Diagnosis of Parkinson's Disease by Boosted Neural Networks

Mehmet Can
mcan@ius.edu.ba

International University of Sarajevo
Faculty of Engineering and Natural Sciences
Hrasnicka Cesta 15, 71000 Sarajevo
Bosnia and Herzegovina

Abstract

A boosting by filtering technique for neural network systems with back propagation together with a majority voting scheme is presented in this paper. Previous research with regards to predict the presence of Parkinson's Disease has shown accuracy rates up to 92.9% [1] but it comes with a cost of reduced prediction accuracy of the minority class. The designed neural network system boosted by filtering in this article presents a significant increase of robustness and it is shown that by majority voting of the parallel networks, recognition rates reach to > 90 in a imbalanced 3:1 imbalanced class distribution Parkinson's Disease data set.

Keywords—Machine learning, Parallel neural networks, boosting by filtering, Parkinson's Disease

1. INTRODUCTION

The cause of Parkinson's disease is unknown, however research has shown that a degradation of the dopaminergic neurons affect the dopamine production to decline [2]. Dopamine is used by the body to control movement, hence the less dopamine that is in circulation the more difficult the person to control the movements and may experience tremors and numbness in extremities. As a direct cause of reduced control of motor-neurons in the central nervous system, the ability of articulating vocal phonetics is reduced. In this case the symptom (the inability to articulate words) is related to the presence of Parkinson's disease and is described as Dysphonia, a reduced functionality of the vocal cords. One of the immediate effects of vocal Dysphonia is that the sound of the words is hardly recognizable [3].

The field of speech processing and development of speech recognition systems have received considerable attention during the last decades. With the availability of portable phones and analyzing methods involving traditional digital signal processing approaches such as hidden Markov models, Kalman filter, short-time frequency analysis and wavelet transforms are successfully used for both speech enhancement and speech recognition applications [4, 5, 6, 7, 8, 9, 10, 11].

Scientific research on vocal recordings of patients that suffer from Parkinson's disease are not abundant. The data set used in this study was collected by Max. A Little et. al. [12] who used support vector machine techniques to distinguish between the people who have normal vocal signals and who suffer from Parkinson's disease. They achieve a classification accuracy of 91.4% but they do not report single class true positive rates. This is noteworthy because of the highly imbalanced sick to healthy ratio (3:1) data class distribution of the Parkinson's disease data set [13].

R. Das [1] has made a comparative study on this data set with regard to neural networks, DMNeural analysis, and regression analysis and decision trees with the presented results of classification accuracy of 92.9%, 84.3%, 88.6% and 84.3% respectively. The analysis was carried out on data exploration of SAS software. Another study by M. Lee et. al. [14] on the imbalanced data problem in biomedical data uses a sampling scheme in collaboration with a naive Bayes classifier to deal with the imbalanced data problem. The sampling pattern starts with a small portion of the data to train the classifier, and then successively to increase the number of training samples regardless of the initial class distribution. This method results in positive predictive rates of 66.2% for normal subjects and 90.0% for subjects with Parkinson's disease.

Neural networks are the tools that should be recalled for any classification job. They are developed enormously since the first attempts made modeling the perceptron architecture six decades ago [15].

The massive parallel computational structure of neural networks is what has contributed to its success in predictive tasks. It has been shown that the approach of using parallel networks is successful with respect to increasing the predictive accuracy of neural networks in robotics [16] and in speech recognition [17]. In the case of the speech recognition application, Lee [17] attempts to forward propagate unlearned data to a neighboring neural network and achieves an increase for the classification accuracy of at most 6.7% compared to a traditional multi-layer neural network approach.

This work presents a parallel networks system which is bound together with a majority voting system in order to further increase the predictive accuracy of a Parkinson's Disease data set based on vocal recordings. It is also proven that forward propagation of untrained data increases the predictive accuracy of the under-representative class.

For the proposed system it is shown with a case study of Parkinson's disease that some of the difficulties with imbalanced data sets are resolved. The type of network used is the standard feedforward back-propagation neural network, since they have proven useful in biomedical classification tasks [18]. The performance of the trained neural networks is evaluated according to the true positive, true negative and accuracy rate of the prediction task. Furthermore the area under the receiver operating characteristic curve and the Mean Squared Error are used as statistical measurements to compare the success of the different models.

