Relatório Atividade #05: RNA Aproximador de Funções

Integrantes:

Felipe Orlando Lanzara R.A: 24.122.055-7

João Vitor Governatore R.A: 24.122.027-6

Introdução

As Redes Neurais Artificiais (RNAs) são modelos computacionais inspirados no funcionamento do cérebro humano e são amplamente utilizadas para resolver problemas complexos de aprendizado de máquina. Entre as diversas aplicações das RNAs, destaca-se a aproximação de funções, uma técnica fundamental para a modelagem de fenômenos que não possuem uma formulação matemática exata.

A aproximação de funções por RNAs tem como objetivo encontrar uma relação entre um conjunto de entradas e saídas de forma precisa, reduzindo ao máximo o erro entre os valores previstos e os valores reais. Esse processo é realizado por meio do treinamento da rede, no qual os pesos das conexões entre os neurônios são ajustados iterativamente para minimizar a função de erro.

O desempenho da rede depende de diversos fatores, como a quantidade de neurônios nas camadas ocultas, o número de iterações de treinamento, a função de ativação utilizada e o algoritmo de otimização escolhido. Encontrar uma configuração adequada é essencial para garantir um equilíbrio entre a capacidade de generalização da rede e o tempo de processamento.

Neste relatório, será explorado o uso de uma RNA para a aproximação de funções, analisando o impacto do número de neurônios e de iterações no erro final da rede. A metodologia envolve a utilização de um Perceptron Multicamadas (MLP) treinado com o algoritmo Adam e funções de ativação não lineares. Os resultados obtidos serão avaliados com base na curva de erro e na precisão da função aproximada em relação à função original.

Exemplos Práticos

Exemplo 1: teste1.npy

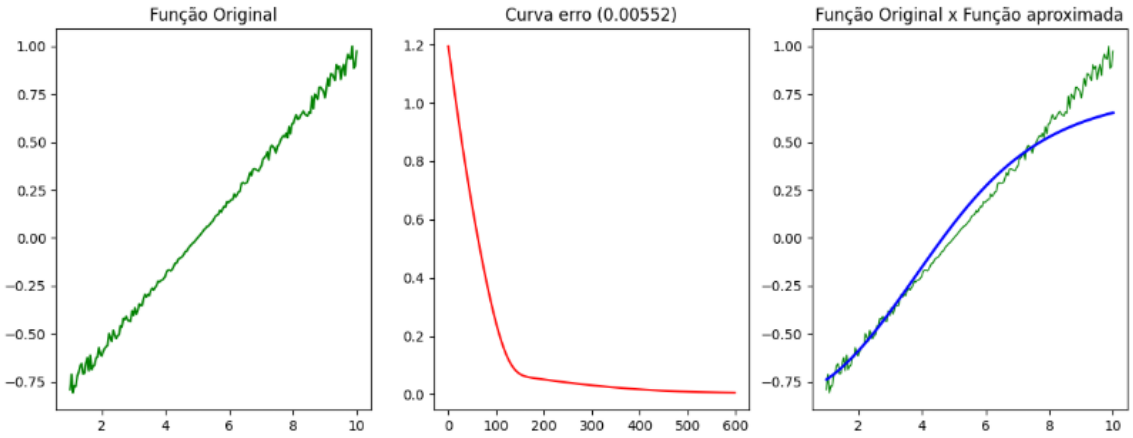
Simulação 01

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 600 iterações com a quantidade de neurônios sendo (5, 2).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0314337446
- Média de erros: 0.019417003521011604
- Menor valor de erro: 0.0011161409879226235

E dessa forma, os gráficos com o melhor resultado são:



