Evaluation of Adam and SGD optimizers on Parkison speech data classifiaction

MATH5836 Data Mining Assignmet 1

Vasiliki Vamvaka - z5251098
School of Mathematics and Statistics
University of New South Wales,NSW
Australia,2052
v.vamvaka@student.unsw.edu.au

***Abstract*—This report summarizes the evaluation of Adam and SGD optimizers applied on the prediction of Parkinson's disease from speech data. The speech data of 40 individuals with 26 voice samples per individual were used, with 20 of the test subjects identified with Parkinsonism. The performance of the two optimizers was assessed based on the confusion matrix. More specifically, the AUC, F1 Score, specificity, and sensitivity metrics were used to estimate the accuracy of the classification performance. The different model parameters investigated were the number of neurons, the learning rate, the momentum rate, and the number of hidden layers for each optimizer, respectively. Both optimizers performed better when having the following characteristics: 25 neurons in one hidden layer, learning rate 0.1 and momentum rate 0.9. However, SGD outperformed ADAM in diagnosing Parkinson's disease from speech data, achieving a maximum accuracy of 69%**

Keywords—SGD, ADAM, ANN, PD diagnosis

I. INTRODUCTION

Parkinson's disease (PD) has been identified as the number two most common neurological disorder among the elderly people with symptoms in speech, behavior, mental processing, and motor reflexes [1]. The main cause of PD has not been established so far, making its diagnosis a rather crucial task [2]. There are two main categories of PD diagnosis, namely the invasive and non-invasive methods, with the latter ones being the most popular due to their low cost and ease of implementation [3]. The non-invasive PD diagnosis is mainly based on voice data and classification to diagnose impairments on speech such as dysphonia, hypophonia, dysarthria and monotone [4].

Recently, speech based PD diagnosis has gain momentum as it is estimated that 90% of the patients suffering from PD, might present certain language difficulties signifying early stages of PD [5]. A study reported in [6] investigates the performance of different machine learning methods for PD diagnosis based on dysphonia measures. The performance was evaluated via 10-fold cross validation. More specifically, seven classifiers were used , namely, linear discriminant analysis (LDA), k-nearest neighbors (k-NN), naïve Bayes (NB), regression trees (RT), radial basis function neural networks (RBFNN), support vector machine (SVM) and Mahalanobis distance (MDC). The authors used 22 voice features from a 295 unbalanced speech dataset, achieving the highest average accuracy of 92% with a standard deviation of 0.02 using SVM. On the other hand, the worst performing classifier was found to be MDC with average accuracy of 63% and a standard deviation of 0.13. To further improve the performance of SVM and achieve better precision and F-measure, the authors suggested including the use of feature extraction techniques. Moreover, in [7] SVM was also used as a binary classifier to detect whether a person is healthy or has PD. In this study feature selection was also considered by empolying a maximum-relevance-minimum-redundancy (mRMR) scheme to reduce the dimensionality of the input features. The dataset comprised by a total number of 195 recordings with 22 features per recording. In addition, the proposed leave-one-individual-out validation scheme was used to eliminate any estimation biasing and assess the classification performance of SVM. After optimizing the SVM with mRMR, the highest accuracy of 81.53% and standard deviation of 2.17% was achieved with a minimal subset of only 4 features. The highest classification accuracy of 99.64% ± 0.01 using SVM for PD detection was reported in [8]. The main focus of this research was to investigate the impact of different dysphonia measurement in PD on the classification rate of SVM. It was found that by considering the variability of measurements across healthy and unhealthy subject, SVM's accuracy could be greatly improved.

Apart from the classic SVM classifier, Artificial Neural Networks (ANNs) have also been employed to diagnose PD based on speech data. An attempt to evaluate the suitability of ANNs for PD diagnosis along with several data pre-processing techniques can be found in [9]. The authors used a Multi-Layer Perceptron (MPL) based ANN that was tested against ten-fold cross validation, 0.8 validation split, and 0.7 validation split across 12 independent experimental runs. At the same time, different pre-processing methods were evaluated using the confusion matrix. Discretize + Resample performed better with ten-fold cross validation and 0.8 validation split giving a ROC of 96.7 and 100 respectively, while Resample+SMOTE was proven to give the best accuracy with an 0.7 validation split with an ROC of 95.8.Another approach to speech-based PD detection is by using Artificial Neural Networks (ANNs) [10]. The ANN was fine-tuned using Levenberg-Marquardt (LM) optimization with a randomized 0.8 validation split parameter. The reported classification accuracy of PD was 94.93%, however the authors did not refer to any limitation or biasing of the proposed ANN. In [11] the authors tried to classify Parkinson's disease based on ANNs. More specifically, feed forward ANNs were employed in both one and multi-dimensional configurations with different architectures in terms of hidden layers and number of neurons per layer. The effect of different feature selection techniques such as SOM, PCA, Pearson's and Kendall's correlation coefficient was assessed according to their accuracy, sensitivity, and specificity. The best performing accuracy of 81.33% was observed using one ANN with 10 neurons and a single hidden layer and Kendall's correlation

coefficient as feature selection method. On the other hand, in the case of multiple ANNs topology, two hidden layers with 10 neurons in each was proven to be the most suitable architecture for PD diagnosis. However, the authors suggested that ANN classification performance on PD speech data must be improved in order to be considered for clinical diagnosis. Finally, the performance of parallel ANNs compared to a single ANN was also investigated in [12]. The authors designed an ANN based on LM training algorithm with 3 hidden layers and 20 neurons per layer. This ANN was then expanded into a parallel configuration consisting of either 3,5,7 or 9 ANNs. In addition, a majority voting decision scheme was applied to the output of the parallel architecture to increase the robustness of the prediction. The unlearned data from each individual ANN was also sequentially fed to the next ANN. The classification accuracy of the proposed architecture on an imbalanced PD speech dataset was measured to be 91.20±1.6% in the case of 9 parallel ANNs, indicating a performance improvement of 8.4% against the single ANN architecture.

From the literature review, it is obvious that SVM represents a more robust and popular method for PD diagnosis based on speech data such as dysphonia measurements. However, the potential of ANNs on speech-based PD diagnosis is great and worth investigating as indicated by certain attempts[11, 12]. Therefore, the main purpose of this report is to investigate the use of ANNs for PD diagnosis on speech data obtained from the UCI database [4]. More specifically, the performance of two optimizers will be investigated namely Adam and SGD for different moment and learning rates as well as ANN with different number of neurons and depth. The accuracy will be assessed across 10 independent experimental runs for a validation split of 0.2. The final performance comparison is presented in terms of the confusion matrix of each optimizer. The paper is organized as follows; in Section II the data set is cited along with our proposed algorithm. Section III outlines our methodology and presents the obtained results. The paper concludes with Section IV

II. Problem Definition and Algorithm

A. Data Set and Task Definition

The main objective of this study was to evaluate the testing and training performance of SGD and Adam optimizer when applying ANNs for PD speech data classification of positive or negative test subjects. The balanced data set given refers to 40 test subjects with 20 healthy individuals and 20 PD diagnosed ones. For each subject, 26 voice samples were recorded, giving a total of 1040 different voice samples. The voice samples included sustained vowels, numbers, words, and short sentences . Furthermore, for 26 handcrafted time-frequency based features were extracted using Praat acoustic analysis software [4]. Therefore, the inputs to the ANN were the features from the dataset and the output was defined as 0 for no PD diagnosed or 1 for PD diagnose as can be seen in Fig.1.

Exploratory analysis was performed on the whole data set, identifying the distribution of the classes and the correlation plots. This dataset was then split into train and test randomly with 60% used for training and 40% used for testing the ANN. After defining the model for the ANN, two different optimizers were investigated in terms of their test and training performance. Initially, the train accuracy and loss as well as the testing accuracy and loss of SGD and Adam were evaluated for different number of hidden neurons in the range of 5 to 25 for a single hidden layer. Then, the learning rate was varied from 0.1 to 1 and the testing and train performance was recoded. Following that, only the momentum rate of SGD was changed from 0.001 to 0.1 observing the loss and accuracy of the fine-tuned ANN. Moreover, the effect of different number of hidden layers starting from 1 to 4 was also investigated in terms of accuracy and loss. Finally, the overall performance of the ANN under SGD or Adam was examined using a confusion matrix and looking at the ROC, AUC, Precision-recall curve and F1 score. The analysis of the optimized ANN was performed for 10 experimental runs to handle the effect of randomness.

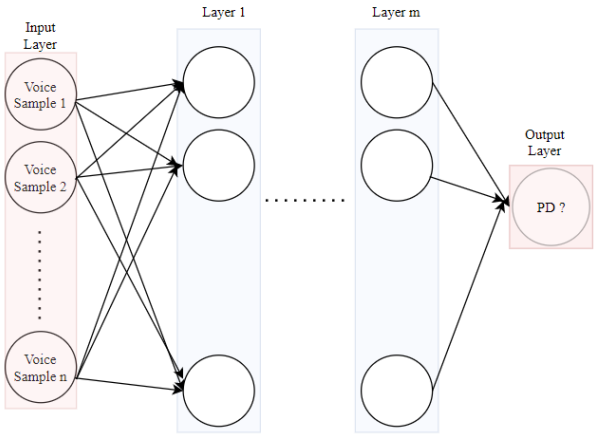

Fig.1. Artificial Neural Network illustration for Parkison's disease classification where Voice Samples represent features

B. Algorithm Definition

The algorithm that was used to assess the performance of SGD and ADAM on the Parkinson's speech data is based on the waterfall model. The idea of the waterfall approach comes from the area of engineering design in which each activity is a linear sequence of the previous one and "flows" towards one direction, hence the name. The first step was the investigation for a number of neurons in the range of 5 to 25, while fixing the learning rate and momentum rate for each optimizer fixed. After assessing the best model, the learning rate was investigated for ADAM and SGD while holding the rest hyperparameters fixed. Then, by using the "best" model from the previous results, a momentum rate was chosen. Finally, some experiments took place for several hidden layers with various number of neurons. A pseudocode of the waterfall approach is illustrated below where *lr* denotes the learning rate and *mr* denotes the momentum rate:

1. *For neurons in (5,10,15,20,25):*
    - *For i = 1, 2…,10 do*
    - *Fit the NN model with lr = t and mr = w*
2. *For lr in (0.1,0.2,0.3,0.5,1):*
    - *For i = 1, 2…,10 do*

- *Fit the NN model with neurons = best from step 1 and mr = w*

3. *For mr in (0.01,0.02,0.04,0.07,0.1,0.9):*
   - *For i = 1, 2…,10 do*
   - *Fit NN model with neurons = best from step 1 and lr = best from step 2*

III. Experimental Evaluation

The methodology followed can be divided into 4 sections. Initially, exploratory analysis was performed to visualize and understand the speech data. This was followed by defining a suitable ANN to be used as the classifier. Based on that ANN, the Adam and SGD optimizers were investigated in terms of number of neurons, learning rate, momentum rate and number of hidden layers. The confusion matrix was then used to evaluate the AUC, F1 score, specificity and sensitivity. According to the testing and training performance of the optimizers their suitability for PD classification based on speech data was extracted.

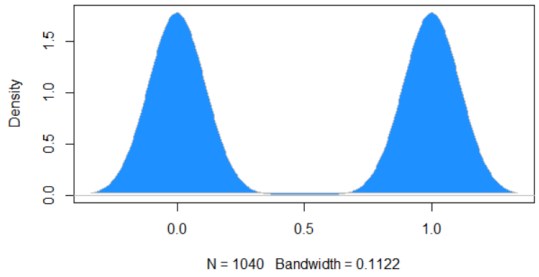

Fig.2. Multivariate normality test

A. *Exploratory analysis and data pre-proccesing*

The goal in exploratory data analysis is to get an understanding of the Parkinson's speech data set. The features of the data set are the subject ID, various speech recordings and the classes of PD diagnosis (1 is for positive PD and 0 is for negative PD). The data set is balanced as there are 520 observations for each class. This is visualized in Fig.2. Following that, the data set is tested for multivariate normality. As it can be seen from Fig.3, the data set has many outliers and does not fall to the straight

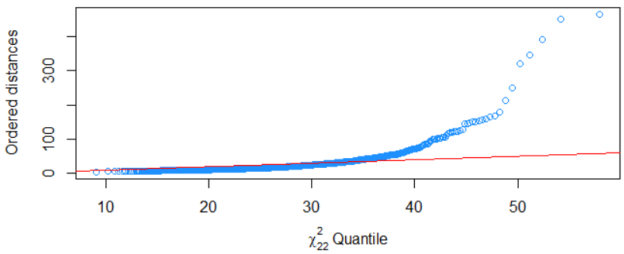

Fig.3. Multivariate normality test

line. In terms of correlation the heatmap correlation matrix is illustrated in Fig.4. Each square in the heatmap indicates the correlation between the variables. Values which are closer to zero indicate that there is not much correlation between the variables, whereas 1 is the perfect linear correlation. Generally, the higher the correlation, the darker the color is. Due to the many features of the data set, Principal Component Analysis (PCA) can also be used to better handle the data. After the PCA analysis, it was found that the first 8 components are enough to explain 90% of

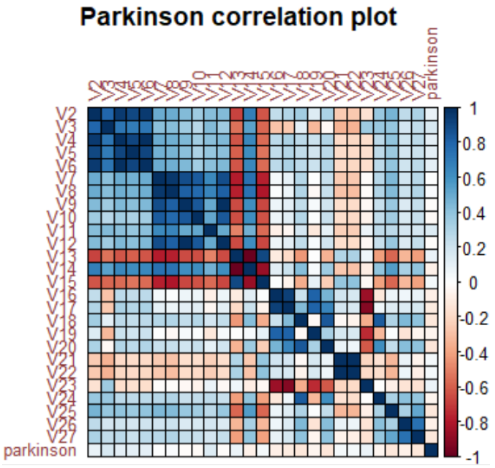

Fig.4. Correlation heat map visualisation

the variance of the data. Fig. 5 represents an example of the first two components, which explain 55.6% of the data's variance. Also, from the PCA visualization, it is obvious that the different classes are clustered together. That could increase the classification difficulty, resulting into poor test accuracy. The last steps before moving to the neural network specifications is to scale the data set and to apply one-hot-encoding. By scaling the data, the mean and standard deviation is calculated for each vector and then each element of the vector is "scaled" according to these findings, by subtracting the mean and dividing with the standard deviation. Scaling is very important in many data analysis techniques and machine learning models, as it makes it easier to compare components with different units and all the variables contribute equally to the analysis. With one-hot-encoding the variables that contain categories are transformed to binary vectors containing 0s and 1s.

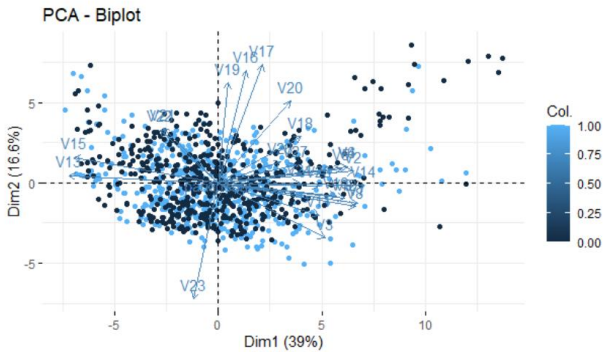

Fig.5. Principal Component Analysis (PCA) visaulization

B. *Articficial Neural Network Definition*

The model for the neural network architecture was created layer-by-layer (sequential model). As an input

layer, all the 26 features from the Parkinson's speech data were used. These features after combined with weights, which are drawn randomly from a Uniform distribution, pass to the following layer. The objective is to minimize the binary crossentropy loss function, through the SGD and ADAM optimizers. Binary crossentropy function, as the name suggests, is used as a loss function in binary classification problems. The binary crossentropy loss function works with logarithmic values, hence it needs to be in the range between 0 and 1. To achieve this the use of the softmax activation function in the last layer of the neural network is essential. The softmax activation function is a generalization of the logistic function. By passing an input of real number to the softmax function, the result is normalized between (0,1) and all the components sum to 1, so they can be interpreted as probabilities. For the hidden layers, the activation function that was used is the Rectified Linear Unit (ReLU), which is a linear function for positive values, outputting either its input if it is positive or zero. The optimization algorithm is iterative, which means that the process occurs in multiple steps and in during each step the performance is improved. For the all the models, 100 epochs were used with a validation set of 0.2. That means, 20% of the data set will not be used to train the model and will be provided to evaluate the loss and the accuracy at the end of each epoch. As a batch size, 50 samples were used to process before the model is updated. Finally, before each of the 100 epochs the training data were shuffled. To handle the uncertainty of the results, the algorithm was run 10 times to gather the average performance measures of the neural network. After the multiple experiments, summary statistics are reported based on the final model configuration.

1) Adam Optimizer Investigation

ADAM optimizer is the name for adaptive moment estimation, and it mostly used to train deep neural networks. The algorithm of ADAM is a combination of AdaGrad and RMSProp and the advantage is that it exploits the first and second moments to compute different learning rates for different parameters [13]. ADAM works well for multilayer

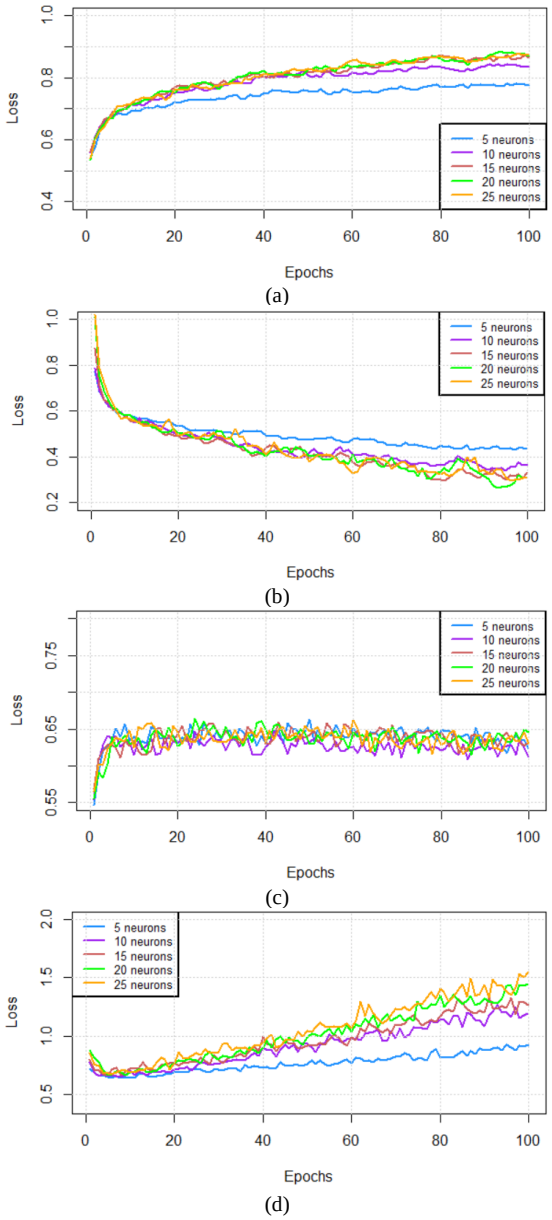

Fig.6. Effect of number of neurons on ADAM's training accuracy (a), trainning loss(b), test accuracy(c) and test loss(d).

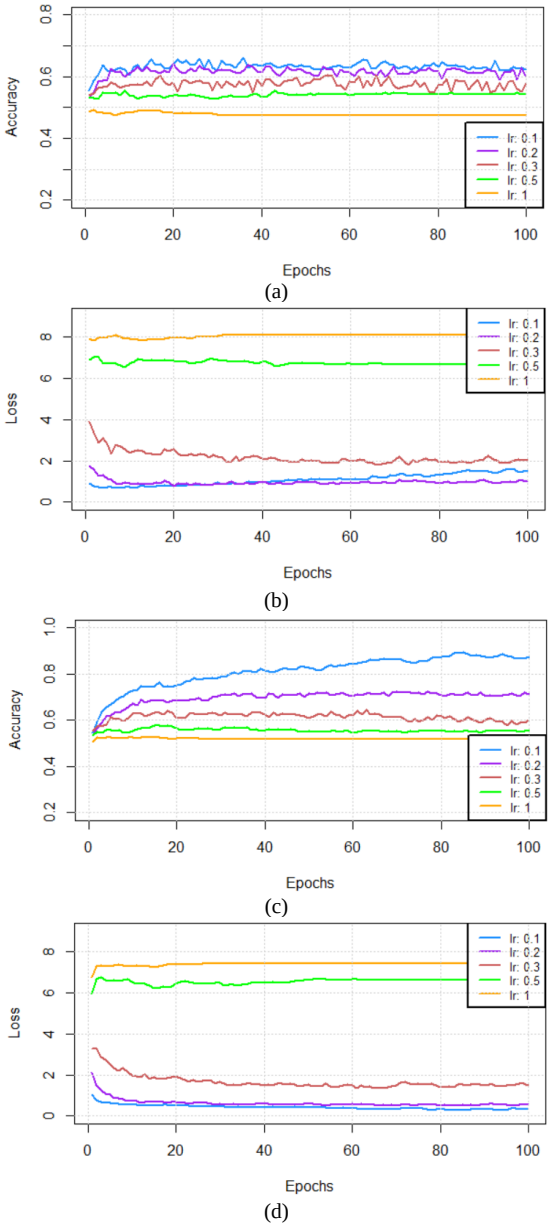

Fig.7. Effect of differnet learning rates on ADAM's training accuracy (a), trainning loss(b), test accuracy(c) and test loss(d).

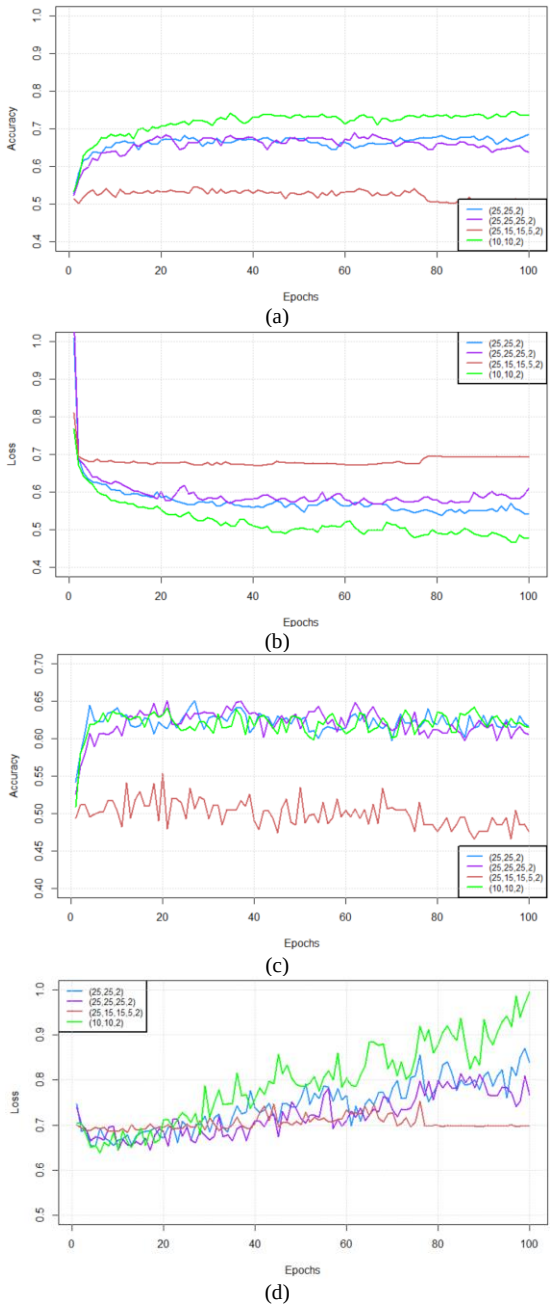

Fig.8. Effect of number of the hidden layers on ADAM's training accuracy (a), trainning loss(b), test accuracy(c) and test loss(d)

networks and high dimensional data, as it has less computational cost than other methods. For our problem, ADAM was investigated for different number of neurons for per layer, several learning rates and for 1,2,3 and 4 hidden layers. Ten experimental runs were implemented for each investigated architecture, the mean value and 95% confidence intervals were reported in Table II for each hyperparameter. In terms of the number of neurons, after experimenting with 5,10,15,20 and 25 the maximum accuracy was achieved with 25 neurons. The mean training accuracy was 81%, although the highest classification accuracy reached 64%. On the other hand, the loss was maximized as more neurons were added to the layer. As it can be seen in Fig.6 (c), test accuracy remains stable for different number of neurons, however the test loss decreases in the case of 5 neurons. The next experiment involved 25 neurons for a single layer model where the learning rate was investigated. As it can be observed from the results in Table II suggest, the best learning rate was 0.1. The neural network achieved the highest accuracy of 63% . As depicted in Fig.7 as the learning rate became larger, the value of the loss function was increased while the accuracy was decreased to almost 50% which means that the classification is random. Finally, for different hidden layers 1 to 4 and different

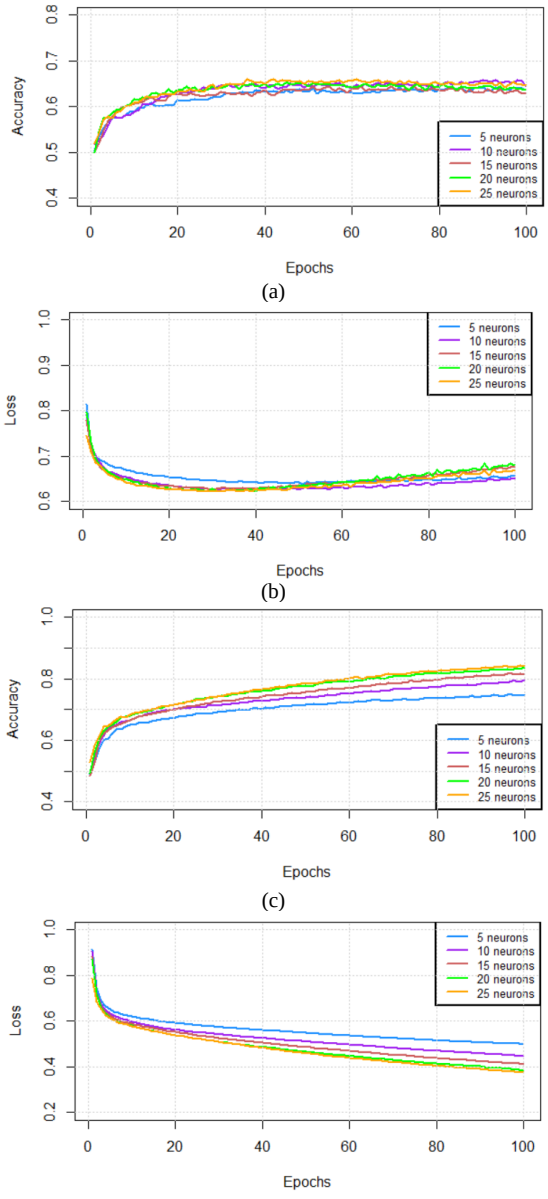

Fig.9. Effect of number of neurons on SGD's training accuracy (a), trainning loss(b), test accuracy(c) and test loss(d)

number of neurons per layer as it can be seen in Fig.8, the best outcomes were obtained when using 1 and 2 hidden layers. A classification accuracy of 64% was achieved with 1 hidden layer and 25 neurons while the second-best accuracy was 61% for 2 hidden layers and 25 neurons per layer. As the number of hidden layers was increased data the train accuracy dropped below 50%, resulting into a test accuracy of 50%. In conclusion, the "best" performed model in terms of ADAM optimizer was the model with 1 hidden layer which included 25 neurons and learning rate 0.1.

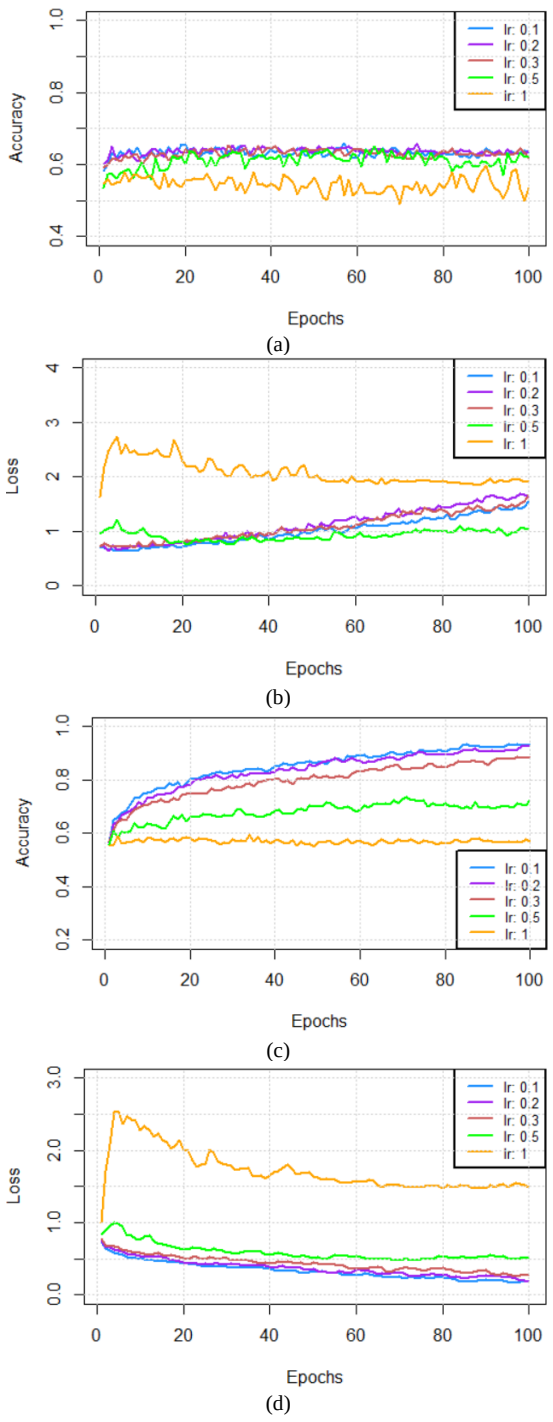

Fig.10. Effect of the learning rate on SGD's training accuracy (a), trainning loss(b), test accuracy(c) and test loss(d)

reduced. During the first experiment the effect of number of neurons was investigated as shown in Fig.9. By referring to Table I, the best classification accuracy of 64% was achieved with 25 neurons. As for the learning rate, again the value of 0.1 produces the best results with accuracy of 64%. As shown in Fig.10 similar results were obtained for learning rates between 0.1 to 0.5 in terms of accuracy and

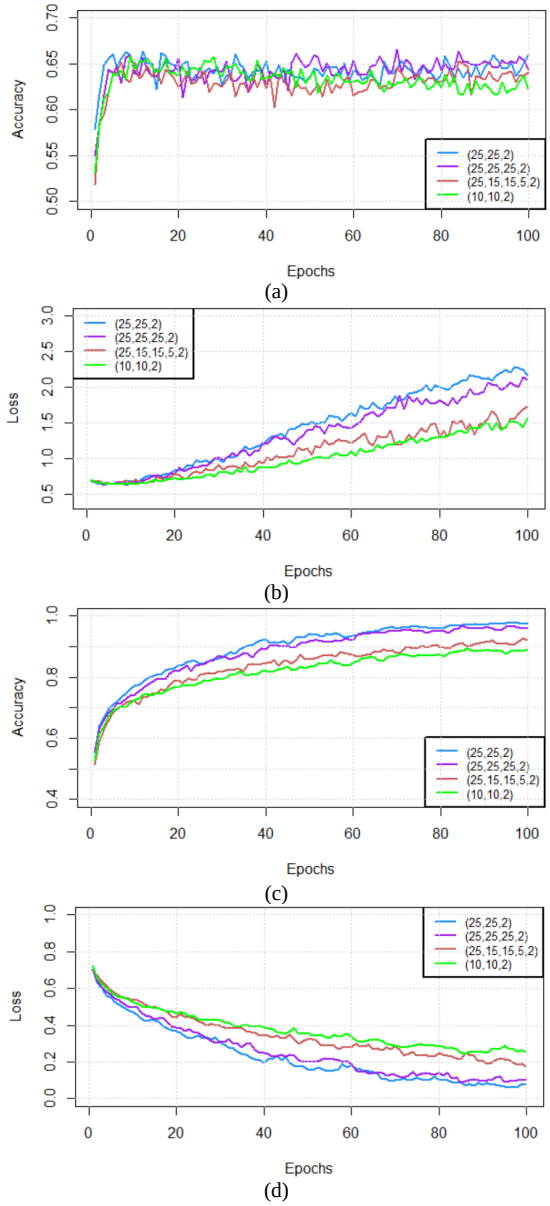

Fig.11. Effect of number of the hidden layers SGD's on training accuracy (a), trainning loss(b), test accuracy(c) and test loss(d)

2) SGD Optimizer Investigation

Stochastic gradient descent (SGD) is an optimizing method to smoothly minimize an objective function. The difference with the classic gradient descent optimization algorithm is that to update the set of parameters, only a random set of the training data is used in each iteration [14]. SGD converges faster the gradient descent, which is helpful in reducing the computational burden especially in high dimensional data sets. A disadvantage of the SGD is that sometimes convergence to the actual minimum is complicated, as it can stay to the local minimum instead of the global minimum. By introducing the idea of momentum, oscillation is loss. On the other hand, for learning rate equal to 1, the loss is around 2.7 and the classification performance is less than 60%.With 25 neurons and learning rate 0.1, the momentum rate was assessed. After evaluation of the results, 0.9 was chosen as the optimum momentum giving a test accuracy of 66%. At last, more hidden layers produce worst results and are overfitting the PD data. As a result, the best test accuracy of 65% was found with 1 hidden layer and 25 neurons. The concluded model for the SGD optimizer is a neural network with 25 neurons in one hidden layer, learning rate 0.1 and momentum rate 0.9.

TABLE I. SGD PERFORMANCE INVESTIGATION FOR DIFFERENT NEURAL NETWORK PARAMETERS FOR N=10 EXPERIMETAL RUNS

	Train				Test				
	Loss		Accuracy		Loss		Accuracy		
	Mean	95% CI	Mean	95% CI	Mean	95% CI	Mean	95% CI	
5	0.5294617	(0.52,0.54)	0.7264423	(0.72,0.74)	0.7045039	(0.67,0.72)	0.6072115	(0.59,0.62)	**Number of Neurons**
10	0.5054703	(0.49,0.52)	0.7493590	(0.73,0.76)	0.7115661	(0.68,0.73)	0.6149038	(0.60,0.62)	
15	0.4832278	(0.47,0.49)	0.7780449	(0.76,0.79)	0.7449631	(0.72,0.76)	0.6173077	(0.60,0.62)	
20	0.4665043	(0.45,0.47)	0.9074519	(0.88,0.93)	0.7370262	(0.7,0.76)	0.6254808	(0.61,0.63)	
25	**0.4554861**	**(0.44,0.46)**	**0.7977564**	**(0.78,0.80)**	**0.7407682**	**(0.71,0.76)**	**0.6293269**	**(0.62,0.64)**	
0.1	**0.5984530**	**(0.45,0.74)**	**0.8458333**	**(0.82,0.87)**	**1.775965**	**(1.57,1.97)**	**0.6396635**	**(0.61,0.64)**	**Learning Rate**
0.2	0.6958352	(0.6,0.78)	0.8325320	(0.80,0.86)	2.013769	(1.86,2.16)	0.6264423	(0.61,0.64)	
0.3	0.6493984	(0.60,0.79)	0.8105769	(0.79,0.83)	1.746525	(1.64,1.84)	0.6211538	(0.60,0.62)	
0.5	0.5860088	(0.60,0.70)	0.7108974	(0.69,0.72)	1.110327	(1.01,1.20)	0.6105769	(0.59,0.62)	
1	1.5328770	(0.95,2.11)	0.5458333	(0.52,0.56)	2.767513	(1.20,2.77)	0.5355769	(0.52,0.55)	
0.01	0.4702984	(0.57,0.58)	0.7855769	(0.77,0.79)	0.7377985	(0.71,0.75)	0.6250000	(0.61,0.63)	**Momentum Rate**
0.02	0.4750522	(0.56,0.58)	0.7815705	(0.77,0.78)	0.7621442	(0.73,0.79)	0.6189904	(0.61,0.62)	
0.04	0.4630520	(0.57,0.56)	0.7818910	(0.77,0.79)	0.7319940	(0.71,0.75)	0.6239231	(0.61,0.63)	
0.07	0.4816285	(0.56,0.57)	0.7774038	(0.76,0.78)	0.7611058	(0.72,0.80)	0.6108173	(0.59,0.62)	
0.1	0.4620094	(0.45,0.47)	0.7927885	(0.78,0.80)	0.7477439	(0.72,0.76)	0.6281250	(0.62,0.64)	
0.9	**0.5626384**	**(0.55,0.57)**	**0.7757885**	**(0.78,0.80)**	**0.7376639**	**(0.72,0.76)**	**0.6581340**	**(0.63,0.66)**	
1	**0.4554861**	**(0.44,0.46)**	**0.7977564**	**(0.78,0.80)**	**0.7407682**	**(0.71,0.76)**	**0.6293269**	**(0.62,0.64)**	**Number of Hidden Layers**
2	0.6916153	(0.64,0.73)	0.8846154	(0.87,0.89)	2.411596	(2.28,2.53)	0.6507212	(0.63,0.66)	
3	0.7046402	(0.62,0.78)	0.8642628	(0.85,0.87)	2.357691	(2.12,2.58)	0.6375000	(0.62,0.64)	
4	0.6307591	(0.54,0.71)	0.8471154	(0.83,0.85)	1.911961	(1.60,2.21)	0.6324519	(0.62,0.64)	

TABLE II. ADAM PERFORMANCE INVESTIGATION FOR DIFFERENT NEURAL NETWORK PARAMETERS FOR N=10 EXPERIMETAL RUNS

	Train				Test				
	Loss		Accuracy		Loss		Accuracy		
	Mean	95% CI	Mean	95% CI	Mean	95% CI	Mean	95% CI	
5	0.5779616	(0.54,0.60)	0.7277244	(0.69,0.75)	1.0494362	(0.95,1.14)	0.6036058	(0.59,0.62)	**Number of Neurons**
10	0.5857607	(0.54,0.62)	0.7775641	(0.76,0.79)	1.3287047	(1.22,1.43)	0.6137019	(0.60,0.62)	
15	0.5724933	(0.52,0.62)	0.8223520	(0.72,0.76)	1.5239148	(1.40,1.64)	0.6213942	(0.62,0.65)	
20	0.6112080	(0.57,0.64)	0.8176282	(0.80,0.82)	1.7797209	(1.50,1.78)	0.6312500	(0.60,0.64)	
25	**0.6602387**	**(0.57,0.74)**	**0.8118590**	**(0.79,0.82)**	**1.7149911**	**(1.51,1.91)**	**0.6209135**	**(0.63,0.64)**	
0.1	**0.6492901**	**(0.60,0.69)**	**0.8100962**	**(0.79,0.82)**	**1.722966**	**(1.57,1.80)**	**0.6204327**	**(0.60,0.63)**	**Learning Rate**
0.2	0.6723217	(0.60,0.74)	0.6910256	(0.66,0.71)	1.190320	(1.05,1.32)	0.5954327	(0.57,0.61)	
0.3	1.6273920	(1.13,2.12)	0.6004808	(0.57,0.62)	2.016304	(1.46,2.01)	0.5704327	(0.55,0.58)	
0.5	6.6311290	(5.83,7.43)	0.5504808	(0.51,0.58)	6.813512	(6.20,7.43)	0.5360577	(0.50,0.56)	
1	7.6239525	(7.42,7.82)	0.5041667	(0.49,0.51)	7.711987	(7.49,7.94)	0.4983173	(0.48,0.51)	
1	**0.6602387**	**(0.57,0.74)**	**0.8118590**	**(0.79,0.82)**	**1.7149911**	**(1.51,1.91)**	**0.6209135**	**(0.63,0.64)**	**Number of Hidden**
2	0.621628	(0.58,0.65)	0.6673077	(0.62,0.70)	0.8813420	(0.79,0.97)	0.5824519	(0.56,0.60)	
3	0.6444787	(0.61,0.67)	0.6307692	(0.57,0.68)	0.7765054	(0.65,0.89)	0.5615385	(0.53,0.59)	
4	0.6931598	(0.69,0.70)	0.5027244	(0.48,0.51)	0.6944136	(0.69,0.70)	0.5091346	(0.50,0.51)	

TABLE III. COMPARIOSN BETWEEN SGD AND ADAM

`	Accuracy	95% CI	Sensitivity	Specificity	F1-score	AUC
SGD	0.65	(0.60,0.69)	0.5634	0.7438	0.6233	0.6536
ADAM	0.62	(0.57,0.67)	0.4554	0.8030	0.5543	0.6292

C. Comparison between ADAM and SGD

Based on the hyperparameters investigations from the previous experiments, the best ANN architecture for both optimizers was found to be 25 neurons in one hidden layer, learning rate 0.1 and momentum rate 0.9. To evaluate and compare the performance of ADAM versus SGD, a classification performance comparison is given on Table III. More specifically, SGD achieved an average better accuracy of 65% with a maximum of 69% as compared to an average of 62% with a maximum of 67% in the case of ADAM. Accuracy and loss visualizations for the best performing models are included in Fig.12 and Fig.13. SGD presents a higher validation loss compared to ADAM while it outperforms ADAM in terms of training accuracy.

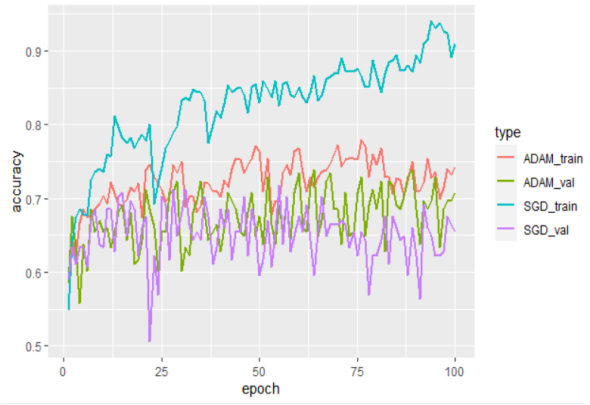

Fig.12. Accuracy comparison between SGD and ADAM

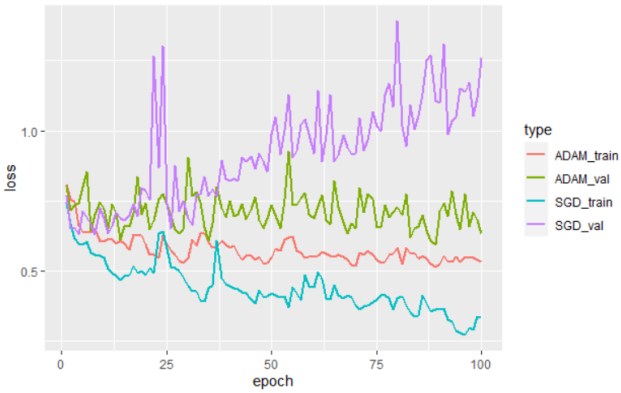

Fig.13. Loss comparison between SGD and ADAM

IV. CONCLUSION

This report summarizes the investigation of the performance of ADAM and SGD optimizer for speech-based PD diagnosis. The best ANN architecture was chosen based on the experimentation with different hyperparameters such as the number of neurons, number of hidden layers, learning and momentum rate. The best architecture was found to be 25 neurons in one hidden layer, learning rate 0.1 and momentum rate 0.9 yielding a maximum accuracy of 69% with SGD optimizer, showing that SGD outperform ADAM on PD diagnosis. As future work, a feature selection technique could be applied to the initial 26 features, reducing the dimensionality and achieve better performance of the ANN.

REFERENCES

[1] E. Ronken and G. J. M. v. Scharrenburg, "Parkinson's Disease",*Solvay Pharmaceuticals Conferences*, Amsterdam: IOS Press, 2002.

[2] J. Jankovic, "Parkinson's disease: clinical features and diagnosis," *J. Neurol. Neurosurgery Psychiatry*, vol. 79, no. 4, pp. 368-376, 2007.

[3] M. A. Little, P. E. McSharry, S. J. Roberts, D. A. E. Costello, and I. M. Moroz, "Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection," *BioMedical Engineering OnLine*, vol. 6, no. 1, p. 23, 2007/06/26 2007.

[4] B. E. Sakar *et al.*, "Collection and Analysis of a Parkinson Speech Dataset With Multiple Types of Sound Recordings," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 4, pp. 828-834, 2013.

[5] A. B. New *et al.*, "The intrinsic resting state voice network in Parkinson's disease," *Human brain mapping*, vol. 36(5), no. 1097-0193 (Electronic), pp. 1951-62, 2015.

[6] S. Lahmiri, D. A. Dawson, and A. Shmuel, "Performance of machine learning methods in diagnosing Parkinson's disease based on dysphonia measures," *Biomedical Engineering Letters*, vol. 8, no. 1, pp. 29-39, 2018/02/01 2018.

[7] C. O. Sakar and O. Kursun, "Telediagnosis of Parkinson's Disease Using Measurements of Dysphonia," *Journal of Medical Systems*, vol. 34, no. 4, pp. 591-599, 2010/08/01 2010.

[8] S. Lahmiri, "Parkinson's disease detection based on dysphonia measurements," *Physica A: Statistical Mechanics and its Applications*, vol. 471, pp. 98-105, 2017/04/01/ 2017.

[9] S. M, M. Martin, and A. Tripathi, "ANN based Data Mining Analysis of the Parkinson's Disease," *International Journal of Computer Applications*, vol. 168, pp. 1-7, 06/15 2017.

[10] A. Yasar, I. Saritas, M. A. Sahman, and A. C. Cinar, "Classification of Parkinson disease data with artificial neural networks," *IOP Conference Series: Materials Science and Engineering*, vol. 675, p. 012031, 2019/11/15 2019.

[11] L. Berus, S. Klancnik, M. Brezocnik, and M. Ficko, "Classifying Parkinson's Disease Based on Acoustic Measures Using Artificial Neural Networks," *Sensors*, vol. 19, p. 16, 12/20 2018.

[12] F. Åström and R. Koker, "A parallel neural network approach to prediction of Parkinson's Disease," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12470-12474, 2011/09/15/ 2011.
[13] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations*, 12/22 2014.
[14] S. Ruder, "An overview of gradient descent optimization algorithms," 09/15 2016.