Artificial Neural Networks for Diagnosis of Parkinson's Disease

Joseph Ritchie

Research School of Computer Science

Australian National University

Abstract. This paper investigates the use of Neural Networks for binary classification of Parkinson's Disease. The data set used was 'Parkinson Speech Dataset with Multiple Types of Sound Recordings', which was obtained from the UCI Machine Learning Repository. After initial results are obtained and compared to previous analyses[1], Heuristic Pattern Reduction is used to investigate how effective diagnosis can be when reducing the data set for training. Results of up to 96% accuracy were obtained with classification on a full dataset, and up to 89% were achieved on a dataset that had been reduced by 37.5%.

Keywords: Parkinson's, Heuristic, Pattern, Reduction, Classification, Neural, Network

1 Introduction

To investigate binary classification of Parkinson diagnosis based on voice recordings, the data set 'Parkinson Speech Dataset with Multiple Types of Sound Recordings' [1] was chosen from the UCI Machine Learning Repository. This data set is already split into two section, the training set and test set. The training set has encoded voice recordings from 20 healthy patients and 20 patients diagnosed with Parkinson's. Each patient has 26 instances, which represent data from different types of recordings (vowels, numbers, short words etc.). The test set has 28 patients, and only includes data from repeated voicing of sustained vowels. This data set was chosen first out of a desire to investigate diagnosis of a disease which has no cure but can be improved with early treatment. Second, it was chosen because it deals with binary classification which is a classic problem in machine learning, and finally each patient in the training data has 26 instances attributed to them with less for the test set. The last factor appeals to the idea of taking a detailed amount of data and being able to generalise results to smaller data sets, since in the real world, the gathering of such detailed data is neither practical or sometimes even possible.

Initially, pytorch was used to implement a neural network using the entire set. The next step was to reduce the amount of data trained on based on the paper 'Heuristic Pattern Reduction' [2]

2 Initial Method

2.1 Pre-processing Data

Pytorch was used to import both sets of data from the relevant text files. The first column was deleted, as it was only a label of the patient, and due to the grouped nature of multiple instances per patient, training in this case was done without randomising order. The last column in the data set was the official diagnosis, which was separated into a target vector. The same was done for the test set.

The training and test data sets were not mixed for the purposes of this paper. The reason for this is that the test set has less information, so the idea was to see if accuracy could be kept while training on a smaller set, and the benefits of randomising or mixing the data sets wasn't deemed as important as testing on less complete information. Particularly worthy of note here is the absence of a UPDRS rating in the test set – this is the most commonly used scale in clinical study of Parkinson's Disease and requires an interview and observation by a qualified professional. In the real world it is not realistic to spend the time and expertise required to rate everyone being tested with a UPDRS scale, and it would be far preferable to be able to diagnose based only on voice recordings.

The final step taken to reduce the complexity of the network and increase accuracy was to normalise the data of both sets – this was chosen over standardisation as the data given does not follow a normal distribution, and on inspection there are not any extreme outliers. The formula used for standardising the data was:

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

2.1 Network Structure

Since this is a binary classification problem, it is appropriate to have a hidden layer, as the absence of a hidden layer would just represent a regression analysis. Various numbers were experimented with for the number of neurons in our (only) hidden layer, with the final number being 24. The rough rules used for the range of experimentation were that the hidden layer should have a number of neurons between the number of input neurons and output neurons in order to avoid overfitting, and the experimental formula:

$$N = \frac{N_s}{x * (N_i + N_o)}$$

Where:

N_i = number of input neurons
N_o = number of output neurons.

The number of output neurons given was 2, so that instead of a positive/negative dichotomy, a number could be assigned to each option, with the largest being taken as the prediction. A final learning rate of 0.001 was used.

In terms of activation functions, the output layer only used a linear function, as we were not classifying results by probability (ie >0.5 = 1), but taking the maximum of each result. The hidden layer used only a sigmoid function:

$$\sigma = \frac{1}{1 + e^x}$$

The loss function used was cross entropy, as is appropriate for binary classification. Finally, multiple optimisation methods were considered; Adam, Momentum, and SGD. The final algorithm chosen was SGD – the existence of multiple instances per patient leaves open the possibility of getting 'stuck' in a local minima, so SGD was considered as a good method to enable the network to jump to new local minima if needed.

2.2 Method of Analysing Data

One of the challenges with this dataset is that there are multiple instances for each patient (28 for training patients, 6 for testing patients). We need to decide whether we would like to combine the instances of each patient into a single instance and run the neural network over that or run them as is and sort the results later. The network was trained over each instance individually, due to the (far) lesser number of instances per patient in the testing data – it seems preferable to train a network to diagnose individual voice recordings and then run it over a smaller amount of such instances (with a similar amount of attributes) than training it to diagnose over 560 attributes for the training set and then diagnosing over the 162 instances in the testing set. The issue of coming up with a single diagnosis for each patient was then resolved by aggregating each diagnosis and determining whether the network made a majority of positive diagnoses: for the training set a threshold of 14/26 positives was required, and for the testing set a threshold of 4/6.

