

A Convolutional Neural Net, some techniques and Augmentation

Micael Mota

I. PROBLEM DESCRIPTION

Given a digit image data-set, build and train a deep neural network using TensorFlow and techniques learned in class including data augmentation to classify the images.

II. ABOUT THE NET

The net architecture follows below sequence:

1. Images with dimension (71x77) as input;
2. Convolutional layer with 32 filters of size (3x3), step size (1x1), padding = 'valid' and relu activation;
3. Max pooling layer with pool size=(2x2) and strides=2
4. Convolutional layer with 64 filters of size (5x5), step size (1x1), padding = 'valid' and relu activation;
5. Max pooling layer with pool size=(2x2) and strides=2
6. A fully connected layer with 128 neurons and relu activation
7. Output an one hot array (1x10) with sigmoid activation.

III. TRAINING

The data-set was randomly divided in 80% for train and 10% for accuracy validation. The network was trained for 50 epochs using Stochastic Gradient Descent (SGD - Mini batches size of 16), learning rate decay and Dropout optimization methods.

IV. AUGMENTATION

For each training batch of size 8 we turned its size into 32 with random translations, random rotations and random scaling the images.

V. RESULTS

	Loss	Accuracy
Training	0.433	0.981
Validation	0.803	0.974

HOW TO USE

Run 'python3 project3.py' and type '1' for training on 'data_part1/train' images or '2' for inference on 'data_part1/test' images. If select 2, the program will output a text file in 'outputs/tf' with one line for each image in test folder with respective prediction.