



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA
DEPARTAMENTO DE ESTATÍSTICA



Comparando Gráfico CUSUM e Redes Neurais Artificiais no contexto do Controle Estatístico de Qualidade

Autora: Karla Faccio
Orientadora: Professora Dra. Liane Werner

Porto Alegre, 13 de Dezembro de 2007.

Universidade Federal do Rio Grande do Sul
Instituto de Matemática
Departamento de Estatística

Comparando Gráfico CUSUM e Redes Neurais Artificiais no contexto do Controle Estatístico de Qualidade

Autora: Karla Faccio

Monografia apresentada para obtenção
do grau de Bacharel em Estatística.

Banca Examinadora:
Professora Dra. Liane Werner
Professor João Henrique Ferreira Flores

Porto Alegre, 13 de Dezembro de 2007.

“Sempre trace seu futuro, mas trace-o a lápis”.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais Ana e Gilberto, tanto pelo apoio emocional quanto financeiro, por estarem sempre ao meu lado, me dando muito carinho, amor... enfim muito obrigada pela dedicação. Agradeço de coração e amo vocês!!!

Não posso deixar de agradecer às minhas queridas avós Aracy e Lourdes, por toda a força que me passaram através de palavras de conforto e muita reza. Valeu!

Agradeço também ao meu maravilhoso mano Rica, você é nota mil! Que durante todo este tempo me compreendeu e me apoiou em tudo.

Agradeço à minha querida professora e orientadora Liane Werner, obrigada por toda a dedicação e paciência, valeu também por todo apoio emocional!

Agradeço ao professor Alberto Cargnelutti Filho pela disponibilidade em me ajudar sempre que possível.

Agradeço à professora Suzi Camey por todas as dicas e idéias que me deste a fim de melhorar este trabalho.

Agradeço ao professor da UNESP Antônio Sérgio Ferraudo por todos os materiais que me passou sobre redes neurais e também a todo o incentivo.

Agora quero agradecer também aos meus queridos colegas, que de uma maneira ou de outra sempre estiveram me auxiliando durante estes quatro anos. André Korzenowski muito obrigado por tudo, você foi um colega e é um amigo nota dez, me ajudou sempre e em tudo, valeu! Queridas colegas Patrícia Simões e Gisele Ferreira, vocês, além de colegas, são amigas incríveis, tenho certeza que a nossa amizade vai ser para a vida toda, obrigada de coração por tudo!! E também quero agradecer ao colega Rodrigo Coster, por toda a ajuda em relação ao software R.

SUMÁRIO

1. INTRODUÇÃO.....	6
1.1 OBJETIVO.....	7
1.2 JUSTIFICATIVA.....	7
1.3 METODOLOGIA.....	9
1.4 ESTRUTURA.....	10
2. REFERENCIAL TEÓRICO.....	11
2.1 Gráficos de Controle.....	11
2.1.1 Gráfico CUSUM.....	13
2.1.1.1 Gráfico CUSUM Tabular.....	14
2.1.1.2 Gráfico CUSUM Máscara V.....	20
2.2 Redes Neurais Artificiais.....	23
2.2.1 Histórico.....	23
2.2.2 Neurônio biológico.....	25
2.2.3 Neurônio artificial.....	27
2.2.4 Rede Neural Artificial (RNA).....	30
2.2.5 Arquitetura de uma Rede Neural.....	30
2.2.6 Rede Perceptron de Múltiplas Camadas (MLP).....	32
2.2.7 Rede RBF – Funções Base Radial.....	34
3. ANÁLISES.....	37
3.1 Metodologia a ser aplicada para análise.....	37
3.2 ETAPA 1 - Geração dos dados.....	39
3.3 ETAPA 2 - Gráfico CUSUM.....	40
3.3 ETAPA 3 - Rede RBF.....	41
3.4 ETAPA 4 - Comparação do Gráfico CUSUM e da rede RBF.....	43
4. CONSIDERAÇÕES FINAIS.....	45
5. REFERÊNCIAS.....	47
7. ANEXO 1.....	49

1. INTRODUÇÃO

Atualmente os consumidores estão cada vez mais exigentes em relação à qualidade dos produtos que pretendem adquirir, sem falar na enorme concorrência, onde sobrevivem apenas as melhores organizações. Então a manutenção e melhoria da qualidade são fundamentais para qualquer empresa que almeje sobreviver neste mercado tão exigente e competitivo. Monitorar o comportamento da variabilidade no processo produtivo é tarefa primordial para o bom andamento da linha de produção das empresas e uma das formas de atingir tal meta é a através do Controle Estatístico de Qualidade (CEQ). O CEQ é formado por um conjunto de ferramentas muito importantes para a obtenção, manutenção e melhoria da qualidade de produtos e serviços produzidos por uma organização (WERNER, 1996).

Este monitoramento pode ser feito através de Gráficos de Controle, tanto os desenvolvidos por Shewhart quanto o gráfico CUSUM. Estes gráficos distinguem as variações comuns (controladas) das variações especiais (não controladas), além de monitorarem a variabilidade do processo ao longo do tempo. A diferença entre os gráficos propostos por Shewhart e o gráfico CUSUM consiste no fato que o último serve para detectar pequenas variações na média que os gráficos de Shewhart não conseguem detectar.

Outra ferramenta que vem crescendo e expandindo suas aplicações na área do Controle Estatístico de Qualidade, principalmente pelo reconhecimento de padrões, são as chamadas Redes Neurais Artificiais (RNA), que podem desempenhar o papel das Cartas de Controle (LEGER et al, 1998). Um dos grandes atrativos das RNAs são as possibilidades de aprender através de exemplos e de generalizar a informação aprendida, pois são técnicas computacionais que possuem um modelo matemático inspirado na estrutura neural de seres inteligentes e que adquirem conhecimento através da experiência (SELAU, 2000).

Face às novas perspectivas com as RNAs em monitorar dados observados ao longo do tempo, neste trabalho será feita a análise de um conjunto de dados simulado, utilizando o Gráfico de Controle CUSUM e Redes Neurais Artificiais (RNA), a fim de comparar os dois métodos em relação à detecção do comportamento destes dados.

1.1 OBJETIVO

Comparar a eficiência do gráfico de controle CUSUM (Gráfico de Soma Cumulativa) com Redes Neurais Artificiais, a fim de verificar qual das duas técnicas consegue detectar melhor o comportamento da série em estudo. Como objetivo secundário, realizar simulação de dados contínuos para concretizar esta comparação

1.2 JUSTIFICATIVA

Estando os clientes cada vez mais exigentes e, conseqüentemente as empresas procurando oferecer produtos e serviços com a melhor qualidade, torna-se necessário utilizar técnicas do controle estatístico de qualidade, dentre as quais os gráficos de controle, para monitorar o processo produtivo (MONTGOMERY, 2000).

Por outro lado, com o avanço dos estudos em relação aos neurônios humanos, surgiram as redes neurais artificiais (RNA), que possuem inúmeras aplicações na prática, dentre as quais a utilização delas na análise de séries temporais.

Como os dados analisados na área de controle de qualidade nas empresas, que são observados ao longo do tempo, podem ser considerados uma série temporal, é viável a sua utilização para este fim. Conforme Kovács (2002, p. 138) “A aplicação de redes neurais em funções voltadas ao operador, particularmente a detecção e diagnóstico de falhas e processamento de alarmes tem sido explorado com algum êxito por vários pesquisadores. Diagnóstico de falhas na operação de sistemas pode ser reduzido a um problema de detecção de padrões, no caso padrões anômalos, e já se sabe que redes neurais são excelentes detectores de padrões”.

Ainda de acordo com Smith (1994), que para o reconhecimento de padrões utilizou Redes Neurais Artificiais por retropropagação do erro, detectando amostras que apresentam mudanças na variabilidade e na média da característica de qualidade de interesse. Os resultados foram comparados aos gráficos de controle tradicionais \bar{X} e R , e verificou-se que para o caso de grandes desvios as RNAs foram idênticas aos gráficos, mas para

pequenos desvios as RNAs foram superiores. Segundo Samohyl et al. (2006) os gráficos de controle de Shewhart são eficientes para detectar grandes mudanças na média do processo, mas para pequenos desvios o gráfico CUSUM é mais sensível.

Logo, seria importante comparar o gráfico CUSUM com redes neurais artificiais, já que o estudo de Smith (1994) mostrou que as RNAs foram mais eficientes para detectar pequenas mudanças na média do processo, propriedade esta presente nos gráficos CUSUM. Esta lacuna já foi investigada por Leger et al. (1998), que utilizaram o gráfico de controle CUSUM e Redes Neurais Artificiais para detecção e diagnóstico de defeitos no modelo do sistema de transporte de energia em um reator nuclear. Os resultados mostraram que um dos tipos de RNA estudado foi o que menos detectou alarmes falsos, e quando combinado com o gráfico CUSUM foi capaz de eliminar todos os alarmes falsos, mas o diagnóstico para os defeitos foi inconclusivo. Quando se utilizou apenas o gráfico CUSUM, este detectou o maior número de alarmes falsos, não sendo assim a técnica mais indicada neste caso.

O estudo comparativo de RNA e gráficos de controle ainda é insipiente e necessita ser aprofundado. Aplicar ambas as técnicas em um processo produtivo simulado que monitorem variáveis contínuas e compará-las vai permitir mostrar uma maneira alternativa de controlar a qualidade nas empresas não se restringindo apenas aos gráficos de controle estatístico, mas sim utilizando uma ferramenta que possui inúmeros benefícios, como as redes neurais artificiais.

Segundo Haykin (2001, p. 28) “é evidente que uma rede neural extrai seu poder computacional através, primeiro, de sua estrutura maciçamente paralelamente distribuída e segundo sua habilidade de aprender e, portanto de generalizar. A generalização se refere ao fato de a rede neural produzir saídas adequadas para entradas que não estavam presentes durante o treinamento. Estas duas capacidades de processamento de informação tornam possível para as redes neurais resolver problemas complexos que são atualmente intratáveis”.

Levando em conta os benefícios da nova proposta (redes neurais), procura-se aplicar os conhecimentos e ferramentas do qual se dispõe, e assim amplia-se a aplicabilidade no controle de qualidade.

1.3 METODOLOGIA

Este trabalho pode ser classificado como uma pesquisa exploratória, que visa proporcionar uma visão geral de um determinado fato, do tipo aproximativo. Um trabalho é de natureza exploratória quando inclui levantamento bibliográfico, entrevistas com pessoas que tiveram (ou têm) experiências práticas com o problema pesquisado e análise de exemplos que estimulem a compreensão. Este tipo de pesquisa tem por objetivo desenvolver, esclarecer e modificar conceitos, proporcionando um maior conhecimento para o pesquisador acerca do assunto, a fim de que esse possa formular problemas mais precisos ou criar hipóteses que possam ser pesquisadas por estudos posteriores (GIL, 1999).

Primeiramente, será estruturado um referencial teórico sobre gráficos de controle e redes neurais artificiais envolvendo levantamento através de livros, artigos científicos, materiais na internet e outros. Esta revisão permitirá uma maior base para realizar a comparação dos métodos envolvidos.

Posteriormente será gerado um conjunto de dados que servirá de referência, e será avaliado o número de pontos fora de controle destes dados através do gráfico CUSUM. Após, serão simuladas 400 séries com o mesmo comportamento do conjunto de dados base e será aplicado o gráfico CUSUM. Seguindo, será modelado, para cada série, uma rede neural artificial, e esta modelagem será comparada com o modelo referência. Por fim, irá ser aplicado um teste de comparação de médias para o número de pontos classificados como diferentes da série referência via método CUSUM e via redes neurais artificiais, para desta forma verificar qual dos dois métodos foi mais eficiente em detectar o comportamento da série base. A geração, simulação e análises dos dados serão feitas com o auxílio do software R v 2.6.0.

1.4 ESTRUTURA

Este trabalho está estruturado em quatro capítulos distintos.

O capítulo 1 apresenta a introdução juntamente com o objetivo traçado, além da justificativa, metodologia a ser utilizada e estrutura da monografia.

No capítulo 2 há todo o referencial teórico necessário para a compreensão das análises que serão apresentadas posteriormente. Este capítulo mostrará as definições e conceitos básicos sobre gráficos de controle e redes neurais artificiais.

O capítulo 3 apresenta em detalhes como os dados foram simulados, após mostra a construção e análises utilizando o gráfico CUSUM e redes neurais artificiais, que por fim apresenta a análise que permitirá a comparação entre estes dois métodos.

O capítulo 4 apresenta as conclusões e sugestões de trabalhos futuros. As conclusões obtidas através do trabalho permitiram ao leitor verificar a contribuição que a RNA tem na área do controle de qualidade.

2. REFERENCIAL TEÓRICO

2.1 Gráficos de Controle

Os gráficos de controle são úteis para monitorar a variabilidade de um processo, e distinguem dois tipos de variação, isto é, separam as causas comuns (variabilidade inerente) das especiais (variabilidade excessiva), informando assim se o processo está ou não sob controle estatístico, em geral, tal distinção é realizada verificando a existência de pontos fora dos limites de controle. Os gráficos de controle são formados por uma linha central (média da estatística monitorada), um par de limites de controle (limite superior e limite inferior), valores da característica da qualidade e número da amostra, sendo que a plotagem dos dados deve ser de forma cronológica (MONTGOMERY, 2000).