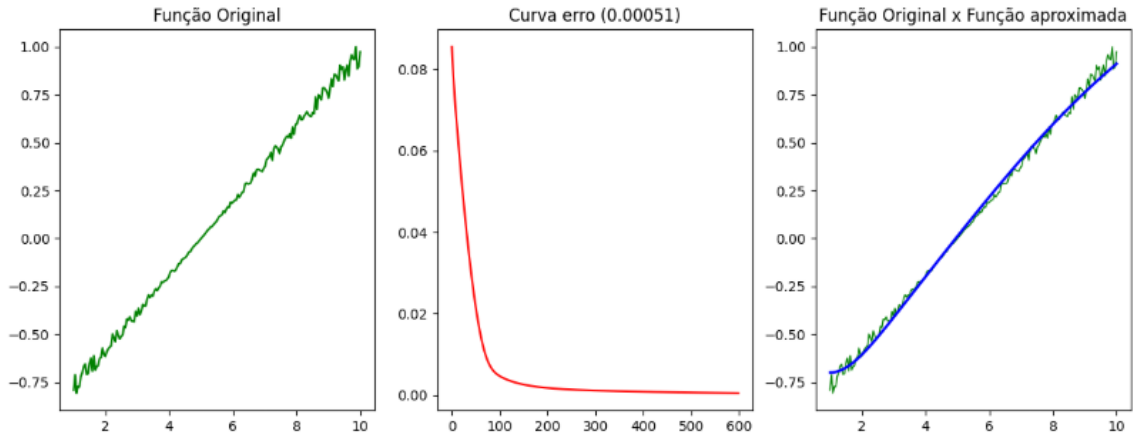
Simulação 02

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 600 iterações com a quantidade de neurônios sendo (10, 5).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0011607864
- Média de erros: 0.0016163655430493566
- Menor valor de erro: 0.0005099707075738473

E dessa forma, os gráficos com o melhor resultado são:



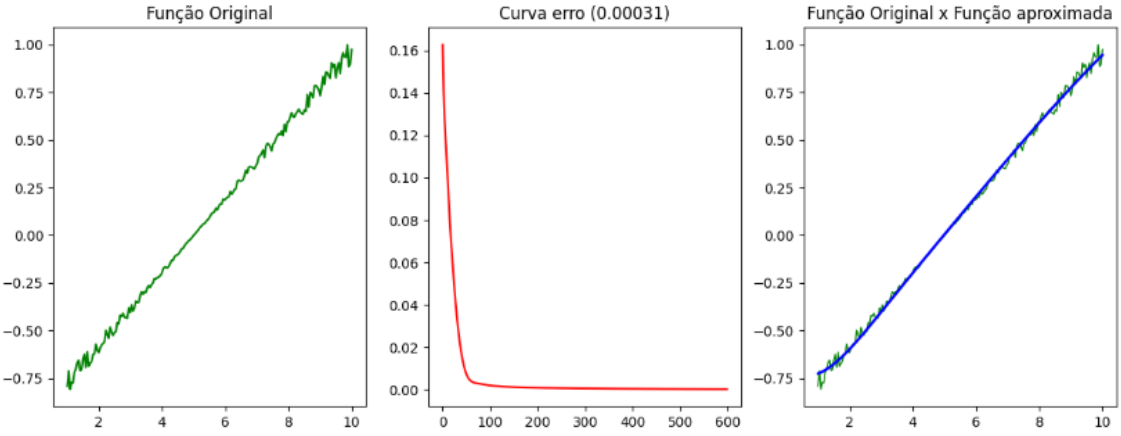
Simulação 03

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 600 iterações com a quantidade de neurônios sendo (20, 15).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0005859946
- Média de erros: 0.0008295523831694731
- Menor valor de erro: 0.00030661691417024943

E dessa forma, os gráficos com o melhor resultado são:



Exemplo 2: teste2.npy

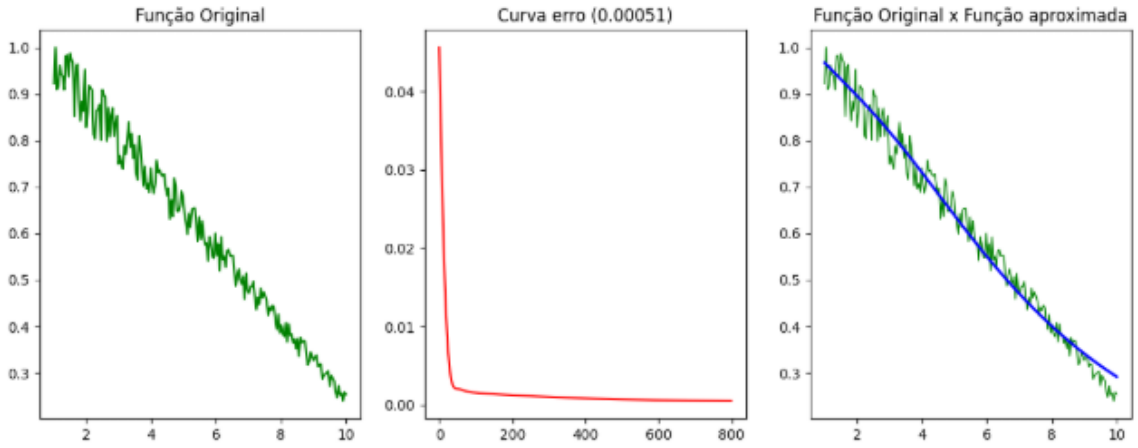
Simulação 01

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 800 iterações com a quantidade de neurônios sendo (5, 2).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0128362588
- Média de erros: 0.007484399711933596
- Menor valor de erro: 0.0005059861201686691

E dessa forma, os gráficos com o melhor resultado são:



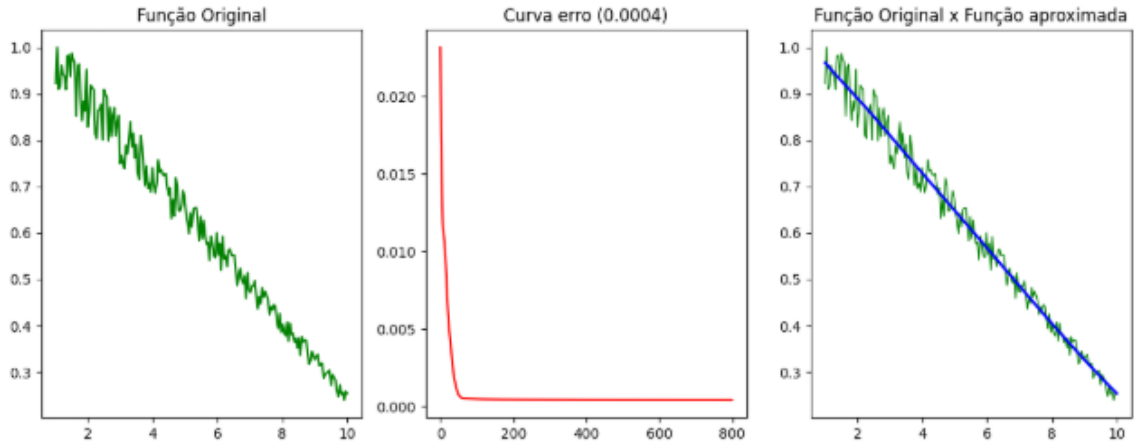
Simulação 02

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 800 iterações com a quantidade de neurônios sendo (10, 5).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0002676144
- Média de erros: 0.0007823975882419768
- Menor valor de erro: 0.00040353846067115785

E dessa forma, os gráficos com o melhor resultado são:



