CSA 250: Deep Learning Project II Report

Brijkumar Chavda

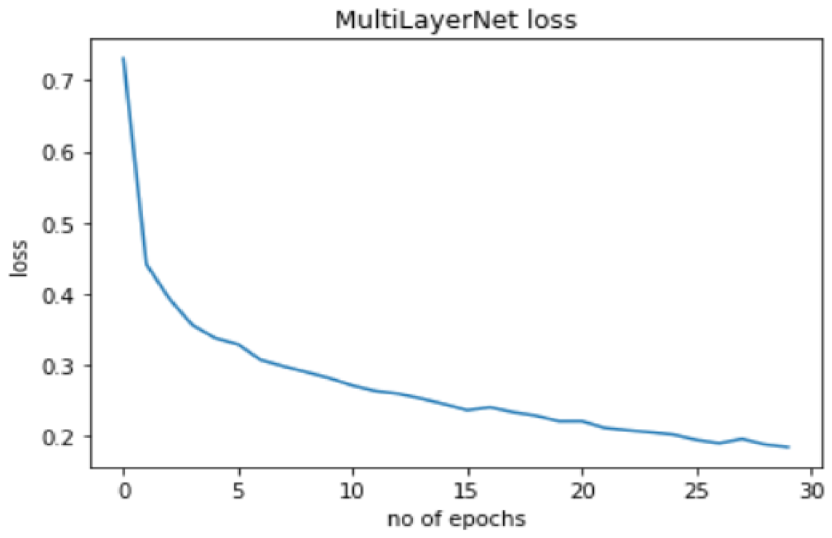
Multi-Layer Perceptron:

- Architecture:
  - Training and validation accuracy are used as parameters for developing architecture of this model.
  - Input Layer:
    - ✓ 784 pixels are taken as input by first normalising (28,28,1) grey scale images (dividing by max value = 255) and then flattening it
  - Hidden Layer:
    - ✓ Number of layers increased starting from two to three while maintaining number of neurons constant 700 to check improvement in validation accuracy. Starting with 700 neurons because input size is 784 and number of neurons in hidden layer could be less than number of neurons in the input.
    - ✓ At first, three layers with 700 neurons are taken but model was overfitting data (train accuracy ~ 95%, validation accuracy ~ 88%), then number of neurons reduced, and dropout were used.
    - ✓ After that number of neurons reduced to 512 and dropout (each layer) of 10% and 20% were checked (train accuracy ~ 94%, validation accuracy ~ 88.5%)
    - ✓ After that number of neurons reduced to 256 and dropout 20% were checked (train accuracy ~ 93%, validation accuracy ~ 89%).
    - ✓ After that one extra layer was introduced and dropout of second layer was removed which leads to train accuracy ~ 93% and validation accuracy ~89.50%.
  - Output Layer:
    - ✓ Output layer consists of 10 neurons because output of it is probability of ten classes so to generate 10-dimensions (10 probability), ten neurons are used.
  - Optimizer:
    - ✓ Adam optimizer was used for optimization during entire experiments. After hidden layers parameter tuning weight decay of Adam optimizer was set to 1e-4, which improves validation accuracy ~89.80%.
  - Loss Function:
    - ✓ Categorical cross entropy loss function is used as it is a multiclass classification problem.

- Confusion Matrix:

-	1	2	3	4	5	6	7	8	9	10
1	843	1	24	14	3	0	100	0	15	0
2	2	981	1	10	3	0	2	0	1	0
3	15	0	857	9	64	0	54	0	1	0
4	23	12	10	897	28	0	24	0	6	0
5	0	1	131	34	778	0	53	0	3	0
6	0	0	0	0	0	967	0	15	2	16
7	108	3	95	26	50	1	705	0	12	0
8	0	0	0	0	0	12	0	974	1	13
9	3	0	1	2	5	1	2	3	983	0
10	0	0	0	0	0	7	1	37	0	955

- Accuracy:
  - Accuracy on test data is 89.40%.
  - Accuracy on train data is ~93.10%.
  - Accuracy on validation data is ~89.80%.
- Multilayer Perceptron Loss vs Number of epochs:



ConvNet:

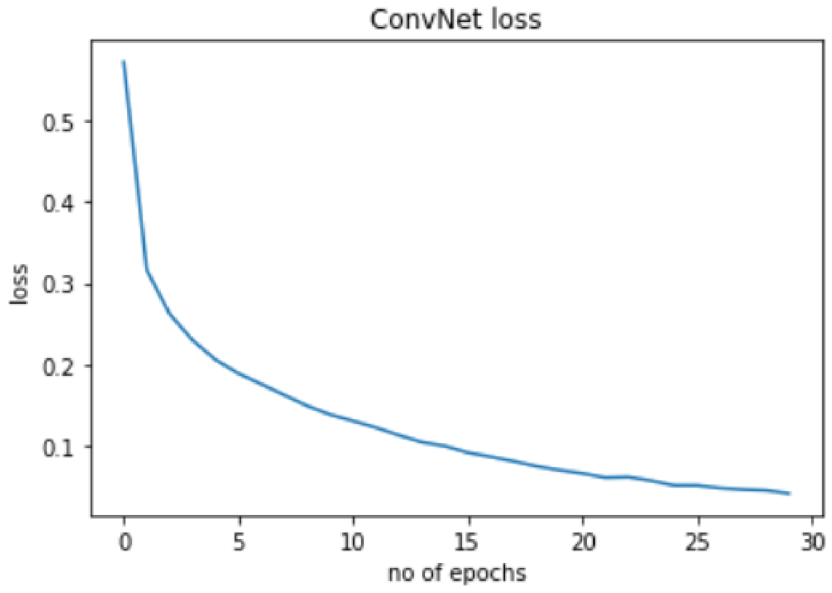
- Architecture:
  - Test data contains some horizontal flips compared with training data for that purpose Keras Image Pre-processing Image Data Generator is used
  - Data generator takes train_images and train_labels as input and generates flipped data which are used for model training.
  - Input and hidden layer:
    - ✓ At first, two set of convolution layer following max pooling layer with kernel of (5,5) and stride of 1 following flatten layer (to make output of convolution layer in the form that can be given in dense layer) and two dense layer of 1024 neurons were checked but training accuracy was ~98% but test accuracy was ~91% only. Here also 1024 neurons because flatten layer output contains 64 * 6 *6 = 2304 so less than that value initialized with ~1000 neurons.
    - ✓ After that kernel size of (3,3) was taken to improve test accuracy as less kernel size consider small details of image (test accuracy ~91.5%). Here simultaneously, number of epochs were reduced to optimum value (50 to 30) as well as batch size from 500 to 300.
    - ✓ After that to improve representation of image data without quickly losing spatial information in place of conv layer followed by max pooling, two conv layers followed by max pooling was used, which improves test accuracy around (~92.80%).
    - ✓ After that as model was overfitting the data (train accuracy ~ 99%), first dropout of 20% (in input as well as in one of the dense layer) were checked and then number of neurons of two dense layers were reduced to half (512 neurons per layer) which improves test accuracy 93.72%.
  - Output Layer:
    - ✓ Output layer consists of 10 neurons because output of it is probability of ten classes so to generate 10-dimensions (10 probability), ten neurons are used.
  - Optimizer:
    - ✓ Adam optimizer was used for optimization during entire experiments.
  - Loss Function:
    - ✓ Categorical cross entropy loss function is used as it is a multiclass classification problem.

- Accuracy:
  - Test data accuracy is 93.72%.
  - Train data accuracy is ~97%.

- Confusion Matrix:

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	850	0	17	13	1	0	113	0	6	0
2	1	989	0	6	2	0	0	0	2	0
3	19	0	917	9	26	0	29	0	0	0
4	5	2	10	943	19	0	19	0	2	0
5	1	0	22	18	931	0	27	0	1	0
6	0	0	0	0	0	985	0	10	0	5
7	64	0	44	19	56	0	814	0	3	0
8	0	0	0	0	0	2	0	983	0	15
9	0	0	2	3	0	0	2	1	992	0
10	0	0	0	0	0	4	0	27	1	968

- CNN Loss vs Number of epochs:


ConvNet loss
loss
no of epochs