Os tipos de gráficos de controle geralmente são classificados de acordo com a maneira que as características da qualidade são avaliadas: por atributos ou por variáveis. Existem tipos de gráficos por atributos e pode-se separá-los em dois grupos, segundo Besterfield (1979), um grupo de gráficos são aqueles que monitoram as unidades não-conformes e fazem parte deste os gráficos *p* e o *np*, e o outro grupo de gráficos são aqueles que monitoram as não-conformidades nas unidades e fazem parte deste os gráficos *c* e *u*. Os gráficos de controle de Shewhart por variáveis, conforme Montgomery (2000) e Werkema (1995) são: gráfico \bar{X} , que monitora a variabilidade entre os subgrupos, e as amostras em cada subgrupo devem ser constantes e possuem dois ou mais elementos; o gráfico *R* (amplitude), que controla a variabilidade dentro dos subgrupos, e as amostras devem ser constantes e preferencialmente possuem entre dois a dez elementos, pois para amostras maiores a amplitude torna-se ineficiente; gráfico *S* (desvio padrão), que também serve para controlar a variabilidade do processo, além de produzir uma estimativa do desvio padrão melhor do que a fornecida pelo gráfico da amplitude.

Mas existem também os chamados gráficos de controle de Shewhart para medidas individuais, que possuem amostras de tamanho unitário. Utilizados principalmente nas indústrias químicas, nos processos com taxa de produção baixa, em testes destrutivos e na inspeção automatizada (toda unidade produzida é avaliada). Em muitas aplicações dos

gráficos para unidades individuais usa-se a amplitude móvel de duas observações consecutivas como base para estimar a variabilidade do processo (MONTGOMERY, 2000). A amplitude móvel é definida segundo a equação (1):

$$MR_i = |x_i - x_{i-1}| \quad (1)$$

Para o gráfico de controle das medidas individuais, os parâmetros são definidos através das fórmulas (2), (3) e (4).

$$LSC = \bar{x} + 3 \frac{\overline{MR}}{d_2} \quad (2)$$

$$\text{Linha Central} = \bar{x} \quad (3)$$

$$LSC = \bar{x} - 3 \frac{\overline{MR}}{d_2} \quad (4)$$

Como os gráficos de Shewhart utilizam informações apenas do último ponto plotado (ao invés de utilizarem toda seqüência de pontos), esta característica faz com que os gráficos de controle de Shewhart sejam relativamente insensíveis a pequenas mudanças no processo, na ordem de 1,5 desvio padrão (ou menos). Outro critério pode ser aplicado aos métodos de Shewhart, como, por exemplo, as regras de sensibilidade e o uso de limites de controle que procuram incorporar a informação de todo o grupo de pontos no procedimento de tomada de decisão. Entretanto, o uso destas regras de sensibilidade suplementares reduz a simplicidade e a facilidade da interpretação dos gráficos de controle de Shewhart. Junto a isso, o uso destas regras pode reduzir o comprimento médio da seqüência (ARL) do gráfico de controle, quando o processo está sob controle, antes de um alarme falso, que poderia ser indesejado (MONTGOMERY, 2000).

Um critério para confirmar o desempenho dos gráficos de controle, é o ARL (Comprimento Médio da Seqüência). Ele tem a função de determinar quantas amostras são necessárias para que os gráficos de controle apresentem um sinal indicativo de estado fora de controle. ARL_0 é o número esperado de amostras antes de um alarme falso ser dado, quando o processo está sob controle estatístico (VARGAS et al, 2004). O cálculo do ARL é obtido através da equação (5).

$$ARL = \frac{1}{\alpha} \quad (5)$$

Onde: α é a probabilidade para qualquer um dos pontos cair fora dos limites de controle quando o processo estiver sob controle. Por exemplo, para ilustrar o ARL de um gráfico 3σ , onde $\alpha=0,0027$, quando o processo estiver sob controle é:

$$ARL_0 = \frac{1}{\alpha} \Rightarrow ARL_0 = \frac{1}{0,0027} = 370$$

Esse valor do ARL_0 encontrado indica que se o processo se mantiver sob controle um sinal de fora de controle ocorrerá, em média, após 370 amostras.

As cartas de controle de Shewhart são substituídas pelo gráfico de controle CUSUM quando há interesse em detectar pequenos desvios do valor nominal. Este gráfico, que acumula as informações mais recentes com as informações anteriores, é considerado altamente eficiente na detecção de causas especiais de variação, que conduzem às não-conformidades na produção (MONTGOMERY, 2000).

2.1.1 Gráfico CUSUM

Gráficos CUSUM, que foram introduzidos por Page em 1954, incorporam diretamente todas as informações na seqüência dos valores amostrais, ou seja, acumulam informações das amostras de um processo ponderando-as igualmente, plotando a soma acumulada dos desvios da amostra proveniente de um valor objetivo (desejado). O que permite que sejam mais eficientes que os gráficos de Shewhart na detecção de pequenas variações na média de um processo.

A técnica de soma acumulada pode ser aplicada tanto na construção do gráfico CUSUM para observações individuais como para observações amostrais das médias de subgrupos. Para as observações individuais, a estatística utilizada é a soma acumulada dos desvios de cada valor individual com relação à medida dada pelo valor objetivo (desejado). E no caso das amostras de tamanho $n > 1$, esta estatística é a soma acumulada dos desvios da média amostral com relação ao valor nominal. Mas os gráficos de soma acumulada são mais eficientes com amostras de tamanho unitário, isto é, são indicados para situações onde o monitoramento do processo é feito através de observações individuais (ALVES, 2003).

O procedimento inicia com o cálculo dos desvios do valor nominal, isto é, a diferença entre o valor observado (ou média amostral) e o valor nominal μ_0 . Tendo este

desvio, a soma acumulada é iniciada. A soma C_i para o i -ésimo período é a soma de todos os desvios do valor nominal desde o período 1 até o período i , dada pela equação (6):

$$C_i = \sum_{j=1}^i (X_j - \mu_0) = (X_i - \mu_0) + C_{i-1}, i \geq 1 \quad (6)$$

Onde: X_j é a j -ésima observação de um gráfico CUSUM para observações individuais

Agora se quiser a soma acumulada para um gráfico CUSUM que utiliza as médias de um subgrupo, usa-se a equação (7):

$$C_i = \sum_{j=1}^i (\bar{x}_j - \mu_0) \quad (7)$$

onde assume-se que as amostras coletadas têm tamanho $n \geq 1$ e \bar{x}_j é a média da j -ésima amostra.

Se o processo se mantém sob controle no valor alvo μ_0 , a soma acumulada definida em (6) e em (7) descreve uma maneira aleatória com média zero. Por outro lado, se a média varia para qualquer valor superior a $\mu_1 > \mu_0$, então uma tendência ascendente vai desdobrar-se na soma cumulativa C_i . Reciprocamente, se a média se altera para um valor abaixo de $\mu_1 < \mu_0$, a soma cumulativa C_i , vai ter uma direção negativa. Considerando isto, se, nos pontos demarcados, uma tendência ascendente ou descendente aparecer, ela deve ser considerada uma evidência que a média do processo mudou, e uma procura pela causa determinante deve ser feita (MONTGOMERY, 2000).

Há duas maneiras de representar os gráficos de soma cumulativa: a forma Tabular e a forma Máscara V. A seguir serão mostrados em detalhes estes dois formatos de gráfico CUSUM.

2.1.1.1 Gráfico CUSUM Tabular

É um algoritmo que utiliza soma cumulativa para calcular as somas acumuladas unilaterais que através do gráfico são comparadas com o intervalo de decisão $H = \pm h$. Mas se um valor da soma for maior do que este intervalo o processo é dito fora de controle. Este procedimento serve para monitor a média de valores individuais ou a média de subgrupos (ALVES, 2003).

Seja x_i cada observação do processo controlado que se supõe sob controle. Supondo ainda que os dados coletados seguem distribuição normal com média μ_0 e desvio padrão σ . Este gráfico é um plano de controle com intervalos bilaterais simétricos que utiliza duas estatísticas unilaterais C_i^+ para detectar mudanças positivas e C_i^- para detectar mudanças negativas. São calculados através das equações (8) e (9),

$$C_i^+ = \max[0, x_i - (\mu_0 + K) + C_{i-1}^+] \quad (8)$$

$$C_i^- = \max[0, (\mu_0 - K) - x_i + C_{i-1}^-] \quad (9)$$

Onde $C_1^+ = C_1^- = 0$, e K (fator de sensibilidade) deve ser escolhido de forma que o valor da soma $\mu_0 + K\sigma$ ou $\mu_0 - K\sigma$ esteja entre a média do processo e a média deslocada que se queira avaliar (SAMOHYL et al, 2006).

Através do algoritmo de soma acumulada são obtidos valores dos desvios de C_i^+ e C_i^- , que são colocados em uma tabela e acumulados sucessivamente. A soma acumulada destes desvios é comparada com um intervalo de decisão H . Se $C_i^+ > H$ ou $C_i^- < H$, então o processo é considerado fora de controle.

O valor de referência K é escolhido considerando o valor μ_1 para qual o processo pode ter mudado e que se está interessado em detectar rapidamente. Geralmente esta mudança é expressa em termos de desvio padrão, e assim, K é a metade do valor que se tem interesse em detectar rapidamente, ou

$$K = \frac{\Delta}{2} \sigma \quad (10)$$

Onde: Δ é a magnitude da variação, ou seja, $\Delta = |\mu_1 - \mu_0|$ e σ é o desvio padrão do processo.

Pode-se afirmar que K está diretamente relacionado com a magnitude da mudança que se deseja detectar através do gráfico CUSUM. Quanto menor K , menor será a faixa de variação que o gráfico será capaz de detectar e maior será a sensibilidade do gráfico (ALVES, 2003).

O gráfico CUSUM Tabular é projetado pela escolha de valores razoáveis para H (intervalo de decisão) e K (valor de referência), recomenda-se que estes parâmetros sejam selecionados para produzir um bom ARL (Comprimento Médio da Sequência).

O melhor modelo, segundo Montgomery (2000), para selecionar o valor de referência K e o intervalo de decisão H , é dado pelas equações de (11) a (13):

$$K = k \frac{\sigma}{\sqrt{n}} \quad (11)$$

$$\text{Limite Superior: } H = h \frac{\sigma}{\sqrt{n}} \quad (12)$$

$$\text{Limite Inferior: } H = -h \frac{\sigma}{\sqrt{n}} \quad (13)$$

Onde: n é o tamanho da amostra, σ é o desvio padrão da variável amostral usada, k é o valor de referência padronizado (freqüentemente usa-se $k=0,5$) e h é o intervalo de decisão padronizado (geralmente $h=4$ ou $h=5$). Estes valores para k e h freqüentemente usados produzem um gráfico CUSUM com boas propriedades do ARL com uma mudança de cerca 1σ na média do processo conforme a tabela 1. E para o caso de observações individuais ($n=1$) deve-se substituir o termo σ / \sqrt{n} por σ , nas equações (11), (12) e (13).

Verifica-se através da tabela 1, que $h=4$ resulta em um $ARL_0= 168$ amostras sob controle, enquanto que $h=5$ resulta em um $ARL_0= 465$ amostras. Se escolhermos $h=4,77$, isso resultará em um CUSUM com $ARL_0= 370$ amostras, o que coincide com o valor de ARL_0 para um gráfico de controle de Shewhart com os limites 3σ habituais. Uma vez k selecionado, escolhe-se h para dar o desempenho desejado de ARL_0 dentro do controle.

Tabela 1. Desempenho do CUSUM Tabular com $k=1/2$ e $h=4$ ou $h=5$

Mudança na média (múltiplo de σ)	$h=4$	$h=5$
0	168	465
0,25	74,2	139
0,50	26,6	38
0,75	13,3	17
1,00	8,38	10,4
1,50	4,75	5,75
2,00	3,34	4,01
2,50	2,62	3,11
3,00	2,19	2,57
4,00	1,71	2,01

Fonte: Montgomery, 2000

Segundo Hawkins (1993a), através a tabela 2, verifica-se os valores para k e os correspondentes valores de h para um $ARL_0=370$. Estes valores de k e h foram calculados com a mesma confiabilidade do que a dos gráficos de Shewhart, ou seja, há uma probabilidade de 0,27% do gráfico apresentar um sinal de alarme sem que a média tenha se deslocado verdadeiramente de seu valor original (alarme falso).

Tabela 2. Valores de k e h que fornecem $ARL_0=370$ sugeridos para o gráfico CUSUM Tabular

K	0,25	0,5	0,75	1,0	1,25	1,5
H	8,01	4,77	3,34	2,52	1,99	1,61

Fonte: Hawkins, 1993^a

O algoritmo de soma acumulada pode ser colocado no formato da tabela 3, formada por três conjuntos de colunas. O primeiro conjunto com os valores observados, o segundo com o desvio, a soma acumulada unilateral C_i^+ e o valor N^+ . O terceiro com o desvio, a soma acumulada unilateral C_i^- e o valor N^- . As quantidades N^+ e N^- indicam o número de períodos consecutivos em que os CUSUMs C_i^+ e C_i^- foram não-nulos, imediatamente após ter sido zero. A contagem N^+ e N^- inicia ou recomeça sempre que o valor C_i^+ ou C_i^- assume valor zero, por exemplo, o contador N^+ registra o número de períodos consecutivos desde que o CUSUM C_i^+ ficou acima do valor zero. Então, o CUSUM tabular indica quando a mudança provavelmente ocorreu (MONTGOMERY, 2000).

Tabela 3. Tabela do algoritmo de soma acumulada

	a				b		
i	X_i	$X_i-(\mu_0+K)$	C_i^+	N_i^+	$(\mu_0-K)-X_i$	C_i^-	N_i^-
1	X_1	$X_1-(\mu_0+K)$	C_1^+	N_1^+	$(\mu_0-K)-X_1$	C_1^-	N_1^-
2	X_2	$X_2-(\mu_0+K)$	C_2^+	N_2^+	$(\mu_0-K)-X_2$	C_2^-	N_2^-
3	X_3	$X_3-(\mu_0+K)$	C_3^+	N_3^+	$(\mu_0-K)-X_3$	C_3^-	N_3^-
...
n	X_n	$X_n-(\mu_0+K)$	C_n^+	N_n^+	$(\mu_0-K)-X_n$	C_n^-	N_n^-

Fonte: Alves, 2003

Se o processo estiver fora de controle o analista deve encontrar as causas especiais e estimar seu valor médio para que possa fazer o ajuste adequado. Então, é muito importante

que o analista compare o valor das somas acumuladas C_i^+ e C_i^- com o intervalo de decisão H para cada amostra i coletada.

