

ICMC - USP

SCC0270 - Redes Neurais

Projeto 2

CIFAR-10 CNN

Docente: Roseli Aparecida Francelin Romero

Aluno: Lucas Antognoni de Castro, 8936951

São Carlos

Novembro 2017

Enunciado

Para o segundo projeto da disciplina foi implementada, treinada e testada uma rede neural para a resolução do problema de classificação de imagens. Codificou-se o modelo de Rede Neural Convolucional (*Convolutional Neural Network*) utilizando o pacote e bibliotecas do **TensorFlow**. Foram utilizados diferentes valores para os parâmetros da rede neural, a fim de verificar o impacto dos mesmo no aprendizado e posterior teste da mesma.

Plataforma utilizada

O referido trabalho foi confeccionado no sistema operacional **Linux** (*Ubuntu*), utilizando um ambiente de desenvolvimento com a **IDE Visual Studio Code** e terminal. O código foi desenvolvido na linguagem de programação **Python** e suas bibliotecas.

Dados

Foi utilizada a base de dados **CIFAR-10** para realizar a classificação de imagens. Os dados foram pré-processados utilizando as bibliotecas **cifar10.py**, **dataset.py** e **cache.py**, disponíveis no seguinte repositório da plataforma Github:

- <https://github.com/Hvass-Labs/TensorFlow-Tutorials>

Os dados são separados em 6 arquivos contendo 10000 imagens cada com os seguintes parâmetros (largura, altura, canais de cores):

[32, 32, 3]

Cinco dos seis arquivos são utilizados para treino (*data_batch_n*) e um para teste (*test_batch*).

Testes e Resultados

As rede neural utilizada apresenta as seguintes configurações:

- Primeira Camada Convolutiva
 - `W_conv1 = weight_variable([3, 3, 3, 16])`
 - `b_conv1 = bias_variable([16])`
- Segunda Camada Convolutiva
 - `W_conv2 = weight_variable([3, 3, 16, 32])`
 - `b_conv2 = bias_variable([32])`
- Terceira Camada Convolutiva
 - `W_conv3 = weight_variable([3, 3, 32, 64])`
 - `b_conv3 = bias_variable([64])`
- Camada Densamente Conectada
 - `W_fc1 = weight_variable([4 * 4 * 64, 2048])`
 - `b_fc1 = bias_variable([2048])`
 - `h_pool3_flat = tf.reshape(h_pool3, [-1, 4*4*64])`
 - `h_fc1 = tf.nn.relu(tf.matmul(h_pool3_flat, W_fc1) + b_fc1)`

- Camada de Dropout
 - `keep_prob = tf.placeholder(tf.float32)`
 - `h_fc1_drop = tf.nn.dropout(h_fc1, keep_prob)`
 - Utilizada para reduzir o overfitting.
- Camada Readout
 - `W_fc2 = weight_variable([2048, 10])`
 - `b_fc2 = bias_variable([10])`
 - `y_conv = tf.matmul(h_fc1_drop, W_fc2) + b_fc2`

Utilizou-se convolução com *stride* de valor 2 e *zero padding*, além de um *max pooling 2x2*, obtendo uma acurácia média de **67,2549%**, durante os testes com **10000** épocas.

Executando

Extraia os arquivos da pasta e em um terminal navegue até o diretório do projeto. Baixe a base de dados *CIFAR-10* para a linguagem de programação python (<https://www.cs.toronto.edu/~kriz/cifar.html>), extraia os arquivos na pasta *"data/"*, de maneira que o caminho para os dados seja *"data/cifar-10-batches-py/"*. Por último, digite o seguinte comando:

```
python3 cnn.py
```

O programa será executado e os resultados serão impressos na tela.