Using Casper algorithm detect real angry emotion

Hao Wen
Research School of Computer Science
Australian National University
Acton ACT 2601 Australia
U5883475@anu.edu.au

Abstract. Detecting the actual perception of the emotion expressed by human face is important and it can effectively improve the diagnostic of mental health. In this study, I used pupillary response patterns as feature variable to classify individual's real and posed anger. The algorithm is Casper network algorithm employing RPROP and SARPROP backpropagation. Also, this paper examined effect of activation function on performance of Casper algorithm. Four main results were found: 1. Casper network has better performance in prediction accuracy and more compact network than simple neural network. 2. There is no significant difference between RPROP and SARPROP in this problem. 3. Sigmoid activation function has higher accuracy than Relu activation function. 4. Casper algorithm has similar accuracy with machine classifier in the published research paper.

1. Introduction

Emotions are an important aspect of human life. People recognize personal emotions by facial expression and tell the individual's mental health as well as social interaction. One of the emotions we experienced frequently in daily life is anger. Anger has valuable qualities and can be beneficial to the human condition. Thus it is plausible that real anger is more useful to reflect mental health. However, acted anger expression without genuine feeling attempts to mislead the perceives. So how to successfully classifying acted and genuine anger is becoming a popular topic. There have been a number of researches made to classifying posed and real emotion. Classifying posed and real smiles has been proved successfully using Pupillary Response(PR) features and classification accuracy was 93.7% with trained machine classifiers[1]. Another study shows the similar result when detecting real or posed anger by using same Pupillary Response(PR) features and machine classifiers[2]. The result from these two studies suggest pupillary response can be used to predict real and posed emotion and reach a higher accuracy than human verbal response.

Simple Neural Network has been a powerful tool for many classification problems. However, simple neural network usually converge slowly and needs a very large topology to achieve a good accuracy. A new network algorithm called Casper[3] was introduced and is shown to produce more compact networks and better performance on some classification problems. Casper is a constructive learning algorithm which builds cascade networks[5] and employs Progressive RPROP to train the whole network. Compare simple neural network, there are two main advantages of Casper algorithm. The first one is, like Cascade algorithm, neurons are added one at a time and are connected to all previous hidden and input neurons. Another advantage is Casper doesn't freeze weights and uses modified version of Resilient Back Propagation(RPROP) algorithm to train the whole network. RPROP[6] is a very different algorithm used to improve backpropagation. It only uses the sign of the gradient and assume the different weights need different step sizes for update, so it considerably accelerates backpropagation learning and can determine the appropriate step size by itself. While RPROP is fast on converging, it suffers from local minima problem. Another study[4] shows SARPROP algorithm, which is RPROP algorithm with Simulated Annealing term, can address local minima problem and increase the rate of convergence. The basic idea is adding noise allows the network to change descent direction and hence help network escape from local minima.

This study aims to investigate (1) if Casper algorithm classify genuine and posed anger successfully and has better result than other algorithm or benchmarks. (2) if Casper algorithm has more compact network than simple neural network. (3) Does SARPROP behaviour better than RPROP. In addition, this study examine the effect of activation function on classification performance.

2. Dataset Description and Pre-Processing

Dataset used in this paper called Anger that comes from a designed experiment[2]. There are 20 participants and each participant watch 20 videos including 10 genuine(True) anger scene and 10 acted (False) anger scene. Pupil size were tracked by a remote Eye Tribe eye gaze tracker. The dataset contains 400 rows and 9 columns. The first two columns are index of participants and video, the third to eighth columns are variables about pupil size. The last column is label which indicates the genuine anger or posed anger. The task is using the pupil size information to predict true or posed anger. Also the first two columns were discarded because it is index number and not helpful to predict. Thus, inputs are six variables including pupil response information and output is the last column that indicates genuine or posed anger. As there are only two values in response variable, so this is a classification problem.

The six input variables are all continuous variables that have different range. For example, Mean variable has range (0.5829, 0.9834) while Diff1 variable has range (0.0011, 0.044). The data need to be scaled into similar range which makes neural network work better. Output variable is categorical variable with two values "Genuine" and "Posed". "Genuine" is coded as 0 and "Posed" is 1.

3. Method

3.1 Casper algorithm

The Casper algorithm construct the network in a similar way to Cascor: starts with a single hidden neuron as initial network and successively adds single hidden neuron from candidate pool to the previous network each time. The new hidden neuron should connect all input neurons and previous hidden neurons. And then use backpropagation algorithm update different weights with different learning rates and train the whole network.

