Summary of the Article "*Neural Networks with Periodic and Monotonic Activation Functions: a Comparative Study in Classification Problems*" by J.M. Sopena, E. Romero and R. Alquézar

Hofbauer Heinz
hofbauer.heinz@gmx.net

Berka Tobias
tobias_berka@yahoo.com

Stütz Thomas
arot.ts@mailexcite.com

1 Introduction

The article discusses a number of reasons why the use of sine activation functions can lead to a significant improve in performance of multilayer feed-forward networks.

A classification system assigns a paricular class to each set of values of variables (the input space).
A learning algorithm leads a system from an initial state to a minimum of error. Backpropagation is most frequently used.

To improve performance of neural networks several modifications of backpropagation have been proposed, but none of these proposals have considered modifying the activation function. Recent experiments with periodic activation functions (sine) have shown that they are superior learning speed, cumputational capacity and generalization.

2 Characteristics of Periodic Activation Functions

The use of a periodic function as classifier divides the input space into an infinitive number of regions, opposed to the two regions created by monotonic activation functions.

Consider the formula

$$g(\vec{x}) = bias_o + \sum_{i=1}^{n_hid} \alpha_i sin(\vec{x}\vec{\omega}_i + bias_i)$$

as a representative sinusoid function. The weights ($\vec{\omega}_i$) are called non-linear parameters. The weights of the connections between layers (α_i) are called linear parameters. The biases can ba considered as weights with a simple transformation.

This allows approximation of more complex functions. Training a neural network of this kind is basically cunstructing a Fourier series, the only difference being the possibility of the network to adjust the values of the non-linear parameters.

Networks with sine functions have much more local minima in the error surface, in which the error correction algorithm easily can be trapped before reaching the global minimum, making the sufficient variation from the initial position of the weights difficult. This probability can be diminished by assigning an initial set of weights that is appropriate for the problem in question.

Non-linear parameters are difficult to adjust, thus they evolve much slower than linear parameters. To enhance the performance proved to be convenient to assign different values of learning raters for each layer.

3 Experiments

Learning speed is measured in the number of epochs required to learn the training set. The learninw was considered to be complete when the examples have been solved at a given tolerance. Partly an other criterion has been applied, when learning all examples was impossible or not convenient for generalization purposes. The architecture used was feed-forward with a layer of hidden units. The activation function was sinusoidal, the initial weights were assigned randomly to a particular interval.
Momentum was set to 0. Without a bias, the network can only produce odd-like functions. An approach to the conversion of the target function into an odd-like function is to rescale the variables of the input space between 0 an 1.

3.1 Two-Spirals Problem

The points of two interlocking spirals are to be identified. The training set comprises 194 points.

Standard BP with simple structure fails at this problem. The best algorithm ("Cascade Correlation") using standard BP with complex architecture needed 1,700 epochs to learn this problem. An algorithm using sine as the activation function in the hidden layer and hyperbolic tangent in the output layer only an average of 42.3 epochs. The range of the initial weights is important. With small ranges learning is not possible.

3.2 Negation N

Input patterns have N+1 bits. The last bit indicates whether the output is either a copy, or the 1-compliment.
With a 4-3-3 architecture using sigmoids 5,000 epochs were required. Using the same non- monotonic architecture as above, only 30 epochs were necessary.

3.3 Separation Problem

In this problem, like in the "two spirals" problem, the standard sigmoid function with BP is trapped in local minima. Using sines instead of sigmoids not only prevents the ANN from being trapped but also allowing it to learn around 180 times faster.

3.4 Quadrants

All points in quadrant one and three have target 1, the others have target 0. This problem seems to be well suited for sigmoid activation functions, but a 2-2-1 architecture with sine as activation function in the hidden layer and sigmoid in the output layer learns faster than sigmoid function network.

3.5 Diabetes

A comparison performed with a medical diagnosis problem. Sigmoid function are unable to learn more than 82% after 70,000 epochs, while sine functions learn more than 90% in 1,500 epochs.

3.6 Sonar

The system must destiguish between sonar signals bounced off a metal cylindar and those off a cylindrically shaped rock.
A sigmoid standard BP obtained a 89.2% generalization after 300 epochs, while the same architecture with sine functions archieved 92.0% after 400 epochs.

3.7 Other Problems

Learning the prediction of the next output of a finite state automata is ten times faster with sines.
A classification task with examples created using biconditional grammar was extremely fast and with a generalization of 98% plus using sines, a problem unsolvable with sigmoids. Generation of fixed length sequences was learned with a generalization of 98.2%, sigmoids also being unable to learn the task.

4 Conclusion

Networks using sine functions are superior in performance to sigmoids (in terms of learning speed, computational capacity and generalization). Tasks impossible to learn for sigmoid functions were easily learned using sine functions. The main problem is finding an initial set of weights in high-dimensional inputs.

5 Questions:

1) What is the error surface?
2) What is a parameter fit?
3) What is the momentum?
4) What is an odd-like function?
5) What is the separation problem?