Profa. Dra. Roseli Aparecida Francelin Romero

Projeto 2: CNN

SCC0270 - Introdução à Redes Neurais.

Igor Martinelli 9006336
Zoltán Hirata Jetsmen 9293272

25 de Outubro de 2018

Sumário

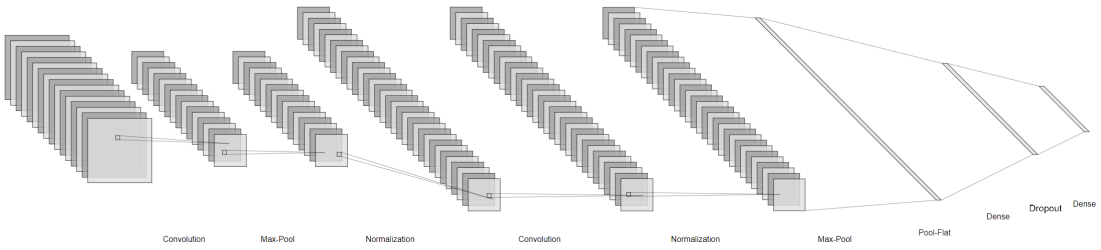
1 Introdução

Este relatório tem por objetivo descrever as etapas desenvolvidas para realizar a implementação da rede CNN com o objetivo de realizar a classificação de 10 instâncias, sendo uma de cada classe, da base de dados CIFAR-10[1].

2 Arquitetura da rede

Para a construção da arquitetura utilizada para a resolução deste problema, baseou-se no tutorial[2] disponibilizado e na rede implementada[3] pela biblioteca tensorflow. Dessa maneira, uma representação gráfica da rede implementada pode ser observada pela figura 1

Figura 1: Representação gráfica da rede utilizada.



A rede apresentada conta com 64 unidades convolucionais em sua primeira camada, seguida de uma camada de *max-pooling*, uma de normalização, outra camada convolucional com 128 unidades seguida de uma camada de normalização e outra de *max-pooling*. Por fim, tem-se uma camada *flatten* e 1 camada densa, com 512 neurônios seguida por uma camada de *dropout* com taxa de *drop* igual a 30% seguida por uma outra camada densa, com 1024 neurônios e, por fim, a camada de saída, com 10 neurônios, cada um especializado em uma classe do conjunto de dados. Nas camadas convolucionais e densas, utilizou-se, como função de ativação, a ReLu (*Rectified linear unit*).

3 Implementação e treinamento

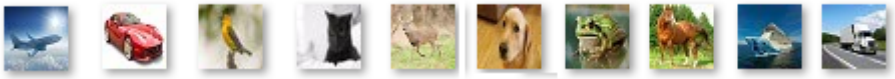
Para a implementação da rede, utilizou-se a biblioteca *tensorflow*. Utilizou-se o conjunto de dados *in-place*, para isso, foi utilizado o módulo *keras.datasets.cifar10*. Para o treino da rede, foram utilizadas 20 *epochs*, taxa de aprendizado igual a 0.01 e função de perda *sparse softmax cross entropy*[4]

[1]https://www.cs.toronto.edu/ kriz/cifar.html
[2]https://www.tensorflow.org/tutorials/estimators/cnn
[3]https://www.tensorflow.org/tutorials/images/deep_cnn
[4]https://www.tensorflow.org/api_docs/python/tf/losses/sparse_softmax_cross_entropy

4 Teste

Após a etapa de treinamento da rede, validou-se a mesma utilizando os exemplos de teste fornecido pelo conjunto de dados e fora obtido uma acurácia de 73.3%, assim, salvou-se o modelo utilizando o método *Estimator*. A etapa seguinte consistiu na obtenção do conjunto de dados utilizado para a validação do modelo, conjunto este composto por um exemplo de cada instância da base de dados CIFAR-10. As imagens coletadas podem ser observadas na figura 2.

Figura 2: Imagens coletadas para o teste.



Após a etapa de teste com as imagens obtidas, computou-se a acurácia para os 10 exemplos, que teve um resultado de 100% e também calculou-se com qual probabilidade cada classe foi selecionada pelo algoritmo. Tais dados podem ser observados com maiores detalhes na tabela 1.

Tabela 1: Resultados obtidos com a execução do modelo gerado.

Original class	**Predicted class**	**Probability**
0	0	0.9936506
1	1	0.99602747
2	2	0.8937165
3	3	0.8628852
4	4	0.94235396
5	5	0.8028489
6	6	0.8394695
7	7	0.8616485
8	8	0.9361846
9	9	0.7603381
Total accuracy		**100%**