

Previsão de Falhas em Manutenção Industrial Usando Redes Neurais

Rubião Gomes Torres Júnior¹
rubtor@attglobal.net

Maria Augusta Soares Machado,
Dsc.¹
mmachado@ibmecrj.br

Jorge Muniz Barreto²
barreto@inf.ufsc.br

1 Faculdades Ibmec/RJ - Rio de Janeiro, RJ
2 UFSC – Florianópolis, SC

Área temática: LG - Logística

RESUMO

A aplicação de dois modelos de previsão de falhas em manutenção industrial utilizando Redes Neurais Artificiais (RNA), é o objetivo manifesto deste trabalho. É formada uma série temporal diária, com base num levantamento de 05 anos de intervenções de paradas para manutenções. Assim objetiva-se introduzir estes modelos e verificar sua capacidade de previsão de paradas nos sistemas. Estes dados geram importantes informações que apóiam decisões gerenciais, para que no futuro seja possível intervir num tempo ótimo, visando prolongar o tempo de operação das plantas e, conseqüentemente, aumentar sua disponibilidade. Os resultados obtidos neste trabalho demonstram a aplicabilidade das Redes Neurais nas previsões de falhas. Confirma-se, ainda, a habilidade das RNAs em efetuar previsão em um conjunto de dados com forte componente de não-linearidade, onde outras técnicas estatísticas tem se mostrado menos eficazes.

1. INTRODUÇÃO

A aplicação de modelos de previsão de falhas em manutenção industrial utilizando Redes Neurais Artificiais (RNA), é o objetivo manifesto deste trabalho.

É uma característica do ambiente industrial contemporâneo, a acirrada competição, o que leva as empresas a procurarem minimizar seus custos. Neste cenário torna-se de extrema importância a coleta e o tratamento de dados de manutenção, visando a real necessidade de reparo dos equipamentos e sistemas da planta.

Atualmente a operação prolongada e eficaz dos sistemas produtivos é exigência vital. Logo se impõe a necessidade de conhecer e controlar quaisquer possíveis falhas que venham a comprometer a missão produtiva. A meta passa a ser o prolongamento da atividade do sistema a plena carga e de modo contínuo, no tempo necessário, sem que seja afetado por defeitos em suas partes integrantes.

A manutenção preditiva apresenta a vantagem de prever o estado dos componentes, informando quando o mesmo apresentará falha, dentro de boa margem de certeza. Para tal, há necessidade de executar aquilo que é chamado de “diagnóstico”. Através do mesmo, ficar-se-á sabendo qual o estado de determinado ou determinados componentes, quando os mesmos apresentarão falhas e como programar a sua substituição antes da ocorrência da situação crítica que leva ao rompimento e conseqüente parada do equipamento (Nepomuceno, 1989).

Neste trabalho é formada uma série temporal diária, com base num levantamento de 05 anos de intervenções de paradas para manutenções em diversos sistemas produtivos na

área de acabamento da Petroflex Ind. e Com.s.a., a maior produtora de borrachas sintéticas da América Latina e uma das maiores do mundo. Assim objetiva-se introduzir modelos baseados em redes neurais e verificar sua capacidade de previsão de paradas nos sistemas, para que, no futuro, seja possível intervir, num tempo ótimo, antes que os sistemas falhem, de forma a prolongar o tempo de operação e, conseqüentemente, aumentar sua disponibilidade.

Modelos baseados em redes neurais vêm sendo utilizados com sucesso na previsão de séries temporais onde há um forte componente de não linearidade. Este fato pode ser explicado pela capacidade que uma rede neural tem de funcionar como um “aproximador universal” de funções contínuas (Cybenko, 1989). Esta abordagem vem sendo utilizada com bastante frequência em séries temporais financeiras (Klimasauskas, 1992), (Draisma, Kaashoek & Dijk, 1993), (Cheng, 1996), (Refenes, 1994), bem como em previsões de consumo de cargas elétricas, (Park, 1991), (Lin, 1996), onde outras técnicas estatísticas têm se mostrado menos eficazes.

