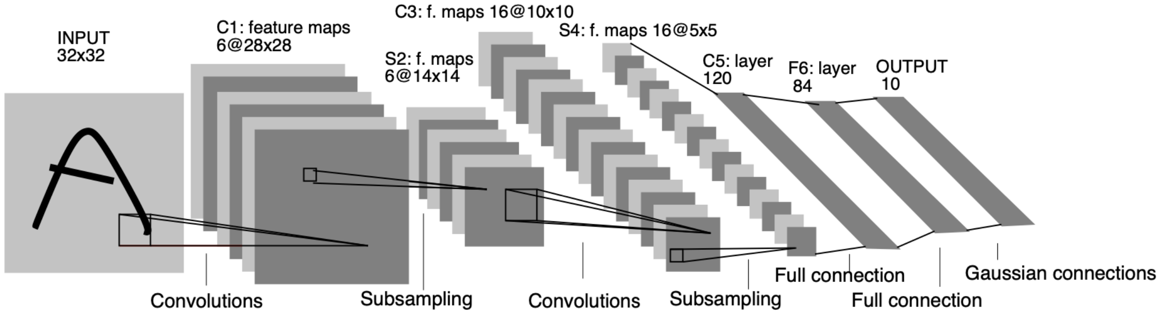
LeNet-5 Report

LeNet-5 is a Convolutional Neural Network which has 7 layers. The input is a 32x32 pixel image. The architecture has been described below:



Layer C1: First convolutional layer with 6 feature maps. Each unit is connected to a 5x5 neighborhood. The size of the feature map is 28x28.

Layer S2: Sub-sampling layer with 6 feature maps of size 14x14. Each unit is connected to a 2x2 neighborhood.

Layer C3: Second convolutional layer with 16 feature maps. Each unit is connected to a 5x5 neighborhood.

Layer S4: Second subsampling layer with 16 feature maps of size 5x5. Each unit is connected to a 2x2 neighborhood.

Layer C5: Third convolutional with 120 feature maps. Each unit is connected to a 5x5 neighborhood.

Layer F6: First fully connected layer with 84 units.

Layer F7: The last layer which has output of 10 units. It has Euclidean Radial Basis Function neurons as activation functions.

Analysis

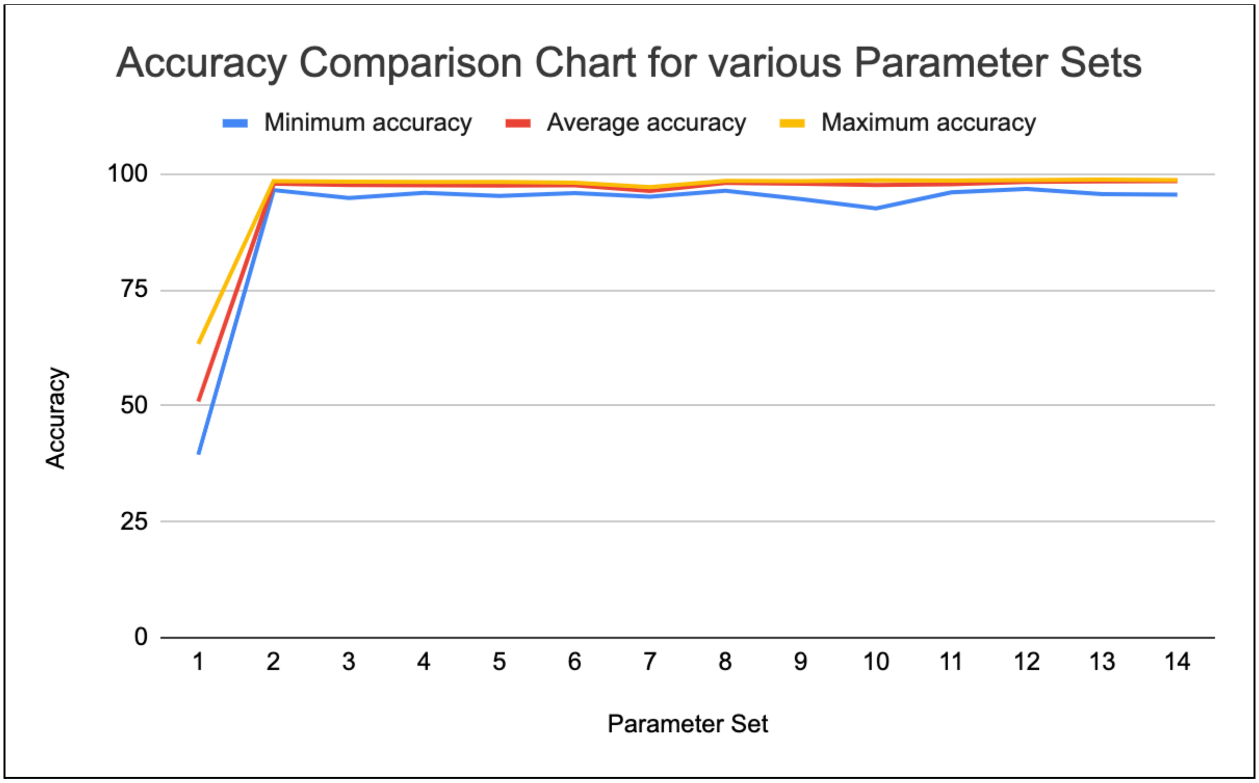
LeNet-5 has been trained and evaluated using 14 different configurations, including training size, test size, epochs, learning rate, and batch size. The minimum, maximum, and average accuracies were determined using these combinations.

