



Relatório do Lab 8 de CT-213

Trabalho 8 – *Deep Learning* *Imitation Learning* com Keras

Aluno:

Bruno Costa Alves Freire

Turma:

T 22.4

Professor:

Marcos Ricardo Omena de Albuquerque Máximo

Data:

19/05/2019

**Instituto Tecnológico de Aeronáutica – ITA
Departamento de Computação**

1. Testando o *framework* Keras

No script `test_keras.py` é feito um teste do *framework* Keras para ajustar as funções `sum_gt_zero` e `xor`. Em comparação com o Lab anterior, isso é feito de maneira muito mais fácil com o Keras, que permite implementar uma rede neural de maneira muito mais simples, basta ir adicionando as camadas uma a uma, definindo as quantidades de neurônios, funções de ativação, mecanismos de regularização, a função de custo e o algoritmo de otimização usado no *back propagation*.

A primeira etapa do teste do Keras foi aplicar a rede neural criada ao aprendizado das funções `sum_gt_zero` e `xor` com e sem o mecanismo de regularização L_2 , com $\lambda = 0,002$. A arquitetura da rede contava com uma camada de entrada com 50 neurônios, e uma de saída com um neurônio. Em ambas as camadas a função de ativação era do tipo sigmoide. Para esse teste, foram gerados *datasets* com 200 pontos e adicionado um certo ruído para corromper esse *dataset*. A rede treinou por 5000 épocas em cada exemplo.

Os resultados obtidos para a função `sum_gt_zero` foram compilados nas figuras de 1 a 5, e os resultados para a função `xor` nas figuras de 6 a 10.

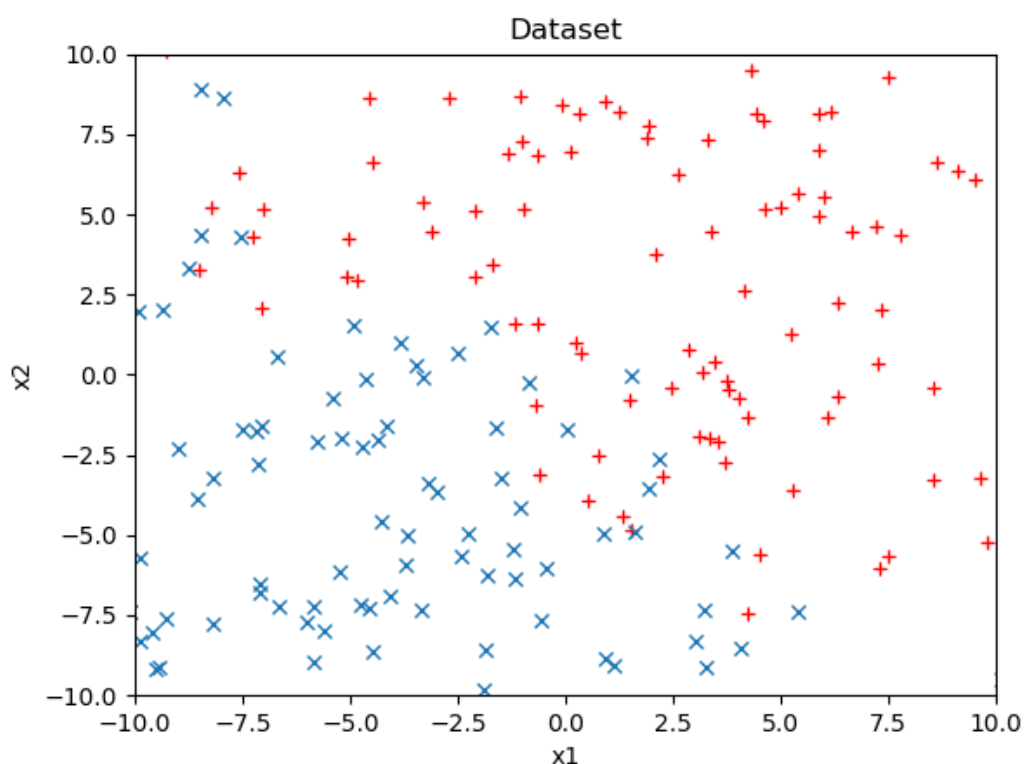


Figura 1: *Dataset* para treinamento da rede neural na função `sum_gt_zero` com ruído adicionado.

