UT-632

Universal Power Law in Feed Forward Networks†

Hideyuki CÂTEAU#

Department of Physics, University of Tokyo
Bunkyo-ku,Hongo, Tokyo, 113 Japan
e-mail:cateau@tkyux.phys.s.u-tokyo.ac.jp

February 1993
Subject Index:Neural Networks

Abstract:

The power law in the pace of the memory, which was previously reported for the encoder, is shown to hold universally for general feed forward networks. An extensive simulation on wide variety of feed forward networks shows this and reveals a lot of interesting new observations.

† Work supported in part by Grant-in-Aid for Scientific Research No.04-2348 from the Ministry of Education, Science and Culture of Japan.
# Fellow of Japan Society for the Promotion of Science.

Suppose that we train a gradient descent feed forward network[1] so that it outputs a vector $\mathbf{t}^{(1)}$, in response to an input vector $\mathbf{x}_{in}^{(1)}$, outputs $\mathbf{t}^{(2)}$ to $\mathbf{x}_{in}^{(2)}$, $\cdots$,$\mathbf{t}^{(M)}$ to $\mathbf{x}_{in}^{(M)}$. As it memorizes these pairs of vectors, the learning slows down due to interferences between different memories. The interference is specifically caused by the fact that an update of the connection weight which is good for one pair of vectors are not good for another pair. Therefore, if $t(M)$ denotes the time necessary to memorize M items, $t(M)$ must be a function of M increasing faster than a linear function of M.

We human being also experience the interference when we try to memorize many items. In the human case, it was reported by a psychologist[2] that the interference can be described by a power law:

$$t(M) = cM^D \ (c, D\text{:constants}, \quad D \sim 2) \ . \tag{1}$$

Now, it is a non-trivial question what kind of functional form of $t(M)$ is relevant to the interference of the feed forward network models. In the case of the back propagation model with a structure of the encoder[3], it was shown[4,5] in a good precision that the interference can be described by the same formula (1).

But at this point, you may well claim that the power law is a special feature of this model, that it does not have any universal meaning.

In order to see the universality of this power law, this time we performed an extensive simulation for wide variety of the gradient descent feed forward networks having various network structures, cost functions, node functions etc. As a result we conclude that the power law is fairly universal independent of the details of the network architecture.

In this short note, we just give a brief report on a part of the results of our simulation, revealing new phenomena found in the generalization. The detailed results and discussions on the implications of the results and on the possible mechanism underlying this power law will be given in the future publication.

Let us consider a feed forward network consisting of N_l layers. In the p-th layer, the neural computation is performed as $x_i^{(p)} = g\left(\sum w_{ij}^{(p)} \, x_j^{(p-1)} - b_i^{(p)}\right)$ where $x_i^{(p)}$ is an output of the i-th neuron in the p-th layer, $w_{ij}^{(p)}$ and $b_i^{(p)}$ are the connection weight and the bias. The input layer has N_{in} neurons, the output layer has N_{out} neurons. The network is trained by the usual gradient descent back propagation algorithm[6,7,8] with a learning coefficient η, a momentum coefficient α, and a cost function $E(\mathbf{x_{out}}, \mathbf{t})$.

The first observation of the power law was achieved[4,5] in the network structure of the encoder[3], namely $N_l = 3$ (one hidden layer) and $N_{out} = N_{in}$ with a sigmoid function as a node, and with a squared sum $E = \frac{1}{2}\sum(x_{i,out} - t_i)^2$ as a cost function.

The constants c and D appearing in (1) were experimentally measured and the exponent D was around two or less. On the other hand it was analytically proved[5] that D does not depend on α nor η.

But the exponent is not always the same value. The exponent does depend on the network structure. With a careful estimate of the error on D [†] this time, we have clarified how the exponent changes in response to the change of the network structure or other factors.

If N_h increases in the encoder model, the exponent is naively expected to go down since the interference must be weakened. However it was shown that this expectation is only correct for N_h not so large. Further increase of N_h makes the exponent go up again, finally the learning gets instable.

The word instable in this note generally means that the number of non-convergent or very slowly convergent events increases. We call such instability an inferior dominance since inferior learners dominate in this case.

Below we examine various modified systems. First of all we have tried truly multi layer ($N_l > 3$) models with arbitrary choices of the number of neurons on each layer. Then we certified that the power law holds well. We have even tried the network without hidden layer (the perceptron[1] [‡]) and verified that the law is not violated at all.

The perceptron is, however, somewhat special in that the exponent is relatively small. The exponents in the multilayer networks are generally found between 1.5 and 2, while the exponents observed in the perceptron were generally close to, but certainly greater than unity, see Figure 1.

Next we change the cost function from the squared sum to the relative entropy[9,10,11]. This completely modifies the structure of the dynamical phase space in which $w_{ij}^{(p)}$ makes motion, so that it is nontrivial whether or not the power law is preserved.

For not so large values of $M(\leq 10)$, however, log t-log M plot shows a linear behavior. But, for large values of M, we have a deviation from the power law. This is only case among

† The errors of the exponents in our simulation are 1 to 2% in general.

‡ In this note we use the word perceptron for the network model without a hidden layer.

all the modifications we have examined that shows a statistically significant deviation from the power law. The deviation looks a wavy one. A negligiblly small one of such has once been observed[4,5].

Thus far we have fixed the node function $g(x)$ to be a sigmoid function: $g(x) = 1/(1 + \mathrm{e}^{-x})$. Clearly, we need to try another node function to believe the universality in question. For this purpose we examine a two parameter deformation of the node function: $g(x, \beta, \rho) = 1/(1 + \mathrm{e}^{-\beta x})^{\rho}$ with positive constants β and ρ.

Under this deformation the power law is preserved again. But we find a new phenomenon with respect to the β dependence of the learning. We study how the learning time for a fixed M, $t(M)$ depends on β. Surprisingly, $t(M)$ is a power function of β for not so large values of β, see Figure 2. Namely we have found another non-trivial power law in the system.

Next we consider about the type of vectors. The input and the output vectors used so far have been of the type $(0, \cdots, 0, 1, 0, \cdots, 0)$ (except the perceptron) which we call an axis type. The generalization to the case of the non-axis type needs some explanation.

For each fixed architecture, we usually perform 30 learning runs starting with different initial conditions. From these data we determine the learning time to plot in two different ways. The first way is to use a mean value of these 30 learning times, the second one is to use the time up to which a fixed proportion r of the 30 runs terminate, in other words if $f(t)$ denotes the histogram of the 30 learning times, the learning time t is determined such that $\int_0^t f(t)dt = r \times 30$. Specifically we chose $r = 0.2$ in this note because the graph obtained in this way is insensitive to the choice of r. We call the former a mean plot, and the latter a superior plot, respectively. When the standard deviation of the learning times is not large, these two ways give the same graph.

The inferior dominance which is especially significant for non-axis cases disturb the plot. But it is refined by the use of the superior plot.

We should recall here that the superior plot allows a natural implementation as a neural network model, while the mean plot does not, as explained in refs. (4),(5). So we do not have to worry about the disturbed mean plot.

The superior plot is a plot in which the behavior of the inferior learners is neglected. The point is that the power law in question is the law for the superior learners. Therefore, we do not have any direct interest in the inferior learners.

Next we comment on the novelty appearing in the non-axis case. We have no direct interest in the inferiors. However, we cannot be completely indifferent to the existence of the inferiors. If the number of inferiors expands greatly (a strong inferior dominance), the superiors should be ruined (the number of superior learners vanishes) because the number of sessions (=the number of subnetworks in the realistic neural implementation) is finite.

This gives rise to a limit of the memory M_c. This is a new bound due to this generalization. For example, in the encoder case M_c is known to be equal to 2^{N_h} because of the binary encoding on the hidden layer. But if we use the non-axis vector, M_c is reduced much for this reason.

In addition to many modifications discussed above, we also examined other possibilities such as Fahlman's acceleration technique[12], a change of the criterion of the convergence, a change of the mode (batch/non-batch), etc. and any combinations of them. Some of the modifications have their own characteristic feature which we will report elsewhere. But the power law has been always correct.

In summary, the power law of the slowing pace of the memory, is a fairly universal law seen in the feed forward models with a gradient descent learning rule. Although the value of the exponents depends on the architecture, they are around two or less in almost all the variations we have examined. It seems highly non-trivial why the power law holds and why it is such universal. It is natural to consider there is some self similar or scale invariant structure underlying this power law. We believe that to seek such a structure will deepen our understanding of the feed forward network models.

Acknowledgements: The author thanks Dr.S.Saito for reading this manuscript carefully.

REFERENCES

1. For a general review see, J.Hertz, A.Krogh and R.G.Palmer,"Introduction to the Theory of Neural Computation", Addison Wesley (1992).
2. M.Foucault, Anée,Psychol.(1913)**vol 19**, pp. 218.
3. D.H.Ackley,G.E.Hinton, and T.J.Sejnowski (1985), "A learning algorithm for Bolzmann Machines".
4. T.Nakajima, N.Fuchikami, H.Câteau, and H.Nunokawa(1992), "Power Law in the Performance of the Human Memory and a Simulation with a Neural Network Model" in proceedings of the international workshop ISKIT'92 in Iizuka, Japan, [archive.cis.ohio-state.edu: pub/neuroprose/nakajima.power.tar.Z].
5. H.Câteau, T.Nakajima, H.Nunokawa, and N.Fuchikami,"Power Law in the Human Memory and in the Neural Network Model", University of Tokyo, preprint UT-616 (1992),[archive.cis.ohio-state.edu: pub/neuroprose/cateau.power.tar.Z].
6. D.E.Rumelhart, G.E.Hinton, and R.J.Williams (1986), Nature **vol.323**,pp.533.
7. B.Widrow and M.E.Hoff (1960), Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record Part **vol.4**, pp.94.
8. S.Amari, (1967), IEEE Trans. EC, **vol.16**,no.3,pp.299.
9. E.B.Baum and F.Wilczek (1988), "Supervised learning of probability distributions by neural networks", In neural network information processing systems (Denver 1987), ed.D.Z.Anderson, pp.52-61, New York:American Institute of Physics.
10. J.J.Hopfield (1987), "Learning Algorithms and probability distributions in feed forward and feed-back networks". Proceedings of the National Academy of Sciences, USA **vol.84**, pp.8429-8433.
11. S.A.Solla, E.Levin, and M.Fleisher (1988), "Accelerated learning in layered neural networks", Complex Systems, **vol.2**, pp.625-639.
12. S.E.Fahlman, (1989) "Fast-learning variations on back-propagation: An empirical Study". In Proceedings of the 1988 Connectionist Models Summer School (Pittsburg 1988), eds.D.Touretzky, G.Hinton, and T.Sejnowski, pp.38-51. San Mateo: Morgan Kaufmann.

Figure Captions

*Figure*1. A bilogplot of $t(M)$ vs M in the case of the perceptron with $N_{in} = N_{out} = 20$. The parameters are set as $\alpha = 0.001, \eta = 2.0$. The constants of the power law are observed to be $c = 36.1$, $D = 1.149 \pm 0.006$.

*Figure*2. The learning time for $M = 2$, $t(M = 2)$ is plotted versus β, where the deformed node function $g(x, \beta, \rho = 1)$ is used. The points are fitted well by the power function $60\beta^{-2.19}$. The network structure used here is 10-4-10, the parameters are set as $\alpha = 0.8, \eta = 0.8$.