

APLICAÇÃO DE SISTEMAS INTELIGENTES EM ENGENHARIA DE POTÊNCIA

Germano Lambert-Torres

Alexandre Pinto Alves da Silva

Escola Federal de Engenharia de Itajubá - EFEI
Grupo de Inteligência Artificial
Av. BPS 1303 - Itajubá - 37500-000 - MG - Brasil

RESUMO

Este artigo tem a finalidade de apresentar o estado-da-arte da aplicação de sistemas inteligentes (sistemas especialistas e redes neurais) na solução dos problemas de sistemas elétricos de potência. Inicialmente, uma pequena descrição teórica é apresentada, para em seguida, mostrar os assuntos que mais tem tido sucesso com estas técnicas. São apresentadas também as características dos problemas que mais se adaptam a cada uma das técnicas. Os sistemas atualmente em uso serão também brevemente analisados mostrando seus pontos fortes e fracos. Para finalizar, é realizada uma análise dos próximos problemas que serão alvo dessas técnicas.

Palavras chaves: Sistemas Especialistas, Redes Neurais, Engenharia de Conhecimento, Operação dos Sistemas de Potência, Planejamento dos Sistemas de Potência.

APPLICATIONS OF INTELLIGENT SYSTEMS TO POWER ENGINEERING

ABSTRACT

This paper presents the state-of-the-art of intelligent system (expert system and neural network) application to electric power system problems. Initially, a short theoretical description is presented, and afterwards, the most successful applications are enumerated. Features that make a problem suitable for being solved by intelligent systems are stressed. Strong and weak points of current technology are pointed. Finally, new perspectives of intelligent system application to power engineering are analyzed.

Keywords: Expert Systems, Neural Networks, Knowledge Engineering, Power System Operation, Power System Planning

1. INTRODUÇÃO

A busca do ser humano em encontrar máquinas inteligentes e que pudessem substituí-lo em suas diversas tarefas data dos primórdios da humanidade. Já nos hieróglifos das pirâmides egípcias são encontradas menções a este fato.

Esse objetivo pareceu estar mais próximo em meados da década de 1950, quando a ciência encontrava-se em um momento de euforia no pós-guerra, pois diversos avanços na lógica matemática e na psicologia cognitiva, possibilitavam representar o cérebro humano. Com o advento e popularização dos computadores, principalmente através das linguagens de terceira geração, passamos a dispor de máquinas capazes de imitar o cérebro. Assim foi criada a ciência chamada *Inteligência Artificial* (IA).

Muito foi prometido nesta fase pelos pesquisadores. Estes pensavam ter encontrado finalmente a máquina inteligente. Na verdade, pouco foi realmente concretizado

pois nem as funções do cérebro foram totalmente metrizadas bem como a ferramenta utilizada dispunha de recursos limitados.

Isto causou um *período negro* para os pesquisadores pois o descrédito foi imediato. Felizmente, alguns deles continuaram seus trabalhos e novamente este ramo da ciência da computação foi ganhando importância e, principalmente, mostrando-se útil para resolver problemas que anteriormente não eram solucionados via computador, ou então, possuíam métodos analíticos ineficientes para resolvê-los.

Este artigo trata exatamente deste novo período da IA e mais precisamente de sua aplicação na solução dos problemas de sistemas elétricos de potência. Para esta aplicação, dois sistemas se sobressairam: os Sistemas Especialistas (SE) e as Redes Neurais (RN). Estes dois sistemas visam imitar o cérebro humano na solução dos problemas. Os SE têm uma visão macroscópica do cérebro, representando suas interações através de relações lógicas, com o uso de por exemplo regras de

produção do tipo *If-Then*. As RN utilizam como elemento básico a estrutura de um neurônio, formando redes de neurônios artificiais, tendo assim uma visão microscópica do cérebro.

Antes de passarmos aos tópicos principais deste artigo, cabe ressaltar que todas as definições, os exemplos e as conclusões expressas são concernentes a aplicação de Sistemas Inteligentes em Sistemas de Potência. Isto é feito para que se possa exemplificar com mais facilidade o atual estado-da-arte no setor.

2. SISTEMAS ESPECIALISTAS

Os Sistemas Especialistas (SE) são o ramo da IA que tenta imitar o cérebro humano através do uso de processos de decisão semelhantes aos utilizados pelos especialistas humanos. Para tal, é organizada uma *base de conhecimento* (BC), que contém todas as informações necessárias de um dado campo do conhecimento. Estas informações são organizadas, na maioria dos casos, em fatos e regras. Os fatos contêm informações não condicionais, como por exemplo: estados dos elementos, sua posição no sistema e suas características. Eles podem ser classificados em dois tipos: estáticos ou dinâmicos, segundo o tempo de atualização desse seus dados.

As regras são declarações condicionais e servem para definir relações entre os fatos ou o encaminhamento de uma dada solução. Na realidade, elas formam a base de sustentação de um SE e sua grande diferença para um programa convencional. Neste último tipo, o conhecimento deve ser depurado e organizado antes ser colocado no interior do programa. E mais, nos programas convencionais somente os problemas inicialmente propostos são resolvidos não havendo a possibilidade da solução de problemas diferentes dos concebidos pelo programador para seu uso.

