INTERNAL MEASURING MODELS IN TRAINED NEURAL NETWORKS FOR PARAMETER ESTIMATION FROM IMAGES

Tian-Jin Feng(1) Z Houkes(2) M J Korsten(2) L J Spreeuwers(2)

(1) Ocean University of Qingdao, P R China (T.J. Feng did this research at (2))

(2) Twente University, The Netherlands

ABSTRACT The internal representations of 'learned' knowledge in neural networks are still poorly understood, even for backpropagation networks. This paper discusses a possible interpretation of the learned knowledge of a network trained for parameter estimation from images. The outputs of the hidden layer are the internal components of the output parameters. The input-to-hidden weight maps, functioning as a kind of internal measuring model of the parameter components, include statistical features of the training set and seem to have a clear physical and geometrical meaning.

1. INTRODUCTION

The modelling of an image, when a "conventional" method is used, is a complex and time-consuming job in actual applications of parameter estimation [1,2]. A neural network uses massively parallel computations and a generalized model to represent various input-output relation by examples rather than by using an explicit model. It holds information of internally created model in a distributed, associative memory.

This has led to the situation that many attempts are made to apply neural networks, while learning and generalization in neural networks are still mystical [3].

The internal representations in trained networks are quite different in various applications. This paper discusses the design and performance of a network for parameter estimation from images, and gives some understanding of the internal measuring strategies developed by the neural network.

To be able to carry out some basic research, some programs are written to generate training and test images of 32*32 pixels, representing a bar positioned in the centre of the image and defined by 3 geometrical parameters: width (w), length (l) and angle (a). The grey levels obey a Gaussian function in the width direction of the bar. The background is black with additive white noise.

Images obeying the parameters of a bar are denoted in the program:

$$\{\text{Image}\} = g\{w,l,a\} \qquad (1)$$

The problem of parameter estimation involves the inverse mapping:

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

In actual applications, that is to estimate the parameters from measured images, and it is difficult to find the model $g\{w,l,a\}$ by traditional methods. A trained neural network has learned to approximate the mapping $F\{\text{image}\}$ and computes the parameters with better accuracies and at a relative high speed (one image per second on SUN SPARC station 1) [2].

2. NETWORK ARCHITECTURE

In understanding the internal representations, a fundamental question is to analyze the relation between the used neural network architecture and the given task.

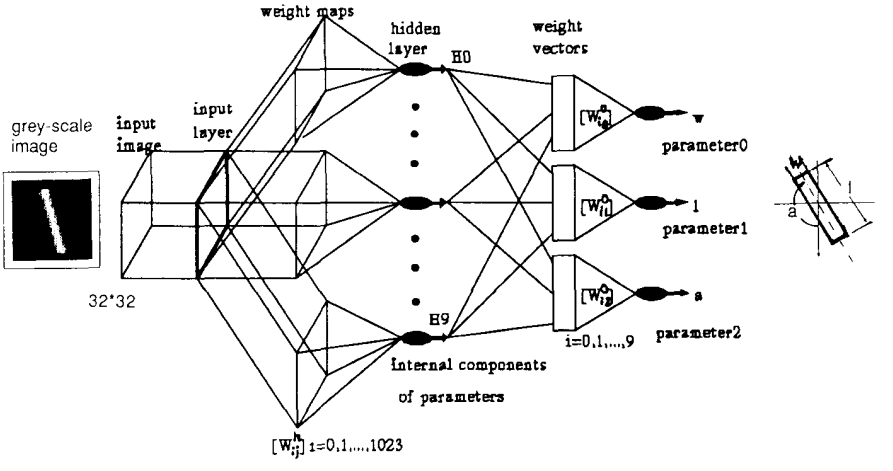


Fig.1. Architecture of an experimental network

In the experiments, the parameters to be estimated depend on the whole input image, i.e., they are computed from the grey levels in the global input space. Therefore, the number of input neurons in our experimental networks equals to the number of pixels in one image, i.e., each neuron of the input layer is fed with the grey level of the corresponding image pixel. The number of output neurons is determined by the number of estimated parameters.

The neurons of the hidden layer are fully connected to the input and output neurons with connecting weights W_{ij} in a multilayer feedforward network. A kind of massively parallel computation, related to the global input image, will take place in the neural networks.

According to our experiments for 32*32 images with three parameters, the network needs only one hidden layer, and the optimal number of neurons of the hidden layer is 10. Incidentally, the number of neurons of the hidden layer takes approximately the logarithm (to the base 2) of the number of inputs (see: figure 1).

Experiments show that the learning rate should be small (=<0.1) and the momentum should not be too large (=<0.5) [2].

3. INTERNAL COMPONENTS OF PARAMETERS

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

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

where the W_{bj} is connected with the bias, a constant 1. The X_i are neuron outputs of the preceding layer, with $1 >= X_i >= 0$ in standard backpropagation networks [5].

For the output layer, the X_i in the equation (3) are the outputs of hidden layer, H_i, and we have:

$$Parameter_j = f(W^o_{bj} + \sum_i W^o_{ij} * H_i)$$

with j=0,1,2. Before the training phase, the weight factors W_{ij}^o are initialized with numbers randomly distributed in the interval [-0.5, 0.5] with zero mean. After training, the outputs of the network, i.e., the parameter values are strongly determined by some large weights. A large positive W_{ij}^o will make $Parameter_j$ larger and a large negative W_{ij}^o will make $Parameter_j$ smaller in the case of a significant H_i.

We divide $[W_{ij}^o]$ into two: a positive part $W_{pj}^o=[+]W_{ij}^o$ and a negative part $W_{nj}^o=[-]W_{ij}^o$, and take their sign out of the sum:

$$Parameter_j = f(W^o_{bj} + \sum_p W^o_{pj} * H_p - \sum_n W^o_{nj} * H_n)$$

Therefore, we call H_p the large internal component of a parameter if it connects to a large positive W_{pj}^o, and call H_n the small internal component of a parameter if it connects to a large negative W_{nj}^o.

According to the sigmoid function, an estimated parameter may be combined from a few large and small components, related to maximum and minimum of a parameter in the training set respectively. In our case, a network can be trained to do in this way, and each output parameter should have more than two internal components (one large and one small component) for better accuracies of the estimate.

Figure 2 and Table 1 show two examples for hidden-to-output weight vectors, W_{ij}^o, in different trained networks. In Table 1, the W_{i0}^o, W_{i1}^o and W_{i2}^o are used for computing the width, length and angle parameter respectively. In other words, H_i are allocated to the three parameters by the hidden-to-output weight vectors. We call H_1 the large width component (w+), for example, since a big positive weight (+4.0), connects with it and makes it important for computing a larger angle parameter, and so on.

It should be noted that each of about 7 outputs of the hidden layer plays a dominant role with respect to one of the parameter components without obvious correlation to others, i.e., an output of hidden layer mainly contributes toward one parameter. But the H_3, H_5 and H_8 are mainly contributing to two parameter components and can be called the compound internal components.

Our experiments indicate that the correlative case will be less when a network is trained better. Obvious correlation occurs if the number of hidden neurons is less than 2 times the number of parameters, and estimating accuracies will be worse. It follows that the minimum of hidden neurons = 6.

Which component of H_i will be representative for which component of parameters, is strongly dependent on the network architecture and training set.

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

TABLE 1. The $[W_{ij}^o]$ in a trained network
(+) and (-): positive and negative numbers which absolute values < 1
a-, l- and w-: small angle, length and width component respectively
a+, l+ and w+: large angle, length and width component respectively

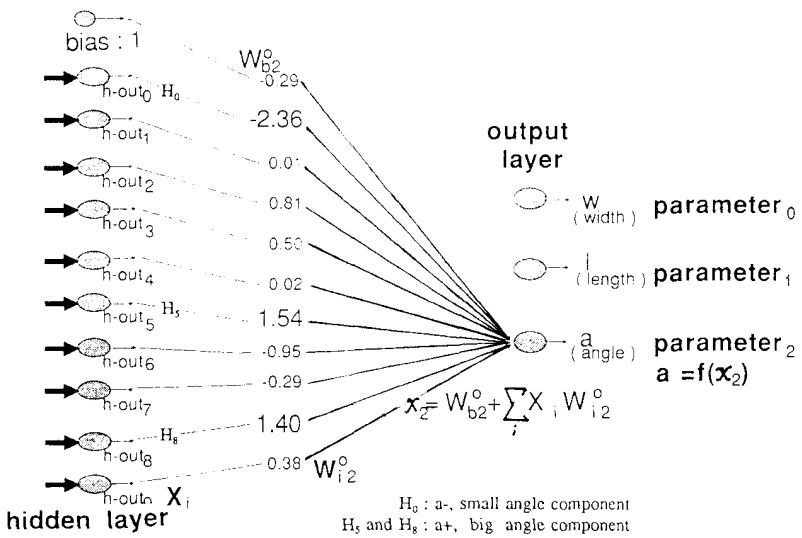


Fig.2. Internal components of the angle parameter in a trained network

4. INPUT-TO-HIDDEN WEIGHT MAPS

As mentioned above, each neuron of the input layer is fed with a grey level of an image pixel. The X_i in the equation (3) for hidden layer are the grey values of input image, and the Y_j are the outputs of hidden layer, H_j or $Component_j$, i.e.

$$Component_j = f(W^h_{oj} + \sum_i W^h_{ij} * X_i)$$

The weighted sum is a sum of products $[W^h_{ij}]$*[image].

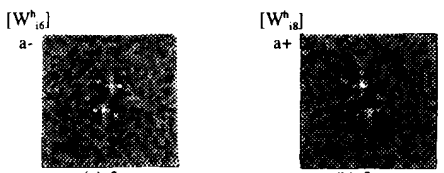
To consider the input-to-hidden weights for each hidden neuron as a map (or a kind of image), correlated to the input image, is an important research approach in our case. Each internal component of parameters depends on this products [image]*[map], and all internal components are combined with hidden-to-output weight vectors to compute the output parameters. This is the so-called massively parallel computation.

Weights are memory units of learned knowledge in networks. The input-to-hidden weight maps are developed through the learning phase to measure the internal components of parameters. They abstract the statistical features from the training images. Being relative to the bars in training set are symmetrical about the centre, there is a similar symmetry in the weight maps.

Now let us see what happens when a 'test image' is offered to a trained network, and how the network links the input image to the higher level concepts, geometrical parameters.

4.1 To measure the angle of a bar

Noting weight maps $[W^h_{i8}]$ and $[W^h_{i6}]$ in Figure 3, there are some larger positive values (white rectangles) distributed in the direction of the large angle in the centre area of $[W^h_{i8}]$



(a) for small angle component

(b) for large angle component

Fig.3. Two weight maps in the trained network [A]
-- The grey levels of small squares represent weights.
-- The white and black squares represent positive and negative weights respectively.

and in the direction of small angle in $[W^h_{i6}]$. On the other hand, larger negative values (black rectangles) take the opposite direction. The distributing angle of larger weights is related to the maximum and minimum of the bar angle parameters in the training images respectively.

It is clear that $[W^h_{i8}]$ will contribute to the large angle component a+ (H_8), and $[W^h_{i6}]$ to the small angle component a- (H_6).

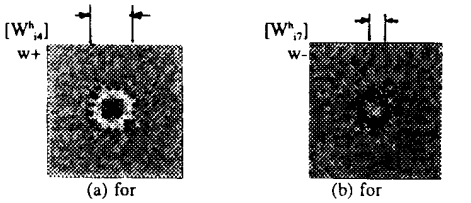
Supposing there is a bar in a test image with a small angle parameter, the map $[W^h_{i6}]$ in Figure 3 will collect grey levels of the bar and fire strongly the neuron number 6 in the hidden layer. The H_6 will get a larger value which causes a smaller angle parameter.

There is a similar result if the bar has a large angle, so that $[W^h_{i8}]$ will make H_8 larger in Figure 3. In the other cases,

both the small and large components will be smaller if the bar has a medium valued angle parameter.

