October 16, 2020

Activity Recognition in Healthy Older People
Project Phase II

Jordan Lanius
CS 4300
University of Missouri-St. Louis

Abstract

As she has grown older, my grandmother has experienced dizziness and infrequent fainting spells. She has a button that she wears around her neck that she can use to call for help in case of a fall, but it is risky to rely on her ability to press it during an emergency. I am interested in the potential for an artificial intelligence to monitor a person's activity and be able to report any concerning behavior.

1 Dataset

The dataset was obtained by the UCI Machine Learning Repository and is titled "Activity recognition with healthy older people using a batteryless wearable sensor Data Set"[1]. It was provided by Roberto Luis Shinmoto Torres, Damith Ranasinghe, and Renuka Visvanathan from the University of Adelaide. For this project, I am only using the S1 setting that uses four RFID reader antennas. There are sixty participants that took part in this trial, with varying amounts of data collected from each.

1.1 Dataset Description

The data was originally separated into different files for each participant. During the cleaning process, where I merged it from multiple files to a single *.csv file, I included columns for the participant number and their gender. I also adjusted the activity column, condensing it from four different activities into two categories: 'In Bed', expressed as 0, and 'Out of Bed', expressed as 1.

In total, the condensed dataset has 52482 rows and 11 columns. For this project, I will be using 8 of the 11 columns for the input and use the 'Activity' column for the output. The input attributes that will be used are:

- Gender of the Participant
- Acceleration reading in G for frontal axis
- Acceleration reading in G for vertical axis
- Acceleration reading in G for lateral axis
- ID of antenna reading sensor
- Received signal strength indicator (RSSI)
- Phase
- Frequency

1.2 Input Data Visualization

Histograms plot the distribution of each input feature. Figure 1 shows these plots for each input column of the dataset.

