**Deep learning – HW2 (CNN)**

**Data:**

Dataset: German Traffic Sign Recognition Benchmark

Splits: We use the data descripted in the csv files. For validation set, we took 20% from the un-augmented train set.

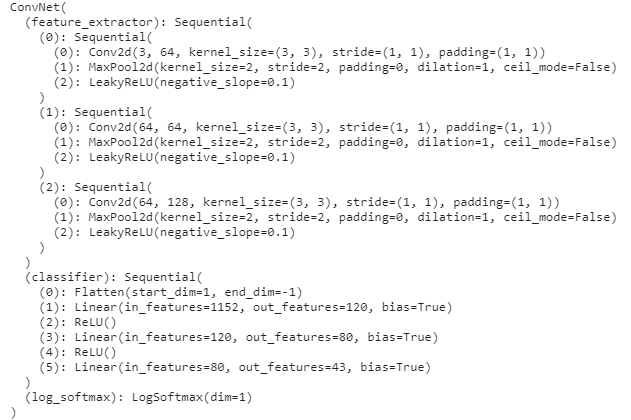
Data augmentation:

We apply augmentation on the train set and concatenate the original (normalized) training set with the augmented one. This approach helped the model to converge faster and proved to be beneficial.

**Model architecture:**

Feature extractor:

Without dropout and batch normalization:

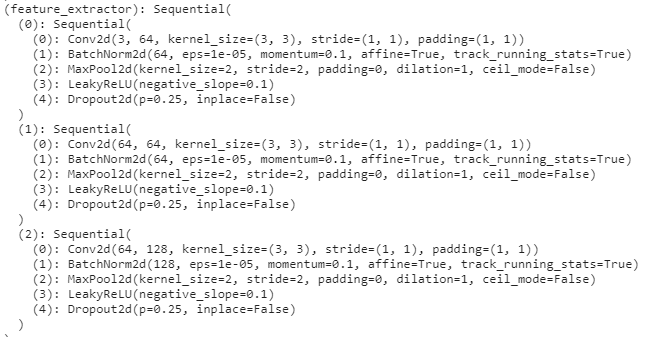


First convolutional layer with output channel = 64, kernel size = 3 and padding = 1 to keep dimensions, followed by max pooling the leaky relu as activation with negative slop = 0.1.

Second convolutional layer with output channel = 64, kernel size = 3 and padding = 1 to keep dimensions, followed by max pooling the leaky relu as activation with negative slop = 0.1.

Third convolutional layer with output channel = 128, kernel size = 3 and padding = 1 to keep dimensions, followed by max pooling the leaky relu as activation with negative slop = 0.1.

With dropout and batch normalization:

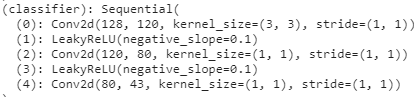
****

Same as before but in this case each of the Conv2d layer will be followed by a batch normalization, and each block will be ended with a dropout with p=0.25 as the probability.

Classifier:

Fully connected classifier:

Fully convolutional classifier:



First, we will lower the height and width to 1 using kernel size = (output height, output width). After, we will apply leaky relu, and continue by 2 layers of 1x1 convolution to decrease the number of channels to be equal to number of classes (43).

**Training:**

Optimizer: Adam.

Loss function: Cross entropy.

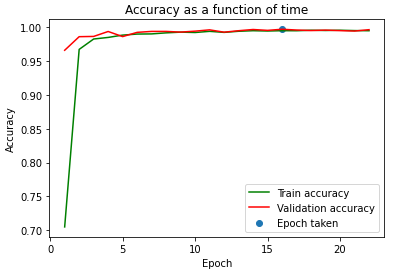
Learning rate: 0.001.

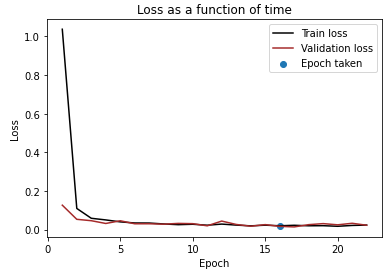
We use decreasing learning rate with step size = 10 and gamma=0.1, which means that every 10 epochs our learning rate will decrease by a factor of 0.1.

We applied Early stopping with patience = 5 to avoid over-fitting and take the best model.

Max epochs = 50.

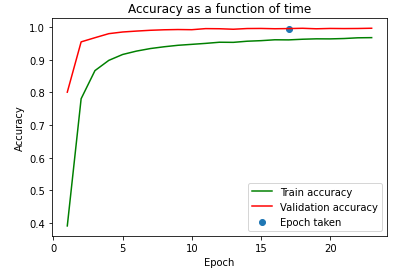


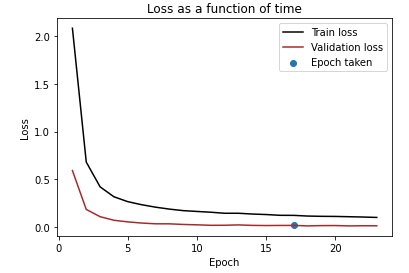




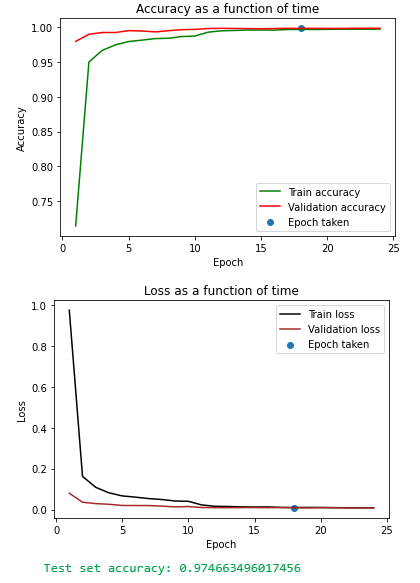
**Model with dropout and batch normalization (fully connected layers):**







**Model with dropout and batch normalization (fully convolutional):**





**GUI print screens of 2 examples:**





Dropout P = 0.5:

