Application of Artificial Neural Network on Speech Signal Features for Parkinson's Disease Classification

A. John Wu[1], **B. Xin Ye**[1], **and C. Shaun-inn Wu**[1]
[1]Department of Computer Science, CSUSM, San Marcos, CA, USA

Abstract - *Since speech is affected in up to 90% of Parkinson's Disease patients, it has drawn interest as a target symptom for diagnostic testing. Using a dataset of speech sample features, multilayer perceptron was trained to accurately classify between Parkinson's Disease patients and healthy individuals. By optimizing the number of neurons in a single hidden layer network, a 0.937 accuracy model was identified. Using the single hidden layer results to inform a search for deep learning models with two hidden layers, accuracy improved to 0.952. Finding high accuracy and often 100% sensitivity, this method could be a very useful screening test for Parkinson's Disease as it is also non-invasive, quick, and can be done remotely.*

Keywords: machine learning, neural networks, Parkinson's Disease, multilayer perceptron

1 Introduction

In recent years, ever increasing amounts of data are being generated and collected about nearly everything. The high volume and number of features of most of this data make it difficult for humans to extract useful conclusions or information. However, it is possible to have computers do the hard work. Machine learning has been used more and more on a multitude of problems. Artificial neural networks are computer programs inspired by the way biological neurons are connected. They can be trained to detect patterns and relationships of data given to them as opposed to directly programmed to do so.

Medical diagnosis is a classification problem. For any disease, signs and symptoms need to be evaluated to categorize a patient as suffering from the disease or not and sometimes into different variants or severities of a disease. Use of neural networks has the potential to assist physicians in making more accurate diagnosis or even finding new patterns that can make the diagnosis. For example, a recurrent neural network trained using electronic medical record data called Doctor AI can classify a patient into the conditions or medications they are likely to have out of a large list of possibilities [1]. Forecasting depressed mood can help with the mental health of many people. Using a person's self-reported mood via a smartphone app, a long short term memory recurrent neural network can categorize if a patient will or will not have a severe depressed mood in the upcoming few days [2].

Parkinson's Disease (PD) is a degenerative disorder of the central nervous system with the most recognizable symptoms resulting from motor system dysfunction [3]. The disease progresses over a long time period and unfortunately has no cure. Current treatments for PD are done to manage symptoms and earlier diagnosis can allow for forming more optimal treatment plans.

Speech is a part of the motor system that is affected in close to 90% of patients [4] [5]. Hypokinetic dysarthria manifests in a variety of ways including affected rate of speech (slow with sudden acceleration or deceleration and involuntary pauses), difficulty with initiation of speech, repetition of words or syllables, reduced loudness/intensity, monoloudness, increased nasality, hoarseness, and imprecise consonant articulation [6]. Thus, since there is no definitive diagnostic test for PD [3], these speech changes can be a useful testing target. Further benefits as a target symptom to use for testing include that it is non-invasive and can conveniently be done without a clinical visit using telemedicine [7].

Recently, a dataset of speech sample features from 188 PD patients and 64 healthy individuals was made publicly available [8]. Using their dataset Sakar et al. [8], explored a variety of classification methods. The accuracy of the methods they explored reached up to 0.86 accuracy, with 0.84 F_1 score and 0.59 MCC. Using previous smaller Parkinson's Speech datasets, studies using different methods eventually reached accuracy of 0.9952 using a complex-valued artificial neural network with k-means clustering-based feature weighting [9]. As such, we were motivated to explore different configurations or variants of artificial neural networks using the larger sample size available with the new dataset. Obtaining improved classification accuracy with a larger dataset demonstrates the potential to expand the use of artificial neural networks on speech for diagnosing or assessing for PD to a generalized population.

2 Multilayer Perceptron

A multilayer perceptron (MLP) is a feed forward artificial neural network that has an input layer, one or more hidden layers of neurons, and an output layer of neurons. Each layer

is fully connected with the next and input moves to output in a feed forward manner through the connections and activations functions of each layer of neurons sequentially. While training, there is a backward propagation of error through the layers to update the weight of each connection [10]. This model is able to distinguish data that is not linearly separable and was able to solve XOR [11].

