EECE6036 HW5

Jack Fliegel

# Problem 1

**System Specification**

As specified in the Homework pdf, I used the same network as in Problem 1 of HW4 but applied pretrained weights (labeled FinalInputs.txt in /HW5\_Datafiles) to the input to hidden layer weights. Case I did not apply training to these weights and Case II did apply training to the weights.

**Results**

A graph with a number of error fractions

Description automatically generated

Figure : Error Fractions vs Epoch HW5

A graph with a line graph

Description automatically generated with medium confidence

Figure : Error Fractions vs Epoch HW4

A chart of training and prediction

Description automatically generated with medium confidenceA chart of training set case ii

Description automatically generated

Figure : Confusion Matrix Training Set (Cases I (a) and II (b))

A chart with a yellow line

Description automatically generatedA chart with a yellow line

Description automatically generated

Figure : Confusion Matrix Test Set (Cases I (a) and II (b))

**Analysis**

*Q1: Did initializing the hidden weights from the autoencoder make training go faster in this this homework compared to that in HW 4 (for both Case I and Case II)?*

Initializing the hidden weights made the training slower and fail to classify for Case I and Case II compared to HW4. As we can see in figure 1 the model has a higher error fraction within 5 epochs compared to HW4’s model where the weights were set at random (not pretrained).

With these results we can assume that the model failed to classify for both cases I and II. I do not believe that the autoencoder found relevant features that could be applied to the model.

*Q2: Did training both layers (Case II) substantially improve performance over training only the output*

*layer (Case I)?*

There was not any significant changes from Case I vs Case II when comparing performance of each. Both seemed to have poor performance and failed to classify in different ways as seen in figures 3 and 4. The accuracy of both cases stayed around 15% which makes sense because the model was failing and only predicting one output for every input.

# Problem 2

**System Description**

**Results**

A group of squares with different shades of orange and black

Description automatically generated

Figure : SOFM heatmap of each digit 0-9

A number in a row

Description automatically generated with medium confidence

Figure : features found for each neuron

**Discussion**

Figure 5 shows that for each digits class there are clear and distinct areas of classification. Each map is scaled differently but we can still see patterns arise and some neurons for certain digit classes ‘win’ more than others.

# Problem 3

**System Description**

**Results**

**Discussion**