The paper is organized as follows; first, the data used in this work is introduced in section 2. The neural network that is boosted by filtering is illustrated in section 3. Results of the research are shown in section 4 which followed by a conclusion.

2. DATA SET OF PARKINSON'S DISEASE

The data used in this study is a voice recording originally done at University of Oxford by Max Little [12]. In the same study a detailed presentation is made on the specificities of the recording equipment as well as in what environment the experiment was conducted. The data consists of 195 recordings extracted from 31 people whom 23 are suffering of Parkinson's disease. The time since first diagnosis of Parkinson's disease was done 0 to 28 years ago and the age of the subjects ranged from 46 to 85 years and a total of 6 vocal sounds were recorded from each subject. For more information on the data set refer to ref. [12]. Furthermore, the data set consists of 22 attributes. Little et al. apply a correlation filter and of these 22 attributes 12 are removed after applying the filter. A corresponding data table for this correlation filter can be seen in appendix table 4. Each correlation coefficient, which is less than 0.95 is considered not to contribute to classification accuracy, thus the attribute is removed. All in all, a total of 10 attributes are kept after the correlation filter has been applied. Table 4 in appendix illustrates which features are kept and which are removed by the correlation filter. Table 3 in the appendix shows gives a brief explanation of meaning of the attributes; references [19, 12] should be consulted for details on how the attributes are derived and what they indicate.

No	**Attribute name**	**Description**
1	MDVP:Jitter(Abs)	Variation in fundamental frequency
2	Jitter:DDP	Variation in fundamental frequency
3	MDVP:APQ	Measures of variation in amplitude
4	Shimmer:DDA	Measures of variation in amplitude
5	NHR	Ratio of noise to tonal components
6	HNR	Ratio of noise to tonal components
7	status	(1)-Parkinson's Disease, (0)-Healthy
8	RPDE	Dynamic complex measurement
9	DFA	Signal fractal scaling exponent
10	D2	Dynamic complex measurement
11	PPE	Non-linear measure of fundamental frequency

Table 1: Table describing the attributes that are not removed after applying the correlation filter or by other reasons mentioned in Little et. al [12] where the exact computations of each measurement is described.

3. ARTIFICIAL NEURAL NETWORKS

Nervous systems existing in biological organism for years have been the subject of studies for mathematicians who tried to develop some models describing such systems and all their complexities. Artificial Neural Networks emerged as generalizations of these concepts with mathematical model of artificial neuron due to McCuloch and Pitts [20] described in 1943 definition of unsupervised learning rule by Hebb [21] in 1949, and the first ever implementation of Rosenblatt's perceptron [22] in 1958. The efficiency and applicability of artificial neural networks to computational tasks have been questioned many times, especially at the very beginning of their history the book "Perceptrons" by Minsky and Papert [23], published in 1969, caused dissipation of initial interest and enthusiasm in applications of neural networks.

It was not until 1970s and 80s, when the back propagation algorithm for supervised learning was documented that artificial neural networks regained their status and proved beyond doubt to be sufficiently good approach to many problems. Artificial Neural Network can be looked upon as a parallel computing system comprised of some number of rather simple processing units (neurons) and their interconnections. They follow inherent organizational principles such as the ability to learn and adapt, generalization, distributed knowledge representation, and fault tolerance. Neural network specification comprises definitions of the set of neurons (not only their number but also their organization), activation states for all neurons expressed by their activation functions and offsets specifying when they fire, connections between neurons which by their weights determine the effect the output signal of a neuron has on other neurons it is connected with, and a method for gathering information by the network that is its learning (or training) rule.

3.1. Architecture

From architecture point of view neural networks can be divided into two categories: feed-forward and recurrent networks. In feed-forward networks the flow of data is strictly from input to output cells that can be grouped into layers but no feedback interconnections can exist. On the other hand, recurrent networks contain feedback loops and their dynamical properties are very important.

The most popularly used type of neural networks employed in pattern classification tasks is the feedforward network which is constructed from layers and possesses unidirectional weighted connections between neurons. The common examples of this category are Multilayer Perceptron or Radial Basis Function networks, and committee machines.

Multilayer perceptron type is more closely defined by establishing the number of neurons from which it is built, and this process can be divided into three parts, the two of which, finding the number of input and output units, are quite simple, whereas the third, specification of the number of hidden neurons can become crucial to accuracy of obtained classification results.

The number of input and output neurons can be actually seen as external specification of the network and these parameters are rather found in a task specification. For classification purposes as many distinct features are defined for objects which are analyzed that many input nodes are required. The only way to better adapt the network to the problem is in consideration of chosen data types for each of selected features. For example instead of using the absolute value of some feature for each sample it can be more advantageous to calculate its change as this relative value should be smaller than the whole range of possible values and thus variations could be more easily picked up by Artificial Neural Network. The number of network outputs typically reflects the number of classification classes.

The third factor in specification of the Multilayer Perceptron is the number of hidden neurons and layers and it is essential to classification ability and accuracy. With no hidden layer the network is able to properly solve only linearly separable problems with the output neuron dividing the input space by a hyperplane. Since not many problems to be solved are within this category, usually some hidden layer is necessary.

With a single hidden layer the network can classify objects in the input space that are sometimes and not quite formally referred to as simplexes, single convex objects that can be created by partitioning out from the space by some number of hyperplanes, whereas with two hidden layers the network can classify any objects since they can always be represented as a sum or difference of some such simplexes classified by the second hidden layer.

Apart from the number of layers there is another issue of the number of neurons in these layers. When the number of neurons is unnecessarily high the network easily learns but poorly generalizes on new data. This situation reminds auto-associative property: too many neurons keep too much information about training set rather "remembering" than "learning" its characteristics. This is not enough to ensure good generalization that is needed.

On the other hand, when there are too few hidden neurons the network may never learn the relationships amongst the input data. Since there is no precise indicator how many neurons should be used in the construction of a network, it is a common practice to build a network with some initial number of units and when it trains poorly this number is either increased or decreased as required. Obtained solutions are usually task-dependant.

3.2 Activation Functions

Activation or transfer function of a neuron is a rule that defines how it reacts to data received through its inputs that all have certain weights.

Among the most frequently used activation functions are linear or semi-linear function, a hard limiting thresh-old function or a smoothly limiting threshold such as a sigmoid or a hyperbolic tangent. Due to their inherent properties, whether they are linear, continuous or differentiable, different activation functions perform with different efficiency in task-specific solutions.

For classification tasks antisymmetric sigmoid tangent hyperbolic function is the most popularly used activation function:

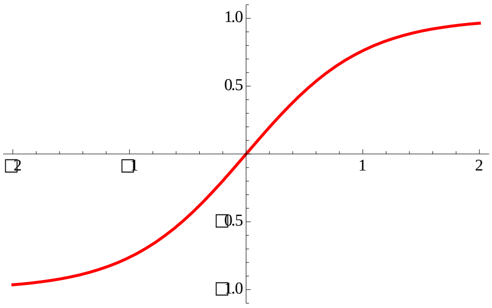


Fig. 1. Antisymmetric sigmoid tangent hyperbolic activation function

3.3 Learning Rules

In order to produce the desired set of output states whenever a set of inputs is presented to a neural network it has to be configured by setting the strengths of the interconnections and this step corresponds to the network learning procedure. Learning rules are roughly divided into three categories of supervised, unsupervised and reinforcement learning methods.

The term supervised indicates an external teacher who provides information about the desired answer for each input sample. Thus in case of supervised learning the training data is specified in forms of pairs of input values and expected outputs. By comparing the expected outcomes with the ones actually obtained from the network the error function is calculated and its minimization leads to modification of connection weights in such a way as to obtain the output values closest to expected for each training sample and to the whole training set.

In unsupervised learning no answer is specified as expected of the neural network and it is left somewhat to itself to discover such self-organization which yields the same values at an output neuron for new samples as there are for the nearest sample of the training set.

Reinforcement learning relies on constant interaction between the network and its environment. The network has no indication what is expected of it but it can induce it by discovering which actions bring the highest reward even if this reward is not immediate but delayed. Basing on these rewards it performs such re-organization that is most advantageous in the long run [22].

The modification of weights associated with network interconnections can be performed either after each of the training samples or after finished iteration of the whole training set.

The important factor in this algorithm is the learning rate η whose value when too high can cause oscillations around the local minima of the error function and when too low results in slow convergence. This locality is considered the drawback of the backpropagation method but its universality is the advantage.

3.4 Architecture of artificial neural networks, Committee Machines

As the base topology of artificial neural network committee machines [24] with the feed-forward multilayer perceptron with sigmoid activation function trained by backpropagation algorithm is used.

