A Study on Backpropagation Networks for Parameter Estimation from Grey-Scale Images

Tian-Jin Feng *
Department of Physics
Ocean University of Qingdao, P. R. China

Z. Houkes M. J. Korsten L. J. Spreeuwers
Department of Electrical Engineering, Twente University
P.O.Box 217, 7500 AE Eschede, The Netherlands

Abstract

The LSE (least Squares Estimator) method of parameter estimation from images is relatively time consuming. Furthermore, the models and the required initial parameters are often difficult to find in actual applications. The advantage of neural networks over the LSE is that a parameter estimation can be designed without explicitness of model functions and which requires no initial values used for estimating. A large number of experiments have been done on the basic research of parameter estimation from images with neural networks. To get a better estimate accuracy of parameters and to decrease needed storage space and computation time, the architecture of networks, the effective learning rate and momentum, and the selection of training set, are investigated. A comparison of network performance to that of the LSE, is made. The internal representations in trained networks, i.e. input-to-hidden weight maps or "measuring models", which include statistical features of training images and have a clear physical and geometrical meaning, and the internal components of output parameters given by outputs of hidden neurons, are presented.

1. Introduction

The research work at the University of Twente on the estimation of parameters from sequences of images began in the early 1980s [1]. These parameters specifically describe the shape, position and orientation of objects. This information is used by e.g. robots to manipulate objects. The parameters are estimated from a set of measurements which consist of grey values from digital images, and a model based prediction of measurements. Usually these models are non-linear. The estimation is therefore performed iteratively using Kalman filters with a model, linearized about a previous estimate [2,3]. This traditional method is time consuming for human designing models, and takes much computing time. However, in robotic applications processing speed is very critical, and in fact, the models and the required initial parameter values are difficult to find.

The neural network is an alternative approach to overcome these deficiencies. It uses a massively parallel computation and is programmed by example rather than by using an explicit model. It holds information in a distributed, associative memory, and it is especially useful for the case of the computation associated with basic cognitive processes such as vision and audition [4,11,12]. On the other hand, this has led to the situation that neural networks are widely applied, but that learning and generalization in neural networks are still poorly understood.

This paper discusses the design and performance of a network for parameter estimation from simple images, makes a comparison with the traditional method, and gives an understanding of the measuring strategy of networks.

As a basic research, standard backpropagation networks were simulated in the software. Some programs were written to generate training and testing images of 32*32 pixels, representing a bar positioned in the centre of the image and with 3 geometrical parameters: width (w), length (l) and angle (a). The grey values followed a Gaussian function in the width direction of the bar (see: Figure 1).

We can say there is an imaging model g{ } in the program:

$$\{\text{Image}\} = g\{w,l,a\} \tag{1}$$

generating images from the parameters of the bar. The background is black with some noise. The problem of parameter estimation is that we need to find out the inverse mapping :

* T.J. Feng did this research at the Department of Electrical Engineering, Twente University, the Netherlands.

CH 3065-0/91/0000-0331 $1.00 © IEEE

$$\{w,l,a\} = F\{Image\} \quad (2)$$

i.e. to estimate the parameters from the measured images in actual applications, but usually it is difficult to find the model (1) and the inverse (2). After training, the network learned to approximate the function F{image}.

2. Network Architecture and Training Method

2.1. Neural network

The network used in the experiments is a three-layer feedforward network, in which each neuron of the input layer is fed with a grey value of a pixel from the image, i.e. there is an input buffer with 32*32 neurons, Nr(input)=1024. The number of output neurons is determined by the number of estimated parameters, i.e. Nr(output)=3. The neurons in the hidden layer are fully connected to input and output neurons with connecting weights W_{ij}. The output function for neurons in hidden and output layers is a sigmoid with a bias.

The standard backpropagation algorithm with momentum, as proposed by McClelland and Rumelhart [8], is used to train the network. This gradient descent method is used to search for optimal setting of the weights. The weights are updated during each epoch, which consists of a cycle through all images in the training set. This process continuous until the number of epochs reaches a certain value. After the training phase for learning internal representations, the network is ready for testing with test images. The network can be trained repeatedly if necessary.

