**Universidade Federal de Minas Gerais**

**Nome:** Guilherme Vinícius Amorim

**Matrícula:** 2017089081

**Data:** 10/2020

**Exercício 10**

O objetivo do exercício desta semana é observar que o MLP é capaz de aproximar qualquer função contínua. Para isso, foi realizada a regressão de um ciclo de uma senoide com backpropagation.

Inicialmente, os dados de treinamento e de testes foram criados, de acordo como uma senoide acrescida e um ruído. A Figura 1 mostra o código em R utilizado para criar esses dados.

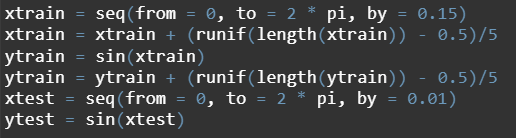


Figura 1: Criação dos dados de treinamento e de teste.

A partir dos dados criados, uma RNA foi desenvolvida com três neurônios na camada escondida e um neurônio na camada de saída com função der ativação linear. Para o ajuste dos pesos z da matriz escondida, o método backpropagation foi utilizado. O código utilizado para o treinamento da rede MLP com o ajuste de peso com backpropagation está em anexo ao final deste relatório.

Como há um fator de aleatoriedade da eficácia do treinamento MLP, treinou-se 5 vezes os dados de treinamento. Para cada treinamento, calculou-se o erro médio quadrático percentual (MSE). A Figura 2 mostra o algoritmo utilizado nos 5 treinamentos. A Figura 3 mostra valor do MSE para cada rodada de treinamento, a média desses 5 valores de erro, e o desvio padrão.

Os parâmetros utilizados no treinamento foram:

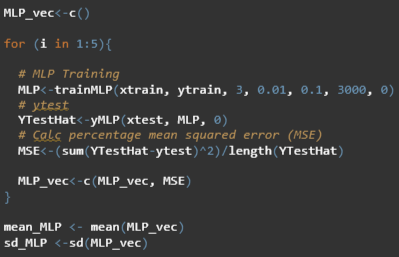


Figura 2: Código utilizado no treinamento e cálculo de MSE.

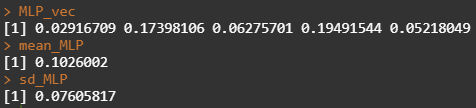


Figura 3: O valor do MSE para cada rodada de treinamento, a média desses 5 valores de erro, e o desvio padrão.

Analisando os resultados, observa-se valores erro altos. No quarto treinamento, o erro percentual foi 20%. Além disso, observa-se um desvio padrão também alto, mostrando que o fator de aleatoriedade tem grande importância no treinamento. A Figura 4 mostra o *plot* das saídas. Os pontos em preto são os dados utilizados no treinamento. As funções em vermelho e azul são as utilizadas para treinamento e a saída treinada pela rede ELM, respectivamente.

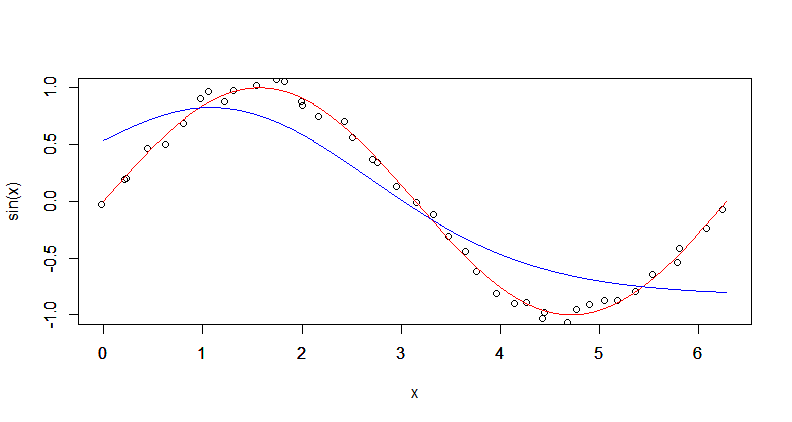


Figura 4: *Plot* das saídas senoidais de teste, treinamento e rede treinada.

## Conclusão

O alto erro no treinamento pode ter ocorrido em função do baixo número de neurônios na camada escondida. Além disso, a tolerância utilizada foi 0,1. Uma tolerância menor pode reduzir o valor de erro. Ademais, o número de dados de treinamento pode ter sido não suficiente. Mais dados de treinamento pode melhorar o resultado esperado.

O algoritmo da rede de treinamento MLP pode ser visto na próxima página.