Mas é possível monitorar também o processo de variabilidade através dos gráficos de controle CUSUM, desde que estes sejam utilizados com observações individuais. Hawkins (1993a) sugere um novo valor padronizado para observações individuais, como é expresso na equação (14).

$$v_i = \frac{\sqrt{|y_i|} - 0,822}{0,349} \quad (14)$$

Onde: $y_i = \frac{X_i - \mu_0}{\sigma}$, $i = 1, 2, \dots, n$ é o valor padronizado de X_i (que possui distribuição normal com média μ_0 e desvio padrão σ).

Logo, o gráfico CUSUM Tabular padronizado é formado pelas equações (15) e (16).

$$S_i^+ = \max\{0, v_i - k + S_{i-1}^+\} \quad (15)$$

$$S_i^- = \max\{0, -k - v_i + S_{i-1}^-\} \quad (16)$$

Onde: S_i^+ e S_i^- são as somas acumuladas padronizadas; $v_i \sim \text{Normal}(0;1)$; μ_0 é o valor alvo do processo; k é o valor de referência (em unidades de desvio padrão) e os valores de k e h são selecionados como no gráfico de CUSUM para controlar a média do processo. Este gráfico padronizado sinaliza quando $S_i^+ > h^+$ ou quando $S_i^- < h^-$.

A figura 1 ilustra um típico gráfico CUSUM bilateral.

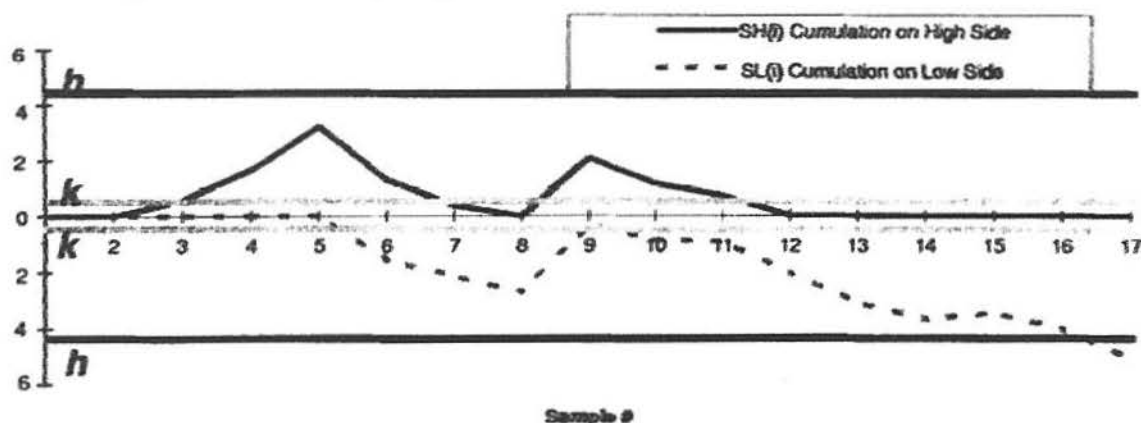


Figura 1. Típico gráfico CUSUM

Fonte: Leger et al, 1998

$$\hat{\mu} = \mu_0 - K - \frac{C_i^-}{N^-}, \text{ caso } C_i^- > H \quad (18)$$

2.1.1.2 Gráfico CUSUM Máscara V

Consiste em um método gráfico alternativo que permite, por inspeção dos pontos representados no gráfico CUSUM, decidir se ocorreu ou não um desvio no valor médio μ_0 desejado no processo.

A máscara V é aplicada a sucessivos valores estatísticos obtidos conforme a equação (19):

$$C_i = \sum_{j=1}^i y_j = y_i + C_{i-1} \quad (19)$$

Onde: y_i é a observação padronizada $y_i = (x_i - \mu_0)/\sigma$.

O método máscara V nada mais é do que uma moldura visual que desempenha o papel semelhante aos limites de controle nas cartas de controle de Shewhart. A máscara é colocada no gráfico com sua origem sobre o último ponto registrado, e assim, os pontos cobertos por ela são uma indicação de que o processo sofreu mudança. Porém, para a construção do gráfico se supõem que a saída do processo tem distribuição Normal e o desvio padrão do processo está sob controle (WERNER, 1996).

Montgomery (2000) adaptou o valor real do ARL_0 para o método máscara V, então neste método o ARL sob controle deve ser $ARL_0 = 1/2\alpha$. Por exemplo, para resultar em um ARL_0 de 370, o valor de α precisa ser igual a 0,00135 e em um ARL_0 de 500 o valor de α precisa ser 0,001. Onde α representa a probabilidade para qualquer um dos pontos cair fora dos limites de controle quando o processo estiver sob controle.

A figura 3 mostra um gráfico CUSUM com a máscara V. O ponto 0 da figura 3 coincide com o último ponto representado no gráfico CUSUM e fica a uma distância igual a d unidades amostrais do vértice P da máscara V. Além disto, d é a distância entre 0 e P, e θ é o ângulo entre um dos braços da máscara V e $d = \overline{0P}$.

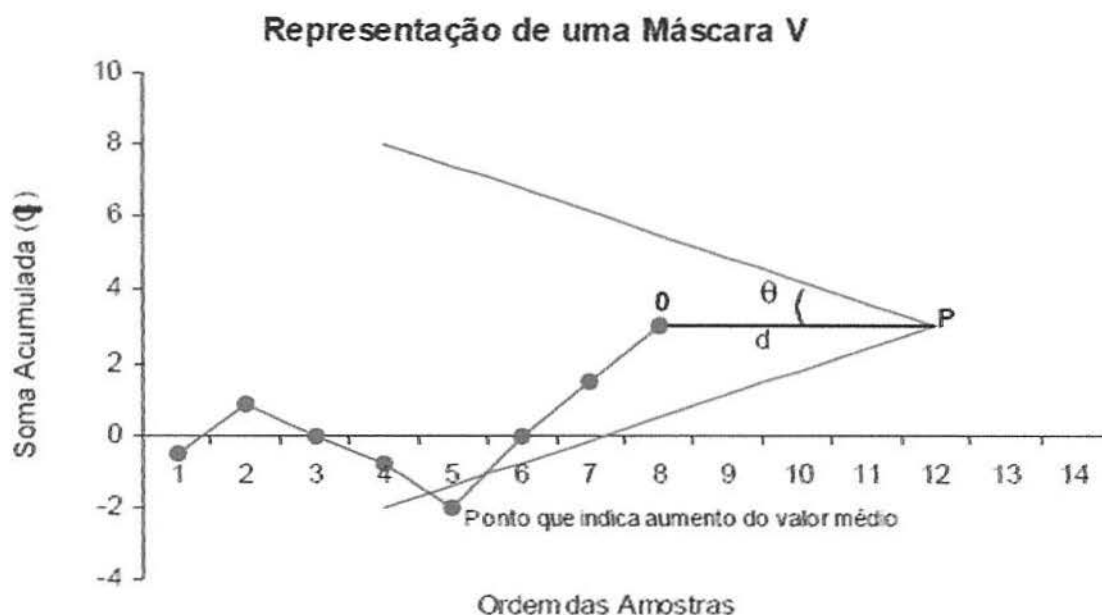


Figura 3. Exemplo de Máscara V

Fonte: Alves, 2003

A linha OP é paralela ao eixo horizontal e os braços são simétricos em relação a ela, formando ângulos iguais a θ . Se todas as somas cumulativas C_1, C_2, \dots, C_i ficarem dentro dos braços da máscara V , o processo estará sob controle. Mas se qualquer ponto ficar fora dos braços da máscara V , o processo é considerado fora de controle.

O desempenho do gráfico CUSUM associado à máscara V depende da escolha dos parâmetros d e θ que permitem a sua construção. E quanto maior forem estes parâmetros menos freqüentes serão as interrupções do processo.

A construção de um gráfico CUSUM pelo método máscara V , segundo Werner (1996), segue basicamente os seguintes passos:

- Obter uma estimativa para $\frac{\sigma}{\sqrt{n}}$.
- Determinar o valor mínimo da mudança D na média que se deseja detectar e calcular $\delta = \frac{D\sqrt{n}}{\sigma}$. Onde: $D = (\mu_1 - \mu_0)$ e δ representa a amplitude ou tamanho da mudança do valor médio do processo em unidades de desvio padrão.

- Determinar o nível de probabilidade no qual as decisões serão tomadas. Para os limites equivalentes aos limites de Shewhart, $\alpha=0,00135$.
- Definir k como fator de escala que é a mudança numérica no valor da variável (escala vertical) por unidade de mudança em distância (entre duas demarcações) na escala horizontal (número do subgrupo).
- Para obter a distância d , começar com $\delta = \frac{D\sqrt{n}}{\sigma}$ na tabela 2 onde $\alpha_0=0,00135$.
- Para obter o ângulo θ , calcule D/k . Substituindo D/k por δ , olhe na tabela 4 e obtenha (ou faça interpolação) θ na linha correspondente a D/k .
- Usando d e θ construir a máscara para definir os limites de controle.

Tabela 4. Valores de d e θ para o tamanho da mudança do valor médio do processo que se pretende detectar.

δ	θ	d
0,2	5°43'	330,4
0,5	14°00'	52,9
0,8	21°48'	20,6
1,0	26°34'	13,2
1,3	32°59'	7,8
1,6	38°40'	5,2
1,8	41°59'	4,1
2,0	45°00'	3,3
2,4	50°12'	2,3
2,8	54°28'	1,7
3,0	56°19'	1,5

Fonte: Werner, 1996

2.2 Redes Neurais Artificiais

Alguns tipos de redes neurais artificiais podem ser observados através da figura 4, que foram classificadas de acordo com o tipo de dado de entrada (dados binários ou dados contínuos) e posteriormente pela forma de aprendizado (supervisionado ou não supervisionado). Mas particularmente neste trabalho será utilizada apenas uma rede neural que trabalhe com valores contínuos e supervisionada: RBF (Função Base Radial).

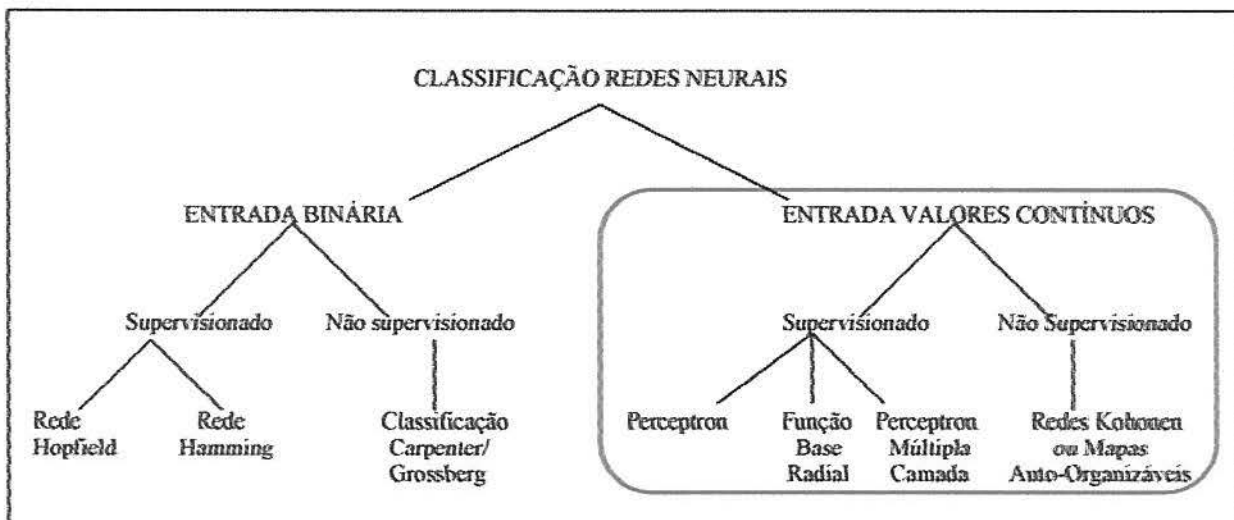


Figura 4. Classificação Redes Neurais
Fonte: adaptado de Lau, 1991

2.2.1 Histórico

A origem da teoria das redes neurais remonta aos modelos matemáticos e aos modelos de engenharia de neurônios biológicos. A célula nervosa foi identificada anatomicamente e descrito pelo neurologista espanhol Ramón y Cajal no século 19 (KOVÁCS, 2002).

As primeiras informações mencionadas sobre a neurocomputação datam de 1943, em artigos de McCulloch e Pitts, em que sugeriram a construção de uma máquina baseada ou inspirada no cérebro humano. Estes autores descreveram um cálculo lógico das redes neurais que unificava os estudos de neurofisiologia e da lógica matemática, pois McCulloch

foi um psiquiatra e neuroanatomista por treinamento, e Pitts foi um matemático que se associou-se a McCulloch em 1942. Porém eles não desenvolveram aplicações práticas para seu trabalho, e os sistemas que propuseram não tinham a capacidade de aprender. Contudo, o artigo de 1943 destes autores estabeleceu as bases para o surgimento das redes neurais artificiais e inteligência artificial (HAYKIN, 2001).