It is interesting to note that the maps $[W^h_{i8}]$ and $[W^h_{i6}]$ in Figure 3, are somewhat similar to an anisotropy of the lateral connections of an inhibitory interneuron in the visual cortex when the angle is also small [4]. It is not strange that there are some similar phenomena between them, since artificial neural networks are inspired by studies of the physiological nervous system, although with artificial simplifications.

4.2 To measure the width of a bar



(a) for large width component

(b) for small width component

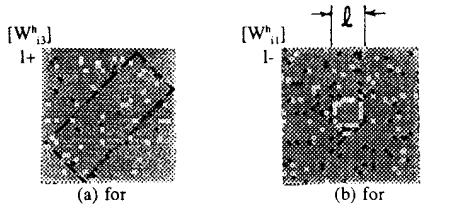
Fig.4 Maps for measuring width (in trained network [A])

Similarly, if a test bar has a larger width, the maps $[W^h_{i4}]$ in Figure 4 will make the 'large width' component H_4 larger, and $[W^h_{i7}]$ will make the 'small width' components smaller. The 'small width' component H_7 in the trained network [A] will be larger only when the bar is narrower. There is a white ring with a radius of 5 pixels in $[W^h_{i4}]$ in Figure 4, and the maximum of the bar widths in training set just is 2*5 pixels. There is a small white area in $[W^h_{i7}]$ which size is just related to the minimum of the bar widths in training set.

4.3 To measure the length of a bar

The measuring of lengths is relative to the whole range where the bars exist. So, the 'learned' information about lengths are more dispersed. That is why the weight values, $[W^h_{i3}]$ in Figure 5, all are smaller than 1 and seem to be noise.

$[W^h_{i1}]$ and $[W^h_{i3}]$ in Figure 5 are more clear, where we change the way to display the maps. The white and black rectangles respectively represent positive and negative weights which absolute values > 0.45, i.e., the larger weights display only.



(a) for large length component

(b) for small length component

Fig.5. Maps to measure length in trained network [B] (training set: 1000 images, amplitude of single = 1.0, noise amplitude = 0.1)

The map $[W^h_{i3}]$ will collect more grey levels and make the large length component larger, if a bar is longer. There are some larger weights (white rectangles), in a small area of $[W^h_{i1}]$, which is related to the minimum of the bar lengths in training images. It is easy to understand that $[W^h_{i1}]$ will make the small length component larger if the bar is short.

The experiments show that the accuracy will be worse and estimated length value will be smaller than the actual parameter sometimes, when the bar in a test image is longer and narrower. This result can be interpreted by the maps. That is, the 'large length' component, which is related to the products $[W^h_{i3}]$*[image], will be smaller when a bar has a smaller width and positions in a special angle space where there are very few large weights in $[W^h_{i3}]$.

It follows that the distribution of angle parameter values in training images influences the accuracy of length-estimation. The accuracy of length estimation will be better if the angle increment, in a uniform distributed parameter space of the training set, is smaller.

5. CONCLUSION

The internal representations of learned knowledge, i.e., the weight distributions in a trained backpropagation network with one hidden layer for parameter estimation from images, can be considered as a kind of internal measuring models used to link the input image with the higher concepts, described by the geometrical parameters w,l,a.

That is, the output parameters are divided into two types: the large and small internal components which are the outputs of the hidden layer. The internal components are computed parallel through out the input-to-output weight maps. The maps seem to have a clear physical and geometrical meaning, to abstract statistical feature from the training set and represent the special measuring strategies developed by the artificial neural networks.
Further research will be carried out to verify the proposed interpretation of the knowledge acquired by the network.

REFERENCES

[1] 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.
[2] Feng T. J., Houkes Z., Korsten M. J., Spreeuwers L. J., "A Study on Backpropagation Networks for Parameter Estimation from Grey-Scale Images", IJCNN 91' Singapore, November 18-21 1991.
[3] Minsky, M. L., Papert, S. (Expanded edition 1988). < Perceptions >. Cambridge. MA: MIT Press.
[4] Ilya A. Rybak, Natalia A. Shevisova, 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.
[5] 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.