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 imangens 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, fortemente inspirado das linhas de código apresentadas na seção Dicas do roteiro do laboratório [1]. Um detalhe importante a ser sitado foi que na criação das camadas de average pooling no Keras, os parâmetros *px* e *py* (no parâmetro *pool_size*) são os tamanhos do Kernel apresentados nos requisitos para cada uma dessas camadas.

A rede então foi treinada utilizando o script *train_lenet5.py* e os resultados desse treinamento foi exibido no *Tensorboard*, mostrado a partir da execução do script *run_tensorboard.py*. Ambos scripts já foram fornecidos previamente no código básico disponibilizado para esse laboratório.

Por fim, a rede foi avaliada no *test set* usando o script *evaluate_lenet5.py*, gerando exemplos gráficos de predições corretas e outras erradas.

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 implentação 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.

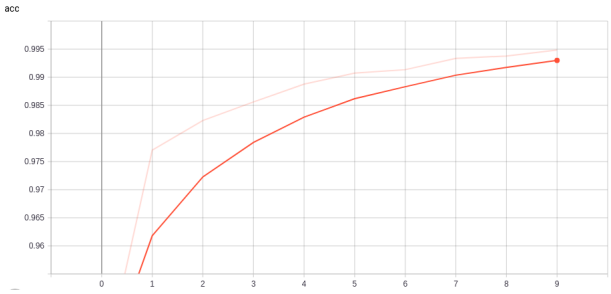

Figura 1. Arquitetura da rede neural usada para o imitation learning. Essa imagem foi apresentada no roteiro [1]

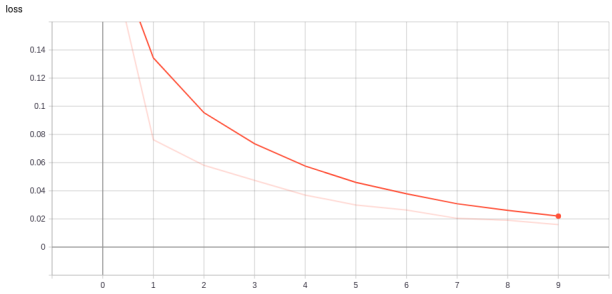

Figura 2. *Dataset* utilizado para o aprendizado da função $soma > 0$

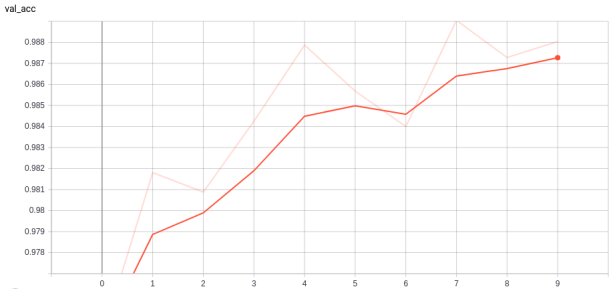

Figura 3. Convergência da função de custo para a função $soma > 0$, sem regularização.

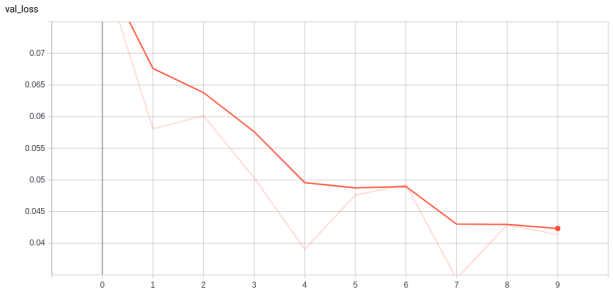

Figura 4. Resultado da classificação por rede neural para a função $soma > 0$, sem regularização.

Analisando os resultados, é possível notar que em todas as situações (com e sem regularização, e para as duas funções) a rede obtive uma classificação satisfatória. A classificação com regularização, entretando, apresentou-se muito mais acertiva, conforme se pode observar pelas 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".

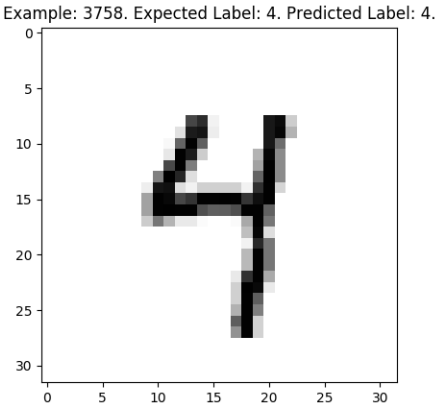

Figura 5. Convergência da função de custo para a função $soma > 0$, com regularização.

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 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.

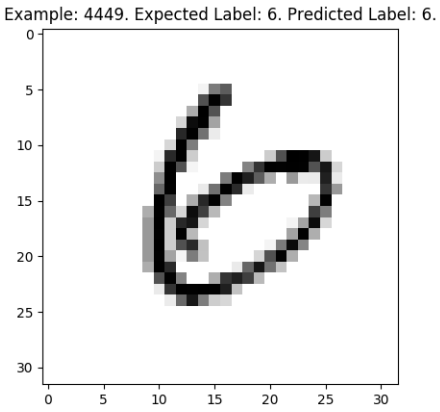

Figura 6. Resultado da classificação por rede neural para a função $soma > 0$, com regularização.

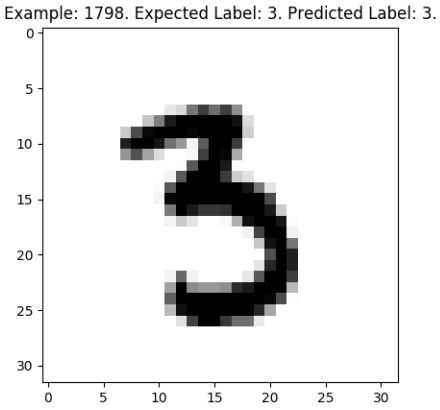

Figura 7. *Dataset* utilizado para o aprendizado da função xor.

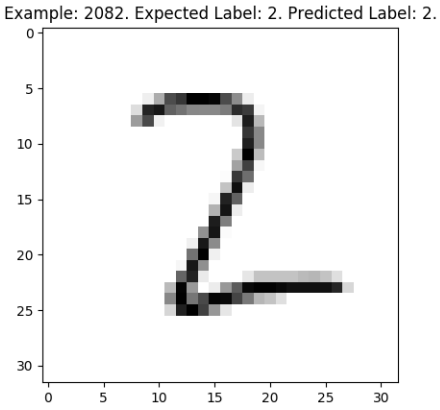

Figura 8. Convergência da função de custo para a função xor, sem regularização.

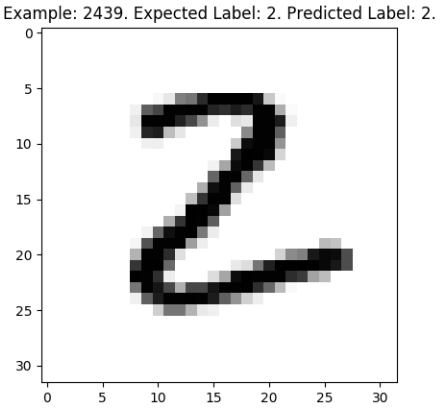

Figura 9. Resultado da classificação por rede neural para a função xor, sem regularização.

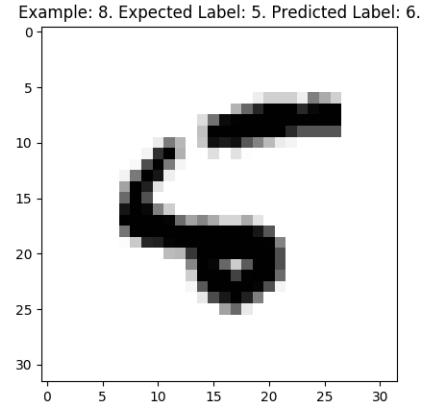

Figura 10. Convergência da função de custo para a função xor, com regularização.

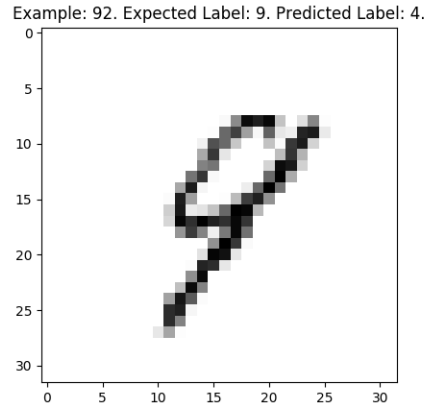

Figura 11. Resultado da classificação por rede neural para a função xor, com regularização.

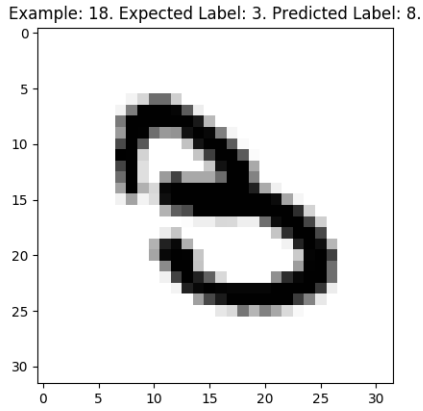

Figura 12. Resultado da classificação por rede neural para a função *soma* > *0*.

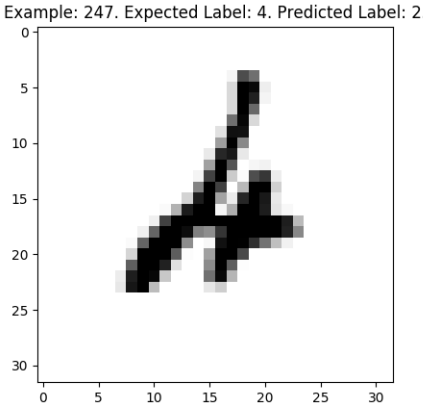

Figura 13. Resultado do *Imitation learning* para Right Ankle Roll.

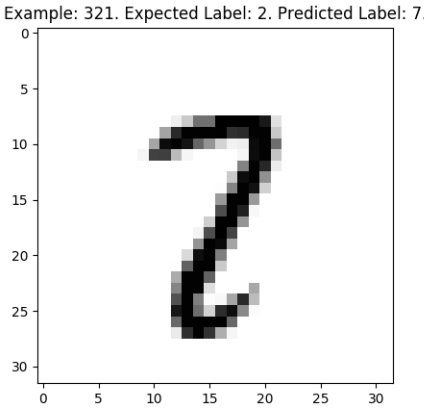

Figura 14. Resultado do *Imitation learning* para Right Hip Pitch.

REFERÊNCIAS

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