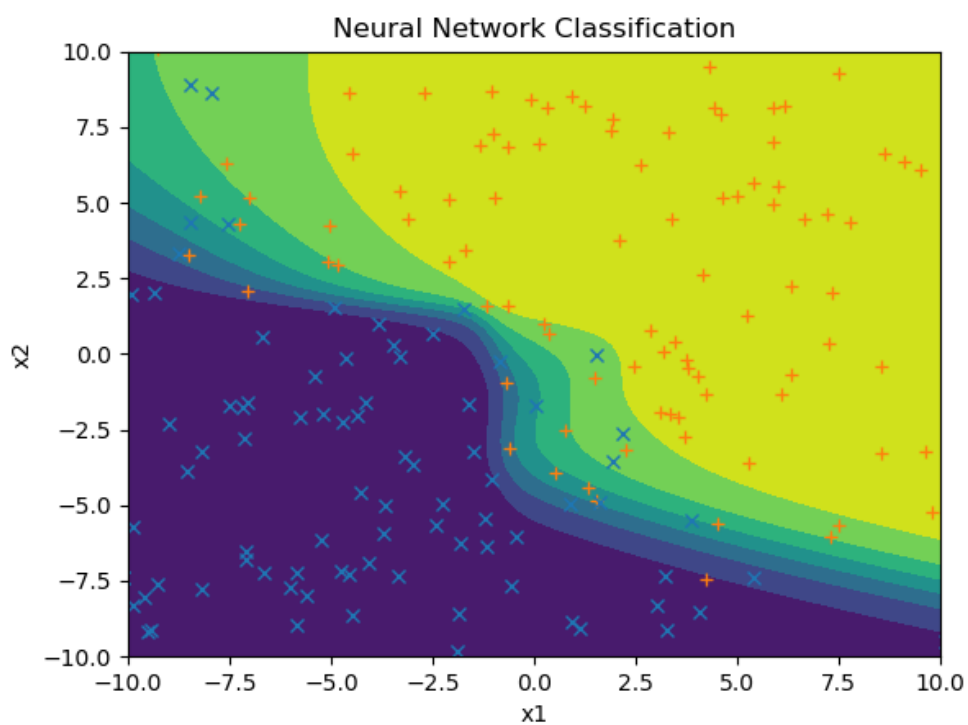


Figura 2: Resultado da classificação do *dataset* da função *sum_gt_zero* pela rede neural sem mecanismos de regularização.

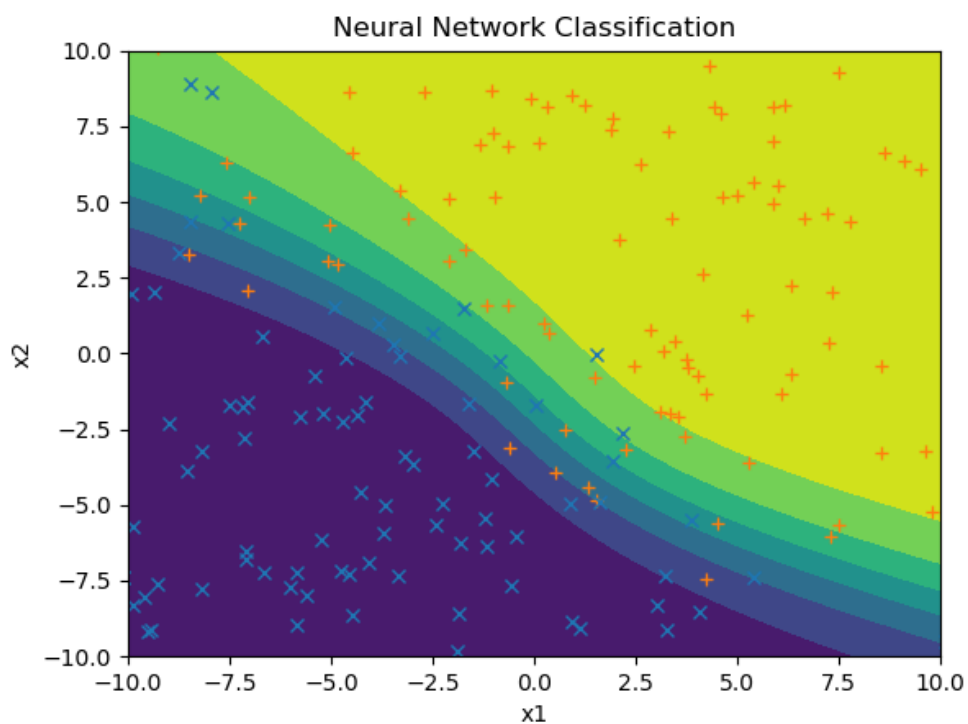


Figura 3: Resultado da classificação do *dataset* da função *sum_gt_zero* pela rede neural com regularização L_2 com $\lambda = 0,002$.

Comparando as figuras 2 e 3 com a classificação obtida pela rede neural do Lab anterior, implementada manualmente, vemos que o desempenho obtido por esta rede não foi tão bom. No entanto, há de se levar em consideração que anteriormente tínhamos uma rede com mais camadas (tudo bem, uma camada a mais apenas), e o dataset não tinha ruído.

Outra coisa que podemos notar é com relação ao efeito da regularização. Na figura 2 podemos ver que as curvas de nível da classificação sem regularização ficou com o formato distorcido, ocasionado pelo fato da rede ter aprendido uma peculiaridade desse dataset corrompido, em vez de abstrair o comportamento normal dos dados. Podemos dizer, de certo modo, que ela se ateve demais ao “macaco” do dataset, isto é, “aprendeu o ruído”, em vez de aprender o verdadeiro comportamento da função. Já na figura 3, podemos ver que com o mecanismo de regularização implementado, tivemos uma aproximação melhor do comportamento geral da função. Essas diferenças serão refletidas na evolução da função de custo ao longo do treinamento, como mostram as figuras 4 e 5.