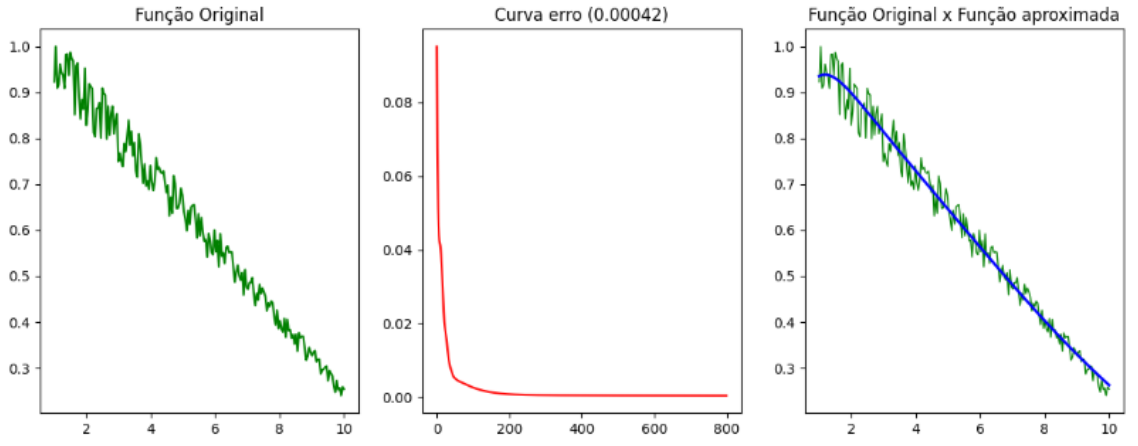
Simulação 03

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 800 iterações com a quantidade de neurônios sendo (20, 15).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0000935647
- Média de erros: 0.0005215974255308415
- Menor valor de erro: 0.0004157050713200531

E dessa forma, os gráficos com o melhor resultado são:



Exemplo 3: teste3.npy

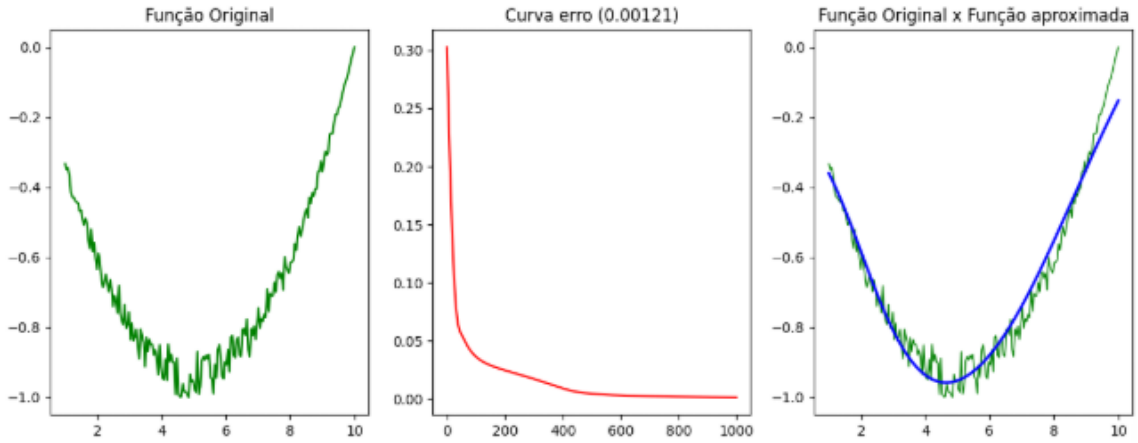
Simulação 01

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 1000 iterações com a quantidade de neurônios sendo (10, 5).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0044744106
- Média de erros: 0.005568547244948258
- Menor valor de erro: 0.0012101801894799367

E dessa forma, os gráficos com o melhor resultado são:



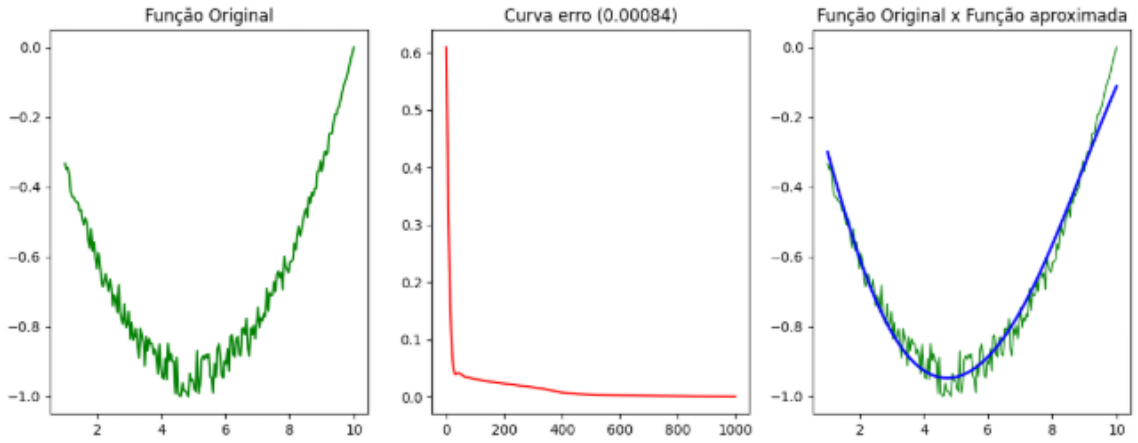
Simulação 02

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 1000 iterações com a quantidade de neurônios sendo (20, 10).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0008920945
- Média de erros: 0.0018089893160694675
- Menor valor de erro: 0.0008390475224833823

