

Relatório do Laboratório 9: Redes Neurais Convolucionais

Isabelle Ferreira de Oliveira
CT-213 - Engenharia da Computação 2020
Instituto Tecnológico de Aeronáutica (ITA)
São José dos Campos, Brasil
isabelle.ferreira3000@gmail.com

Resumo—Esse relatório documenta a implementação, treino e teste da rede neural LeNet-5 usando o *dataset* MNIST, que consiste num conjunto grande de imagens anotadas de dígitos decimais escritos à mão. Assim, será reproduzido um trabalho clássico da literatura de Redes Neurais Convolucionais (CNNs), que foi realizado originalmente por Yann LeCun.

Index Terms—LeNet-5, MNIST, Redes Neurais Convolucionais, Keras, Tensorflow

I. IMPLEMENTAÇÃO

Para a implementação da rede neural conforme os parâmetros requisitados pelo roteiro do laboratório [1], era necessário utilizar do código de adição de camadas a uma rede, apresentado na seção Dicas do roteiro do laboratório [1].

Tendo em vista que em *keras.activations* não há função de ativação Leaky ReLU, utilizou-se a recomendação sugerida pelo roteiro [1] para usar Leaky ReLU no Keras, ou seja, foi adicionado uma camada do tipo LeakyRelu após ter definido uma camada (usando função de ativação linear).

Além disso, para configurar o modelo, setou-se o parâmetro *loss* da função *compile()* para *losses.mean_squared_error*, uma vez que foi utilizado erro quadrático.

Por fim, para treinar, o tamanho do *batch* foi o tamanho total de entradas, para que seja usado todo o *dataset* em cada iteração do treinamento.

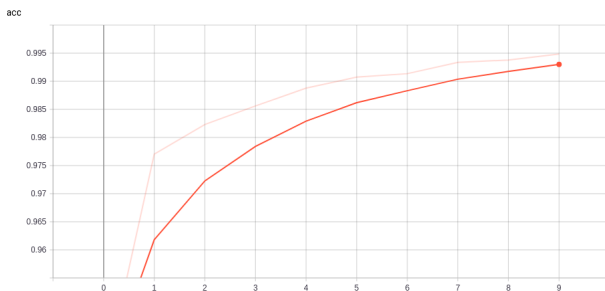


Figura 1. Arquitetura da rede neural usada para o imitation learning. Essa imagem foi apresentada no roteiro [1]

II. RESULTADOS E CONCLUSÕES

A. Estudo de implementação de Rede Neural com Keras

O código do arquivo *test_keras.py* foi estudado para se entender a utilização do *framework* Keras na implementação

de redes neurais. O que foi aprendido resultou no texto escrito na Introdução.

B. Análise do efeito de Regularização

Esse arquivo *test_keras.py* continha a implementação do aprendizado das funções "soma maior que zero" e "xor" para diferentes valores de regularização.

Após a execução desse arquivo, obteve-se os resultados apresentados nas Figuras de ?? a ??, sendo as Figuras ?? e ?? os *dataset* utilizados para as funções "soma maior que zero" e "xor", respectivamente.

Analisando os resultados, é possível notar que em todas as situações (com e sem regularização, e para as duas funções) a rede obteve uma classificação satisfatória. A classificação com regularização, entretanto, apresentou-se muito mais acertiva, conforme se pode observar pela comparação das Figuras ?? e ?? para "soma maior que zero" e ?? e ?? para "xor".

A questão da convergência da função de custo também pode ser comparada. Para os casos com regularização, a convergência se deu bem antes em número de épocas. Isso pode ser observado nas Figuras ?? e ??, para "soma maior que zero" e nas Figuras ?? e ?? para "xor".

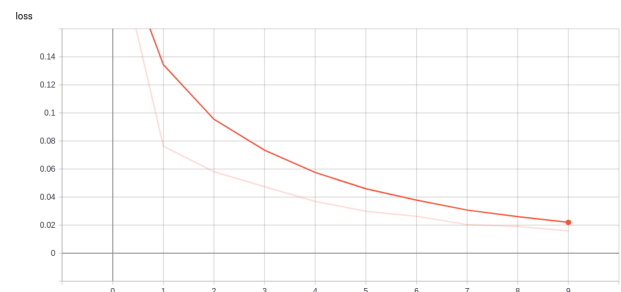


Figura 2. *Dataset* utilizado para o aprendizado da função *soma > 0*

C. Imitation Learning

Após a implementação da rede neural com Keras conforme o explicado na seção Implementação, os resultados obtidos estão apresentados nas Figuras de ?? a ?? . A comparação entre os gráficos de azul (curva original) e laranja (função aprendida por rede neural) dessas figuras demonstra que a implementação