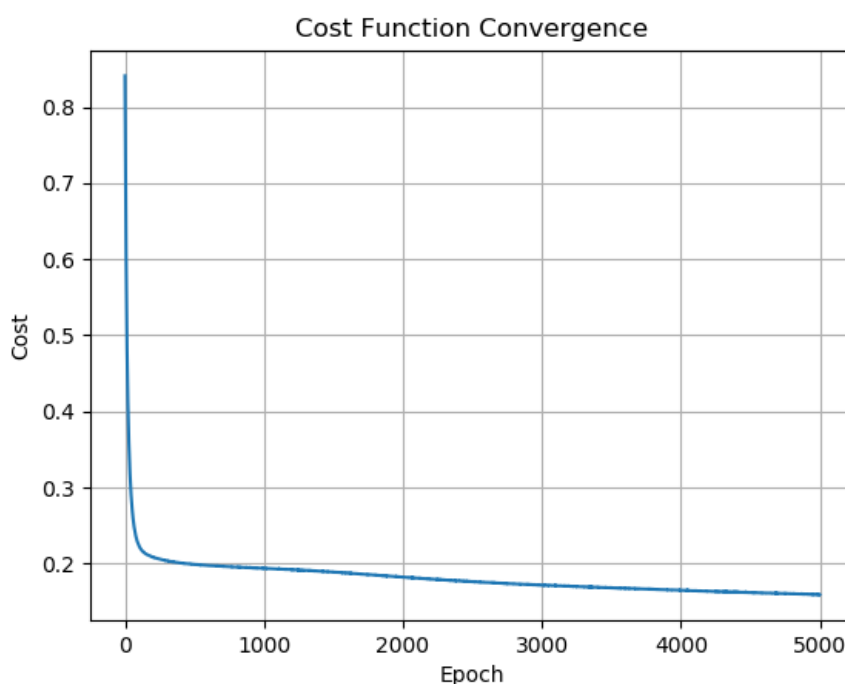


Figura 4: Evolução da função de custo da classificação do *dataset* da função `sum_gt_zero` pela rede neural sem regularização.

Pela figura 4, temos a impressão de que era apenas uma questão de tempo (executar o treinamento por mais épocas) até que a rede neural se encaixasse cada vez mais ao formato específico do *dataset*, ou seja, incorporasse totalmente o ruído ao seu modelo.

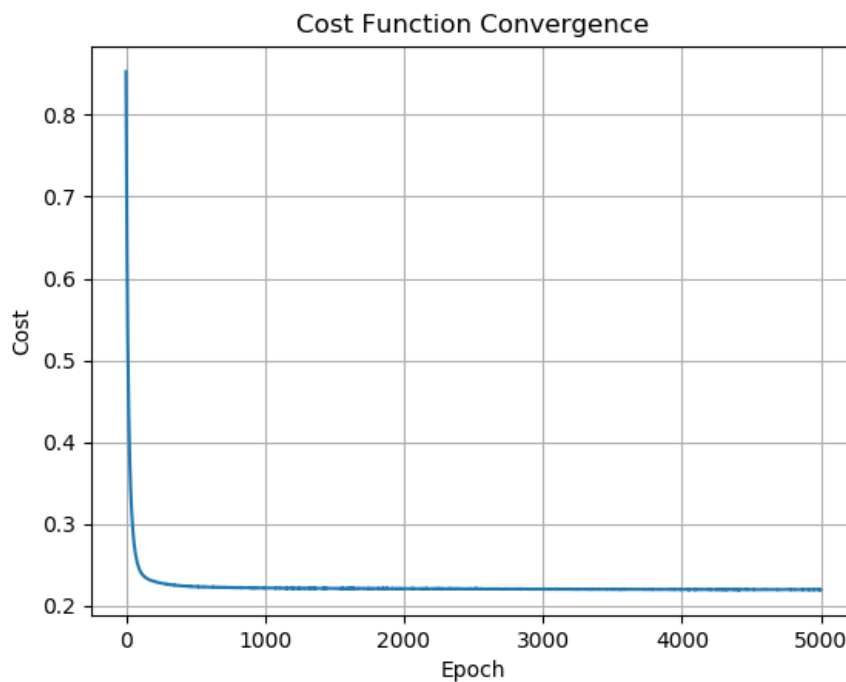


Figura 5: Evolução da função de custo da classificação do *dataset* da função `sum_gt_zero` pela rede neural com regularização L_2 de $\lambda = 0,002$.

Na figura 5 podemos notar uma aparente estagnação da função de custo num patamar estritamente acima do 0. Ou seja, a rede neural não converge para assimilar totalmente o *dataset*, mostrando uma melhor capacidade de ignorar o ruído presente no mesmo.

Da figura 6 em diante, analisamos o desempenho da rede neural no aprendizado da função `xor`.

