Computer Vision
CSCI-GA.2272-001 - Fall 2017
Assignment 3

Lakshay Sharma
ls4170@nyu.edu

Submitted: Dec. 7, 2017

1 Introduction

This assignment is a Kaggle competition on The German Traffic Sign Recognition Benchmark [1]. The objective it to produce a model that gives the highest possible accuracy on the test portion of this dataset.

2 Dataset

The dataset has over 50,000 images, each belonging to one of 43 classes.
We split this into train-val-test sets with the following number of samples:

- Training set: 35339
- Validation set: 3870
- Test set: 12631

2.1 Visual inspection

[1] http://benchmark.ini.rub.de/?section=gtsrb&subsection=news

Figure 1: GTSRB dataset examples

2.2 Normalization

The image pixels (after re-scaling) are normalized to zero-mean and standard deviation of 1.

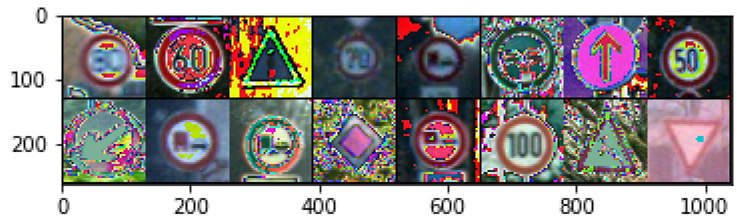

Figure 2: Normalized samples

2.3 Distribution of classes

The following shows a distribution of the classes in the training and validation sets.

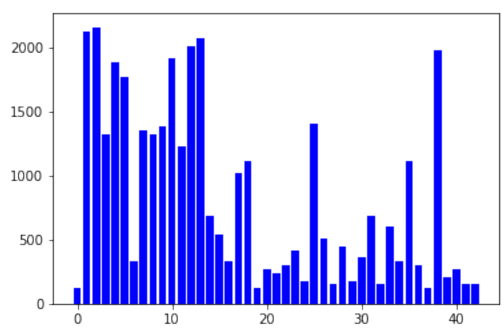

Figure 3: Training set class distribution

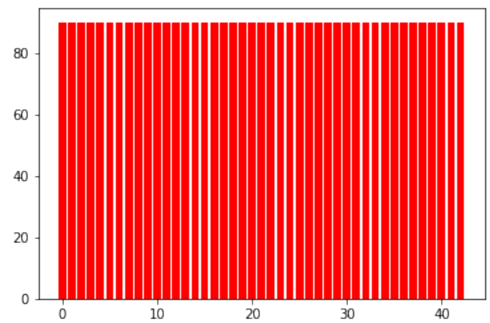

Figure 4: Validation set class distribution

3 Experiments

Unless specified otherwise, the fixed training parameters are as follows:

Batch size: 64
Optimization method: stochastic gradient descent
Learning rate: 0.01
SGD momentum: 0.5

3.1 Baseline model

The baseline model provided as part of this assignment is described below.

Input (3x32x32)
Conv 5x5 10 fm
Max-pool 2x2
ReLU
Conv 5x5 20 fm
Dropout p=0.5
Max-pool 2x2
ReLU
FC 500 -> 50
Dropout p=0.5
FC 50 -> 43
Softmax 43

Table 1: Baseline model architecture

Train acc.	Val acc.	Test acc.
.8773	.9225	-

Table 2: Baseline model results (after 20 epochs)

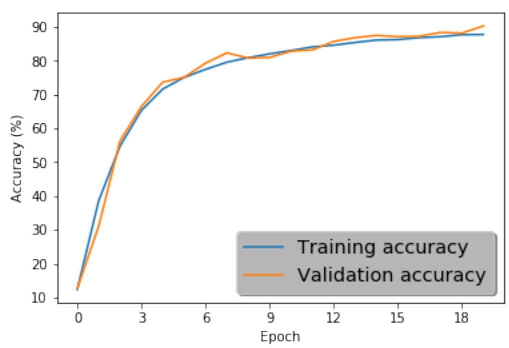

Figure 5: Baseline model epoch vs. accuracy

3.2 Increasing width

The baseline model was modified to create a wider model. Compared to the baseline model, this model fits the training data better, and also has better accuracy on the validation data.

