Heart Disease Prediction Based on Health Records

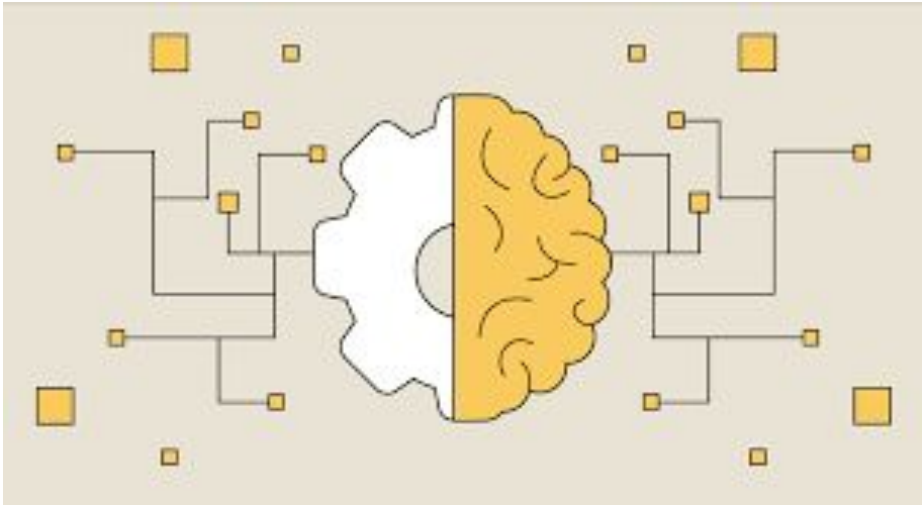
By Aaron Coppeta and Tyler Merrill

Introduction

- Heart disease is the leading cause of death in the US [1]
- 1 in every 5 deaths [1]
- Heart disease cost the US an estimated 239.9 billion dollars each year [1]
- Risk factors [2]
  - High blood pressure
  - high cholesterol
  - Diabetes
  - Smoking
  - Obesity
  - Age
  - Low physical activity

Methods - Dataset

- Used kaggle dataset for cardiovascular disease [3]
  - 70,000 data from patients
- Results gained from medical examination from patients
- Features
  - Ages, height, weight, gender, systolic and diastolic blood pressure, cholesterol, glucose, smoking, alcohol intake, physical activity
  - Target column: presence or absence of cardiovascular disease



Methods - Test Parameters

- Intermediate Linear Layers
  - 0, 1, 2
- Learning rate
  - 1e-1, 1e-3, 1e-6
- Initial Layer Width
  - 10, 128, 512, 1024
- Data Scaling
  - Min-max versus standard deviation scaling
- Training Epochs
  - The best model after the first iterations was analyzed from 20-100 training epochs in increments of 10 epochs.

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.flatten = nn.Flatten()
        self.l1 = nn.Linear(12, 10)
        self.l2 = nn.Linear(10, 8)
        self.l3 = nn.Linear(8, 5)
        self.l4 = nn.Linear(5, 2)
```

0 Intermediate same-in, same-out layers

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.flatten = nn.Flatten()
        self.l1 = nn.Linear(12, 10)
        self.l2 = nn.Linear(10, 8)
        self.l3 = nn.Linear(8, 8)
        self.l4 = nn.Linear(8, 5)
        self.l5 = nn.Linear(5, 2)
```

