Adaptive Performance Network? Is it the Answer to our Prayers?

A.R. Kian Abolfazlian
Computer Science Department
Aarhus University
Ny Munkegade, Bldg. 540
Dk-8000 Arhus C
Denmark

September 1994

Abstract

[Bak] has introduced a new class of adaptive networks, which tries to function in a volatile environment. The resulting dynamics, called *adaptiue performance* networks, show fluctuation over a wide range of time scales. We introduce some guide lines for training and testing these new type of networks, which will be used to examine the power of these networks.

1 INTRODUCTION

The traditional artificial neural networks are somehow dependant on an external teacher, whose responsiblity is to make network adapt in a specific environment. These networks have been used in the development of technologies for pattern recognition, and emphasis has been on maximizing their capacity as a learning machine, in respect to statistical models, without any effort to refine them, 80 they can produce "a real brain functio". I realise,

that may be an overstatement of the fact, but it is not completley 50 either. The question is, if it is possible to find more brain-like kind of artificial neural networks. Therefore as an attempt to answer this question, the notion of *Adaptiue Performance Networks* **APN**s is introduced by [Bak]. APNs are actually very simple McCulloch-Pitts networks (MPNs) with a global evalutive feedback signal and a reinforcement rule with no need of an external teacher. Like in the original MPNs here a neuron either is in the firing mode ($n_j = 1$), or in the non-firing mode ($n_i = 0$). The activation h_i of the i'th neuron is computed as usual as:

$$h_i = \sum_j \omega_{ij} n_j$$

where w_{jj} is weight of the connection from the j'th neuron to the i'th one. A neuron fires if its activation exceeds a threshold value θ, i.e.

$$n_i' = sgn(h_i - \theta_i)$$

where $sgn()$ is the usual sign function. Let us now suppose that our APN has N neurons, of which C neurons represent an *output region*, whose output we accept as the output of APN. The modification of a synapse, i.e. the change of the weight of a connection, called here *reinforcement*, is allowed to depend only locally on the coherence in activity. The goal of reinforcement is to strengthen firing paths that lead to a positive feedback, and weaken the paths resulting in a negative feedback. This is done here using the following reinforcement rule,

$$\omega_{i,j}' = \frac{\hat{\omega}_{i,j}}{\sum_i \hat{\omega}_{i,j}}$$

where,

$$\hat{\omega}_{i,j} = \omega_{i,j} + \mathcal{F}(\omega_{i,j}) n_i n_j$$

where $\mathcal{F}$ is a simple logistic function,

$$\mathcal{F}(x) = \kappa x(1-x) + \psi$$

with $|\kappa| << 1$. APNs has the constraint $\forall i,j : 0 < \omega_{ij} < 1$, which implies that all connections are excitatories. Anyhow with small changes of the reinforcement rule we would be able to use both excitatory and inhibitory connections. Possible noise is represented by ψ which we choose randomly in space and time from a uniform distribution of values between $-\psi_0$ and $+\psi_0$. It is now obvious, that if we want to keep the feedback of the system as versatile as possible, we can't allow it to be task specific, otherwise what would be the use of this when we have *error back-propagation* procedure! Thus the feedback has to be just *evaluatiue*. Our feedback is then based on just 2 possiblities[1], namely **YES** or **NO**, (**RIGHT** or **WRONG**), and due to this constraint κ takes 2 values

$$\kappa(\mathbf{YES}) = \kappa_+ \text{, and , } \kappa(\mathbf{NO}) = \kappa_-$$

which are used upon the behaviour of the APN[2] Furthermore an *internal* control mechanism is introduced. On the assumption, that it is essential for the performance that the number of firing neurons in the output region, i.e. $\mathcal{A} = \sum_{i \in C} n_i$ kept at a minimum, this internal control machanism is represented by a constraint on the threshold value θ. If $\mathcal{A}$ exceeds a predetermined value $\mathcal{A}_0$ the threshold θ is increased, while θ is lowered if $\mathcal{A} < \mathcal{A}_0$, i.e.

$$\theta' = \theta + \varphi \mho(\mathcal{A} - \mathcal{A}_0)$$

where

$$\mho(x) = \begin{cases} 1 & : \quad x > 0 \\ 0 & : \quad x = 0 \\ -1 & : \quad x < 0 \end{cases}$$

[1]This is done for the sake of simplicity.

[2]The behaviour is rewardered globaly, if it includes the desired action (κ_+), otherwise punished κ_-. This means, that if the desired output is a part of the firing action in the net's output region, then we use κ_+, and otherwise κ_-. In each case the value of $\kappa()$ is chosen globally, i.e. the same value is used for every weight in the net.

and, $0 < \varphi << 1$. $\mathcal{A}_0$ is small compared to $\mathcal{A}_{max} = |C|$.

2 TRAINING

The external teacher paradigm plays a special role in the realization of whole concept of training and testing. The reason is that these external teachers are almost all dependant on a differentiable transfer function, which among other things implies that we just need to look at a pattern of data once to get it propagated through the network, i.e. the signal coming through the input region gets all the way up into the output region. This is one of the non-biological features of ANNs, which has been forced into the field by the learning algorithms based on the external teacher paradigm. In the case of human beings it is the fact, that not every stimulus will be get the whole way through from the input region to the output region. This is actually one the ways, in which we get rid of the noise. This particular feature is preserved in APNs. Due to the very nature of APNs being McCulloch-Pitts networks, it is not the case that the input signal will always be propagated the whole way through. Here the internal control machanism comes into action. By looking at the same pattern of data you can build a channel from the input region, through the network's layers, into the output region. This period of *silence* can be interpreted as the time which the system has to use to adjust itself to the new environment, so it can come up with any response. Due to this we have to redefine the whole procedure of training. That is, that while we train the network, we have to allow it to look at each pattern in a period, afterwhich we can start using the reinforcement rule . Here we have two options.
Firstly, we can choose to let the network look at each pattern once and reinforce, afterwhich we change the pattern and so on. This may seem to be a *hard* training procedure, but it can be compared to the online learning, which has been used as a serious training procedure[Hertz].
Secondly we can let the network look at each pattern in a period of *thought*, in which the network reinforce itself again and again by means of using the internal control mechanism and the reinforcement rule right after each other. This period of thought has turned out to be very important. The longer it is, the easier it gets to learn the right response again. The problem here is, that the firing pattern corresponding to a pattern of data is weakened while

we train the network on another pattern. If the period of thought has been long, it strengthens the particular firing pattern in a way, so even though we train the net on another pattern it gets easier to restablish it again.

3 RESULTS

We used an APN built as a grid of 64 by 128 artificial neurons for the purpose of training. At first we used 1 - 1 *corresponding methode*, where we assigned 1 neuron in the input region to 1 neuron in the output region ($\mathcal{A}_0 = 1$). The APN was very good at learning this pattern and did learn the pattern after 100 epochs in average for a sample of 32 different patterns of data[3]. An example of this is shown in the figure 1, where the firing pattern eventually takes shape as a path from the firing input neuron to the desired output neuron[4].
Figure 1:
1-pattern training

The next step was 1-pattern training, where the input pattern now consistet of more than one firing neurons. For each output neuron we constructet 32 rantom generated input patterns and trained the network by them. The result was again satisfactory. APN could learn the patterns after 266 epochs in average.
Now the obvious question was, whether APN could learn the whole data set as it can be done by ordinary artificial neural networks. But the results were disappointing. APN showed a very poor ability to preserve its inner structure achieved during the periot of training on one pattern when using the Hard training (On-line learning) procedure both in case of single and multiple firing input patterns. On the other hand training by the usage of the period of thought, where APN was given time to learn a particular pattern of data completely before it was introduced to a different one seemed to work in a peculiar manner. In the case of Single firing input pattern it was able, if the period of thought was long enough, to build a internal kernel-like structure, which can be seen in the Figure 2.

[3]Here we have, that the period of silence is equal to 100.

[4]The topline in the figure shows the“input” region, while the bottom line shows the “output” region.

Figure 2:
Kernel-like structure

Figure 3:
Usage of the Kernel-like structure

By training in 1024 epochs in average it had a 6eneralisation of 80 % on the train- ing set. The kernel-like structure in the APN was used like this. When APN was presented to a learnet training pattern it uset the internal control machanism in order to make a path from the firing input neuron to the kernel-like structure and from the kernel-like structure to the desiret output neuron. This is shown in Figure 3. The period of silence here was equal to 7 epochs in average.

The case of Multipe firing input patterns showed to be harder to learn from APN due to the lower percentage of generalisation (63 % in the best trial) and the longer period of thought (1017 in average). In this case again the learning was due to the kernel-like structure.

4 FUTURE WORK

APNs show a more structured type of learning, although the learning is not very impressive, and that is why they could be very interesting to explore as an alternative to traditional artificial neural networks for the purpose of simulating the cognitive processes. There are obviously many open questions about APN's, that still have to be worked on. The structure and appearence of the so-called *kernel structure* has to be explored. The question of training on the data set with the multiple firing output neurons is still open, although the author has his doubts about the success of this kind of training.

References

[Bak] P. Bak & D. Stassinopolus, "A Simplc Model of Brain", Brookhaven National Laboratory, Upton NY 11973, U.S.A., 1994, (Also in the pro-

ceedings of WCNN94)

[Hertz] John Hertz, Anders Krogh & Richard G. Palmer, "Introduction to the Theory of Neural Computation", Addison Wesky Publishing Company.