4
Redes Neurais Artificiais – RNAs

Redes neurais artificial (RNA) são algoritmos que se baseiam no comportamento do cérebro humano. Dessa forma, imita a estrutura massivamente paralela do cérebro, com capacidade de adquirir, armazenar e utilizar conhecimentos experimentais (HAYKIN, 2001). Apresentam como característica a robustez e tolerância à falhas, ou seja, mesmo diante do mau funcionamento de determinada neurônio a rede neural se mantém estável, produzindo resultados confiáveis.

Seu funcionamento é bastante semelhante ao sistema biológico em dois aspectos: o conhecimento é obtido através de processos de aprendizado e a densidade das sinapses (conexões entre neurônios) é a chave para se armazenar os conhecimentos adquiridos.

A estrutura das RNAs é formada por diversas unidades de processamento, chamadas neurônios artificiais, que se encontram interligadas. Essa rede de neurônios artificiais se comunica através de sinais e são capazes de representar comportamentos complexos.

No decorrer deste capítulo 4, serão apresentados um pouco dos eventos e pesquisas que levaram ao desenvolvimento das RN. Serão discutidos seus princípios de funcionamento, arquiteturas em que podem ser apresentadas, lógica e formas de treinamento assim como outros aspectos importantes.

4.1
Histórico

Embora os computadores atualmente consigam realizar operações matemáticas com uma velocidade muito superior ao que nossos cérebros podem processar informação, não existe máquinas ou *software* que consigam resolver problemas simples do dia a dia. Por exemplo, o processo de andar conversando com outra pessoa e no caminho reconhecer objetos sem nunca tê-los vistos especificamente é tão cotidiano que nem raciocinamos conscientemente sobre esses feitos. Nosso sistema nervoso consegue resolver esses problemas de como locomover o corpo para andar, manter atenção dando

continuidade à uma conversa e reconhecer padrões de objetos, tudo ao mesmo tempo e sem problemas. Fazemos isso recebendo (órgãos sensoriais) as informações do ambiente, enviando sinais para a nossa rede neural (cérebro), que processa as informações recebidas e envia um sinal para nossos músculos atuarem no ambiente externo com uma resposta, figura 29.

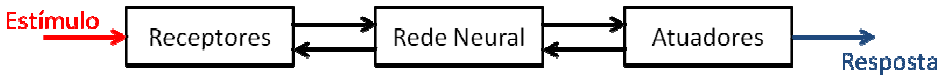

Figura 29 - Esquemático de direção de propagação de sinais durante o processamento de informações pelo cérebro (Rede Neural natural).

Em contrapartida o cérebro também recebe informações sobre as respostas dadas ao ambiente externo, enviando novos sinais para os receptores do nosso organismo, realizando um processo de retroalimentação.

Diante desse poder de interpretação, surgiu o interesse de desenvolver um sistema artificial capaz de assemelhar-se ao cérebro.

Dessa forma, as redes neurais artificiais foram idealizadas de forma a serem capazes de alcançar a capacidade apresentada pelo cérebro humano em identificar, reconhecer e classificar padrões, processando grandes quantidades de informações em tempo reduzido.

As pesquisas em inteligência computacional se esforçam para reorganizar o processamento de informações, deixando de lado o conceito linear e sequencial empregado até então, e passando a considerar o processamento paralelo de informações. Assim, a ideia de redes de processadores de informações ganhou força. Esses processadores foram desenvolvidos incorporando muito dos processos realizados por neurônios no cérebro humano.

A figura 30 mostra a representação da organização e entrelaçamento existentes entre os neurônios no cérebro humano.

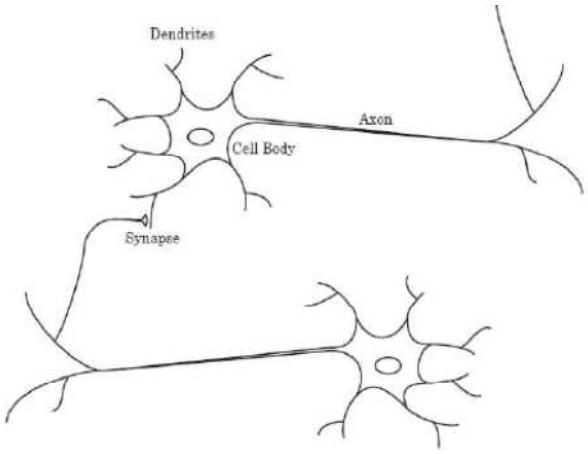

Figura 30 - Ilustração da organização existente entre os neurônios naturais.