In committee machines approach, a complex computational task is solved by dividing it into a number of computationally simple tasks and then combining the solutions to those tasks. In supervised learning, computational simplicity is achieved by distributing the learning task among a number of experts, which in turn divides the input space into a set of subspaces. The combination of experts is said to constitute a committee machine. Basically, it fuses knowledge acquired by experts to arrive at an overall decision that is supposedly superior to that attainable by anyone of them acting alone. The idea of a committee machine may be traced back to Nilsson [19] (1965); the network structure considered therein consisted of a layer of elementary perceptrons followed by a vote-taking perceptron in the second layer.

Committee machines are universal approximators. They may be classified into two major categories:

1. Static structures. In this class of committee machines, the responses of several predictors (experts) are combined by means of a mechanism that does not involve the input signal, hence the designation "static." This category includes the following methods:

- Ensemble averaging, where the outputs of different predictors are linearly combined to produce an overall output.
- Boosting, where a weak learning algorithm is converted into one that achieves arbitrarily high accuracy.

2. Dynamic structures. In this second class of committee machines, the input signal is directly involved in actuating the mechanism that integrates the outputs of the individual experts into an overall output, hence the designation "dynamic."

3.5 Boosting

Boosting is a method that belongs to the "static" class of committee machines. Boosting is quite different from ensemble averaging. In a committee machine based on ensemble averaging, all the experts in the machine are trained on the same data set; they may differ from each other in the choice of initial conditions used in network training. By contrast, in a boosting machine the experts are trained on data sets with entirely different distributions; it is a general method that can be used to improve the performance of any learning algorithm.

Boosting' can be implemented in three fundamentally different ways:

1. Boosting by filtering. This approach involves filtering the training examples by different versions of a weak learning algorithm. It assumes the availability of a large (in theory, infinite) source of examples, with the examples being either discarded or kept during training. An advantage of this approach is that it allows for a small memory requirement compared to the other two approaches.

2. Boosting by subsampling. This second approach works with a training sample of fixed size. The examples are "resampled" according to a given probability distribution during training. The error is calculated with respect to the fixed training sample.

3. Boosting by reweighting. This third approach also works with a fixed training sample, but it assumes that the weak learning algorithm can receive "weighted" examples. The error is calculated with respect to the weighted examples.

In this paper Boosting by filtering is used. This algorithm is due to Schapire [25] (1990). The original idea of boosting described in Schapire (1990) is rooted in a distribution free or probably approximately correct (PAC) model of learning. To be more specific, the goal of the learning machine is to find a hypothesis or prediction rule with an error rate of at most ε, for arbitrarily small positive values of ε, and this should hold uniformly for all input distributions.

In boosting by filtering, the committee machine consists of three experts or subhypotheses. The algorithm used to train them is called a boosting algorithm. The three experts are arbitrarily labeled "first," "second," and "third." The three experts are individually trained as follows:

1. The first expert is trained on a set consisting of N_2 examples.

2. The trained first expert is used to filter another set of examples by proceeding in the following manner:
Flip a fair coin; this in effect simulates a random guess.
If the result is heads, pass new patterns through the first expert and discard correctly classified patterns until a pattern is misclassified. That misclassified pattern is added to the training set for the second expert.

If the result is tails, do the opposite. Specifically, pass new patterns through the first expert and discard incorrectly classified patterns until a pattern is classified correctly. That correctly classified pattern is added to the training set for the second expert.

Continue this process until a total of N_1 examples has been filtered by the first expert. This set of filtered examples constitutes the training set for the second expert.

By following this coin flipping procedure, it is ensured that if the first expert is tested on the second set of examples, it would have an error rate of 1/2. In other words, the second set of N_1 examples available for training the second expert has a distribution entirely different from the first set of N_2 examples used to train the first expert. In this way the second expert is forced to learn a distribution different from that learned by the first expert [26].

3. Once the second expert has been trained in the usual way, a third training set is formed for the third expert by proceeding in the following manner:

• Pass a new pattern through both the first and second experts. If the two experts agree in their decisions, discard that pattern. If, on the other hand, they disagree, the pattern is added to the training set for the third expert.

• Continue with this process until a total of N_1 examples have been filtered jointly by the first and second experts. This set of jointly filtered examples constitutes the training set for the third expert.

The third expert is then trained in the usual way, and the training of the entire committee machine is thereby completed.

