Deep learning Lab
Uni Freiburg

Assignment 2
Maryna Kapitonova

Classifying MNIST with CNN in TensorFlow

Introduction
The task was to implement convolutional neural network (CNN) for MNIST digits classification using TensorFlow and evaluate its performance.

Implementation
CNN consists of two convolutional layers (16 3*3 filters and stride 1), interspersed with ReLu layers and a max pooling layer. After the convolutional layer a fully connected layer with 128 units and softmax layer are added.
Network is trained by optimizing the cross-entropy loss with stochastic gradient descent.
Validation performance is saved after each epoch (learning curves)

Convolutional layer:
Accepts a volume of size W1 x H1 x D1
Requires four hyperparameters:

- Number of filters K = 16
- Their spatial extent F
- The stride S
- The amount of zero padding P

Produces a volume of size W2 x H2 x D2

- W2 = (W1 - F + 2P)/S + 1
- H2 = (H1 - F + 2P)/S + 1
- D2 = K

With parameters sharing it introduces F*F*D weights per filter, for a total of (F8F8D1)*K weights and K biases
In the output volume, the d-th depth slice (of size W2xH2) is the result of performing valid convolution of the d-th filter over the input volume with a stride S, and then offset by d-bias

Common setting K - power of 2
Kernel width of convolution
Kernel height of convolution

Pooling layer
Makes the representations smaller and more manageable
Operates over each activation map independently

Using other optimizer

Results:

Validation losses for different learning rates using GPU computing