2. REDES NEURAIS

2.1 DEFINIÇÃO

As Redes Neurais Artificiais (RNA) são processos de informações tecnológicas inspiradas em estudos do cérebro e do sistema nervoso (Trippi; Turban, 1993).

Haykin (1994) define RNA como:

“As RNAs são processadores, massivamente paralelos e distribuídos, que têm uma propensão natural para armazenar o conhecimento proveniente da experiência e torná-lo útil. Desta forma, assemelhando-se ao cérebro humano em dois aspectos:

1. conhecimento é adquirido pela rede através de um processo de aprendizado e
2. as intensidades das conexões entre neurônios, conhecidas como pesos sinápticos, são usadas para armazenar o conhecimento”.

Motivados pelo desejo de se entender e simular o funcionamento do cérebro, os modelos de redes neurais vêm se desenvolvendo, ao longo dos anos, através de generalizações de modelos matemáticos. Devido à sua capacidade de aprender através de exemplo e de generalizar a informação aprendida, as RNAs estão propondo soluções interessantes em diversas áreas como a administração de empresas, economia, medicina, agronomia, engenharia, etc.

O modelo matemático de um neurônio artificial foi primeiramente idealizado pelos pesquisadores W. S. McCulloch e W. H. Pitts (Zurada, 1992) no ano de 1943. a principal limitação do neurônio de McCulloch e Pitts é que ele foi proposto com pesos fixos, não ajustáveis, só conseguindo, assim, implementar funções linearmente separáveis.

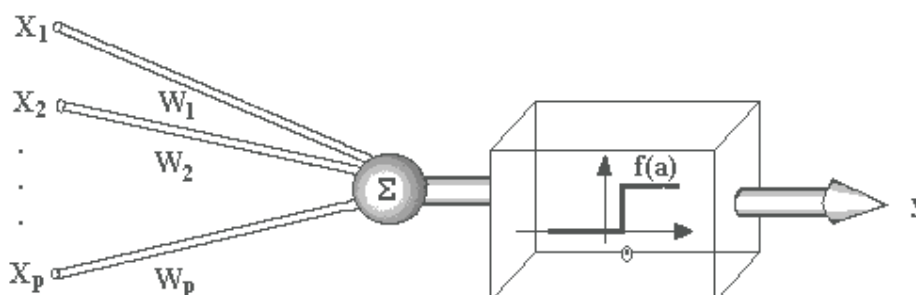


Figura 1 - Neurônio de McCulloch

Na figura 1, os sinais de entrada são representados na figura por X_p , e devem chegar ao neurônio ao mesmo tempo para que sejam processados e enviados para a saída.

Os pesos, representados por W_p , são atributos de grande relevância no neurônio artificial, pois representam o grau de importância (intensidade) que determinada entrada possui em relação ao neurônio em questão. O papel do peso é multiplicar o sinal X_p na entrada da sinapse conectada a um neurônio.

Matematicamente, os pesos são vistos como um vetor de valores $[W_1, W_2, \dots, W_p]$ para um neurônio, ou uma matriz de pesos, coleção de vetores, para um conjunto de neurônios.

A função soma adiciona as entradas ponderadas pelos pesos respectivos. O sinal de excitação de um neurônio é o resultado do somatório do produto dos sinais de entrada representados pelo vetor $[X_1, X_2, \dots, X_p]$.

Na seqüência, o processo encontra a função de ativação $f(a)$ juntamente com o limiar (threshold) θ . Se o valor da resultante da soma entre o produto dos sinais de entrada pelos respectivos pesos atingiu o limiar, o mesmo é repassado; caso contrário, o sinal não será transferido. A função de ativação funciona como um limitante à amplitude da saída do neurônio, ou seja, a entrada é normalizada dentro de um intervalo fechado.

Finalmente y é a saída do neurônio.

No estado da arte da computação neural não é necessário ficarmos limitados ao nosso entendimento das redes neurais biológicas. A despeito da extensiva pesquisa da neurobiologia e psicologia, importantes questões permanecem sobre como o cérebro e a mente trabalham. Esta é a principal razão para que os modelos de computação neural não sejam muito limitados aos atuais sistemas biológicos, (Turban; Trippi, 1993).