O próximo desenvolvimento significativo das redes neurais foi dado por Hebb em 1949, que apresentou pela primeira vez uma regra de aprendizagem para modificação sináptica. Este autor propôs que a conectividade do cérebro é continuamente modificada conforme um organismo que vai aprendendo tarefas funcionais diferentes e que agrupamentos neurais são criados por tais modificações (HAYKIN, 2001).

Hebb serviu de inspiração para que muitos outros pesquisadores perseguissem a mesma idéia. E embora muito tenha sido estudado e publicado nos anos que seguiram (1940-1950), estes serviram mais como base para desenvolvimento posterior que para o próprio desenvolvimento.

Uma nova abordagem para o problema de reconhecimento de padrões foi introduzida por Frank Rosenblatt em 1958 no seu trabalho sobre o *perceptron*, um método inovador de aprendizagem supervisionada, dando origem aos modelos baseados no aprendizado por correção de erros.

Após Rosenblatt, Bernard Widrow (1960-1962), com a ajuda de estudantes, desenvolveram um novo tipo de elemento de processamento de redes neurais chamado de *Adaline* (*ADaptive LINear Element*) e mais tarde a sua generalização multidimensional *Madaline* (*multiple-adaline*), equipado com uma poderosa lei de aprendizado, que diferente do *perceptron* ainda permanece em uso. A contribuição importante deste trabalho foi a invenção de um princípio de treinamento super poderoso para as redes *adalines* conhecido como a *Regra Delta* ou algoritmo de aprendizado LMS (*Least Mean Square*). Além disto, Widrow também fundou a primeira companhia de hardware de neurocomputadores e componentes (KOVÁCS, 2002).

Os modelos *perceptron* e *Adaline* foram muito criticados por Minsky e Papert (1969), pois apontaram a impossibilidade destes modelos serem capazes de solucionarem problemas que não fossem linearmente separáveis. Então como ficou comprovado o baixo

poder computacional destes modelos, as pesquisas nesta área ficaram estagnadas até meados dos anos 80.

Nos anos 80, John Hopfield se interessou pela neurocomputação e criou um modelo de rede com conexões recorrentes, onde o aprendizado era não-supervisionado. Apenas em 1986 Rummelhart, Hinton e Williams inventaram o poderoso algoritmo de retropropagação do erro para redes de neurônios de múltiplas camadas, resolvendo assim uma das limitações fundamentais para o treinamento de redes complexas (KOVÁCS, 2002).

Surgiram na mesma época as redes de Kohonen ou Mapas Auto-Organizáveis (*Self-Organizing Maps – SOM*), modelos neurocomputacionais desenvolvidos pelo professor Teuvo Kohonen (Universidade Técnica de Helsínquia – Finlândia), sendo utilizadas principalmente como análise de agrupamentos e apropriadas para tratar dados envolvendo comportamentos não lineares.

Na década de oitenta também se destacaram Carpenter e Grossberg, pois estabeleceram um novo princípio de auto-organização conhecido como teoria da ressonância adaptativa (*ART-Adaptive Resonance Theory*), a teoria envolve uma camada de reconhecimento de baixo para cima e uma camada generativa de cima para baixo (princípio de projeções para frente/para trás) (HAYKIN, 2001).

A partir da década de 90 houve um aumento considerável em relação às inúmeras aplicações e modelos de redes neurais, consolidando assim a importância destas tanto no estudo da inteligência artificial quanto nas mais diversas áreas.

2.2.2 Neurônio biológico

A idéia dos neurônios como constituintes estruturais do cérebro foi introduzida por Ramón y Cajál (1911). Estima-se que existam aproximadamente 10 bilhões de neurônios no cérebro humano e 60 trilhões de sinapses ou conexões, logo o cérebro é considerado uma estrutura extremamente eficiente. Os neurônios estão conectados uns aos outros através de sinapses, e juntos formam uma grande rede, chamada Rede Neural. A sinapse química é uma conexão simples que pode impor ao neurônio receptivo excitação ou inibição, mas não ambas (HAYKIN, 2001).

O sistema nervoso humano pode ser visto como um sistema de três estágios, onde o centro é o cérebro (representado pela rede neural), que recebe continuamente informação,

percebe-a e toma decisões apropriadas. Os receptores, que convertem os estímulos do corpo humano ou do ambiente externo em impulsos elétricos que transmitem informação para a rede neural (cérebro), e os atuadores, que convertem impulsos elétricos gerados pela rede neural em respostas discerníveis como saídas do sistema. Havendo assim, em todo o sistema, a presença de realimentação (HAYKIN, 2001).

Verifica-se na figura 5, que o neurônio biológico é delimitado por uma fina membrana celular, e a partir do corpo celular ou soma, que é o centro dos processos metabólicos da célula nervosa, projetam-se extensões filamentosas, os dentritos, e o axônio. E o sentido de propagação do impulso em um neurônio segue a seqüência: dentrito => corpo celular => axônio (KOVÁCS, 2002).

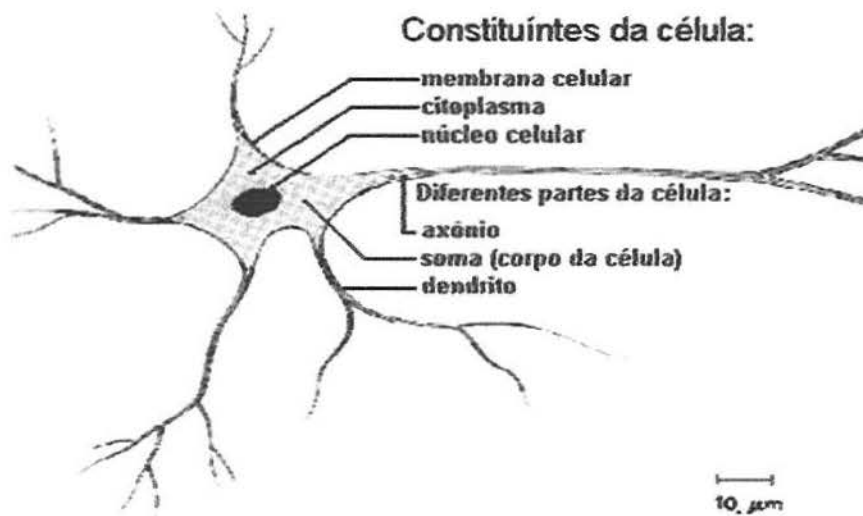


Figura 5. Neurônio biológico

Fonte: Ferraudo, 2006

Então, o neurônio biológico pode ser visto como um dispositivo computacional do sistema nervoso, composto de muitas entradas e uma saída. As entradas são formadas através das conexões sinápticas que conectam os dentritos aos axônios de outras células nervosas. Os sinais que chegam por estes axônios são pulsos elétricos conhecidos como impulsos nervosos ou potenciais de ação e constituem a informação que o neurônio processa para produzir como saída um impulso nervoso no seu axônio (KOVÁCS, 2002).

As redes neurais naturais não imprimem sinais negativos, pois sua ativação é avaliada pela frequência com que emite pulsos, frequência esta com pulsos contínuos e

positivos. Já as redes neurais artificiais podem vir a transmitir sinais negativos e apresentam uniformidade, ao contrário das redes naturais, que apresentam uniformidade somente em alguns pontos do organismo.

2.2.3 Neurônio artificial

O neurônio artificial é uma estrutura lógico-matemática que tenta simular o comportamento, a forma e as funções de um neurônio biológico. As entradas substituem os dendritos e os pesos simulam as sinapses ligando as entradas ao corpo celular. Os estímulos captados pelas entradas são processados pela função soma, e o limiar de disparo do neurônio biológico é substituído pela função de transferência. Observa-se pela figura 6, um comparativo entre o neurônio artificial e o natural (FERRAUDO, 2006).

A figura 6 mostra os elementos que compõem um neurônio artificial, ou seja: o vetor X (estímulos de entradas no neurônio); o vetor W (pesos ou ligações sinápticas); o vetor Y as saídas segundo o valor de ativação alcançado pelo neurônio; o θ_k (*bias*- termo independente), que é somente conectado aos neurônios de saída, ou seja, é tomado tipicamente de fora do corpo do neurônio e conectado a ele usando uma entrada adicional que permanece fixa (por exemplo: +1).

O sinal total que entra no corpo de processamento de um neurônio artificial é geralmente chamado de *net*, cujo valor é calculado através da simples multiplicação do sinal que chega ao neurônio pelo peso da sinapse em questão, ou seja, *net* = Sinal de entrada (*input*) x Peso, podendo gerar um sinal tanto positivo (excitatório) quanto negativo (inibitório). E como os neurônios possuem um grande número de dendritos e, portanto, podem estabelecer simultaneamente várias sinapses com outros neurônios da rede neural, o cálculo do *net* deve ser generalizado para n entradas, isto é, representa a soma de todas as combinações lineares $\left(\sum_i^n W_i X_i\right)$ entre os pesos sinápticos (W_i) e as entradas (X_i) fornecidas ao neurônio (FERRAUDO, 2006).

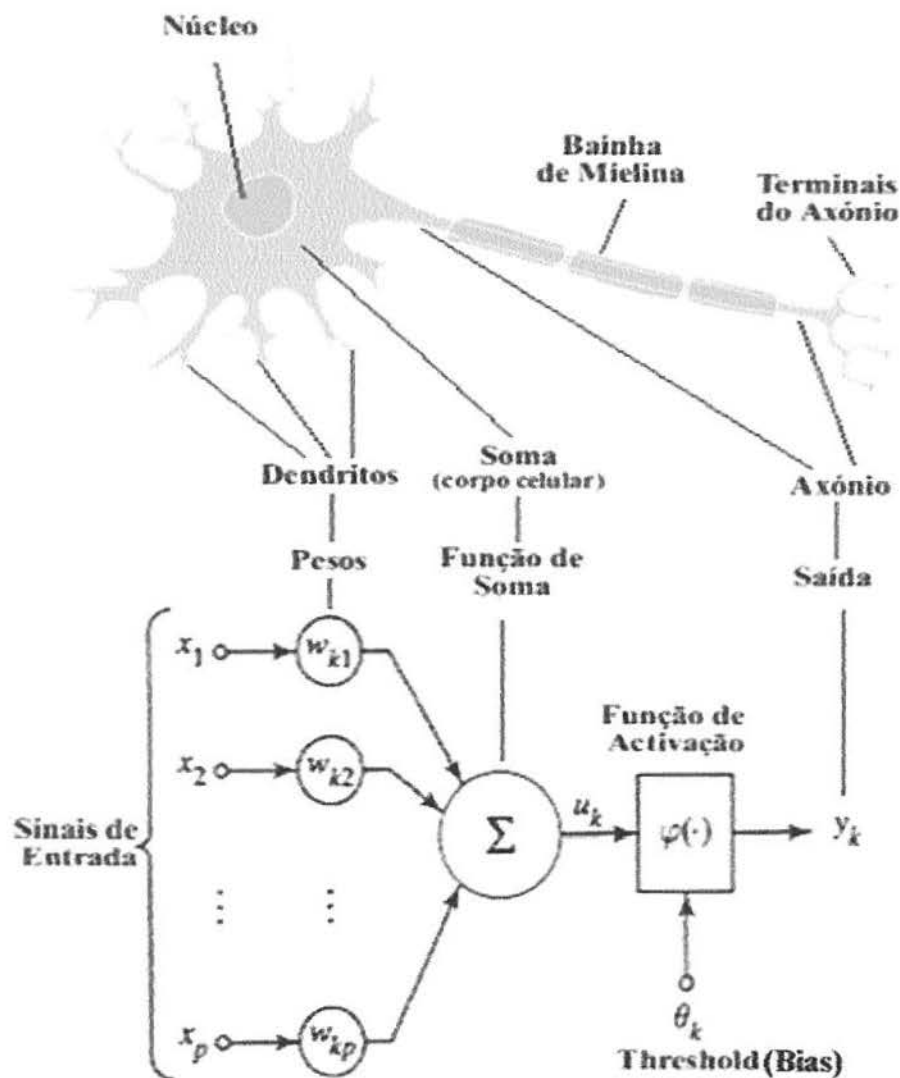


Figura 6. Neurônio biológico em contraponto com o neurônio artificial

Fonte: <http://bioinformatics.ath.cx/index.php?id=110>

Há também a *função de ativação (ou de transferência)*, que produz a saída do neurônio, cujo objetivo principal é introduzir uma não-linearidade no modelo. Pois sem esta função o sistema seria linear e o poder computacional, tanto do neurônio quanto da rede neural, seria prejudicado, impossibilitando assim a solução de vários problemas complexos.

As principais funções de ativação utilizadas são: (i) limiar; (ii) linear por partes; (iii) linear e (iv) sigmóide, que encontram-se respectivamente na figura 7(a) a 7(d).