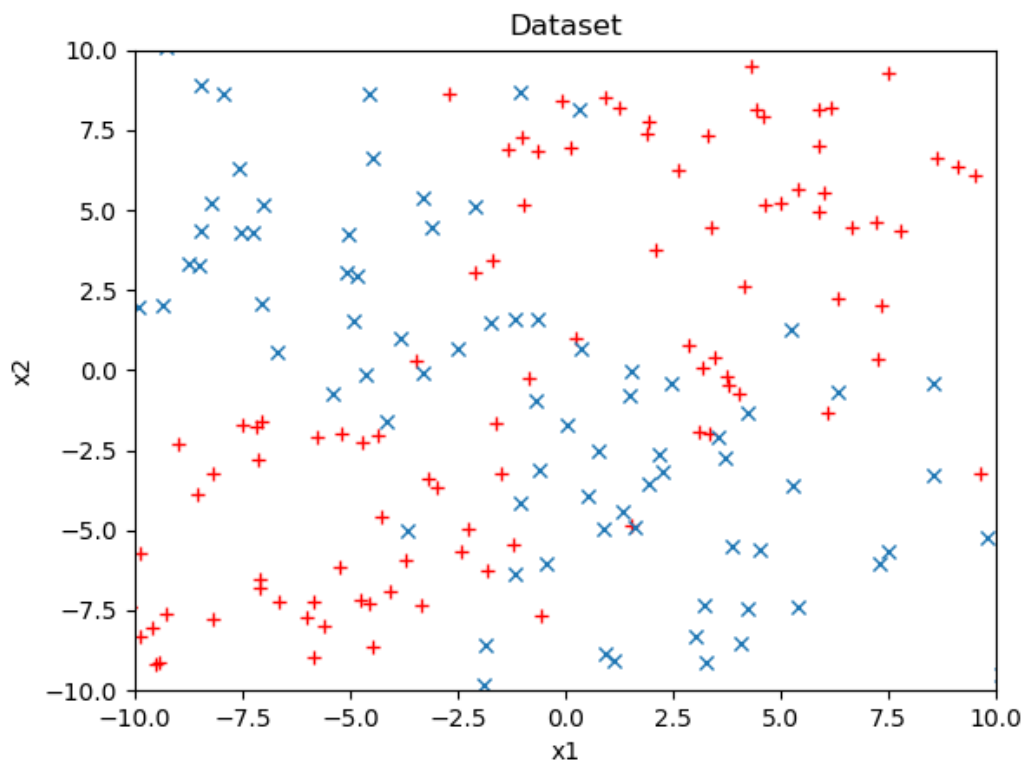


Figura 6: *Dataset* para treinamento da rede neural na função `xor` com ruído adicionado.

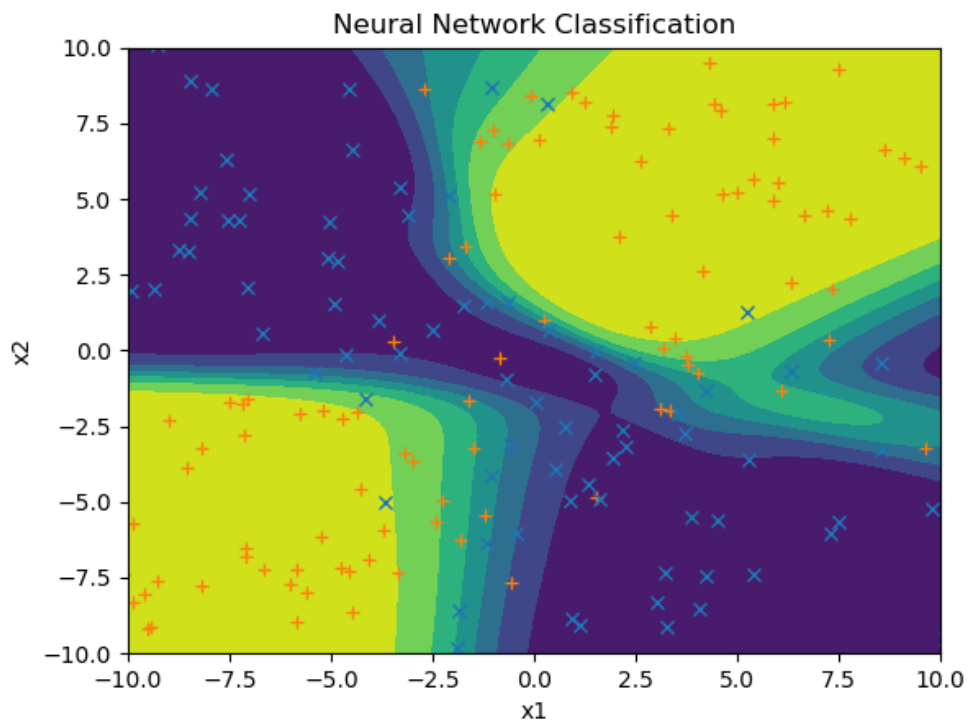


Figura 7: Resultado da classificação do *dataset* da função `sum_gt_zero` pela rede neural sem mecanismos de regularização.

