Network Architecture

- 3 linear layers with ReLU activation on the first layer and Tanh activation on the last layer

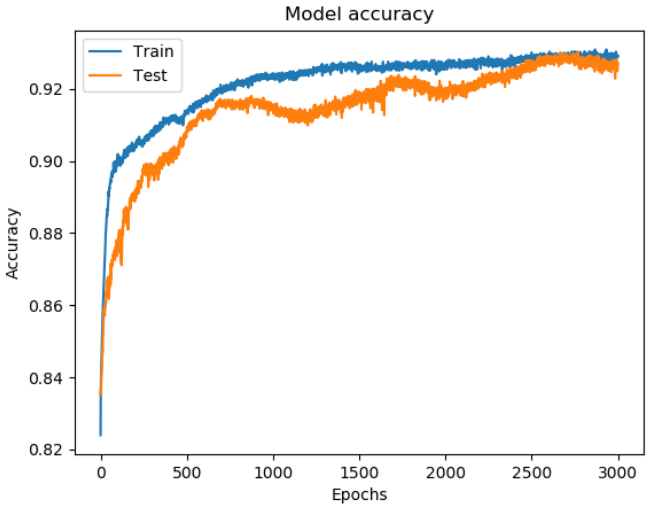

Figure 1: Accuracy over number of epochs.

Age	EducationNum	Hours-per-week	Label	Prediction
47	9	40	1	0
45	13	60	1	0
35	12	44	0	1

Table 1: Misclassified examples.

Performance The model is trained for 3000 epochs. The training accuracy reached 92.9% and the test accuracy reached 92.7% (Figure 1). The mean for both bias vectors and weight matrices are around 0, but the values themselves vary a lot. The first parameter seems to have very high values or very low values (Figure 2). The model trained on the GPU for about 1 hour.

Table 1 shows examples that were misclassified. Only 3 features are shown. Age refers to the age of the person, EducationNum refers to the number of years that the person was education, and Hours-per-week refers to the number of hours that the person has worked. The network has classified two people who are in their 40s as making less than 50K, but they actually do make more than 50K. The network also misclassified someone in their 30s as making more than 50K, but they actually make less than 50K.

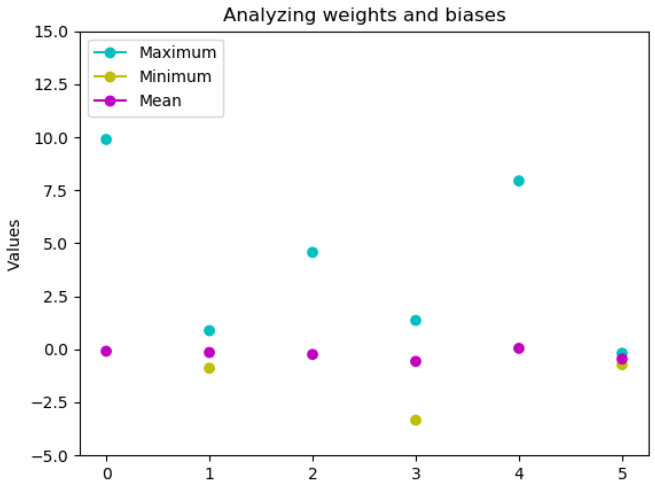

Figure 2: Final weights and biases of trained model. There are three colored points along each x-coordinate because these are the values associated with a parameter: maximum, minimum, and mean. These values indicate the largest, smallest, and average of values in the weight matrix/bias vector.

Explanation For this problem, I first tried adding more layers and making the network deeper. However, accuracy could not increase as the network trained for an extended period of time. Then, I tried the usual techniques like batch normalization and dropout. Those methods seem to make accuracy even lower. So, in the end, I just trained a simple feedforward network for several epochs. This seemed to be most effective at increasing training and test accuracy. I think this may be because the data contains many categorical variables. Unlike images, batch normalization and dropout may not be effective. Also, adding too much nonlinearity may not be an advantage. The mapping between inputs and labels may be a very simple function. So, adding nonlinearity may make the mapping between inputs and predictions too complicated.

For the last layer, I added a Tanh activation function rather than a ReLU function. This seemed to improve accuracy by 1 to 2%. Then, I used a cross entropy loss. This loss could also provide a probability of the person making more than 50K. If loss is close to 1, then the person is likely to make more than 50K. If the lose is close to 0, then the person is not likely to make more than 50K. This is why I chose to use cross entropy for this model.