E dessa forma, os gráficos com o melhor resultado são:



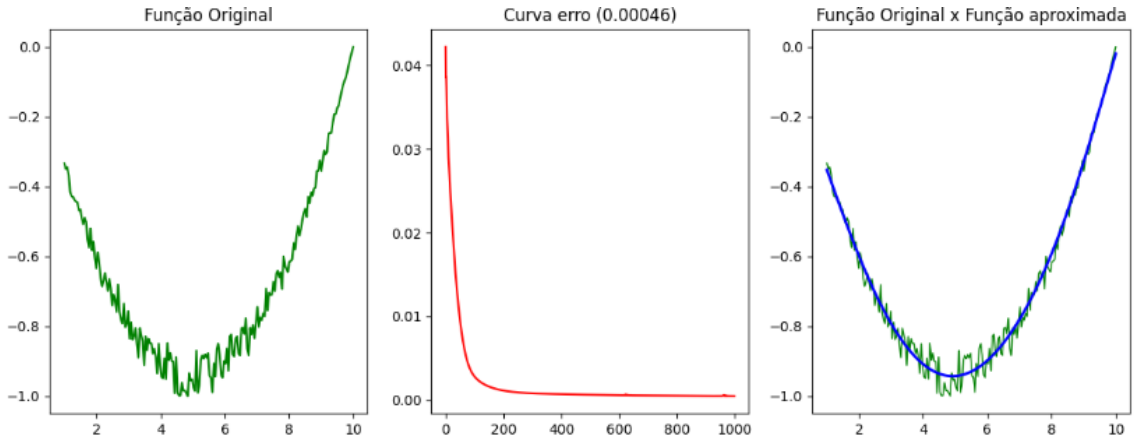
Simulação 03

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 1000 iterações com a quantidade de neurônios sendo (30, 20).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0001640134
- Média de erros: 0.0007135276529801391
- Menor valor de erro: 0.0004610446305975839

E dessa forma, os gráficos com o melhor resultado são:



Exemplo 4: teste4.npy

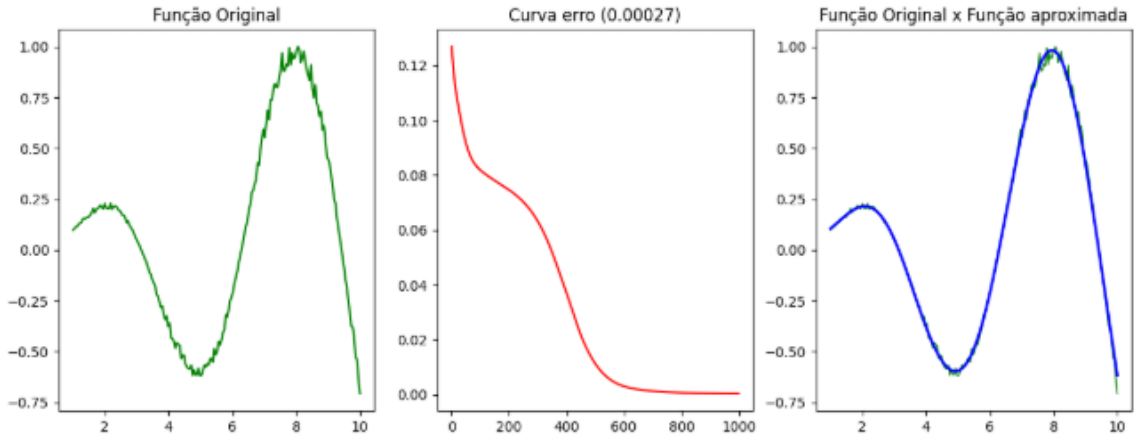
Simulação 01

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 1000 iterações com a quantidade de neurônios sendo (15, 10).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0283863076
- Média de erros: 0.053862482233546484
- Menor valor de erro: 0.0002730194734775859

E dessa forma, os gráficos com o melhor resultado são:



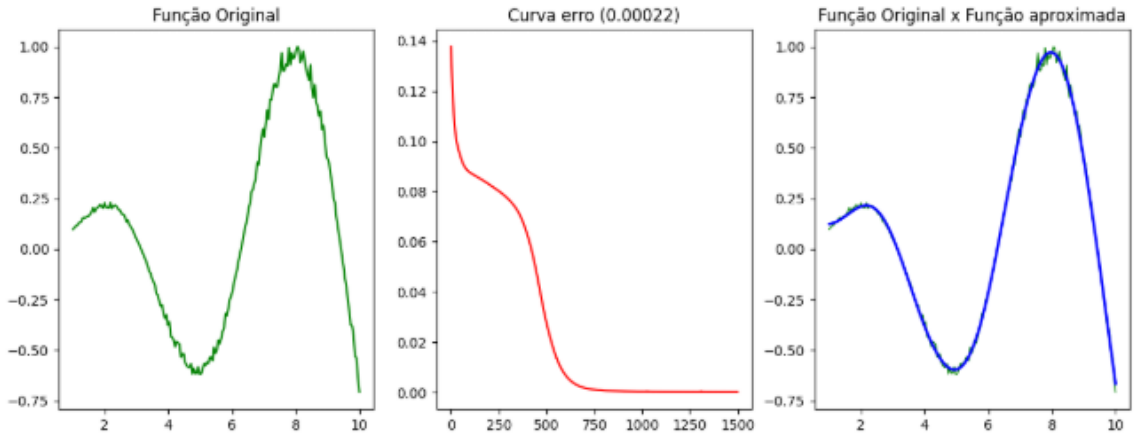
Simulação 02

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 1500 iterações com a quantidade de neurônios sendo (15, 15).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0260280146
- Média de erros: 0.022894886358122514
- Menor valor de erro: 0.0002180263937867538

E dessa forma, os gráficos com o melhor resultado são:



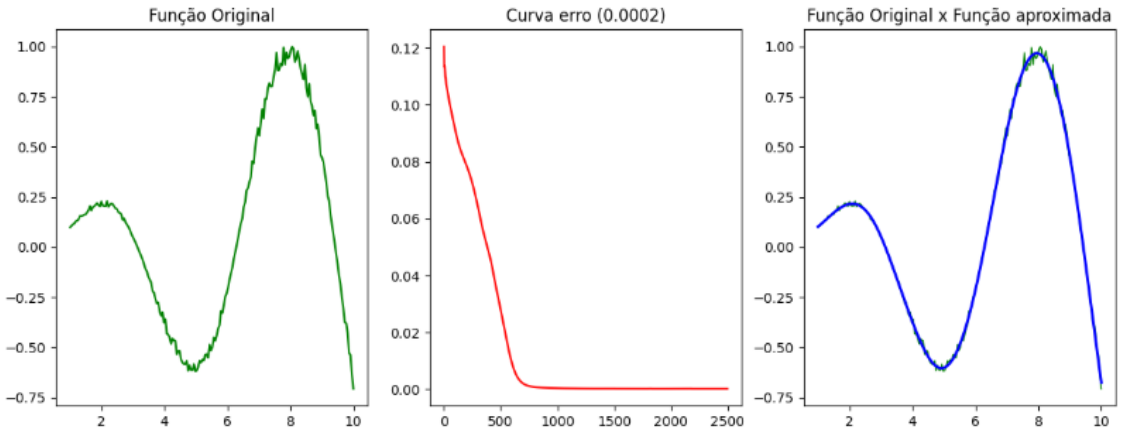
Simulação 03

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 2500 iterações com a quantidade de neurônios sendo (20, 15).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0000268633
- Média de erros: 0.0002348148388114958
- Menor valor de erro: 0.00020229617870964092

E dessa forma, os gráficos com o melhor resultado são:



Exemplo 5: teste5.npy

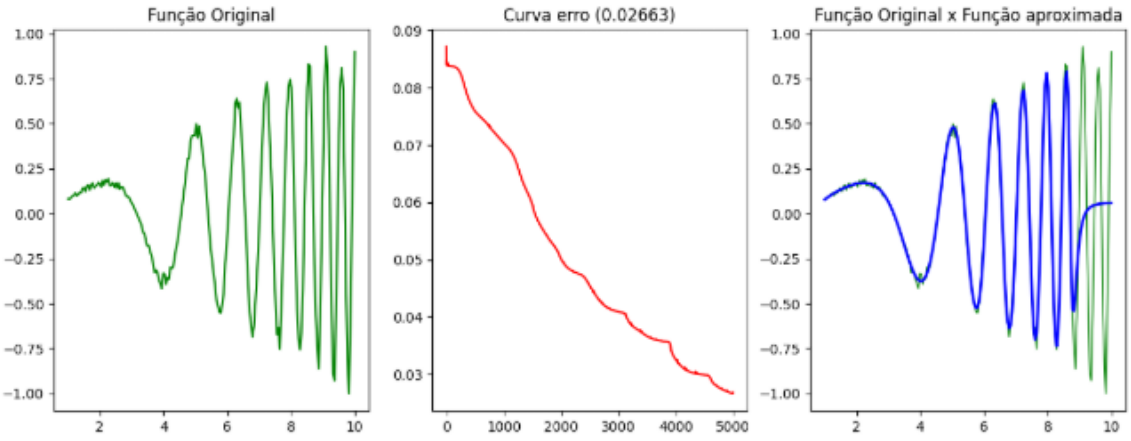
Simulação 01

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 5000 iterações com a quantidade de neurônios sendo (40, 20, 10).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0090634725
- Média de erros: 0.03906006787508779
- Menor valor de erro: 0.026629310793727445

E dessa forma, os gráficos com o melhor resultado são:



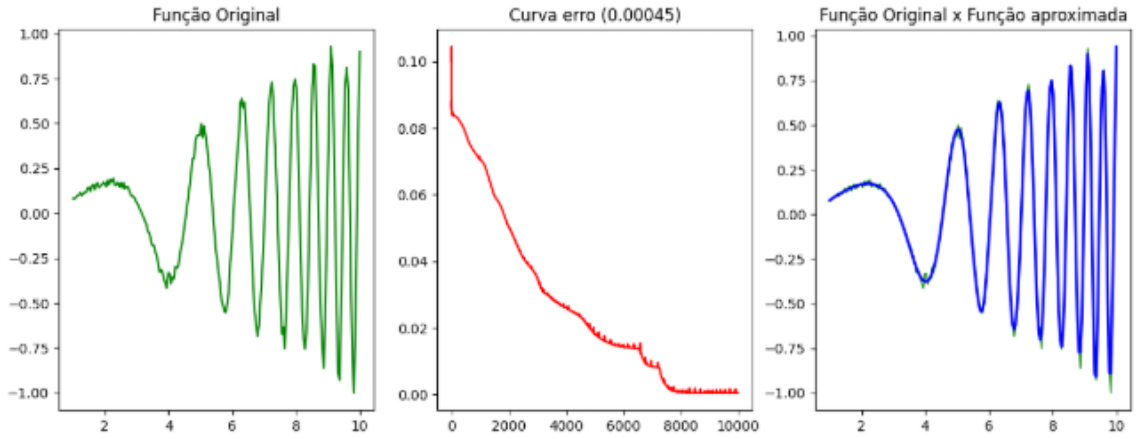
Simulação 02

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 10000 iterações com a quantidade de neurônios sendo (80, 40, 20).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0076105982
- Média de erros: 0.007978510829930067
- Menor valor de erro: 0.00045097797698656595

E dessa forma, os gráficos com o melhor resultado são:



Simulação 03

Neste primeiro exemplo, chegamos à conclusão de que seria necessário fazer 15000 iterações com a quantidade de neurônios sendo (120, 60, 40).

Após executarmos 10 vezes, chegamos esses valores:

- Desvio padrão: 0.0000385827
- Média de erros: 0.00044813482375297
- Menor valor de erro: 0.0003885823878116999

E dessa forma, os gráficos com o melhor resultado são: