BITS Pilani
K K Birla Goa Campus

COURSE TITLE: MACHINE LEARNING (BITS F464)

PREPARED BY :

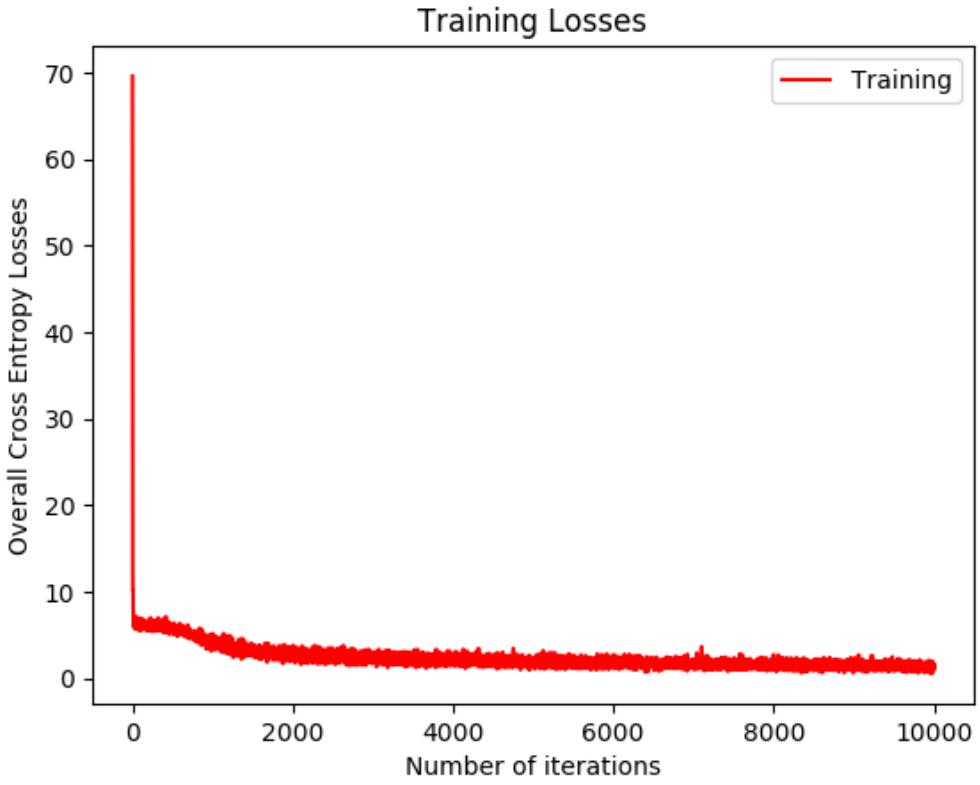
MAITHELI BISWAS
HARITHA D.
KASHYAP GAJERA