Pode-se observar que as conexões entre um neurônio o outro é realizada através de sinapses. Esse conceito foi levado para a área de inteligência artificial e é utilizado para representar as conexões entre os processadores ou neurônios artificiais.

Historicamente, o primeiro modelo de neurônio artificial apareceu na década de 40. Em 1943 o psiquiatra e neuroanatomista Warren Mc Culloch juntamente com o matemático Walter Pitts trabalharam para desenvolver uma máquina que utilizava um modelo de neurônio artificial chamado *Psychon*, que apesar de interessante ainda não possuía poder de aprendizado.

Em 1949, é proposto o primeiro modelo de aprendizado para um conjunto de neurônios interligados, a partir da atualização dos pesos sinápticos. Esse modelo ficou conhecido como Regra de Hebb. Um pouco mais tarde, no ano de 1951, o pesquisador Marvin Minsky desenvolveu o primeiro neurocomputador em que o processamento de informação foi utilizado como inspiração para outros trabalhos posteriores.

Em 1958, alguns pesquisadores liderados por Frank Rosenblatt e Charles Wightman desenvolveram o primeiro neurocomputador bem sucedido. Seu trabalho deu força para o desenvolvimento do algoritmo do *Perceptron*, baseado nas regras de *Hebb*. No início da década de 60, Widrow e Hoff realizaram importantes contribuições ao modelo *Perceptron* (redes de uma única camada), introduzindo o conceito de erro médio quadrático.

Em 1969, Minsky e Papert demonstraram matematicamente a impossibilidade de se obter resolução de problemas linearmente inseparáveis através do Perceptron.

Um grande impulso nos estudos dos ANNs foi dado na década de 1980. Em 1982 John Hopfield propôs uma rede neural diferente do *Perceptron*. O seu modelo propunha uma rede com conexões recorrentes entre os neurônios. Desta forma o sinal não se propagava somente para frente. Alem disso, o seu aprendizado era conduzido de forma não supervisionada.

Outras contribuições foram realizadas nesta década, entretanto, somente em 1986 que Rumelhart, Hilton e Williams desenvolveram o algoritmo baseado no *Perceptron* de múltiplas camadas (MLP), treinadas com algoritmos de aprendizado por retro propagação de erro (Back-propagation). Este modelo possibilitou a resolução de qualquer tipo de problema.

4.2
Neurônio Artificial

O neurônio artificial é a base para o funcionamento das ANNs. Sua estrutura é simples e pode ser entendida como um processador que realiza uma soma ponderada dos sinais de entrada, representados por x_1 até x_n, pelos pesos. Estes, que podem ser conhecidos como parâmetros da rede, representam a memória da rede e são adquiridos através da experiência obtida por diversas apresentações dos padrões de entrada. A ativação de um sinal de saída depende do valor dessa soma e de uma função denominada função de ativação. Na figura 31, a representação de um neurônio artificial com seus pesos, sinais de entrada e função de ativação está demonstrado.

É importante ressaltar que a função de ativação representa a parte não linear de cada neurônio, sendo de fato o único lugar aonde a não linearidade se encontra.

O termo b_k, é conhecido como termo polarizador ou *bias.* A sua função é de indicar o ponto em que a função de ativação se encontra acima do eixo, elevando ou reduzindo a entrada da função de ativação. A sua definição é obtida através do mesmo processo que otimiza o valor dos pesos.

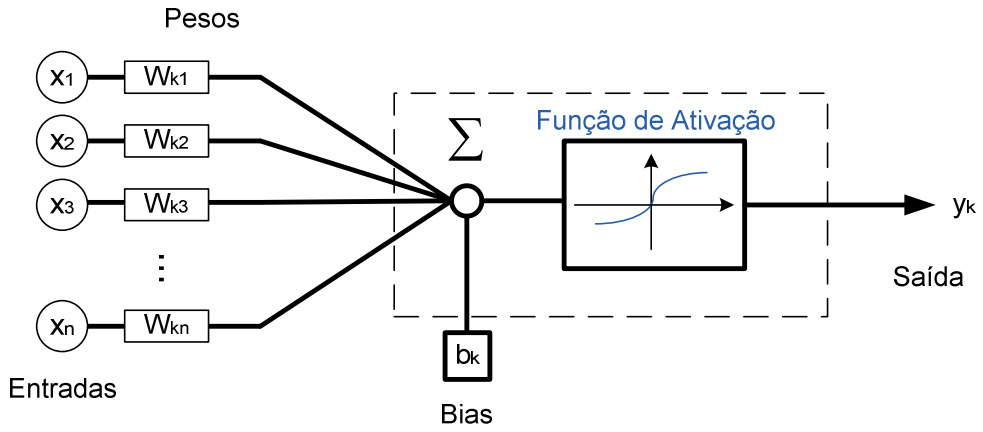

Figura 31 - Representação de um neurônio artificial.

Matematicamente o comportamento de um neurônio artificial pode ser representado pela equação 3,1.

$$y_k = F(\sum_{i=1}^{n}(x_i w_{ki}) + b_k) \quad (3,1)$$

onde:

x_i – é o i-ésimo padrão de entrada

w_{ki} – é o i-ésimo peso do neurônio k

b_k – função bias do neurônio k

F – função de ativação do neurônio

y_k – sinal emitido pelo neurônio k

Os pesos e a bias são os parâmetros que são estimados por uma RNA durante o seu treinamento. Após o treinamento seus valores são fixados, permanecendo constates.

4.3
Funções de Ativação

Existem algumas formas de função de ativação. Entretanto as mais utilizadas são a função degrau, linear, sigmoide (ou logística) e hiperbólica (HAYKIN, 2001). A forma da função de ativação (figura 32) está relacionada com a amplitude do intervalo da resposta de um neurônio.

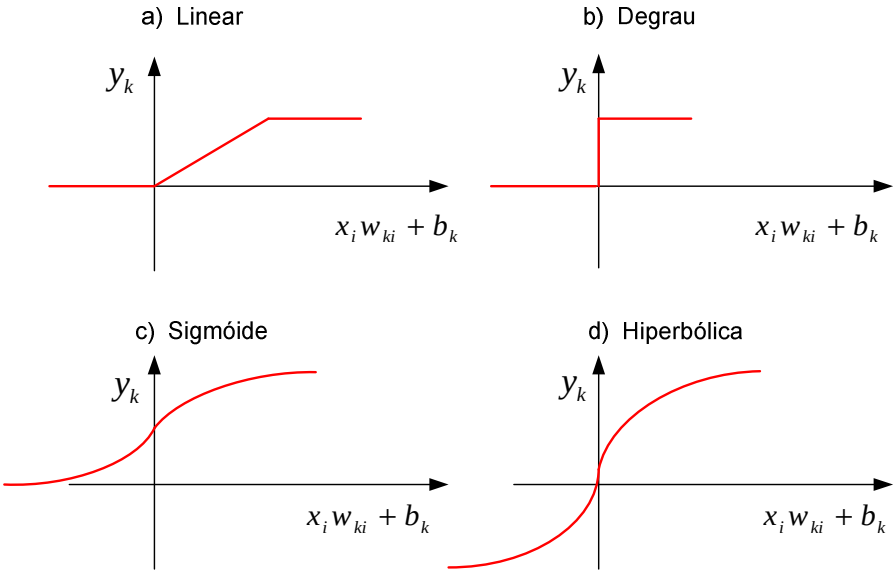

Figura 32 - Formato das principais funções de ativação.

A determinação da função de ativação é um dos fatores importantes para o bom funcionamento da RNA. Não se aconselha o uso, por exemplo, da função hiperbólica quando o estudo envolvem resultados somente positivos (visto que esse tipo de função aceita valores negativos). Já o uso das funções lineares e degrau seria indicado em situações em que se deseja evitar efeitos de saturação.

4.4 Arquitetura

O comportamento do sistema é controlado pela estrutura das ligações definidas pela sua arquitetura (ou topologia), pelos valores atribuído por cada conexão (pesos sinápticos) e a resposta de cada neurônio modulada pela função de ativação. Assim, a arquitetura e o número de neurônios em uma RNA é fundamental para o seu bom funcionamento, tendo influência direta no seu poder de processamento. Na figura 33 é mostrada a arquitetura de uma RNA, com a camada de entrada, camada escondida (intermediárias) e camada de saída.

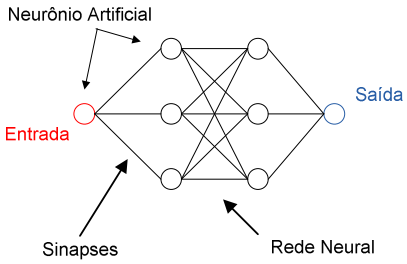

Figura 33 - Representação gráfica de uma topologia de rede neural.

Pode-se classificar a arquitetura da uma RNA como:

- Redes de arquitetura Recorrente
- Redes de arquitetura Não Recorrente.

As redes não recorrentes são definidas por não possuírem realimentação das suas saídas para as entradas. São ditas "sem memória" e a sua estrutura é formada por uma (figura 34-a) ou mais camadas (figura 34-b).

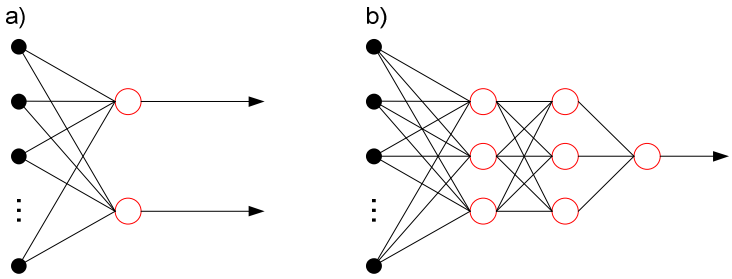

Figura 34 - Na figura *a)* esta representado uma RNA com uma camada (*Perceptron*) enquanto na figura b) a RNA possui múltiplas camadas (MLP).

Nesse tipo de arquitetura não é permitido ligações (transferência de sinal efetivo) de um neurônio à outro de uma camada anterior, ou mesmo da mesma camada. É importante diferenciar aqui os sinais efetivos e informações de erro. O sinal efetivo é que não encontra conexões para neurônios das mesma camada ou de camadas anteriores, as informações de erro podem e são retro propagadas em muitos casos.

Um exemplo de uma redes organizadas em camadas são as redes chamadas de *feedfoward.*

Já as redes recorrentes possuem como principal característica conexões de neurônios entre camadas ou com camadas anteriores. Por isso mesmo, as ANN recorrentes são chamadas de redes com memória. Nesse tipo de rede, a arquitetura não

se encontra amarrada a estruturas de camadas. Podem ser encontrados redes parcialmente recorrentes (figura 35) ou totalmente recorrentes (figura 36).

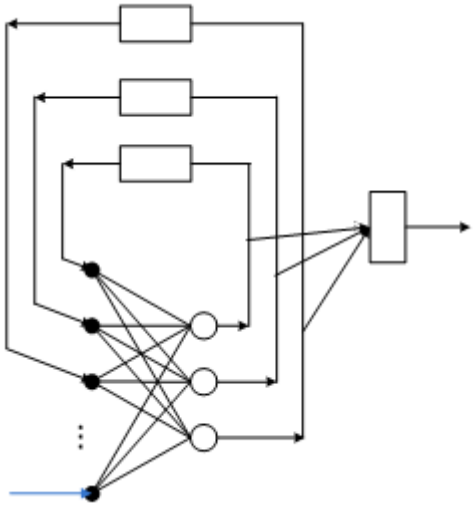

Figura 35 - Rede parcialmente recorrente (Rede de Elman).

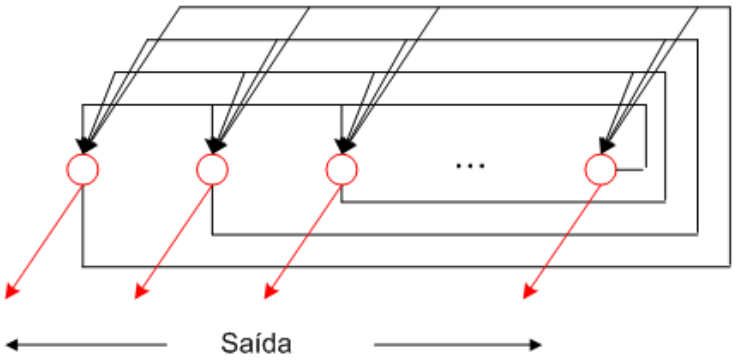

Figura 36 - Rede totalmente recorrente (Rede de Hopfield).

As redes recorrentes são interessantes em algumas aplicações temporais, uma vez que conseguem gravar informações em tempo anterior, possibilitando uma melhor previsão para o futuro.

4.5
Formas de Aprendizado - Treinamento.

O treinamento de uma RNA é a fase em que o os pesos em cada conexão nos neurônios são sistematicamente ajustados de forma a transformar o sinal de entrada em

um sinal de saída desejado ou consistente. Em outras palavras, o objetivo do treinamento é determinar a intensidade das conexões entre os neurônios. O ajuste dos pesos da rede é realizado até que os mesmos convirjam para um determinado valor. A densidade das conexões é definida pela quantidade de conexões ligadas (pesos com valores entre 1 e -1, exceto zero) e conexões desligadas (pesos com valores iguais ou muito próximos a zero).