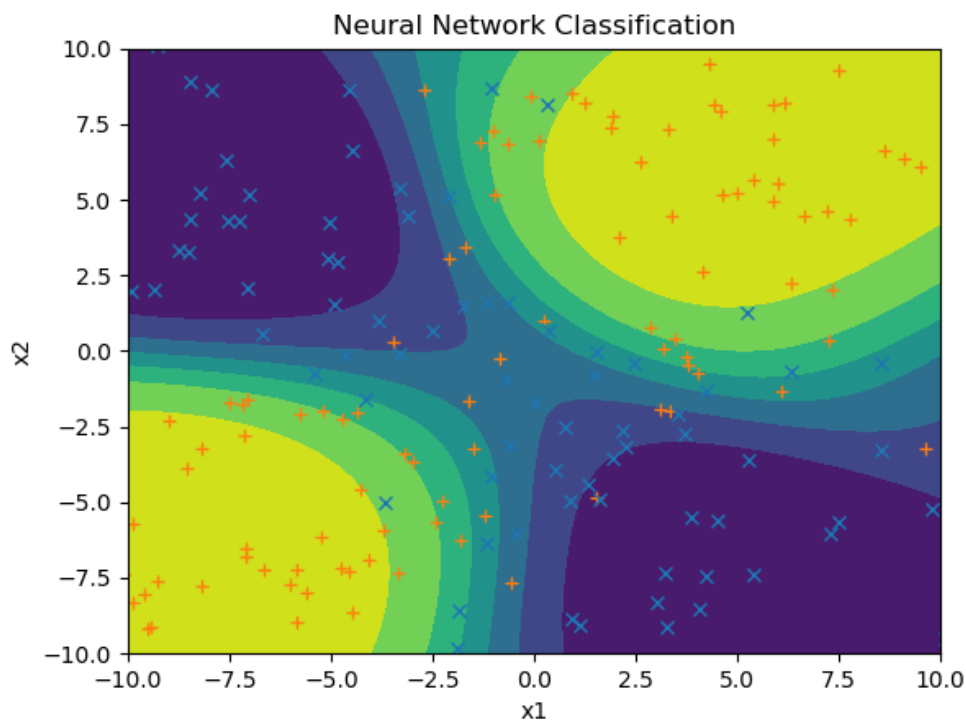


Figura 8: Resultado da classificação do *dataset* da função `sum_gt_zero` pela rede neural com regularização L_2 com $\lambda = 0,002$.

Comparando as figuras 7 e 8 com o resultado da rede neural do Lab anterior, temos até uma relativa melhora no formato das curvas de nível.

Contudo, novamente é evidente que a rede neural aqui implementada sem regularização (e também a do Lab anterior) aprenderam demais algumas particularidades dos respectivos datasets. Nesta ocasião, a rede neural claramente se ajustou ao ruído do dataset. Por outro lado, ao implementar a regularização, obtivemos uma geometria das curvas de nível mais simétrica, e próxima do comportamento real da função.

Vamos agora analisar a convergência da função de custo.

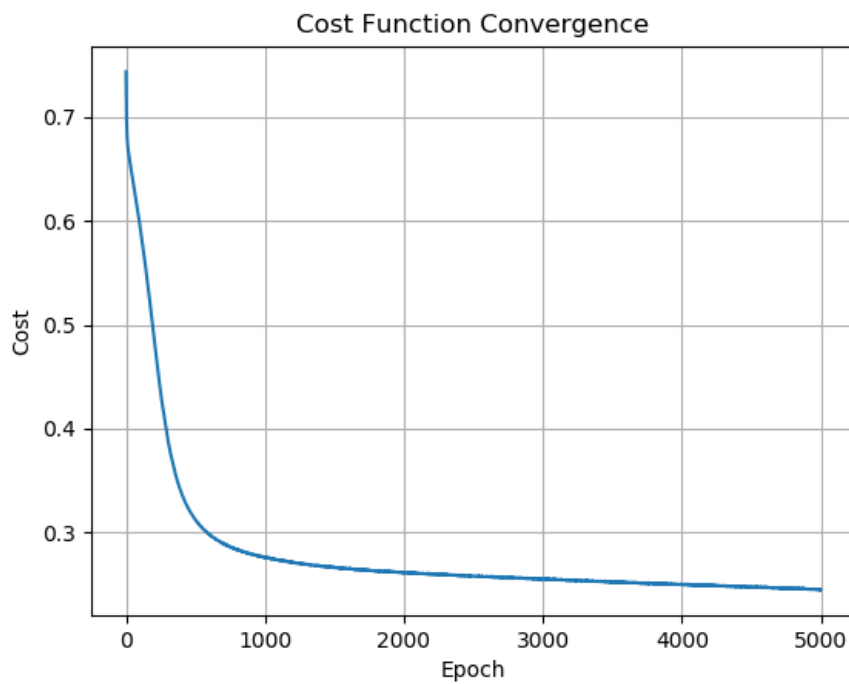


Figura 9: Evolução da função de custo da classificação do *dataset* da função `xor` pela rede neural sem regularização.

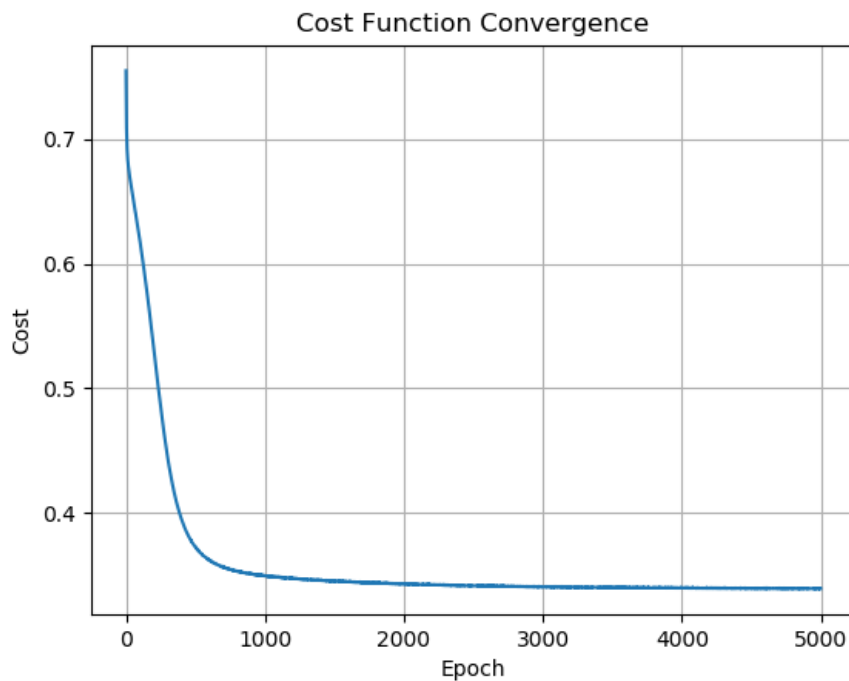


Figura 10: Evolução da função de custo da classificação do *dataset* da função `xor` pela rede neural com regularização L_2 de $\lambda = 0,002$.

Novamente podemos ver que a regularização possui um efeito de retardar a convergência da função de custo (o que é condizente com o objetivo de evitar a assimilação de ruído), tanto em valor final quanto em velocidade.

2. Usando o Keras para executar o *Imitation Learning*

Agora vamos utilizar o Keras para criar uma rede neural capaz de aprender o movimento de caminhada de um robô humanoide, por meio da técnica de *Imitation Learning*. Vamos utilizar uma rede neural com 3 camadas densas, com 75, 50 e 20 neurônios respectivamente. Nas duas primeiras camadas é utilizada a função de ativação Leaky ReLU com $\alpha = 0,01$, e na última a ativação é linear. A função de perda utilizada será quadrática, e em cada iteração do treinamento é utilizado todo o *dataset*. Por fim, não é utilizado nenhum mecanismo de regularização, uma vez que o objetivo é aprender o movimento das juntas do robô com bastante fidelidade aos dados colhidos.

Os resultados do modelo obtido pela rede neural para o movimento das juntas de arfagem do joelho, de rolamento e arfagem da cintura, e do rolamento e da arfagem do calcanhar constam nos gráficos das figuras de 11 a 15.

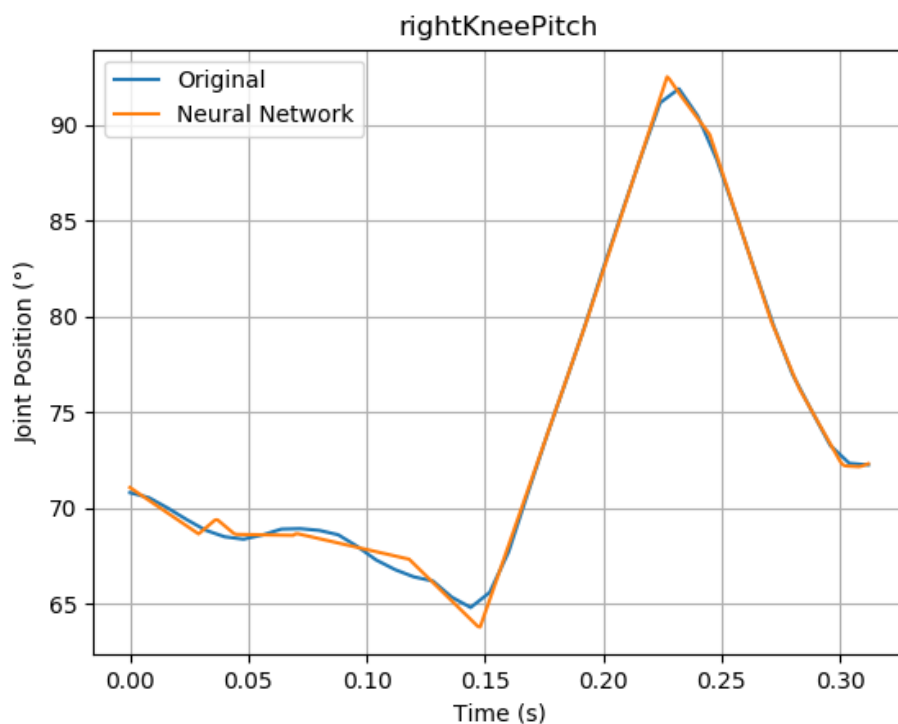


Figura 11: Comparação da posição angular da junta de arfagem do joelho direito original e aprendida pela rede neural.

