

# Agradecimentos

A Deus que me concedeu luz, paz, saúde, energia, inteligência, para levar adiante a missão de estudar.

Ao meu esposo Wellington e aos meus filhos Gabriel, Luísa Gabriela e Lívia Gabriela, que me proporciona uma vida equilibrada e cheia de paz. Obrigada por me acompanhar e me incentivar nas horas mais difíceis. Obrigada pela compreensão e por estarem sempre ao meu lado.

A minha mãe Marlene Rita, por sempre apoiar os meus projetos, me dando força e incentivo para seguir em frente.

Ao meu pai Walmir Marcolino, que apesar de não estar mais entre nós, sempre me deu força para seguir com meus estudos, e hoje mais que nunca estaria feliz com a minha missão cumprida.

A todos os meus irmãos e amigos que nos momentos difíceis, me deram força pra seguir essa caminhada.

Ao professor Dr. Edwirde Luiz e a professora Msc. Ana Patricia, pela amizade, pelo respeito, pela dedicação e pelos incentivos concedidos a mim no desenvolvimento deste trabalho e na minha vida profissional.

A todos os professores do Curso, por todos os conhecimentos e experiências transmitidos, que muito contribuíram para a realização de meus trabalhos e que levarei comigo por toda a minha vida.

Aos colegas de estudo, especialmente a Nathielly, Arielly, Wanessa, Adriana, Analu e Samara pela companhia diária, pela amizade, pelos conhecimentos transmitidos, pelo companheirismo e principalmente pela paciência ao longo desses quatro anos de estudos.

Obrigada a todos, por sonharem comigo e me ajudarem a concretizar o meu sonho.

# Resumo

Este trabalho apresenta uma maneira de implementar suporte à decisão de usar redes neurais artificiais (RNA) juntamente com estatística univariada e multivariada. Na proposta, a RNA do tipo perceptron de múltiplas camadas (MLP) e dois experimentos. No primeiro experimento, usou-se uma regressão polinomial de grau 7, responsável pelo ajuste e predição de uma função específica  $y = x \cos(x)$  perturbada com um erro  $e \sim N(0, 1)$ . Em que, a rede (MLP) e a regressão polinomial de grau 7 apresentaram resultados satisfatórios. Além disso, esse primeiro experimento também tem como objetivo identificar o desempenho das técnicas usadas compará-las com o erro médio quadrático gerado por ambas as técnicas. Para o segundo experimento, criou-se uma função do tipo  $y = humps(x)$  perturbada com um erro  $e \sim N(0, 1)$  de forma que a rede MLP e uma regressão polinomial de grau 19 que aproximem a função de Humps perturbada, avaliando o desempenho de ambas com finalidade de compará-las, da mesma forma que no primeiro experimento, além de compará-las, nesse segundo experimento usou-se também o coeficiente de determinação  $R^2$  para medir a qualidade do modelo.

**Palavras-chaves:** Regressão polinomial, Redes neurais MLP, Função *humps*, Ajuste e Predição.

# Abstract

This paper presents a way to implement the support the decision to use artificial neural networks (RNA) with univariate and multivariate estatística. In the proposal, the RNA perceptron ltiplas layers of mu (MLP) and two experiments, at first, we used a returns the polynomial of degree 7, respon sible for adjustment and predictive C to a C function to especifica troubled with an error randomly river. Where the network (MLP) and returns the polinomila grade 7 showed good results. Al in addition, this study also aims to identify the performance of t used ecnicas compared the edition m squared error generated by both técnicas. For a second experiment, we created a function of the type  $y = \text{humps}(x)$  and disrupted with an  $e \sim N(0, 1)$  so that the MLP network and returns the polynomial that approximate the function of the c humps disturbed by evaluating the performance of both in order to compare them, in the same manner as in the first experiment.

**Key-words:** Regression polinomial, Redes neurais MLP, adjustment and prediction, Fit and predicting.

# Sumário

## Lista de Figuras

## Lista de Tabelas

<b>1</b>	<b>Introdução</b>	p. 11
<b>2</b>	<b>Fundamentação Teórica</b>	p. 13
2.1	Histórico . . . . .	p. 13
2.2	Paralelismo entre os modelos estatísticos e os neurais . . . . .	p. 14
2.2.1	Relação entre redes neurais e regressão . . . . .	p. 14
2.3	Principais características de uma rede neural . . . . .	p. 15
2.4	Neurônio artificial . . . . .	p. 15
2.5	Funções de ativação parcialmente diferenciáveis . . . . .	p. 17
2.5.1	Função degrau ou indicadora . . . . .	p. 17
2.5.2	Função degrau bipolar ou função sinal . . . . .	p. 17
2.5.3	Função de ativação rampa simétrica . . . . .	p. 18
2.6	Funções de ativação totalmente diferenciáveis . . . . .	p. 19
2.6.1	A função logística . . . . .	p. 19
2.6.2	A função de ativação hiperbólica . . . . .	p. 20
2.6.3	A função gaussiana . . . . .	p. 20
2.6.4	A função linear . . . . .	p. 21
2.7	Arquitetura das redes neurais . . . . .	p. 21
2.7.1	Treinamento da RNA: supervisionado e não supervisionado . . . . .	p. 22

2.7.2	Principais arquiteturas de redes neurais artificiais . . . . .	p. 23
2.8	Perceptron múltiplas camadas (MLP) . . . . .	p. 24
2.8.1	Processo de treinamento do <i>Perceptron</i> . . . . .	p. 24
2.9	Análise de regressão . . . . .	p. 25
2.9.1	Origem da Regressão . . . . .	p. 25
2.9.2	Regressão . . . . .	p. 26
2.9.3	Tipos de regressão . . . . .	p. 26
2.9.4	Regressão linear . . . . .	p. 26
2.9.5	. Estimação de Parâmetros . . . . .	p. 28
2.9.6	O coeficiente de correlação . . . . .	p. 28
2.9.7	Coefficiente de determinação . . . . .	p. 29
2.9.8	Regressão polinomial . . . . .	p. 29
2.9.8.1	Polinômios com uma regressora . . . . .	p. 30
<b>3</b>	<b>Aplicação</b>	p. 32
3.1	Regressão polinomial . . . . .	p. 34
3.2	Resultados e discussão . . . . .	p. 35
<b>4</b>	<b>Conclusão</b>	p. 41
<b>5</b>	<b>Referências Bibliográficas</b>	p. 42
	<b>Apêndice A – Apêndice</b>	p. 44

# Lista de Figuras

1	Esquema de unidade McCulloch-Pitts . . . . .	p. 16
2	Função de ativação degrau . . . . .	p. 17
3	Função degrau ou bipolar . . . . .	p. 18
4	Função de ativação rampa simétrica . . . . .	p. 18
5	Função de ativação logística . . . . .	p. 19
6	Comportamento do gráfico com a influência da contante $\beta$ na função de ativação logística . . . . .	p. 19
7	Função hiperbólica . . . . .	p. 20
8	Função de ativação gaussiana . . . . .	p. 21
9	Função de ativação linear . . . . .	p. 21
10	Rede <i>perceptron</i> . . . . .	p. 23
11	Exemplo de <i>Perceptron</i> Multicamadas . . . . .	p. 33
12	Gráfico da Função $x\cos(x) + \varepsilon$ . . . . .	p. 36
13	Gráfico de dispersão da função $x\cos(x)$ perturbada, regressão polinomial de grau 7, e o ajuste da rede neural MLP. . . . .	p. 36
14	Formas de onda do erro de previsão para a rede neural MLP (a) e formas de onda do erro da regressão polinomial (b) . . . . .	p. 37
15	Gráfico da função de Humps no intervalo de -0,5 a 1,5 . . . . .	p. 38
16	Gráfico de dispersão da função humps(x) perturbada, regressão polinomial de grau 19, e o ajuste da rede neural MLP . . . . .	p. 39
17	Valor ajustado pela rede versus o valor da variável independente com a Rede MLP (a) e Valor ajustado pela rede versus o valor da variável independente com a regressão polinomial (b) . . . . .	p. 39

18	Formas de onda do erro da regressão polinomial (a) e Formas de onda do erro da rede MLP (b) . . . . .	p.40
----	---	------

# Lista de Tabelas

1	Lista términos e expressões equivalentes de Redes Neurais e Estatística	p. 14
2	Técnicas equivalentes entre redes neurais e estatística . . . . .	p. 15
3	Erros de regressão e MLP . . . . .	p. 37
4	Coefficiente de determinação para a regressão polinomial e a rede perceptron múltiplas camadas (MLP). . . . .	p. 38



# 1 Introdução

Em sistemas inteligentes, as redes neurais artificiais (RNA) são modelos matemáticos que se assemelham às estruturas neurais biológicas e que tem capacidade computacional adquirida por meio de aprendizado e generalização. O aprendizado em RNA consiste na fase em que a rede neural absorve dados e, a partir destes, modifica seus parâmetros de entrada. Esta etapa pode ser considerada como uma adaptação da RNA às características intrínsecas de um problema, em que se procura cobrir um grande espectro de valores associados as variáveis pertinentes. Isto é feito para que a RNA adquira, através de uma melhora gradativa, uma boa capacidade de resposta para o maior número de situações possíveis (BRAGA, CARVALHO e LUDEMIR, 2000).

Por sua vez a generalização de uma RNA está associada à sua capacidade de dar respostas coerentes para dados não apresentados a ela durante o treinamento. Espera-se que uma RNA treinada tenha uma boa capacidade de generalização independentemente de ter sido controlada durante o treinamento. No entanto, atualmente, boa parte das pesquisas na área visa o desenvolvimento de modelos e técnicas de aprendizado que tenham algum controle de generalização (SILVA, 2010).

Para Braga, Carvalho e Ludemir (2000), bem como Kovács (2006), o processamento da informação em RNAs é feito por meio de estruturas neurais artificiais em que o armazenamento e o processamento da informação são realizados de maneira paralela e distribuída, por elementos processadores relativamente simples. Cada elemento processador corresponde a um neurônio artificial, também conhecido como modelo de McCulloch e Pitts, ou simplesmente modelo MCP. Uma das características mais importantes das RNAs é que as mesmas são aproximadores universais de funções multivariáveis contínuas. Em outras palavras, qualquer problema de aproximação de funções contínuas pode ser resolvido por meio de RNA, independente do número de variáveis envolvidas.