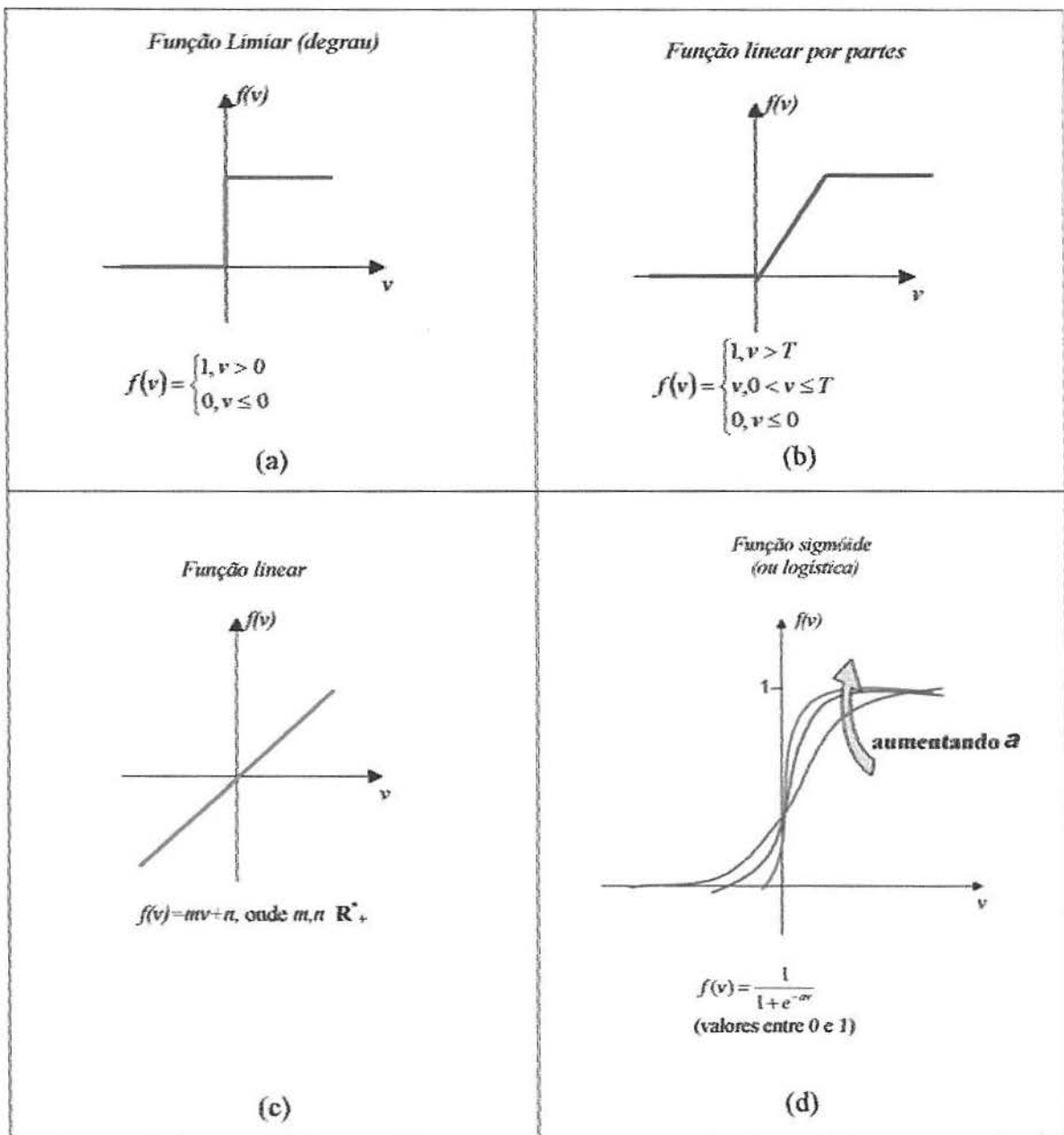


Figura 7. Principais funções de ativação

Fonte: Ferraudo, 2006

2.2.4 Rede Neural Artificial (RNA)

Uma camada é uma estrutura contendo uma quantidade de neurônios e os dados de entrada através de funções matemáticas e fornece o resultado (saída da rede). As entradas e saídas podem ser denominadas, respectivamente, de camada de entrada e camada de saída, enquanto que as camadas internas são ditas camadas intermediárias ou camadas ocultas (FERRAUDO, 2006). Então, Redes Neurais Artificiais (RNAs) são sistemas paralelos compostos por unidades de processamentos (neurônios) que processam determinadas funções matemáticas. Tais unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, e na maioria dos modelos estas conexões estão associadas a pesos, cujas ponderações se referem ao aprendizado do conhecimento representado nas entradas de cada neurônio.

O uso de redes neurais proporciona algumas propriedades úteis:

- **Não-linearidade:** é uma propriedade muito importante, particularmente se o mecanismo físico responsável pela geração do sinal de entrada for não-linear.
- **Generalização:** as redes podem aprender através de exemplos e apresentar respostas coerentes para entradas não vistas no treinamento.
- **Adaptabilidade:** as redes se adaptam a novos ambientes através de alterações nos pesos sinápticos. Também podem ser projetadas para alterarem seus pesos em tempo real ou para operarem em ambientes que variem com o tempo.
- **Informação Contextual:** processam as informações contextuais de forma natural, uma vez que o processamento de um neurônio é afetado pelo processamento de outros neurônios da rede.
- **Uniformidade:** A notação pode ser utilizada em diferentes domínios de aplicações, pois todas as arquiteturas de rede são formadas por neurônios. É possível utilizar um mesmo algoritmo de aprendizagem e teorias em outras aplicações.

2.2.5 Arquitetura de uma Rede Neural

O número de camadas e o número de neurônios ótimo são encontrados empiricamente, pois ainda não existe uma metodologia capaz de fornecer uma quantidade de camadas e neurônios ótima, assim muitas arquiteturas devem ser treinadas num mesmo

conjunto de dados até que seja encontrada a que contenha um número de camadas e neurônios que permita uma boa generalização. Mas se estes números forem além do suficiente pode levar a rede a memorizar os dados de treinamento (*overfitting*), e se ocorrer o contrário, pode forçar a rede a gastar tempo a mais na tentativa de otimizar uma representação.

Então, a propriedade primordial para uma rede neural é a sua habilidade de aprender a partir de seu ambiente e de melhorar o seu desempenho através da aprendizagem (HAYKIN, 2001). Ainda segundo Haykin (2001 p.75), a aprendizagem no contexto de redes neurais é “um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre”.

Existem vários tipos de treinamento de redes, que podem ser separados em dois grandes grupos: Aprendizado Supervisionado e Aprendizado Não Supervisionado. E nada impede que alguma camada possa trabalhar com o método supervisionado enquanto que outra camada da mesma rede trabalhe com o método não supervisionado (FERRAUDO, 2006).

- Aprendizado Supervisionado: Neste tipo de aprendizado a saída é comparada com a desejada, e ajustes pertinentes são realizados nos pesos sinápticos de modo a reduzir esta distância. O algoritmo de aprendizado supervisionado mais utilizado é a regra delta (correção de erros) e a sua generalização para rede de múltiplas camadas, o poderoso algoritmo de *backpropagation* (retropropagação do erro).

- Aprendizado Não Supervisionado (Auto-Organização): neste tipo de aprendizado não existe uma saída desejada para comparação. Pelos padrões contidos nas entradas, as unidades experimentais são classificadas automaticamente em grupos. Esse tipo de rede utiliza os neurônios como classificadores e as entradas são os elementos para a classificação. Envolve competição e colaboração entre os neurônios da rede. Uma das vantagens está na ausência da supervisão (saída desejada), mas a desvantagem está no conjunto de dados para treinamento, que deve ser redundante para que a rede consiga abstrair características em seu treinamento.

- Aprendizado Híbrido: neste aprendizado há uma fase não-supervisionada seguida de uma fase supervisionada, um exemplo de rede que pode utilizar este tipo de aprendizagem é a rede com Função Base Radial (RBF).

2.2.6 Rede Perceptron de Múltiplas Camadas (MLP)

No final da década de 1950, Rosenblatt na Universidade de Cornell, deu prosseguimento às idéias de McCulloch, criando uma rede de múltiplos neurônios do tipo discriminadores lineares. Estas redes são geralmente chamadas de *perceptrons de múltiplas camadas (MLP- Multilayer perceptron)*, e conseguem obter soluções para problemas linearmente não separáveis. Um *perceptron* é uma rede com uma topologia apresentada na figura 8, com neurônios dispostos em várias camadas. Os neurônios que recebem diretamente as entradas da rede constituem a camada de entrada, e os demais que recebem como entradas as saídas daqueles da camada de entrada constituem as camadas ocultas, até a camada final que é a camada de saída (KOVÁCS, 2002).

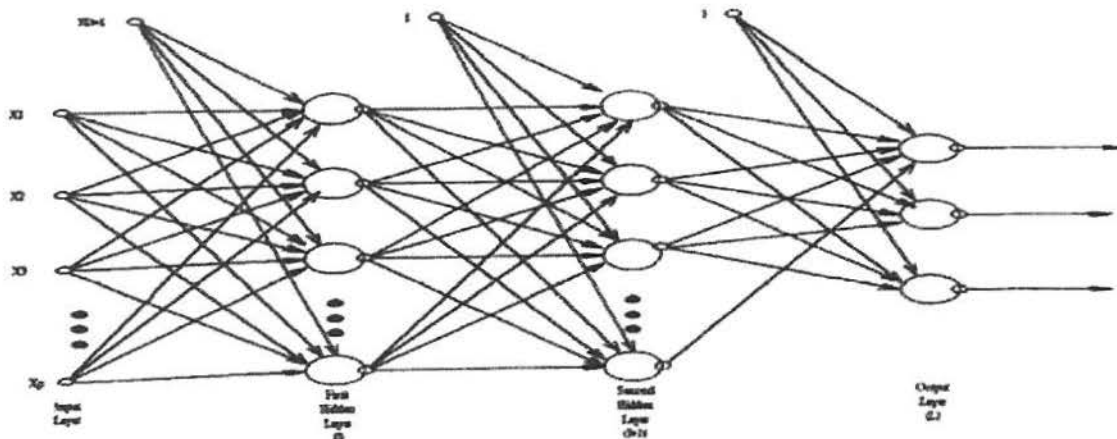


Figura 8. Rede Perceptron Múltipla Camada

Fonte: Leger et al, 1998

Esta rede utiliza o treinamento supervisionado através do algoritmo *backpropagation*, que é baseado na regra de aprendizagem por correção de erro. Ou seja, a aprendizagem por retropropagação de erro consiste de dois passos através das diferentes camadas da rede: um passo para frente, a propagação, e um passo para trás, a retropropagação. Durante o passo de propagação, os pesos sinápticos da rede são todos

fixos, enquanto que na retropropagação os pesos (w) são todos ajustados de acordo com uma regra de correção de erro. Em resumo, a resposta real é subtraída da resposta alvo e produz um sinal de erro que é propagado para trás da rede, contra a direção das conexões sinápticas. Os pesos são ajustados de modo que a resposta real da rede se torne a mais próxima da resposta desejada (SELAU, 2000).

Então, o algoritmo de aprendizado por *backpropagation*, que é utilizado para redes de múltiplas camadas, compara o valor predito na saída da rede com o valor desejado da classe da amostra. A partir desta comparação é calculado o erro quadrático médio que é retropropagado pela rede, no sentido da camada de saída até a camada de entrada, modificando os pesos sinápticos, de forma a minimizar o erro. Geralmente os pesos convergem para valores que tornam este aceitável e o treinamento é finalizado. Na figura 9 há uma representação da retropropagação para uma rede com três camadas.

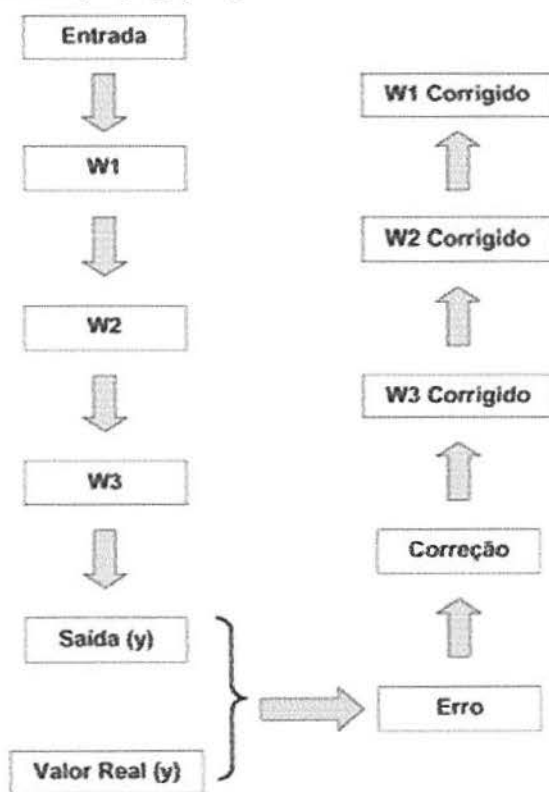


Figura 9. Representação da retropropagação em uma RNA

Fonte: Mello et al, 2001

2.2.7 Rede RBF – Funções Base Radial

De acordo com o ponto de vista de que o projeto de uma rede neural pode ser um problema de ajuste de curva, aprender é equivalente a encontrar uma superfície, em um espaço multidimensional que forneça o melhor ajuste para os dados de treinamento, com o critério de “melhor ajuste” sendo medido em um sentido estatístico. Generalização é equivalente ao uso desta superfície multidimensional para interpolar os dados de teste. Então, tal ponto de vista é a motivação por trás do método das funções de base radial, onde as unidades ocultas fornecem um conjunto de “funções” que constituem uma “base” arbitrária para os vetores de entrada, quando eles são expandidos sobre o espaço oculto: estas funções são chamadas de funções de base radial (HAYKIN, 2001).

As redes neurais RBF possuem apenas uma camada intermediária definida por funções base radial (uma por neurônio) com atribuição, principalmente, dos centros Gaussianos por decomposição do espaço de dados em subespaços cujas predições neste caso possuem menor erro que as redes MLP. Na figura 10 pode-se observar uma rede RBF em sua forma mais básica, ou seja, envolve três camadas. A camada de entrada é constituída por nós de fonte que conectam a rede ao seu ambiente. A segunda camada, a única camada oculta da rede, aplica uma transformação não-linear do espaço de entrada para o espaço oculto; na maioria das aplicações o espaço oculto é de alta dimensionalidade, pois quanto mais alta for a dimensão do espaço oculto mais precisa será a aproximação. E a camada de saída é linear, fornecendo a resposta da rede ao padrão de ativação aplicado à camada de entrada (HAYKIN, 2001).