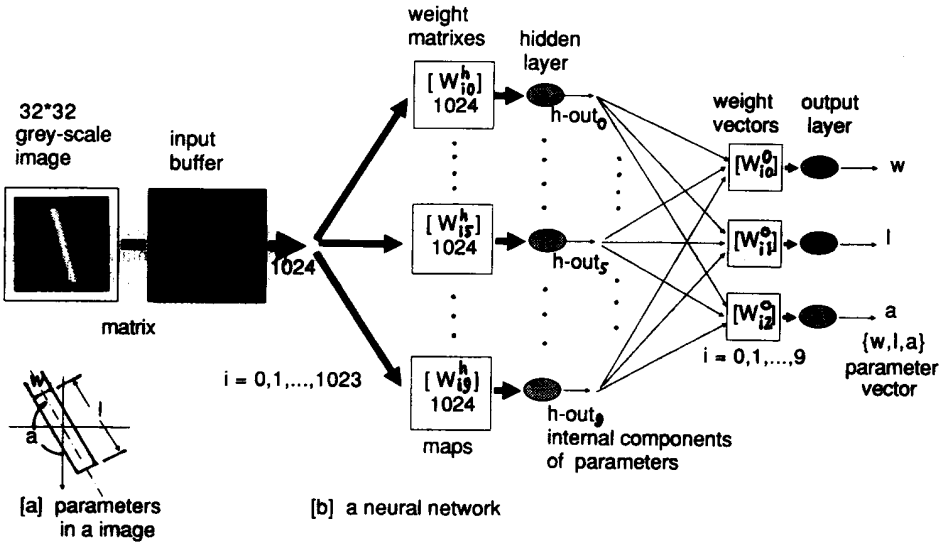

Figure 1. Architecture of the network
maps : internal representations of statistical features of training images
$h\text{-}out_j$: outputs of the hidden layer, as the internal components of output parameters

2.2 Hidden layer size

A question is how to determine the number of hidden neurons, Nr(hidden). The number of hidden neurons must be sufficient to realize a certain function. On the other hand, the training time of a network which has a larger number of connecting weights will be longer and more storage cells have to be allocated. The problem is that an excessively large hidden layer may be unreliable or unstable. In terms of the algebraic projection analysis [6], the solution may wander in the weight space where there are either too few hidden neurons (overdetermined case) or too many hidden neurons (underdetermined case).

According to our experiments, a better architecture of networks has one hidden layer, and the optimal number of neurons in the hidden layer is 10. Taking a large number did not improve the performance, whereas taking fewer neurons got worse performance. Incidentally, the number of neurons in hidden layer takes approximately

the logarithm (to the base 2) of the number of inputs.

2.3 Training set

Once the network architecture has been chosen, the training set and training time will determine the mapping represented by the network and its estimate accuracy. How to select a training set to accomplish a better performance is important for any application of networks. The theoretical analysis to select an optimal training set has not yet been established [7].

We used a program to generate a training set which consisted of 512 images. The parameters in the training set were uniformly distributed in the scaled range of 0 to 1 with an increment 0.125.

After above basic training, we found the results of estimate length were worse than others when the bars were in narrower width (see: section 4.2). A special retraining was made for an improvement on estimation accuracies (see: section 5). The bars in the special training set had more different lengths (from 0 to 1 with an increment 0.083) and narrow widths (from 0.05 to 0.25).

3. Learning Rate and Momentum

According to Rummelhart, "a kind of momentum was in weight-space that effectively filters out high-frequency variation of the error surface in the weight space"[8]. Sometimes the learning speed can be increased by using a momentum term in the backpropagation learning rule. But no clue is given as to how they should be set. Our experiments indicated that the learning curve oscillated obviously, the 'rms' error was bigger and the learning process took more time when a very small learning rate (0.01) and a very big momentum (0.99) were used, as proposed in [5].

The problem of an optimal matching of the learning rate and momentum has not been resolved for the general case. The error as a function of weights can be seen as an error surface in a high-dimensional weight space, and properties of error surface are quite different in various applications. We found an effective and practical choice as follows.

1) The learning rate should be small (=< 0.1) and momentum should not be too large (=< 0.5). For example, they were 0.1 and 0.5 respectively in above experiments.

2) They may be all larger if the architecture of networks and projects to be resolved are simpler. For example, the learning rate and momentum were 0.6 and 0.8 respectively in some sequential function syntheses and a curve fitting problem with 2 to 20 inputs, 4 to 10 neurons in hidden layer and 1 to 3 outputs.

4. Learned Internal Representations

Although it is difficult to understand the learned internal representations in a network, because too many neurons are involved and the representations are distributed over the network, many authors pay attention to this problem [9,10,11,12]. The internal representations in trained networks are quite different in various applications, and our interest is in understanding the measuring strategy developed by networks. After thinking carefully out the relationship of input images, output parameters and the architecture of the network, some interesting results were discovered.