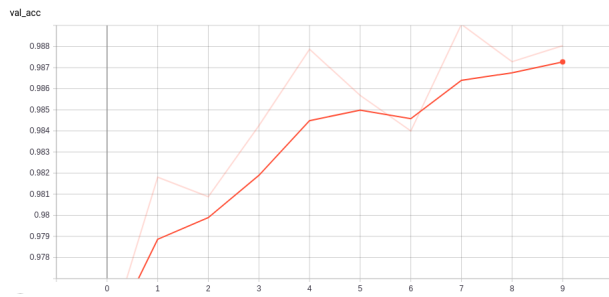


Figura 3. Convergência da função de custo para a função *soma* > 0, sem regularização.

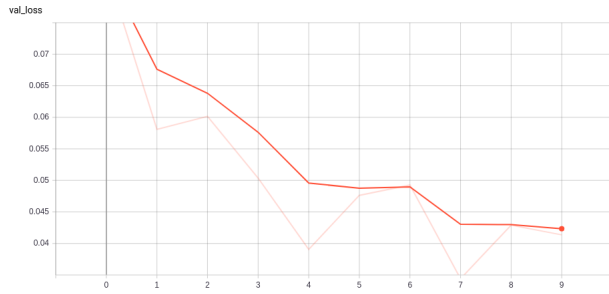


Figura 4. Resultado da classificação por rede neural para a função *soma* > 0, sem regularização.

aconteceu de maneira satisfatória, uma vez que as funções ficaram bastante semelhantes.

Tendo em vista o que foi apresentado, pode-se notar, por fim, que algoritmos de *Deep leaning* e o *framework* Keras realmente se demonstraram eficazes em realizar aprendizado por imitação.

REFERÊNCIAS

- [1] M. Maximo, “Roteiro: Laboratório 8 - Imitation Learning com Keras”. Instituto Tecnológico de Aeronáutica, Departamento de Computação. CT-213, 2019.

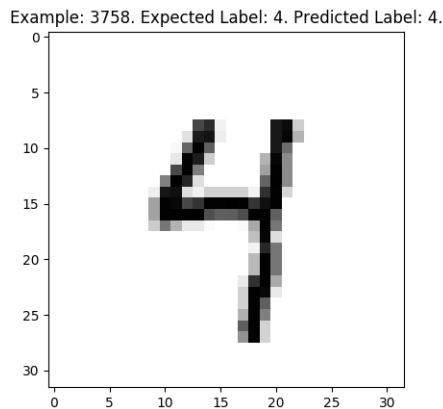


Figura 5. Convergência da função de custo para a função *soma* > 0, com regularização.

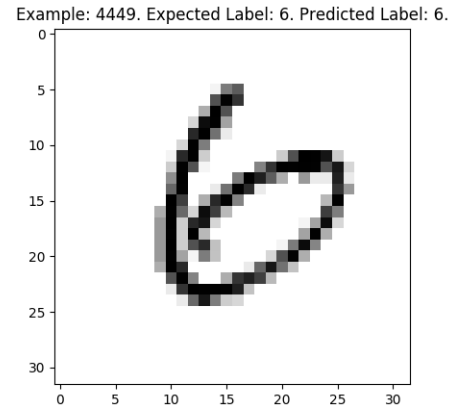


Figura 6. Resultado da classificação por rede neural para a função *soma* > 0, com regularização.

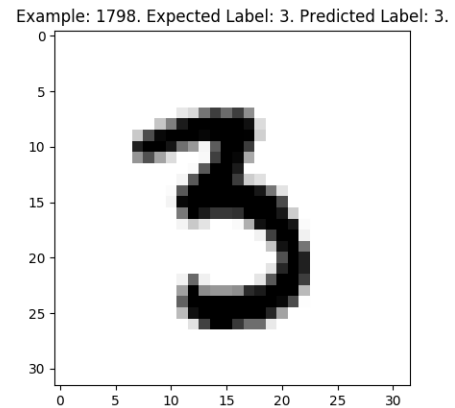


Figura 7. Dataset utilizado para o aprendizado da função *xor*.

