

1 Architecture

Input The input is a 3D array with three 2D feature maps of size 32 x 32.

Stage 1

1. The first layer applies 64 filters to the input map, each being 5x5. The receptive field of this first layer is 5x5, and the maps produced by it are therefore 64x28x28.
2. This linear transform is then followed by a non-linearity (tanh)
3. and an L2-pooling function, which pools regions of size 2x2, and uses a stride of 2x2 so the receptive field is now 7x7. The result of that operation is a 64x14x14 array, which represents a 14x14 map of 64-dimensional feature vectors.
4. Finally, there is a subtractive normalization step with kernel constructed from a 1D Gaussian of length 7.

Stage 2 Stage 2 is a repetition of Stage 1. The result is a 64x5x5 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 '0' to '9'.

2 Learning Techniques

We did not use dropout and used the extra training data with 604388 examples.

3 Training Procedure

We used a Learning rate of .001. We used no mini-batches, and set the momentum and weight decay to 0. We used negative log-likelihood as loss function. We decided to work without a validation sample and thus used all train data for training.

The training error was

The test error was