Pode-se dizer ainda que, a habilidade de manipular dados imprecisos faz com que as redes neurais sejam extremamente eficazes em tarefas em que especialistas não estão à

disposição ou um conjunto de regras não pode ser facilmente formulado. O campo de pesquisa em redes neurais é bastante amplo e inclui entre outras áreas, como estatística, robótica, controle, sistemas de produção, reconhecimento de padrões, processamento de imagens, física, geofísica, tomografia, engenharia, biomédica, análise financeira, sistemas de potência, gerenciamento e pesquisa operacional. Visto com isso a importância das RNAS no desenvolvimento das pesquisas.

Este trabalho tem por objetivo estudar e identificar o desempenho do ajuste realizado pela perceptron multicamadas (MLP) e compará-las com uma regressão polinomial de grau 7 e uma função polinomial de grau 19. Para ambos ajustes serão observados os valores de erros médios quadráticos. A função do tipo  $y = x \cos(x)$  perturbada com um erro aleatório  $e \sim N(0, 1)$  e por meio da função *humps*, também com um erro  $e \sim N(0, 1)$  de forma que a rede MLP e a regressão polinomial aproximem esta específica função perturbada, avaliando o desempenho de ambas com finalidade de compará-las, além disso.

## 2 Fundamentação Teórica

### 2.1 Histórico

As redes neurais foram desenvolvidas na década de 40, pelo neurofisiologista McCulloch e pelo matemático Walter Pitts da Universidade de Illinois, que fizeram uma analogia entre células nervosas vivas e o processo eletrônico num trabalho publicado sobre os neurônios formais. O trabalho consistia num modelo de resistores variáveis e amplificadores representando conexões sinápticas de um neurônio. Na mesma época, o psicólogo Donald Hebb descobriu a base de aprendizado nas redes neurais quando explicou o que ocorre, em nível celular, durante o processo de aprendizagem no cérebro.

Anos depois, em 1959, foi apresentada a rede Adaline e, em 1960, a rede Perceptron, por Frank Rosenblatt. A rede Adaline, abreviação de Adaptive Linear Network, pela primeira vez, imitava o cérebro humano. Em 1969, Marvin Minsky e Seymour Papert, publicaram um livro chamado *Perceptron*, no qual, resumiram e criticaram seriamente a pesquisa neste ramo da inteligência artificial, apresentando exemplos de limitação do uso dessas redes. Com isto, houve então uma retração na evolução de pesquisas sobre redes neurais, e só então em 1982, Rumelhart e McClelland, voltaram com novas pesquisa sobre o assunto.

Desde então, diversos modelos de redes neurais artificiais têm surgido com o propósito de aperfeiçoar e aplicar esta tecnologia. Algumas destas propostas tendem a aperfeiçoar mecanismos internos da rede neural para aplicação na indústria e negócios, outras procuram aproximá-las ainda mais dos modelos biológicos originais.

De acordo com Haykin (2001), pode-se definir uma RNA como sendo um processador maciço e paralelamente distribuído, constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental, e aplicá-lo para resolver problemas. As redes neurais artificiais são modelos computacionais inspirados no sistema nervoso de seres vivos, elas possuem a capacidade de aquisição e manutenção do

conhecimento (baseado em informações) e podem ser definidas como um conjunto de unidades de processamento, caracterizadas por neurônios artificiais, que são interligados por um grande número de interconexões (sinapses artificiais), sendo as mesmas representadas aqui por vetores/matrizes de pesos sinápticos.

## 2.2 Paralelismo entre os modelos estatísticos e os neurais

Sabe-se que a estatística compreende um conjunto de métodos que servem para buscar, organizar, resumir e analisar dados, assim como para extrair conclusões e tomar decisões. É por tanto natural esperar que os métodos utilizados nas redes neurais artificiais são de algum modo paralelos (ou similares) aos empregados na estatística moderna.

### 2.2.1 Relação entre redes neurais e regressão

A Tabela 1, mostras as expressões equivalentes, ou muito parecidas, de ambos os campos, o das redes neurais artificiais e a estatística, com isso é possível ver a relação que há entre esses dois campos de estudo.

Tabela 1: Lista términos e expressões equivalentes de Redes Neurais e Estatística

Redes Neurais	Estatística
Entrada	Variáveis independentes
Saída	Variáveis dependentes
Aprendizagem	Estimação
Pesos	Parâmetros
Generalização	Interpolação
Conjunto de treinamento	Amostra
Resíduos	Erros
Arquitetura	Modelo
Aproximação de função	Regressão

Muitas modelos de redes neurais são similares (em ocasiões iguais) a técnicas estatísticas, tais como modelos lineares generalizados, regressão polinomial, regressão não paramétrica, regressão projeção. A maior parte das redes neurais pode aprender a generalizar a partir de dados com ruídos (erros) similarmente aos métodos estatísticos.

É possível ver na Tabela 2, a equivalência entre algumas técnicas estatísticas e as redes neurais.

Tabela 2: Técnicas equivalentes entre redes neurais e estatística

Técnica de redes neurais	Técnicas Estatísticas
Rede de segunda ordem	Regressão quadrática
Redes de ordem superior	Regressão polinomial
Aprendizado supervisionado	Regressão
Propagação <i>feedforward</i>	Predição

## 2.3 Principais características de uma rede neural

Dentre as características de uma rede neural as principais são:

- i) Adaptação por experiência: as adaptações dos parâmetros internos da rede, tipicamente seus pesos sinápticos, são ajustadas a partir da apresentação sucessiva de exemplos o que possibilita a aquisição do conhecimento por experimentação;
- ii) Capacidade de aprendizado: por intermédio da aplicação de um método de treinamento, a rede consegue aprender o comportamento das variáveis armazenando assim as informações;
- iii) Habilidade de generalização: após o processo de treinamento da rede, essa é capaz de generalizar o conhecimento adquirido, pois o mesmo armazena os conhecimentos nos pesos das conexões, possibilitando com isso solucionar outros problemas que se encaixem nesse sistema de aprendizado.
- iv) Cada rede neural artificial representa uma arquitetura de processamento.

## 2.4 Neurônio artificial

Para uma melhor definição do que são redes neurais artificiais é importante mostrar como se constitui as suas unidades básicas. Assim, é possível ver na Figura 1, a descrição funcional do *k-ésimo* neurônio de uma rede, os diversos sinais de entrada advindos do meio externo (aplicação) que são espelhados pelo conjunto  $x_1, x_2, x_3, \dots, x_n$ . Com isso, a relevância de cada uma das entradas  $x_i$  do neurônio é então executada por meio de suas multiplicações pelos respectivos pesos sinápticos  $w_1$ , ponderando-se, portanto, todas as informações externas que chegam ao neurônio. Assim, torna-se possível verificar que a saída do corpo celular artificial, denotado por  $u$ , é a soma ponderada de suas entradas (BRAGA, 1998).

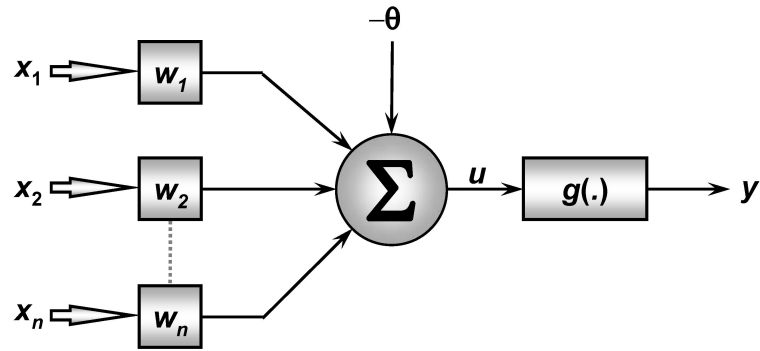


Figura 1: Esquema de unidade McCulloch-Pitts

Assim, considerando a Figura 1, verifica-se que o neurônio artificial é constituído de sete elementos básicos, ou seja:

- i) Sinais de entrada  $(x_1, x_2, \dots, x_n)$
- ii) Pesos sinápticos  $(w_1, w_2, \dots, w_n)$ , representam os pesos ou parâmetros da rede, ou ainda, funcionam como a memória de aprendizados anteriores.
- iii) Combinador linear  $(\Sigma)$ , que representa a combinação linear dos pesos, corresponde a soma ponderada da entrada pelos pesos.
- iv) Limiar de ativação  $(\theta)$ , indica o ponto em que a função se encontra em cima do eixo, e assim, definem o domínio de valores de saída.
- v) Potencial de ativação  $(u)$ , é o resultado produzido pela diferença do valor produzido entre o combinador linear e o limiar de ativação.
- vi) Função de ativação  $(g(\cdot))$ , tem objetivo é limitar a saída do neurônio dentro de um intervalo de valores razoáveis a serem assumidos pela sua própria imagem funcional.
- vii) Sinal de saída  $(y)$ , consiste do valor final produzido pelo neurônio em relação a um determinado conjunto de sinais de entrada, podendo ser também utilizado por outros neurônios que estão sequencialmente interligados.

Com isso, agora pode-se falar sobre as funções de ativação desses neurônios, que geralmente são utilizadas com dois propósitos: limitar a saída do neurônio e introduzir a não linearidade no modelo.

$$y = x_1w_1 + x_2w_2 + \dots + x_nw_n + \theta \quad (2.1)$$

## 2.5 Funções de ativação parcialmente diferenciáveis

As funções de ativação parcialmente diferenciáveis são aquelas que possuem pontos cujas derivadas de primeira ordem são inexistentes. As três principais funções desse grupo são: função degrau ou indicadora, função degrau bipolar ou função sinal e função identidade ou rampa (HAYKIN, 2001).

### 2.5.1 Função degrau ou indicadora

Nesta função os resultados obtidos assumirão valores unitários positivos quando o potencial de ativação do neurônio for maior ou igual a zero; caso contrário, o resultado assumirá valores nulos, ou seja, dado pela função:

$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u < 0 \end{cases} \quad (2.2)$$

A representação gráfica da função (2.2) é mostrada na Figura 2, em que é possível observar o seu comportamento:

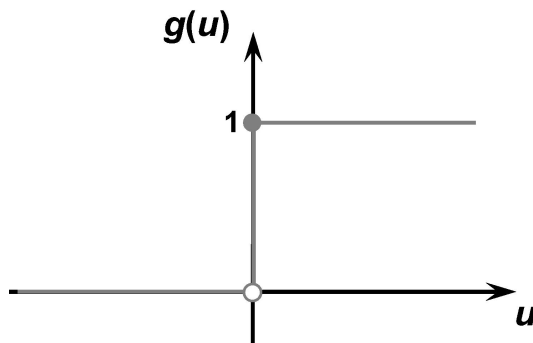


Figura 2: Função de ativação degrau

### 2.5.2 Função degrau bipolar ou função sinal

A ativação desta função assumirá valores unitários positivos quando o potencial de ativação do neurônio for maior que zero; valor nulo quando o potencial for também nulo; e valores unitários negativos quando o potencial for menor que zero. Segundo Haykin (2001), esta definição decreve a propriedade do "tudo-ou-nada" do modelo de RNA de

McCulloch e Pitts (1943). Seguindo a seguinte notação matemática:

$$g(u) = \begin{cases} 1, & \text{se } u > 0 \\ 0, & \text{se } u = 0 \\ -1, & \text{se } u < 0 \end{cases} \quad (2.3)$$

A função degrau ou sinal (2.3) é representada pela Figura 3, em que é visível que a função ocorre nos dois pólos do gráfico, ou seja, temos um pólo positivo e outro pólo negativo.

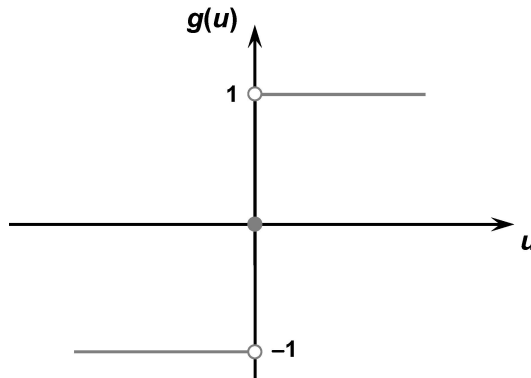


Figura 3: Função degrau ou bipolar

### 2.5.3 Função de ativação rampa simétrica

Os valores retornados são iguais aos próprios valores dos potenciais de ativação. Tendo como função matemática:

$$g(u) = \begin{cases} a, & \text{se } u \geq 0 \\ u, & \text{se } -a \leq u \leq a \\ -a, & \text{se } u < 0 \end{cases} \quad (2.4)$$

A representação gráfica da função (2.4) é vista na Figura 4:

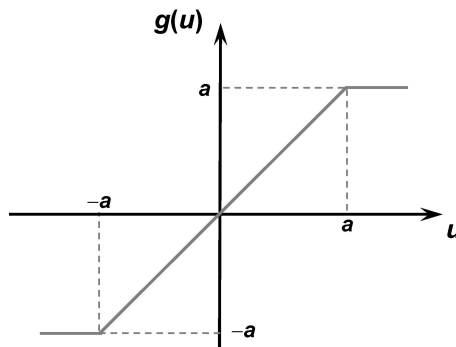


Figura 4: Função de ativação rampa simétrica



## 2.6 Funções de ativação totalmente diferenciáveis

As funções de ativação totalmente diferenciáveis são aquelas cujas derivadas de primeira ordem existe e são conhecidas em todos os pontos de seu domínio de definição. As quatro principais funções pertencentes a este grupo, e que podem ser empregadas em redes neurais artificiais são a função logística, a tangente hiperbólica, a gaussiana e a função linear (HAYKIN, 2001).

### 2.6.1 A função logística

Nesse tipo de função o resultado de saída produzido pela aplicação assumirá sempre valores reais entre zero e um.

$$g(u) = \frac{1}{1 + e^{-\beta u}} \quad (2.5)$$

em que,  $\beta$  é uma constante real, que está associada a inclinação da função, como pode ser observado na Figura 5

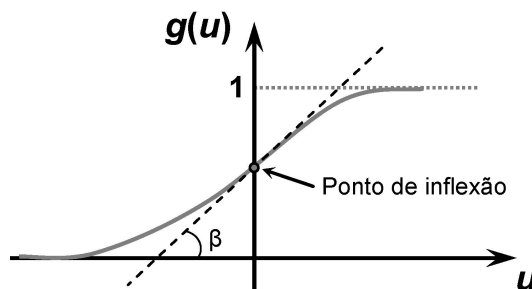


Figura 5: Função de ativação logística

A Figura 6 mostra o comportamento da função de ativação logística (2.5) em relação à variação do parâmetro de inclinação que é a constante real  $\beta$ .

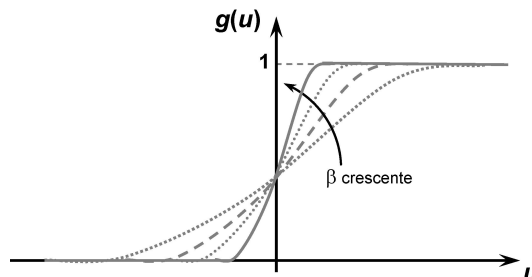


Figura 6: Comportamento do gráfico com a influência da contante  $\beta$  na função de ativação logística

Assim, é possível ver claramente através da Figura 6, que a medida que  $\beta$  tende

a infinito, o formato geométrico da função se assemelha com o formato da função de ativação degrau, no entanto, ao contrário da função degrau a função logística é totalmente derivável.

### 2.6.2 A função de ativação hiperbólica

Nesse tipo de função, os valores assumem valores positivos e negativos. E segue a seguinte expressão matemática, que será mostrada graficamente na Figura 7:

$$g(u) = \frac{1 - e^{-\beta u}}{1 + e^{-\beta u}} \quad (2.6)$$

Para a função de ativação hiperbólica (2.6), tem-se a seguinte Figura 7:

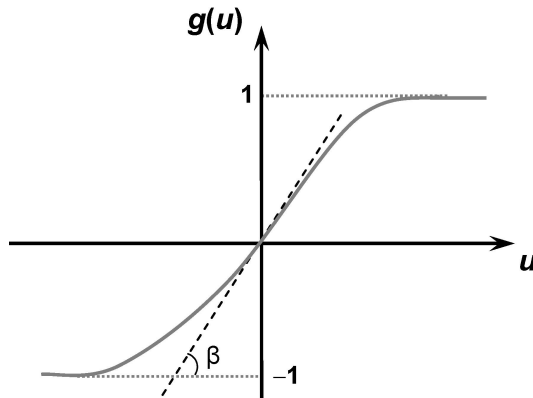


Figura 7: Função hiperbólica

### 2.6.3 A função gaussiana

Na função de ativação gaussiana, a saída do neurônio produzirá resultados iguais para aqueles valores de potencial de ativação ( $u$ ) que estejam posicionados a uma mesma distância média do centro, em que sua curva será simétrica em relação ao mesmo. A função gaussiana é dada por:

$$g(u) = e^{-\frac{(u-c)^2}{2\sigma^2}}, \quad (2.7)$$

tem-se que  $c$  é um parâmetro que define o centro da função gaussiana e  $\sigma$  será o desvio padrão associado à mesma, isto é, esse parâmetro mostrará o quão disperso está a curva em relação ao centro. A representação gráfica desta função (2.7) é ilustrada na Figura 8.

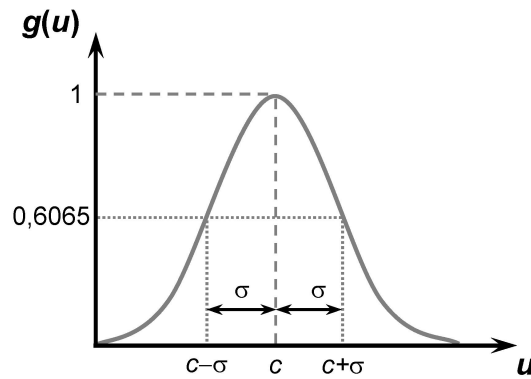


Figura 8: Função de ativação gaussiana

### 2.6.4 A função linear

Na função de ativação linear ou função identidade, os resultados de saída são sempre iguais aos valores atribuídos ao potencial de ativação ( $u$ ), e sua expressão matemática será:

$$g(u) = u \quad (2.8)$$

Essa função (2.8) é representada graficamente na Figura 9.

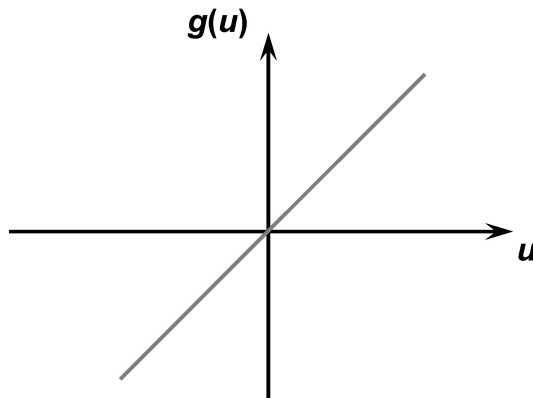


Figura 9: Função de ativação linear

## 2.7 Arquitetura das redes neurais

A arquitetura é um fator de grande importância, pois é a partir dela que definimos a capacidade de processamento de uma RNA. A escolha correta do número de conexões é decisiva para um treinamento bem sucedido. A arquitetura de uma rede neural é basicamente dividida em dois tipos:

- i) Redes não recorrentes: são as redes que não possuem realimentação, também conheci-

das como sem memórias. Sua estrutura pode ser com de camada única e também multicamada. Neste tipo de arquitetura não existem conexões ligando um neurônio de uma camada a outro neurônio de uma camada anterior nem a um neurônio de uma mesma camada. Para este temos como exemplo as redes do tipo *feedforward*.

