

Deep Learning Course – Assignment 2

Convolutional neural networks

Submission date: 20/05/21, 23:55

Please submit the assignment via Moodle

General Instruction:

- Solitary submissions or in pairs.
- The code must be written in Python 3.7.1 and run on GPU in google colab
- You are allowed to use only Pytorch, numpy, Tkinter, pandas and matplotlib as external libraries
- The code must be reasonably documented

In this assignment you'll implement a module for traffic sign classification using CNNs. You'll decide about the architecture and training details. You'll also build a GUI for your traffic sign classifier.

Dataset:

- Use the following dataset from Kaggle:
<https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign>
The dataset contains more than 50,000 images of different traffic signs. It is further classified into 43 different classes
- For loading it to google colab you can upload it to google drive using the following instructions (not mandatory):
<https://towardsdatascience.com/downloading-datasets-into-google-drive-via-google-colab-bcb1b30b0166>
- Load the train and test examples, resize the images to size of 3*30*30.
- Split the training examples to training and validation sets.
- Build Dataset and DataLoader for the training set, validation set and test set.
- Use batch_size = 64, num_workers=4

Model:

- Design a CNN model for traffic sign classification.
- You should decide about the architecture: number of convolutional layers, number of kernels in each layer, size of kernels, number and size of fully connected layers, activation functions, etc.
- Please define the following models:
 1. Model without dropout and batch normalization.
 2. Model with dropout and batch normalization.
 3. Model with dropout and batch normalization and without FC layers (convert your network to fully convolutional network as we saw in the tutorial).

Training:

- Write the training procedure for our task
- Decide what is the best optimizer for this task, learning rate and other hyperparameters
- Use Cross Entropy loss function
- Please use early stopping regularization
- Train the 3 models

Evaluation:

- Evaluate your 3 models on the test set.

GUI:

- Build a graphical user interface for our traffic signs classifier with Tkinter. Tkinter is a GUI toolkit in the standard python library.
- In the GUI, we'll have an option to upload an image
- The image will be shown on the GUI
- We'll have a button used to classify the image. Please use the attached dictionary to present the details about the sign instead of a label.
- Please use the third model for the GUI

Submission instructions:

Your pdf report should contain:

- Model architecture description, training procedure (regularizations, optimization details etc).
- Two convergence graphs for each of the 3 models - loss as a function of time (epochs), and accuracy as a function of time. Each graph should depict both training and validation performance.
- Report the test accuracy for each of the 3 models.
- Explain the difference between the 3 models based on the number of parameters, graphs and results.
- Add a short summary of your attempts (architecture, optimizers, etc.) and summary your conclusions.
- GUI print screens of 2 examples

In addition, you should also supply:

- Code able to reproduce your results.
- GUI code.
- A function that loads the third trained network and returns the average error on the test set and the accuracy (check save and restore models in Pytorch as we talked in the tutorial).

Grade's policy:

- Successful submission - 55 points
- Report - 25 points
- Competition - 20 points