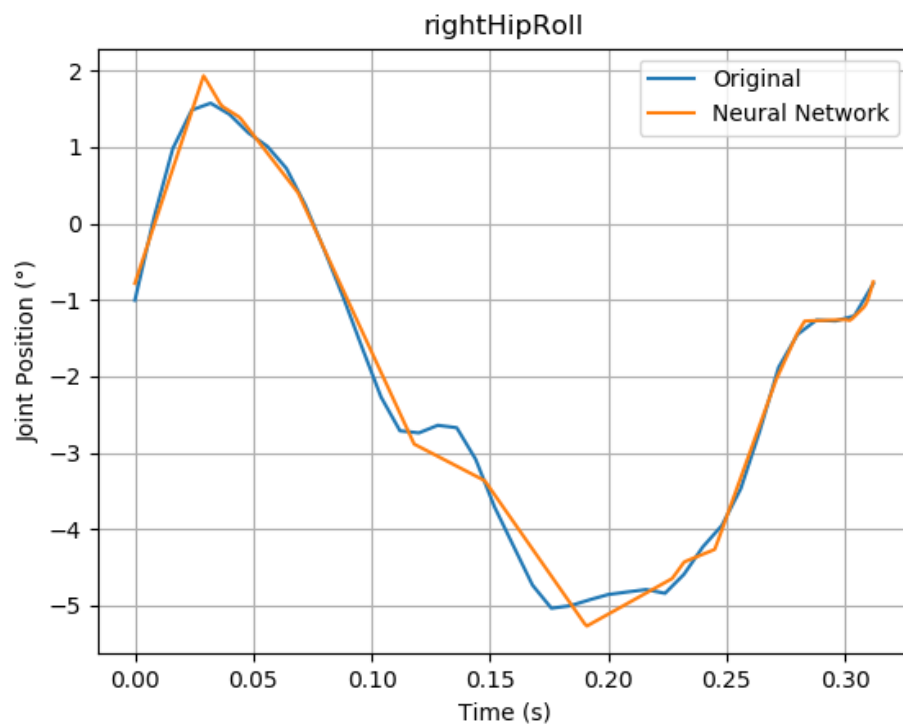


Figura 12: Comparação da posição angular da junta direita de rolamento da cintura original e aprendida pela rede neural.

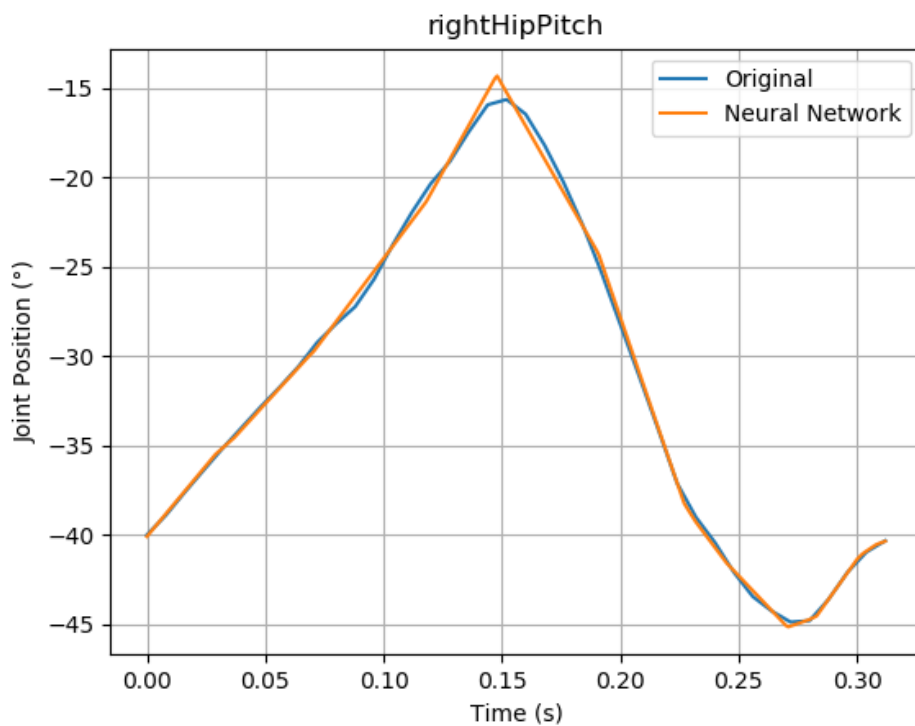


Figura 13: Comparação da posição angular da junta direita de arfagem da cintura original e aprendida pela rede neural.

