Albert-Ludwigs-Universität Freiburg, Institut für Informatik
Andreas Eitel
Deep Learning for Autonomous Driving Laboratory
Summer term 2018

Exercise 1

Topics: traffic sign recognition, CNN

Submission deadline: May 04, 2018

Submit to: `eitel@informatik.uni-freiburg.de`

General Notice

This exercise should be solved individually. The source code of programming exercises should be submitted as a link to your github/bitbucket repository. Put learning curve plots and tables in a report (max. 1 page).

We will be using PyTorch for the programming exercises https://github.com/pytorch/pytorch

Figure 1: German Traffic Sign Recognition Benchmark (GTSRB) dataset

Task 1: Getting familiar with pytorch and the pool computers

Setup pytorch on the pool computers in a virtual python environment/anaconda and get started with a CNN tutorial.

- Connect to the main server: ssh -X <username>@login.informatik.uni-freiburg.de
- Do not run processes on the main server. Connect to one of the pool computers there instead. Preferably use tfpool25-46 or tfpool51-63 they provide the best GPUs: ssh tfpoolXX
- Replace XX with a 2 digit number of the computer you want to connect. Before running make sure no one else is using the selected computer. Also

make sure your program is running on the GPU and not on the CPU! Do not use 100% of the GPU go for 80% memory capacity so others can still use the computer. Terminal: who top nvidia-smi

- Start a screen session: screen -S my_training
- Detach from screen using: ctrl+a+d
- Login back into screen: screen -ls screen -r my_training
- Write down on which computer you started your screen session.

Task 2: Train a CNN on the German Traffic Sign Recognition Benchmark

The task is to train a CNN classifier for recognizing different traffic signs from 40 classes. The GTSRB dataset can be found here http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset. It contains more than 50,000 images from 40 classes.

- Process the dataset (resize images to 32×32, mean subtraction, split up official GTSRB training set into training 80% and validation 20%)). http://pytorch.org/tutorials/beginner/data_loading_tutorial.html
- Start with a standard architecture (several layers of 3×3 conv filters, stride 1 and 2×2 max pooling, stride 2 and ReLU activation, one fully connected layer with 512 neurons, one softmax layer). Similar to VGG-16 for CIFAR-10.
- Train the CNN network using cross-entropy loss with stochastic gradient decent until convergence. Make sure to plot the validation performance after each epoch in a figure, e.g., as learning curve.

Task 3: Optimize the CNN performance

- Try different learning rates and plot the learning curve for $\{0.01, 0.001, 0.0001\}$
- Try to improve the performance of your architecture on the validation set. Things you can try out/change: dropout, weight decay, activation functions (PReLU, ELUs), batch norm, number of filters, filter kernel size
- Report your performance on the official test set after re-training with the best architecture found on the validation set.