

Deep Learning

Programming Assignment 2

April 7, 2019

Group Members	
Name	Roll No.
Akul Gupta	B16006
Arpit Batra	B16047
Ayush Meghwani	B16127

Faculty Mentor - Prof. Aditya Nigam.

Contents

1	On given assignment's network	3
1.1	MNIST dataset	3
1.1.1	Confusion Matrix	3
1.1.2	Accuracy	3
1.2	Line Dataset	3
1.2.1	Confusion Matrix	3
1.2.2	Accuracy	4
2	On our network	4
2.1	Network Structure (same for both lines and MNIST datasets)	4
2.2	MNIST dataset	5
2.2.1	Confusion Matrix	5
2.2.2	Accuracy	5
2.3	Line Dataset	5
2.3.1	Confusion Matrix	5
2.3.2	Accuracy	6
2.3.3	Inferences	6

1 On given assignment's network

1.1 MNIST dataset

1.1.1 Confusion Matrix

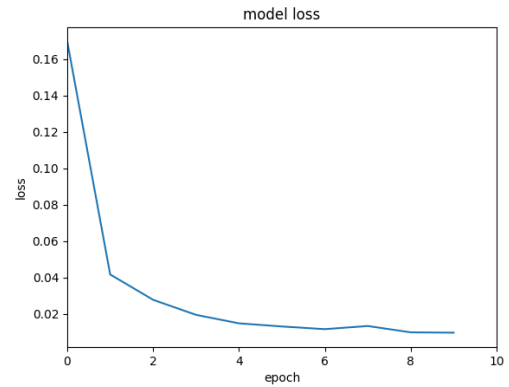
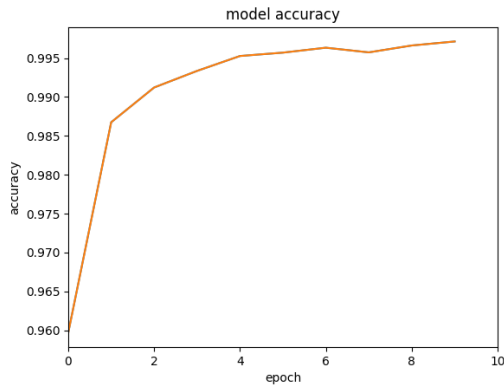
976	0	1	0	0	1	1	1	0	0
0	1133	2	0	0	0	0	0	0	0
1	4	1023	0	0	0	0	4	0	0
0	0	3	992	0	11	0	1	0	3
0	1	1	0	976	0	3	0	0	1
2	0	0	0	0	889	1	0	0	0
8	3	0	0	7	4	936	0	0	0
0	2	4	3	0	0	0	1018	0	1
6	2	16	1	7	10	3	3	916	10
2	5	2	1	12	10	0	3	0	974

1.1.2 Accuracy

	C ₀	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇	C ₈	C ₉
precision	0.980	0.985	0.972	0.994	0.974	0.961	0.991	0.988	1.0	0.984
recall	0.995	0.998	0.991	0.982	0.993	0.996	0.977	0.990	0.940	0.965
f_score	0.988	0.991	0.981	0.988	0.983	0.978	0.984	0.989	0.969	0.974

Accuracy: 98.96%

Loss: 0.041458



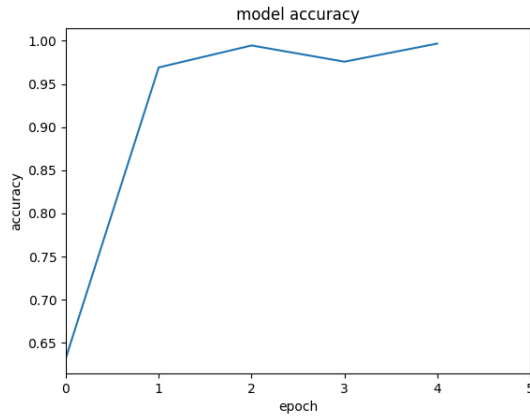
1.2 Line Dataset

1.2.1 Confusion Matrix

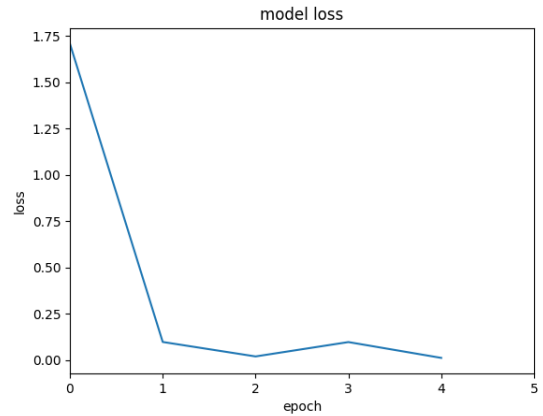
As the size of confusion matrix is 96x96, its not possible to show it here.