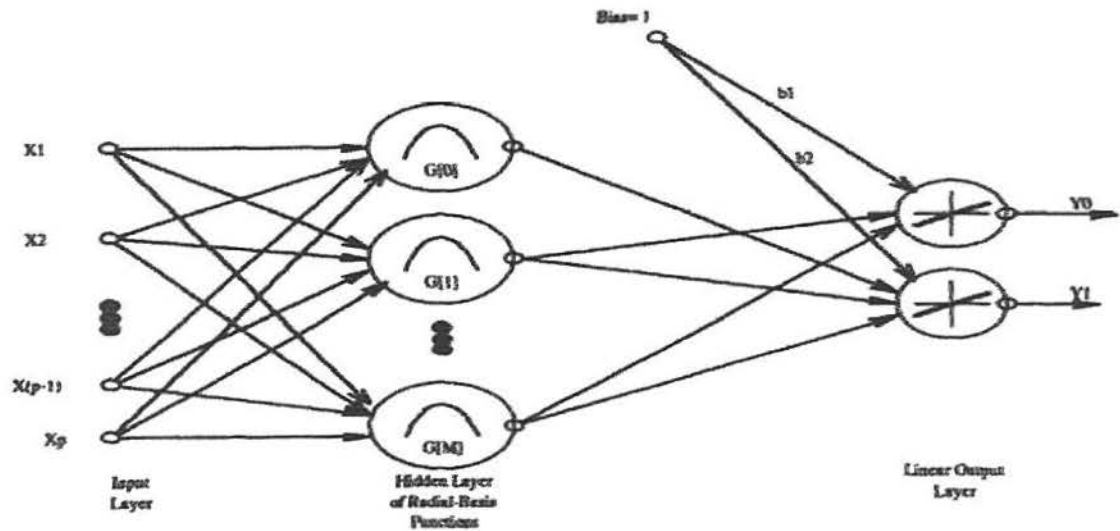


Figura 10. Rede Neural RBF – Função Base Radial

Fonte: Leger et al, 1998

Como o principal objetivo da rede é aproximar uma função do mapeamento entre entrada e saída, o uso da RBF consiste na determinação de uma função que tenha a forma da equação (20).

A camada de saída é usada como combinação linear da saída dos nós ocultos, e é encontrada de acordo com a equação (20),

$$y_i = \sum_{m=0}^M w_{mi} G_m + b_i, \quad m=0,1,\dots,M \quad (20)$$

Onde: M é o número de nós ou centros da camada oculta, w_{mi} é o peso entre o centro m e o neurônio de saída i , G_m é a saída do neurônio padrão m e b_i representa o bias da saída i . Geralmente, funções não-normalizadas Gaussianas centradas em c_i são usadas. São definidas conforme a equação (21):

$$G(\|x - c_i\|) = \exp\left(\frac{-\|x - c_i\|^2}{\sigma_i^2}\right) \quad (21)$$

Onde $\|x - c_i\|$ é a distância euclidiana entre o vetor de entrada p dimensional x e o i -ésimo centro, c_i . σ_i é a amplitude ou desvio padrão na função Gaussiana centrada em c_i .

As funções Gaussianas evitam que as RBFs produzam um valor alto de saída para observações muito distantes de seus centros. Isto ajuda a diminuir a extrapolação do erro

que ocorre quando os dados de teste caem além do intervalo do conjunto de treinamento. Pode-se verificar através da equação (21) que a função Gaussiana tem seu valor máximo de 1 quando $\|x - c_i\|$ é igual a zero e tem valor decrescente quando $\|x - c_i\|$ tende ao infinito (BALESTRASSI, 2000).

Há diferentes estratégias de aprendizado que podem ser usadas para determinar os pesos sinápticos deste tipo de rede. Estas estratégias dependem de quais parâmetros livres da rede estão sendo ajustados. Os parâmetros da rede são os números de centros, localização dos centros (c_i) e a amplitude dos centros (σ_i). A estratégia de aprendizagem simples é selecionar aleatoriamente os centros da RBF para o treinamento dos dados e assumi-los como fixos. As funções conseguem ser normalizadas pelas amplitudes através da equação (22) (LEGER et al, 1998).

$$\sigma = \frac{d}{\sqrt{2M}} \quad (22)$$

onde, d : distância máxima entre os centros escolhidos e M : número de centros.

A fórmula (22) assegura que as funções de base radial individuais não sejam pontiagudas demais ou planas demais, pois estas duas condições extremas devem ser evitadas.

Por fim, os pesos da camada de saída (w_{it}) podem ser calculados pela fórmula (23).

$$\mathbf{w} = \mathbf{G}^+ \mathbf{d} \quad (23)$$

Onde: \mathbf{d} é a matriz dos vetores da saída e \mathbf{G}^+ é a pseudoinversa da matriz \mathbf{G}

As redes RBF convergem rapidamente para um erro mínimo o que diminui em muito o tempo de processamento quando comparado às redes MLP, apesar das MLP serem capazes de generalizar em regiões de espaço de entradas onde há poucos dados de treinamento (LEGER et al, 1998). E outra grande diferença entre a rede RBF e a rede MLP, é que aquela consegue buscar um mínimo local e já esta consegue apenas um mínimo global mostrando assim que a rede RBF é mais confiável (HAYKIN, 2001).

3. ANÁLISES

3.1 Metodologia a ser aplicada para análise

Primeiramente é gerado um conjunto de dados que servirá de referência, que neste caso escolheu-se arbitrariamente o modelo AR (2), e avalia-se o número de pontos fora de controle destes dados através do gráfico CUSUM, com parâmetros que permitem uma equivalência ao gráfico de Shewhart em termos de ARL, neste caso ($h = 4,77$ e $k = 0,5$). Após, são simuladas 400 séries de tamanho 50 períodos, com o mesmo comportamento do conjunto de referência.

Após, será aplicado o gráfico CUSUM para cada uma das 400 séries simuladas e será criada a estatística d_1 em cada período de tempo t . Esta estatística consiste no número de pontos classificados como diferentes da série referência.

Também será modelado, para cada uma das 400 séries, uma rede neural artificial. Neste caso a rede neural a ser usada é uma RBF, pois em seu estudo Leger et al (1998) mostraram que a rede neural RBF detectou um menor número de alarmes falsos do que a rede neural MLP, indicando que a primeira é mais sensível para verificar o controle de um processo. Mas no presente estudo não será avaliado os alarmes falsos como no estudo de Leger et al (1998), será verificado a capacidade de cada um dos dois métodos (CUSUM e RNA) em conseguir detectar o comportamento da série referência.

Após a modelagem da RNA será aplicado o gráfico de Shewhart para medidas individuais obtendo-se a estatística d_2 em cada período de tempo t , e esta estatística é o número de pontos classificados como diferentes da série referência.

Por fim, será aplicado um teste - *t-student* ou *Kolmogorov-Smirnov* - para comparar as médias das estatísticas d 's encontradas via método CUSUM e via redes neurais artificiais. Tal análise, que também será realizada no software R v. 2.6.0 como as simulações, visa verificar qual dos dois métodos foi mais eficiente em detectar o comportamento da série base.

A figura 11 apresenta o procedimento utilizado para facilitar a operacionalização da metodologia e viabilizar a comparação do gráfico CUSUM e uma RBF no software R, e no anexo I encontram-se as rotinas computacionais utilizadas neste trabalho.

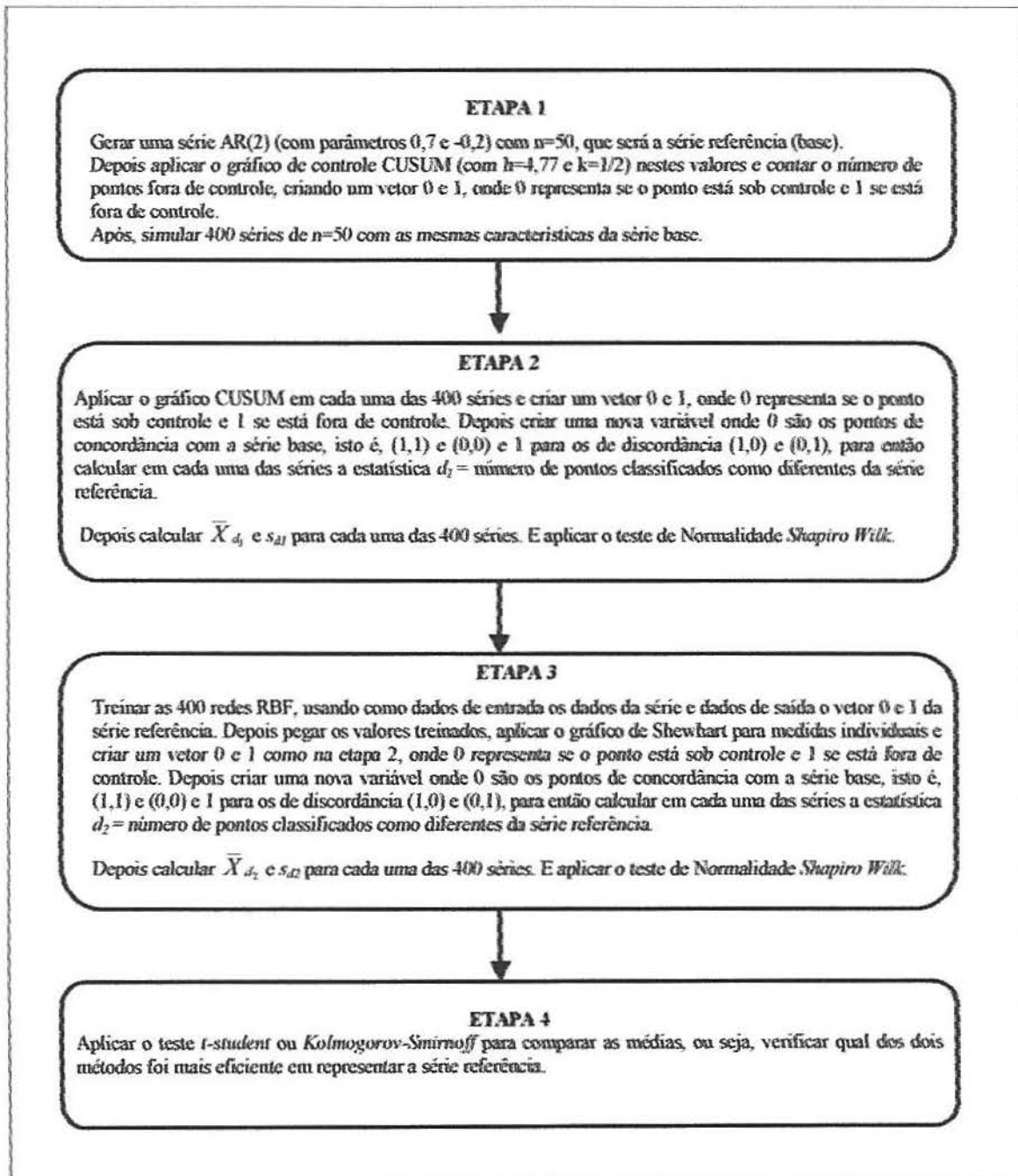


Figura 11. Metodologia proposta para realizar a comparação do CUSUM e da RBF

3.2 ETAPA 1 - Geração dos dados

Os dados gerados possuem a forma de um modelo Auto-Regressivo(2), ou seja, AR(2): $Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + a_t$, com os parâmetros 0,7 e -0,2 e tamanho da série igual a 50. Para poder comparar os métodos, os dados gerados passam a representar medidas “reais” de determinada característica de qualidade a ser estudada (parâmetro). Além disto, as 400 séries geradas, de tamanho 50, também serão um modelo AR(2) com os mesmos parâmetros da série de referência e estes serão as séries a serem utilizadas nas análises.

Após aplicou-se o gráfico de controle CUSUM Tabular (com parâmetros $h = 4,77$ e $k = 1/2$) na série referência criou-se um vetor 0 e 1, onde 0 representa se o ponto está sob controle e 1 se o ponto está fora de controle e depois contou-se o número de pontos fora de controle, neste caso encontrou-se 15 pontos fora de controle.. O critério de escolha dos valores dos parâmetros, ou seja, de $h = 4,77$ e $k = 1/2$, teve como base a tabela 2 do capítulo 2, que indica os valores para k e os correspondentes valores de h para um $ARL_0=370$, e estes valores de k e h foram calculados com a mesma confiabilidade do que a dos gráficos de Shewhart, assim o gráfico CUSUM possui propriedades semelhantes aos gráficos de Shewhart, permitindo assim a comparação de ambos. A figura 12 mostra o gráfico CUSUM Tabular da série base, onde há 15 pontos fora de controle detectados.