Input (3x32x32)
Conv 5x5 16 fm
Max-pool 2x2
ReLU
Conv 5x5 128 fm
Dropout p=0.5
Max-pool 2x2
ReLU
FC 3200 -> 128
Dropout p=0.5
FC 128 -> 43
Softmax 43

Table 3: Wider model architecture

Train acc.	Val acc.	Test acc.
.9810	.9271	-

Table 4: Wider model results (after 20 epochs)

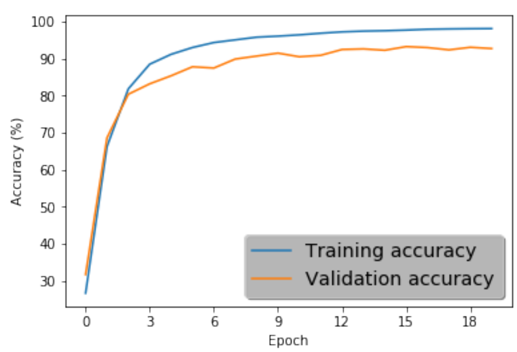

Figure 6: Wider model epoch vs. accuracy

3.3 Increasing depth

The baseline model was modified to create a deeper (and slightly narrower in the FC layers, as a consequence) model. This model performs worse than the original baseline model, as well as the wider model. It is also slower to converge.

Input (3x32x32)
Conv 5x5 10 fm
Max-pool 2x2
ReLU
Conv 5x5 10 fm
Dropout p=0.5
Max-pool 2x2
ReLU
FC 160 -> 160
FC 160 -> 100
Dropout p=0.5
FC 100 -> 43
Softmax 43

Table 5: Deeper model architecture

Train acc.	Val acc.	Test acc.
.9104	.8369	-

Table 6: Deeper model results (after 20 epochs)

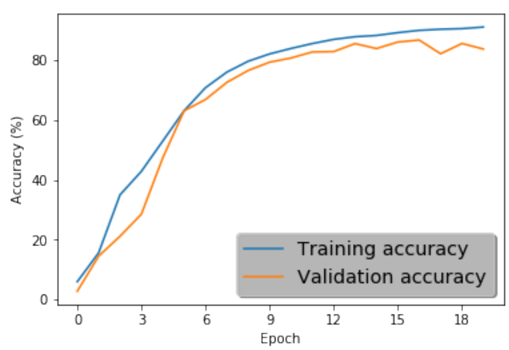

Figure 7: Deeper model epoch vs. accuracy

3.4 Increasing input dimensions

Prior to this model, the input images were of dimension 32x32 pixels. The input dimensions are increased to 64x64, and it is seen with a model similar to the wider model (3.2) that this yields improvements in prediction accuracy.

Input (3x64x64)
Conv 5x5 16 fm
Max-pool 2x2
ReLU
Conv 5x5 128 fm
Dropout p=0.5
Max-pool 2x2
ReLU
FC 21,632 -> 128
Dropout p=0.5
FC 128 -> 43
Softmax 43

Table 7: 64x64 input size model

Train acc.	Val acc.	Test acc.
.9879	.9325	-

Table 8: 64x64 input size, (after 20 epochs)

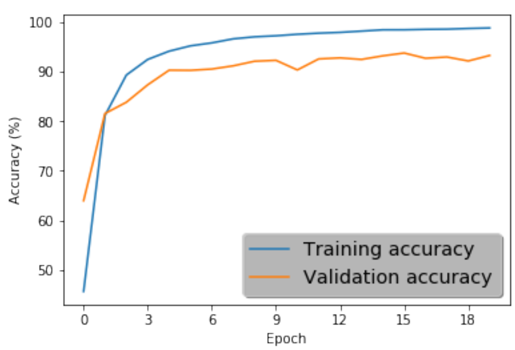

Figure 8: 64x64 input size, epoch vs. accuracy

3.5 Increasing dropouts

Based on the experiments prior to this model, a wider and deeper model (extra convolution layer, and extra masks at each convolution layer) is created. Various dropout layer combinations were tried, and it was found that, in most cases, increasing the number of dropout layers (specially for the conv layers) improved accuracy significantly. Note: the best validation result was found to be after the 17^{th} epoch (accuracy .9617). The test accuracy reported below is for the parameters following this epoch.

Input (3x64x64)
Conv 5x5 16 fm
Max-pool 2x2
ReLU
Conv 5x5 128 fm
Dropout p=0.5
Max-pool 2x2
ReLU
Conv 5x5 256 fm
Dropout p=0.5
Max-pool 2x2
ReLU
FC 6,400 -> 300
Dropout p=0.5
FC 300 -> 43
Softmax 43

Table 9: Increased dropout model architecture

Train acc.	Val acc.	Test acc.
.9908	.9604	.9637

Table 10: Increased dropout model results (after 20 epochs) (test acc. results for parameters after 17^{th} epoch)

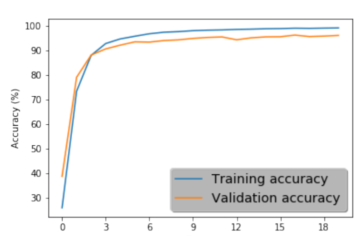

Figure 9: Increased dropout model results, epoch vs. accuracy

3.6 Applying batch normalization

Applying batch normalization was found to yield significant performance improvements. The model below was tested with various permutations of batch normalization at the conv layers, and the best seeming model was then trained for 50 epochs. Note: the best validation result was found to be after the 28^{th} epoch (accuracy .9813). The test accuracy reported below is for the parameters following this epoch.

Input (3x64x64)
Conv 5x5 16 fm
Batch norm
Max-pool 2x2
ReLU
Conv 5x5 128 fm
Dropout p=0.5
Max-pool 2x2
ReLU
Conv 5x5 256 fm
Batch norm
Dropout p=0.5
Max-pool 2x2
ReLU
FC 6,400 -> 300
Dropout p=0.5
FC 300 -> 43
Softmax 43

Table 11: Batch normalized model architecture

Train acc.	Val acc.	Test acc.
.9976	.9638	.9830

Table 12: Batch normalized model results (after 30 epochs) (test acc. results for parameters after 28^{th} epoch)

.

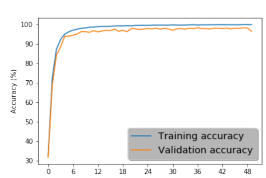

Figure 10: Batch normalized model, epoch vs. accuracy

3.7 Fine-tuned model

Fine tuning the model from 3.6 yields some improvements in prediction accuracy. After initial training as described, the model is re-trained with a smaller learning rate (0.001) for 50 epochs. The best parameters are found to be after epoch 2, and the results for this are shown below.

Train acc.	Val acc.	Test acc.
.9988	.9826	.9840

Table 13: Batch normalized model results (after epoch 2)

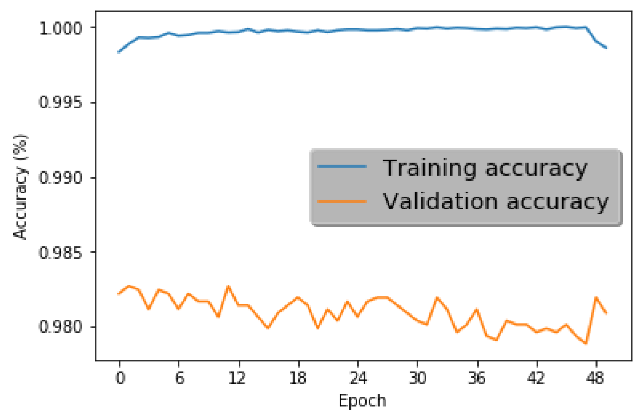

Figure 11: Fine-tuned normalized model, epoch vs. accuracy