1.2.2 Accuracy

Accuracy: 99.84%



Loss: 0.006556



2 On our network

2.1 Network Structure (same for both lines and MNIST datasets)

1. 3x3 Convolutional Layer with 32 filters and stride of 1.
2. ReLU Activation Layer.
3. 3x3 Convolutional Layer with 64 filters and stride of 1.
4. ReLU Activation Layer.
5. 2x2 Max Pooling layer with a stride of 1
5. fully connected layer with 128 output units.
6. ReLU Activation Layer.
7. Droupout layer of 0.5
- 8 Dense layer with number of classes as output (10 for mnist and 96 for lines dataset) with sigmoid activation function

2.2 MNIST dataset

2.2.1 Confusion Matrix

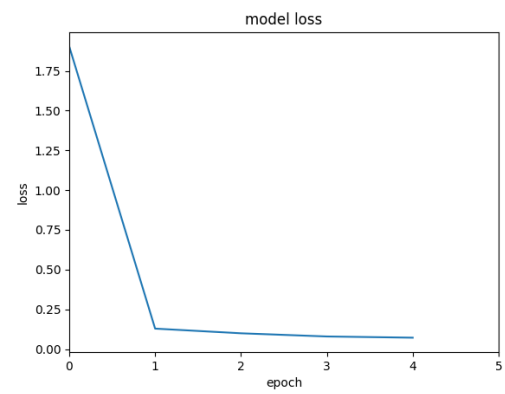
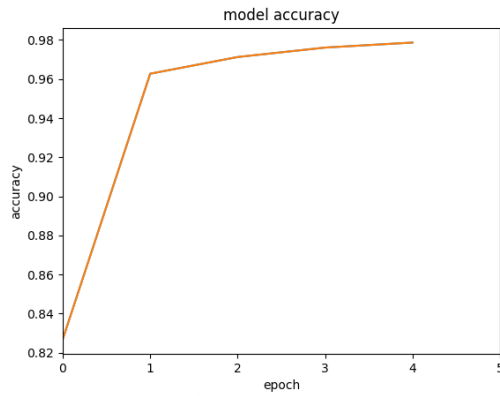
$$\begin{bmatrix} 972 & 1 & 1 & 0 & 1 & 0 & 2 & 1 & 1 & 1 \\ 0 & 1132 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 3 & 1018 & 0 & 3 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 1 & 1004 & 0 & 2 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 976 & 0 & 1 & 0 & 0 & 5 \\ 2 & 0 & 0 & 6 & 0 & 880 & 3 & 0 & 0 & 1 \\ 3 & 4 & 0 & 0 & 1 & 2 & 946 & 0 & 2 & 0 \\ 0 & 2 & 8 & 1 & 0 & 0 & 0 & 1013 & 1 & 3 \\ 9 & 1 & 3 & 3 & 3 & 1 & 1 & 1 & 946 & 6 \\ 2 & 6 & 0 & 2 & 6 & 3 & 0 & 5 & 0 & 985 \end{bmatrix}$$

2.2.2 Accuracy

	C ₀	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇	C ₈	C ₉
precision	0.981	0.985	0.986	0.987	0.985	0.991	0.991	0.986	0.994	0.983
recall	0.991	0.997	0.986	0.994	0.993	0.986	0.987	0.985	0.971	0.976
f_score	0.986	0.991	0.986	0.990	0.989	0.988	0.989	0.985	0.983	0.979

Accuracy: 98.72%

Loss: 0.041458



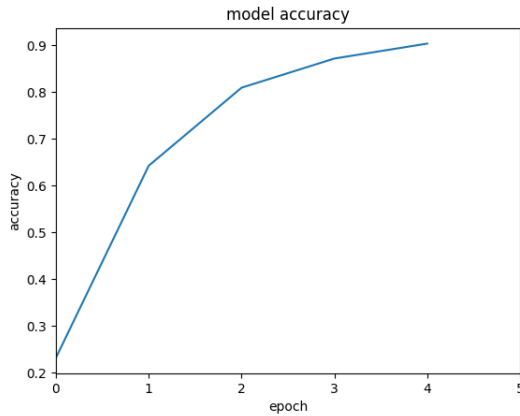
2.3 Line Dataset

2.3.1 Confusion Matrix

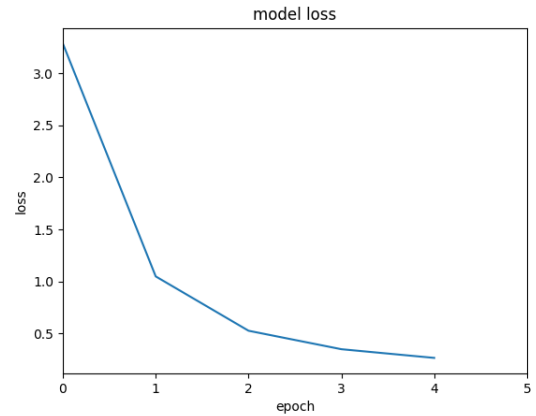
As the size of confusion matrix is 96x96, its not possible to show it here.

2.3.2 Accuracy

Accuracy: 99.86%



Loss: 0.021535



2.3.3 Inferences

1. There was no need to run many epochs for both line and MNIST datasets because while training it was seen that training data accuracy goes to 99 percent or above after 2 or 3 epochs only and loss value is very small.
2. Also, it was tried by our side that if we increase many no. of convolution layers or increase the no. of filters in it, the accuracy decreases. the reason behind is due to overfitting.
3. Also, we choose sigmoid activation instead of RELU or softmax, training accuracy decrease.
4. In case of using sigmoid function, we need to run greater number epochs in order to achieve higher frequency as compare to number of epochs in RELU/softmax function.