv.1 A porcentagem do Erro (MAPE) de Treinamento e Validação para Rede Neural.

	Modelo	(λ)	Treinamento (MAPE) %	Teste (MAPE) %
Tensão Escoamento	2	0.3	0.2447	0.4644
Tensão Última	3	10	0.6518	1.4858
Alongamento	2	30	20.1042	35.2466

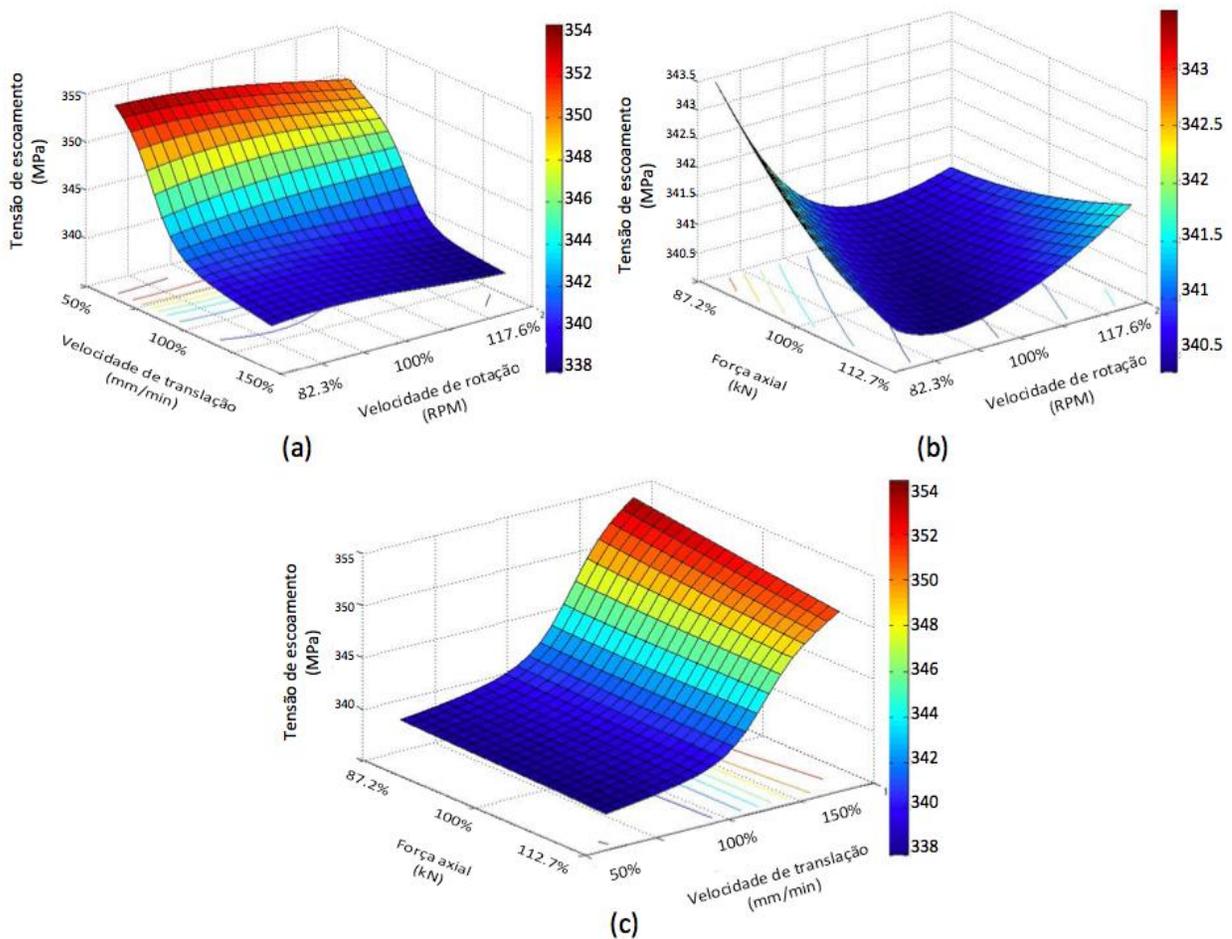


Figura iv.2 Superfície de Resposta da Tensão de Escoamento por o Algoritmo RNA; (a) VR-VT; (b) VR-FA; (c) VT-FA;

iv.3 Todos os Modelos do Regressão Polinomial

Tabela iv.2 Resultados de Treinamento e Teste para os Modelos Desenvolvidos de Rede Neural.

Modelo	Função	Camada Oculta	Tensão Escoamento		Tensão Última		Alongamento	
			Treinamento (MSE)	Test (MSE)	Treinamento (MSE)	Test (MSE)	Treinamento (MSE)	Test (MSE)
Modelo 1	Log-Sig	5	0.4368	5.6661	29.0425	58.0873	6.5147	2.6906
Modelo 2	Tan-Sig	10	0.3644	5.5815	22.5510	57.0028	6.5147	2.6906
Modelo 3	Log-Sig	10	0.3719	5.6707	18.7563	56.8892	6.5147	2.6906
Modelo 4	Log-Sig	20	0.2518	4.2820	11.5465	52.4000	6.5147	2.6906
Modelo 5	Tan-Sig	5-10	2.2432	8.7595	32.6998	62.6340	6.5147	2.6906
Modelo 6	Log-Sig	10-10	0.6599	5.2117	10.9836	56.1358	6.5147	2.6906
Modelo 7	Log-Sig	20-15	0.4583	3.8459	9.1933	53.5623	6.5147	2.6906
Modelo 8	Log-Sig	20-20	0.3337	2.4282	8.4045	44.2259	6.5147	2.6906
			Treinamento (R^2)	Test (R^2)	Treinamento (R^2)	Test (R^2)	Treinamento (R^2)	Test (R^2)
Modelo 1	Log-Sig	5	0.9601	0.7139	0.6658	0	0	0
Modelo 2	Tan-Sig	10	0.9730	0.7365	0.6860	0	0	0
Modelo 3	Log-Sig	10	0.9746	0.7078	0.8007	0	0	0
Modelo 4	Log-Sig	20	0.9830	0.7279	0.8773	0.0021	0	0
Modelo 5	Tan-Sig	5-10	0.8488	0.5887	0.5464	0	0	0
Modelo 6	Log-Sig	10-10	0.9555	0.7315	0.8833	0.0339	0	0
Modelo 7	Log-Sig	20-15	0.9579	0.7503	0.9130	0.339	0	0
Modelo 8	Log-Sig	20-20	0.9691	0.7809	0.9107	0.366	0	0