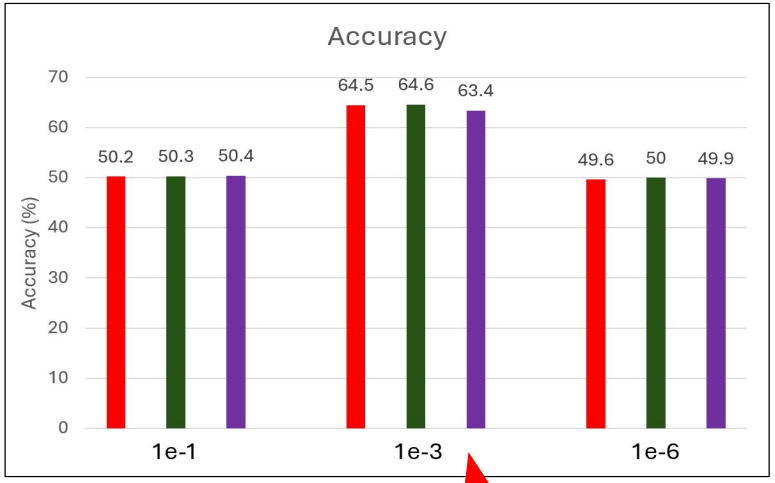
1 Intermediate same-in, same-out layer

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.flatten = nn.Flatten()
        self.l1 = nn.Linear(12, 10)
        self.l2 = nn.Linear(10, 8)
        self.l3 = nn.Linear(8, 8)
        self.l4 = nn.Linear(8, 8)
        self.l5 = nn.Linear(8, 5)
        self.l6 = nn.Linear(5, 2)
```

2 Intermediate same-in, same-out layer

Results

1. # of Layers & 2. Learning Rate



Best Learning Rate

0 Intermediate same-in, same-out layer

1 Intermediate same-in, same-out layers

2 Intermediate same-in, same-out layers

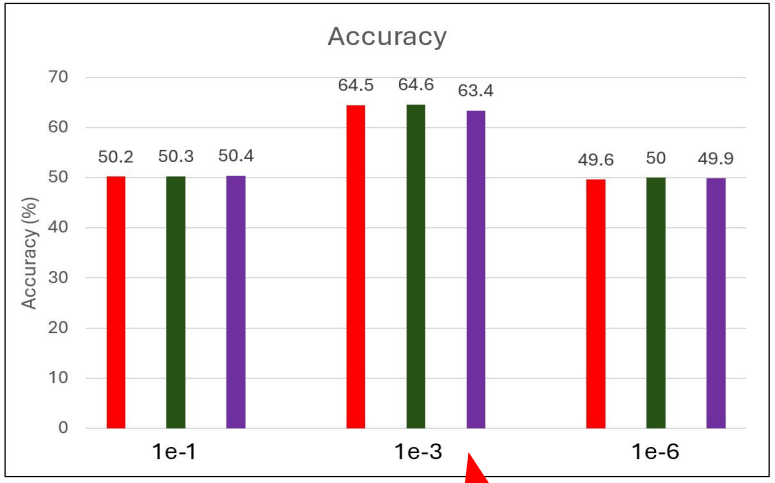
Best Architecture

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.flatten = nn.Flatten()
        self.l1 = nn.Linear(12, 10)
        self.l2 = nn.Linear(10, 8)
        self.l3 = nn.Linear(8, 8)
        self.l4 = nn.Linear(8, 5)
        self.l5 = nn.Linear(5, 2)
```

1 Intermediate same-in, same-out layer

Results

1. # of Layers & 2. Learning Rate



0 Intermediate same-in, same-out layer

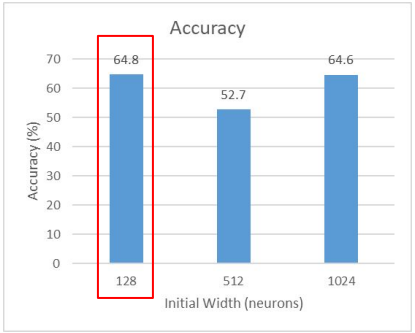
1 Intermediate same-in, same-out layers

2 Intermediate same-in, same-out layers

Best Learning Rate

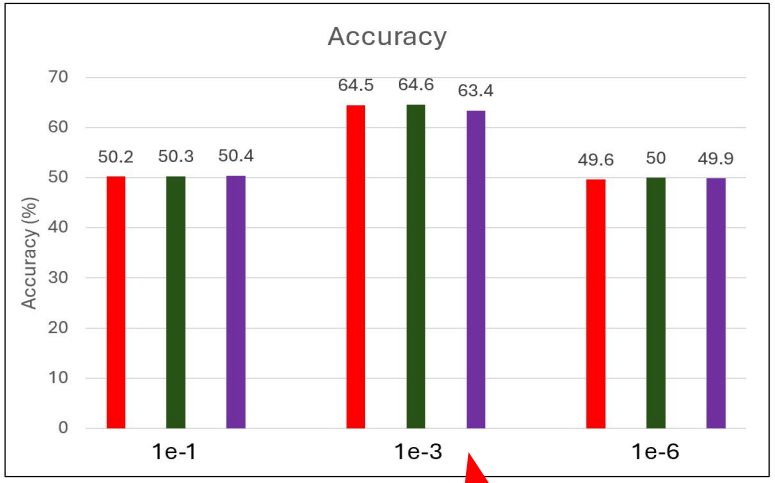
Best Architecture

3. Initial Layer Width



Results

1. # of Layers & 2. Learning Rate



0 Intermediate same-in, same-out layer

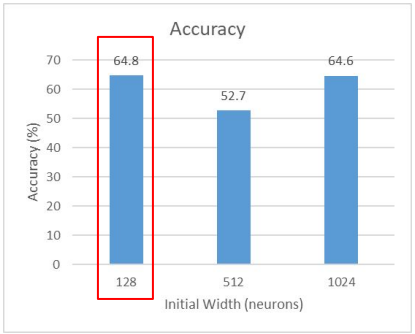
1 Intermediate same-in, same-out layers

2 Intermediate same-in, same-out layers

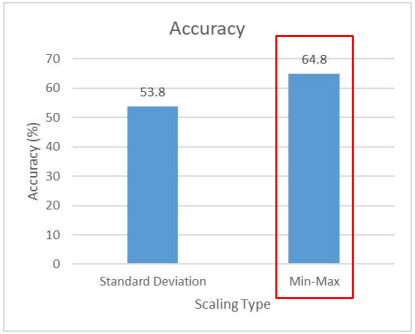
Best Learning Rate

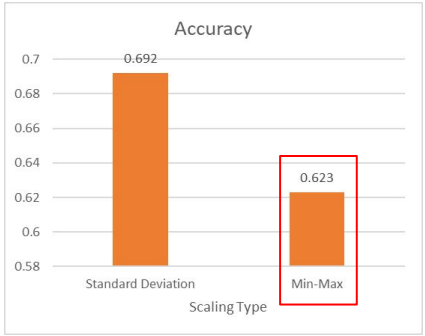
Best Architecture

3. Initial Layer Width



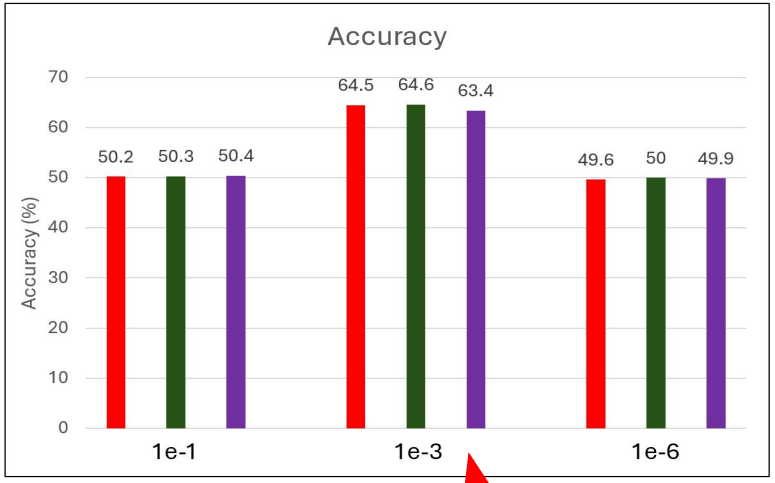
4. Data Scaling Method





Results

1. # of Layers & 2. Learning Rate



0 Intermediate same-in, same-out layer

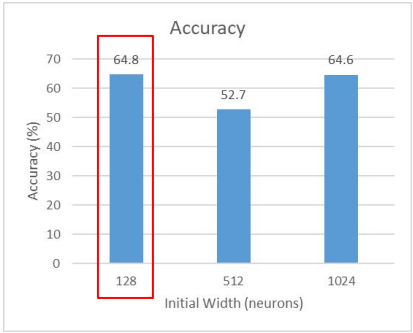
1 Intermediate same-in, same-out layers

2 Intermediate same-in, same-out layers

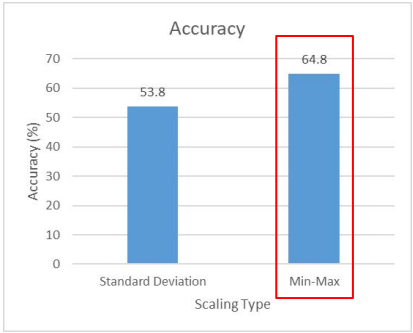
Best Learning Rate

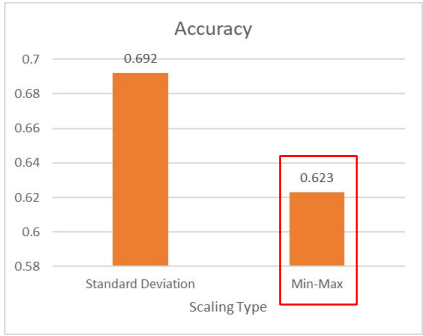
Best Architecture

3. Initial Layer Width

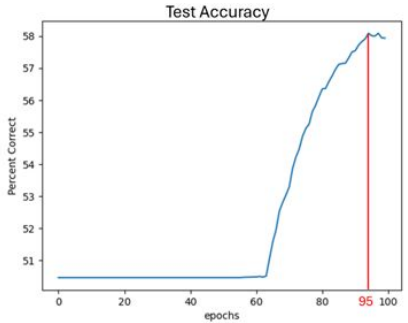


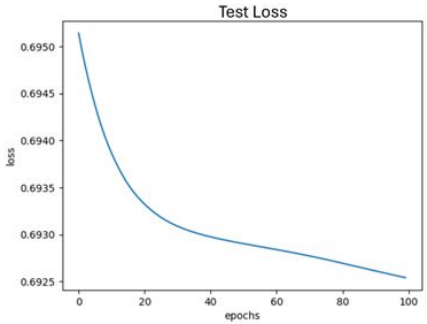
4. Data Scaling Method





5. Training Epochs





Discussion

- Modified learning rate, epochs, layers, width of layers, and scaling method
  - Learning rate: 1e-3
  - Epochs: ~95
  - Layers: 1 intermediate layer
  - Width of layers: 128
  - Scaling methods: min max
- The model has an accuracy of 64.7% and an average loss of 0.624
  - Model is not overfitting
  - Model is not only predicting one value
  - Although model is more accurate than random chance further work is needs to be done to validate and optimize this model
- Can be used to indicate if individuals are at risk for heart disease based off quick and simple measurements
  - Can be used to streamline healthcare process for heart disease

Reference

[1]Centers for Disease Control and Prevention, “Heart Disease Facts,” Centers for Disease Control and Prevention, May 15, 2023. https://www.cdc.gov/heartdisease/facts.htm

[2] CDC, “Heart Disease and Stroke,” www.cdc.gov, Oct. 07, 2020. https://www.cdc.gov/chronicdisease/resources/publications/factsheets/heart-disease-stroke.htm#:~:text=The%20Nation

[3]“Cardiovascular Disease dataset,” www.kaggle.com. https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset?resource=download