A2 homework submission
Team: the gurecki
Deep Learning 2015, Spring

David Halpern
Department of Psychology
New York University
david.halpern@nyu.edu

Anselm Rothe
Department of Psychology
New York University
ar3918@nyu.edu

Alex Rich
Department of Psychology
New York University
asr443@nyu.edu

1 Architecture

1.1 Input

The input is a 3D array with three 2D feature maps of size 32×32. We preprocessed the training data by normalizing across features so that each feature had a mean of 0 and a standard deviation of 1 and then used the same normalization for the test data. The data was also normed per pixel so each pixel across images had a mean of 0 and a standard deviation of 1. Each image is padded with zeros of size $2 \times 2 \times 2 \times 2$ in order to make for better convolutional filter learning.

1.2 Model

The first layer applies 23 7×7 convolutional filters with a stride of 2. These filters have a ReLU activation function and are fed into a max pooling layer which takes the max of each 3×3 region with a step size of 2×2. Each filter is now 22×22. This they are all fed into a linear layer with 50 outputs nodes. The outputs are then run through a Tanh activation function and finally put through another linear layer with 10 output nodes, one for each category. The model then uses log softmax to convert energies into choices.

2 Learning Techniques

We used dropout to train the convolutional layers.

3 Training Procedure

We used the to train our model. The model was trained using stochastic gradient descent with a batch size of 128, a momentum of .9, no weight decay and a learning rate of .01. In addition, the learning rate was halved every 20 epochs. We used the typical negative log likelihood criterion as the loss function. We used a validation set of 500 of the original 5000 labeled training examples. The optimization procedure was run for 100 epochs over the remaining 4500 training data using GPUs.

At the end of training, our performance on training data was 82.4%, our performance on the validation set was 54.2 % and our performance on test was 56.775%

4 Experiments

These experiments did not help to boost the performance and thus where not included in our final submission.

4.1 Unlabeled data

As per Xiang Zhang's suggestions from class [2], e attempted to use uniform prescription to train our model using the unlabeled data. In order to do this, we used KL Distance as our loss function so that the target for the unlabeled data could be a uniform distribution on all of the categories. The targets for the labeled data were a delta function with all probability mass on the true category. We tried using various amounts of unlabeled data from 500 to all 100000 but this did not seem to improve performance.

4.2 Data augmentation

Following Dosovitskiy et al. (2014), we converted all images into HSV color space. We inflated the train data set by a factor of 3 and applied each of the following transformations (thereby flipping a 0.5 coin to decide whether or not to apply the transformation):

1. A up-left or right-down shift by 2-10 pixels (randomly determined)
2. A horizontal flip
3. A rotation by 6-10 degrees clockwise or counterclockwise

4.3 Model Averaging

We also attempted model averaging, by running seven different iterations of the training procedure starting with different random seeds and then allowing the seven models to "vote" on each image, with the modal response being chosen. Interestingly, this also had no effect on performance, even though there were many images on which the models disagreed.

References

[1] Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., & Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. *Advances in Neural Information Processing Systems*, pp. 766-774.

[2] Xiang Zhang (2015). *How to Train STL-10 Well*. http://goo.gl/xJGvyH