4.1 Hidden-to-output weight vectors and internal components of parameters

In the used model of neurons the output function,f(.), is a sigmoid, and the output of j_{th} neuron in a layer is:

$$Y_j = f(W_{oj} + \sum_i W_{ij} * X_i) \qquad (3)$$

where the W_{oj} is connected with the bias, a constant 1. The X_i are neuron outputs of preceding layer, and $1 >= X_i >= 0$ in standard backpropagation networks. Before training phase, W_{ij} are initialized with random values which are uniformly distributed in the interval [-0.5 0.5]. After training, the big positive W_{ij} will make Y_j bigger and the negative W_{ij} will make Y_j smaller.

Table 1 shows values of hidden-to-output weight vectors, W^o_{ij}, in a trained network. The X_i in the equation (3) here are the outputs of hidden layer, $h\text{-}out_j$, which are allocated to 3 parameters by the hidden-to-output weight vectors. So, we call $h\text{-}out_j$ the internal components of parameters. For example, we call $h\text{-}out_1$ the wide width component, w+, since a big positive weight, +4.0, connects with it and makes it important for computing a bigger angle parameter, and so on. Each parameter, as an output of output layer, has more than two internal components which are connected to the big positive or negative W^o_{ij}.

Table 1. values of hidden-to-output weight vectors of an experiment network
$[W^o_{i0}]$, $[W^o_{i1}]$ and $[W^o_{i2}]$: the width, length and angle weight vector respectively
(+) and (-) represent positive and negative numbers which absolute values < 1 .
a-, l- and w- : small angle, short length and narrow width component respectively
a+, l+ and w+ : big angle, long length, and wide width component respectively
as outputs of hidden layer, connected with W^o_{ij}.

W^o_{ij}	bias	(i:) 0	1	2	3	4	5	6	7	8	9
W^o_{i0}	(-)	-3.5	+4.0	+3.8	+1.9	+7.7	-2.5	(+)	-2.2	-2.0	(+)
W^o_{i1}	(+)	(+)	(+)	(+)	+2.1	(-)	(+)	(-)	(-)	(-)	-3.7
W^o_{i2}	-2.0	(-)	(+)	(+)	(+)	(+)	+2.6	-2.8	(+)	+2.4	(+)
with	(-a)	w-	w+	w+	l+ w+	w+	a+ w-	a-	w-	a+ w-	l-

It should be noted that each of about 7 outputs of the hidden layer plays a role of a parameter component without obvious correlation with another component, i.e. an output of the hidden layer mainly determine one parameter as an output of the output layer. For example, the 'h-out_6 * (-2.8)' will make the parameter angle smaller and 'h-out_8 * 2.4' will make it bigger, so the 'h-out_6' can be called a small angle component and 'h-out_8' a big angle component. The h-out_3, h-out_5 and h-out_8 are constituted of two parameter components, i.e. they have influences simultaneously on two parameters. For example, the h-out_3 is constituted of l+ and w+.

Our experiments indicate that the correlative cases is less when a network is trained better. There will occur obvious correlation if the number of hidden neurons is less than 2 times Nr(parameter), and the estimate accuracy will be worse. It follows that the minimum of hidden neurons = 6 (see: section 2.2).

4.2 Input-to-hidden weight maps, internal measuring models

Figure 2 shows input-to-hidden weight maps, $[W^h_{ij}]$, being relative to pixels of input image. The X_i in the equation (3) here are the grey values of the input image, and Y_j are h-out_j. Because the bars in training set are symmetrical about the centre, there is a relative symmetry in weight maps. These weight maps have a clear geometrical meaning and represent the statistical features of training images.

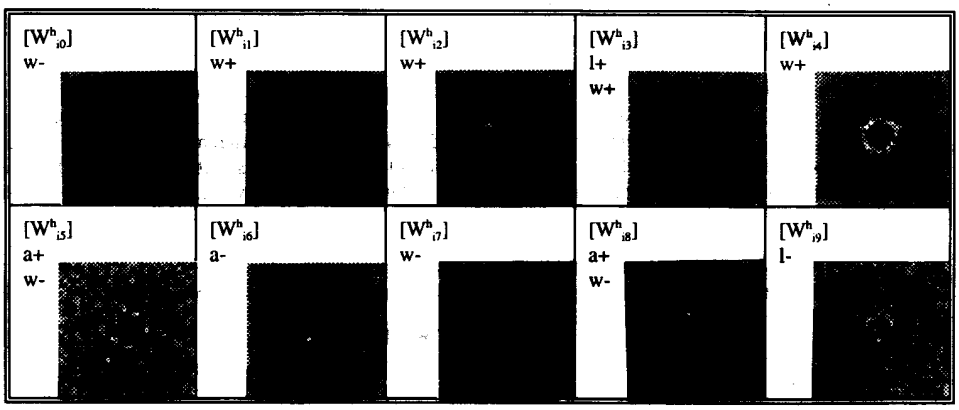

Figure 2. Input-to-hidden weight maps, internal measuring models
Grey values of small rectangles represent weights.
White and black rectangles represent positive and negative weights respectively.