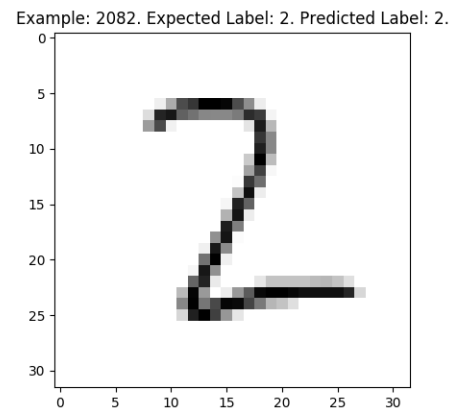


Figura 8. Convergência da função de custo para a função *xor*, sem regularização.

Example: 2439. Expected Label: 2. Predicted Label: 2.

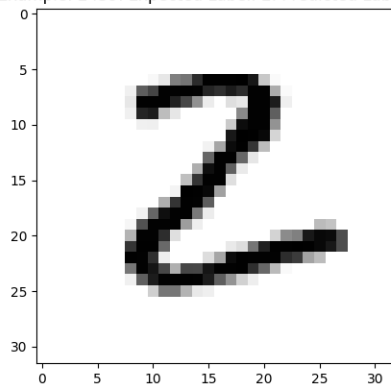


Figura 9. Resultado da classificação por rede neural para a função *xor*, sem regularização.

Example: 18. Expected Label: 3. Predicted Label: 8.

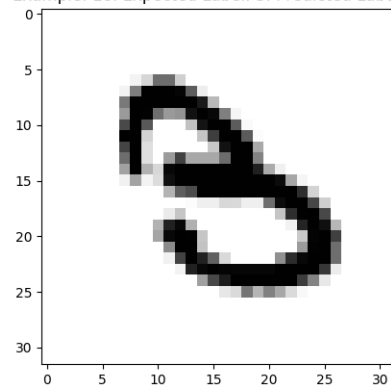


Figura 12. Resultado da classificação por rede neural para a função *soma > 0*.

Example: 8. Expected Label: 5. Predicted Label: 6.

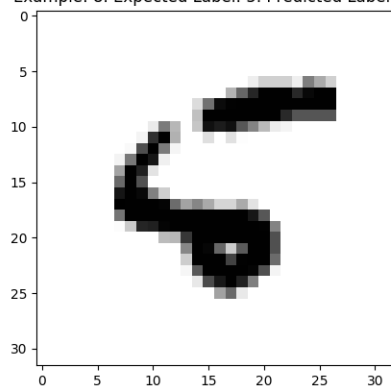


Figura 10. Convergência da função de custo para a função *xor*, com regularização.

Example: 247. Expected Label: 4. Predicted Label: 2.

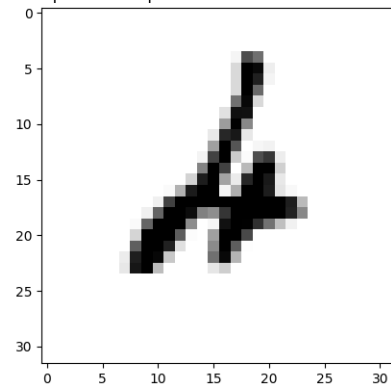


Figura 13. Resultado do *Imitation learning* para Right Ankle Roll.

Example: 92. Expected Label: 9. Predicted Label: 4.

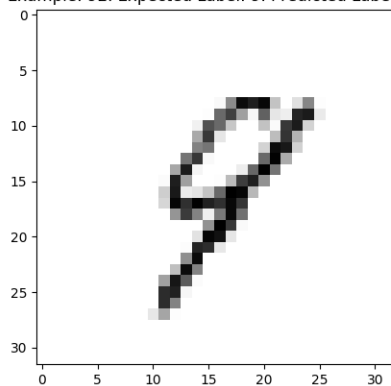


Figura 11. Resultado da classificação por rede neural para a função *xor*, com regularização.

Example: 321. Expected Label: 2. Predicted Label: 7.

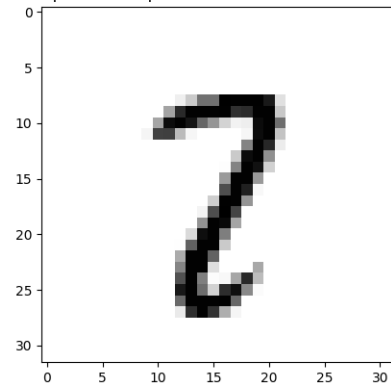


Figura 14. Resultado do *Imitation learning* para Right Hip Pitch.