Let N_2 denote the number of examples that must be filtered by the first expert to obtain the training set of N_1 examples for the second expert. Note that N_1 is fixed, and N_2 depends on the generalization error rate of the first expert. Let N_3 denote the number of examples that must be jointly filtered by the first and second experts to obtain the training set of N_1 examples for the third expert.

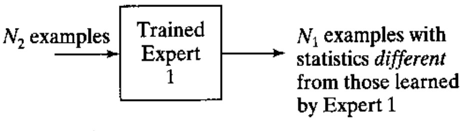
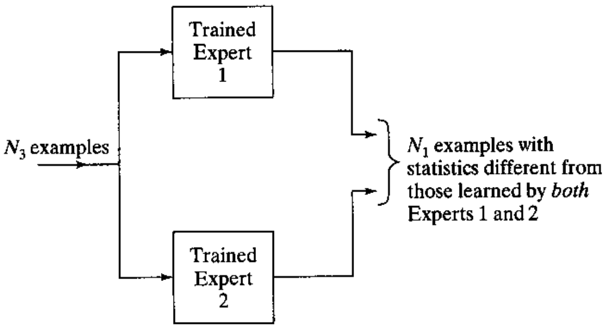

N_2 examples
Trained Expert 1
N_1 examples with statistics *different* from those learned by Expert 1

Fig. 2a.Filtering of data by Expert 1


N_3 examples
Trained Expert 1
Trained Expert 2
N_1 examples with statistics different from those learned by *both* Experts 1 and 2

b. Filtering of data by Expert 2 and 3

Fig. 2.The three-point filtering procedure

With N_1 examples also needed to train the first expert, the total size of data set needed to train the entire committee machine is $N = N_1 + N_2 + N_3$. However, the computational cost is based on 3N1 examples because N1 is the number of examples actually used to train each of the three experts. We may therefore say that the boosting algorithm described herein is indeed "smart" in the sense that the committee machine requires a large set of examples for its operation, but only a subset of that data set is used to perform the actual training.

Table 2. Number of samples used at each stage of the training-testing processes.

	N_1	N_2	N_3	Test
Sick	147	52	42	50
Healthy	48	48	70	50

Another noteworthy point is that the filtering operation performed by the first expert and the joint filtering operation performed by the first and second experts make the second and third experts, respectively, focus on "hard-to-learn" parts of the distribution.

During the training stage, the performances of committee members, are shown in Table 3.

Table 3. Positives, and Negatives in training stage.

	First	Second	Third
True positive %	83.67	42.31	4107
True negative %	47.92	79.17	51.55
False positive %	52.18	20.83	48.45
False negative %	16.33	57.69	58.93

In the theoretical derivation of the boosting algorithm originally presented in Schapire (1990)[25], simple voting was used to evaluate the performance of the committee machine on test patterns not seen before. Specifically, a test pattern is presented to the committee machine. If the first and second experts in the committee machine agree in their respective decisions, that class label is used. Otherwise, the class label discovered by the third expert is used. However, in experimental work presented in Drucker et al.[27-28] (1993,1994), it has been determined that addition of the respective outputs of the three experts yields a better performance than voting. For example, in the optical character recognition (OCR) problem, the addition operation is performed simply by adding the "digit 0" outputs of the three experts, and likewise for the other nine digit outputs.

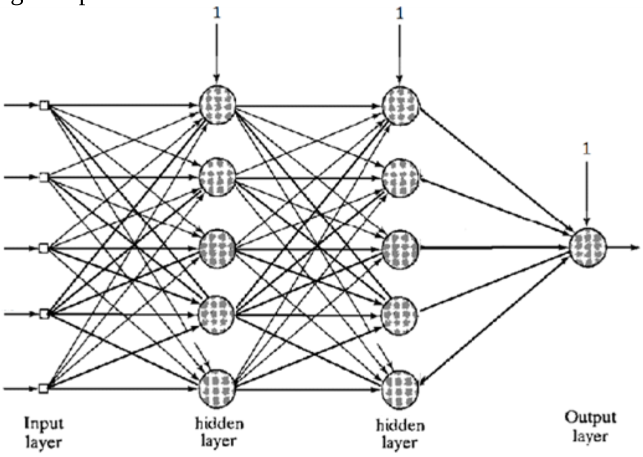


Fig. 3. Signal flow graph of each of the three expert machines with two hidden layers.