3.1.1 Candidate pool

Every time when adding a new hidden neuron, the new neuron come from a candidate pool which has 10 candidate hidden neurons. Each candidate neurons has different randomly initialized weights and was connected to current network. The candidate neurons that produces the lowest loss of whole network was selected as the best one.

3.1.2 Learning rate

There are three different learning rate for Casper algorithm as shown in Figure 1. Learning rate L1 is for region 1 where the weights connect all inputs and hidden neurons to new hidden neuron. Learning rate L2 is for region 2 where the weights connected from new hidden neuron to output neurons. Learning rate L3 is for region 3 where include all the old weights from previous network. Usually the relative value is L1>>L2>L3. The reason is high value of L1 allows the new hidden neuron to learn quickly the remaining network error. At the same time, L2 and L3 allows the other neurons to reduce the network loss as well but with little interference.

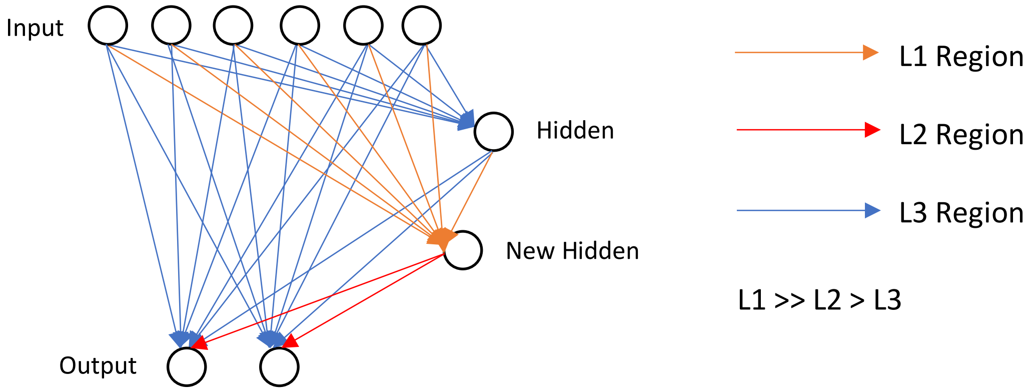

Figure 1. Casper algorithm

3.2 Backpropagation algorithm

3.2.1 RPROP algorithm

Casper use RPROP algorithm for network training. RPROP is a gradient descent algorithm only uses the sign of the gradient. Also, it assumes the different weights need different step size for updates, which vary throughout the process. The basic idea is if the error gradient for a given weight had the same sign in two consecutive epochs, we increase its step size because the optimal value may be far away. On the other hand, we decrease the step size if the sign switched. Finally update weights with step size. RPROP algorithm can eliminate the noisy effect of the size of gradient and avoid to get stuck with extreme weights because of shallow slope in the activation function.

3.2.2 SARPROP algorithm

Casper can also use SARPROP algorithm for network training. SARPROP algorithm is based on RPROP algorithm and makes use of weight decay (Simulated Annealing term) as a mean to increase the rate of convergence for some problem and avoid local minima. There are two main enhancements.

1. A noise factor is introduced. Noise is added to a weight when both the error gradient changes sign in successive epoch and the magnitude of the update value is less than a value proportional to the current loss. This will allow the weight to jump out of local minima. In the formula *error* corresponds to loss, *T* corresponds to the temperature factor, k_3 is constant, *r* is a random number between 0 and 1, *epoch* is the epoch number.

$$\Delta_{ij}(t) = \Delta_{ij}(t-1) * \eta^{-} + k_3 * r * error * 2^{-T*epoch}$$

2. Weight decay term to the error function. There are two different forms from different papers [3, 4]. The new error gradients are shown below where k_1 is constant number which effects the magnitude of weight decay, *T* is temperature factor, *epoch* is the epoch number, *sign* returns sign(positive or negative) of its operand, *HEpoch* is the number of epochs elapsed since the addition of the last hidden neuron:

$$\frac{\partial E}{\partial W_{ij}}^{SARPROP} = \frac{\partial E}{\partial W_{ij}} - k_1 * W_{ij} * 2^{-T*epoch}$$

$$\frac{\partial E}{\partial W_{ij}}^{SARPROP} = \frac{\partial E}{\partial W_{ij}} - k_1 * sign(W_{ij}) * {W_{ij}}^{2} * 2^{-0.01*HEpoch}$$

3.3 Activation function

Sigmoid function and Relu function has been widely used in neural network implementations. This study will mainly examine these two frequently used activation functions. Also can try other functions in the future study. Sigmoid function transfer the input value to range zero to one. Also for backpropagation process, it reduce the error by maximum a quarter in each layer. So as the network goes deeper, more knowledge from the data will be lost and vanishing gradient problem will appear . Relu function output zero if the input is less than zero, and raw output otherwise. So the derivative is just 1 and there is no squeezing effect. But one disadvantage is Relu loss the information from negative part of input value.