Já o aprendizado é o algoritmo utilizado para modificar o valor dos pesos de forma a garantir que a RNA responda adequadamente ao problema proposto. Nesse sentindo, o ajuste dos parâmetros da rede através de regras de aprendizado bem estabelecidas é o que define um algoritmo de aprendizagem. Existem diferentes regras de aprendizado. Entre as mais conhecidas vale a pena citar: Adaline, Backpropagation, Competitive Learning, Delta Rule (Mendel, 1995; Timoszczuk, 2004).

Os procedimentos de treinamento podem ser classificados por treinamento não supervisionado ou supervisionado.

No treinamento não supervisionado, é inexistente um valor de referência (alvo de resposta). Dessa forma, os resultados obtidos pela RNA não podem ser comparados para gerar informações de erro que guiem a atualização dos pesos. Logo, este tipo de treinamento é caracterizado pela comparação entre os próprios sinais de entrada. Assim, a rede descobre padrões característicos (correlações) dos dados de treinamento.

Já o treinamento supervisionado utiliza um valor de alvo a ser atingido pela RNA. Dessa forma, dado um valor de entrada a rede gera um valor de saída que será comparado ao alvo pré-determinado, gerando um valor de erro. Esse erro então é utilizado para ajustar os pesos da rede de forma a minimiza-lo. Normalmente a soma do erros quadráticos da rede é o fator usado como parâmetro de treinamento. O algoritmo mais utilizado neste caso é o chamado algoritmo de retropropagação de erro (*Backpropagation*). Este algoritmo funciona não alterando os pesos de cada conexão entre os neurônios, quando o sinal passa da camada de entrada para a saída. Nesse ponto um valor de erro é gerado através da comparação da saída com o valor alvo, e esse erro é retropropagado pro toda a rede, atualizando os valores dos pesos sinápticos.

O aprendizado é adquirido através da apresentação de exemplos relacionados com o problema que se deseja solucionar. Dessa forma, é de suma importância a apresentação de um histórico em dados para que seja possível desenvolver um

algoritmo em redes neurais. Quando não há mais alterações significantes nesses pesos o rede realizou a aprendizagem.

4.6
Modelagem de uma RNA

A modelagem de uma RNA leva em conta a escolha dos dados que servirão para treinar a rede e garantir a sua capacidade de generalização, a melhor forma de apresentação desses dados (normalização, codificação, filtragem, etc.) e a escolha do número de neurônios em cada camada.

O primeiro passo da modelagem de uma ANN é sem dúvida com relação aos dados relacionados ao problema em questão, os quais estão disponíveis na forma de um histórico de casos. Como a rede aprende através destes, é de suma importância que os mesmos sejam representativos do problema ou do conhecimento que se deseja extrair. Nesta etapa divide-se o conjunto de dados em 3 subconjuntos: Conjunto de treinamento, conjunto de validação e conjunto de teste. Normalmente o conjunto de treinamento representa um valor em torno de 70% de todos os dados, enquanto o percentual dos conjuntos de validação e teste giram por volta de 20% e 10% respectivamente.

4.6.1
Validação cruzada e generalização e testes do modelo

Já foi observado que o treinamento é realizado apresentando-se os padrões à rede. A apresentação de todo o conjunto de padrões de treinamento corresponde ao que se convencionou chamar de uma época para a rede neural. Geralmente são necessárias algumas épocas para que uma rede esteja treinada e pronta para produzir resultados.

Entretanto, para que a RNA não reconheça somente os padrões do conjunto de treinamento um procedimento de validação é executado. Essa validação, ou validação cruzada é feita da seguinte forma. O treinamento da RNA é iniciado apresentando todos os padrões do conjunto de treinamento. Esse procedimento é repetido por algumas épocas, em média de 10 a 20. Neste instante a rede é testada com dados do conjunto de validação. O erro é então calculado e armazenado. Novamente o treinamento é reiniciado e executado por mais algumas épocas (mesmo número de épocas utilizadas

anteriormente). Obviamente o erro associado ao treinamento é sempre inferior ao erro associado ao teste de validação. O que se avalia então não é a relação entre estes erros mas sim a curva de erro de validação, que deve atingir um mínimo para determinada época. Após esse mínimo de validação, mesmo com o erro de treinamento decrescendo, o de validação tende a crescer, como mostrado na figura 37.