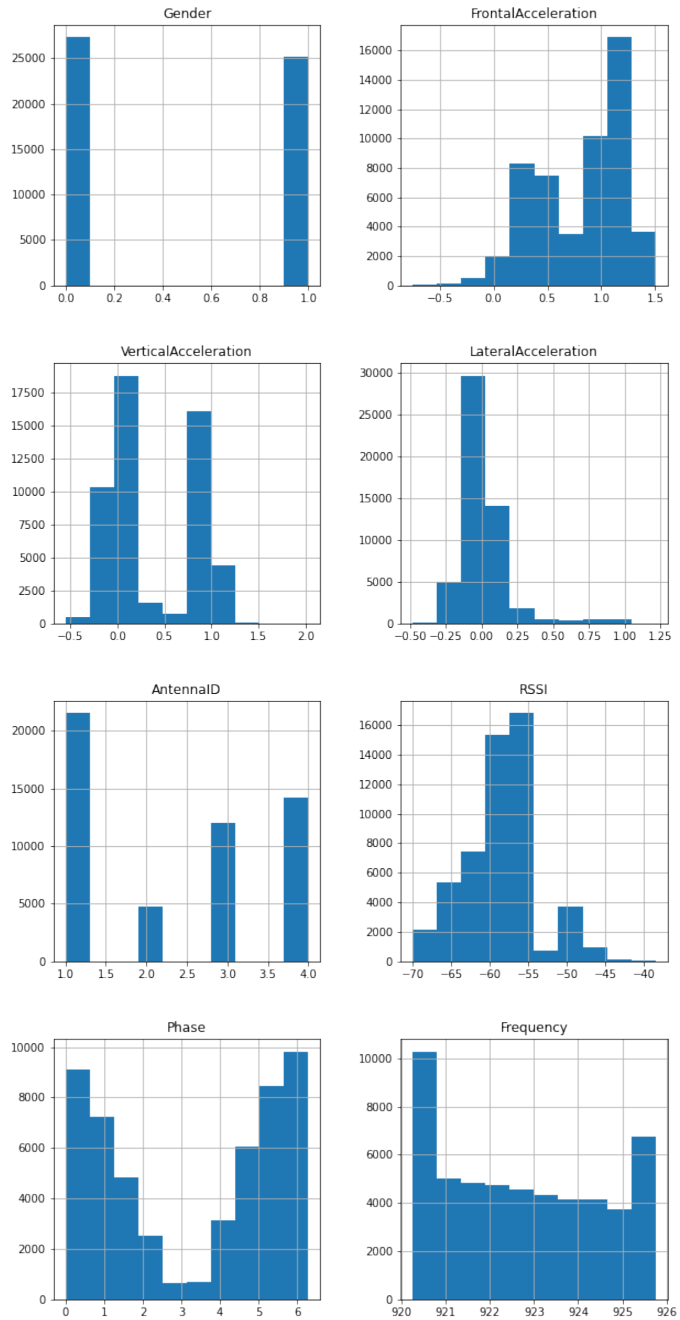


Figure 1: Input Data Distribution Histograms

Table 1 contains the minimum and maximum values of each attribute as well as the calculated mean and standard deviation values that are used for normalization.

Table 1: Input Feature Statistics

	Min	Max	Mean	Std
Gender	0.0	1.0	0.479993	0.499604
Frontal Acceleration	-0.74808	1.5032	0.805042	0.39636
Vertical Acceleration	-0.55349	2.0302	0.377804	0.468899
Lateral Acceleration	-0.48121	1.2178	0.00771	0.180674
Antenna ID	1.0	4.0	2.360752	1.261542
RSSI	-70.0	-38.5	-58.430814	4.61122
Phase	0.0	6.2817	3.275907	2.240341
Frequency	920.25	925.75	922.762261	1.693769

1.3 Output Data Visualization

Figure 2 shows the distribution of the binary output. There is some imbalance between the two. About 87.9% (46145/52482) of the sensor observations are classified as 'In Bed' while only 12.1% (6337/52482) are classified as 'Out of Bed'. While significant, this imbalance isn't extreme and there are still plenty of observations of both classes.

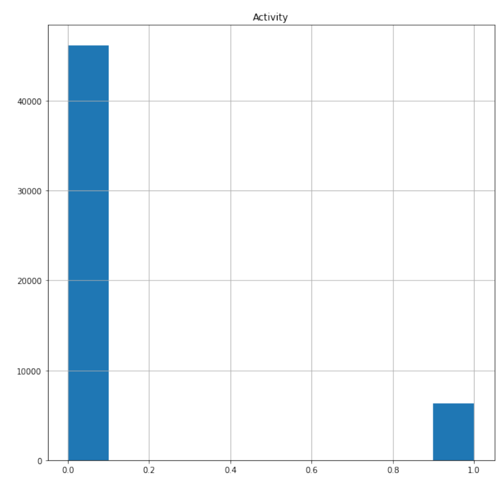


Figure 2: Output Data Distribution Histogram

2 Data Processing

2.1 Data Splitting

The data has been shuffled randomly and separated into two groups. 70% of the dataset has been allocated as training data and the remaining 30% of the dataset has been reserved as validation data.

The validation data will be used to evaluate the different architectures explored in Section 3. A model isn't effective if it is only accurate with the training data.

2.2 Data Normalization

Neural networks work best when data has been distributed uniformly. The best way to achieve this is to apply a normalization technique to the dataset. For this project I have used Standardization which utilizes Equation 1.

$$X = (X - mean)/std \tag{1}$$

3 Modelling

3.1 Sigmoid Activation

The sigmoid activation function is often used to perform logistic regression. This occurs when only one neuron is used in the neural network.

3.1.1 Performance

Table 2, given below, compares the performance statistics for five different architectures that use the sigmoid activation function. The models labeled 'Relu' use the rectified linear activation function on all but the final layer of neurons. The models labeled 'Pure' use the sigmoid activation function on every layer of neurons.