3 Dataset

A dataset of features from speech samples of PD patients was made available via the UCI Machine Learning Repository on November 5, 2018 [8]. The data was generated from vocal samples of 252 individuals. 188 PD patients consisted of 107 men and 81 women with ages ranging from 33 to 87 with an average of 65.1 and a standard deviation of 10.9. 61 healthy individuals consisted of 23 men and 41 women with ages ranging from 41 to 82 with an average of 61.1 and standard deviation of 8.9. Each individual provided 3 samples of sustained phonation of vowel /a/ sound recorded through a microphone set at 44.1 KHz for a total of 756 samples [8].

The provided dataset contains data features already processed from the speech samples. It also provides gender, PD status, and an ID number that can be used to identify which 3 samples are from one user. The speech features are in sections based on the technique used to extract them from the speech sample.

The first section is 21 baseline features made up of 5 jitter, 6 shimmer, 5 fundamental frequency parameters, 2 harmonicity parameters, 1 Recurrence Period Density Entropy, 1 Detrended Fluctuation Analysis, and 1 Pitch Period Entropy features. Time frequency features are made up of 3 intensity, 4 formant frequency, and 4 bandwidth features. Vocal fold features consist of 3 glottis quotient, 6 glottal to noise excitation, 7 vocal fold excitation ratio, and 6 empirical mode decomposition features. There are 84 Mel Frequency Cepstral Coefficients based features. Next, there are 182 wavelet transform based features. Finally, using a tunable Q-factor wavelet transform (TQWT) method that has not previously been applied to PD speech analysis, 432 features are generated.

All features in the available dataset already consisted of numeric values. All features except for ID number and PD status were used as possible input for neural network models. After reading in these features for all samples, each feature was normalized. First, for every feature, its mean was subtracted from each sample's value for that feature and then divided by the variance of that feature. Next, each feature was scaled by subtracting the minimum value of each feature and then dividing by the difference between the maximum and minimum value. This resulted in all feature values being in the range of 0 to 1 based on their original value.

As a classification problem, output was formatted with one output for each category. In this case, there were only two categories of PD or healthy individual and the PD status from the dataset was used to label output. So, a healthy individual had output label of [1,0] and a PD patient would be [0,1].

4 Evaluation Metrics

Models will primarily be compared using accuracy. To this end, every sample in the test data will be run through the model. The number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) will be totaled. Then, accuracy (Acc) of the model will be calculated using:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

F_1 score and Matthews Correlation Coefficient (MCC) were used to evaluate classification methods using this dataset and will also be calculated. F_1 score is the harmonic average of precision and recall. It calculates to a value in the range of 0 to 1 with 1 resulting from perfect precision and recall (no false positives or false negatives). It is calculated with this formula:

$$F_1\ score = 2\left(\frac{precision \times recall}{precision + recall}\right) \quad (2)$$

$$precision = \frac{TP}{TP+FP} \quad (3)$$

$$recall = \frac{TP}{TP+FN} \quad (4)$$

Matthews Correlation Coefficient is a useful measure of accuracy for unbalanced datasets. This dataset is unbalanced with 188 Parkinson's Disease patients and 64 healthy individuals. MCC can be calculated as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (5)$$

If any of the additions in the denominator result in 0, then the entire denominator will be set as 1.

Sensitivity (Sen), specificity (Spe), positive predictive value (PPV), and negative predictive value (NPV) are important performance measures for diagnostic tests. These performance metrics can be calculated using these formulas:

$$sensitivity = \frac{TP}{TP+FN} \quad (6)$$

$$specificity = \frac{TN}{TN+FP} \quad (7)$$

$$PPV = \frac{TP}{TP+FP} \quad (8)$$

$$NPV = \frac{TN}{TN+FN} \quad (9)$$

5 Preliminary Study

In the Sakar et al. study, for MLP, they reported accuracies ranging from 0.75 to 0.78 when using individual non-TQWT feature subsets. They reported an accuracy of 0.85 using specific TQWT settings and 0.84 when using the top 50 features out of all feature subsets as selected by mRMR filter. The use of the top 50 features was reported to be done to avoid the curse of dimensionality [8].

