Critique of Backpropagation Applied to Handwritten Zip Code Recognition

This paper presents almost the full-scale CNNs used currently except for the creation of many activation functions as against the only sigmoid function utilized then. Let us dive into the experiment conducted which is now popularly known as MNIST.

The Dataset

The dataset was collected from the handwritten zip codes on posts of US Mail. Each of these digits were segmented and brought down to scale of 16×16 $(= 256)$ pixel image. The image is in grey scale due to the linear transformation of bringing down to scale and the grey scaled values vary between -1 to $+1$.

The Network Architecture

Input: 16×16 *image.*

Kernel Size: 5×5

Activation Function: Sigmoid

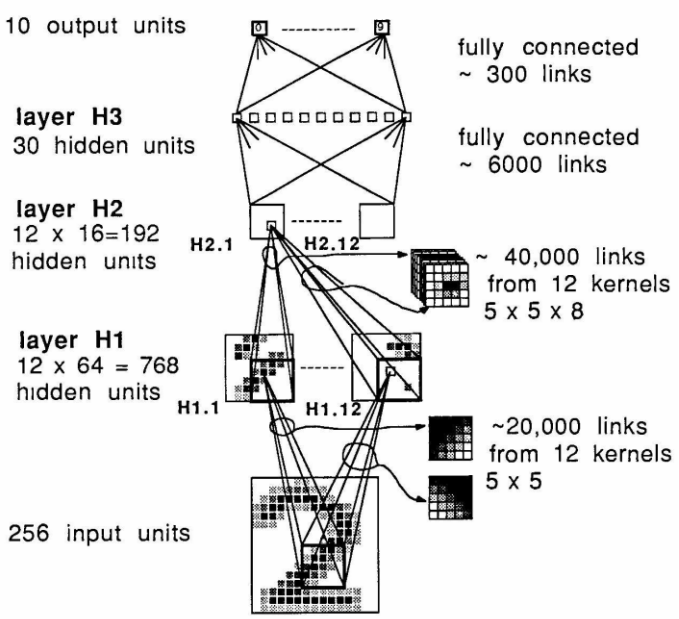
Hidden Layers:

H1: 8×8 $(images) \times 12$

H2: $4 \times 4 \times 12$

H3: 30 *units*

Output: 10 *units* $(0$ to $9)$



The network is fully connected and there are no connections made by hand as has been done in other experiments. The kernel size is 5×5 as mentioned above in Hidden Layers 1 and 2 and both the layers have 12 kernels/filters. The filter weights are learnt through backpropagation after random initialization using uniform distribution in the range $-2.4/F$ to $2.4/F$. The filters are convoluted across the image with strides of 2. Since each filter has the same weights for the whole image, it is termed as ***'weight sharing'.***

The output having 10 units denoting the 10 classes are one hot encoded. The Loss Function used is Mean Squared Error. Back Propagation is done as usual, i.e., computing the gradients of each of the weights and updating them. Stochastic Gradient Descent was utilized for minimizing the error wherein backpropagation is performed for each forward pass of an example.

The notable results include the achievement of 0.14% error on the training set while 5% on the test set. Another parameter for evaluation considered is the no. of rejections for achieving a certain error rate in the test set. In this case, to achieve 1% error rate on the test set, 12.5% rejections on the test set had to be made.

Other kernel sizes like 3×3 were tried but did no yield a better result. Another important point that must be noted that it took 3 days for the model to train on one of the best systems that existed then! However, the inference time is quite fast at 10 to 12 classifications per second and more than 30 classifications per second on normalized images.

This is paper is extremely relevant as this is probably the first paper that shows the application of Convolutional Neural Networks on real world datasets!