

8º Trabalho de Redes Neurais

Péricles Lopes Machado

Resumo—Neste trabalho, uma rede neural RBF é treinada para simular a equação de recorrência:

$$x_k = x_{k-1} + \frac{ax_{k-s}}{1 + x_{i-s}^c} - bx_{k-1} + 0.1N, \quad (1)$$

onde N é uma variável aleatória de distribuição normal com $\mu = 0$ e $\sigma^2 = 1$, $a = 0.2$, $b = 0.1$, $c = 10$ e $s = 17$.

Os resultados são comparados com os obtidos no trabalho 7.

1 A REDE NEURAL

A rede neural possuía 7 neurônios, cujos centros associados eram escolhidos de forma aleatória dentro da base de dados filtrada contendo em torno de 3 mil amostras da série x_k . A distância D , utilizada para avaliar a distância entre uma entrada e o centro associado ao neurônio, foi a euclidiana. A função de ativação utilizada foi $u_i(\vec{x}) = e^{-\beta D(\vec{x} - \vec{c}_i)}$, onde c_i é o centro associado ao neurônio i e o parâmetro livre β , nos testes realizados, foi 0, 1.

2 A GERAÇÃO DOS CONJUNTOS DE TREINAMENTO, VALIDAÇÃO E TESTE

Inicialmente, foram gerados $M = 10000$ amostra do sinal (1), no formato $v_i = (x_{i-1}, x_{i-s}, x_i)$, com $i = s + 1, \dots, M$, onde $x_n = 0$, para $n = 1, 2, \dots, s$. O conjunto de amostras S gerado foi dividido em três partes U_0, V, T , onde U_0 contém 70% do conjunto S e é filtrado para se gerar o conjunto de treinamento, V é o conjunto de validação e contém 20% do conjunto S e T é o conjunto de teste e contém 10% de S . A partição do conjunto S em U_0, V e T foi realizada de forma aleatória.

Para tornar mais eficiente o treinamento, foi realizada uma filtragem pra eliminar dados de entrada com grande semelhança. A medida de semelhança entre duas amostras de entrada, $\vec{v}_i = (x_{i-1}, x_{i-s}, x_i)$ e $\vec{v}_k = (x_{k-1}, x_{k-s}, x_k)$, foi a função dada na equação(2).

$$D(\vec{v}_i, \vec{v}_k) = \sqrt{\langle \vec{d}_{i,k}, \vec{d}_{i,k} \rangle}, \quad (2)$$

onde $\vec{d}_{i,k} = \vec{v}_i - \vec{v}_k$.

O algoritmo de filtragem é o apresentado a seguir:

- $\vec{x}_{ant} = (0, 0, 0)$.
- Define-se uma distância mínima ϵ que as amostras tem de ter com relação a x_{ant} para entrar no conjunto de treinamento U .

- Para cada elemento v_k de U_0 , verifica-se o valor de $D(v_k, x_{ant})$, se este valor for maior que ϵ , então v_k entra para o conjunto U e x_{ant} passa ser igual a v_k .

A utilização desse algoritmo, com um $\epsilon = 0.1$, permitiu reduzir o conjunto de treinamento em 61.51%.

3 O TREINAMENTO

O método de mínimos quadrados foi utilizado para ajustar os pesos das conexões entre a camada intermediária e a camada de saída. A equação (3) é utilizada para ajustar a matriz de pesos w que possui $N_{neuronios} \times N_{saidas}$ valores.

$$Gw = b, \quad (3)$$

onde G , com $G_{ij} = e^{-\beta D(x_i, c_j)}$ e $N_{amostras} \times N_{neuronios}$ valores, é uma matriz que relaciona cada entrada do banco de dados de treinamento com um neurônio j , calculando a influência do neurônio j sobre a entrada i . b é uma matriz com $N_{amostras} \times N_{saidas}$ valores contendo as saídas esperadas para cada amostra x_i apresentada. Cada amostra x_i contém M dimensões.

Pelo método de mínimos quadrados o valor de w pode ser calculado utilizando-se a *pseudo-inversa* de G .

$$G^T G w = G^T b \quad (4)$$

$$w = [G^T G]^{-1} G^T b \quad (5)$$

4 RESULTADOS

Utilizando a rede RBF foi possível reduzir o erro quadrático no teste para 0,099612, um resultado próximo ao obtido com o *backpropagation* no trabalho anterior (0,098). Mas, para se obter esse erro, o tempo de treinamento foi bem reduzido, já que o *mínimos quadrados* não sofre com problemas de mínimos locais que afetam significativamente o método *backpropagation*.

Este experimento ilustra uma característica importante da técnica de redes neurais com funções de base radial, o ajuste dos pesos é bem mais simples e é possível obter resultados muito bons em um tempo de treinamento reduzido. No caso, o treinamento se resume na filtragem da *database* e no cálculo da *pseudo-inversa* da matriz w .

A figura 1 ilustra os bons resultados obtidos.

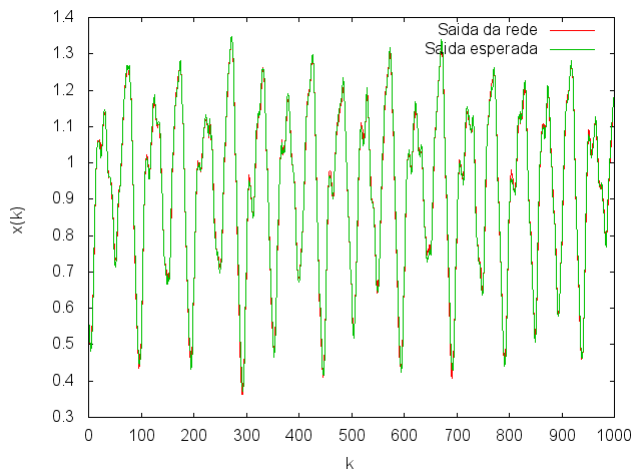


Figura 1. Comparação da saída obtida pela rede RBF com a saída esperada.

5 CONCLUSÃO

Neste trabalho, uma rede neural utilizando funções de base radial é utilizada para simular a mesma série temporal do trabalho 7. Com um tempo de processamento bem menor e utilizando a mesma *database* e a mesma quantidade de neurônios foi possível ajustar os pesos da rede para se obter um erro similar ao obtido no trabalho anterior. As principais dificuldades na utilização dessa técnica são a escolha de um valor apropriado para o parâmetro livre *beta* e a determinação dos centros associados a cada neurônio da rede.

Além disso, mostrou-se como técnicas de treinamento sem a utilização do gradiente do erro podem ser utilizadas para um ajuste rápido dos pesos de uma rede neural. Embora, a técnica requeira mais espaço de memória, isso é facilmente compensado pelo fato do tempo de ajuste dos pesos ser bastante reduzido.