To establish a baseline of approximately in the same level of accuracy, initial experiments with a single hidden layer MLP were conducted using educated guesses at possible settings for network shape, activation functions, etc. This and all further testing were done in Python 3.7 using the Keras library [13] with TensorFlow [14] as its backend. Numpy [15] and Scikit-Learn [16] Python libraries were also utilized.

For a quick initial test, an MLP model using all 753 features as input, one hidden layer of 11 nodes, and 2 output nodes was used. All features were used for the simplicity of not needing to make any guesses regarding feature selection. Some method of feature selection could be used if it was found that using all 753 features resulted in problems with the curse of dimensionality at any point. This was never found to be the case and all features continued to be used as input for all experiments. Two output nodes were used to account for the two possible labels of PD or no PD. The number of neurons for the hidden layer was a fortunate choice of a random prime number. Initial weights for all connections were generate randomly.

Dataset was divided into 50% for training, 25% for validation, and 25% for testing. The initial testing accomplished by simply using the first half as training set, the third quarter as validation set, and the fourth quarter as testing set. This produced an accuracy of 0.660 over 10 tests with the best test having an accuracy of 0.772. However, since the dataset was not evenly distributed, this split results in each set having disparate percentages of PD patients (78.6% in training, 77.8% in validation, and 63.5% in testing) that might have resulted in formation of less accurate decision regions. The randomness of the initial weights and the possibility of multiple local minimas seem to explain the variation in final accuracies.

Randomizing the samples into sets should produce more comparable sets. Due to the dataset having 3 samples from each person, randomly selecting the training, validation, and testing sets could result in some samples from the same person being used in training and testing causing a potential false increase in accuracy. Using the preliminary model with a fully randomized splitting results in 0.851 average accuracy over 10 different trainings. While randomizing but ensuring all 3 samples of each person were only in 1 set has an average accuracy of 0.831 over 10 different trainings.

Next, testing with one sample from each person showed an average accuracy of 0.861 over 10 different trainings. Since the results appeared to be comparable, this trimmed dataset using the first sample of each individual in the full dataset was used for all further testing. This simplifies randomizing to training, validation, and testing sets. Furthermore, using 3 similar samples could tend toward overfitting as similar noise in the data could show up multiple times. Finally, each epoch of training will be faster by only having fewer samples.

With labels being 0 and 1, an activation of "hard sigmoid" was used at the end of the model to push the output further toward one category or the other. At the very beginning, using the sigmoid activation function for neurons resulted in final output values in the range of 0.5 to 1. This was less than optimal since the threshold value for deciding between labels was 0.5. Thus, it was quickly switched to using TanH as the activation function for neurons. This allowed for the output values to range from 0 to 1.

At this point, the dataset has 252 samples overall. Before doing any further experiments, training, validation, and testing sets with relatively similar ratios of between PD and healthy individual samples was obtained by splitting using a variety of random seeds and checking the percentage in each set with PD. This eventually generated a training set of 126 samples with 73.8% PD, validation set of 63 samples with 73.0% PD, and test set of 63 samples with 77.8% PD.

6 Single Hidden Layer MLP

With many settings and model parameters decided during preliminary testing, the major parameter for finding an accurate MLP for classifying PD would be the shape of the MLP. It is difficult to know now many neurons will produce the best model. Thus, optimizing this parameter will be done through experimentation. With a single hidden layer, it should be relatively easy because the number of neurons in only one layer needs to be changed.

Based on preliminary testing success with 11 internal neurons, MLP models starting with 1 hidden layer neuron and up to 100 hidden layer neurons were generated. As with the preliminary testing, each model was run 10 times to find their average accuracy. If it had appeared that accuracy was still trending higher near the end of this range, then testing would have been continue using models with over 100 hidden layer neurons.

