# Deep learning: A1 homework submission

David Halpern
Anselm Rothe
Alex Rich

February 9, 2015

## 1 Architecture

**Input** The input is a 3D array with three 2D feature maps of size $32 \times 32$. 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 were also then locally normalized using contrastive normalization with a $13 \times 13$ Gaussian kernel.

**Stage 1**

1. The first layer applies 64 convolutional filters to the input map, each being $5 \times 5$, with a stride of 1. The receptive field of this first layer is $5 \times 5$, and the output produced by it is therefore $64 \times 28 \times 28$.
2. This linear transform is then followed by a non-linearity (tanh).
3. and an L2-pooling function, which pools regions of size $2 \times 2$, and uses a stride of $2 \times 2$ so the receptive field is now $7 \times 7$ The result of that operation is a $64 \times 14 \times 14$ array, which represents a $14 \times 14$ map of 64-dimensional feature vectors.
4. Finally, there is a subtractive normalization step with kernal constructed from a 1D Gaussian of length 7.

**Stage 2** Stage 2 is a repetition of Stage 1. The result is a $64 \times 5 \times 5$ array.

**Stage 3** In Stage 3 we first flatten all the features to get 3200 features. This gets fed into a two layer linear neural net with a hidden layer of size 128. The output is a vector of size 10, where each value indicates the likelihood of the labels '1' to '10' (0).

## 2 Learning Techniques

We did not use dropout and used the extra training data with for a total of 604388 unique examples.

## 3 Training Procedure

We trained our model using stochastic gradient descent with a learning rate of .001, no mini-batches, and momentum and weight decay set to to 0. We chose negative log-likelihood as loss function. We did not implement validation and thus validation and thus all trian data was used for training. The optimization procedure was run for three epochs over all training data.

The model's performance on the training data in the final epoch was 97.73%. The performance on the test data was 95.37%.