- As we increased the number of epochs, we noticed that the accuracy began to improve
- It has also been observed that training and testing size had little impact on the network's accuracy
- With a learning rate of 0.1, we found that accuracy dropped dramatically, with an average accuracy of 50.91 %

Table shown below that shows minimum, maximum and average accuracies with different set of parameters:

Training size	Test size	Number of epochs	Learning rate	Batch size	Minimum accuracy	Maximum accuracy	Average accuracy
90	10	20	0.1	100	39.45237994	63.35714459	50.91071397
90	10	10	0.001	100	96.52380943	98.45238328	97.88809597
80	20	10	0.001	100	94.82142925	98.30952287	97.58214355
70	30	10	0.001	100	95.9523797	98.27777743	97.56031692
60	40	10	0.001	100	95.23809552	98.30357432	97.50416696
50	50	10	0.001	100	95.84761858	98.10476303	97.527619
90	10	20	0.01	100	95.09524107	97.16666937	96.33928597
90	10	20	0.001	100	96.38095498	98.52380753	98.04285645
90	10	20	0.001	200	94.59523559	98.45238328	97.87023813
90	10	20	0.001	500	92.59523749	98.61904979	97.60714352
90	10	20	0.001	32	96.04762197	98.54761958	97.75238186
90	10	300	0.001	40	96.78571224	98.64285588	98.26412698
90	10	100	0.0005	64	95.66666484	98.80952239	98.37214267
90	10	500	0.001	256	95.52381039	98.69047403	98.42399971

Comparison Chart



Conclusion

Overall, the network achieved best results with minimum accuracy of 96.78 % when it was trained on 300 epochs, a maximum accuracy of 98.80 % with a learning rate of 0.001 and 100 epochs, and an average accuracy of 98.42 % with 500 epochs and 256 batch size, which is quite impressive. The learning rate, on the other hand, has been a crucial factor in changing the accuracy and achieving the desired results.

References

https://www.geeksforgeeks.org/countplot-using-seaborn-in-python/

https://pytorch.org/tutorials/beginner/blitz/neural_networks_tutorial.html

https://www.codestudyblog.com/cs2201py/40119135921.html

https://pytorch.org/docs/stable/generated/torch.flatten.html

https://towardsdatascience.com/implementing-yann-lecuns-lenet-5-in-pytorch-5e05a0911320

https://www.kaggle.com/code/alekseysergeev/pytorch-cnn-lenet5-russian-version/notebook

http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf