Relatório Projeto 2

Aluno: João Guilherme Madeira Araújo

Número USP: 9725165

Relatório do Projeto 2:

O exercício pediu a implementação de um modelo CNN(Convolutional Neural Network) para classificação de imagens usando o dataset CIFAR, em seguida geramos 10 exemplos de imagens, um de cada classe para a rede fazer inferências. O framework utilizado foi o Tensorflow com Keras, o programa que treina é o cnn.py e o que faz as inferências é o test.py.

Datasets

Foi usado o dataset CIFAR, que contém 60000 imagens RGB de 32x32 em 10 classes distintas, sendo elas avião, carro, pássaro, gato, veado, cachorro, sapo, cavalo, navio e caminhão.Das imagens 50000 fazem parte do conjunto de treinamento e os 10000 restantes do conjunto de testes. Cada imagem foi lida em formato de tensor 32x32x3 do tipo ponto flutuante e em seguida cada célula foi normalizada para valores entre 0 e 1. Os exemplos criados a mão para inferência estão na pasta dataset e tem o nome classe.in, onde classe é a sua classe..

A Topologia da Rede

O código-fonte da Convolutional Neural Network está no arquivo cnn.py. O arquivo contém a implementação de uma CNN de 10 camadas, que podem ser divididas em 3 blocos de 3 camadas e a camada densa final. Os blocos consistem em uma camada convolucional com filtros 3 x 3, função de ativação exponential linear unit e regularização L2, em seguida fazemos uma Batch Normalization e outra camada idêntica a primeira, outra Batch Normalization e terminamos com maxpool com filtros 2x2 e dropout. As diferenças entre os blocos está no número de filtros das camadas convolucionais: 32 no primeiro bloco, 64 no segundo e 128 no último, e a probabilidade de dropout: 20% no primeiro, 30% no segundo e 40% no terceiro. A camada densa de saída tme 10 neurônios e ativação Softmax.

Treinamento

O treinamento durou 30 épocas com função de perda entropia cruzada e método de otimização AdaDelta. O Modelo já treinado está no arquivo cifar.h5. Como se pode notar foi necessário o uso de muitos regularizadores e técnicas para evitar overfitting, como L2, Batch Normalization e dropout, devido ao elevado número de parâmetros e profundidade da rede.

Conclusões e Teste

A rede alcançou 90% de acurácia no treinamento e 83% nos testes como se pode ver na Figura 1, além disso conseguiu classificar corretamente 9 das 10 imagens testadas, como mostrado na Figura 2. Podemos concluir que Convolutional Neural Networks tem excelente capacidade de classificar imagens, porém vemos fraquezas tanto na necessidade de um número elevado de exemplos para treino quanto na precisão e cautela necessárias em projetá-las para evitar problemas de overfitting.

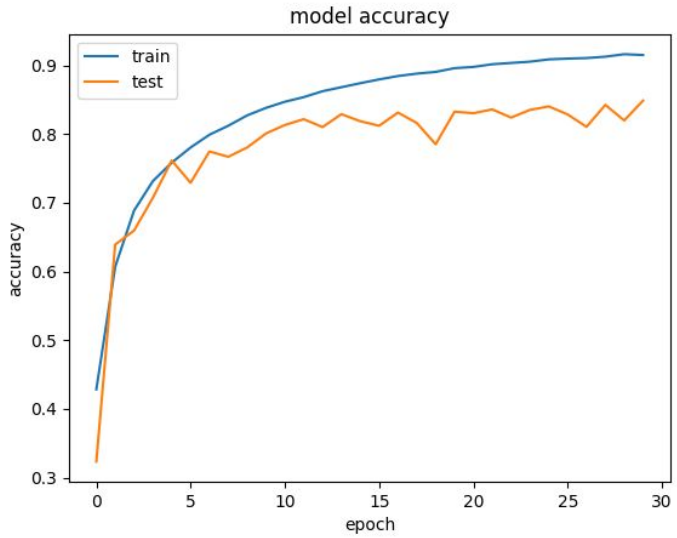

Figura 1: Acurácia do Modelo em função do número de épocas de treinamento

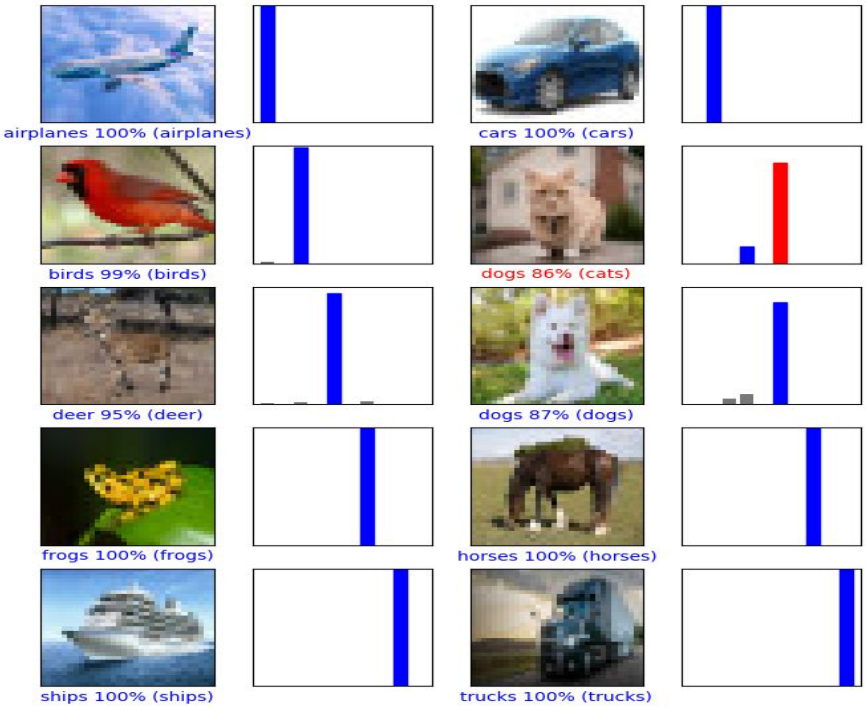

Figura 2: Imagens de teste e predições do modelo