STAT 6685 HWK5

Cody Grogan
A02313514

November 16, 2023

Problem 1

1)

PCA is a mathematical technique for capturing the directions of the highest variance in an input. This then allows it to reduce the dimension of the input. An Auto encoder is trying minimize the representation of the input while still being able to reconstruct the input. PCA and Auto encoders are similar because they have the same objective which is to preserve the most data possible in a lower dimensional representation.

2)

The differences between a linear and convolutional auto encoder is exactly what you'd think. Convolutional autoencoders use convolutional layers to reduce the dimension which takes advantage of the locality of information in the image. Where as a linear auto encoder uses only fully connected layers to reduce the dimension of the input not accounting for locality.

3)

a)

Where one is trying to predict future values for a stock given an input window of a fixed size.

b)

An example of this is music generation where the model is given a single note and the RNN needs to output multiple notes or a song.

c)

An example of this would be predicting the tone of a sentence (Whether it is negative positive or indifferent) where inputing the entire sentence is important in determining the tone.

d)

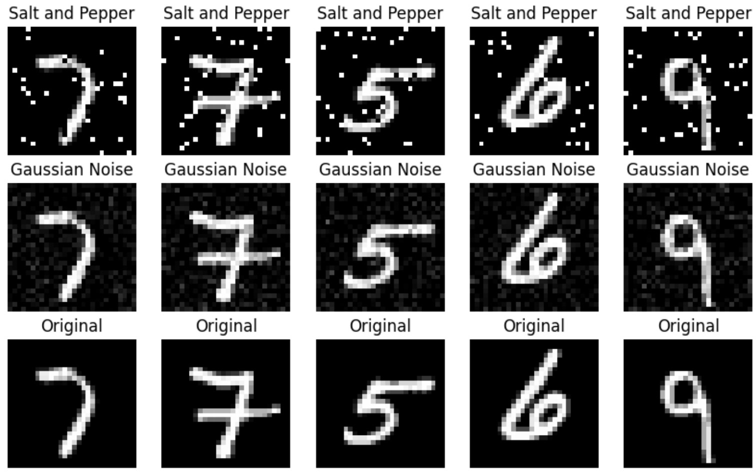
An example of this may be a text summarizer where the input is a long section of text and the output needs to be smaller.

4)

For any task where you do not have information about the future. As in stock prediction one does not have information about the future and if they did they wouldn't need a NN.

Problem 2

1)



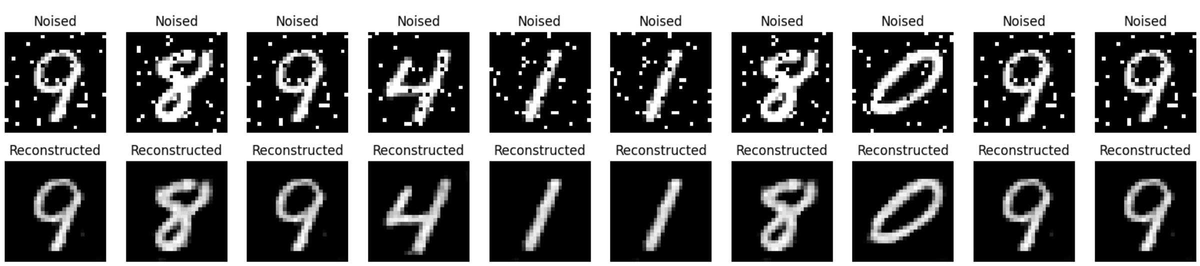
3)

For this section I used MSE loss, Adam optimizer (lr=0.001), Batch size of 128, and 40 Epochs

	Training Loss	Validation Loss
Salt and Pepper	0.0039	0.0040
Gaussian Noise	0.0037	0.0037

Table 1: Table of Training and validation losses for both networks

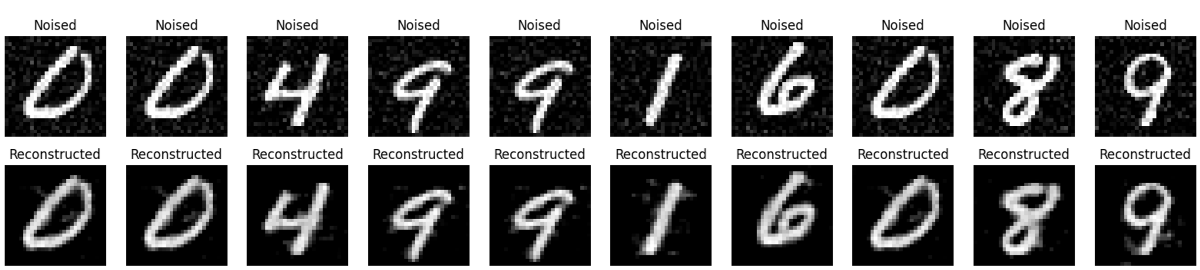
Salt and Pepper Images



With a test loss of 0.004

The figure above shows the autoencoder can accurately denoise the images with rather good accuracy. However the locality of how the network is determining the true value for the pixels is apparent in the small artifact on the 3rd 9. There is a light spot in the background because the noise had salted pixels in a localized region.

Gaussian Noise Images



With a test loss of 0.0064

The figure above shows the autoencoder can denoise the data but not as well as salt and pepper noise. This is likely because the gaussian noise can output values that wouldn't be out of place. This is obvious in the picture of the 9 where the tail has an extra swoop to it.

Problem 3

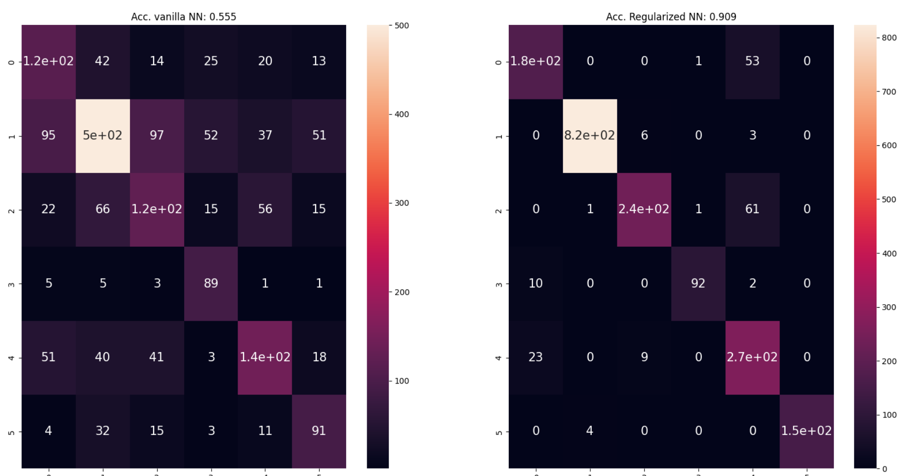
3)

Imputer type	MSE Loss
Simple Mean	0.0399
Simple Most Freq	0.0662
KNN	0.0179
Auto Encoder	0.0143

Table 2: MSE for imputers

The table above shows that the simple imputation techniques are not match for the more complex ones. However the KNN and Autoencoder take significantly more time to compute. Of the complex methods the Autoencoder performed the best with KNN as a close second.

Problem 4



The confusion matrices show there is a significant performance increase when using geometry regularization on the unlabelled portion of the data. This is evident from the high numbers on the diagonal (correct predictions) for the geometry regularized model. This also shows multitask learning can allow a network to learn things about the data without labels.