Deep Learning :: Assignment A1 February 9, 2015
By Team DeepBlue: Priyank Bhatia, Emil Christensen, and Peter Varshavsky

Architecture

The architecture of the model provided is a convolutional neural network with two stages and a classifier. The input has 3 feature maps, each 32x32 pixels.The first stage performs convolutions using a tanh squashing function with a 5x5 filter to produce 64 feature maps to which L2 pooling with pooling size 2x2 is applied. A subtractive normalization module using a Gaussian kernel of size 7 is then applied before feeding the 64 14x14 outputs to the next stage.The second stage performs the convolutions and poolings with the same filter sizes, pool sizes, number of feature maps, and normalization function as the first stage except that the 16-dim feature maps are projected into 256-dim maps before feeding the resulting data into a 2-layer non linear classifier with 128 hidden units, which uses the tanh function for non linear transformation. A Class Negative Log-Likelihood (nll) Loss function is used along with Stochastic Gradient Descent (SGD) for optimization and learning rate of 1e-3, mini-batch size of 1, and momentum 0 number of max iterations is equal to 1. Despite our attempts to improve the model, we found that the original model (as provided by Clement Farabet's tutorial) yielded the best results.

Learning Techniques

Non Linear Transformation Functions	**Test Accuracy**
Tanh	89.24
ReLU	86.12

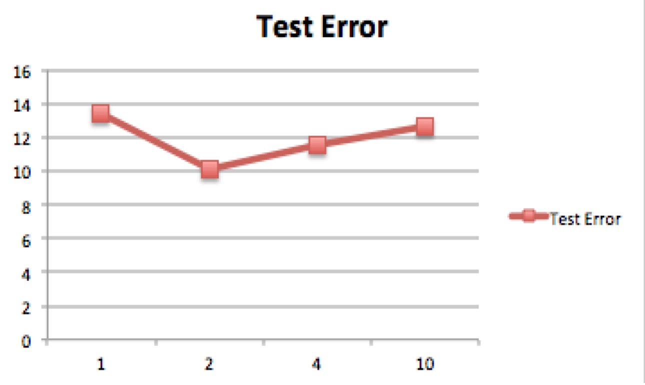
We compared performance of LP pooling, max-pooling and average pooling and found below results after 1 epoch:

Pooling Function	**Test Accuracy**
MaxPooling	85.302
LPPooling	89.24
Average Pooling	84.8

Below values were noted for test error for poolsize =1,2,4,10 after 1 epoch and P=2 was seen to be showing the least error. Also LP Pooling used and it was showing accuracy of 92.5 on test set after 3 epochs.

Poolsize vs Test Error Graph

PoolSize	**Test Error**
1	13.4
2	10.1
4	11.5
10	12.6



LP Pooling (with poolsize=2) was seen to be showing most optimal results with test accuracy of 92.5% after 3 epochs. Different loss functions were used to identify the most optimal loss function. Their performance is shown below:

Loss Function	Test Accuracy
NLL	89.24
Multi Margin	85.056

Optimization Function

The optimization function Stochastic Gradient Descent (SGD) was seen to be showing the most optimal results compared to LBFGS and CG optimization algorithm.

Optimization Function	Test Accuracy
SGD	89.24
LBFGS	84.78
CG	85.24

Cross Validation

The SVHN classification dataset contains 32x32 images with 3 color channels. The train set is divided in four ways to produce validation sets for cross validations. In this we use 4 different models to evaluate the performance of each model.

Model	CV1 Accuracy (Test Accuracy)	CV2 Accuracy (Test Accuracy)	CV3 Accuracy (Test Accuracy)	CV4 Accuracy (Test Accuracy)
Original	90.97% (90.53%)	90.37% (90.44%)	90.8% (90.85%)	90.87% (90.7%)
learning rate 1e-2 & L2 Pooling	89.98% (90.89%)	90.47% (90.73%)	90.88% (90.85%)	90.24% (90.29%)
learning rate 1e-2 & L4 pooling	90.1% (90.74%)	90.76% (91.29%)	91.19% (90.98%)	91.26% (91.35%)
learning rate 1e-3 & L4 pooling	90.3% (90.44%)	89.95% (90.02%)	90.07% (90.98%)	89.76% (91.35%)

Results

When the same four models were trained on the extra data set the original model performed best with test accuracy of 95.31%. Model 2 achieved 93.51% test accuracy after 1 epoch. Model 3 achieved 94.92% test accuracy after 3 epochs.Model 4 achieved 94.63% test accuracy after 3 epochs.

Reference

Sermanet, Pierre, Chintala, Soumith, and LeCun, Yann. “Convolutional neural networks applied to house numbers digit classification.” CoRR, abs/1204.3968, 2012a.

A Note On Collaboration

While working on this assignment, we (team Deep Blue) collaborated closely with the members of team Elucid (Eduardo Franco, Jiamin Xuan and Yen-Cheng Liu). This includes sharing models, insights and results. Our final model submission - here and on Kaggle - are products of our collaboration and thus have identical results.
Code available here: http://cims.nyu.edu/~erc399/result.lua