- ii) Redes recorrentes: estas são redes que possuem realimentação, por isso, essas redes recorrentes respondem a estímulos dinamicamente, ou seja, após aplicar uma nova entrada, a saída é calculada e então realimentada para modificar a entrada. Por isso, essas redes são ditas com memória. A realimentação ocorre da saída de cada neurônio da camada oculta para todos os neurônios da mesma camada. Além dessa camada recorrente, a rede pode apresentar várias outras camadas do tipo MLP e pode também ter uma ou várias saídas. Neste tipo de rede o algoritmo utilizado para o treinamento é o de retropropagação do erro. Entretanto, é preciso tomar cuidado na implementação deste algoritmo uma vez que num dado instante de tempo a rede recebe não apenas as entradas externas, mas também as realimentadas da saída da primeira camada oculta obtidas no instante anterior.

### 2.7.1 Treinamento da RNA: supervisionado e não supervisionado

De acordo com Braga (2007), a etapa de treinamento, o objetivo do treinamento de uma RNA é fazer com que a aplicação de um conjunto de entradas produza um conjunto de saídas desejado ou no mínimo consistentes. Durante o processo de treinamento, os pesos da rede gradualmente convergem para determinados valores, de modo que aplicação dos vetores de entrada produza as saídas necessárias. O aprendizado relaciona-se a maneira pela qual a modificação dos parâmetros ocorre. Neste processo os parâmetros são ajustados através de um processo estimulado pelo ambiente no qual a rede está inserida. O algoritmo de aprendizagem refere-se a regras bem definidas para a solução de um problema de aprendizado. Os algoritmos diferem entre si pela forma como se dá o ajuste dos pesos. Os procedimentos de treinamento e aprendizado podem ser classificados em dois tipos:

- i) Supervisionado - Acontece da seguinte forma: existe um vetor de entrada e um vetor alvo que se deseja na saída. O processo funciona aplicando-se o vetor de entrada e, então, a saída fornecida pela rede e comparada com o vetor de resposta desejado. Dessa comparação obtém-se um sinal de erro. O erro encontrado é realimentado através da rede e os pesos são atualizados segundo um algoritmo determinado a fim

de minimizar este erro. O algoritmo de aprendizado supervisionado mais utilizado é o *Backpropagation*. Portanto, o aprendizado supervisionado pressupõe um erro de saída. É importante que os valores alvo (resposta desejada) sejam escolhidos dentro do intervalo da função de ativação.

- ii) Não supervisionado - Nesse treinamento não existe um vetor de resposta desejada, logo, não existem comparações que forneçam um sinal de erro. Nesta situação, são fornecidas a rede condições para realizar uma medida independente da tarefa que deve ser aprendida e os parâmetros livres da rede são otimizados em relação a esta medida.

## 2.7.2 Principais arquiteturas de redes neurais artificiais

A Figura 10 ilustra uma das principais arquiteturas de redes neurais, a Perceptron que é constituída de  $n$  sinais de entrada, e somente uma saída, pois a mesma é composta de um único neurônio.

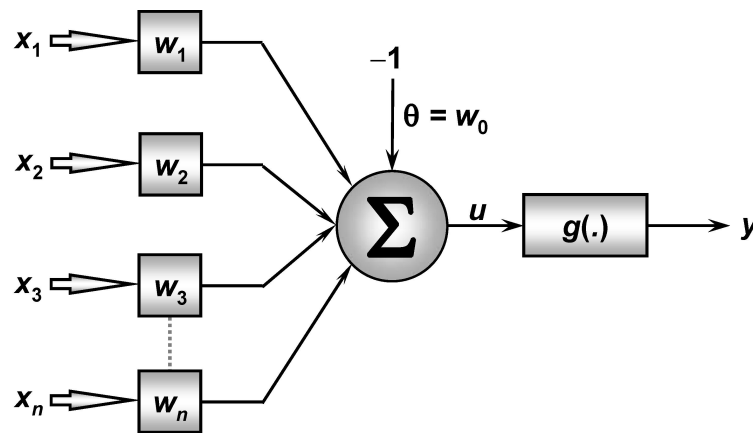


Figura 10: Rede *perceptron*

Esse tipo de rede neural que não tem camada oculta só consegue classificar padrões que sejam linearmente separáveis (ou seja, padrões que se encontram em lados opostos de um hiperplano). Esse tipo de rede tem como principais características (HAYKIN, 2001):

- i) Sua função de ativação será sempre degrau.
- ii) Sua arquitetura é composta de apenas uma camada.
- iii) Seu algoritmo de aprendizado é supervisionado.
- iv) Seus valores entrada/saída são os números binários (-1, +1).

## 2.8 Perceptron múltiplas camadas (MLP)

A MLP é uma rede do tipo unidirecional que inclui várias camadas escondidas entre as da entrada e saída, permitindo estabelecer regiões de decisão mais complexas que as do perceptron de uma só camada (FAUSSET, 1994). Esse tipo de rede é constituído da seguinte forma: uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. O sinal se propaga sempre para a frente, camada por camada. Esse tipo de rede é capaz de construir uma superfície no espaço de entrada. Uma rede MLP com uma camada escondida e neurônios com função de ativação sigmoideal (com um número adequado de neurônios na camada escondida) é um mapeador universal, isto é, pode aproximar com uma precisão arbitrária qualquer região de decisão contínua.

O treinamento de uma rede MLP é uma aprendizagem com correção do erro, ou seja, aprendizagem supervisionada (MICHALEWICZ, 2000). Para adaptar os pesos, calcula-se o erro no  $i$ -ésimo neurônio usando-se um erro derivado da camada mais próxima da saída. Este erro é o erro de saída retropropagado e devidamente escalado. A sensibilidade é automaticamente calculada pela regra da cadeia (BENDER, 1996).

O uso do algoritmo de retropropagação (*backpropagation*) para o treinamento de redes MLP é sistemático. Esse algoritmo consiste em calcular o erro na saída da rede e retropropagá-lo pela rede, modificando os pesos para minimizar o erro da próxima saída. O algoritmo baseia-se no método do gradiente descendente cuja idéia central é fazer modificações proporcionais ao gradiente do erro. A direção do gradiente é onde o erro é minimizado. Um ponto interessante a ser observado é que os erros de classificação e o erro médio quadrático podem variar em sentidos opostos. O erro médio quadrático é sensível à diferença entre a resposta desejada e a atual, enquanto que o número de classificações erradas é uma quantidade digital que depende somente da maior saída (BISHOP, 1995).

### 2.8.1 Processo de treinamento do *Perceptron*

O ajuste dos pesos e limiar do Perceptron visando-se propósitos de classificação de padrões que podem pertencer a uma das duas únicas classes possíveis, é realizado por meio da regra de aprendizado de Hebb (HEBB, 1949). Resumidamente, se a saída produzida pelo Perceptron está coincidente com a saída desejada, os pesos sinápticos e limiares da rede serão então incrementados (ajuste excitatório) proporcionalmente aos valores de seus sinais de entrada; caso contrário, ou seja, a saída produzida pela rede é diferente do

valor desejado, os pesos sinápticos e limiar serão então decrementados (inibitório). Este processo é repetido, sequencialmente para todas as amostras de treinamento, até que a saída produzida pelo Perceptron seja similar à saída desejada de cada amostra. Uma das grandes vantagens dessa técnica é que o perceptron multicamada é um aproximador universal de funções. Isso significa que desde que os pesos sejam bem adaptados e a rede dispõe um número suficiente de neurônios escondidos, o cálculo desejado é atingido.

## 2.9 Análise de regressão

A análise de regressão é uma técnica estatística para investigar e modelar a relação entre variáveis, sendo uma das mais importantes na análise de dados (FREEDMAN et al., 2004).

### 2.9.1 Origem da Regressão

O termo regressão teve origem com Francis Galton ao verificar que, mesmo havendo uma tendência de pais altos terem filhos altos e de pais baixos terem filhos baixos, a altura média de filhos de pais de uma dada altura tendia a se deslocar, ou a regredir (daí ter começado a fazer uso deste termo) para a altura média da população como um todo. A lei de regressão universal de Galton foi confirmada mais tarde por Karl Pearson, tendo este coletado mais de mil registros das alturas dos membros de grupos de famílias, verificando que a altura média dos filhos de um grupo de pais altos era inferior à altura de seus pais e que a altura média dos filhos de um grupo de pais baixos era superior à altura de seus pais. Assim, tanto os filhos altos como os baixos "regrediram" em direção à altura média da população.

Na análise de regressão, busca-se estabelecer a existência e o grau de dependência estatística entre as variáveis aleatórias, ou seja, aquelas que têm distribuição de probabilidade. A análise de regressão tem como objetivo estimar o valor médio de uma variável, a variável dependente, com base nos valores de uma ou mais variáveis adicionais, variáveis explicativas. Neste tipo de análise, a variável dependente é estocástico, enquanto as variáveis explicativas não são estocásticas para a maior parte. A análise de regressão ganhou popularidade devido ao grande número de pacotes estatísticos que incluem e por ser um "processo robusto que se adapta a uma infinidade de aplicações científicas e executivo para a tomada de decisões."

## 2.9.2 Regressão

A análise de regressão, bem como a correlação tem a ver com a análise simultânea de duas ou mais variáveis, como quando mede-se vários atributos em um mesmo indivíduo (idade, peso, altura, etc.) e há interesse em saber qual é a relação entre eles. No caso da correlação estima-se o grau em que duas variáveis covariam ou variam juntas, isto é, em sua interdependência. Em geral, usa-se a análise de regressão para testar a hipótese sobre a possível relação entre duas variáveis  $Y$  e  $X$  casualidade para prever  $Y$  em termos de  $X$ , e para explicar as mudanças em  $Y$  em relação às mudanças em  $X$ . Em geral, através desta análise explica-se as mudanças na variável dependente em relação a uma combinação linear da variável independente.