Predicting SVHN sequence of up to 5 numbers

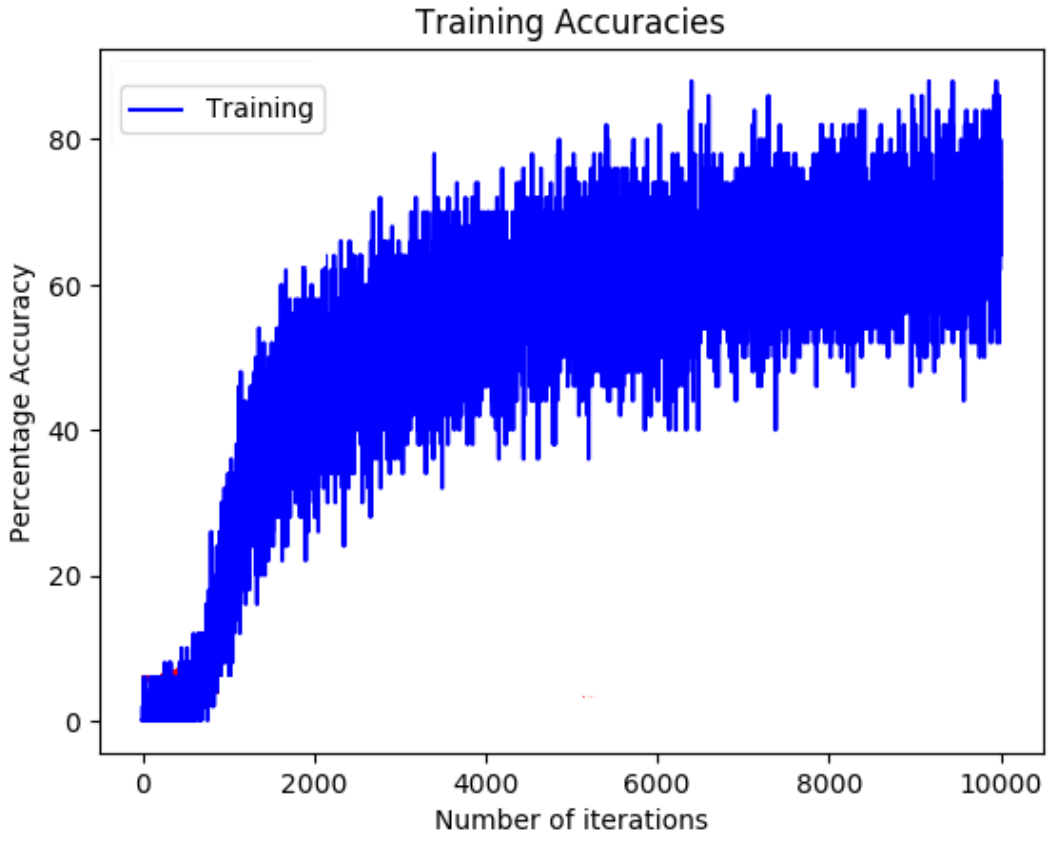
Design choices:

1. Using Tensorflow
2. 5 classifiers (1 per digit)
3. 11 bits (one-hot) to represent each label (0-9 and NaN)
4. Normalized the train data-
   For obtaining all data in same scale
5. Converted to greyscale -
   This is computationally less complex because of higher in dimensionality, and still does not provide guaranteed better results.
   Also, the test-time was mentioned to be limited, so we optimized on this.
6. Images are resized as CNNs accept only fixed size inputs: width = 32, height = 32, channels = 1
7. Filter size= 5 x 5, learning rate=0.001
8. All weights are initially sampled from a Gaussian distribution;
   biases are initalized to 1.0 constant.
9. Layer1: (5,5,3,16)
   Layer2: (5,5,16,16)
   Layer3: (1024, 128)
   Layer4: (128,10)
10. Keep probability value=0.93
    The reason that randomly ignoring nodes is useful is because it prevents inter-dependencies from emerging between nodes (I.e. nodes do not learn functions which rely on input values from another node), this allows the network to learn a more robust relationship.
    Implementing dropout has much the same effect as taking the average from a committee of networks.
11. AdamOptimizer
    It uses **moving averages of the parameters** (momentum)
    This enables Adam to use a larger effective step size, and the algorithm will converge to this step size without fine tuning.
12. Batch size= 50
13. Randomsampling of batches, 20000 iterations.
14. Relu: $f(x) = Max(0, x)$
    To add non-linearity to the network, so as to be able to computer non-linear functions.
    It can be computed more efficiently compared to more conventional activation functions like the sigmoid and tanh, without making a significant difference to generalisation accuracy.
15. Max pool
    Since we are concerned with whether the feature is present in a particular locality or not. Averaging in the region will reduce this probably of finding the feature.

Training Loss function graph:



Training Accuracy graph:



The network:

ImagePreprocess -> Conv1->Relu1->Pool1->Conv2->Relu2->Pool2->Flatten->FC1
->Relu3->FC01 ->Softmax01
->FC02 -> Softmax02
->FC03 -> Softmax03
->FC04 -> Softmax04
->FC05 ->Softmax05

1. Images are resized to 32x32x1
2. Batch is passed through Convolutional Layer 1 with weights (5,5,3,16) (images are padded with zeros so as to not shrink the dimension of the output, also to not lose information from image)
3. The output is passed through Relu layer
4. Max pooled using ksize=[1,2,2,1] and strides=[1,2,2,1]
5. Output is passed through Convolutional Layer 2 with weights (5,5,16,16), padded wth zeros.
6. Ouput is passed through Relu layer
7. Max pooled using ksize=[1,2,2,1] and strides=[1,2,2,1]
8. Output is Flattened and passed to 5 fully connected layers without dropout
9. AdamOptimizer is used to minimize the sum of crossentropy losses of the classifiers