Nos SE, o conhecimento é colocado em sua *forma bruta* sem grandes depurações ou organizações iniciais. Ele é escrito na BC e a resposta é achada pelo *motor (ou máquina) de inferência* (MI). Esta segunda parte de um SE é a responsável por realizar o processo de inferência no conhecimento disponível, achando respostas parciais e a resposta final. O MI trabalha utilizando relações de lógica formal e independe do tipo de conhecimento existente na BC.

Normalmente, o MI utiliza uma "memória de trabalho" ("working memory") para escrever os fatos que vão sendo inferidos durante o processo de busca de uma solução. Esta memória funciona como um rascunho que é jogado fora quando o trabalho é concluído [1].

Finalmente, a terceira parte de um SE é a *interface com o usuário* (UI), que deverá ser amigável, podendo ter, segundo o caso, gráficos e desenhos ilustrativos e fazer sua comunicação via linguagem natural. A diferença deste tipo de interface para a dos programas convencionais é que esta deve conter um módulo de explicação de como o SE chegou a uma dada resposta. Isto é uma característica importante dos SE, ou seja,

poder fornecer ao usuário com que tipo de raciocínio o sistema resolveu um dado problema.

2.1 Quais são os tipos de problemas que podem ser resolvidos pelos SE ?

Como o próprio nome especifica, estes são sistemas especialistas e não generalistas. Isto significa que eles devem ser concebidos para resolver problemas em uma área restrita do conhecimento. É importante notar que é possível construir um conjunto de SE para uma área mais abrangente, porém, deve-se ter em mente que este desenvolvimento será individual, ou seja, por classes de problemas. Após os desenvolvimentos individuais, escreve-se um gerenciador para o uso dos SE desenvolvidos.

Então seriam características desejáveis para um SE, quanto ao desenvolvimento do sistema [2]:

- (a) estar em um domínio bem conhecido e delimitado,
- (b) existir um consenso sobre como deve ser realizada uma dada solução,
- (c) existir ao menos um especialista envolvido no desenvolvimento, para que ele possa explicar qual a sua linha de raciocínio,

quanto a escolha do problema [3]:

- (d) não existir uma fórmula ou um procedimento analítico bem conhecido e de bom desempenho
- (e) existir escassez de especialistas humanos para a solução do problema
- (f) existir dados ou informações não precisas vindas do sistema.

Desta forma, alguns problemas de Sistemas de Potência podem ser adaptados as condições acima [4]:

- (a) por não existir uma fórmula ou um procedimento analítico conhecido;
por exemplo: planejamento da manutenção e restauração de sistemas
- (b) pelos métodos analíticos serem parcialmente conhecidos;
por exemplo: controle de tensão, avaliação de contingências
- (c) pelos métodos analíticos serem ineficientes;
por exemplo: análise de contingências, localização de faltas

Como contra-exemplo, ou seja, problemas que não se deve aplicar esta técnica pode-se citar o cálculo do fluxo de potência, cálculo de correntes de curto-circuito, entre outras.

2.2 Quais são as áreas de desenvolvimento de SE em Sistemas de Potência ?

Existem desenvolvimentos de SE em diversas áreas ligadas aos Sistemas de Potência, sendo as principais [3]:

- (a) *diagnóstico* - onde procura-se a causa de um mau funcionamento de um equipamento ou do sistema,
- (b) *prescrição* - recomendações ou ações para corrigir um dado problema,
- (c) *análise* - monitoração das leituras e informações do sistema com o objetivo que acompanhar seu funcionamento e definir seu estado,
- (d) *previsão* - procura-se estabelecer as conseqüências de uma determinada ação ou grupo de ações,
- (e) *seleção* - criação de um rol de possibilidades de soluções ou cenários e a escolha de um ou um grupo segundo alguns critérios,
- (f) *configuração* - configurar ou reconfigurar objetos, cenários ou sistemas segundo alguns limitantes e critérios,
- (g) *planejamento* - desenvolvimento da seqüência de ações para atingir um determinado objetivo dentro de um tempo desejado,
- (h) *controle* - combinação de diversas áreas acima, incluindo análise e prescrição,
- (i) *instrução* - utilização no treinamento através da instrução inteligente assistida por computador.

Em um levantamento que levou em consideração artigos publicados, relatórios internos e sistemas consultados por diversas forças-tarefas da CIGRÉ [5,6] e do IEEE [7,8], entre 1986 a 1993, chegou-se a seguinte divisão por áreas de problemas:

- Operação de sistemas	23%
- Operação de usinas	16%
- Diagnóstico de problemas da geração	8%
- Análise de segurança on-line	8%
- Restauração de sistemas	7%
- Automação da distribuição	7%
- Restauração de subestações	5%
- Planejamento da transmissão	3%
- Instrução e treinamento	3%
- Diagnóstico de problemas da transmissão ..	2%
- Outros	18%

2.3 Quais são as etapas do desenvolvimento de um SE ?

Existem várias metodologias que podem ser aplicadas no desenvolvimento de um SE. Uma das mais aceitas foi proposta por Hayes-Roth *et al.* em [9], a qual contém cinco estágios evolutivos: identificação, conceituação, formulação, implementação e testes.

Durante a etapa de *identificação* são determinados os participantes do desenvolvimento e os recursos necessários, bem como as sub-metas e objetivos parciais do projeto.