Um dos pressupostos principal desta análise é a existência de uma relação de causalidade entre variáveis. Assim, toda vez que faz-se uma análise de regressão, assumi-se que a variável independente  $X$ , está relacionada com a variável  $Y$ . Assim descreve-se a relação funcional entre duas variáveis. O objetivo da análise de regressão será a predição dos valores da variável dependente para valores não observado das variáveis independentes (MONTGOMERY, 2001).

## 2.9.3 Tipos de regressão

i) Considerando o número de variáveis independentes consideradas na análise:

- Uma variável independente, a análise é denominado análise de regressão simples.
- Diversas variáveis independentes, a análise é chamado de Análise de Regressão Múltipla.

ii) Por tipo de relação funcional entre as variáveis:

- Linear: Equação da reta ou do plano.
- Não-Lineares: Equação exponencial, polinomial, geométrica, etc.

## 2.9.4 Regressão linear

Ainda segundo Freedman (2004), importância da distribuição bidimensional consiste em investigar como uma variável afeta a outra. Esta pode ser uma causa e efeito sub-



sidiária, por exemplo, a quantidade de chuva (causa) leva a um aumento na produção agrícola (efeito). Ou o aumento do preço das mercadorias, resultando em uma diminuição das quantidades exigidas.

De acordo com Bussab (1988), Se utilizarmos um sistema de coordenadas cartesianas para representar a distribuição bidimensional, obtemos um conjunto de pontos conhecido como o diagrama de dispersão, que a análise qualitativa para estudar a relação entre duas variáveis. O próximo passo é a determinação das funções subsidiárias entre as duas variáveis X e Y que melhor se ajusta à distribuição bidimensional. É chamado de regressão linear quando a função é linear, ou seja, requer a determinação dos parâmetros: a inclinação e do intercepto da reta de regressão.

$$y = ax + b. \quad (2.9)$$

A regressão também nos permite determinar o grau de dependência da série de valores X e Y, a previsão do valor e estima-se que seria obtida por um valor x que não está na distribuição. Para determinar a equação da linha que melhor se ajusta aos dados, temos o chamado erro  $e_i$  diferença  $y_i - y$ , entre as  $y_i$  valor observado e o valor ajustado  $y = ax_i + b$ . A abordagem do ajuste é tomado como aquele em que a variação é mínima, quer dizer, deve ser soma mínima.

$$s = \sum_{i=1}^N e_i^2 = \sum_{i=1}^N (y_i - (ax_i + b))^2 \quad (2.10)$$

O problema da regressão linear simples entre X e Y variáveis é reduzida para o cálculo da linha de regressão que melhor representa a sua distribuição conjunta. Os dados são apresentados como uma matriz de duas colunas:

$$\begin{pmatrix} (x_1, y_1), \\ (x_2, y_2) \\ (x_3, y_3) \\ \dots \\ (x_n, y_n), \end{pmatrix}$$

em que  $(x_i, y_i)$ , com  $i = 1, 2, \dots, n$ , é o *i-ésimo* par observados. Pretende-se ajustar um modelo de como

$$y = ax_i + b + e_i \quad (2.11)$$

Sob as seguintes hipóteses:

- i) A variável resposta  $y_i$ , depende do  $x_i$  variável explicativa linear (com um declive e interceptar  $b$ ), acrescido de um factor aleatório  $e_i$  residual.
- ii) Os resíduos têm distribuição normal com média 0 e variância  $\sigma^2$  desconhecida.
- iii) Estes factores aleatórios são independentes.

### 2.9.5 . Estimação de Parâmetros

Os parâmetros da linha de regressão  $a$  e  $b$ , são calculados pelo critério dos mínimos quadrados, o que leva aos

$$\frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad b = \bar{y} - a\bar{x} \quad (2.12)$$

Sendo

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

ambas as medidas de variáveis estatísticas.

### 2.9.6 O coeficiente de correlação

A definição, o coeficiente de correlação como

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (2.13)$$

que tem apenas os valores no intervalo  $[-1,1]$ , nos dá uma ideia de quão longe o ajuste linear é razoável:

- i) Se o valor de  $r$  está próximo de  $-1$ , o ajuste é razoavelmente bom, observações, distribuídas  $(x_i, y_i)$  em torno de uma linha de inclinação negativa.
- ii) Se o valor de  $r$  é próximo de  $0$ , o ajuste não é aceitável, indicando que não há relação linear entre as variáveis.
- iii) Se o valor de  $r$  está próximo de  $+1$ , o ajuste é aceitavelmente bom, observações, distribuídas  $(x_i, y_i)$  em torno de uma linha de inclinação positiva.

### 2.9.7 Coeficiente de determinação

De acordo com Bussab (1988), o coeficiente de determinação, também chamado de  $R^2$  é uma medida de qualidade do modelo em relação a sua habilidade de estimar corretamente os valores da variável resposta  $Y$ . O  $R^2$  indica quanto a variável resposta é explicada pela variância das variáveis explicativas. Seu valor está no intervalo de 0 a 1, ou seja, quanto maior o valor do  $R^2$  mais explicativo é o modelo.

$$R^2 = SQ_{reg}/SQ_{total} \quad (2.14)$$

com  $0 \leq R^2 \leq 1$ . Em que  $SQ_{reg}$  é a soma de quadrado da regressão e  $SQ_{total}$  é a soma de todas as variáveis. O coeficiente de determinação  $R^2$  nos fornece a porcentagem da variação total de  $Y$  explicada pela variável  $X$ . Esse coeficiente pode ser obtido também pelo coeficiente de correlação de Pearson ( $r$ ), ou seja:

$$R^2 = (r)^2 \quad (2.15)$$

A estatística  $R^2$  deve ser usada com cuidado, pois o  $R^2$  grande não implica necessariamente que o modelo de regressão seja um bom preditor linear, pois podem haver modelos não lineares que representem melhor os dados.

### 2.9.8 Regressão polinomial

Segundo Montgomery e Peck (2001), o modelo de regressão polinomial (MRP) é um caso especial do modelo de regressão linear geral  $Y = X\beta + \varepsilon$ . Este modelo pode conter uma, duas ou mais regressoras (variáveis independentes) as quais podem estar em diversas potências. As regressoras são expressas comumente como o desvio de suas respectivas médias porque nos modelos polinomiais tem-se frequentemente termos altamente correlacionados. Expressar as regressoras como o desvio de sua média pode, em muitos casos, auxiliar na redução da multicolinearidade. Em outros casos, mesmo centrando as regressoras na média, pode-se continuar tendo níveis altos de multicolinearidade. Os polinômios ortogonais podem ser úteis nessas situações.

### 2.9.8.1 Polinômios com uma regressora

Suponha o modelo polinomial  $Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2^2 + \dots + \beta_k X_{1i}^k + \varepsilon_i$  para  $i = 1, \dots, n$ . Para tais modelos, as colunas da matriz  $X$  serão, geralmente, não ortogonais e, além disso, se aumentarmos a ordem do polinômio por adicionar um termo  $\beta_{k+1} X_{1i}^{k+1}$ , devemos recalculer a inversa  $(X'X)^{-1}$  e as estimativas dos parâmetros de ordem menor deverão mudar. Se, no entanto, for ajustado o modelo:

$$Y_i = \alpha_0 P_0(x_i) + \alpha_1 P_1(x_i) + \alpha_2 P_2(x_i) + \dots + \alpha_k P_k(x_i) + \varepsilon_i, \quad i = 1, \dots, n$$

em que  $P_u(x_i)$  é a  $u$ -ésima ordem do polinômio ortogonal definido de tal modo que,

$$\begin{cases} \sum_{i=1}^n P_r(x_i) P_s(x_i) = 0, & r \neq s \quad (r, s = 0, 1, 2, \dots, k) \\ P_0(x_i) = 1, \end{cases}$$

o modelo torna-se, então,  $Y = X\alpha + \varepsilon$  em que a matriz  $X$  é:

$$X = \begin{bmatrix} P_0(x_1) & P_1(x_1) & \dots & P_k(x_1) \\ P_0(x_2) & P_1(x_2) & \dots & P_k(x_2) \\ \vdots & \vdots & \vdots & \vdots \\ P_0(x_n) & P_1(x_n) & \dots & P_k(x_n) \end{bmatrix}$$

a qual tem colunas ortogonais e, portanto,

$$X'X = \begin{bmatrix} \sum_{i=1}^n P_0^2(x_i) & 0 & \dots & 0 \\ 0 & \sum_{i=1}^n P_1^2(x_i) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \sum_{i=1}^n P_k^2(x_i) \end{bmatrix}$$

Os cinco primeiros polinomiais  $P_j(x_i)$  para o caso dos níveis de  $X$  serem igualmente espaçados são dados por:

$$\begin{aligned} P_0(x_i) &= 1 \\ P_1(x_i) &= \lambda_1 \left[ \frac{x_i - \bar{x}}{d} \right] \\ P_2(x_i) &= \lambda_2 \left[ \left( \frac{x_i - \bar{x}}{d} \right)^2 - \left( \frac{n^2 - 1}{12} \right) \right] \\ P_3(x_i) &= \lambda_3 \left[ \left( \frac{x_i - \bar{x}}{d} \right)^3 - \left( \frac{x_i - \bar{x}}{d} \right) \left( \frac{3n^2 - 7}{20} \right) \right] \\ P_4(x_i) &= \lambda_4 \left[ \left( \frac{x_i - \bar{x}}{d} \right)^4 - \left( \frac{x_i - \bar{x}}{d} \right)^2 \left( \frac{3n^2 - 13}{14} \right) + \frac{3(n^2 - 1)(n^2 - 9)}{560} \right], \end{aligned}$$

sendo  $d$  = espaço (distância) entre os níveis de  $X$  e  $\lambda_j$  = constantes escolhidas de modo

aos polinomiais terem valores inteiros. Valores numericos desses polinomiais podem ser encontrados, por exemplo, em Montgomery e Peck (2001). Os estimadores de M.QO de  $\alpha$  são encontrados por:

$$\hat{\alpha} = (X'X)^{-1}X'Y \text{ e, portanto, } \hat{\alpha}_j = \frac{\sum_{i=1}^n P_j(x_i)y_i}{\sum_{i=1}^n P_j^2(x_i)} \text{ para } j = 0, 1, \dots, k.$$

Como  $P_0(x_i) = 1$ , segue que  $\hat{\alpha}_0 = \bar{y}$ . Ainda, temos que

$$SQRes(k) = Y'Y - \sum_{j=1}^k \hat{\alpha}_j [P_j(x_i)y_i]$$

e que a soma de quadrados da regressão para qualquer parâmetro do modelo, a qual não depende dos outros parâmetros no modelo, é dada por:

$$SQReg(\alpha_j) = \hat{\alpha}_j \sum_{i=1}^k P_j(x_i)y_i$$

Se precisarmos acessar a significância do termo de ordem mais alta, devemos testar  $H_0 : \alpha_k = 0$  e, para isso, usamos a estatística de teste F dada por:

$$F_0 : \frac{SQreg(\alpha_k)}{SQres(k)/(n - k - 1)}.$$

Note que se a ordem do polinômio é mudada para  $k + r$ , somente os  $r$  novos coeficientes devem ser calculados uma vez que os  $k$  coeficientes que já estavam no modelo não mudam devido a propriedade de ortogonalidade dos polinomiais. Assim, um ajuste sequencial do modelo é computacionalmente fácil. Polinomiais ortogonais podem também ser construídos e usados nos casos em que os  $X$ 's não são igualmente espaçados.