2.2 APRENDIZAGEM

Aprendizagem é o processo pelo qual os parâmetros de uma rede neural são adaptados por meio de um estímulo contínuo do ambiente, no qual a rede está operando, sendo o tipo específico de aprendizagem realizada definido pela maneira particular como ocorrem os ajustes realizados nos parâmetros (Haykin, 1994).

Sem dúvida, o grande atrativo das redes neurais está na capacidade de aprendizagem, onde a rede extrai informações relevantes de padrões apresentados gerando uma representação própria do problema.

O processo de aprendizagem pode ser simplificado da seguinte forma (Trippi; Turban, 1993):

“As Redes Neurais Artificiais aprendem com seus erros. Normalmente o processo de aprendizagem (ou treinamento) envolve três fases:

- 1 - Compute a saída;
- 2 - Compare a saída com as respostas desejadas;
- 3 - Ajuste os pesos e repita o processo.

Usualmente o processo de aprendizado começa pelos pesos randomizados. A diferença entre a saída atual (Y ou Y_t) e a saída desejada (Z) é chamada de Δ . O objetivo é minimizar o Δ (ou no melhor reduzi-lo a zero). A redução do Δ é feita pela mudança incremental dos pesos”.

2.3 O ALGORITMO “BACKPROPAGATION”

As redes do tipo MLP tem sido utilizadas com sucesso para a solução de vários problemas envolvendo altos graus de não linearidade. Seu treinamento é do tipo supervisionado e utiliza um algoritmo muito popular chamado retro-propagação do erro (error backpropagation). Este algoritmo é baseado numa regra de aprendizagem que “corrige” o erro durante o treinamento (Haykin, 1994).

O algoritmo “backpropagation” foi criado por Rumelhard, Hinton e Williams em 1986 (Zurada, 1992), (Haykin, 1994), a partir da generalização da regra de aprendizado “Widrow-Hoff”, que fora introduzida em 1960-1962 para redes do tipo “feedforward perceptron”. A regra de aprendizado “Widrow-Hoff”, também conhecida como “Regra Delta” – LMS (minimização do erro médio quadrático) – que ajusta os pesos das conexões entre os neurônios da rede de acordo com o erro, ou seja, esta regra tem como objetivo encontrar um conjunto de pesos e polarizações que minimizem a função erro,

$$(1) \quad E = \frac{1}{2} \sum_{p=1}^R \sum_{i=1}^S (y_{p,i} - y'_{p,i})^2$$

onde,

R = número de padrões ou vetores de entrada;

S = número de neurônios de saída – dimensão do vetor de saída;

$y_{p,i}$ = saída desejada no i-ésimo neurônio, quando p-ésimo padrão é apresentado;

$y'_{p,i}$ = saída obtida pela rede no i-ésimo neurônio, quando o p-ésimo padrão é apresentado.

A alteração dos pesos $W_{i,j}$ da regra “Widrow-Hoff” é calculada da seguinte maneira:

$$(2) \quad \Delta W_{i,j} = - \eta \partial E / \partial W_{i,j}$$

onde, η = taxa de aprendizado e $\partial E / \partial W_{i,j}$ é a derivada parcial do erro em relação ao peso da respectiva conexão – gradiente. A principal restrição na minimização do erro no sentido do gradiente decrescente é que a função de transferência do neurônio tem que ser monotônica e diferenciável em qualquer ponto.

O algoritmo “Backpropagation” (BP) refere-se a uma regra de aprendizagem que consiste no ajuste dos pesos e polarizações da rede através da retropropagação do erro encontrado na saída. A minimização é conseguida realizando-se continuamente – a cada interação – a atualização dos pesos e das polarizações da rede, no sentido oposto ao do gradiente da função no ponto corrente, ou seja, proporcionalmente ao negativo da derivada do erro quadrático em relação aos pesos correntes. Trata-se, portanto, de um algoritmo de treinamento supervisionado, determinístico, de computação local, e que implementa o método do gradiente decrescente nas somas dos quadrados dos erros.