Subsequently, the best few models would be trained and tested 100 times while recording TP, FP, TN, and FN. This would be used to further evaluate accuracy and the other performance metrics.

7 Two Hidden Layer MLP

With a two hidden layer MLP, there are naturally two parameters to adjust that can change the shape of the MLP. Using the same search strategy as with the single hidden layer would result in a huge number of models to test. This could just be doing a grid search using all combinations of 1 to 100 neurons in each hidden layer. Thus, a comprehensive search would require testing of 1000 models. It could be a feasible way to go with sufficient time and computational power.

It might be possible to use the single hidden layer MLP results to make some decisions about how many neurons to use for

each layer when adding a hidden layer. The number of neurons needed to be accurate in single layer might be an indication of the number of different patterns in the data that were useful for the task. Thus, once the single layer model testing produces an upper value after which adding neurons either does not improve accuracy or shows decreased accuracy, then that could be the basis for the shapes of the two layer models that might be viable to try.

With deep learning, the later layers use the previous layers output as their features. Conservatively, this might mean the first hidden layer should also be allowed to find up to the same number of patterns from the input as the single layer models. For simplicity, the range of the second layer could also go up to the upper value of the single layer models. Even using this reasoning to limit the maximum neurons in each layer could still result in a very inefficient comprehensive search. It is probably unnecessary to go up to the upper value in each layer since deep learning has the possibility of using less computation units compared to a network with similar performance using fewer layers. [12]

Furthermore, it might not be necessary or efficient to initially cover every single value in the range. The performance of using 20 neurons verses 21 neurons should likely be relatively similar. Thus, it might be possible to initially cover the search area by testing models spread out over the search area. For this study, an initial guess of each of the two hidden layers of the deeper learning model needing up to half the neurons needed in the single layer MLP. To quickly cover the range, all the prime numbers in the range will be used rather than every value in the range. This should be able to cover the range well without using a specific set interval between test points.

With the second layer acting upon the output of the first layer, it could be finding combinations of the first layer's outputs that can solve the task. With this reasoning, it could be possible to multiply the number of neurons in each layer together to get an estimation of the number of overall patterns that the model can handle. Therefore, concentrating on combinations of amounts neurons that might approximately be able to find the same number of patterns as the upper value of the single layer models might be a viable strategy. Such as if the upper value for single layer models was 30, then models using 6 and 5, 7 and 4, 8 and 4, and so on could be tested as possibilities for highly accurate two layer models.

Just as with the single layer models, the best few models found using the outlined search strategy would be further evaluated by being trained and tested 100 times while recording TP, FP, TN, and FN. This would be used to further evaluate accuracy and the other performance metrics.

8 Results and Analysis

8.1 Single Hidden Layer MLP

After running every shape single hidden layer MLP up to 100 neurons ten times each, average accuracies for each model were calculated. The best model in this initial test had an average accuracy of 0.908 and used 64 neurons in the hidden layer. The top models by average accuracy are shown in Table 1. These top models underwent further testing and evaluation with more performance metrics.

TABLE I. TOP INITIAL SINGLE HIDDEN LAYER MODELS

Neurons	64	53	14	24	31	36	47
Average Accuracy	0.908	0.905	0.903	0.903	0.903	0.903	0.903

The average accuracy of every model tested is shown in Figure 1. This shows that many models exceeded or were close to 0.90 in average accuracy. Over the range of approximately 10 to 65 neurons, most models were relatively close to the best accuracy. Above this range, accuracy declines and does not appear likely to improve again with models with even more neurons.

The top models listed in Table 1 were each trained starting with random weights 100 times while recording epochs and clock time until early stopping. For every instance run, testing to determine TP, FP, TN, and FN with the testing set was done and then used to calculate the accuracy, F_1 score, MCC, sensitivity, specificity, PPV, and NPV of that instance.

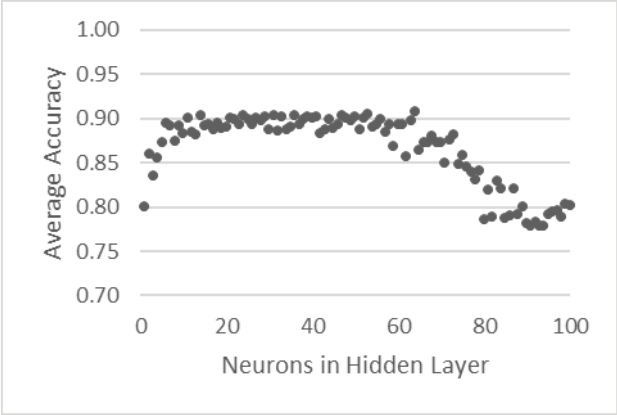

Fig. 1. Average Accuracy of Single Hidden Layer MLP Models.

The average of the performance metrics of each model from this testing is shown in Table 2 while the best performing instance of each model is shown in Table 3. Overall, each of these models continued to have high average performance.

On average, the model using 47 hidden layer neurons was the best performing with just over 0.901 accuracy, F_1 score of 0.940, and MCC of 0.699. The other models were not far behind. All models had high average sensitivity, PPV, and NPV. As can be seen in Table 3, nearly every model tested

had at least one instance where accuracy was 0.937, F_1 score of 0.961, and MCC of 0.813. These instances all had sensitivity and NPV of 1.00 as there were no false negatives. This indicates the test is very good at finding all individuals with PD and that a negative result is very reliable to mean the individual does not have PD. PPV of 0.925 means that a positive test result is highly likely to mean the tested individual has PD. Sensitivity of 0.714 means the test is relatively capable of finding health individuals but there are probably enough false positives to require further testing of test positive individuals.

TABLE II. AVERAGE METRICS OF TOP SINGLE LAYER MLPS

Neurons	Accuracy	F_1 score	MCC	Sensitivity	Specificity	PPV	NPV
14	0.894	0.937	0.676	0.995	0.548	0.886	0.956
24	0.891	0.935	0.667	0.996	0.525	0.881	0.982
31	0.897	0.938	0.689	0.996	0.555	0.887	0.982
36	0.896	0.938	0.687	0.997	0.545	0.885	0.989
47	0.900	0.940	0.699	0.995	0.568	0.890	0.981
53	0.891	0.935	0.652	0.994	0.536	0.884	0.914
64	0.872	0.925	0.544	0.993	0.451	0.867	0.762

TABLE III. BEST PERFORMING INSTANCE OF EACH TOP SINGLE LAYER MLP

Neurons	Accuracy	F_1 score	MCC	Sensitivity	Specificity	PPV	NPV
14	0.936	0.960	0.812	1.000	0.714	0.924	1.000
24	0.920	0.951	0.763	1.000	0.642	0.907	1.000
31	0.936	0.960	0.812	1.000	0.714	0.924	1.000
36	0.936	0.960	0.812	1.000	0.714	0.924	1.000
47	0.936	0.960	0.812	1.000	0.714	0.924	1.000
53	0.936	0.960	0.812	1.000	0.714	0.924	1.000
64	0.936	0.960	0.812	1.000	0.714	0.924	1.000

8.2 Two Hidden Layer MLP

Since accuracy in the single hidden layer MLP appears to start decreasing after approximately 65 neurons as seen in Figure 1, initial testing with two hidden layer MLP models will use combinations of all prime numbers up to 31 for the neurons in each layer. Since initial survey of single layer models had highest average accuracy with 64 neurons, testing of two hidden layer MLP models where the number of neurons in the hidden layers multiplied to a value near 64 (56 to 72) was also used as a search strategy for potentially accurate models. The prime number grid search generates 121 models to evaluate while the "64 combination" search generated 61 models to evaluate.

The average accuracies of an initial survey of 10 different trainings of each model are visualized in Figure 2 and Figure 3. The prime number grid search shows many peaks and valleys in average accuracy. There is not as clear of a plateau of high accuracy as seen in Figure 1 with single hidden layer models. However, there appears to be an overall trend with better accuracy in higher numbers of neurons in the first hidden layer and lower numbers of neurons in the second hidden layer. The "64 combination" search method generated models that formed a curve through the prime number grid area. These form a continuum of low first layer-high second layer neuron count to low-moderate neuron count in both layers to high first layer-low second layer neuron count. Similar to the grid search, there appear to be higher accuracy in the high first layer-low second layer neuron models.

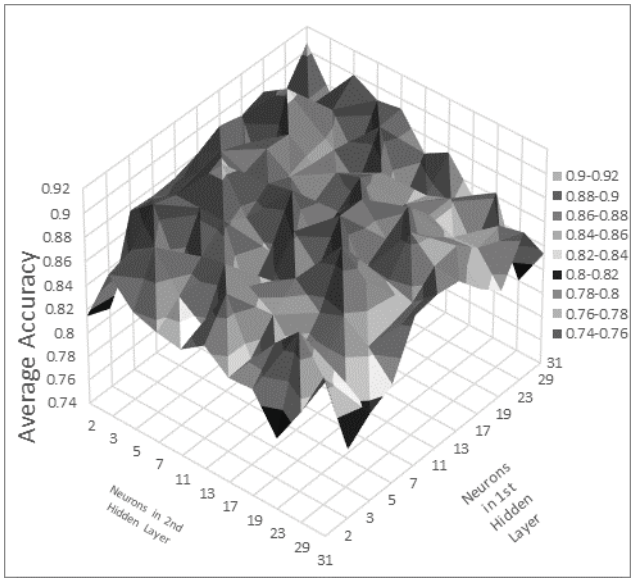

Fig. 2. Average Accuracy of Two Hidden Layer Models (Prime Number Grid).

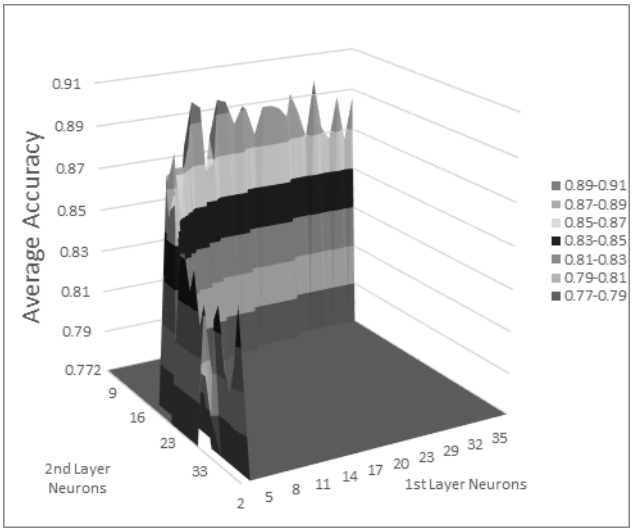

Fig. 3. Average Accuracy of Two Hidden Layer Models (Combination Search).

The best models with over 0.9 average accuracy from either of these search methods are shown in Table 4. The first 4 models are from the prime number grid method and the last model is from the "64 combination" method. As with the single layer evaluation, these top models were each trained

starting with random weights 100 times while recording epochs and clock time until early stopping. For every instance run, testing to determine TP, FP, TN, and FN with the testing set was done and then used to calculate the accuracy, F_1 score, MCC, sensitivity, specificity, PPV, and NPV of that instance.

The average performance metric values for the best two layer models from the initial survey are shown in Table 5. The 31-5 neuron model was the best average performer with 0.886 accuracy, 0.931 F_1 score, and MCC of 0.641. All the models tested were very close in performance metrics. These results are very similar to that of the single layer models seen in Table 2 with the single layer models averaging just slightly higher.

TABLE IV. TOP INITIAL TWO HIDDEN LAYER MODELS

Layer 1	11	31	23	31	11
Layer 2	5	2	3	5	6
Average Accuracy	0.906	0.906	0.905	0.900	0.900

TABLE V. AVERAGE METRICS OF TOP TWO LAYER MLPS

Neurons								
Layer 1	**Layer 2**	**Acc**	**F_1 score**	**MCC**	**Sen**	**Spe**	**PPV**	**NPV**
11	5	0.870	0.922	0.558	0.971	0.517	0.880	0.751
23	3	0.881	0.927	0.621	0.962	0.598	0.897	0.806
31	2	0.872	0.923	0.563	0.970	0.528	0.883	0.752
31	5	0.885	0.930	0.640	0.973	0.581	0.894	0.870
11	6	0.879	0.927	0.607	0.979	0.532	0.883	0.860

TABLE VI. BEST PERFORMING INSTANCE OF EACH TOP TWO LAYER MLP

Neurons								
Layer 1	**Layer 2**	**Acc**	**F_1 score**	**MCC**	**Sen**	**Spe**	**PPV**	**NPV**
11	5	0.936	0.960	0.812	1.000	0.714	0.924	1.000
23	3	0.952	0.970	0.860	1.000	0.785	0.942	1.000
31	2	0.952	0.970	0.860	1.000	0.785	0.942	1.000
31	5	0.936	0.960	0.812	1.000	0.714	0.924	1.000
11	6	0.936	0.960	0.812	1.000	0.714	0.924	1.000

However, when examining the best performing instances of each two layer model shown in Table 6, two models had instances that were able to train to a better accuracy than any instance of any single layer model. Both 23-3 and 31-2 models had instances reaching 0.952 accuracy with F_1 score of 0.970 and MCC of 0.861. This is rather significant as the best performing single layer models only had 4 false positives and 0 false negatives. This means these instances of the two layer models were able to correctly label one of those 4 false positives as a true negative.

9 Conclusion

Multilayer perceptron is potentially a very useful classification method for diagnosing Parkinson's Disease using extracted speech features. After systematically surveying many network shapes, some two hidden layer deep learning MLP was found to be able to reach 0.952 accuracy with F_1 score of 0.970 and MCC of 0.861. This is greater accuracy than prior work on this dataset using MLP or other classification methods (including naïve Bayes, logistic regression, k-nearest neighbors, random forest, support vector machine). After prior success implementing multilayer perceptron, it can still be possible to further improve accuracy by optimizing hyperparameters and adding more layers for deep learning.

As demonstrated by optimizing the number of neurons in the single hidden layer MLP, the models in the optimal range of neurons were on average 10% more accurate. While it was relatively quick to run through 100 different models, a broad plateau of higher accuracy models was found for this data. This suggests that it would be efficient to survey network shapes at higher increments initially and then subsequently do more detailed evaluation of network shapes around the better performing initial shapes to potentially find higher accuracies. However, a comprehensive search could still be used if allowed by having sufficient time or computational power available.

While the single hidden layer models already reached high accuracy, in this case, using deep learning with a two hidden layer model did improve accuracy slightly. Having the first layer generate features for the second layer may have allowed for a more complex decision area to be generated. Thus, after being properly trained, a deeper model could be able to more correctly map a decision between Parkinson's Disease samples and healthy individual samples that are relatively close in the data space.

In this case, using results from single hidden layer models was able to guide the search for two hidden layer models. Two methods were used. In the first, the range of the number of neurons in each of the two layers was restricted using the highest neuron single layer models that had high average accuracy. By seeing that a broad range of models all had similarly high accuracy in the single layer experiments, values to use in the range were spaced out to cover the range without needing to comprehensively try every value. In the second method, models where the first and second layer neuron counts multiplied together to be close to the highest single layer model neuron count were included. This would allow the two layer models to have approximately the same amount of combinations of patterns as the patterns the single hidden layer could recognize. Both of these methods were found to be able to discover viable two layer models that were at least approximately as accurate as the top single layer models. A couple of the two layer models found this way demonstrated

instances with the highest accuracy at Parkinson's Disease classification out of all models evaluated.

While it does seem that 753 features for 756 samples might result in difficulties with the curse of dimensionality, it did not arise as a problem in this case. Having all features available to train on may have allowed the MLP to find a better decision area than it would if restricted to specific subsets of the data or with algorithmically decided top features. In general, the apparent best practice would be to try initially using as much data as is available for input and then subsequently use some method to cut down the number of features only if necessary.

After optimizing the shape of single and two layer multilayer perceptrons for this research, the best models were almost always able to identify every or nearly every Parkinson's Disease patient. With the best instances finding all Parkinson's Disease patients, this method achieved 100% sensitivity. With zero false negatives, the negative predictive value was 1.00 and receiving a negative test result would indicate very high confidence to exclude a person from having Parkinson's Disease. Specificity was quite high at 78.6% in the best instances as a few healthy individuals still ended up as false positives. However, positive predictive value was still very reliable at 0.942. Use of multilayer perceptron for classification using speech samples has the potential to be a very good screening test for Parkinson's Disease. In addition to the high sensitivity of this method, it would be quick, non-invasive, and would not require a clinical or laboratory visit

10 References

[1] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart and S. Jimeng, "Doctor AI: Predicting Clinical Events via Recurrent Neural," JMLR Workshop Conf Proc, vol. 56, pp. 301-318, August 2016.

[2] Y. Suhara, Y. Xu and A. Pentland, "DeepMood: Forecasting Depressed Mood Based on Self-Reported Histories via Recurrent Neural Networks," In Proceedings of the 26th International Conference on World Wide Web (WWW '17), pp. 715-724, 2017.

[3] J. Jankovic, "Parkinson's disease: clinical features and diagnosis," Journal of Neurology, Neurosurgery & Psychiatry, vol. 79, pp. 368-376, 2008.

[4] J. Logemann, H. B. Fisher, B. Boshes and E. R. Blonsky, "Frequency and Cooccurrence of Vocal Tract Dysfunctions in the Speech of a Large Sample of Parkinson Patients," Journal of Speech and Hearing Disorders, vol. 43, pp. 47-57, 1978.

[5] A. K. Ho, R. Iansek, C. Marigliani, J. L. Bradshaw and S. Gates, "Speech Impairment in a Large Sample of Patients with Parkinson's Disease," Behavioural Neurology, vol. 11, no. 3, pp. 131-137, 1998.

[6] L. Brabenec, J. Mekyska, Z. Galaz and I. Rektorova, "Speech disorders in Parkinson's disease: early diagnostics and effects of medication and brain stimulation," Journal of Neural Transmission, vol. 124, no. 3, p. 303–334, March 2017.

[7] A. Tsanas, M. Little, P. McSharry and L. Ramig, "Accurate telemonitoring of parkinsons disease progression by noninvasive speech tests," IEEE Transactions on Biomedical Engineering, vol. 57, no. 4, pp. 884-893, April 2010.

[8] C. O. Sakar, G. Serbes, A. Gunduz, H. C. Tunc, H. Nizam, B. E. Sakar, M. Tutuncu, T. Aydin, M. E. Isenkul and H. Apaydin, "A comparative analysis of speech signal processing algorithms for Parkinson's disease classification and the use of the tunable Q-factor wavelet transform," Applied Soft Computing, vol. Volume 74, pp. 255-263, January 2019.

[9] H. Gürüler, "A novel diagnosis system for Parkinson's disease using complex-valued artificial neural network with k-means clustering feature weighting method," Neural Computing and Applications, vol. 28, no. 7, p. 1657–1666, July 2017.

[10] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, p. 533–536, 1986b.

[11] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by error propagation," in Parallel distributed processing: explorations in the microstructure of cognition, vol. 1, Cambridge, MA: MIT Press, 1986a, pp. 318-362.

[12] Y. Bengio, "Learning Deep Architectures for AI," Foundations and Trends in Machine Learning, vol. 2, no. 1, pp. 1-127, 2009.

[13] F. Chollet and Others, Keras, 2015.

[14] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

[15] T. E. Oliphant, A Guide to NumPy, Trelgol Publishing, 2006.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, T. Bertrand, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.