### 3 Aplicação

Para desenvolvimento do presente trabalho, usou-se a regressão polinomial, para ajustar o modelo e comparar os dados dos dois experimentos, assim foi possível comparar os ajustes da regressão polinomial e o ajuste da rede (MLP), e verificar qual dos dois melhor se adequou aos exemplos demonstrados.

Assim, gerou-se duas situações: para o primeiro experimento utilizou-se um conjunto de dados fictícios para o treinamento da rede neural e criou-se uma função do tipo  $y = x \cos(x)$  perturbada com um erro  $e \sim N(0, 1)$  no um intervalo de -6 a 4, e para o segundo experimento, foi suposto que a função que queremos inferir é uma função do tipo  $y = humps(x)$  perturbada com um erro  $e \sim N(0, 1)$ , em que a função *humps* é definida pela seguinte expressão matemática:

$$humps(x) = \frac{1}{(x - 0,3)^2 + 0,01} + \frac{1}{(x - 0,9)^2 + 0,04} - 6 + 0,9rand(300, 1),$$

em que,  $x \in [-0,5 : 1,5]$ .

Nos dois casos, avaliou-se se a rede MLP e uma regressão polinomial se aproximem a função  $y = x \cos(x)$  perturbada e da função *humps* perturbada.

Antes de dar início aos experimentos que darão suporte a esse trabalho, foi feito um exemplo, para um melhor entendimento de como funciona uma rede neural artificial tipo perceptron multicamada, utilizando um algoritmo backpropagation, há necessidade de se definir a priori diversas variáveis e parâmetros auxiliares que serão usados para tal propósito, e assim possibilitar um melhor desenvolvimento do mesmo. Baseando-se nisso, apresenta-se um conjunto de variáveis que norteiam a derivação do algoritmo.

De acordo com esse exemplo, é possível ver, como funciona o comportamento de uma rede perceptron multicamadas, considerando (Figura 11), composto de duas entradas ( $x_0 = -1$ ), ( $x_1 = 1$ ) e ( $x_2 = 2$ ), três neurônios na primeira camada escondida, dois neurônios na segunda camada escondida e um neurônio de saída, assim, será possível entender melhor o desempenho dos exemplos posteriores. Considera-se também que a

tangente hiperbólica será assumida como função de ativação para todos os neurônios, veja:

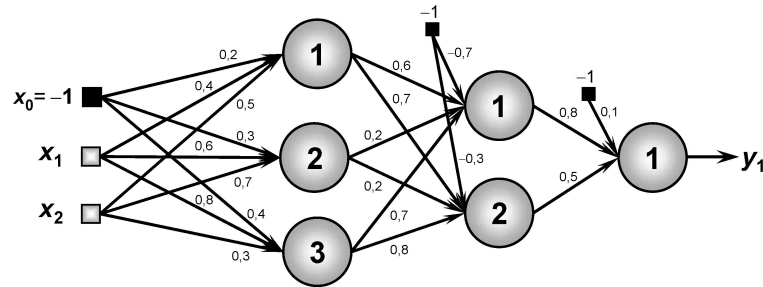


Figura 11: Exemplo de *Perceptron* Multicamadas

Cálculos:

$$W_{ji}^{(1)} = \begin{bmatrix} 0,2 & 0,4 & 0,5 \\ 0,3 & 0,6 & 0,7 \\ 0,4 & 0,8 & 0,3 \end{bmatrix}$$

$$W_{ji}^{(2)} = \begin{bmatrix} -0,7 & 0,6 & 0,2 & 0,7 \\ -0,3 & 0,7 & 0,2 & 0,8 \end{bmatrix}$$

$$W_{ji}^{(3)} = \begin{bmatrix} 0,1 & 0,8 & 0,5 \end{bmatrix}$$

Assumindo-se um sinal de entrada definido por  $(x_0 = -1)$ ,  $(x_1 = 1)$  e  $(x_2 = 2)$ , os vetores  $I_j^{(1)}$  e  $Y_j^{(1)}$  seriam então representados por:

$$I_j^{(1)} = \begin{bmatrix} 0,2(-1) + 0,4(1) + 0,5(2) \\ 0,3(-1) + 0,6(1) + 0,7(2) \\ 0,4(-1) + 0,8(1) + 0,3(2) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,7 \\ 1 \end{bmatrix}$$

$$Y_j^{(1)} = \begin{bmatrix} \tanh(1,2) \\ \tanh(1,7) \\ \tanh(1) \end{bmatrix} = \begin{bmatrix} 0,83 \\ 0,93 \\ 0,76 \end{bmatrix} \begin{bmatrix} -1 \\ 0,83 \\ 0,93 \\ 0,76 \end{bmatrix}$$

em que, os argumentos usados na função tangente hiperbólica ( $\tanh$ ) estão em radianos.

Os vetores  $I_j^{(2)}$  e  $Y_j^{(2)}$  referentes a segunda camada neural seriam representados por:

$$I_j^{(2)} = \begin{bmatrix} -0,7(-1) + 0,6(0,83) + 0,2(0,93) + 0,7(0,76) \\ -0,3(-1) + 0,7(0,83) + 0,2(0,93) + 0,8(0,76) \end{bmatrix} = \begin{bmatrix} 1,916 \\ 1,675 \end{bmatrix}$$

$$Y_j^{(2)} = \begin{bmatrix} \tanh(1,916) \\ \tanh(1,675) \end{bmatrix} = \begin{bmatrix} 0,96 \\ 0,93 \end{bmatrix} \begin{bmatrix} -1 \\ 0,96 \\ 0,93 \end{bmatrix}$$

Finalmente, os vetores  $I_j^{(3)}$  e  $Y_j^{(3)}$  referentes a terceira camada neural seriam representados por:

$$I_j^{(3)} = [0,1(-1) + 0,8(0,96) + 0,5(0,93)] = [1,13]$$

$$Y_j^{(3)} = [\tanh(1,13)] = [0,81]$$

Na última expressão dispensa-se a inserção do termo  $Y_0^{(3)} = -1$ , pois já se trata da última camada neural, sendo que o valor de  $Y_1^{(3)}$  é a própria saída de  $Y_1$ , produzida por esta rede.

### 3.1 Regressão polinomial

A escolha de uma função a ser ajustada aos dados deve levar em consideração as características do fenômeno em estudo, há funções especiais para determinados problemas. Quando não há uma função especial costuma-se utilizar as funções polinômios como uma aproximação ao comportamento dos dados. Para esse estudo usou-se a suposição de que, em um intervalo de variação dos dados, o comportamento da função será para o primeiro experimento, uma regressão polinomial de grau 7, e para o segundo experimento uma regressão polinomial de grau 19, os graus dos polinômios foram ajustados após vários testes, em que buscava-se o grau que ajustaria melhor os dados. Procurou-se adaptar um polinômio de acordo com o desenvolvimento da rede (MLP) a qual será comparada. Para ajustar polinômios de grau superior podemos deduzir fórmulas. No entanto, estas serão bem mais complicadas, difíceis de utilizar.

Para o primeiro experimento, em que usou-se uma função do tipo  $y = x \cos(x)$  per-



turbada com um erro  $e \sim N(0, 1)$  no um intervalo de -6 a 4, utilizou-se uma regressão polinomial de grau 7:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2^2 + \beta_3 X_3^3 + \beta_4 X_4^4 + \beta_5 X_5^5 + \beta_7 X_7^7 + \varepsilon$$

e para o segundo experimento, em que usou-se uma função do tipo  $y = humps(x)$  perturbada com um erro  $e \sim N(0, 1)$ , utilizou-se uma regressão polinomial de grau 19:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2^2 + \dots + \beta_{19} X_{19}^{19} + \varepsilon$$

A partir desses modelos, obtivemos os resultados que se seguem.

## 3.2 Resultados e discussão

Para o primeiro experimento, em que usou-se a função do tipo  $y = x \cos(x)$  perturbada com um erro  $e \sim N(0, 1)$  no um intervalo de -6 a 4, foi desenvolvido um algoritmo usando o software Matlab 7.1, com o propósito de ajustar e prever o desempenho da rede, investigando-se duas estruturas diferentes para essa simulação:

- O perceptron de múltipla camada consiste de um nó de entrada, uma primeira camada oculta com 4 neurônios, uma segunda camada oculta, com um neurônio linear de saída. O perceptron foi treinado com gradiente scale. O treinamento foi realizado com uma amostra de tamanho 100. O coeficiente da taxa de aprendizagem da rede foi de 0,01.
- Para cada ponto dado no conjunto de treinamento foi processado cerca de 100 vezes. A partir da função objetivo principal desse trabalho, temos a função do tipo  $y = x \cos(x)$  perturbada, observada na Figura 12 .