2.3 Analysing Results

After achieving a single diagnosis for each patient, the results were put into a confusion matrix. Accuracy was then measured by the equation:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Where:

TP = True positive (model correctly predicts positive)

TN = True negative (model correctly predicts negative)

FP = false positive (model incorrectly predicts positive)

FN = False negative (model incorrectly predicts negative)

Recall is a useful measurement to include, as it determines what percentage of the actual positive diagnoses is classified correctly. was measured by the equation:

$$Recall = \frac{TP}{TP + FN}$$

Finally, precision is also included, and shows the percentage of the time that positive classifications made by the algorithm are correct. This is measured by:

$$Precision = \frac{TP}{TP + FP}$$

Results for the neural networks described above, when run over 4000 epochs are below:

	Training Set	**Test Set**
Accuracy	0.77	0.96
Recall	0.95	0.96
Precision	0.61	1

This is a result higher than expected when comparing to the highest accuracy obtained by previous studies on this data set using the most similar methods [2] – previous accuracy was 82%. Of note here is that previous methods involved using the training set also for validation, whereas this paper used each data set separately. Some speculation as to why this difference occurs may be investigated by looking at the confusion matrix of data on the training set, which is more insightful than the test set, since it includes half negative/healthy patients, as opposed to all sick patients:

$$\begin{bmatrix} 12 & 8 \\ 1 & 19 \end{bmatrix}$$

Where the labels are from top left to bottom right: TN, FP, FN, TP.

From this we can see that the network is far more aggressive in classifying patients as positive incorrectly – there is only one case of incorrectly classifying a no, but 8 cases of incorrectly classifying true. So then given a test set which is made up of exclusively positive people, it makes sense that the network will predict correctly a very high % of the time. This is not necessarily a good feature of the network, however in the real world (provided adequate

professional diagnosis methods) it may be more preferable to overdiagnose, so that the false positives can be confirmed as such but the high positives can confirmed and treated as such. There are fewer victims of Parkinson's 'slipping through the cracks'.

One reason for this 'aggressive' positive diagnosis may be the existence of the UPDRS in the training data set. The value of this attribute is by far the attribute with the biggest difference between healthy and sick patients, being always 1 for a healthy patient and up to 40 for some sick patients. Intuitively, the removal of this should result in less aggressive positive diagnosis, since the network cannot learn patterns of positive patients as quickly without such an obviously indicative attribute like the UPDRS. As a final experiment before moving onto pattern reduction, we remove the UPDRS from the training set, both to test the nature of the high frequency of positive diagnosis on the test set and also to discover the value of UPDRS in the real world, and how realistic diagnosis on voice may be if unaided by a time-costly UPDRS evaluation. Indeed, we can see that when we remove this and run the network with all the same parameters and number of epochs we get for less accuracy:

	Training Set	**Test Set**
Accuracy	0.62	0.64
Recall	0.50	0.64
Precision	0.40	1

3 Heuristic Reduction Approach

Using the general idea outlined in 'Heuristic Pattern Reduction' [2], we start to remove patterns from our training set. Intuition tells us that due to the existence of UPDRS, we will be able to dramatically result our training set. Indeed, we can see that removing the last quarter of both sick and healthy patients, we get:

	Training Set	**Test Set**
Accuracy	0.85	0.93
Recall	0.80	0.93
Precision	0.47	1

So, we want a heuristic we can use to remove an even larger amount of patterns from our data set and still achieve high accuracy for positive diagnosis. Since this is not a similar dataset to the one used for grading students [1], we create our own heuristic – one that is immediately intuitive and highly relevant to this data set and using a measurement that has been a theme already used thus far; the UPDRS. Note that this need not be restricted to analysis on Parkinson's – any dataset with a feature that varies so consistently between classes might be able to be reduced with the same heuristic.

Since the high UPDRS ratings are the most standout feature, we keep all of those and only include 5 of the 20 healthy patients to train on. This is ideal for usage in medical diagnosis, as diagnosed patients should already have a UPDRS rating, so if reduction is possible with good results then neural networks can be trained without going to the effort of having a professional run a full UPDRS rating on as many healthy patients in order to get samples for the network. Using this much smaller sample of 25 patients instead of 40, we get:

	Training Set	**Test Set**
Accuracy	0.80	0.89
Recall	1	0.89
Precision	1	1

We arrive at an encouraging result – using a heuristic of a pattern with a result that varies much more than all others, we can reduce the size of our training set significantly while keeping strong levels of diagnosis.

4 Conclusion and Future Work

Neural networks show much promise for the diagnosis of Parkinson's Disease. As can be seen, accurate diagnosis (whether positive or negative) occurs at a fairly high rate, and considering this is a disease that benefits from early treatment it is an important area of research. Future work will be able to expand and improve upon this greatly. Possible areas of research are:

-Using Convoluted Neural Networks to extract features automatically. This may allow for important features only to be extracted and make computation more efficient, although it would require the original recordings which is not available.
-Using Deep Learning to pass the training data through many more layers than has been done here.
-More efficient methods to deal with multiple recordings per person. It may be that certain types of recordings are more powerful for diagnosis, or certain combinations of recordings contain information when analysed together.

References

1. Sakar Erdogu, Betul, and Isenkul, M.Erdem, and Sakar, C. Oran, and Sertbas, Ahmet, and Gurgen, Fikret, and Delil, Sakir, and Apaydin, Julya, and Kursun, Olcay. 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
2. Gedeon, T.D, and Bowden, T.G (1992). Heuristic Pattern Reduction, International Joint Conference on Neural Networks
3. Erdogdu Sakar, B., Isenkul, M., Sakar, C.O., Sertbas, A., Gurgen, F., Delil, S., Apaydin, H., Kursun, O., 'Collection and Analysis of a Parkinson Speech Dataset with Multiple Types of Sound Recordings', IEEE Journal of Biomedical and Health Informatics, vol. 17(4), pp. 828-834, 2013