Adult Earning Dataset: Phase 2

February 2022

Contents

1 Introduction

Artificial Intelligence is a powerful tool to use for predictions. In a world were personal data is has value, it can be useful to determine the wealth of a person. This concept has potential for many applications from determining appropriate demographics for advertising to offering a fair starting salary to a college graduate. In this project, we will build a neural network for binary classification to determine whether a person makes over $50k a year. We will test a variety of structures of the neural network in order to determine which one will be appropriate for our task. We will accomplish this task using only 13 key features about a person. We will then build a more efficient neural network by determining which features negatively impact our accuracy. We will remove any unwanted features from our final model. This tool of artificial Intelligence is only a powerful tool when trained with appropriate data and in a appropriate model. By determining the best model for our data and removing unnecessary features, we will aim to be as accurate as possible with our final model.

Task: Salary Prediction
The prediction task is to determine whether a person makes over 50K a year.

2 About the Project Phases

I decided to switch datasets during phase II of the project. My original dataset was dealing with poker hands and was not happy with the heavily unbalanced results of the "poker hands" (due to basic statistics). I decided to switch to a dataset that dealt with determining the scores of math tests dependent on test prep, race, gender, parents education, etc. I had issues while running the neural network and would constantly get accuracy levels of around 1.3%. I tried manipulating the neural network (adding/removing layers/nodes, epochs, and changing the batch size). I spent too much time trying to fix my issue before I looked into the data and decided that it was going to require more manipulation before it would be ready for the neural network. I decided to switch my dataset again to this adult earnings dataset. This dataset is far more easier to work with since I am still a novice in this field of AI. Much of Phase I had to be reworked to be applied to this new dataset and to better understand the data.

Link to OverLeaf Project:
`https://www.overleaf.com/read/hkttfbbdhtzp`

3 The Dataset and Its Source

Each record is a person's atrribute listed below.

3.1 Attributes

1. Age: [continuous]
2. Workclass: Private, Selfempnotinc, Selfempinc, Federalgov, Localgov, Stategov, Withoutpay, Neverworked
3. Final Weight (fnlwgt): [continuous]
4. Education: Bachelors, Somecollege, 11th, HSgrad, Profschool, Assocacdm, Assoc-voc, 9th, 7th8th, 12th, Masters, 1st4th, 10th, Doctorate, 5th6th, Preschool.
5. Educationnum: [continuous]
6. Maritalstatus: Married-civ-spouse, Divorced, Nevermarried, Separated, Widowed, Married-spouseabsent, MarriedAFspouse
7. Relationship: Wife, Ownchild, Husband, Not-in-family, Otherrelative, Unmarried
8. Race: White, AsianPacIslander, Amer-Indian-Eskimo, Other, Black
9. Sex: Female, Male
10. Capital-gain: [continuous]
11. Capital-loss: [continuous]
12. Hours-per-week: [continuous]
13. Native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(GuamUSVIetc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, ElSalvador, Trinadad and Tobago, Peru, Hong, Holand-Netherlands
14. Earnings: greater than 50K, lesser than or equal to 50K

3.2 About

The dataset was sourced from UCI Machine Learning Repository. It was created by Ronny Kohavi and Barry Becker. Extraction was done by Barry Becker from the 1994 Census database.
Source: http://archive.ics.uci.edu/ml/datasets/Adult

Splitting the Data

Before splitting the data, the data was shuffled in order remove any predetermined "traits" in the order of the original dataset. The data was then split, since the dataset is large (n = 32,561), 30% of the data was used for training (n = 9768) the model and 70% of the data was used for validation (n= 22,793). To ensure the normalization of the training data didn't disturb the normalization of the validation data (and vice-versa), the data was normalized after the split.

4 Normalizations

In order to normalize the data, each of the input variables will be normalized based on their range, known as Min-Max Normalization. For this method, the following equation can be employed on each input variable:

$$\acute{x} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

This method will bring all data into the range [0, 1], so that the coefficients are representative of the potential of each input variable to change the output. If this method is insufficient, then traditional standardization can be used instead:

$$\acute{x} = \frac{x - \bar{x}}{\partial}$$

4.1 Normalized Histograms

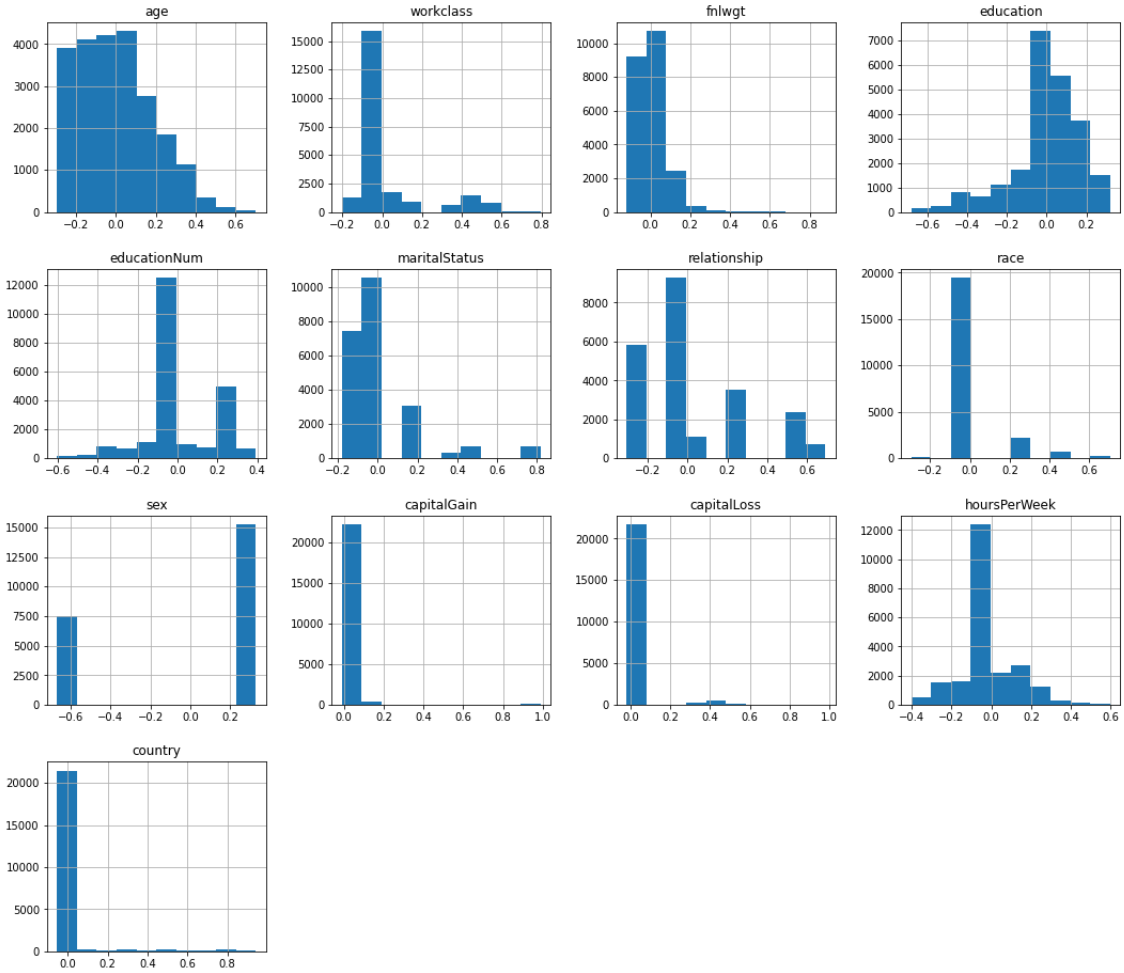

Figure 1: Histograms of Normalized Dataset

5 Options of Model Selections

All models were ran with 100 epochs and a batch size of 64. Since I am still a novice at this, I took the opportunity to run multiple alternative models to learn how each one effects the learning rates. Observing the Model Accuracy graph, it should be noted that the closer the line is to the top left hand corner, the more accurate the neural network is. Also it should be noted that in the Model Loss graph, the closer the line is to the lower right hand corner the less loss there is.

5.1 Base Model

This base model has a basic architecture of a single input and a single output layer.

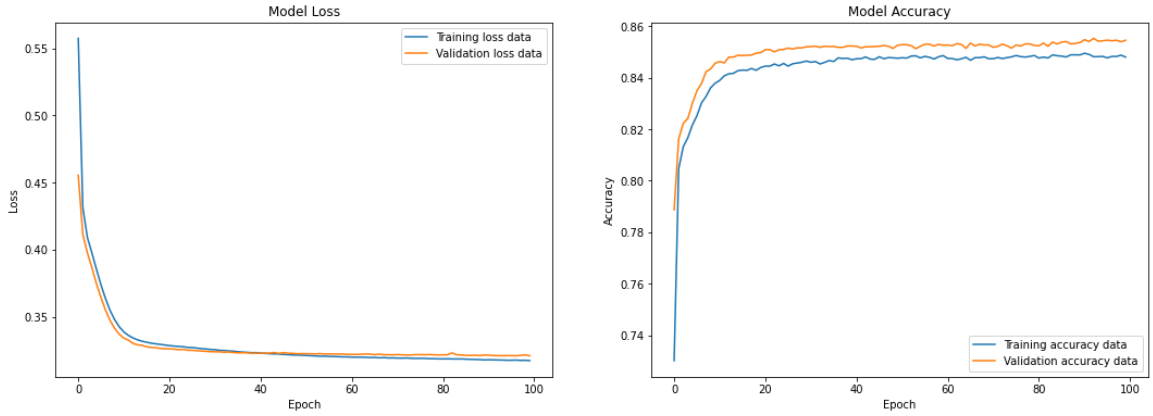

Figure 2: Learning Curve of Base Model

5.2 Model with Middle Layer

This model has an architecture of an input layer, a hidden layer, and an output layer.

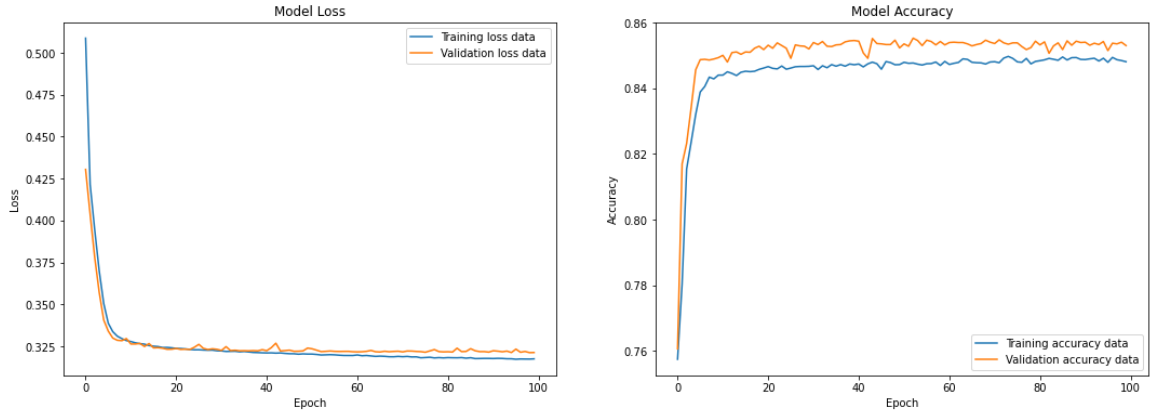

Figure 3: Learning Curve of Model with Hidden Layer

5.3 Model with 4 Layers

This model has an architecture of an input layer, two hidden layers, and an output layer.