Noting weight maps $[W^h_{i8}]$ and $[W^h_{i6}]$, there are some bigger positive values distributed in the direction of big angle in $[W^h_{i8}]$ (see: angle parameter in Figure 1 [a]), similarly in the direction of the small angle in $[W^h_{i6}]$. On the other hand, bigger negative values take the opposite direction in $[W^h_{i8}]$ and $[W^h_{i6}]$. It is clear that $[W^h_{i8}]$ will join to compute the big angle component (a+, h-out_8), and $[W^h_{i6}]$ to compute the small angle component (a-, h-out_6).

It is interesting to notice that maps $[W^h_{i8}]$ and $[W^h_{i6}]$ are somewhat similar to an anisotropy model of the iso-orientation domain, which is about angle also, in the visual cortex model [13].

Noting maps $[W^h_{i2}]$,$[W^h_{i4}]$,$[W^h_{i1}]$ and $[W^h_{i0}]$,$[W^h_{i7}]$, and that the maximum of width is 10 pixels in our experiments, there are some bigger values distributed within a radius of 5 pixels in $[W^h_{i2}]$,$[W^h_{i1}]$ and $[W^h_{i4}]$, and they will join computing of the wide width component (w+). There are oppositive cases of $[W^h_{i0}]$ and $[W^h_{i7}]$ which have influence over the narrow width component (w-).

Noting $[W^h_{i9}]$ and $[W^h_{i3}]$, there are some bigger positive values in $[W^h_{i9}]$ distributed near to the centre to compute short length component (l-), but far from centre in $[W^h_{i3}]$ for computing the long length component (l+). Because the measurement of lengths is relative to whole area where the bars exist, there are some little-bigger weights (about 0.8, they seem noise) distributed in the same larger area, i.e. the information of lengths is more dispersed and sometimes is lost when the width is narrower. That is why the average accuracy of estimate lengths could be worse than others. The values of width and angle have to collect near to the centre as some maps show, because many bars have only short lengths.

Now let's see what happen when a test image is shown on the network. Supposing there is a bar in the test image which has a small angle parameter, the input-to-hidden weight map, $[W^h_{i6}]$, will collect the grey values of the bar and only the correct neuron, i.e. the number 6 neuron in the hidden layer fires strongly, and a larger value, $h\text{-}out_6$, makes the angle parameter smaller. The small and big angle components both will be small if the bar has a middle angle parameter, and so on.

It is meaningful that the network has learned to use the internal representations to link the input image to the higher level concepts, geometrical parameters. It can be said, therefore, that input-to-hidden weight maps are a kind of internal measuring models of parameter components.

5. Comparison with LSE

As has been said, there is no general method for an analytical inversion of given non-linear relation (1). There exist however numerical algorithms, such as the extended Least Squares Estimator (LSE), as shown in Figure 3.

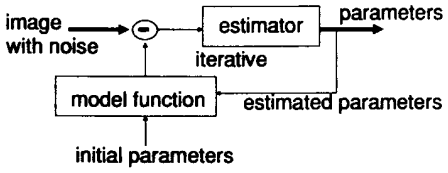

Figure 3. LSE method
using special algorithm

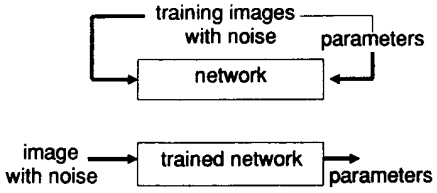

Figure 4. Network method
by example rather than algorithm

By LSE the parameters converge to the optimal values during a number of iterations. While performing an iteration the LSE estimates the difference between actual parameters and the previous guess, with which the guess is updated [3]. Neural networks take a rather different way, as shown in Figure 4.

To compare LSE with network performance, the same program was used to generate 10 32*32 images with

Table 2. A comparison of network performance to that of LSE
amplitude of image signal = 1.0 amplitude of noise = 0.1
$: when initial parameter values close to targets R : retrained network

performance \ comparison	LSE		Network	
training time	do not need		about 150 hr. +140 hr. (R)	
estimate time (of 1 image)	about 10 seconds		about 1 second	
rms : average (for 10 images)	0.038 ($ case)	0.130	0.067	0.051 (R)
best one worst one	0.002 ($ case)	0.520	0.012 (R)	0.163 (R)

random parameter values for testing. The network was trained with a training set of 512 images and 8000 epochs. To retrain, a new training set of 480 images, in which bars had more different values of length in narrower width, was used and the network was retrained for 6000 epochs. The results are shown in Table 2.

The LSE will give a high accuracy of 0.002 when initial parameter values are close to target values, but it will not converge or will reach wrong results sometimes if initial values are quite different from the targets. And the advantage of networks over the LSE simply is the capability of learning internal representations by examples.

6. Conclusions and Discussions

1) The application of artificial neural networks to the parameter estimation from grey-scale images is a meaningful work. A trained network can use the internal representations, i.e. measuring models, to link the input image to higher level concepts, parameters, and give better accuracies at a relative high speed.
2) The requirements of the LSE method of parameter estimation from images are the models and the initial parameters which are often difficult to find in actual applications. The requirements of the neural network is the training set that we need to collect as many samples as we can.
3) In some cases, to estimate parameters from images a trained network can be used first, and then the LSE is used for higher estimation accuracy using results given by the network as initial parameter values, if the model function was known.
4) A fundamental challenge is that we don't know how to incorporate a priori knowledge into networks. For example, there are some areas in pictures, in which there is not any object present to estimate, therefore the input-to-hidden weights of relative areas are not needed to train, and so on. We could get a smaller training set and take shorter training time if we are allowed to do that.

References

[1] Houkes Z., "Motion Parameter Estimation in TV-Pictures", In: NATO ASI series, Vol F2: Images Processing and Dynamic Scene Analysis, Ed.by Th. S. Huang, Berlin Springer Verlag, pp249-263, 1983.
[2] Houkes Z., Korsten M.J., "Considering Shape from Shading as An Estimation Problem", Proceeding of SPIE/SPSE Symposium on Electronic Imaging: Image Processing Algorithm and Technique, Santa Clara, CA, USA, 11-16 Feb. 1990, Vol.1244, pp 56-67.
[3] Korsten M. J., Houkes Z., "The Estimation of Geometry and Motion of A Surface from Image Sequences by Means of Linearization of A Parametric Model", Comp. Vision, Graph. and Im. Proc. Vol.50, No.1, pp 1-28. 1990.
[4] Spreeuwers L.J., "A Neural Network Edge Detector", accepted for SPIE/SPSE's Symposium on Electronic Imaging: Science and Technology, San Jose, Cal., USA, 24 Feb.-1 March 1991, Proc. SPIE Vol.1451.
[5] Lloyd G. Allred, Gary E. Kelly, "Supervised Learning Techniques for Backpropagation Networks", June 17-21,1990. IJCNN Conference, p I-727.
[6] S. Y. Kung, J. N. Hwang, "An Algebraic Projection for Optimal Hidden Units Size and Learning Rates in Back-Propagation Learning", IEEE International conference on Neural networks, July 1988.
[7] M. Wann, T. Hediger, N.N. Greenbaun, "The Influence of Training Sets on Generalization in Feed-Forward Neural Networks", IJCNN, San Diego, California, June 17-21, 1990.
[8] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, Parallel Distributed Processing (PDP): Exploration in the Microstructure of Cognition (Vol.1). "8 Learning Internal Representation by Error Propagation". pp.318-362, MIT Press, Cambridge, 1986.
[9] R. Paul Gorman, Terrence J. Sejnowski, "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets", Neural Networks, Vol.1. pp.75-89, 1988.
[10] Garrison W. Cottrell, Paul Munro, David Zipser, "Learning Internal Representation from Grey-Scale Images: An Example of Extensional Programming", In Proc. 9th Annual Conference of the Cognitive Science Society, pp.461-473.
[11] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, Kevin J.Lang, "Phoneme Recognition Using Time-Delay Neural Networks", IEEE Transactions on Acoustics, Speech, and Signal Processing. Vol.37. No.3, March 1989.
[12] Teuvo Kohonen, "The Self-Organizing Map", Proceedings of The IEEE, VOL.78, NO.9, September 1990.
[13] Ilya A. Rybak, Natalia A. Shevtsova, Lubov N. Podladchikova, Alexander V. Golovan, " A Visual Cortex Domain Model and Its Use for Visual Information Processing", Neural Networks, Vol.4. pp.3-13. 1991.