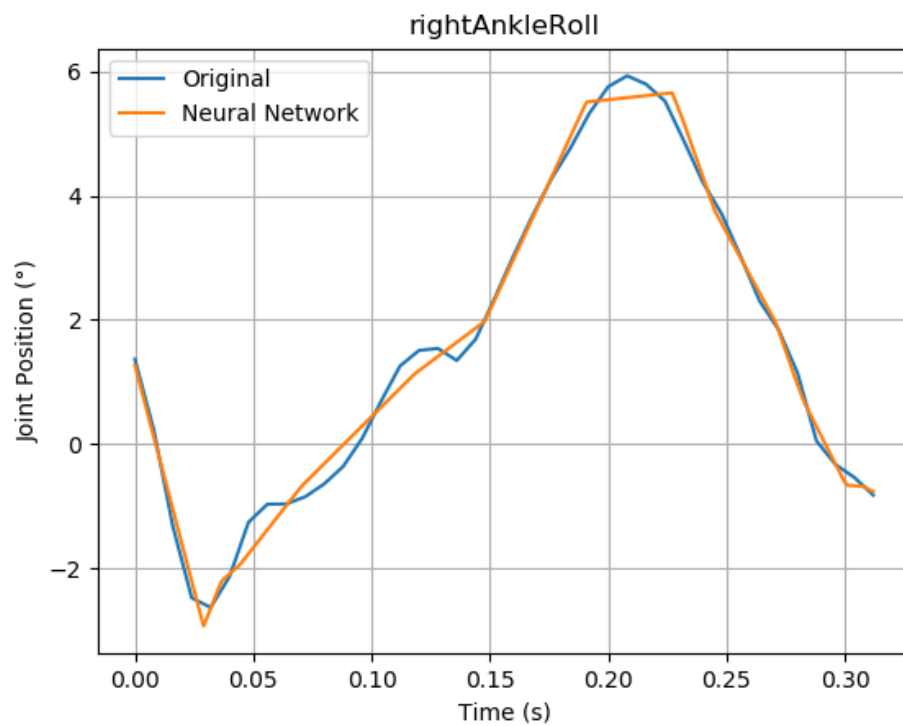


Figura 14: Comparação da posição angular da junta direita de rolamento do calcanhar original e aprendida pela rede neural.

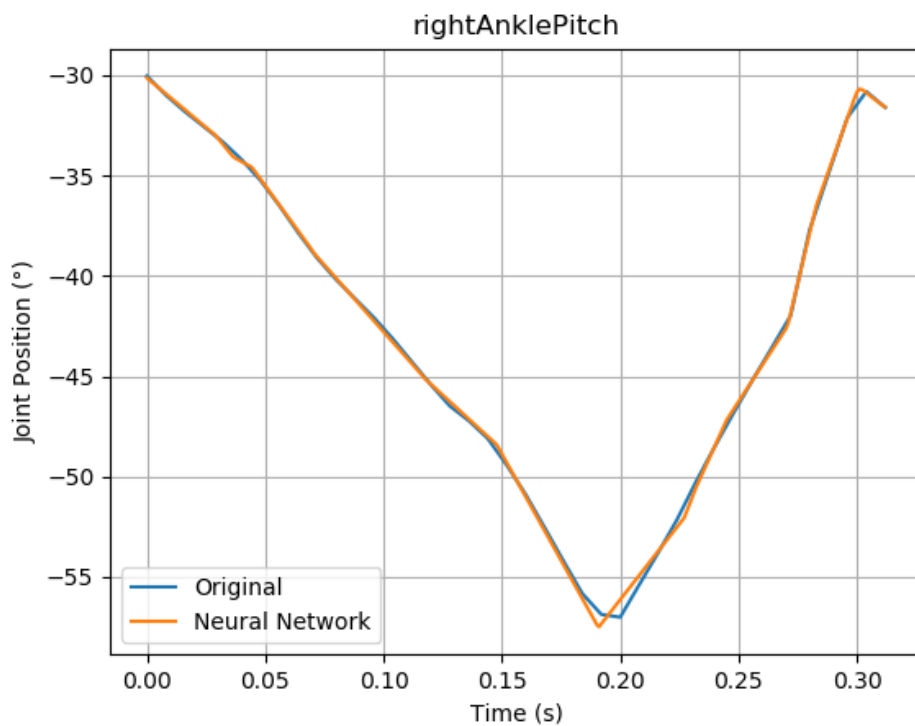


Figura 15: Comparação da posição angular da junta direita de arfagem do calcanhar original e aprendida pela rede neural.

Podemos avaliar o desempenho da rede como bom, a partir das figuras de 11 a 15, uma vez que as curvas obtida pelo aprendizado se aproximam bem das originais, com apenas alguns trechos levemente destoantes, principalmente no ajuste das juntas de rolamento do calcanhar e da cintura.