A topologia da arquitetura da rede que utiliza esta regra de aprendizagem é formada, geralmente, por uma ou mais camadas escondidas (intermediárias) de neurônios não-lineares (com função de propagação sigmoideal) e uma camada de saída de neurônios lineares. Devido a grande difusão da arquitetura da rede a que esta regra de aprendizagem se aplica, é comum referir-se a ela com o nome da própria regra de aprendizagem, ou seja, rede BP. Redes BP com polarizações com, no mínimo, uma camada intermediária de neurônios lineares na saída, são teoricamente capazes de realizar a aproximação de qualquer função matemática, sendo ainda bastante utilizadas na associação e classificação de padrões.

3 ESPAÇOS AMOSTRAIS

Na Petroflex, atualmente são acompanhados 23(vinte e três) motivos de paradas, ou seja, equipamentos ou sistemas que, no caso de falha, param o sistema produtivo.

Devido a natureza deste trabalho, optou-se por escolher apenas 01 (um) motivo relevante de parada de sistemas para verificar se o comportamento diário do mesmo, pode ser previstos por redes neurais.

Com o intuito de verificar o comportamento das séries temporais relativas as quantidades de horas paradas diariamente para cada motivo escolhido, foram organizados

os dados produzindo o somatório diário ao longo dos 05 (cinco) anos, ou seja, uma série de 1826 dias corridos com suas respectivas horas paradas.

Assim sendo, segue o gráfico do motivo de parada estudado neste trabalho, feito no software Matlab®, mostrando o dia que ocorreram falhas e quantas horas permaneceram inoperante.

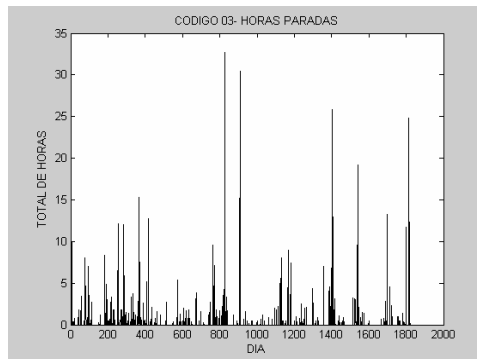


Figura 2 – Gráfico da série temporal, horas reais – código 03

4. CONSTRUÇÃO DA REDE

Após a normalização, os dados relativos a cada código foram divididos em 03(três) blocos para diferentes finalidades: treinamento, validação e teste. Tendo em vista o objetivo do trabalho em foco e a quantidade de dados de entrada para cada rodada, 1826, optou-se por dividi-los em três grupos iguais, garantindo assim uma consistente massa de dados para teste. Sendo assim, os valores das séries temporais ficaram distribuídos da seguinte forma:

- 01 a 608 – Treinamento.
- 609 a 1216 – Validação.
- 1217 a 1824 – Teste.

O software utilizado foi o Matlab® através da Neural Network Toolbox. Este programa utiliza para redes feedforward, o algoritmo backpropagation usando gradiente decrescente nomeado por “trainlm”. O computador utilizado foi um AMD Athlon 2400, 1.99Ghz.

Visando acelerar a fase de treinamento, os dados foram normalizados (Haykin, 2001), também proporcionando dados de entrada entre 0 e 1. A função de ativação escolhida foi a Logística Sigmoidal, obtendo-se saídas também compreendidas entre 0 e 1 facilitando as medidas de acurácia.

Na fase de treinamento e validação, o processo somente é interrompido por dois mecanismos pré-determinados. O primeiro diz respeito a um padrão mínimo de erro na saída, determinado no caso em foco, na fase de retropropagação, pelo erro quadrático médio, e o segundo mecanismo de interrupção é devido ao número máximo de interações permitidas (epochs). Foi pré-definido que para todas as redes testadas seriam fixados dois valores para o erro médio, 0,01 e 0,005 e um limite de 30.000 interações.(Limites inicialmente definido, tendo como base alguns testes realizados com os dados e por serem valores de aceitável precisão no caso em questão).

Para a taxa de aprendizado e o momentum, dois importantes parâmetros que auxiliam na aceleração do treinamento, optou-se por utilizar o “default” do Matlab®, o ‘traingd’, ou seja, automaticamente o programa altera e adapta a taxa de aprendizado e os valores do momentum a medida que o erro médio quadrático diminui.

No tocante à arquitetura da rede, foi utilizada apenas uma camada oculta, tendo em vista que, esta topologia mais simples, já é suficiente para possibilitar a aproximação de funções contínuas e não-lineares das entradas. Assim, com o objetivo de encontrar a melhor

arquitetura, ou grupo de redes que adquiram as melhores capacidades de aprendizado, partiu-se de 01(um) neurônio na camada oculta, aumentando este número sem limites iniciais pré-determinados e verificando-se o comportamento das redes, tanto na fase de treinamento e validação acompanhando o parâmetro de erro, quanto na fase de resultados dos testes realizados.

Em relação ao parâmetro de erro, foi utilizado, como técnica para mensuração da acurácia do modelo de previsão, a relação entre os valores da saída da rede e de treinamento, para isto, foi plotado, a saída da rede e o conjunto de validação (valor esperado), em gráficos superpostos e em cores diferentes com o intuito de obter preliminarmente uma análise visual, e as diferenças, os valores dos erros encontrados, foram mostrados graficamente. Os menores valores de erros encontrados determinam as melhores redes.

Com base neste parâmetro de avaliação foram realizadas simulações com o intuito de separar um primeiro conjunto de redes, que demonstrassem genericamente, as topologias (números de neurônios na camada escondida) mais viáveis. Assim foram separadas para uma segunda fase de treinamentos seis arquiteturas com 02, 03, 04, 05, 10 e 20 neurônios. Nesta fase, assim como nas anteriores, cada arquitetura corresponde a duas redes, ou seja, uma para erro médio quadrático de 0,01 e outra de 0,005 e as duas com máximo de 30.000 interações, inicialmente, e se fosse verificado a necessidade de mais interações, novas simulações seriam realizadas. Para cada configuração foram repetidas no mínimo 10(dez) vezes cada arquitetura. Não houve preocupação com o tempo dos treinamentos.

Ainda com o intuito de verificar a capacidade de previsão e adequação a valores futuros, foi comparada a saída da rede ao conjunto de teste; esta diferença também produziu um erro que foi mostrado graficamente. Posteriormente a rede ainda foi simulada para este conjunto de teste e a saída desta rede simulada comparada com o valor esperado, ou seja, ao próprio conjunto de teste.

Numa nova etapa, mantendo-se o mesmo número de neurônios na camada intermediária, para a mesma série temporal, os conjuntos de treinamento, validação e teste, foram reorganizados de sorte a fornecerem para uma nova rede, com três entradas distintas em instantes denominados t-1, t-2 e t-3, seguidos dos respectivos valores de validação e teste. A intenção é a de verificar mais uma vez a capacidade de previsão da rede neural para os mesmos dados, porém em instantes diferentes. Assim, para esta nova arquitetura e para cada código, foram testadas várias redes, analisadas as performances na fase de treinamento e validação e comparadas as saídas com os dados de teste, configurando um erro, posteriormente analisado. Esta mesma rede também foi simulada para os dados de teste e a diferença entre esta nova saída e o próprio dado de teste mostrado graficamente.

5. RESULTADOS

Preliminarmente, no decorrer das simulações realizadas, verificou-se a necessidade de, para a mesma arquitetura, simular várias vezes, com o intuito de encontrar o melhor grupo de resultados e tentar buscá-los em simulações futuras.

Código 03 (Sistema de Ácido Soro e Água Mãe).

Melhor topologia: 03 (três) neurônios na camada escondida.

Rede iniciada para 30.000 interações.

Limite de Erro Médio Quadrático – 0,005

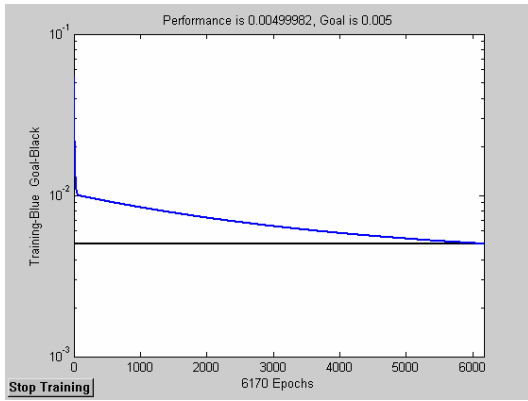


Figura 3– Performance da rede neural na fase de treinamento e validação

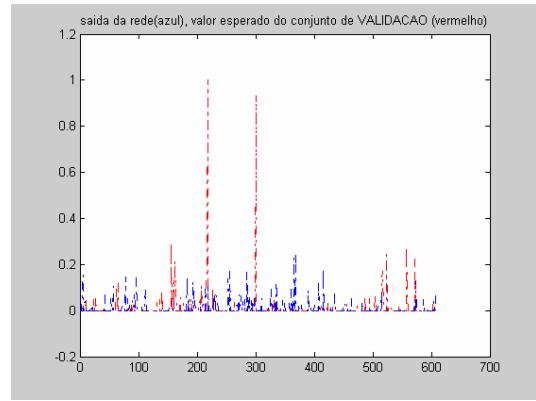


Figura 4– Gráficos superpostos da saída da rede e os dados para validação

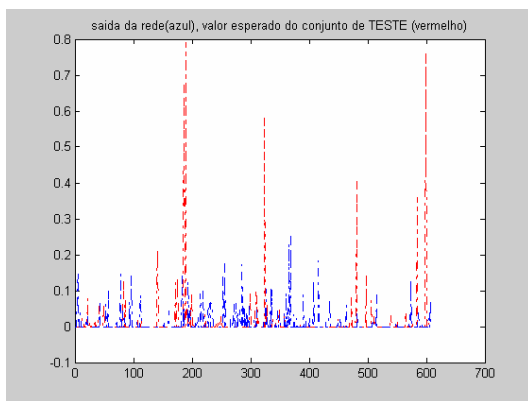


Figura 5 – Gráfico do erro da rede para o conjunto de validação

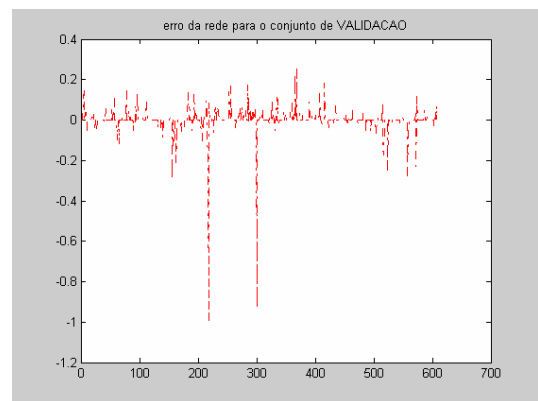


Figura 6 – Gráficos superpostos da saída da rede e os dados de teste - código 03

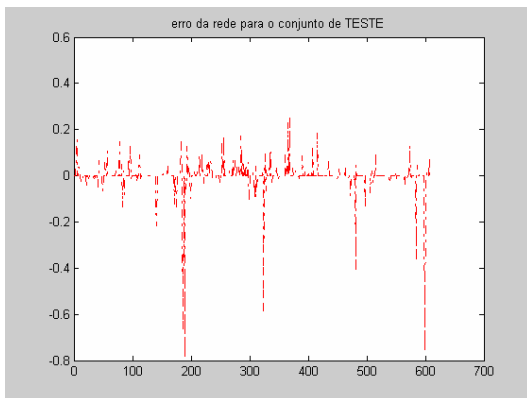


Figura 7 – Gráfico do erro da rede para o conjunto de teste

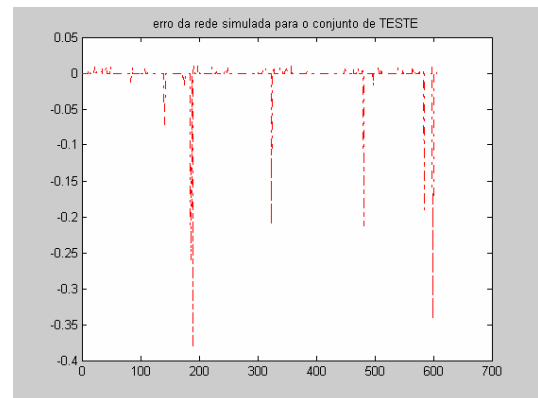


Figura 8 – Gráfico do erro da rede simulada para o conjunto de teste

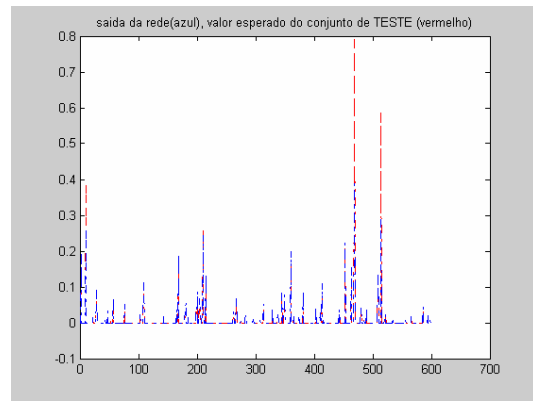
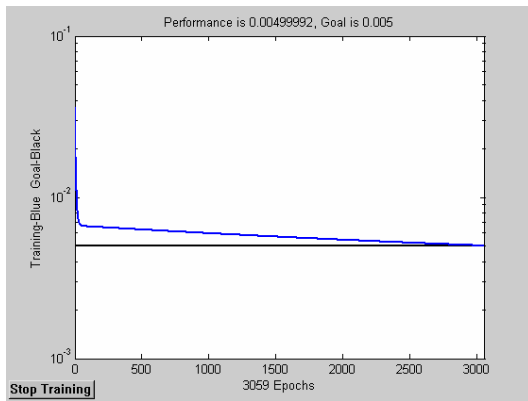


Figura 9 – Performance da nova rede com 03 entradas, fase de treinamento e validação

Figura 10 – Gráficos superpostos da saída da rede e do conjunto de teste

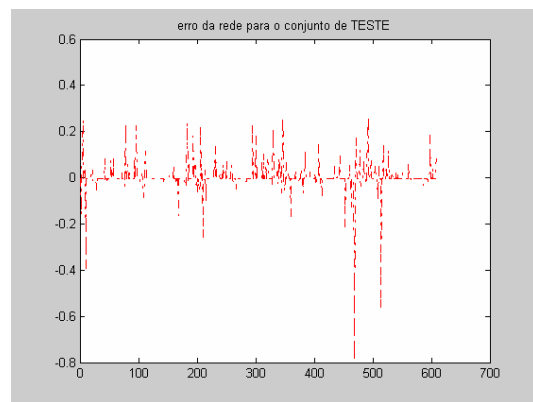
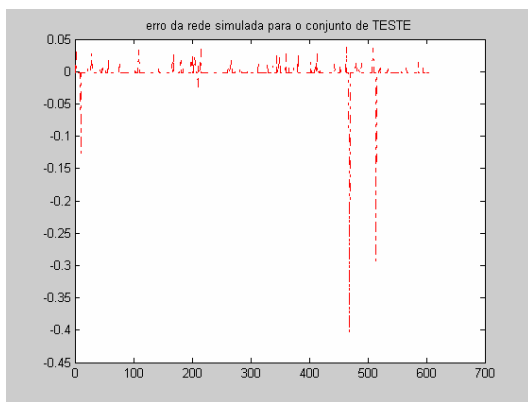


Figura 11 – Gráfico do erro da saída da rede em relação ao conjunto de teste

Figura 12 – Gráfico do erro da saída da rede simulada em relação ao conjunto de teste

6. CONCLUSÕES

Os resultados obtidos neste trabalho demonstram a aplicabilidade das Redes Neurais nas previsões de paradas para a manutenção na área industrial da Petroflex. Confirma-se, ainda, a habilidade das RNAs em efetuar previsão em um conjunto de dados com forte componente de não-linearidade, onde outras técnicas estatísticas tem se mostrado menos eficazes.

Encontrar modelos neurais para previsões de séries temporais de falhas em sistemas, tem cada vez mais aberto espaço no campo de pesquisas, principalmente devido à demanda do mercado e trata-se, sem dúvida, de uma técnica que irá evoluir bastante na área de manutenção industrial, visando subsidiar importantes tomadas de decisões gerenciais. Conta ainda com a possibilidade de ser inserida nos sistemas computacionais que atualmente já fornecem, como no caso em estudo, informações básicas, colhidas na sua operação diária e ter um custo substancialmente menor em relação a outras técnicas de manutenção preditiva.

Técnicas preditivas como as aqui apresentadas, trabalhando em conjunto com o planejamento de manutenção, podem, a médio prazo, se cuidadosamente implementadas e acompanhadas, fornecerem um substancial aumento de horas operacionais disponíveis

Uma recomendação para trabalhos posteriores, seria a de um modelo particularizado para um determinado conjunto de dados, que diariamente alimentado, encontre as melhores redes neurais que se adaptam a estes dados e forneça a previsão da data de possíveis

paradas dos sistemas, bem como a quantidade de horas em que este conjunto permanecerá inoperante.

7. REFERÊNCIAS

- Braga, A.; Carvalho, A.; Ludermir, T. (2000) *Redes Neurais Artificiais: Teoria e Aplicações*, Livros Técnicos e Científicos, Rio de Janeiro.
- Cheng, W. Wagner, L. Lin, C. (1996) Forecasting the 30-year U.S. Treasury Bond with a System of Neural Networks. *NeuroVe\$t Journal*, January/February, p 13.
- Cybenko, G. (1989) Aproximation by Superpositions of a Sigmoidal Function. *Math. Control Signal Systems*, p304-314.
- Draisma, G., Kaashoek, J. F. e Dijk, H. K. (1994) A Neural Network Applied to Economic Time Series. Erasmus University Rotterdam, Econometric Institute. Report 9474B.
- Haykin, Simon. (2001) *Redes Neurais – Princípios e Práticas*. 2. ed. Porto Alegre : Bookman.
- Haykin, Simon. (1994) *Neural Networks a Comprehensive Foundation*. New York: Macmillan Collge Publishing.
- Klimasauskas, C. C. (1992) Accuracy and Profit in Trading Systems. *Advanced Technology for Developers*. pp 10-14.
- Lin, K., Subbarayan, S., Shoults, R. R., Manry, M.T., Kwan, C., Lewis, E. L., Naccarino, J. (1996) Comparison of Very Short -Term Load Forecasting Techniques - *IEEE Power Systems*, Vol. 6, No 2.
- Nepomuceno, L. X. (1989) *Técnicas de Manutenção Preditiva*. Volume 1. São Paulo: Ed. Edgard Blücler.
- Park, D. C., El-Sharkawi, M. A., Marks II, L. E. (1991) Atlas and M. J. Damborg Eletric, Load Forecasting Using an Artificial Neural Network *IEEE Power Systems*, Vol. 6, No 2.
- Trippi, R. Robert; Turban, Efraim. (1993) *Neural Networks in Finance and Investing: using artificial intelligence to improve real world performace*. Salem: Probus Publishing Company.
- Zurada, J. M. (1992) *Intruducion to Artificial Neural Systems*. Boston, Mass: West Publishing Company.
- Refenes, A. N.; Zaprakis, A.; Francis, G. (1994) Stock Performance Modeling Using Neural Networks: A Comparative Study with Regression Models, vol. 7, No. 2, p. 375-388.