The number of input terminals equaled the number of attributes in the human voice data, thus it is eleven. There are two hidden layers with eleven neurons within each of three neural networks in the committee machine for preserving generalization properties but achieving convergence during training with tolerance at most 0.14 for all training samples recognized properly.

For all structures of artificial neural networks, only one output is produced. Actually, it was possible to use a single output and by interpretation of its active state as one class and inactive output state the second class the task would have been solved as well, but with such approach the text is attributed to either one or another author and classification is binary. Algorithm results in a decision about attribution of paragraphs whose textual description entered as inputs.

4. RESULTS AND DISCUSSION

To perform the boosting by filtering technique, we the training data are chosen in a special way described in Section 3.5. A balanced set of 50-50 positive and negative members are chosen from available data for testing. During the testing stage, the performances of committee members and success in the final decision are shown in Table 4.

Table 4. False positives, and false negatives in testing stage.

	First	Second	Third	Majority
True positive %	84	56	70	74
True negative %	54	78	74	74
False positive %	46	22	26	26
False negative %	16	44	30	26

It has been shown that parallel neural networks, when boosted by filtering, in combination with a majority voting increase performance of true recognition rates in an imbalanced data set.

The data set is very unbalanced with regard to the class distribution. This, in combination with the small sample size, makes it difficult to train any type of classifier to predict the presence of Parkinson's disease.

Out of 195 samples, 75.4% are Parkinson's disease type and the remainder is of healthy character. Although it is seen that, adding multiple copies of the samples in the smaller population helps balancing population imbalance, a common problem with imbalanced data sets is that they can increase to high false positive rates.

Traditionally, the problem with false positive predictions is dealt with over- or undersampling [22]. However techniques to adjust the sample distribution sometimes overweight the benefits of generalising the classifier. Any modification to the data set is merely artificial alternatives to the problem of inadequate training data. In this paper, it has been demonstrated that parallel neural networks are strong at adjusting the imbalanced data set problem.

False positive rates up to 25 - 30% of the positive class have been reported [29] in the literature. It has been demonstrated in this study that a true positive rate up to 74% of each class can be achieved by using three parallel networks. This is a significant improvement compared to previously demonstrated results

5. CONCLUSIONS

A system has been presented consisting of parallel distributed neural networks with two hidden layers, boosted by the use of filtering, and a majority voting system. The different expertise of the committee members increases the robustness of the system. An empirical investigation demonstrates that it is possible to achieve >90% true positive rate for each class in a Parkinson's disease data set with class distribution of 3:1 ratio.

REFERENCES

[1] R. Das, A comparison of multiple classification methods for diagnosis of Parkinson disease, *Expert Systems with Applications* 37 (2) (2010) 1568 – 1572.

[2] D.M. R, J. J. L, *Harrison's Principles of Internal Medicine*, 17th Edition, The McGraw-Hill Company, Inc, http://www.accessmedicine.com/content.aspx? aID=2905868, 2009, Ch. 366. Parkinson's Disease and Other Extrapyramidal Movement Disorders.

[3] A. H. Ropper, M. A. Samuels, *Adams and Victor's Principles of Neurology*, 9th Edition, The McGraw-Hill Companies, Inc, http://www.accessmedicine.com/content.aspx? aID =3633872, 2009, Ch. 23. Disorders of Speech and Language.

[4] L. A. da Silva, M. B. Joaquim, Noise reduction in biomedical speech signal processing based on time and frequency Kalman filtering combined with spectral subtraction, *Computers and Electrical Engineering* 34 (2008) 154 – 164.

[5] Q. Yan, S. Vaseghi, E. Zavarehei, B. Milner, J. Darch, P. White, I. Andrianakis, Kalman tracking of linear predictor and harmonic noise

models for noisy speech enhancement, *Computer Speech and Language* 22 (1) (2008) 69 – 83.

[6] J. J. Sroka, L. D. Braida, Human and machine consonant recognition, *Speech Communication* 45 (4) (2005) 401 – 423.

[7] M. D. Skowronski, J. G. Harris, Applied principles of clear and Lombard speech for automated intelligibility enhancement in noisy environments, *Speech Communication* 48 (5) (2006) 549 – 558.

[8] A. Esposito, M. Marinaro, *Nonlinear Speech Modeling and Applications*, Vol. 3445 of Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 2005, Ch. Some Notes on Nonlinearities of Speech, pp. 1–14.

[9] A. Hussain, M. Chetouani, S. Squartini, A. Bastari, F. Piazza, *Progress in Nonlinear Speech Processing*, Vol. 4391, Springer Berlin / Heidelberg, 2007, Ch. Nonlinear Speech Enhancement: An Overview, pp. 217–248.

[10] J. McDonough, K. Kumatani, T. Gehrig, E. Stoimenov, U. Mayer, S. Schacht, M. Wölfel, D. Klakow, *Machine Learning for Multimodal Interaction*, Vol. 4892/2008, Springer-Verlag Berlin Heidenberg, 2007, Ch. To Separete Speech, A System For Recognizing Simultaneous Speech, pp. 283 – 294.

[11] M. J. F. Gales, *Model-based techniques for noise robust speech recognition*, Ph.D. thesis, Gonville and Caius College (September 1995).

[12] M. A. Little, P. E. McSharry, E. J. Hunter, L. O. Ramig, Suitability of dysphonia measurements for telemonitoring of parkinson's disease, *IEEE Transactions on Biomedical Engineering* 56 (4) (2009) 1015–1022.

[13] F. Aström, R. Koker, A case study of parallel neural networks,

[14] M. S. Lee, J.-K. Rhee, B.-H. Kim, B.-T. Zhang, *Aesnb: Active example selection with naive Bayes classifier or learning from imbalanced biomedical data*, 2009 Ninth IEEE International Conference on Bioinformatics and Bioengineering (2009) 15–21.

[15] M. L. Minsky, and S. A. Papert, (1988). *Perceptrons,* Expanded Edition. Cambridge, MA: MIT Press. Original edition, 1969.

[16] R. Koker, Reliability-based approach to the inverse kinematics solution of robots using Elman's networks, *Engineering Applications of Artificial Intelligence* (18) (2008) 685 – 693.

[17] B. J. Lee, Parallel neural networks for speech recognition, *Neural Networks,*1997., International Conference on 4 (9-12) (1997) 2093–2097.

[18] M. A. Mazurowskia, P. A. Habasa, J. M. Zuradaa, J. Y. Lob, J. A. Bakerb, G. D. Tourassib, *Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance*, *Advances in Neural Networks Research*: IJCNN '07, 2007 International Joint Conference on Neural Networks IJCNN '07 21 (2-3) (2008) 427–436.

[19] M. A. Little, P. E. McSharry, S. J. Roberts, D. A. Costello, I.M. Moroz, Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection, *BioMedical Engineering OnLine* 2007 6 (23).

[20] W. S. Mcculloch, and W.Pill's. (1943). "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics*, 5:115-133. Reprinted in Anderson& Rosenfeld [1988], pp. 18-28.

[21] D. O. Hebb, (1949). *The Organization of Behavior.*New York: John Wiley & Sons. Introduction and Chapter 4 reprinted in Anderson & Rosenfeld [1988], pp. 45-56.

[21] Rosenblatt, E, 1958.The Perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, vol. 65, pp. 386-408.

[22] H. Tang, K. C. Tan, and Z. Yi, *Neural Networks: Computational Models and Applications*, Springer-Verlag Berlin Heidelberg 2007.

[23] N. J. Nilsson, *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*, New York: Mcffraw-Hill, 1965.

[24] S. Haykin, *Neural Networks A Comprehensive Foundation*, Second Edition, Prentice-Hall, Inc., Simon & Schuster,A Viacom Company Upper Saddle River, New Jersey 07458, 1999.

[25] R. E. Schapire, R.E, The strength of weak learnability, *Machine Learning*, vol. 5, pp.197-227, 1990.

[26] R. E. Schapire, 1997. Using output codes to boost multiclass learning problems, Machine Learning: *Proceedings of the Fourteenth International Conference*, Nashville, TN.

[27] H. Drucker, C. Cortes, L.D. Jackel, and Y. LeCun, Boosting and other ensemble methods. Neural Computation, vol. 6, 1994, pp.1289-1301

[28]H. Drucker, R.E. Schapire, and P. Simard, Improving performance in neural networks using a boosting algorithm, *Advances in Neural Information Processing Systems*, vol. 5, 1993, pp. 42--49, Cambridge, MA: MIT Press.

[29] D. O. Hebb, *The Organization of Behavior*. New York: John Wiley & Sons, 1949. Introduction and Chapter 4 reprinted in Anderson & Rosenfeld, 1988, pp. 45-56.