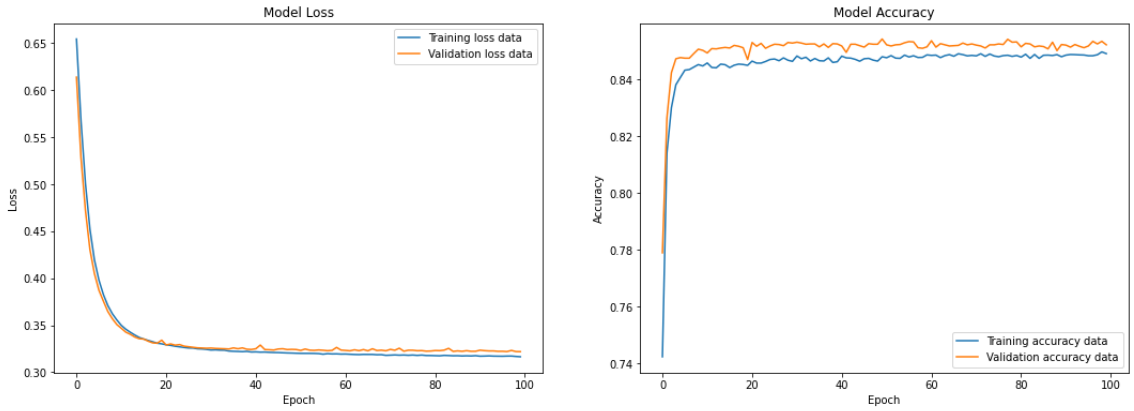

Figure 4: Learning Curve of Model with 2 Hidden Layers

5.4 Model with 6 Layers

This model has an architecture of an input layer, four hidden layers, and an output layer.

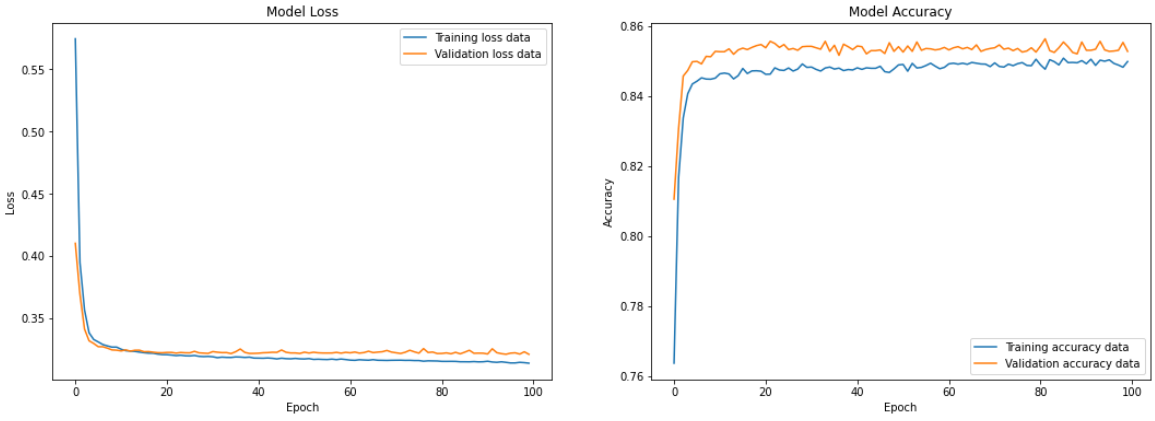

Figure 5: Learning Curve of Model with Hidden Layer

5.5 Model with Overfitting

This model has an architecture of an input layer, a hidden, and an output layer with overfitting by a factor of 10 on the first and hidden layer.

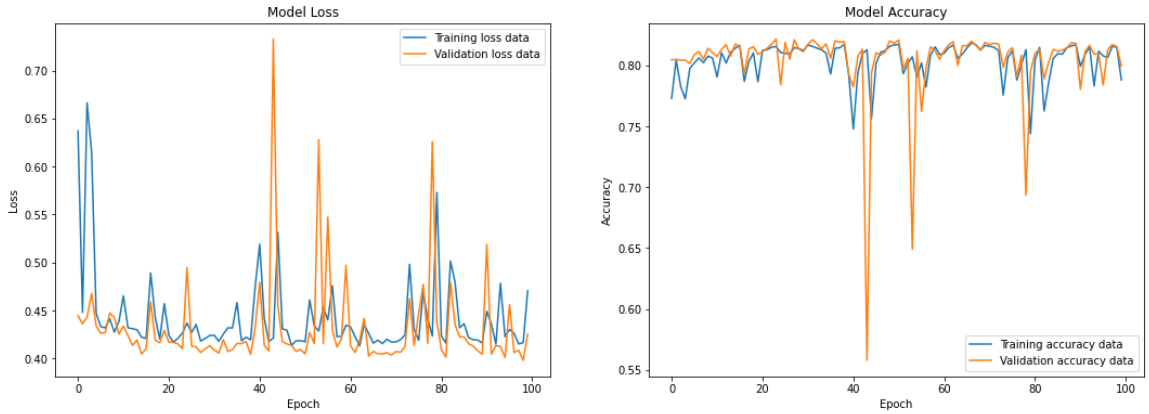

Figure 6: Learning Curve of Model with Overfitting