Pattern Classification using Quadratic Neuron: An Experimental Study

by

Ganesh Yaparla, Rhishi Pratap Singh, Garimella Ramamurthy

in

8th International Conference on Computing Communication and Networking Technologies 2017

Report No: IIIT/TR/2017/-1

Centre for Communications
International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2017

Pattern Classification using Quadratic Neuron: An Experimental Study

Yaparla Ganesh and Rhishi Pratap Singh
Signal Processing and Communication Research Center
International Institute of Information Technology,
Hyderabad, India
Email: ganesh.yaparla, rhishi.pratap@research.iiit.ac.in

Garimella Rama Murthy
Signal Processing and Communication Research Center
International Institute of Information Technology,
Hyderabad, India
Email: rammurthy@iiit.ac.in

***Abstract*—In this paper, we present a study done on quadratic neurons to solve the pattern classification problems. The paper compares the classification results obtained by quadratic neural network(QUAD) with normal single and multilayer perceptron(MLP). Examples with randomly generated toy datasets are used for understanding and visualization. The standard datasets such as Iris, MNIST and others are used for extensive comparison and interesting experiments. Different architectures of QUAD neural net has also been tested. Obtained results are better in comparison to conventional multilayer perceptron. This experimental study motivates the authors to study usage of QUAD neurons in deep neural networks.**

***Index Terms*—Multilayer perceptrons, Multi-layer neural network, Artificial neural networks**

I. Introduction

The class of problems dealing with the classification of data, which are not linearly separable, are called hard problems. To solve such hard problems, a multilayer feed forward neural network with non-linear processing units and hidden layers is used. It is known that every hidden layer in MLP extracts some higher order correlations of previous layers. Deeper the layer in the network, more specific are the features that it extracts. The idea of achieving convex like non-linear decision surfaces at hidden layers in MLP motivates us to use QUAD neurons in classification problems. It is also clear that we can never achieve exactly smooth decision surfaces in MLP which we can, using QUAD neural networks. Hence, it is makes sense to use QUAD neurons in neural networks rather than a big stack of perceptrons in some arrangement doing the same job.

The second order generic equation in two dimensional space is $Ax^2 + By^2 + Cxy + Dx + Ey + F = 0$, if parameters A, B, C, D, E, F are learnt, then complex decision surfaces can be formed with ease. In this paper we have done classification using quadratic neurons (QUAD), which learns these parameters and generates second order non-linear decision boundaries. Single QUAD neuron can be referred as a perception with second order encoded input features.

The initial work in this direction was done by [1] and [2]. The work shows the capabilities of QUAD neural net junction. But the idea was not clearly conveyed to use this in classification problems. More work [3], [4] is done towards the clustering using the QUAD.

The rest of the paper is divided into five sections. In next section, single quadratic neural network is discussed. It is shown that how a single neuron can produce non-linear decision surfaces which can classify linearly separable and linearly non-separable classes. Next the capabilities of QUAD are extended and neural net with single layer is discussed. Different neural network topologies and their results with standard datasets such as Iris and MNIST [5] are compared. In further sections, the capability of QUAD neuron is examined with multilayer QUAD network, PCA and hybrid neural architectures.

II. Single Quadratic Neuron Model

Fig.1 shows a quadratic neuron. For two features x, y

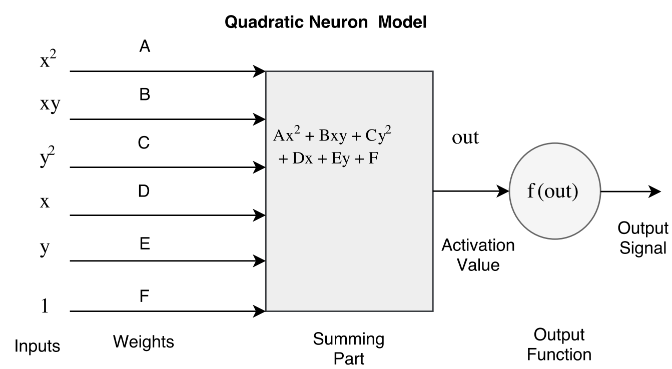

Fig. 1: Generic model of quadratic neuron with 2 features

The following equations describe the operation of the quadratic neuron model

$$\text{Activation: } out = Ax^2 + Bxy + Cy^2 + Dx + Ey + F \quad (1)$$

$$\text{Output Signal: } f(out) \quad (2)$$

f can be any activation function such as heaviside step function, hyperbolic tanget, etc.

$$\text{Error: } \delta = b - f(out) \quad (3)$$

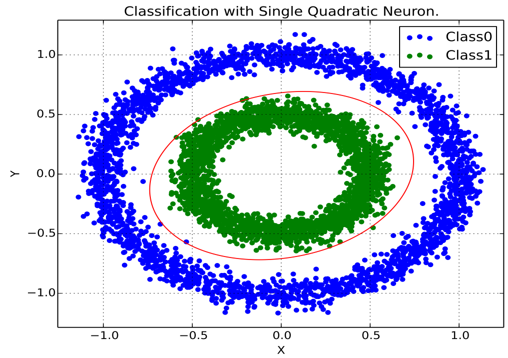

Fig. 2: Concentric separable classes

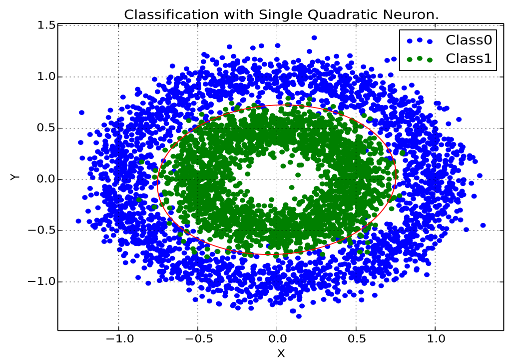

Fig. 3: Concentric separable classes

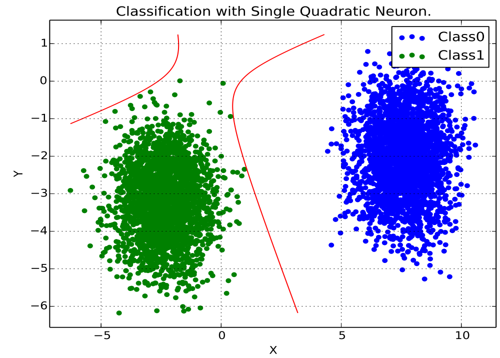

Fig. 4: Concentric separable classes

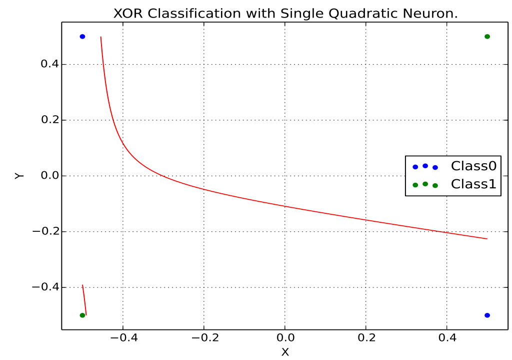

Fig. 5: Concentric separable classes

$$\text{Weight Update: } \Delta w(n+1) = \Delta w(n) - \eta \delta a(n) \quad (4)$$

where $a(n)$ is the input vector at n^{th} iteration. b is the desired output. we randomly initialize weights i.e A, B, C, D, E, F and train neural network using back propagation(same as perceptron learning), hence it follows same convergence law. If the classes are second order separable the learning law converges to final set of weights in finite number of steps. It holds true for linear separable classes as well because linear separability also holds elliptical or in general second order separability.

The implementation of the quadratic neuron is done in python. The datasets are used from the python-sklearn package [6]. Testing of MLP procedures are done using functions available in python-sklearn. In this section, test cases for MLP are run with 1 hidden layer and 10 neurons. There are 2000 data points in total. The training and testing datasets have been split in 70:30. Initial learning rate is kept at .01. These values are the same for entire section.

A. Toy data set test cases

First test case has two elliptically separable classes which are concentric. These classes are not linearly separable. In case of MLP, minimum of 13 hidden neurons were required to obtain accuracy of 100%, while only a single QUAD obtains 100% accuracy. Fig.[2] shows these two classes and the separating boundary learnt by single QUAD neuron.

Second test case is same as the first one but the data points are not separable elliptically. It is hard to draw any decision boundary. As the classes are not linearly separable the learning for QUAD continues till the highest accuracy is achieved for training dataset. QUAD neuron creates an elliptical boundary for the classification as per fig.[3].

Two linearly separable blobs have been taken in third test case. This test case emphasizes the point that if two classes are linearly separable, those are second order separable as well. Result in fig.[4] shows that a hyperbolic surface has been created by QUAD neuron.

XOR is another standard problem posed for nonlinear classifiers. QUAD neuron classifies both the classes using a hyperbolic surface. The decision boundary can be seen in fig.[5].

Table[I] shows the comparison of results with MLP. It is clear that, the performance of QUAD neuron is very competitive.

III. Single layer Quadratic Neural Net

From previous section, it is clear that a single QUAD neuron is capable of classifying two classes which are second order separable. For multi-class classification, a layer of such

Classifier	Accuracy
MLP	100%
QUAD	100%

(a) Elliptically and linearly separable classes

Classifier	Accuracy
MLP	97.9%
QUAD	97.5%

(b) Elliptically non-separable classes

TABLE I: Accuracy comparison of MLP with single quadratic neuron classifier.

neurons would correctly classify the data points. Fig.6 shows the architecture of such a neural network.

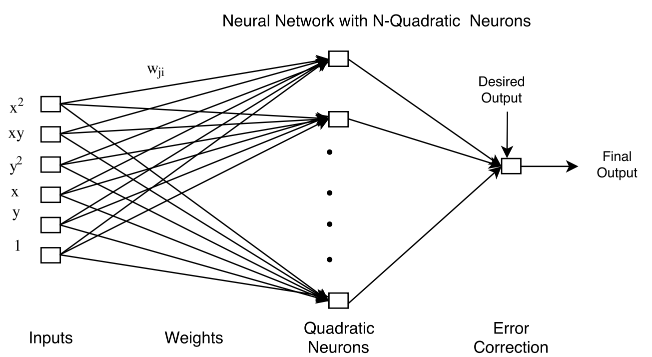

Fig. 6: Single layer QUAD neural network

The learning rule or the weight update rule is same as the previous section. The only difference is every QUAD neuron in the first layer acts as one vs rest classifier. For MLP, there are changes in number of hidden layers and number of hidden neurons in a layer. These specifications are mentioned in the test cases.

A. Test Case: multiple blobs case

There are 5000 data points in total. Again, train and test data sets have a ratio of 70:30. Each blob represents a class. Using 4 and 7 QUAD neurons, the classification of four class and seven class problems is done. From fig.[7] and fig.[8], it is visible that single layer QUAD neural network is creating second order decision surfaces for classification of different classes.

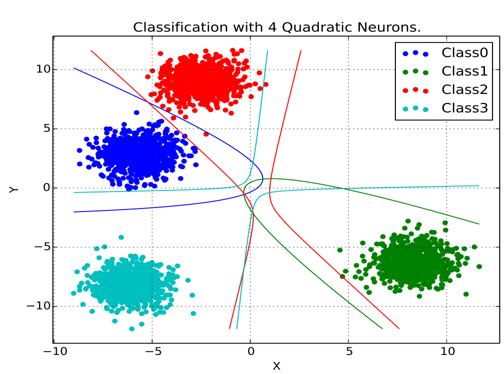

Fig. 7: Classification using 4 QUAD neurons.

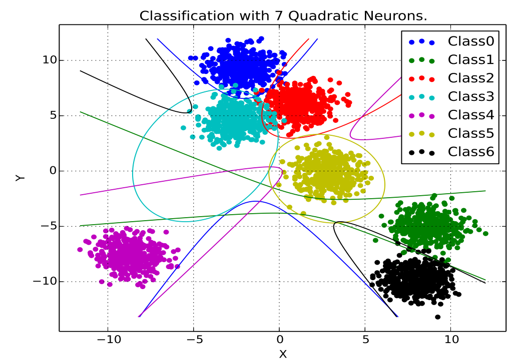

Fig. 8: Classification using 7 QUAD neurons.

Classifier	Accuracy
MLP	100%
QUAD	99.83%

(a) 4 blobs case

Classifier	Accuracy
MLP	97.13%
QUAD	96.83%

(b) 7 blobs case

Classifier	Accuracy
MLP	100%
QUAD	100%

(c) Iris datset with 2 features

Classifier	Accuracy
MLP	100%
QUAD	100%

(d) Iris datset with all features

TABLE II: Accuracy comparison of MLP with single layer QUAD neuron classifier.

For solving the same problem with MLP, 2 layers of hidden neurons are used. Both first and second hidden layers have 10 neurons. If changes are made in number of neurons and number of hidden layers, the accuracy varies. Hence, it signifies that if patterns are second order separable, it is required to find optimal set of hidden neurons and layers in case of MLP. While, required QUAD neurons for classification are total number of output classes. This signifies the simplicity involved while training the neural net with QUAD neurons.

B. Test Case: Iris Dataset

Iris data set has three classes Setosa, Versicolour, and Virginica, with 4 features sepal length, sepal width, petal length and petal width. There are total 150 data points. For each class there are equal amount of data points.

Iris dataset is tested in two ways. First, only sepal length and petal width features are chosen for classification. The reason for the selection is that, with these features dataset becomes almost linearly separable. As there are three classes, a QUAD neural network with 3 neurons classifies the test set accurately. The result is shown in fig.[9].

Next the original dataset with all the features is tested. Again a QUAD neural network with 3 neurons is used. Table[IIc] and [IId] shows the comparison of accuracies with MLP.

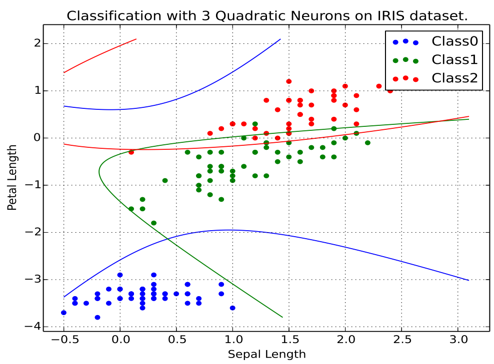

Fig. 9: Classification using 3 QUAD neurons on Iris dataset with two features.

IV. Neural Net with Hidden quadratic neurons

Given a multi class classification problem, in general not all classes will be QUAD separable, so we need more than one QUAD neuron to classify a single class. Multilayer QUAD neural network has been used to solve such problems. Fig.[10] shows the architecture of such a neural network.

Dimensionality is a curse here. If the input space is very high dimensional, then corresponding inputs for QUAD neurons increase exponentially. For high dimensional dataset such as MNIST, the requirements for processing and memory increases heavily. Dimensionality reduction techniques should be used while preprocessing data. Principal component analysis(PCA) and auto encoders are the common dimensionality reduction techniques which give good results in our case.

MNIST dataset has been used for testing the neural net with hidden quadratic neurons. First, the dimensionality is reduced using PCA. Original MNIST dataset has 784 dimensional and using PCA it is reduced to 50. Then, the inputs are given to QUAD neuronal net as per fig.[10]. Stochastic gradient descent has been used to train the network with constant learning rate of 0.08. Early stopping has been used to avoid overfitting. The test is repeated with multiple hidden neurons and the results are displayed in table[IIIa]. Tests are also run with 2 hidden layers, first hidden layer has QUAD neurons, while next hidden layer is composed on linear neurons. The test is done with multiple combinations of hidden neurons and the results are in table[IIIb].

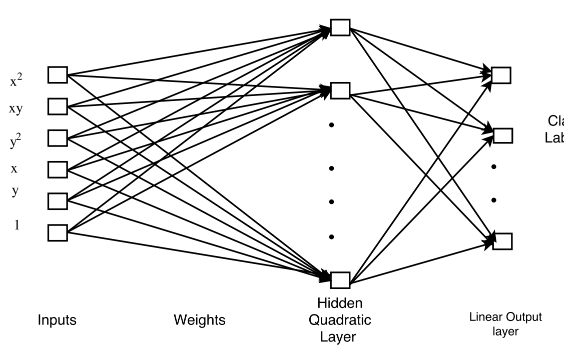

Fig. 10: Architecture with cross correlation terms

No. Hidden Neurons	Accuracy
200	98.38%
300	98.33%
400	98.37%
500	98.38%

(a) Neural net with 1 hidden layer

No. Hidden Neurons	Accuracy
200, 125	98.30%
300, 125	98.32%
400, 125	98.46%
500, 125	98.40%

(b) Neural net with 2 hidden layers

TABLE III: Accuracy with hidden QUAD neuron layer and PCA

V. Neural Net with Hidden quadratic neurons and no cross correlations terms

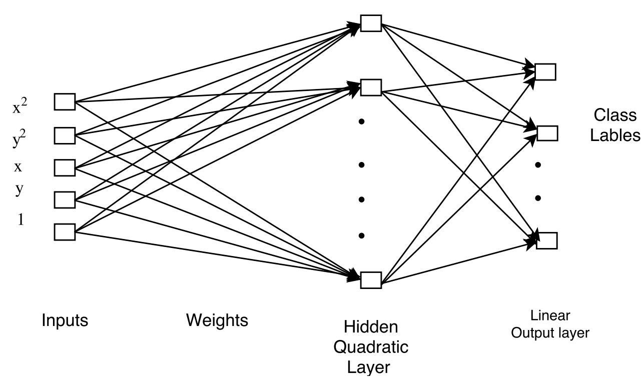

Fig. 11: Architecture without cross correlation terms

Given some decision boundary we can always come up with a set of axes parallel second order curves which can encompass the whole region with some error rate, which depends upon number of second order curves used. If the number of attributes of a classification problem are N, the Fan-In to every QUAD is N(direct terms) + $\binom{N}{2}$ (cross correlated terms) + N (self correlations). Fan-In of the quad neuron will increase in the order of N^2, which is because of the cross correlated terms.

If N is very large it doesn't seem reasonable to train such network, So we omit cross correlated terms and increase the number of QUAD neurons in hidden layer , which increases fan-In of network in order of N and does reasonably well in classification. Architecture of such a network is in fig.[11].

This case has also been tested with MNIST dataset in two categories. Tests are run with 1 and 2 hidden layers, with and without PCA as preprocessing. PCA is used to decrease the dimensionality of feature space from 784 to 50. Stochastic gradient descent has been used to train the network with constant learning rate of 0.08. Early stopping has been used to avoid overfitting.The inputs are given without cross-correlated terms as per fig.[11]. The results are shown in table[IV]. In tables[IVb,d], it is emphasized that even after converting the data to very low dimensionality, we were able to obtain very good results on the test data set. As PCA transforms the data mostly along the axes, hence we can omit cross correlation terms at the first hidden layer which barely deceases the accuracy on test set.

No. Hidden Neurons	Accuracy
200	98.27%
300	98.32%
400	98.33%
500	98.30%

(a) Neural net with 1 hidden layer

No. Hidden Neurons	Accuracy
200	98.08%
300	98.31%
400	98.34%
500	98.37%

(b) Neural net with 1 hidden layer and PCA

No. Hidden Neurons	Accuracy
200, 125	98.17%
300, 125	98.39%
400, 125	98.44%
500, 125	98.60%

(c) Neural net with 2 hidden layers

No. Hidden Neurons	Accuracy
200, 125	98.26%
300, 125	98.43%
400, 125	98.38%
500, 125	98.51%

(d) Neural net with 2 hidden layers and PCA

TABLE IV: Accuracy with hidden QUAD neurons and no cross co-relation terms.

The results in tables[IV] emphasize that good results have been achieved over MNIST dataset with less number of neurons in hidden layers. Most of the job that is being done in the second Hidden layer in MLP is being done in the first layer of QUAD neural network itself. so, with less number of neurons classification is possible.

The other dataset that has been used is human activity recognition with smart phones [7]. There are six classes walking, walking_upstairs, walking_downstairs, sitting, standing, laying with 561 features. The points in training and testing datasets are in 70:30 ratio.

From these experiments, it is clear that there is comparatively less overfitting in QUAD neural network than MLP. Results show that after achieving maximum accuracy with some configuration in case of MLP, if the number of hidden neurons in first layer are increased, the accuracy of the test set decreases more steeper in case of MLP than QUAD network. One important observation is that QUAD neural net converges faster than MLP. Refer fig.[12] and fig.[13]. Maximum claimed accuracy on MNSIT with MLP and no distortions is 98.47% [5] and with QUAD neural net the max obtained accuracy is 98.60%. Maximum claimed accuracy on human activity recognition with smart phones datset is 95.2% and with QUAD neural net we are able to achieve 96.4%.

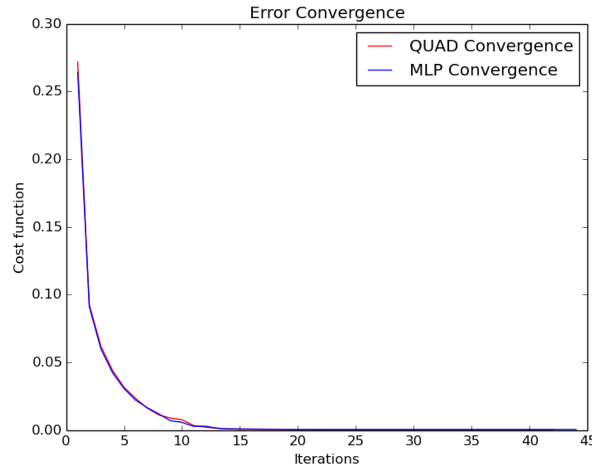

Fig. 12: Convergence comparison with MLP for MNIST dataset.

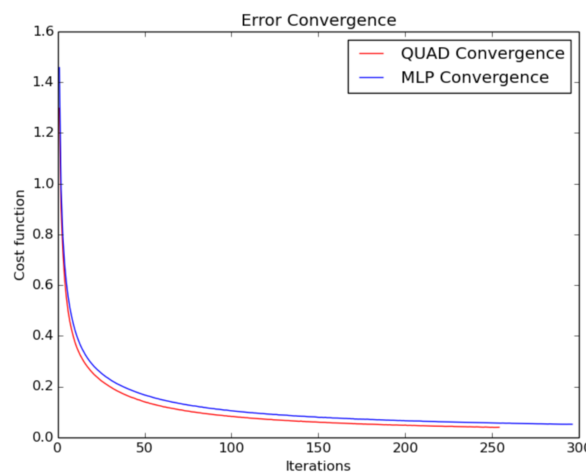

Fig. 13: Convergence comparison with MLP for phone activity dataset.

VI. Hybrid Neural Net with Hidden quadratic and linear neurons

Ideally we expect our decision surfaces to be aligned with shape of data. It should neither overfit nor underfit. If the data points are linearly separable in some neighborhood, it makes sense to classify it using a straight line rather than a second order curve. It gives the motivation to use linear neurons along with QUAD neurons in hidden layer to achieve good performance over test data. The architecture of such neural network is shown in fig.[14].

There are many advantages of using hybrid architecture. It is more generalized, it converges at a faster rate compared to other architectures, it requires less number of neurons for classification and comparatively less number of hidden layers

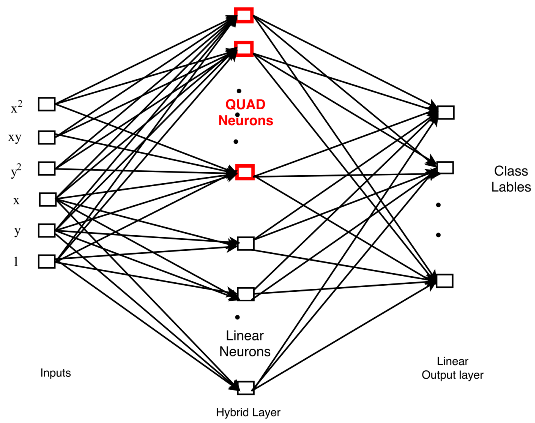

Fig. 14: Proposed Architecture with combination of QUAD neurons and linear neurons in hidden layer.

We got an average accuracy of 98.68% on MNIST test dataset using 200 linear neurons, 500 QUAD neurons in first hidden layer and 250 neurons in second hidden layer. We have implemented this in keras[8] a deep learning library. Accuracy can be further increased by hyperparameter optimization techniques.

VII. Future Work and Conclusion

In this research paper we have presented the experimental study done on QUAD neural network. For better understanding, the visualization of separating hyperplanes for two dimension classes is shown. We have achieved more accuracy on MNSIT dataset than standard MLP based implementations. It is clear that with less number of neurons, QUAD neural network is capable of achieving better accuracy against MLP. Usage of QUAD neurons in clustering needs to be exploited by regularizing the cost function. This trick makes sure that QUAD neural net always learns closed curves.

References

[1] Nicholas DeClaris and Mu-chun Su. A novel class of neural networks with quadratic junctions. In *Systems, Man, and Cybernetics, 1991.'Decision Aiding for Complex Systems, Conference Proceedings., 1991 IEEE International Conference on*, pages 1557–1562. IEEE, 1991.

[2] Nicholas DeClaris and M-C Su. Introduction to the theory and applications of neural networks with quadratic junctions. In *Systems, Man and Cybernetics, 1992., IEEE International Conference on*, pages 1320–1325. IEEE, 1992.

[3] Mu-Chun Su and Ta-Kang Liu. Application of neural networks using quadratic junctions in cluster analysis. *Neurocomputing*, 37(1):165–175, 2001.

[4] Mu-Chun Su, Nicholas DeClaris, and Ta-Kang Liu. Application of neural networks in cluster analysis. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, volume 1, pages 1–6. IEEE, 1997.

[5] Yann Lecun and Corinna Cortes. The MNIST database of handwritten digits.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[7] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *ESANN*, 2013.

[8] François Chollet. Keras. https://github.com/fchollet/keras, 2015.