É possível observar claramente no comportamento desse gráfico, o que vem a facilitar as comparações entre as redes neurais artificiais e a regressão polinomial a qual esta sendo comparada no decorrer desse estudo.

Para esse modelo foi feita um ajuste usando a regressão polinomial:

$$y = -0.0003x^7 + 0.0004x^6 + 0.0201x^5 - 0.0079x^4 - 0.3408x^3 + 0.0282x^2 + 0.7051x + 0.0239$$

A aproximação da função empregando uma MLP também está discriminada na Figura 13. Para o ajuste e predição a arquitetura da rede neural MLP foi composta por: 4

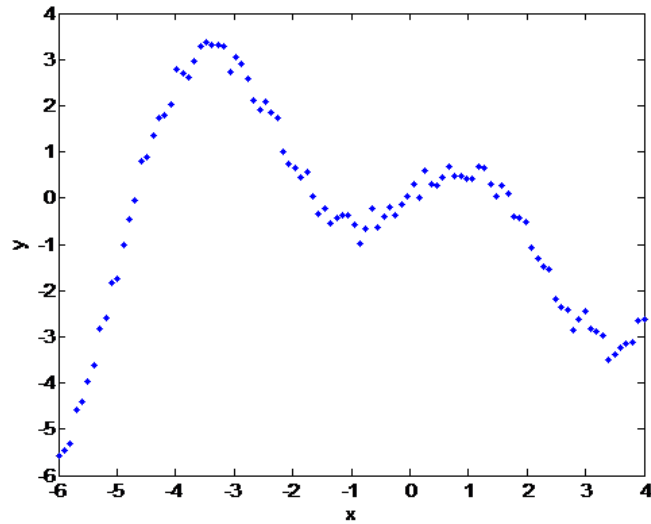


Figura 12: Gráfico da Função  $xcos(x) + \epsilon$

camadas ocultas, uma camada de entrada e uma de saída.

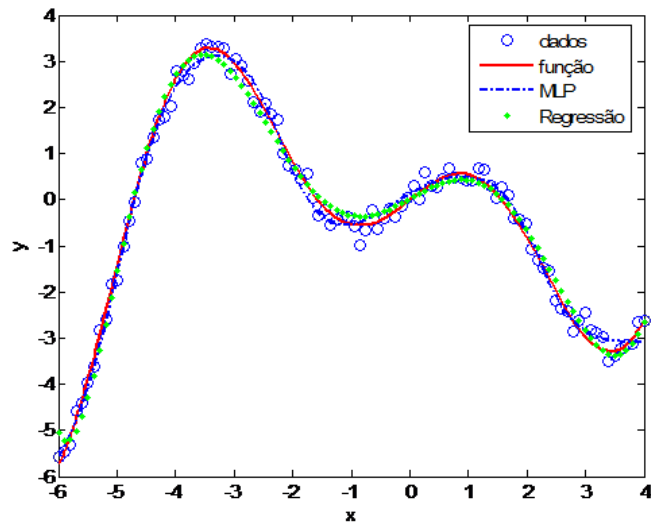


Figura 13: Gráfico de dispersão da função  $xcos(x)$  perturbada, regressão polinomial de grau 7, e o ajuste da rede neural MLP.

Após o treinamento da rede e comparação com a regressão polinomial de grau 7, obtivemos a Tabela 3:

Em que pode-se ver que, o erro da regressão polinomial e o erro da MLP, tanto no erro quadrado médio como no desvio padrão foram iguais.

A Figura 14, apresenta a forma de onda da predição de  $\hat{y}$  computado pela MLP. Esta figura inclui também a forma de onda real  $y$ . Estas duas formas de onda são dificilmente

Tabela 3: Erros de regressão e MLP

	Erro da regressão	Erro do modelo neural MLP
Erro quadrado médio	0,6540	0,6540
Desvio-padrão residual	0,0809	0,0809

distinguíveis entre si. O treinamento foi monitorado todo o tempo, a Figura 16 mostra os erros quadráticos médio da rede MLP. Observa-se que os valores estão muito próximos de zero, indicando também um ótimo ajuste gerado pela aproximação de função MLP.

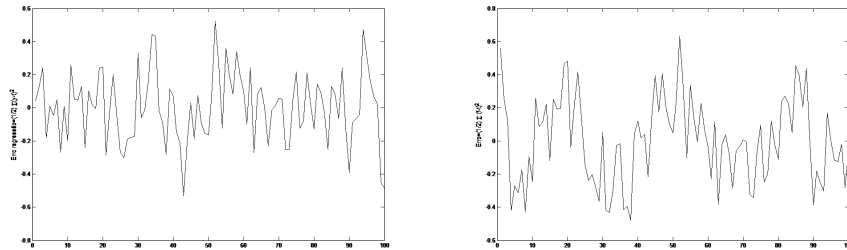


Figura 14: Formas de onda do erro de previsão para a rede neural MLP (a) e formas de onda do erro da regressão polinomial (b)

De forma análoga a Figura 14 apresenta a forma de onda da previsão  $\hat{y}$  computada pela MLP treinado com o algoritmo (ver apêndice A). Estas duas formas de onda têm o mesmo erro de predição.

Já no segundo experimento, foi feito um ajuste com a função *humps*, então para esse desenvolvimento, criou-se uma função do tipo  $y = humps(x)$  perturbada com um erro  $e \sim N(0, 1)$  de forma que a rede MLP e uma regressão polinomial que aproximem a função de *humps* perturbada, avaliando do desempenho de ambas com finalidade de compará-las.

Algumas observações a respeito da aplicação da rede MLP são aqui ressaltadas:

- Consiste de um nó de entrada, uma primeira camada oculta com 17 neurônios, uma segunda camada oculta, com um neurônio linear de saída. A função de otimização utilizada foi gradiente conjugado escalado. Utilizou-se a função de ativação linear para todos os neurônios, com taxa de aprendizagem de 0,01. O treinamento foi realizado com uma amostra de tamanho 300.
- Para cada ponto dado no conjunto de treinamento foi processado cerca de 100 vezes.

Todo o modelo desse trabalho foi realizado com uma regressão polinomial de grau 19, com um modelo do tipo:

$$Y = \beta_0 + \beta_1x + \beta_2x^2 + \beta_3x^3 + \dots + \beta_{19}x^{19} + \varepsilon$$

A partir da função objetivo desse segundo experimento, a função de Humps, em que  $y = \text{humps}(x)$  perturbada em um intervalo -0,5 a 1,5, tem-se a Figura 15:

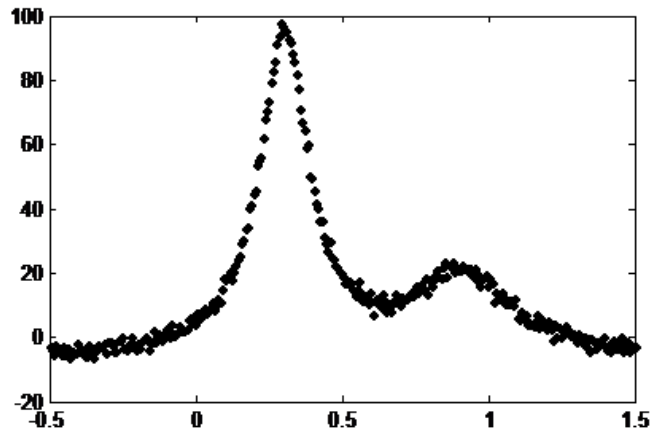


Figura 15: Gráfico da função de Humps no intervalo de -0,5 a 1,5

Encontrando a estimação dos parâmetros do modelo polinomial obteve-se os coeficientes da equação 3.1:

$$y = 0.0508 - 0.4460x + 1.6067x^2 - 2.9220x^3 + 2.3800x^4 + 0.3850x^5 - 2.2417x^6 + 0.1598x^7 + 0.5338x^8 - 0.0000x^9 + 0.0000x^{10} - 0.0000x^{11} + 0.0000x^{12} - 0.0000x^{13} + 0.0000x^{14} - 0.0000x^{15} + 0.0000x^{16} - 0.0000x^{17} + 0.0000x^{18} - 0.0000x^{19} \quad (3.1)$$

A aproximação da função empregando uma MLP e a regressão polinomial estão discriminadas na Figura 16:

Pela Tabela 4, percebe-se que o modelo de regressão explicou 99,72% da variação e que o modelo da rede neural MLP explicou 96,7% da proporção da variação total que foi explicada (devido) ao ajuste polinomial. Pode-se observar que os dados são bastante ajustados, corroborando o alto valor do coeficiente de determinação  $R^2$ .

Tabela 4: Coeficiente de determinação para a regressão polinomial e a rede perceptron múltiplas camadas (MLP).

Modelo	$R^2$ (%)
Regressão	99,72%
MLP	96,70%

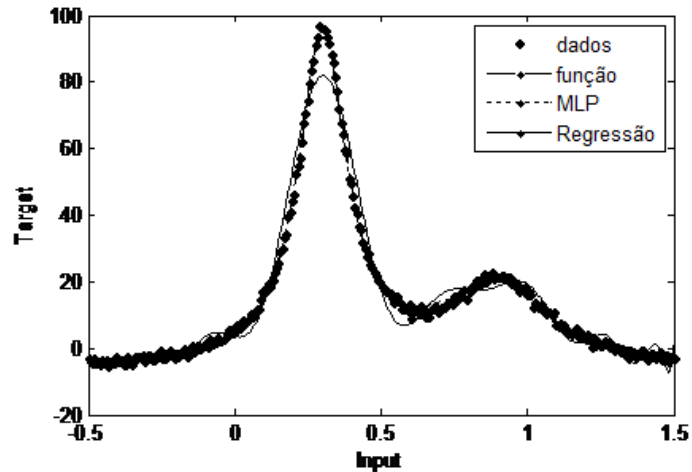


Figura 16: Gráfico de dispersão da função humps(x) perturbada, regressão polinomial de grau 19, e o ajuste da rede neural MLP

A dispersão da Figura 17 mostra uma correlação satisfatória entre o ajuste da rede MLP e a variável dependente de ambos os modelos. Este ajuste é confirmado através da estatística  $R^2$ , visto na Tabela 4.

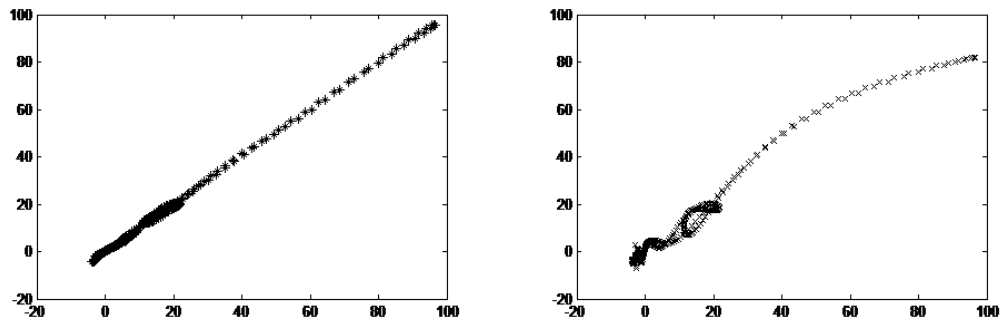


Figura 17: Valor ajustado pela rede versus o valor da variável independente com a Rede MLP (a) e Valor ajustado pela rede versus o valor da variável independente com a regressão polinomial (b)

A Figura 18 mostra os ajustes de ambos os modelos. Em que pode-se ver que não houve um aumento ou uma diminuição da dispersão dos dados em torno da reta na medida em que  $x$  cresce. Assim, a adequação do ajuste e as suposições do modelo podem ser verificadas através de uma análise dos resíduos, pois, esses dois gráficos mostram uma dispersão uniforme dos pontos em torno da reta ajustada, sem observações individuais discrepantes.

Pode-se observar também que a regressão polinomial adapta-se melhor que a rede neural MLP, pois se a dispersão dos resíduos da rede é bem inferior que a dispersão dos resíduos da rede neural. Porém, ambos os modelos apresentaram um ajuste satisfatório,

conforme se observa na Figura 16 e também é mostrado na Tabela 4.

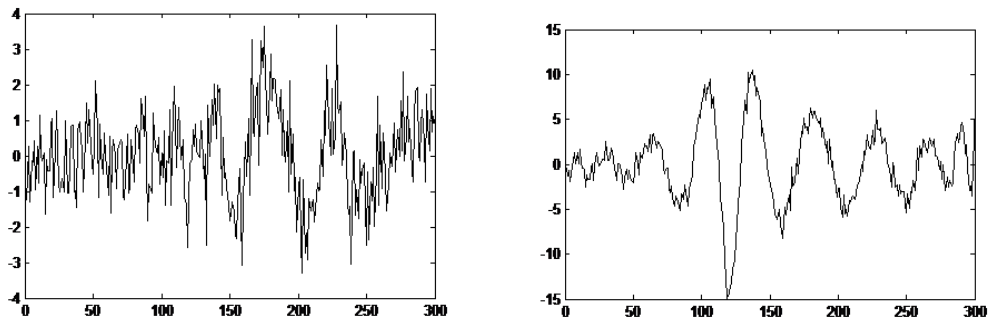


Figura 18: Formas de onda do erro da regressão polinomial (a) e Formas de onda do erro da rede MLP (b)

## 4 Conclusão

Apartir dos resultados obtidos nos experimentos, conclui-se que, no primeiro experimento, a curva gerada pelo modelo neural e o modelo de regressão produziram respostas de generalização satisfatória, visto que seus formatos ficaram mais próximos daquele esperado para a função  $y = x \cos(x)$ , que estava sendo comparada aos modelos. Nesse caso, o erro médio quadrático foi similar para os dois modelos propostos, revelando que houve ajuste e predição significativos. Vale salientar que existem outros modelos neurais como, por exemplo, função de base radial (RBF) bem como outros modelos estatísticos para ajuste e predição.

Já no segundo experimento, observa-se a partir das análises que as curvas estimadas pelo modelo neural e o modelo de regressão produziram respostas de generalização também satisfatórias, visto que o formato da regressão polinomial de grau 19 ficou mais próximo daquele esperado para a função  $y = \text{humps}(x)$ . Nesse caso, o erro médio quadrático foi similar para os dois modelos propostos, revelando que houve ajuste e predição significativos, também observou-se nesse exemplo o coeficiente de determinação  $R^2$ , que mostrou que a regressão polinomial de grau 19 explicou 99,72% da variação total do modelo e a rede neural explicou 96,7% da variação total do modelo. Não se deseja exaurir este assunto visto que podem ser usadas outras arquiteturas para uma melhor aproximação.

## 5 Referências Bibliográficas

- BENDER, E. A., *Mathematical Methods in Artificial Intelligence*, IEEE Computer Society Press, 1996.
- BISHOP, C. M., *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.
- BRAGA, Antônio P.; CARVALHO, André C. P. L. F.; LUDERMIR, Teresa B. *Redes Neurais Artificiais: teoria e aplicações*. Rio de Janeiro: LTC, 2007.
- BUSSAB, W. O. *Análise de variância e regressão*. 2.ed. São Paulo: Atual, 1988.
- Cont,Valdeir Del Francis Galton: Eugenia e hereditariedade São Paulo,2008
- CORRAR, Luiz J.; PAULO, Edilson; DIAS, José M. Filho. *Análise Multivariada: para os cursos de contábeis, Atuarias e Financeiras*. São Paulo: Atlas, 2007.
- FONSECA, Jairo Simon da; MARTINS, Gilberto de Andrade; TOLEDO, Geraldo Luciano. *Estatística Aplicada*; 2.ed. São Paulo: Atlas, 1985.
- FREEDMAN, L. S., FAINBERG, V., KIPNIS, V., MIDTHUNE, D. e CARROLL, R. J. A new Method for Dealing with Measurement Error in Explanatory Variables of Regression Models, *Biometrics*.2004.
- HAYKIN, S. *Redes neurais: principios e práticas*, 2. ed. Porto Alegre: Bookman Companhia Editora, 2001.
- KOVÁCS, Zolt László. *Redes neurais artificiais: fundamentos e aplicações: um texto básico*. 4. ed. São Paulo: Editora Livraria da Física, 2006.
- MCCULLOCH, W.S. E PITTS, S. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 1943.
- MICHALEWICZ, Z., and Fogel, D. B., *How to Solve It: Modern Heuristics*, Springer-Verlag, 2000.
- MONTGOMERY, D.C.; PECK, E.A e VINING, G.G. *Introducion to Linear Regression*



*Analysis, Third Edition.* Jonh Wiley, New York.2001.

SILVA, Edwirde L. Métodos de reducción de Redes neuronales RBF usando la decomposicion QLP para aplicaciones de ajuste de dados y clasificación, tesis de doctorado, Granada, Espanã, 2007.

SILVA, I.N.; Spatti, D.H., Flauzino, R.A., *Redes Neurais artificiais: para ciências aplicadas.* São Paulo: Artiber,2010.

REIS, Elizabeth. *Estatística Multivariada Aplicada*, 2 ed., Lisboa: Abril, 2001.

RIBEIRO, José L.; CATEN, Carla ten. Projeto de experimentos. Porto Alegre: Universidade Federal do Rio Grande do Sul, Escola de Engenharia, Programa de Pós-Graduação em Engenharia de Produção, 2003.

ROSENBLATT, F. Principles of Neurodynamics: Perceptrons and the theory of brain nechanisms. New York: Spartan Books, 1962.

RUMELHART, D. E., MCCLELLAND, J. L. Parallel distributed processing: explorations in the microstructure of cognition, Psychological and biological models, MIT Press, Cambridge, MA, 1986.

## APÊNDICE A – Apêndice

### Rotina: Matlab

# Algoritmo utilizado para o primeiro experimento: Experimento 1:

ndata = 100; (Número de pontos de dados)

noise = 0.2; (ruído ou erro)

x=unifrnd (-6,4,100,1); (gerar uma uniforme entre -6 ate 4, de tamanho 100x1)

x=linspace(-6,4,100)'

randn('state', 1)

t sem ruído = xcos(x); (função sem ruído)

plot(x,t sem ruído,'.'); (gráfico do xcos(x) sem ruído ou sem erros)

t = xcos(x)+ noise\*randn(ndata, 1); (função com ruído, randn= números aleatórios)

plot(x,t,'.'); (desenhar o gráfico)

O QUE A MLP PRECISA PARA AJUSTAR O COSSENO

nin = 1; (Número de entrada)

nhidden = 4; (Número de camadas escondidas)

nout = 1; (Número de saída - 1 saída = uma coluna de numeros (=100))

alpha = 0.01; (Coeficiente da taxa de aprendizagem)

CRIANDO E INICIALIZANDO OS VETORES DE PESOS DA REDE

(net = mlp(nin, nhidden, nout, 'linear', alpha))

OTIMIZANDO A REDE

```

options(1) = 1;          (mostrar o erro na hora do treinamento)

options(14) = 100;      $\sharp$(ciclos de treinamentos)

TREINANDO USANDO UMA ESCALA DE GRADIENTES

plotvals = linspace(-6,4,100)

plotvals = unifrnd(-6,4,100,1)

plotvals = sort(plotvals)

plotvals = [0:0.01:1]'; (predição)

y = mlpfwd(net, plotvals); (grar a MLP de acordo com a predição)

fh1 = figure

plot(x, t, 'ob')

hold on

xlabel('Input')

ylabel('Target')

axis([-6 4 -6 4])

plot(fx, fy, '-r', 'LineWidth', 2)

plot(plotvals, y, '-.', 'LineWidth', 2)

f = saida da regressão

f = polyval(P,x)

plot(x,f,'-g','LineWidth', 2)

legend('dados', 'função', 'MLP','Regressão')

t = funcao + ruido (1x100)

f = valores da regressão (1x100)

Gráfico - forma do erro do ajuste de regressão

plot(f-t, '-k')

```