Na *conceituação*, os especialistas e/ou engenheiros de conhecimento definem quais os elementos chaves do conhecimento, como conceituá-los e relacioná-los, e a estrutura geral do conhecimento. Em seguida na etapa de *formulação*, estes conceitos e relações são expressos dentro de uma representação formal que está ligada à linguagem ou sistema de engenharia de conhecimento escolhido.

Na fase de *implementação*, o engenheiro de conhecimento combina e detalha o conhecimento através de regras/fatos e estruturas de inferência definindo assim um protótipo do SE. Na fase de *testes*, deve-se avaliar o desempenho do protótipo e revisá-lo conforme instruções do especialista. Nesta fase, o módulo de explanação de como a resposta foi achada é bastante importante, pois não deve-se somente achar a resposta correta, mas achá-la através do raciocínio correto.

Pronto o protótipo, o SE estará pronto para ser executado de maneira "off-line", ou seja, dentro das salas de estudo, para que sofra um processo maior de validação ("debugging"). Em seguida, o SE passa a um modo de operação chamado "on-line", ou seja, ele já se encontra em uso, por exemplo, ao lado da mesa do operador, recebendo informações do sistema, processando-as e fornecendo seus resultados para avaliação do usuário. Este processo continua até que as respostas do SE sejam amplamente aceitas pelos usuários, nesta fase o sistema passa ao modo de operação "real-time". Neste, o SE além de realizar os processos de inferência, aplica-os diretamente ao sistema elétrico, fornecendo ao usuário somente a nova situação do sistema depois da mudança.

Nenhuma dessas etapas, nem modos de operação, devem ser ignoradas sob pena de criarmos sistemas falhos ou com algum grau de rejeição por parte do usuário.

Quanto a este último aspecto, é importante notar que o conhecimento contido na BC deve abranger não somente aspectos teóricos como aspectos práticos. Uma das formas de diminuir a rejeição por parte dos usuários do sistema (que temem perder seu lugar de trabalho) é incorporá-los no desenvolvimento do SE, principalmente, na etapa de conceituação e formulação do problema. Além disto, é importante, quando o SE estiver sendo implantado, que existam palestras expositivas sobre suas vantagens e limitações afim de diminuir as barreiras que possam vir a existir.

2.4 Quais são as ferramentas para o desenvolvimento de um SE ?

Por se tratarem os SE de programas computacionais, existem algumas linguagens e sistemas próprios para se desenvolver este tipo de programa, são elas: LISP, PROLOG e Sistemas de Engenharia de Conhecimento. As duas primeiras são linguagens, enquanto o último é uma classe de softwares destinados a confecção de SE.

A linguagem LISP ("LISt Processor") data do final da década de 50 e foi inicialmente proposta por John McCarthy. Ela trabalha com listas ordenadas, sendo o próprio programa uma lista. Existem máquinas dedicadas totalmente a esta linguagem, as chamadas *máquinas lisp*.

A linguagem PROLOG (PROgramming in LOGic) foi formulada inicialmente pelo grupo do Professor Alain Colmaurer, na década de 70. Ela trabalha utilizando cálculo de predicados e foi escolhida pelos japoneses

como linguagens oficial do projeto dos computadores de quinta geração.

Os Sistemas de Engenharia de Conhecimento ("Knowledge Engineering Systems") são softwares que contêm toda a estrutura montada para receber o conhecimento de uma dada área. Eles possibilitam que uma pessoa construa um SE, mesmo sem conhecer suas técnicas de representação e processos de inferência. São ideais para testes iniciais de idéias ou desenvolvimentos de protótipos [10].

Quando as linguagens convencionais (Fortran, Pascal, Basic, etc.), elas não são instrumentos adequados para a construção de SE, pois não possuem a flexibilidade necessária para conterem programas não algorítmicos. Não obstante a literatura apresenta alguns sistemas escritos nessas linguagens, que inicialmente funcionam bem, mas com o passar do tempo e o aumento da BC, começam a ter problemas de manutenção e consistência [11]. A única exceção é feita a linguagem C, que apesar de algorítmica possui certa flexibilidade para a representação do conhecimento [12, 13]. Ela é a recomendada para computadores de grande porte que não possuam as linguagens de IA.

2.5 Quando um SE deve ser utilizado e quais as suas limitações?

Os SE podem ser utilizados por diversos grupos de pessoas, desde o próprio especialista, ajudando-o em atividades intelectuais de alto nível, ao jovem operador do sistema, auxiliando-o na tomada de decisão em situações críticas [14]. Outras funções de um SE são [15]:

- (a) padronizar as operações dentro da companhia, que de outro modo poderia ser corroida nos diversos níveis e localidades,
- (b) preservar o conhecimento, principalmente quando das aposentadorias e afastamentos repentinos,
- (c) treinar ou relembrar os usuários em diversas funções do sistema elétrico,
- (d) executar as decisões semelhantes as dos especialistas mesmos em momentos críticos e na presença de um grande volume de dados (ou na falta de alguns deles),
- (e) substituir os especialistas em atividades rotineiras e triviais.

As principais limitações dos SE residem no fato deles somente resolverem problemas de áreas muito bem definidas, bem como a falta de senso comum e sua incapacidade de resolver problemas por analogia. Uma outra dificuldade é a incorporação de novos conhecimentos, que necessitam do engenheiro de conhecimento para a sua validação e incorporação.