Table 2: Performance comparison for various sigmoid activation architectures

	Accuracy	Precision	Recall	F1 Score	Validation Accuracy
Single-Layer	91.62%	82.50%	39.26%	0.53	91.65%
Relu Double-Layer	97.76%	97.44%	83.70%	0.90	97.85%
Pure Double-Layer	97.15%	99.97%	76.54%	0.87	97.27%
Relu Triple-Layer	98.29%	97.75%	87.95%	0.93	98.15%
Pure Triple-Layer	98.23%	98.82%	86.42%	0.94	98.39%
Relu Quadruple-Layer	98.56%	98.47%	89.49%	0.94	98.39%
Pure Quadruple-Layer	98.34%	98.17%	87.95%	0.93	98.34%

3.1.2 Summary

As seen above, all of the neural networks performed better than a basic logistic regression model. Between the double-layer models, the one that used the linear activation function on every layer performed the best. There is no significant difference between the triple-layer models, although the one that only used the sigmoid activation function performed a bit better with the validation data. Neither of the quadruple-layer models were able to improve the accuracy in any meaningful way and any further growth risks overfitting.

3.2 Linear Activation

The linear activation function is often used to perform linear regression. This occurs when only one neuron is used in the neural network.

3.2.1 Performance

Table 3, given below, compares the performance statistics for five different architectures that use the linear activation function. The models labeled 'Relu' use the rectified linear activation function on all but the final layer of neurons. The models labeled 'Pure' use the linear activation function on every layer of neurons.

Table 3: Performance comparison for various linear activation architectures

	Accuracy	Precision	Recall	F1 Score	Validation Accuracy
Single-Layer	87.94%	90.62%	0.65%	0.01	88.07%
Relu Double-Layer	97.43%	94.91%	83.25%	0.89	97.55%
Pure Double-Layer	87.92%	86.36%	0.43%	0.01	88.05%
Relu Triple-Layer	98.14%	97.92%	86.55%	0.92	98.10%
Pure Triple-Layer	87.97%	93.02%	0.90%	0.02	88.10%

3.2.2 Summary

Overall, using the linear activation function didn't produce any significantly better results than using the sigmoid activation function. The models that performed the best utilized the rectified linear activation function for their input and hidden layers. In fact, using the linear activation function for every layer led to worse performances.

3.3 Output Inclusion

One interesting thing to test is how the neural network responds when given the output as an input option. As seen in Table 4, the performance has maxed out at 100%. It may look great, but this model is unusable because the goal of this project is to develop a neural network that can interpolate from data that is not already classified.

Table 4: Performance when the output is included as input

Accuracy	Precision	Recall	F1 Score	Validation Accuracy
100.00%	100.00%	100.00%	1.00	100.00%

4 Model Evaluation

Going forward, I will use the Pure Triple-Layer Sigmoid Activation model, found in Table 3.1.1, as the Neural Network model.

4.1 Performance Comparison

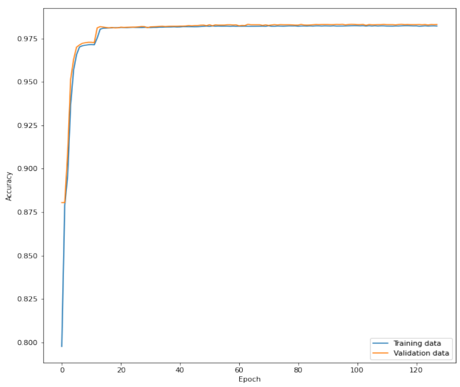
Table 5, given below, compares the performance statistics for the three major architectures explored above. It's purpose is to establish the effectiveness of the selected neural network to both the simple logistic regression and the simple linear regression.

Table 5: Performance comparison for selected architectures

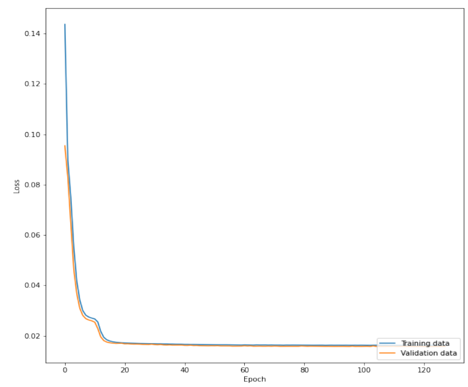
	Accuracy	Precision	Recall	F1 Score	Validation Accuracy
Neural Network	98.23%	98.82%	86.42%	0.94	98.39%
Logistic Regression	91.62%	82.50%	39.26%	0.53	91.65%
Linear Regression	87.94%	90.62%	0.65%	0.01	88.07%

4.2 Learning Curves

Learning curves are a great tool to see how effective a neural network is on data that wasn't used during training. A model is useless if it cannot accurately classify new data.

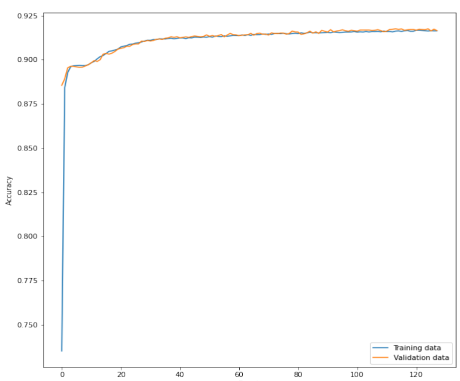


(a) Accuracy vs. Epoch

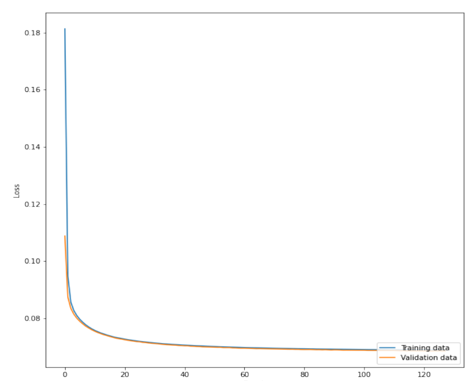


(b) Loss vs. Epoch

Figure 3: Neural Network Learning Curves



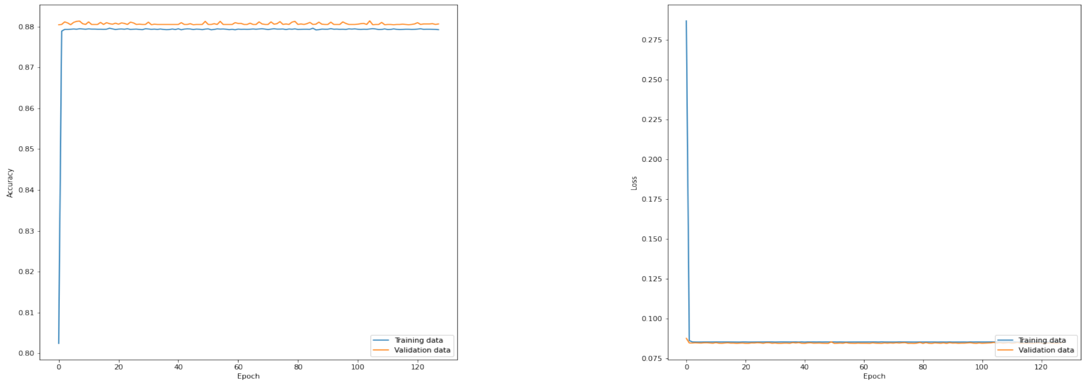
(a) Accuracy vs. Epoch



(b) Loss vs. Epoch

Figure 4: Logistic Regression Learning Curves

All three figures show very clean learning curves. The neural network shown in Figure 3 quickly attains a very high accuracy after only a few epochs and very slowly continues to rise. The logistic regression shown in Figure 4 has a more gradual slope, but still reaches about the same level of accuracy as the neural network. The linear regression shown in Figure 5 almost immediately attains its max accuracy and barely has any room to improve.



(a) Accuracy vs. Epoch

(b) Loss vs. Epoch

Figure 5: Linear Regression Learning Curves

5 Conclusion

This phase of the project focused on selecting a good neural network architecture that effectively classifies newly introduced data. The development of this algorithm is the ultimate purpose of this project. Once a model has been selected, it can be streamlined in the future to produce an effective and efficient tool that can tell what activity is being performed by an elderly adult.

Experimentation has resulted in the selection of a triple-layer model that uses the sigmoid activation function on all levels. It provides a very high accuracy and larger networks are less efficient while only increasing accuracy a small amount.

References

[1] Shinmoto Torres, R. L., Ranasinghe, D. C., Shi, Q., Sample, A. P. (2013, April). Sensor enabled wearable RFID technology for mitigating the risk of falls near beds. In 2013 IEEE International Conference on RFID (pp. 191-198). IEEE.