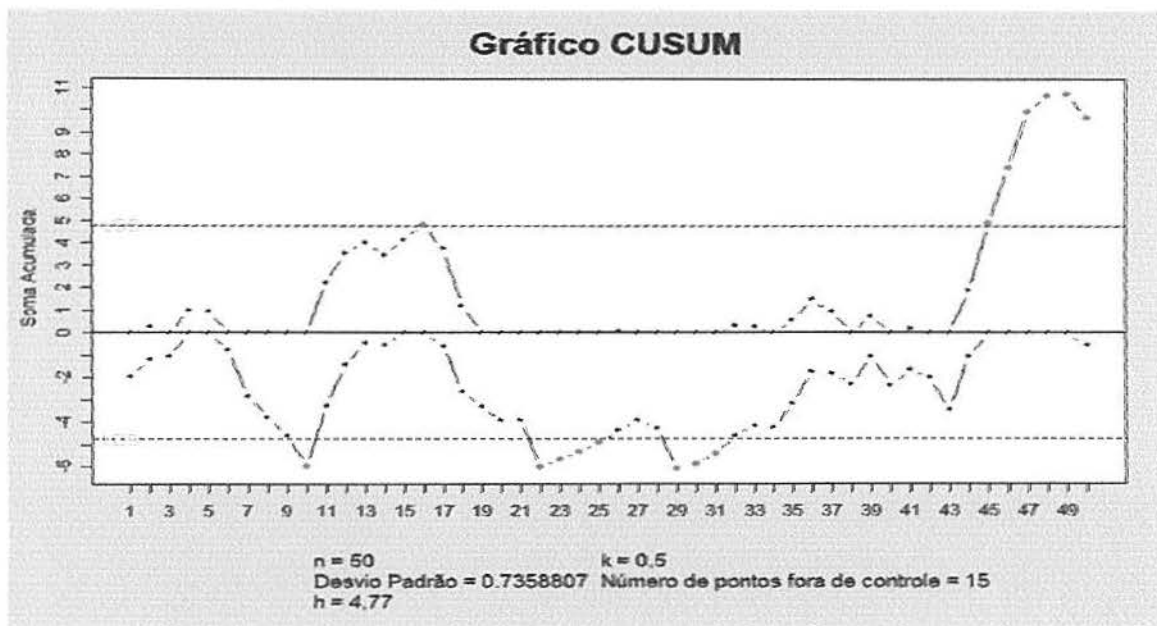


Figura 12. Gráfico CUSUM da série referência

3.3 ETAPA 2 - Gráfico CUSUM

Nesta etapa, aplicou-se o gráfico CUSUM Tabular (com $h = 4,77$ e $k = 1/2$) em cada uma das 400 séries simuladas e contou-se o número de pontos fora de controle criando um vetor 0 e 1 como na etapa 1. Após comparou-se o número de pontos classificados como diferentes da série referência, criando desta forma uma nova variável, onde 0 são os pontos de concordância com a série referência, isto é, os pontos (1,1) e (0,0) e 1 para os de discordância com a série referência, isto é, (1,0) e (0,1). Após calculou-se em cada uma destas séries uma estatística d_1 , ou seja, contou-se o número de pontos discordantes. Por fim, calculou-se \bar{X}_{dt} e s_{dt} , que estão mostrados na tabela 5. Posteriormente aplicou-se o teste de normalidade *Shapiro Wilk*, podendo-se perceber, pelos resultados da tabela 5, que as estatísticas d_j 's não possuem distribuição normal a 5% de significância, pois o p-valor é igual a 0,01562. Isto indica que não será possível testar as médias das estatísticas d_1 e d_2 utilizando o teste *t-student*, sendo adequado aplicar o teste não-paramétrico *Kolmogorov-Smirnov*. Através da figura 13 verifica-se o histograma da diferença d_1 , mostrando a não normalidade desta, já confirmada pelo teste de *Shapiro Wilk*.

Analisando os resultados da tabela 5, percebe-se que utilizando o gráfico CUSUM a média da estatística d_1 foi de 24,3, isto é, em média este método apresentou 24,3 pontos que discordaram da série referência e com desvio padrão de 5,17 pontos.

Assim, a média mostrada na tabela 5, será comparada com a média encontrada pelo método RBF, a fim de verificar se há equivalência nos métodos ou mostrar qual deles foi mais eficiente em conseguir detectar o comportamento da série referência.

Tabela 5. Resultados d_1

Média: 24,3
Variância Amostral: 26,72
Desvio Padrão Amostral: 5,17
Teste de normalidade <i>Shapiro Wilk</i> SW = 0,991, p-valor = 0,01562

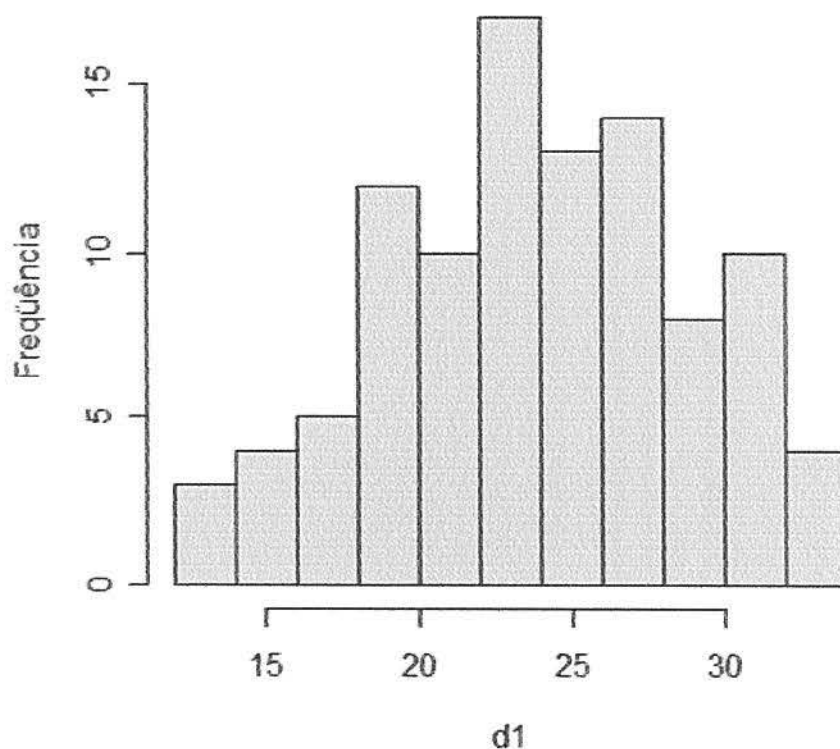


Figura 13. Histograma da diferença d_1

3.3 ETAPA 3 - Rede RBF

Sabe-se que os parâmetros livres da rede RBF são os números de centros, localização dos centros (c_i) e a amplitude dos centros (σ_i). Na fase de treinamento, o número máximo de neurônios, amplitude dos centros e erros são especificados pelo usuário. A função de treinamento inicia com um neurônio e calcula-se a soma de quadrados do erro, e assim mais neurônios vão sendo adicionados até que se chegue ao erro especificado ou o número máximo de centros tenha sido utilizado. E a localização dos centros também se baseia na soma de quadrados do erro mínima (LEGER et al, 1998).

Então, treinou-se 400 redes RBF usando como dados de entrada os dados de cada uma das 400 séries e como dados de saída o vetor 0 e 1 da série referência, onde 0 representa se o ponto está sob controle e 1 se o ponto está fora de controle. O tipo de treinamento foi supervisionado e para se chegar no número de neurônios ideal da camada oculta de cada série foi-se variando o número de neurônios (de 2 a 10 neurônios) e o número de iterações de treino (10 a 210, com pulos de 25 em 25), até a rede que obtivesse o

menor erro médio absoluto. Assim, não existe apenas uma rede, mas sim várias redes diferentes, o que se observa na tabela 6, mostrando a frequência de séries com determinada quantidade de neurônios na camada central e que possuem o menor erro médio absoluto.

Tabela 6. Número de Neurônios da camada oculta e a frequência de séries simuladas

Número neurônios da camada oculta	Frequência (número de séries)
2	103
3	53
4	45
5	32
6	27
7	29
8	35
9	32
10	44

Após, pegou-se os valores das redes que obtiveram o melhor ajuste para cada uma dessas 400 redes e aplicou-se o gráfico de controle de Shewhart para medidas individuais e contou-se o número de pontos fora de controle criando um vetor 0 e 1 como na etapa 1 e na etapa 2. Depois, computou-se em cada uma das séries a estatística d_2 , isto é, contou-se o número de pontos classificados como diferentes da série referência, criando desta forma uma nova variável, onde 0 são os pontos de concordância com a série referência, isto é, (1,1) e (0,0) e 1 para os de discordância com a série referência, isto é, (1,0) e (0,1). Calculou-se \bar{X}_{d_2} e s_{d_2} , que estão na tabela 7. Posteriormente aplicou-se o teste de normalidade *Shapiro Wilk* e nota-se, pelos resultados da tabela 6, que as estatísticas d_2 's não possuem distribuição normal a 5% de significância, pois o p-valor é menor do que $2,2e-16$. Indicando que é mais adequado testar as médias das estatísticas d_1 e d_2 utilizando um teste não-paramétrico, tal como o teste de *Kolmogorov-Smirnov*.

Através da figura 14 observa-se o histograma da diferença d_2 , mostrando a não normalidade desta, já confirmada pelo teste de *Shapiro Wilk*. E analisando os resultados da tabela 7, verifica-se que utilizando a rede RBF a média da estatística d_2 foi de 15,1, isto é, em média este método discordou em 15,1 pontos da série referência e com desvio padrão de 1,52 unidades.

Tabela 7. Resultados d_2

Média: 15,1

Variância Amostral: 2,30

Desvio Padrão Amostral: 1,52

Teste de normalidade *Shapiro Wilk* SW = 0,7427, *p*-valor < 2,2e-16

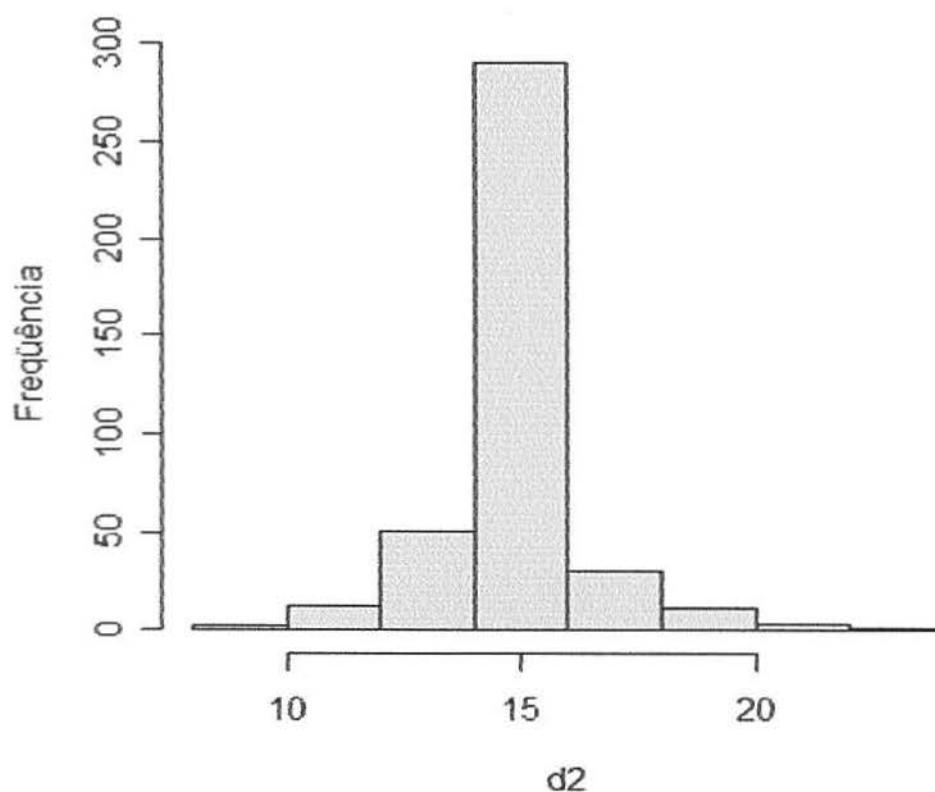


Figura 14. Histograma da diferença d_2

3.4 ETAPA 4 - Comparação do Gráfico CUSUM e da rede RBF

A última etapa consiste em aplicar o teste de *Kolmogorov-Sminorv* para verificar se há diferença significativa entre as médias das estatísticas d_1 e d_2 , verificando qual dos dois métodos foi mais eficiente em detectar o comportamento da série referência.

Na tabela 8, com um 1% de significância, rejeita-se a hipótese de igualdade de médias, assim pode-se dizer que há diferença significativa entre as médias das estatísticas d_1 e d_2 . Então, como a média do número de pontos discordantes em relação à série base é

menor no método RBF, o uso de redes neurais artificiais foi mais eficiente, conseguindo representar as séries mais realisticamente, ou seja, similarmente à série de referência, pois se afastou “pouco” da realidade, “concordando” mais com as classificações (fora ou sob controle) dos pontos da série base, logo este método conseguiu detectar melhor o comportamento da série referência. Já o método CUSUM se “afastou” mais da realidade, isto é, o número de pontos classificados como diferentes dos da série referência foi superior, se comparado com a RNA. E através da tabela 9, observam-se as médias das estatísticas. Portanto, neste estudo, a utilização de RNAs foi mais eficaz em mostrar a realidade do que a carta de controle CUSUM.

Tabela 8. Resultados Teste K-S

Teste de <i>Kolmogorov-Smirnov</i>	
$H_0 : \mu_{d_1} = \mu_{d_2}$	
$H_1 : \mu_{d_1} \neq \mu_{d_2}$	
$D = 0,835, p\text{-valor} < 2,2e-16$	

Assim, através deste trabalho, pode-se perceber uma maneira alternativa e eficaz para se controlar a qualidade nas empresas em geral, isto é, utilizando ferramentas da inteligência artificial ao invés de gráficos de controle estatísticos tradicionais.

Tabela 9. Médias das diferenças

Média de d_1	Média de d_2
24,3	15,1

4. CONSIDERAÇÕES FINAIS

Monitorar o processo produtivo de maneira eficaz é cada vez mais almejado pelas indústrias, uma vez que o mercado exige constantemente produtos e serviços com a mais alta qualidade. Este monitoramento geralmente é realizado pelo CEP (Controle Estatístico de Processo), podendo ser através dos gráficos de controle, tanto os de Shewhart quanto outros tipos mais refinados (CUSUM, EWMA, etc). Mas este monitoramento também pode ser realizado pelas redes neurais artificiais, que são excelentes no quesito reconhecimento de padrões.

