CS5621 Machine Learning – Assignment

Git Repository link : [AmilaFernando/MNIST-digit-classification (github.com)]

1.
    a.
        i. Model Explanation – I have used a keras **Sequential model**. First layer is a Conv2D layer with 32 nodes and kernel size (3, 3) with activation function "**relu**". Second layer is a **max pooling** layer with Kernel size of (2, 2). Again, third layer is a Convolution layer with 64 nodes and (3, 3) kernel size and "**relu**" activation function. After that there is a **max pooling** layer of size (2,2) . Then the output is flattened and feed to the final layer which uses "**softmax**" activation function.
        ii. ReLu activation function is used for first two convolution layers because it is proven to work well with neural networks. Softmax is used as the activation function for the final layer. Then we get a probability array which summed up to 1 with 10 outputs. We can then take the prediction which has highest probability.
        iii. This dataset contains around 60000 train data rows and 10000 test data rows. Each image is 28*28 pixels in size. No color data present.
    b.

Epoch 1/15 - accuracy: 0.8171

Epoch 2/15 - accuracy: 0.9739

Epoch 3/15 - accuracy: 0.9815

Epoch 4/15 - accuracy: 0.9852

Epoch 5/15 - accuracy: 0.9871

Epoch 6/15 - accuracy: 0.9890

Epoch 7/15 - accuracy: 0.9906

Epoch 8/15 - accuracy: 0.9917

Epoch 9/15 - accuracy: 0.9934

Epoch 10/15 - accuracy: 0.9945

Epoch 11/15 - accuracy: 0.9945

Epoch 12/15 - accuracy: 0.9947

Epoch 13/15 - accuracy: 0.9957

Epoch 14/15 - accuracy: 0.9969

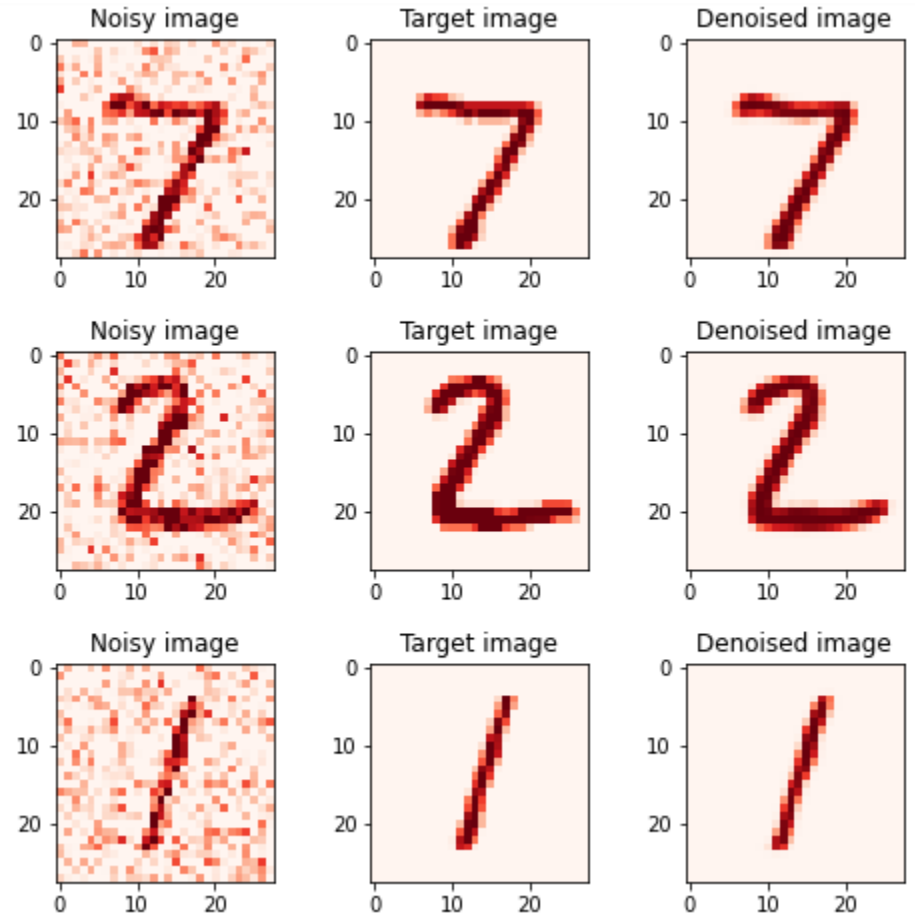
Epoch 15/15 - accuracy: 0.9966

2.

Noise Factor	Accuracy
0.25	0.9857000112533569
0.40	0.9739000201225281
0.50	0.96670001745224

3. Added 0.25 noise factor and modified the model several times and tested for accuracy. It is notices that when the number of layers grows, the accuracy will decrease. This happens for both convolution layers as well as Dense layers. The accuracy was good when there are two convolutional layers with 64 nodes each and one dense layer. Which means noise will not be detected as a feature since number of layers is not sufficient to do that. But a smaller number of layers means a smaller number of important features represent in the model. So, there should be another mechanism.

   Next, I used autoencoders as suggested by Keras blog [1] post to denoise the images. When the new model is trained and evaluated, the loss is 0.07263. Below image is a visualization of noisy image and prediction.



References

1. Keras blog: Building Autoencoders in Keras