4. Model Design

The main idea of model design is adding one hidden unit each time to Casper network and simple neural network until it reaches a predefined number of hidden unit, and record the classification accuracy on testing data for each adding hidden unit. For example, the predefined number of hidden unit is 10. Add first hidden neuron to Casper and simple neural network, record the accuracy on testing data, and then adding the second hidden neuron and record the accuracy, keep adding until it adds to 10 hidden neurons. After that, find out at which hidden neuron the network achieve the highest accuracy. This hidden unit and accuracy will be regarded as the best hidden neuron and best accuracy. Repeat the whole process ten times and take average on best hidden neuron and best accuracy. The reason of predefined number is I found when adding hidden neurons, the prediction accuracy will become stable or lower at some points and then after that it will increase (Figure 2). If I stop the algorithm when accuracy of current hidden neurons doesn't improve compare to last hidden neuron, I will miss the latter hidden neuron probably with highest accuracy. So in this study, I set two predefined numbers that are 10 and 20, the results show the best number of hidden neuron are all below 10 for Casper.

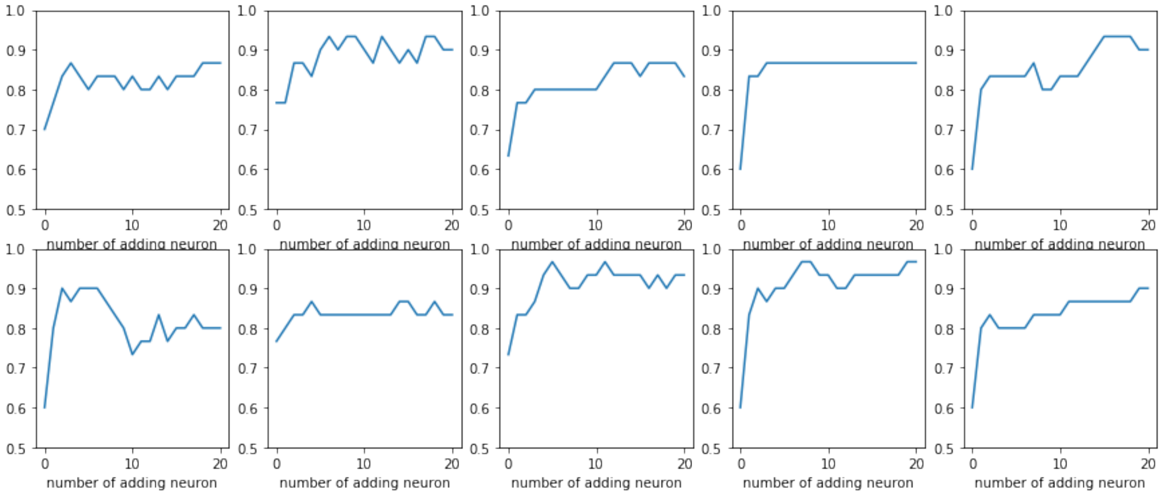

Figure 2. Test accuracy at hidden neurons on 10 repetitions

4.1 Model setting

Before running the whole model, I need to design the model and set conditions for different investigations. Totally, there are 16 different combinations and 8 for each algorithm as shown in below diagram (Figure 3).

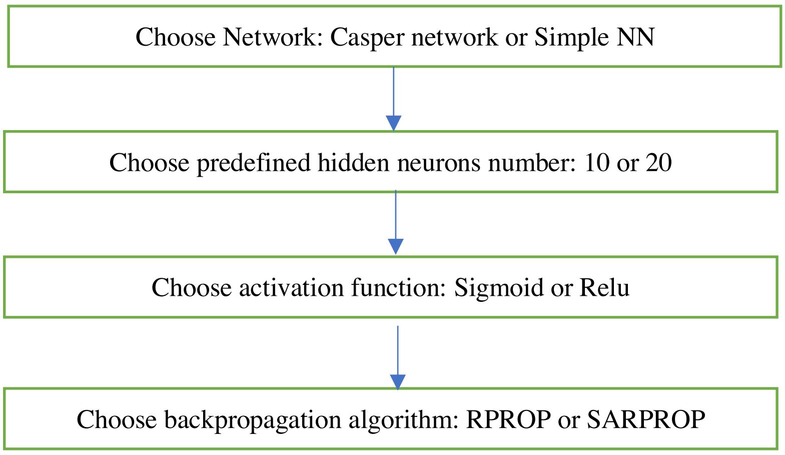
Choose Network: Casper network or Simple NN
Choose predefined hidden neurons number: 10 or 20
Choose activation function: Sigmoid or Relu
Choose backpropagation algorithm: RPROP or SARPROP

Figure 3. Model setting process

4.2 Selecting appropriate learning rate

Also, I need to select an appropriate learning rate for network in order to maximize their performance before running the whole model. There are 7 candidate learning rates: 0.0001,0.001,0.003,0.01,0.03,0.1 and 0.3. For convenience, I set L3 as base learning rate and set L1 = 200*L3, L2 = 5*L3. This is consistent with previous relative value where L1>>L2>L3. The selecting process is choose one learning rate from candidate as base learning rate(L3), set other conditions (activation function is sigmoid, predefined number is 10, backpropagation algorithm is RPROP) and keep it fixed. Run the whole model and select the learning rate with best performance. The determination of performance is in later section. Same process and conditions apply to simple neural network. The result are for Casper network, the best base learning rate is 0.003 and for simple neural network, the best learning rate is 0.001.

4.3 Procedures of whole model:

1. Initialize the network with one hidden unit. Save the parameter as old parameters.
2. Adding the new hidden unit to previous network using Casper algorithm. Randomly initialize the new weights and update the old parameter as well as new parameters together with different learning rates (L1, L2 and L3). Train the new network 100 epochs on training data.
3. Apply the trained network on validation data and save the parameters and loss value.
4. Repeat step 2 to 3 ten times and get a candidate pool with 10 candidate hidden neurons.
5. Choose the hidden unit with the lowest loss value as the best one and assign the parameters to new network.
6. Apply the new network on testing data and calculate the prediction(classification) accuracy.
7. Save the parameter as old parameters and repeat step 2 to 6 until it adds up to predefined number(10 or 20).
8. Repeat the whole process from step 1 to 7 ten times and save the best prediction accuracy and best hidden neuron for each repetition.
9. Average best number of hidden neurons and best accuracy on repetition. This is the result for this network algorithm.

The whole process for simple neural network is very similar and much simpler than Casper network. Simple neural network doesn't need candidate pool.

4.3 Determine the performance

There are two main factors determine the performance of network. The first one is convergence rate. This is measured by the average number of best hidden neurons. If the model reach the highest testing accuracy with less hidden neurons, which means the model converge quickly.

Another one is accuracy on testing data. The higher prediction accuracy is, the better the network is. Also, stand deviation of best hidden neurons and best testing accuracy is a good indicator to show the stable of algorithm.

5. Result and Discussion

Pre-processing data is randomly divided into training data(around 70%), validation data(around 15%) and testing data(around 15%). Also, each dataset is guaranteed to have balanced label which means the number of genuine anger and posed anger is equal in each dataset. This is beneficial to train the network and test performance on both labels. Finally, the training dataset size is 280, validation dataset size is 56 and testing dataset size is 64.

To test the effectiveness of the Casper algorithm, I compare its performance against simple neural network with one hidden layer, results from trained machine classifiers[1], human verbal response[1] and random guess accuracy 50%. Also, I compare the performance between Relu and sigmoid activation function, Rprop and SARprop algorithm. In Casper, a number of parameters require setting. The following (standard) parameters values were used for RPROP[3]: $\eta^{+} = 1.2, \eta^{-} = 0.5, \Delta_{\max} = 50, \Delta_{\min} = 1 * 10^{-6}$. $L1 = 0.6, L2 = 0.015, L3 = 0.003$. The following parameters values were used for SARPROP[4]: $k_1 = 0.01, k_2 = 0.1, k_3 = 3, T = 500, r = 0.5$ and other parameters are same as RPROP. Results are shown in Table 1 and Table 2.

Table 1

		Simple NN							
		Adding hidden units up to 10				Adding hidden units up to 20			
		Sigmoid		Relu		Sigmoid		Relu	
		RPROP	SARPROP	RPROP	SARPROP	RPROP	SARPROP	RPROP	SARPROP
Number of best hidden	average	7.2	7.1	6.9	7.0	13.1	11.9	16.0	15.4
	median	8.0	7.0	7.0	6.5	13.5	14.5	17.5	15.5
	std	2.2715	2.3	2.1189	1.9493	4.1581	5.7349	4.0	3.2311
Best accuracy on testing set	average	0.8533	0.85	0.8038	0.7865	0.91	0.8833	0.8538	0.8442
	median	0.8333	0.85	0.8077	0.7788	0.9	0.8666	0.8461	0.8365
	std	0.04	0.0401	0.0224	0.0303	0.03	0.0223	0.0196	0.025

Table 2

		Casper Algorithm							
		Adding hidden units up to 10				Adding hidden units up to 20			
		Sigmoid		Relu		Sigmoid		Relu	
		RPROP	SARPROP	RPROP	SARPROP	RPROP	SARPROP	RPROP	SARPROP
Number of best hidden	average	5.4	3.8	6.0	4.7	7.6	8.0	5.6	5.9
	median	5.5	3.0	5.5	4.5	5.5	6.0	5.5	4.0
	std	2.7640	2.4819	2.0493	1.7916	5.4808	5.2725	2.8705	5.3749
Best accuracy on testing set	average	0.8966	0.85	0.8596	0.8461	0.9066	0.94	0.8346	0.8365
	median	0.8833	0.8666	0.8653	0.8365	0.9	0.95	0.8461	0.8269
	std	0.0348	0.0654	0.0455	0.0516	0.0388	0.0442	0.0784	0.0414

5.1 Compare the performance between simple NN and Casper

From Table 1 and Table 2, Casper Algorithm always has smaller number of best hidden neurons in any situation. On average the number of best hidden neurons of Casper is 5.875 while simple neural network is 10.575. Figure 4 shows range of best hidden neurons for Casper is [2,11] and Simple NN is [8,20]. Apparently most best hidden neurons in Casper are smaller than simple NN. This result shows Casper algorithm indeed has more compact network than simple neural network and it converges quickly. Also, in terms of prediction accuracy on testing data, the simple NN has accuracy on average of all condition is 84.81%, and Casper Algorithm is 87.13%, which shows Casper has better prediction than simple NN. Also, Casper has the highest accuracy that is 94% among all the situations. The standard deviation for both hidden number and accuracy are relative small and there are no significant difference between two networks.

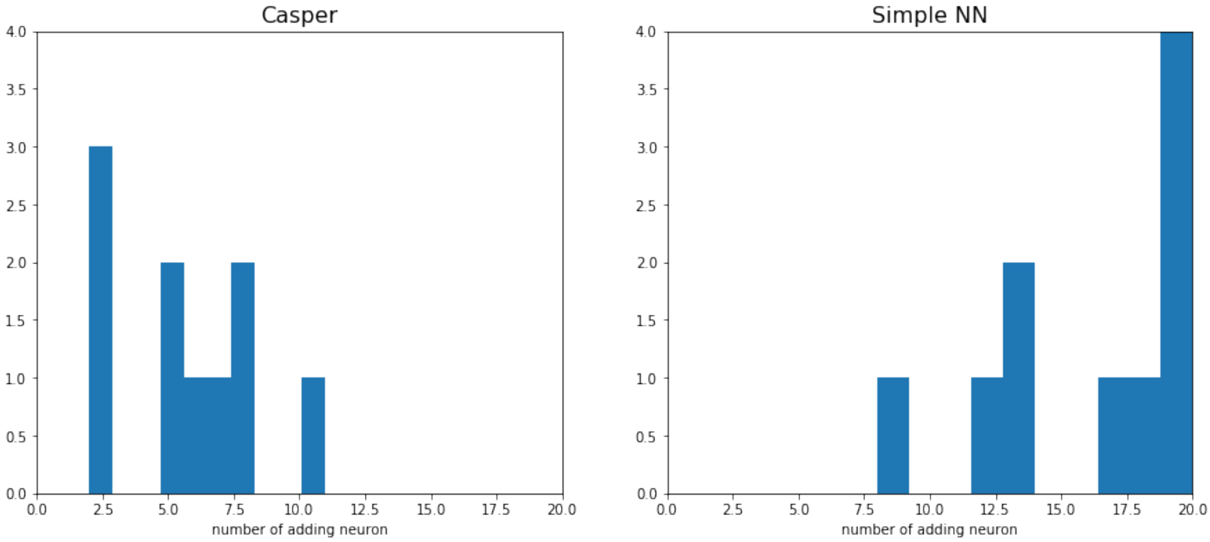

Figure 4. Best hidden unit histogram for Casper network and Simple neural network (predefined number is 20, RPROP, Relu)

5.2 Compare the performance between RPROP and SARPROP

For simple NN, there is no much difference in terms of number of best hidden neurons between RPROP and SARPROP. Also, the accuracy of RPROP is slightly bigger than SARPROP but it is not significant. For Casper algorithm, it is not always RPROP is better than SARPROP or SARPROP is better than RPEOP in terms of both number of hidden neurons and accuracy. So weight decay (Simulated Annealing) didn't have significant improvement for the algorithm in this case. Two possible reasons can explain this situation. First one is there are not many local optimas in loss function. So the advantage of Simulated Annealing is not obvious. The second reason is parameter tuning of Simulated Annealing. There are five more parameters(k1, k2, k3, T and r) and value from previous research[4] may not suitable to this problem. I should try hyperparameter tuning to maximize performance of Simulated Annealing.

5.3 Compare the performance between Relu and Sigmoid

In terms of number of hidden neurons, there is no consistent advantage of either Relu or Sigmoid activation function. But one interesting thing is Sigmoid always has higher accuracy than Relu. On average, the accuracy of Sigmoid is 88.62% while Relu only has 83.31%. One possible reason could be Relu function loss the information on the negative part of input value because it outputs zero if the input is less than 0.

5.4 Compare the performance of Casper with published research paper[2] and random guess accuracy 50%

Table 3

	Casper network	Machine classifiers	Verbal response	Random guess
Mean accuracy	94%	95%	60%	50%

I mainly compare mean accuracy because only mean accuracy is available on published research paper[2]. The table shows Casper network has significant higher accuracy than verbal response and baseline. Also, the accuracy between Casper and Machine classifiers is very close though Casper is slightly lower. This result suggests like other machine classifiers Casper is a powerful tool to classify the genuine and posed anger.

6. Conclusion and Future Work

In summary, this study showed that Casper network can classify genuine and posed anger by using pupillary response feature and reach a high accuracy. Also, Casper network is shown to produce networks which is more compact in terms of hidden neurons. Further, this study can be extended to detect other emotions such as smile and sad.

There are some limitations in this study that can be improved in future. Because of random initialisation, network is not stable and the result is not totally same every time. Variation of best hidden units and accuracy is large in some situation. In future, I would increase the repetition time to 100 that makes the result more stable. I would also try k-fold cross validation that avoid over fitting problem and obtain more stable results. Another limitations is learning rate and hyperparameters. This study fix learning rate once it is selected at the beginning. However, in order to achieve the best performance for each algorithm and situation, I would adjust learning rate and hyperparameters according to different situations.

References

1. Md Zakir Hossain, Tom Gedeon. Classifying posed and real smiles from observers' peripheral physiology. Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare. May 2017. Pages 460–463 (2017).
2. Lu Chen, Tom Gedeon, Md Zakir Hossain, Sabrina Caldwell Are you really angry? Detecting emotion veracity as a proposed tool for interaction. Proceedings of the 29th Australian Conference on Computer-Human Interaction November 2017 Pages 412–416 (2017).
3. Treadgold N.K., Gedeon T.D. A cascade network algorithm employing Progressive RPROP. In: Mira J., Moreno-Díaz R., Cabestany J. (eds) Biological and Artificial Computation: From Neuroscience to Technology. IWANN (1997).
4. Treadgold, N. & Gedeon, Tom. The SARPROP Algorithm: A Simulated Annealing Enhancement To Resilient Back Propagation (1997).
5. Fahlman, S.E., and Lebiere, C. The cascade-correlation learning architecture. In Advances in Neural Information Processing II, Touretzky, Ed. San Mateo, CA: Morgan Kauffman, 1990, pp. 524-532 (1990).
6. Riedmiller, M. and Braun, H. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: Ruspini, H., (Ed.) Proc. of the ICNN 93, San Francisco, pp. 586-591(1993).