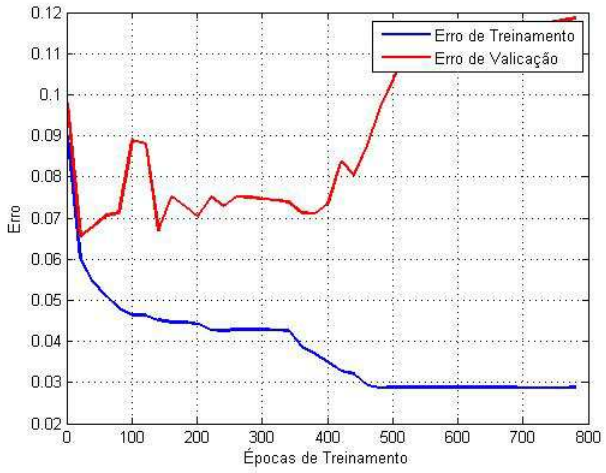

Figura 37 - Relação entre o erro de treinamento e o erro de validação em uma RNA do tipo MLP com treinamento por *Backpropagation*.

Neste ponto é correto afirmar que apesar da rede melhorar o seu desempenho com os dados do conjunto de treinamento ela começa a perder poder de generalização. A rede está atendo-se somente os padrões de treinamento e não mais conseguindo representar outros padrões. Esse fenômeno é conhecido como super treinamento ou *overfitting*.

O teste final é executado com a rede que obteve o melhor desempenho (relacionado ao erro de validação) e com os dados do conjunto de teste (os quais nunca foram apresentados a RNA em questão).

4.6.2
Tratamento dos padrões de entrada para as RNA

Tanto na hora da realização do seu treinamento quanto para o seu teste e aplicação real, é imprescindível que os padrões sejam previamente tratados. Os processos mais utilizados para o tratamento desses padrões são a codificação, filtragem e normalização.

O processo de codificação (como uma codificação binária) é necessário em alguns casos, visto que eventualmente estes não são representados por valores numéricos. Por exemplo, um problema forneça para a RNA uma entrada como o mês do ano de determinada ocorrência, não pode ser realizada com o nome do mês em questão.

A filtragem dos dados disponíveis para o treinamento também pode ser interessante para o melhor funcionamento da rede. Isso é especialmente verdade quando em um conjunto de informações somente uma determinada parcela representa o fenômeno estudado. Dessa forma ao se retirar todo o resto, a complexidade do conhecimento que a rede deve representar diminui, aumentando a eficiência da ANN.

Por fim é importantíssimo a normalização dos dados antes de que os mesmos sejam fornecidos à rede. Isso é melhor entendido para um caso em que os dados de entrada são formados por diferentes variáveis, as quais possuem ordens de grandeza distintas. Dessa forma, os dados de entrada com ordem de grandeza superior acabam por mascarar o restante. Outra justificativa para normalizar os dados vem do fato de que os neurônios da RNA sempre transmitirem sinais que variam de -1 à +1.

É interessante ressaltar ainda que a normalização dos dados deve deixar uma margem (tanto inferior quanto superior), visto que nem sempre é possível se trabalhar com informações que sejam completamente representativas do problema analisado. Dessa forma, ao se estabelecer essa margem de segurança evita-se a RNA se confunda com valores fora da curva de normalização provocados por dados de valor inferior ou superior aos apresentados para seu treinamento.

4.6.3
Número de neurônios

Outro fator que influencia na modelagem da RNA é o número de neurônios contido, principalmente na(s) camada(s) escondida(s). Isso por que o número de neurônios na camada escondida acaba definido o grau de complexidade do modelo da RNA. Se o número de neurônios é baixo para o problema, ou seja, se o modelo tem complexidade inferior à necessária, a rede não irá apresentar bons resultados (não fará uma boa representação do caso em estudo). Entretanto se o número de neurônios é excessivo, a rede terá uma complexidade muito maior que o necessário e irá com isso aprender o que não precisa, como ruído.

O mesmo cuidado na quantidade de neurônios nas camadas de entrada e saída deve ser considerado. Isso por que, alem de também aumentarem ou reduzirem o grau de complexidade do modelo da rede, os neurônios na camada de entrada pode apresentar correlação e dessa forma se tornarem redundantes (o que prejudica o funcionamento adequado da RNA).