3. REDES NEURAIS

A aplicação de SE requer que:

- (a) o conhecimento para resolver o problema exista,
- (b) o conhecimento possa ser representado (codificado),

- (c) o SE resultante atenda aos requisitos de operação e as restrições de tempo de desenvolvimento.

Sistemas Especialistas têm sido criticados por requererem enorme esforço para construção (aquisição de conhecimentos) e manutenção da BC. *Experiência* é escassa em muitas áreas, conseqüentemente a codificação de conhecimento pode ser limitada na prática. Ainda mais, devido a grande complexidade na representação de conhecimento em SE, o aprendizado automático é muito difícil. Experiência na forma de histórico de casos resolvidos pode ser a única fonte de conhecimento nestes casos. Redes Neurais (RN) possuem a vantagem de extrair conhecimento automaticamente a partir de exemplares de casos resolvidos, podendo desta forma descobrir novas relações entre conceitos e hipóteses.

3.1 Modelos de Redes Neurais Artificiais

Uma RN artificial é uma estrutura de processamento de informação não algorítmica, na forma de um grafo direcionado com cada nó, denominado neurônio, tendo diversas entradas e um único sinal de saída que pode ser propagado por diferentes canais de comunicação com outros neurônios. Por não algorítmica entende-se que uma sessão de treinamento se faz necessária para que a rede desenvolva capacidade de processamento. O sinal de saída de cada neurônio depende exclusivamente de suas entradas.

A Figura 1 mostra a representação genérica de um neurônio. O operador somatório calcula a soma dos sinais de entrada do neurônio, ponderados pelos pesos das respectivas conexões. A função $f(\cdot)$ é dita de ativação e é aplicada ao resultado do somatório. A função $f(\cdot)$ pode ser um degrau, uma função sigmoidal, uma reta, etc.

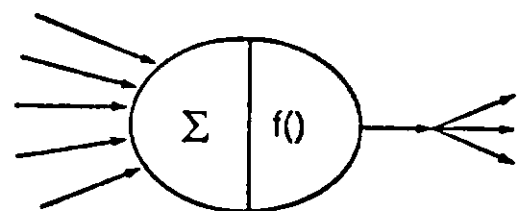


Figura 1: Representação Matemática de um Neurônio.

O treinamento de uma rede neural consiste em estimar (adaptar) os pesos das conexões entre neurônios de acordo com uma regra pré-definida (algoritmo de treinamento). Diversas arquiteturas de redes neurais têm sido definidas. Existem basicamente duas formas de treinamento: supervisionado e não-supervisionado. Entre os modelos de RN que são treinados de forma supervisionada encontram-se: os perceptrons, as redes de Hopfield, as LVQs e a Funcional Link [16, 17]. Entre as que são treinadas de maneira não-supervisionada estão: o mapa auto-organizável de Kohonen e as arquiteturas ART propostas por Grossberg.

Como os perceptrons de múltiplas camadas e os mapas auto-organizáveis de Kohonen são os modelos mais utilizados em sistemas de potência, seus algoritmos de treinamento são descritos a seguir.

3.1.1 Treinamento Supervisionado de Perceptrons com Múltiplas Camadas

Redes neurais do tipo perceptron de múltiplas camadas, treinadas com a regra de retro-propagação de erro [18], têm sido as mais utilizadas em problemas de sistemas de potência, pois podem aproximar quaisquer funções e regiões de decisão. Um perceptron de múltiplas camadas pode ser representado por um grafo direcionado sem realimentação, conforme a Figura 2. Uma rede com três camadas é mostrada, embora múltiplas camadas intermediárias possam ser usadas. Na Figura 2, cada camada só está conectada à camada imediatamente subsequente, mas em geral pode haver conexões diretas entre camadas não-adjacentes. Ainda nesta figura, as conexões entre camadas são globais, isto é, cada neurônio está conectado a todos os neurônios da camada imediatamente subsequente. A regra de retro-propagação de erro não requer que as conexões sejam globais.

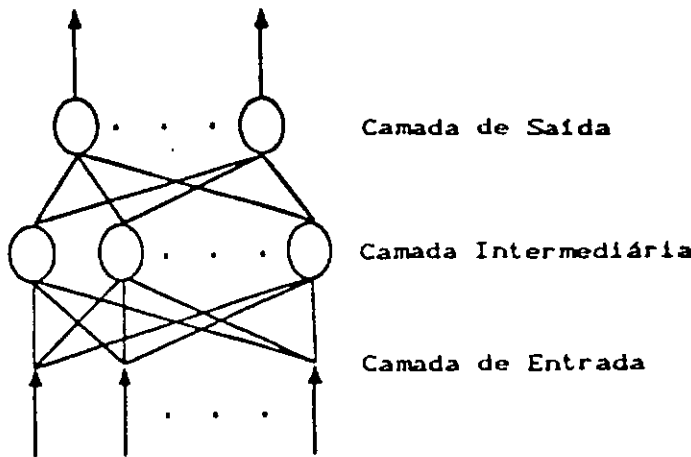


Figura 2: Perceptron de Múltiplas Camadas.
Nota: Os nós da camada de entrada são apenas condutores de sinal.

A regra de aprendizado retro-propagação de erro é um procedimento que modifica os pesos das conexões da rede de modo a reduzir o erro entre o sinal de saída gerado pela rede e o sinal desejado. Este ajuste ocorre a medida que os exemplares do conjunto de treinamento são apresentados à rede.

Para simplificar a descrição do algoritmo de aprendizado retro-propagação de erro, um perceptron de 3 camadas, globalmente conectado, tal como mostrado na Figura 2 é considerado. Os neurônios das camadas de entrada, intermediária e saída são representados pelos sub-índices i , j e k , respectivamente. Logo, se x_i denota os sinais de entrada da RN, a entrada total de um neurônio da camada intermediária é:

$$z_j = \sum_i x_i w_{ij}$$

onde w_{ij} é o peso da conexão entre as unidades i e j .

A saída de uma unidade na camada intermediária é:

$$s_j = f(z_j)$$

onde $f(\cdot)$ é a função de ativação.

Uma função de ativação conveniente é a seguinte função sigmóide:

$$f(z_j) = 1 / \{1 + e^{-(z_j - \theta_j) / \theta_0}\}$$

A função acima produz uma saída que varia continuamente de 0 a 1. O parâmetro θ_j serve como "bias", posicionando a região de transição da função. O parâmetro θ_0 determina o "ganho" da transição.

No processo de treinamento, para cada par de padrões (entrada/saída) apresentado à rede, a saída da mesma para o conjunto corrente de pesos das conexões e "bias" das funções de ativação é comparada ao sinal de saída desejado. Os erros dos neurônios da camada de saída podem ser representados por:

$$e_k = s_{d,k} - s_k$$

onde $s_{d,k}$ é a saída desejada para o neurônio, e s_k é o sinal de saída gerado. A função de erro total pode ser escrita como:

$$E = \frac{1}{2} \sum_k e_k^2$$

Aprender significa modificar os pesos das conexões e os "bias" das funções de ativação de forma a minimizar a função de erro total acima, usando um procedimento de descida em gradiente (a inicialização do processo é feita aleatoriamente utilizando-se valores de pequena magnitude). Resumidamente, o aprendizado via regra de retro-propagação de erro consiste do seguinte processo iterativo (para cada par de padrões de treinamento):

- 1) Propagar o sinal de entrada pela rede;
- 2) Comparar o sinal de saída obtido com o valor desejado;
- 3) Calcular e retro-propagar a medida de erro (começando pela camada de saída) através da rede; e
- 4) Minimizar o erro a cada estágio, ajustando os pesos das conexões de acordo com a seguinte equação:

$$\Delta w_{jk} = \eta \delta_k s_j$$

onde o parâmetro η é a taxa de aprendizado. O sinal de erro δ_k em um neurônio k da camada de saída é calculado por:

$$\delta_k = (s_{d,k} - s_k) s_k (1 - s_k)$$

e o sinal de erro δ_j para um neurônio j pertencente a camada intermediária é calculado por:

$$\delta_j = s_j (1 - s_j) \sum_k \delta_k w_{jk}$$

Os "bias" das funções de ativação são obtidos tornando θ_j equivalente ao peso de uma conexão w_{0j}/w_{0k} , que liga um canal de entrada com sinal sempre igual a 1 aos neurônios j e k .

Um algoritmo para o treinamento de perceptrons de múltiplas camadas, apropriado para conjuntos de treinamento de grande porte, é o Optimal Estimate Training 2 (OET2) proposto em [19, 20, 21].

Quando um perceptron de múltiplas camadas é treinado para atuar como classificador ou interpolador, é necessário evitar um super-ajuste dos dados do conjunto de treinamento, de modo a obter boa capacidade de generalização (isto é, prover resultados corretos para exemplares não pertencentes ao conjunto de treinamento). Isto pode ser alcançado realizando-se uma busca pela arquitetura neural mais simples (aquela com menor número de neurônios e conexões) que produza a menor taxa de erro para o conjunto de teste selecionado (que deve ser escolhido independentemente do conjunto de treinamento).

3.1.2 Treinamento Não-Supervisionado de Mapas Auto-Organizáveis de Kohonen

Um Mapa Auto-Organizável de Kohonen identifica as similaridades entre os exemplares e forma agrupamentos que estejam associados a certas classes do problema [22].

No treinamento não-supervisionado somente os padrões de entrada são apresentados à rede. Logo, nos mapas auto-organizáveis de Kohonen não existe uma saída desejada para produzir uma função de erro, cuja minimização levaria ao treinamento da rede e consequente determinação dos pesos das conexões. No caso das redes de Kohonen os pesos são determinados dinamicamente, de tal modo que vetores de entrada similares quando apresentados à rede irão excitar neurônios vizinhos, definindo assim os agrupamentos. O número de grupos é no máximo igual ao número de neurônios.

A rede de Kohonen está ilustrada na Figura 3. Na rede mostrada os neurônios formam um "grid" bi-dimensional, sendo que cada um desses neurônios está conectado a todos os componentes do vetor de entrada. É realizado então um mapeamento do espaço n -dimensional (vetor de entrada) para o espaço bi-dimensional. A rede de Kohonen divide o espaço dos vetores de entrada em sub-espacos, de acordo com a densidade dos vetores de entrada em cada sub-espaco. Além disso, os protótipos (o protótipo de um grupo é o melhor representante deste grupo) associados a sub-espacos vizinhos são

representados pelos vetores de pesos relacionados a neurônios vizinhos. A esta propriedade denomina-se preservação da vizinhança topológica do espaço dos vetores de entrada.

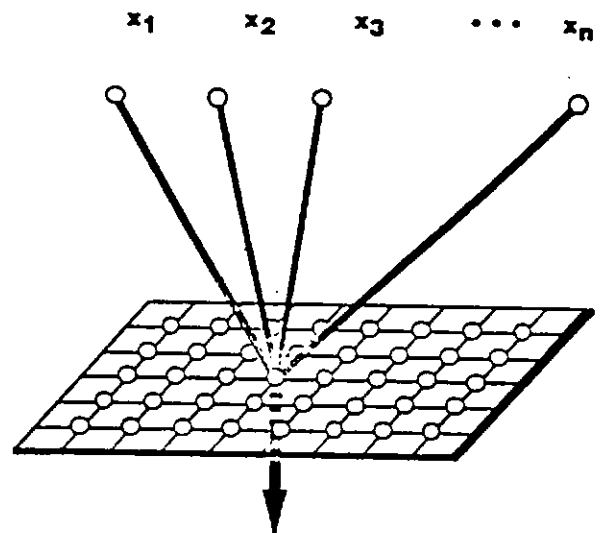


Figura 3: Mapa Auto-Organizável de Kohonen.

O número de neurônios não depende da dimensão do vetor de entrada, nem do tamanho do conjunto de treinamento. É importante notar porém, que uma escolha inadequada da arquitetura da RN, com um número pequeno de neurônios, pode comprometer a capacidade de representação de agrupamentos.

Na fase de treinamento, os padrões de entrada são apresentados à rede aleatoriamente e somente o neurônio cujo vetor de pesos associado é o mais próximo (distância Euclidiana) do vetor de entrada será excitado. O vetor de pesos deste neurônio declarado "vencedor" é então atualizado, sendo atualizados também os vetores de pesos correspondentes a seus vizinhos. Esta atualização é realizada de acordo com uma função de vizinhança $\alpha(t, i, c)$, sendo esta função monotonicamente decrescente no tempo t e no grau de vizinhança dos neurônios i e c . O treinamento pode ser descrito pelos seguintes passos:

- 1) $t = 0 \rightarrow$ inicializar os pesos w_i aleatoriamente.
- 2) Escolher aleatoriamente um vetor de entrada x do conjunto de treinamento X .
- 3) Determinar o neurônio c cujo vetor de pesos é o mais próximo do vetor de entrada.

$$\|w_c(t) - x\|_2 = \min \|w_i(t) - x\|_2$$

para todo i

- 4) Atualizar o vetor de pesos w_c do neurônio c e os vetores de pesos w_i de seus vizinhos.

$$w_i(t+1) = w_i(t) + \alpha(t, i, c)(x - w_i(t))$$

para todo i

- 5) Fazer $t = t + 1$.
- 6) Se $\alpha(t, i, c) > \epsilon$ voltar para o passo 2.
Senão \rightarrow FIM.

Pode-se verificar no algoritmo que quando o número de vetores já apresentados à rede for significativamente grande, não ocorrerá mais atualização. Neste ponto, espera-se que a rede esteja treinada e seja capaz de formar agrupamentos que permitam uma correta classificação dos novos casos a serem apresentados. Convém ressaltar que para isto é necessário que o conjunto de treinamento seja representativo do problema em estudo.

Encerrada a fase de ajuste dos pesos da rede, esta pode ser testada através da apresentação de novos vetores de entrada, observando-se os neurônios excitados em cada caso para posteriormente análise dos agrupamentos formados. Este procedimento pode ser descrito pelos seguintes passos:

- 1) Selecionar um vetor de entrada x .
- 2) Selecionar w_c tal que

$$\|w_c(t_{max}) - x\|_2 = \min_i \|w_i(t_{max}) - x\|_2$$

sendo $w_c(t_{max})$ os pesos da rede treinada.

3.2 Redes Neurais em Sistemas de Potência

A primeira aplicação de redes neurais na solução de problemas de sistemas de potência foi em análise de estabilidade (reconhecimento de padrões) no início da década de 70 [23]. Técnicas de reconhecimento de padrões procuram definir uma sistemática para a seleção de variáveis (ou combinação de variáveis) que sejam significativas para classificar uma dada situação. Os classificadores descritos nas primeiras aplicações relacionadas à análise de estabilidade proporcionavam como resposta as classes "estável" e "instável". Perceptrons sem camadas intermediárias eram utilizados para tal fim. O desempenho insatisfatório destes classificadores, devido a limitações tal como a hipótese de separabilidade linear entre classes, levou ao quase abandono desta área de pesquisa. Ainda na década de 70 foi proposta a utilização do modelo neural GMDH (Group Method of Data Handling) para o problema de previsão de carga [24], e em 1982 o uso de memórias associativas na análise de segurança foi estudado [25].

O renascimento do interesse em RN ocorreu em 1986 por ocasião da proposição do algoritmo de retro-propagação de erro para treinamento de perceptrons de múltiplas camadas. Desde então, as RN passaram a ser utilizadas na síntese de mapeamentos complexos. A nova onda de interesse em RN e o surgimento nos anos 90 de hardware para RN destinado a realização de tarefas complexas têm dado um novo impulso na pesquisa de suas aplicações em engenharia de potência.

Redes neurais têm sido propostas para a solução de problemas que não podem ser tratados apropriadamente por métodos analíticos ou por SE. Previsão de carga [26], análise de segurança [27], diagnóstico de faltas [28] e falhas em componentes do sistema [29], estimação de estado [30, 31], projeto de estabilizadores [32], comissionamento de unidades geradoras [33] e controle

de máquinas elétricas [34] estão entre os principais problemas de engenharia de potência tratados via RN. Estes problemas possuem geralmente uma ou mais das seguintes características:

- (a) complexidade de modelagem → previsão e modelagem de carga, projetos de estabilizadores e estimação de estado;
- (b) natureza combinatória → comissionamento de unidades geradoras e análise de segurança;
- (c) inexistência de algoritmos numéricos que satisfaçam o requisito de tempo-real → análise de segurança;
- (d) tratamento de grande quantidade de dados em tempo-real → diagnóstico de faltas.

Em síntese, as RN vêm sendo utilizadas predominantemente em problemas de regressão, classificação e otimização combinatória. A seguir estão listados os benefícios da utilização de RN em alguns dos problemas mencionados anteriormente:

- (a) previsão de carga → modelos neurais plásticos podem captar interações não-lineares complexas entre variáveis, que melhoram a precisão e a robustez das previsões;
- (b) análise de segurança → RN proporcionam soluções em tempo-real;
- (c) comissionamento de unidades geradoras → RN permitem a incorporação de qualquer tipo de restrição, além de possibilitar a obtenção rápida de soluções;
- (d) estimação de estado → RN permitem a filtragem correta das medidas obtidas pelo SCADA durante problemas de comunicação, e podem ainda estimar valores de pseudo-medidas nas áreas não-observáveis do sistema;
- (e) projeto de estabilizadores → RN possuem a capacidade de tratar não-linearidades arbitrárias, ao contrário das técnicas tradicionais de controle.

Mais de 500 artigos descrevendo aplicações de RN em engenharia de potência foram publicados nos últimos 6 anos. Um trabalho revisando 150 publicações na área [35] é a referência mais abrangente sobre o assunto até o momento. Um levantamento recente estabeleceu a seguinte distribuição entre as aplicações:

- Previsão de carga	20.5%
- Análise de segurança	19%
- Controle	15%
- Diagnóstico de faltas	14%
- Identificação de sistemas	12%
- Despacho econômico e comissionamento de unidades	7%
- Processamento de alarmes	2.5%
- Proteção	2%
- Outras	8%

4. CONCLUSÕES

A diferença fundamental entre os métodos analíticos e a abordagem via sistemas inteligentes é que enquanto os primeiros modelam o sistema de interesse para em

seguida utilizar uma forma algorítmica de solução, os sistemas inteligentes modelam diretamente um processo de solução. A "base de conhecimentos" de um sistema inteligente representa a maneira pela qual o problema é resolvido, e não uma descrição do mesmo.

A principal desvantagem da utilização de RN em comparação aos SE é a natureza "opaca" do mapeamento sintetizado pelas primeiras. Ao contrário dos SE, as RN não têm a capacidade de justificar as suas respostas. Além disso, as RN podem não explorar regras heurísticas que um operador poderia utilizar para resolver um problema. Sistemas especialistas são apropriados para codificar estas regras heurísticas. Consequentemente, existe a necessidade de se combinar o conhecimento específico dos operadores com as RN formando assim os chamados sistemas inteligentes híbridos.

REFERÊNCIAS

- [1] B. Valiquette, G. Lambert-Torres & D. Mukhedkar - "An expert system based diagnosis and advisor tool for teaching power operation emergency control strategies", IEEE Transactions on Power Systems, Vol. 6, No. 3, August 1991.
- [2] L. Galbraith, M. Al-Najjar & A.J.G. Babu - "Expert system in engineering", IEEE AES Magazine, February 1988.
- [3] EPRI Report, "Artificial intelligence technologies for power system operation", Report Np. EL-4323.
- [4] CIGRÉ Task Force 39-03 - "Exploring user requirements of expert systems in power system operation and control", ELECTRA, No. 146, February 1993.
- [5] CIGRÉ Task Force 38-06-03 - "Practical use of expert systems in planning and operation of power systems", ELECTRA, No. 146, February 1993
- [6] M. Akimoto *et al.* - "Development of an expert system for operation planning of bulk power system", CIGRÉ 1992 Session SC38, Paris, August 1992.
- [7] Z.Z. Zhang, G.S. Hope & O.P. Malik - "Expert systems in electric power systems - A bibliography survey", IEEE Transactions on Power Systems, Vol. 4, No. 4, October 1989
- [8] C.C. Liu *et al.* - "Special issue on knowledge based systems in electric power systems", Proceedings of the IEEE, May 1992.
- [9] F. Hayes-Roth, D.A. Waerman & D.B. Lenat - *Building Expert Systems*, Addison-Wesley, 1984.
- [10] G. Lambert-Torres & D. Mukhedkar - "An expert system for AC/DC substation grounding design of a power system", 2nd Symposium on Expert Systems Application to Power Systems, Seattle, August 1989.
- [11] CIGRÉ Task Force 38-06-03 - "Practical use of expert systems in planning and operation of power systems - Phase II - Maintenance and testing procedures", a ser publicado na ELECTRA.
- [12] G. Lambert-Torres, B. Valiquette & D. Mukhedkar - "Bus load forecasting using fuzzy techniques", CEA Engineering and Operating Transactions, Vol. 29, 1990.
- [13] G. Lambert-Torres *et al.* - "A hybrid expert system for power system protection and control", Proceedings of 4th Symposium on Expert Systems Application to Power Systems, Melbourne, January 1993.
- [14] H. Tanaka *et al.* - "Research & development on expert systems applications to power systems in Japan", Plenary Session Paper at 3rd. Symposium on Expert Systems Application to Power Systems, Tokyo, April 1991.
- [15] B.F. Wollemborg & T. Sakaguchi - "Artificial intelligence in power system operations", Proceedings of IEEE, Vol. 75, No. 12, December 1987.
- [16] J. Hertz, A. Krogh & R.G. Palmer - *Introduction to the Theory of Neural Computing*, Addison-Wesley, 1991.
- [17] Y.H. Pao - *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, 1989.
- [18] D.E. Rumelhart, G.E. Hinton & R.J. Williams - "Learning internal representations by error propagation", in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, MIT Press, 1986, pp. 318-362.
- [19] A.P. Alves da Silva, V.H. Quintana & G.K.H. Pang - "Neural networks for topology determination of power systems", Proceedings of the 1st Forum on Applications of Neural Networks to Power Systems, Seattle, July 1991, pp. 297-301.
- [20] A.P. Alves da Silva, A.M. Leite da Silva, J.C.S. de Souza & M.B. Do Coutto Filho - "State forecasting based on artificial neural networks", Proceedings of the 11th Power Systems Computation Conference, Avignon, August 1993, pp. 461-467.
- [21] A.P. Alves da Silva & V.H. Quintana - "Pattern analysis in power system state estimation", a ser publicado no International Journal of Electrical Power & Energy Systems.
- [22] T. Kohonen - "Self-organized formation of topologically correct feature map", Biological Cybernetics, Vol. 43, 1982, pp. 59-69.
- [23] C.K. Pang, F.S. Prabhakara, A.H. El-Abiad & A.J. Koivo - "Security evaluation in power systems using

- pattern recognition", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-93, May/June 1974, pp. 969-976.
- [24] T.S. Dillon, K. Morsztyn & K. Phua - "Short term load forecasting using adaptive pattern recognition and self-organizing techniques", Proceedings of the 5th Power Systems Computation Conference, Cambridge, 1975, pp. 1-16.
- [25] EPRI - "Feasibility of using associative memories for static security assessment of power system overloads", Final Report, EL-2343, Project 1047-2, April 1982.
- [26] Y.-Y. Hsu & C.-C. Yang - "Design of artificial neural networks for short-term load forecasting. Part I: Self-organising feature maps for day type identification. Part II: Multilayer feedforward networks for peak load and valley load forecasting.", IEE Proceedings - C, Vol. 138, September 1991, pp. 407-418.
- [27] M. Aggoune, M.A. El-Sharkawi, D.C. Park, M. Damborg & R.J. Marks II - "Preliminary results on using artificial neural networks for security assessment", IEEE Transactions on Power Systems, Vol. 6, May 1991, pp. 890-896.
- [28] A.P. Alves da Silva, A.H.F. Insfrán, P.M. da Silveira & G. Lambert Torres - "Fault localization estimation via associative memories", a ser publicado nos Proceedings of the International Conference on Intelligent System Application to Power Systems, Montpellier, September 1994.
- [29] T. Baumann, A. Germond & D. Tschudi - "Impulse test fault diagnosis on power transformers using Kohonen's self-organizing neural network", Proceedings of the 3rd Symposium on Expert System Application to Power Systems, Tokyo, April 1991, pp. 642-647.
- [30] A.P. Alves da Silva, V.H. Quintana & G.K.H. Pang - "Solving data acquisition and processing problems in power systems using a pattern analysis approach", IEE Proceedings - C, Vol. 138, July 1991, pp. 365-376.
- [31] A.P. Alves da Silva, V.H. Quintana & G.K.H. Pang - "Associative memory models for data processing", International Journal of Electrical Power & Energy Systems, Vol. 14, February 1992, pp. 23-32.
- [32] Y. Zhang, O.P. Malik, G.S. Hope & G.P. Chen - "Application of an inverse input/output mapped ANN as a power system stabilizer", IEEE Power Engineering Society Winter Meeting, Columbus, January 1993, Paper 93WM 130-5 EC.
- [33] H. Sasaki, M. Watanabe, J. Kubokawa, N. Yorino & R. Yokoyama - "A solution method of unit commitment by artificial neural networks", IEEE Transactions on Power Systems, Vol. 7, August 1992, pp. 974-981.
- [34] L.E. Borges da Silva, A.P. Alves da Silva, G. Lambert Torres & G. Roy - "An alternative neural network training algorithm for real time control", Proceedings of the IEEE International Symposium on Industrial Electronics, Santiago, May 1994, pp. 72-76.
- [35] The CIGRÉ Task Force 38-06-06 Members & A.P. Alves da Silva - "Artificial neural networks for power systems - a literature survey", International Journal on Engineering Intelligent Systems, Vol. 3, December 1993, pp. 133-158.