Este trabalho foi desenvolvido com o intuito de comparar dois métodos para controlar certa característica de qualidade nas empresas, ou seja, utilizar gráfico de controle CUSUM e rede neural artificial RBF e compará-los em relação à eficiência na detecção do comportamento da série referência. Estes dois métodos, aparentemente tão distintos, empregados nesta monografia possuem inúmeros benefícios e particularidades, mas avaliou-se a capacidade de cada método em representar mais fidedignamente a série referência, isto é, resultar em uma menor média do número de pontos classificados como diferentes entre a série referência e as séries modeladas por cada método.

Através do resultado deste estudo simulado observou-se que o método RBF foi o que obteve a menor média do número de pontos classificados como diferentes em relação à série base e que diferiu significativamente, pelo teste de *Kolmogorov-Smirnov* a 1% de significância, do método CUSUM. Assim, a rede RBF foi a que representou a série referência mais fidedignamente, sendo assim, ao nível de significância estabelecido, o método mais eficiente para representar o comportamento destes dados simulados, se comparado ao CUSUM.

Com este estudo ficou comprovado que as redes neurais artificiais, que são uma maneira alternativa, são mais eficazes para monitorar a variabilidade nos processos produtivos. Além disto, possuem inúmeros benefícios, como a capacidade de aprenderem através de exemplos e de generalizarem as informações. Mostrando aos leitores como a inteligência artificial pode ser aplicada com sucesso inclusive no controle estatístico da qualidade, incentivando assim a estes virem a desenvolver trabalhos futuros utilizando-as inclusive em outras áreas.

Para possíveis trabalhos futuros sugere-se:

- Comparar redes neurais artificiais RBF e MLP;
- Simular uma série de tamanho maior;
- Fazer uma aplicação com dados reais.

5. REFERÊNCIAS

- ALVES, C. C. *Gráficos de controle CUSUM: um enfoque dinâmico para a análise estatística de processos*. Dissertação de Mestrado em Engenharia de Produção, UFSC, Florianópolis, 2003.
- BALESTRASSI, P. P. *Identificação de padrões em gráficos de controle estatístico de processos, em tempo real, utilizando séries temporais e redes neurais artificiais*. Tese de Doutorado em Engenharia de Produção, UFSC, Florianópolis, 2000.
- BESTERFIELD, Dale H. *Quality Control*. Prentice-Hall, INC, 1979.
- FERRAUDO, A. S. *Introdução às Redes Neurais Artificiais*, 2006.
- GIL, A.C. *Métodos e técnicas de pesquisa social*. Editora Atlas, 1999.
- HAYKIN, Simon. *Redes Neurais: Princípios e Práticas*. 2ª edição. Editora Bookman, 2001.
- HAWKINS, D. M. A CUSUM for a Scale Parameter. *Journal of Quality Technology*, pp. 228-231, 1981.
- HAWKINS, D. M. Cumulative Sum Control Charting: An Underutilized SPC Tool. *Quality Engineering*, pp. 463-477, 1993a.
- KOVÁCS, Z. L. *Redes Neurais Artificiais: Fundamentos e Aplicações*. 3ª edição. Editora Livraria da Física, 2002.
- LAU, C. *Neural Networks: Theoretical Foundations and Analysis*. IEEE PRESS, 1991.
- LEGER, R. P.; GARLAND, Wm. J.; POEHLMAN, W. F. S. Fault detection and diagnosis using statistical control charts and artificial neural networks. *Artificial Intelligence in Engineering*, vol. 12, pp 35-47, 1998.
- MELLO, C.; DE CERQUEIRA, E. O.; DE ANDRADE, J. C.; POPPI, R. J. Rede Neurais e suas aplicações em calibração multivariada. *Química Nova*, vol.24, nº 6, São Paulo, 2001. Disponível na web em http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-40422001000600025. Acesso: 27 de setembro de 2007.

MONTGOMERY, D. C. *Introduction to Statistical Quality Control*. 4th Edition, New York: John Wiley, 2000.

SAMOHYL, R. W., ALVES, C. C. *O monitoramento de processos industriais via gráficos de Controle CuSum*, 2006. Disponível na web em <http://www.qualimetria.ufsc.br/artigos.htm>. Acesso: 06 de setembro de 2007.

SELAU, L. P. R. *Redes Neurais no contexto Estatístico*. Monografia (Graduação em Estatística). Instituto de Matemática, Departamento de Estatística, UFRGS, Porto Alegre, 2000.

SMITH, A. E. X-bar and R Control Chart Interpretation using Neural Network computing. *International Journal of production Research*, vol. 32, nº 2, pp 309-320, 1994.

VARGAS, V. C. C., LOPES, L. F. D., SOUZA, A. M. Comparative study of the performance of the CuSum and EWMA control charts. *Computers & Industrial Engineering*, vol. 46, pp 707-724, 2004.

WERKEMA, M. C. C. *Ferramentas Estatísticas Básicas para o Gerenciamento de Processos*. Volume 2. Belo Horizonte, MG: Fundação Christiano Ottoni, Escola de Engenharia da UFMG, 1995.

WERNER, L. *Controle Estatístico de Qualidade*, Série B nº 32, Instituto de Matemática, Departamento de Estatística da UFRGS. Porto Alegre, Março 1996.

7. ANEXO 1

Abaixo seguem as rotinas das análises realizadas neste trabalho com o auxílio do software R v 2.6.0.

```
# Matrix Para Redes Neurais
# m2.rn(dados,na,nt,lm,diff)
#
# na          = Vetor com as entradas (valores sempre positivos, ex. 2
#              significa 2 dados atrás, padrão = 1).
# nt          = Número de entradas reservadas para checar (padrão = 10)
# lm          = Lógico. Cria uma coluna com os valores de regressão linear
#              (Padrão=FALSE)
#
# Exemplo:   m2.rn(AirPassengers,n=24,...)

m2.rn = function(dados,classe=ref$scusum.serie,n=1,nt=10,lm=FALSE,...) {
  out = list()
  t <- length(dados) - nt
  out$inp.x <- cbind(dados[1:t])
  out$inp.y <- cbind(classe[1:t])
  out$check.x <- cbind(dados[(1+t):(t+nt)])
  out$check.y <- cbind(classe[(1+t):(t+nt)])
  return(out)
}
```



```

# Redes RBF - Retorna a rede
#
# rede(dados,neurons,it...)
#
# dados          = Saída do m2.m
# neurons        = Número de neuronios na camada central
# it             = Número máximo de iterações (Padrão: 2/3 da quantidade de dados)

rede <- function(dados,neurons,it=dados*2/3,...) {
  if (neurons < 2) { stop("O Número de neuronios deve ser maior que 1")
  }

  data<-
  rbfttrain(dados$inp.x,neurons,dados$inp.y,sigma=NaN,visual=FALSE,maxit=it)
  return(data)
}

# Cusum - Retorna o numero de termos fora dos limites
#
# kusum(dados)
#
# dados          = dados
# h              = h (Padrão = 4.77)
# k              = k (Padrão = 1/2)

```

```

kusum <- function(dados,h=4.77,k=1/2,plot=0) {
  cusum <- qcc(dados,type="xbar.one",plot=plot)
  cusum <- cusum(cusum, decision.int=h, se.shift=k)
  cusum <- cusum$cusum
  wow = rep(0,length(dados))
  wow[cusum$pos > h] <- 1
  wow[cusum$neg < -h] <- wow[cusum$neg < -h] + 1
  out = list()
  out$soma <- sum(wow)
  wow[wow > 1] <- 1
  out$serie <- wow
  return(out)
}

```

Medidas de Acurácea

med.a(dados,valores)

#

dados = Datos observados

valores = Datos obtidos

```

med.a <- function(dados,valores,...) {

```

```

  out = vector()

```

```

  valores = matrix(valores,nrow=length(dados))

```

```

for (i in 1:ncol(valores)) {
  coluna = valores[,i] - dados
  em <- mean(coluna)
  eam <- mean(abs(coluna))
  eqm <- mean(coluna^2)
  epm <- mean(coluna/dados)
  epam <- mean(abs(coluna/dados))
  out = cbind(out,c(em,eam,eqm,epm,epam))
}
rownames(out) <- c("EM", "EAM", "EQM", "EPM", "EPAM")
return(out)
}

# Número de Neuronios - Retorna a melhor rede e as entradas usadas
#
# nneuron(dados,entradas,it,redes...)
#
# dados          = Saída do m2.m
# neurons        = Vetor com a quantidade de neuronios a ser testada
# it             = Iterações de treino (Padrão = 2/3 da quantidade de dados)
# redes          = Quantidades de redes a serem feitas com cada neuronio
(Padrão = 10)

nneurons <-

```

```

function(dados,neurons,it=as.integer(length(inp$inp.x)*2/3),redes=10,...) {
  if (min(neurons) < 2) { stop("Número de neuronios deve ser maior que
  1") }
  redes = max(redes)
  ajust <- vector()
  prevs <- vector()
  redess <- list()
  for (i in neurons) {
    for (j in 1:redes) {
      rede <- rede(dados,i,it)
      redess[[length(redess)+1]] <- rede
      prevs <-
      cbind(prevs,rbf(dados$check.x,rede$weight,rede$dist,rede$neurons,rede$sigma))
    }
  }
  ajust <- med.a(dados$check.y,prevs)
  out = list()
  min = which.min(ajust[2,])
  out$rede = redess[[min]]
  out$ajust = ajust[,min]
  out$inp = dados
  out$it = it
  return(out)
}

```

```

# Previsão

# prev(dados)

#

# dados          = Saida do nneurons

#

prev <- function(dados) {
  inp <- dados$inp
  rede <- dados$rede

  rbf(rbind(inp$inp.x,inp$check.x),rede$weight,rede$dist,rede$neurons,rede$sigma)
}

# cmc(dados,neurons,ent,it,redes,...)

#

# dados          = Valores observados

# neurons        = Vetor com o número de neuronios a serem testados

# ent            = Uma lista com as entradas a serem testadas. Cada item deverá
                 ter pelo menos 2 termos, onde o primeiro é
                 #          um valor lógico 0 ou 1 (falso e verdadeiro, respectivamente), que
                 significa se terá (verdadeiro) ou não
                 #          terá (falso) a linha de tendencia, e o restando o número das
                 entradas (seguindo a explicação do comando

#          m2.rn

```

```

# it          = Vetor com as iterações a serem testadas
# redes      = Número de redes a serem testadas para cada combinação de N e
it

cmc <- function(dados,neurons,ent,it,redes=10,...) {
  en = length(ent)
  ite = length(it)
  dados <- m2.rn(dados)
  tudo = list()
  ajust = vector()
  for (i in 1:en) {
    for (j in 1:ite) {
      n <- ent[[i]]
      lm <- n[1]
      n <- n[2:length(n)]
      nn <- nneurons(dados=dados,neurons=neurons,n=n,it=it[j],lm=lm,...)
      tudo[[i+(j-1)*ite]] <- nn
      ajust <- cbind(ajust,nn$ajust)
    }
  }
  min <- which.min(ajust[5,])
  return(tudo[[min]])
}

```

```

# Combinações
#
# comb(valores)
#
# valores      = Vetor com os dados a serem combinados

comb <- function(valores) {
  out = list()
  for (i in 1:length(valores)) {
    a <- combn(valores,i)
    for (j in 1:ncol(a)) {
      out[[length(out)+1]] <- c(0,as.vector(a[,j]))
    }
  }
  return(out)
}

# Requisitando pacotes necessários
require(qcc)
require(neural)
require(utils)

## Etapa 1
serie <- arima.sim(50,model=list(ar=c(.7,-.2),model=c(2,0,0)))
ref <- list()

```

```

ref$serie <- serie
ref$sum.soma <- ksum(serie,plot=1)$soma
ref$sum.serie <- ksum(serie)$serie
windows()

series <- arima.sim(50,model=list(ar=c(.7,-.2),model=c(2,0,0)))

for (i in 2:400) {
  series <-
  rbind(series,arima.sim(50,model=list(ar=c(.7,-.2),model=c(2,0,0))))
}

## Etapa 2

d1 <- vector()
for (i in 1:400) {
  a <- ksum(series[i,])
  d1[i] <- sum(xor(a$serie,ref$sum.serie))
}

shapiro.test(d1)

mean(d1)

var(d1)

hist(d1,col="lightblue",ylab="Frequência",main="")

windows()

## Etapa 3

d <- Sys.time()

```



```

for (j in 1:40) {
  b = Sys.time()
  for (i in 1:10) {
    print(c(j,i))
    a <- series[length(redes1)+1,]
    wow <- cmc(a,2:10,ent,seq(10,210,25))
    redes1[[length(redes1)+1]] <- wow
    redes <- rbind(redes,as.real(prev(wow)))
  }
  c = Sys.time() - b
  print(c)
  file <- paste("c:\\karla\\sim",j,".RDATA",sep="")
  save.image(file)
}
f <- Sys.time() - d
print(f)
d2 <- vector()
for (i in 1:400) {
  sh <- qcc(redes[i,],type="xbar.one",plot=0)
  v <- rep(0,50)
  v[shewhart.rules(sh, limits = sh$limits, run.length =
qcc.options("run.length"))$beyond.limits] <- 1
  d2[i] <- sum(xor(v,ref$cusum.serie))
}

```

```
shapiro.test(d2)
mean(d2)
var(d2)
hist(d2,col="lightblue",ylab="Frequência",main="")
```

```
## Etapa 4
```

```
ks.test(d2,d1,alternative = "two.sided")
```

```
ks.test(d2,d1,alternative = "less")
```

```
ks.test(d2,d1,alternative = "greater")
```

