Priors for Infinite Networks

Radford M. Neal

Technical Report CRG-TR-94-1
Department of Computer Science
University of Toronto
10 King's College Road
Toronto, Canada M5S 1A4

E-mail: radford@cs.toronto.edu

1 March 1994

Abstract

Bayesian inference begins with a prior distribution for model parameters that is meant to capture prior beliefs about the relationship being modeled. For multilayer perceptron networks, where the parameters are the connection weights, the prior lacks any direct meaning — what matters is the prior over functions computed by the network that is implied by this prior over weights. In this paper, I show that priors over weights can be defined in such a way that the corresponding priors over functions reach reasonable limits as the number of hidden units in the network goes to infinity. When using such priors, there is thus no need to limit the size of the network in order to avoid "overfitting". The infinite network limit also provides insight into the properties of different priors. A Gaussian prior for hidden-to-output weights results in a Gaussian process prior for functions, which can be smooth, Brownian, or fractional Brownian, depending on the hidden unit activation function and the prior for input-to-hidden weights. Quite different effects can be obtained using priors based on non-Gaussian stable distributions. In networks with more than one hidden layer, a combination of Gaussian and non-Gaussian priors appears most interesting.

1 Introduction

Bayesian inference is an alternative to conventional training for neural networks, with advantages that include the automatic determination of the appropriate degree of "regularization", the quantification of the uncertainty in predictions, and the possibility of comparison with other models (MacKay 1991, 1992; Buntine and Weigend 1991; Neal 1992, 1993). The starting point for Bayesian inference is a prior distribution over the model parameters, which for a multilayer perceptron ("backprop") network are the connection weights and unit biases. In the Bayesian framework, this prior distribution is meant to capture our prior beliefs about the relationship we are modeling with the network. When training data is obtained, the prior distribution is updated to a posterior parameter distribution, which is then used to make predictions for test cases.

A problem with this approach is that the meaning of the weights in a neural network is obscure, making it hard to design a prior distribution that expresses our beliefs. Furthermore, a network with a small number of hidden units can represent only a limited set of functions, which will generally not include the true function. Hence our actual prior belief will usually be that the model is simply wrong.

I propose to address these problems by focusing on the limit as the number of hidden units in the network approaches infinity. Several workers (e.g. Funahashi 1989) have shown that in this limit a network with one layer of hidden units can approximate any continuous function defined on a compact domain arbitrarily closely. An infinite network will thus be a reasonable "non-parametric" model for many problems. Furthermore, it turns out that in the infinite network limit we can easily analyse the nature of the priors over functions that result when we use certain priors for the network weights. This allows us to sensibly select an appropriate prior based on our knowledge of the characteristics of the problem.

In practice, of course, we will have to use networks with only a finite number of hidden units. The hope is that our computational resources will allow us to train a network of sufficient size that its characteristics are close to those of an infinite network.

Note that in this approach one does *not* restrict the size of the network based on the size of the training set — rather, the only limiting factors are the size of the computer used and the time available for training. Experience training networks by methods such as maximum likelihood might lead one to expect a large network to "overfit" a small training set, and perform poorly on later test cases. This does not occur with Bayesian learning, provided the width of the prior used for hidden-to-output weights is scaled down in a simple fashion as the number of hidden units increases, as required for the prior to reach a limit.

This assumes, of course, that the implementation of Bayesian inference used produces the mathematically correct result. Achieving this is not trivial. Methods based on making a Gaussian approximation to the posterior (MacKay 1991, 1992; Buntine and Weigend 1991) may break down as the number of hidden units becomes large. Markov chain Monte Carlo methods (Neal 1992, 1993) produce the correct answer eventually, but may sometimes fail to reach the true posterior distribution in a reasonable length of time. In this paper, I do not discuss such computational issues; my aim instead is to gain insight through theoretical analysis, done with varying degrees of rigour, and by sampling from the prior, which is much easier than sampling from the posterior.

For most of this paper, I consider only networks that take I real-valued inputs, x_i, and produce O real-valued outputs given by functions $f_k(x)$, computed using a layer of H sigmoidal hidden units with values $h_j(x)$:

$$f_k(x) = b_k + \sum_{j=1}^{H} v_{jk} h_j(x) \tag{1}$$

$$h_j(x) = \tanh\Big(a_j + \sum_{i=1}^{I} u_{ij} x_i\Big) \tag{2}$$

At times, I will consider networks in which the tanh activation function is replaced by a step function returning -1 for negative arguments and $+1$ for positive arguments. (Learning for networks with step-function hidden units is computationally difficult, but these networks are sometimes simpler to analyse.) Networks with more than one hidden layer are discussed in the final section.

2 Priors converging to Gaussian processes

Most past work on Bayesian inference for neural networks has used independent Gaussian distributions as the priors for network weights and biases. In this section, I will investigate the properties of priors in which the hidden-to-output weights, v_{jk}, and the output biases, b_k, have Gaussian distributions, with standard deviations of σ_v and σ_b. It will turn out that as the number of hidden units increases, the prior over functions implied by such priors converges to a Gaussian process. These priors can have smooth, Brownian, or fractional Brownian properties, as determined by the covariance function of the Gaussian process.

For the priors that I consider in detail, the input-to-hidden weights, u_{ij}, and the hidden unit biases, a_j, also have Gaussian distributions, with standard deviations σ_u and σ_a, though for the fractional Brownian priors, σ_u and σ_a are not fixed, but depend on the value of common parameters associated with each hidden unit.

2.1 Limits for Gaussian priors

To determine what prior over functions is implied by a Gaussian prior for network parameters, let us look first at the prior distribution of the value of output unit k when the network inputs are set to some particular values, $x^{(1)}$ — that is we look at the prior distribution of $f_k(x^{(1)})$ that is implied by the prior distributions for the weights and biases.

From equation (1), we see that $f_k(x^{(1)})$ is the sum of a bias and the weighted contributions of the H hidden units. Under the prior, each term in this sum is independent, and the contributions of the hidden units all have identical distributions. The expected value of each hidden unit's contribution is zero: $E[v_{jk}h_j(x^{(1)})] = E[v_{jk}]E[h_j(x^{(1)})] = 0$, since v_{jk} is independent of a_j and the u_{ij} (which determine $h_j(x^{(1)})$), and $E[v_{jk}]$ is zero by hypothesis. The variance of the contribution of each hidden unit is finite: $E[(v_{jk}h_j(x^{(1)})^2] = E[v_{jk}^2]E[h_j(x^{(1)})^2] = \sigma_v^2 E[h_j(x^{(1)})^2]$, which must be finite since $h_j(x^{(1)})$ is bounded. Defining $V(x^{(1)}) = E[h_j(x^{(1)})^2]$, which is the same for all j, we can conclude by the Central Limit Theorem that for large H the total contribution of the hidden units to the value of $f_k(x^{(1)})$ becomes Gaussian with variance $H\sigma_v^2 V(x^{(1)})$. The bias, b_k, is also Gaussian, of variance σ_b^2, so for large H the prior distribution of $f_k(x^{(1)})$ is Gaussian of variance $\sigma_b^2 + H\sigma_v^2 V(x^{(1)})$.

Accordingly, to obtain a well-defined limit for the prior distribution of the value of the function at any particular point, we need only scale the prior variance of the hidden-to-output weights according to the number of hidden units, setting $\sigma_v = \omega_v H^{-1/2}$, for some fixed ω_v. The prior for $f_k(x^{(1)})$ then converges to a Gaussian of mean zero and variance $\sigma_b^2 + \omega_v^2 V(x^{(1)})$ as H goes to infinity.

Adopting this scaling for σ_v, we can investigate the prior joint distribution of the values of output k for several values of the inputs — that is, the joint distribution of $f_k(x^{(1)}), \ldots, f_k(x^{(n)})$, where $x^{(1)}, \ldots, x^{(n)}$ are the particular input values we chose to look at. An argument paralleling that above shows that as H goes to infinity this prior joint distribution converges to a multivariate Gaussian, with means of zero, and with covariances of

$$\begin{aligned} E[f_k(x^{(p)})f_k(x^{(q)})] &= \sigma_b^2 + \sum_j \sigma_v^2 E[h_j(x^{(p)})h_j(x^{(q)})] && (3) \\ &= \sigma_b^2 + \omega_v^2 C(x^{(p)}, x^{(q)}) && (4) \end{aligned}$$

where $C(x^{(p)}, x^{(q)}) = E[h_j(x^{(p)})h_j(x^{(q)})]$, which is the same for all j. Distributions over functions of this sort, in which the joint distribution of the values of the function at any finite number of points is multivariate Gaussian, are known as *Gaussian processes*; they arise in many contexts, including spatial statistics (Ripley 1981), computer vision (Szeliski 1989), and computer graphics (Peitgen and Saupe 1988).

The prior covariances between the values of output k for different values of the inputs are in general not zero, which is what allows learning to occur. Given values for $f_k(x^{(1)}), \ldots, f_k(x^{(n-1)})$, we could explicitly find the predictive distribution for the value of output k for case n by conditioning on these known values to produce a Gaussian distribution for $f_k(x^{(n)})$. This procedure may indeed be of practical interest, though it does require evaluation of $C(x^{(p)}, x^{(q)})$ for all $x^{(p)}$ in the training set and $x^{(q)}$ in the training and test sets, which would likely have to be done by numerical integration.

The joint distribution for the values of *all* the outputs of the network for some selection of values for inputs will also become a multivariate Gaussian in the limit as the number of hidden units goes to infinity. It is easy to see, however, that the covariance between $f_{k_1}(x^{(p)})$ and $f_{k_2}(x^{(q)})$ is zero whenever $k_1 \neq k_2$, since the weights into different output units are independent under the prior. Since zero covariance implies independence for Gaussian distributions, knowing the values of one output for various inputs does *not* tell us anything about the values of other outputs, at these or any other input points. It will make no difference whether we train one network to produce two outputs, or instead use the same data to train two networks, each with one output. (I assume here that these outputs are not linked in some other fashion, such as by the assumption that their values are observed with a common, but unknown, level of noise.)

This independence of different outputs is perhaps surprising, since the outputs are computed using shared hidden units. However, with the Gaussian prior used here, the values of the hidden-to-output weights all go to zero as the number of hidden units goes to infinity. The output functions are built up from a large number of contributions from hidden units, with each contribution being of negligible significance by itself. Hidden units computing common features of the input that would be capable of linking the outputs are therefore not present. Dependencies between outputs could be introduced by making the weights to various outputs from one hidden unit be dependent, but if these weights have Gaussian priors, they can be dependent only if they are correlated. Accordingly, it is not possible to define a Gaussian-based prior expressing the idea that two outputs might show a large change in the same input region, the location of this region being unknown *a priori*, without also fixing whether the changes in the two outputs have the same or opposite sign.

The results in this section in fact hold more generally for any hidden unit activation function that is bounded, and for any prior on input-to-hidden weights and hidden unit biases (the u_{ij} and a_j) in which the weights and biases for different hidden units are independent and identically distributed. The results also apply when the prior for hidden-to-output weights is not Gaussian, as long as the prior has zero mean and finite variance.

2.2 Priors that lead to smooth and Brownian functions

I will start the detailed examination of Gaussian process priors by considering those that result when the input-to-hidden weights and hidden biases have Gaussian distributions. These turn out to give locally Brownian priors if step-function hidden units are used, and priors over smooth functions if tanh hidden units are used. For simplicity, I at first discuss only networks having a single input, but Section 2.5 will show that the results apply with little change to networks with any number of inputs.

To begin, consider a network with one input in which the hidden units compute a step function changing from -1 to $+1$ at zero. In this context, the values of the input weight, u_{1j}, and bias, a_j, for hidden unit j are significant only in that they determine the point in the input space where that hidden unit's step occurs, namely $-a_j/u_{1j}$. When the weight and bias have independent Gaussian prior distributions with standard deviations σ_u and σ_a, the prior distribution of this step-point is Cauchy, with a width parameter of σ_a/σ_u.

Figure 1 shows functions drawn from the prior distributions for two such networks, one network with 300 hidden units and one with 10 000 hidden units. Note that the general nature of the functions is the same for the two network sizes, but the functions from the larger network have more fine detail. This illustrates that the prior over functions is reaching a limiting distribution as H increases.

(In this and subsequent figures, the functions shown are not necessarily the first that were generated. Some selection was done in order to ensure that typical features are displayed, and to find pairs of functions that fit together nicely on a graph, without overlapping too much. In all cases, the functions shown were selected from a sample of no more than ten functions drawn from the prior.)

The variation in these functions is concentrated in the region around $x = 0$, with a width of roughly σ_a/σ_u. Within this region, the function is locally Brownian in character, as a consequence of being built up from the many small, independent steps contributed by the hidden units. Far from $x = 0$, the functions become almost constant, since few hidden units have their steps that far out. For the remainder of this paper, I will confine my attention to the properties of functions in their central regions, where all points have approximately equal potential for being influenced by the hidden units.

In contrast, when tanh hidden units are used, the functions generated are smooth. This can be seen by noting that all the derivatives (to any order) of the value of a hidden unit with respect to the inputs are polynomials in the hidden unit value and the input-to-hidden weights. These derivatives therefore have finite expectation and finite variance, since the hidden unit values are bounded, and the

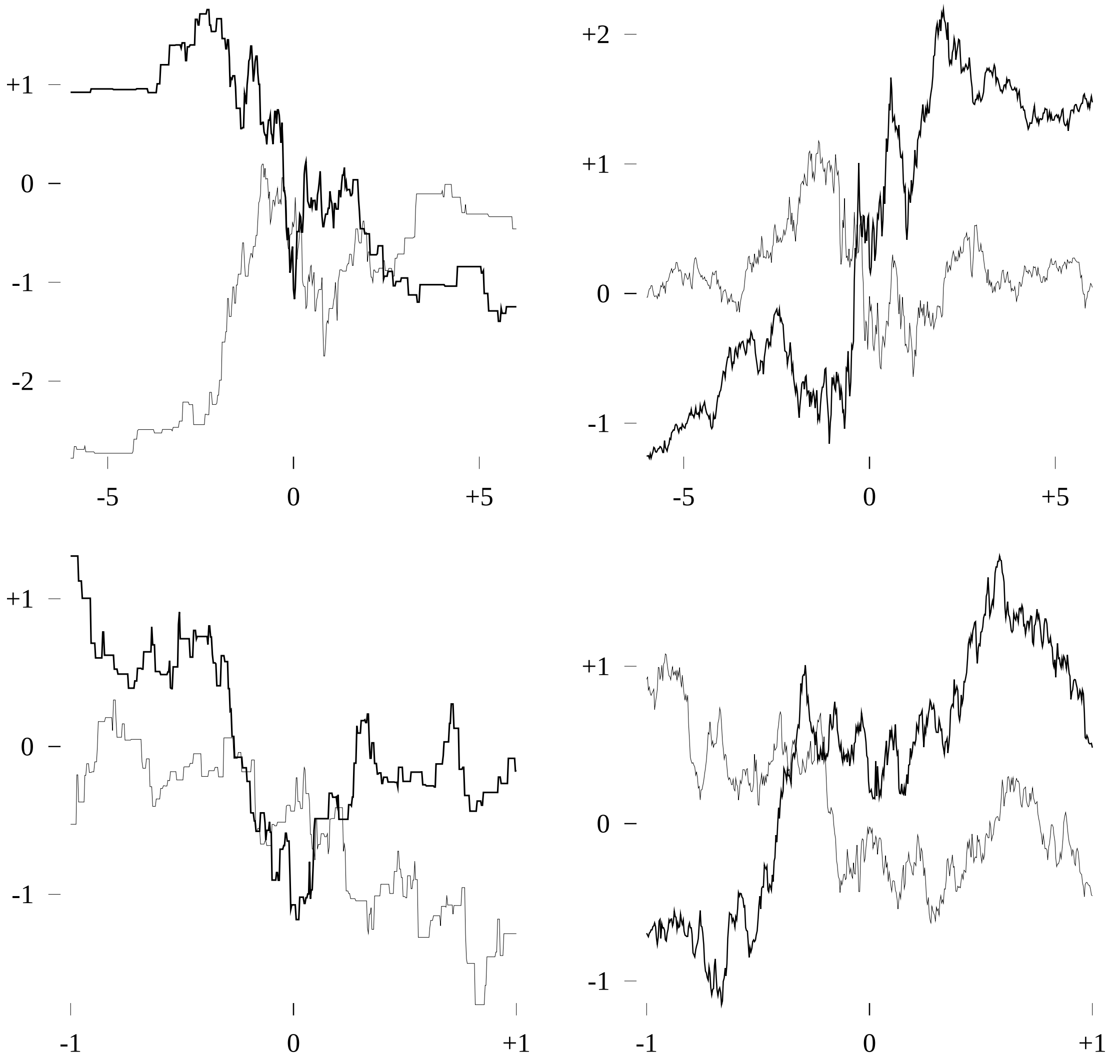

Figure 1: Functions drawn from Gaussian priors for networks with step-function hidden units. The two functions shown on the left are from a network with 300 hidden units, the two on the right from a network with 10 000 hidden units. In both cases, $\sigma_a = \sigma_u = \sigma_b = \omega_v = 1$. The upper plots show the overall shape of each function; the lower plots show the central area in more detail.

weights are from Gaussian distributions, for which moments of all orders exist. At scales greater than about $1/\sigma_u$, however, the functions exhibit the same Brownian character that was seen with step-function hidden units.

The size of the central region where the properties of these functions are approximately uniform is roughly $(\sigma_a + 1)/\sigma_u$. To see this, note that when the input weight is u, the distribution of the point where the hidden unit value crosses zero is Gaussian with standard deviation $\sigma_a/|u|$. The influence of a hidden unit with this

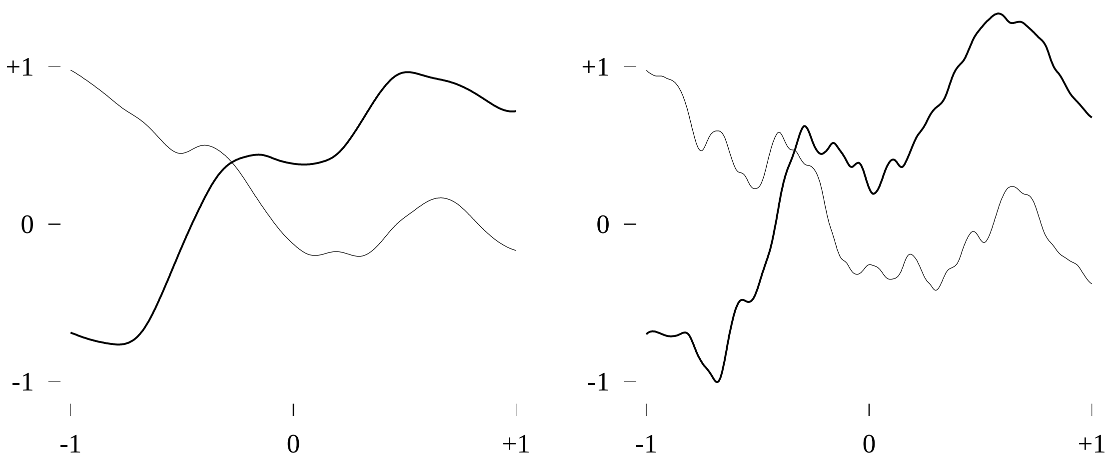

Figure 2: Functions drawn from Gaussian priors for a network with 10 000 tanh hidden units. Two functions drawn from a prior with $\sigma_u = 5$ are shown on the left, two from a prior with $\sigma_u = 20$ on the right. In both cases, $\sigma_a/\sigma_u = 1$ and $\sigma_b = \omega_v = 1$. The functions with different σ_u were generated using the same random number seed, the same as that used to generate the functions in the lower-right of Figure 1. This allows a direct evaluation of the effect of changing σ_u. (Use of a step function is equivalent to letting σ_u go to infinity, while keeping σ_a/σ_u fixed.)

input weight extends a distance of about $1/|u|$, however, so points within about $(\sigma_a + 1)/|u|$ of the origin are potentially influenced by hidden units with input weights of this size. Since the probability of obtaining a weight of size $|u|$ declines exponentially beyond $|u| = \sigma_u$, the functions will have similar properties at all points within a distance of about $(\sigma_a + 1)/\sigma_u$ of the origin.

Functions drawn from priors for networks with tanh hidden units are shown in Figure 2.

2.3 Covariance functions of Gaussian priors

A Gaussian process can be completely characterized by the mean values of the function at each point, zero for the network priors discussed here, along with the covariance of the function value at any two points, given by equation (4). The difference between priors that lead to locally smooth functions and those that lead to locally Brownian functions is reflected in the local behaviour of their covariance functions. From equation (4), we see that this is directly related to the covariance of the values of a hidden unit at nearby input points, $C(x^{(p)}, x^{(q)})$, which in turn can be expressed as

$$\begin{aligned} C(x^{(p)}, x^{(q)}) &= \tfrac{1}{2}\left(V(x^{(p)}) + V(x^{(q)}) - E[(h(x^{(p)}) - h(x^{(q)}))^2]\right) && (5) \\ &= V - \tfrac{1}{2} D(x^{(p)}, x^{(q)}) && (6) \end{aligned}$$

where $V(x^{(p)}) \approx V \approx V(x^{(q)})$, for nearby $x^{(p)}$ and $x^{(q)}$, and $D(x^{(p)}, x^{(q)})$ is the expected squared difference between the values of a hidden unit at $x^{(p)}$ and $x^{(q)}$.

For step-function hidden units, $(h(x^{(p)})-h(x^{(q)}))^2$ will be either 0 or 4, depending on whether the values of the hidden unit's bias and incoming weight result in the step being located between $x^{(p)}$ and $x^{(q)}$. Since the location of this step will be approximately uniform in the local vicinity, the probability of the step occurring between $x^{(p)}$ and $x^{(q)}$ will rise proportionally with the separation of the points, giving

$$D(x^{(p)}, x^{(q)}) \quad \sim \quad |x^{(p)} - x^{(q)}| \tag{7}$$

where $\sim$ indicates proportionality for nearby points. This behaviour is characteristic of Brownian motion.

For networks with tanh hidden units, with Gaussian priors for the bias and incoming weight, we have seen that the functions are smooth. Accordingly, for nearby $x^{(p)}$ and $x^{(q)}$ we will have

$$D(x^{(p)}, x^{(q)}) \quad \sim \quad |x^{(p)} - x^{(q)}|^2 \tag{8}$$

We can get a rough idea of the behaviour of $D(x^{(p)}, x^{(q)})$ for all points within the central region as follows. First, fix the input-to-hidden weight, u, and consider the expectation of $(h(x-s/2) - h(x+s/2))^2$ with respect to the prior distribution of the bias, a, which is Gaussian with standard deviation σ_a. With u fixed, the point where the hidden unit's total input crosses zero will have a prior distribution that is Gaussian with standard deviation $\sigma_a/|u|$, giving a probability density for the zero crossing to occur at any point in the central region of around $|u|/\sigma_a$. We can now distinguish two cases. When $|u| \gtrsim 1/s$, the transition region over which the hidden unit's output changes from -1 to $+1$, whose size is about $1/|u|$, will be small compared to s, and we can consider that $(h(x-s/2) - h(x+s/2))^2$ will either be 0 or 4, depending on whether the total input to the hidden unit crosses zero between $x-s/2$ and $x+s/2$, which occurs with probability around $(|u|/\sigma_a)s$. When $|u| \lesssim 1/s$, $(h(x-s/2) - h(x+s/2))^2$ will be about $(|u|s)^2$ if the interval $[x-s/2, x+s/2]$ is within the transition region, while otherwise it will be nearly zero. The probability of $[x-s/2, x+s/2]$ lying in the transition region will be about $(|u|/\sigma_a)(1/|u|) = 1/\sigma_a$. Putting all this together, we get

$$E_a[(h(x-s/2) - h(x+s/2))^2] \quad \approx \quad \begin{cases} c_1(|u|/\sigma_a)s & \text{if } |u| \gtrsim 1/s \\ c_2(|u|^2/\sigma_a)s^2 & \text{if } |u| \lesssim 1/s \end{cases} \tag{9}$$

where c_1, c_2, ... are constants of order unity. Taking the expectation with respect to the symmetrical prior for u, with density $p(u)$, we get

$$E_{a,u}[(h(x-s/2) - h(x+s/2))^2] \quad \approx \quad \frac{c_1 s}{\sigma_a}\int_{1/s}^{\infty} u\, p(u)\, du \;+\; \frac{c_2 s^2}{\sigma_a}\int_0^{1/s} u^2\, p(u)\, du \tag{10}$$

Finally, if we crudely approximate the Gaussian prior for u by a uniform distribution over $[-\sigma_u, +\sigma_u]$, with density $p(u) = 1/2\sigma_u$, we get

$$\begin{aligned} D(x-s/2, x+s/2) &= E_{a,u}[(h(x-s/2) - h(x+s/2))^2] \\ &\approx \frac{1}{\sigma_a} \begin{cases} c_3\, \sigma_u^2 s^2 & \text{if } s \lesssim 1/\sigma_u \\ c_4\, \sigma_u s \;+\; c_5/\sigma_u s & \text{if } s \gtrsim 1/\sigma_u \end{cases} \end{aligned} \qquad (11)$$

Thus these functions are smooth on a small scale, but when viewed on scales significantly larger than $1/\sigma_u$, they have a Brownian nature characterized by $D(x-s/2, x+s/2)$ being proportional to s.

2.4 Fractional Brownian priors

It is natural to wonder whether a prior on the weights and biases going into hidden units can be found for which the resulting prior over functions has *fractional Brownian* properties (Falconer 1990, Section 16.2), characterized by

$$D(x^{(p)}, x^{(q)}) \;\sim\; |x^{(p)} - x^{(q)}|^\eta \qquad (12)$$

As above, values of $\eta = 2$ and $\eta = 1$ correspond to smooth and Brownian functions. Functions with intermediate properties are obtained when $1 < \eta < 2$; functions "rougher" than Brownian motion are obtained when $0 < \eta < 1$.

One way to achieve these effects would be to change the hidden unit activation function from $\tanh(z)$ to $\text{sign}(z)|z|^{(\eta-1)/2}$ (Peitgen and Saupe 1988, Sections 1.4.1 and 1.6.11). However, the unbounded derivatives of this activation function would pose problems for gradient-based learning methods. I will describe a method of obtaining fractional Brownian functions with $1 < \eta < 2$ from networks with tanh hidden units by altering the priors for the hidden unit bias and input weights.

To construct this fractional Brownian prior, we associate with hidden unit j a value, A_j, that controls the magnitude of that hidden unit's incoming weights and bias. Given A_j, we let the incoming weights, u_{ij}, have independent Gaussian distributions with standard deviation $\sigma_u = A_j\omega_u$, and we let the bias, a_j, have a Gaussian distribution with standard deviation $\sigma_a = A_j\omega_a$. We give the A_j themselves independent prior distributions with probability density $p(A) \propto A^{-\eta} \exp(-(\eta-1)/2A^2)$, where $\eta > 1$, which corresponds to a Gamma distribution for $1/A_j^2$. Note that if we integrate over A_j to obtain a direct prior for the weights and biases, we find that the weights and biases are no longer independent, and no longer have Gaussian distributions.

To picture why this setup should result in a fractional Brownian prior for the functions computed by the network, consider that when A_j is large, $h_j(x)$ is likely to be almost a step function, since σ_u will be large. (A_j does not affect the distribution of the point where the step occurs, however, since this depends only on σ_a/σ_u.) Such near-step-functions produced by hidden units with A_j greater than some limit will contribute in a Brownian fashion to $D(x^{(p)}, x^{(q)})$, with the contribution rising in direct proportion to the separation of $x^{(p)}$ and $x^{(q)}$. However, as this separation increases, the value of A_j that is sufficient for the hidden unit to behave as step-function in this context falls, and the number of hidden units that effectively behave as step functions rises. The contribution of such hidden units to $D(x^{(p)}, x^{(q)})$ will therefore increase faster than for a Brownian function. The other hidden units with small A_j will also contribute to $D(x^{(p)}, x^{(q)})$, quadratically with separation, but for nearby points their contribution will be dominated by that of the units with large A_j, if that contribution is sub-quadratic.

We can see this in somewhat more detail by substituting $\sigma_u = A_j\omega_u$ and $\sigma_a = A_j\omega_a$ in equation (11), obtaining

$$E_{a,u}[(h(x-s/2) - h(x+s/2))^2] \;\approx\; \frac{1}{\omega_a}\begin{cases} c_3\, A_j\omega_u^2 s^2 & \text{if } A_j \lesssim 1/s\omega_u \\ c_4\,\omega_u s \;+\; c_5/A_j^2\omega_u s & \text{if } A_j \gtrsim 1/s\omega_u \end{cases} \tag{13}$$

Integrating with respect to the prior for A_j, we get

$$\begin{aligned} &D(x-s/2, x+s/2) \\ &\quad\approx\; \frac{c_3\,\omega_u^2 s^2}{\omega_a}\int_0^{1/\omega_u s} A\, p(A)\, dA \;+\; \frac{c_4\,\omega_u s}{\omega_a}\int_{1/\omega_u s}^{\infty} p(A)\, dA \;+\; \frac{c_5}{\omega_a\omega_u s}\int_{1/\omega_u s}^{\infty} A^{-2} p(A)\, dA \end{aligned} \tag{14}$$

The mode of $p(A)$ is at $((\eta-1)/\eta)^{1/2}$. Before this point $p(A)$ drops rapidly, and can be approximated as being zero; after this point, it drops as $A^{-\eta}$. The integrals above can thus be approximated as follows, for $\eta \neq 2$:

$$D(x-s/2, x+s/2) \;\approx\; \frac{1}{\omega_a}\begin{cases} c_6\,\omega_u^\eta s^\eta \;+\; c_7\,\omega_u^2 s^2 & \text{if } s \lesssim (\eta/(\eta-1))^{1/2}/\,\omega_u \\ c_8\,\omega_u s \;+\; c_9/\omega_u s & \text{if } s \gtrsim (\eta/(\eta-1))^{1/2}/\,\omega_u \end{cases} \tag{15}$$

When $1 < \eta < 2$, the s^η term will dominate for small s, and the function will have fractional Brownian properties; when $\eta > 2$, the s^2 term will dominate, producing a smooth function; $\eta = 2$ is a special case, for which $D(x{-}s/2, x{+}s/2) \sim s^2 \log(1/s)$.

Fractional Brownian functions drawn from these priors are shown in Figure 3. Figure 4 shows the behaviour of $D(x-s/2, x+s/2)$ for the same priors, as well as for the priors used in Figures 1 and 2.

2.5 Networks with more than one input

The priors discussed here have analogous properties when used for networks with several inputs. In particular, the value of the network function along any line in input space has the same properties as those described above for a network with a

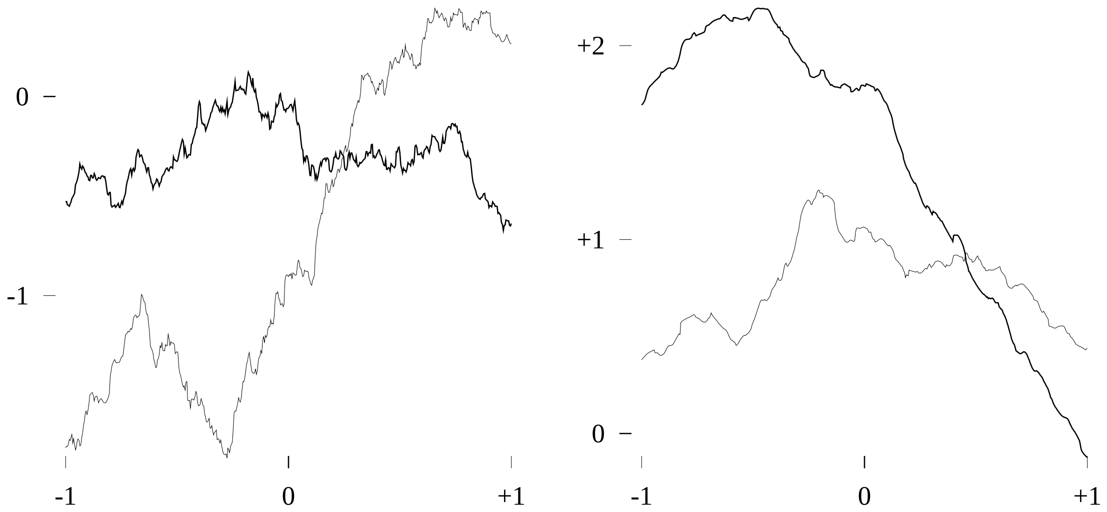

Figure 3: Functions drawn from fractional Brownian priors for a network with 10 000 tanh hidden units. Two functions drawn from a prior with $\eta = 1.3$ are shown on the left, two from a prior with $\eta = 1.7$ on the right. In both cases, $\omega_a = \omega_u = \sigma_b = \omega_v = 1$.

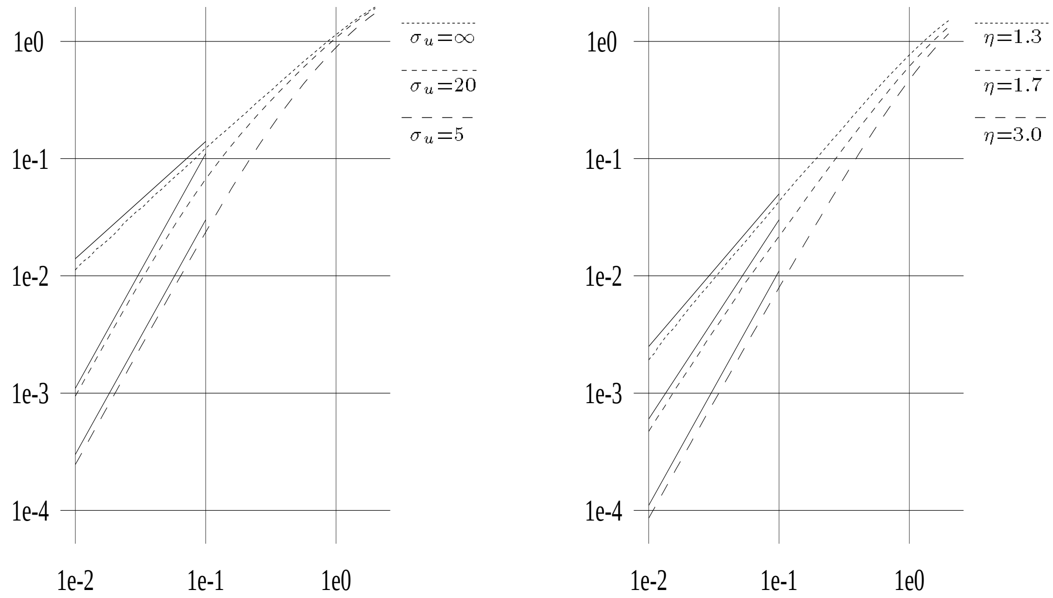

Figure 4: Behaviour of $D(x - s/2, x + s/2)$ as s varies for Brownian, smooth, and fractional Brownian functions. The plots on the left are for the Brownian prior used in Figure 1, and the smooth priors used in Figure 2; those on the right are for the fractional Brownian priors used in Figure 3, as well as for a similar prior on the A_j with $\eta = 3$, which leads to a smooth function. All values are for $x = 0.2$. They were computed by Monte Carlo integration using a sample of 100 000 values drawn from the prior for the bias and weight into a hidden unit; the values are hence subject to a small amount of noise. Note that both scales are logarithmic, so that a function proportional to s^η should appear as a straight line of slope η. Straight lines of the expected slopes are shown beside the curves to demonstrate their convergence for small s.

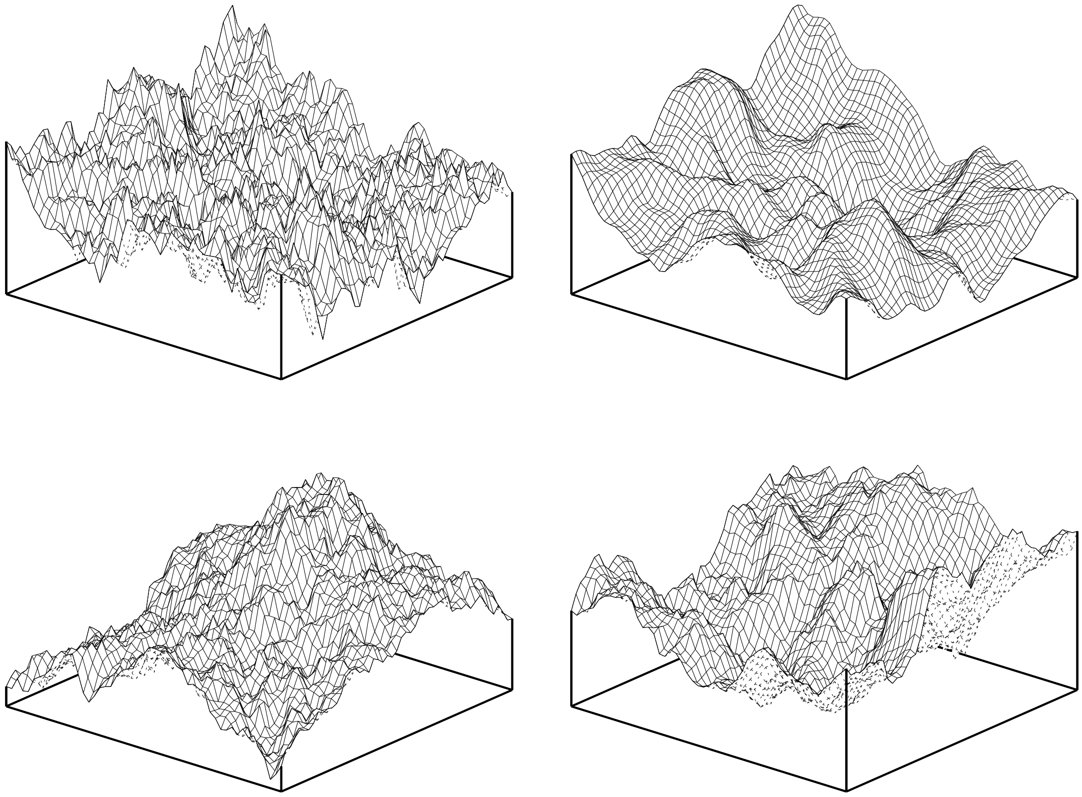

Figure 5: Functions of two inputs drawn from Gaussian priors. The function in the upper left is from a network with 10 000 step-function hidden units, that in the upper right from the corresponding network with tanh hidden units, using the same random number seed. In both cases, $\sigma_a = \sigma_u = 10$. The two lower functions are from networks with tanh hidden units, using fractional Brownian priors. The function in the lower left has $\eta = 1.3$, that in the lower right $\eta = 1.7$. In both cases, $\omega_a = \omega_u = 1$. The plots show the input region from -1 to $+1$.

single input. Since all the priors discussed are invariant with respect to rotations of the input space, we may confine our attention to lines obtained by varying only one of the inputs, say the first. Rewriting equation (2) as

$$h_j(x) \;=\; \tanh\Big(u_{1j}x_1 \;+\; a_j \;+\; \sum_{i=2}^{I} u_{ij}x_i\Big) \qquad (16)$$

we see that when $x_2, \ldots, x_I$ are fixed, they act simply to increase the variance of the effective bias. This merely spreads the variation in the function over a larger range of values for x_1.

Figure 5 shows functions of two inputs drawn from Brownian, smooth, and fractional Brownian priors.

2.6 Hierarchical models

Often, our prior knowledge will be too unspecific to fix values for σ_b, ω_v, σ_a (or ω_a), and σ_u (or ω_u), even if we have complete insight into their effects on the prior. We may then wish to treat these values as unknown *hyperparameters*, giving them higher-level prior distributions that are rather broad. One benefit of such a *hierarchical model* is that the degree of "regularization" that is appropriate for the task can be determined automatically from the data (MacKay 1991, 1992).

Insight gained into the nature of the prior distributions produced for given values of the hyperparameters is still useful even when we do not plan to use such information to fix the hyperparameters to particular values, in that such insight allows us to judge whether the range of possibilities offered by a hierarchical model is adequate for our problem. For example, the results above concerning fractional Brownian motion suggest that it might be useful to make η a hyperparameter, to allow the fractional Brownian character of the function to be determined by the data.

3 Priors converging to non-Gaussian stable distributions

Although we have seen that a variety of interesting priors over functions can be produced using Gaussian priors for hidden-to-output weights and output biases, these priors are in some respects disappointing.

One reason for this is that it may be possible to implement Bayesian inference for these priors, or for other Gaussian process priors with similar properties, using standard methods based directly on the covariance function, without any need for an actual network. We may thus need to look at different priors if Bayesian neural networks are to significantly extend the range of models available. (On the other hand, it is possible that the particular covariance structure created using a network might be of special interest, or that control of the covariance via hyperparameters might most conveniently be done in a network formulation.)

Furthermore, as mentioned earlier, with Gaussian priors the contributions of individual hidden units are all negligible, and consequently, these units do not represent "hidden features" that capture important aspects of the data. If we wish the network to do this, we need instead a prior with the property that even in the limit of infinitely many hidden units, there are some individual units that have non-negligible output weights. Such priors can indeed be constructed, using prior distributions for the weights from hidden to output units that do not have finite variance.

3.1 Limits for priors with infinite variance

The theory of *stable distributions* (Feller, 1966, Section VI.1) provides the basis for analysing the convergence of priors in which hidden-to-output weights have infinite variance. If the random variables $Z_1, \ldots, Z_n$ are independent, and all the Z_i have the same symmetric stable distribution of index α, then $(Z_1 + \cdots + Z_n)/n^{1/\alpha}$ has the same distribution as the Z_i. Such symmetric stable distributions exist for $0 < \alpha \leq 2$, and for each index they form a single family, varying only in width. The symmetric stable distributions of index $\alpha = 2$ are the Gaussians of varying standard deviations; those of index $\alpha = 1$ are the Cauchy distributions of varying widths; the densities for the symmetric stable distributions with most other indexes have no convenient forms.

If independent variables $Z_1, \ldots, Z_n$ each have the same distribution, one that is in the *normal domain of attraction* of the family of symmetric stable distributions of index α, then the distribution of $(Z_1 + \cdots + Z_n)/n^{1/\alpha}$ approaches such a stable distribution as n goes to infinity. All distributions with finite variance are in the normal domain of attraction of the Gaussian. Distributions with tails that (roughly speaking) have densities that decline as $z^{-(\alpha+1)}$, with $0 < \alpha < 2$ are in the normal domain of attraction of the symmetric stable distributions of index α (Feller, 1966, Sections IX.8 and XVII.5).

We can define a prior on network weights in such as fashion that the resulting prior on the value of a network output for a particular input converges to a non-Gaussian symmetric stable distribution as the number of hidden units, H, goes to infinity. This is done by using independent, identical priors for the hidden-to-output weights, v_{jk}, with a density whose tails go as $v_{jk}^{-(\alpha+1)}$, with $\alpha < 2$. For all the examples in this paper, I use a t-distribution with density proportional to $(1 + v_{jk}^2/\alpha\sigma_v^2)^{-(\alpha+1)/2}$. The prior distribution of the contribution of a hidden unit to the output will have similar tail behaviour, since the hidden unit values are bounded. Accordingly, if we scale the width parameter of the prior for hidden-to-output weights as $\sigma_v = \omega_v H^{-1/\alpha}$, the prior for the total contribution of all hidden units to the output value for a particular input will converge to a symmetric stable distribution of index α. If the prior for the output bias is a stable distribution of this same index, the value of the output unit for that input, which is the sum of the bias and the hidden unit contributions, will have a prior distribution in this same stable family. (In practice, it may not be convenient for the bias to have such a stable distribution as its prior, but using a different prior for the bias will have only a minor effect.)

To gain insight into the nature of such priors, we can look at the expected number of hidden-to-output weights lying in some small interval, $[w, w + \epsilon]$, in the limit as H goes to infinity. For a given H, the number of weights in this interval using the prior that is scaled down by $H^{-1/\alpha}$ will be the same as the number that would be in the interval $[wH^{1/\alpha},\ wH^{1/\alpha} + \epsilon H^{1/\alpha}]$ if the unscaled prior were used. As H increases, this interval moves further and further into the tail of the unscaled prior distribution, where, by construction, the density goes down as $v^{-(\alpha+1)}$. The probability that a particular weight will lie in this small interval is thus proportional to $\epsilon H^{1/\alpha}(wH^{1/\alpha})^{-(\alpha+1)} = \epsilon w^{-(\alpha+1)}H^{-1}$. The expected total number of weights from all H hidden units that lie in the interval $[w,\ w+\epsilon]$ is therefore proportional to $\epsilon w^{-(\alpha+1)}$, in the limit as H goes to infinity.

Thus, whereas for Gaussian priors, all the hidden-to-output weights go to zero as H goes to infinity, for priors based on symmetric stable distributions of index $\alpha < 2$, some of the hidden units in an infinite network have output weights of significant size, allowing them to represent "hidden features". As an aside, the fact that the number of weights of each size has non-zero expectation means that the prior can be given an alternative formulation in terms of a Poisson process for hidden-to-output weights. (Note that though such a process could be defined for any α, it gives rise to a well-defined prior over functions only if $0 < \alpha < 2$.)

The above priors based on non-Gaussian stable distributions lead to prior distributions over functions in which the functions computed by different output units are independent, in the limit as H goes to infinity, just as was the case for Gaussian priors. This comes about because the weights to the various output units from a single hidden unit are independent. As H goes to infinity, the fraction of weights that are of significant size goes to zero, even while the actual number of such weights remains non-zero. There is thus a vanishingly small chance that a single hidden unit will have a significant effect on two different outputs, which is what would be needed to make the two outputs dependent.

However, with non-Gaussian priors, we can introduce dependence between outputs without also introducing correlation. One way to do this is use t-distributions that are expressed as mixtures of Gaussian distributions of varying scale. With each hidden unit, j, we associate an output weight variance hyperparameter, $\sigma_{v,j}^2$. As a prior, we give $1/\sigma_{v,j}^2$ a Gamma distribution with shape parameter $\alpha/2$ and mean σ_v. Given a value for this common hyperparameter, the weights out of a hidden unit, v_{jk}, have independent Gaussian distributions of variance $\sigma_{v,j}^2$. By integrating over the hyperparameter, one can see that each hidden-to-output weight has a t-distribution with index α, as was the case above. Now, however, the weights out of a single hidden unit are dependent — they are all likely to have similar magnitudes, since they depend on the common value of σ_v.

This setup therefore leads to single hidden units that compute common features affecting the value of many outputs, without fixing whether these effects are in the same or different directions.

Figure 6: Functions drawn from Cauchy priors for networks with step-function hidden units. Functions shown on the left are from a network with 150 hidden units, those on the right from a network with 10 000 hidden units. In both cases, $\sigma_a = \sigma_u = \sigma_b = \omega_v = 1$.

3.2 Properties of non-Gaussian stable priors

In contrast to the situation for Gaussian process priors, whose properties are captured by their covariance functions, I know of no simple way to characterize the distributions over functions produced by the priors based on non-Gaussian stable distributions. I will therefore confine myself in this section to illustrating the nature of these priors by displaying functions sampled from them.

As before, we can begin by considering a network with a single real input and a single real output, with step-function hidden units. Figure 6 shows two functions drawn from priors for such networks in which the weights and biases into the hidden units have independent Gaussian distributions and the weights and bias for the output have Cauchy distributions (the stable distribution with $\alpha = 1$). Networks with 150 hidden units and with 10 000 hidden units are shown, for which the width parameter of the Cauchy distribution was scaled as $\sigma_v = \omega_v H^{-1}$. As is the case for the Gaussian priors illustrated in Figure 1, the general nature of the functions is the same for the small networks and the large networks, with the latter simply having more fine detail. The functions are clearly very different from those drawn from the Gaussian prior that are shown in Figure 1. The functions from the Cauchy prior have large jumps due to single hidden units that have output weights of significant size.

When the prior on hidden-to-output weights has a form that converges to a stable distribution with $0 < \alpha < 1$, the dominance of small numbers of hidden units becomes even more pronounced than for the Cauchy prior. For stable priors with $1 < \alpha < 2$, effects intermediate between the Cauchy and the Gaussian priors are obtained. These priors may of course be used in conjunction with tanh hidden units. Figure 7 illustrates some of these possibilities for functions of two inputs.

An infinite network whose prior is based on a stable distribution with a small α can be used to express whatever valid intuitions we may sometimes have that might otherwise lead us to use a network with a small number of hidden units. With a small α, the contributions of a small subset of the hidden units will dominate, which will be good if we in fact have reason to believe that the true function is close to one that can be represented by a small network. The remaining hidden units will still be present, however, and able to make any small corrections that are needed to represent the function exactly.

4 Priors for networks with more than one hidden layer

I will conclude with a preliminary look at priors for multilayer perceptron networks with more than one layer of hidden units, starting with networks in which the outputs are connected only to the last hidden layer, each hidden layer after the first has incoming connections only from the preceding hidden layer, and the first hidden layer has incoming connections only from the inputs.

Consider such a network with several layers of step-function hidden units, with all the weights and biases having Gaussian prior distributions. Assume that the standard deviation of the weights on the connections out of a hidden layer with H units is scaled down by $H^{-1/2}$, as before. We are again interested in the limiting distribution over functions as the number of hidden units in each layer goes to infinity.

Figure 8 shows functions of one input drawn from this prior for networks with one, two, and three hidden layers. The function value is shown by a dot at each of 500 grid points in the central region of the input space. (This presentation shows the differences better than a line plot does.) With one hidden layer, the function is Brownian, as was already seen in Figure 1. With two hidden layers, the covariance between nearby points falls off much more rapidly with their separation, and with three hidden layers, this appears to be even more pronounced.

This is confirmed by numerical investigation, which shows that the networks with two and three hidden layers satisfy equation (12) with $\eta \approx 1/2$ and $\eta \approx 1/4$, respectively. For networks where only the first hidden layer is connected to the inputs, it should be true in general that adding an additional hidden layer with step-function units after what was previously the last hidden layer results in a reduction of η by a factor of two. To see this, note first that the total input to one of the hidden units in this new layer will have the same distribution as the output

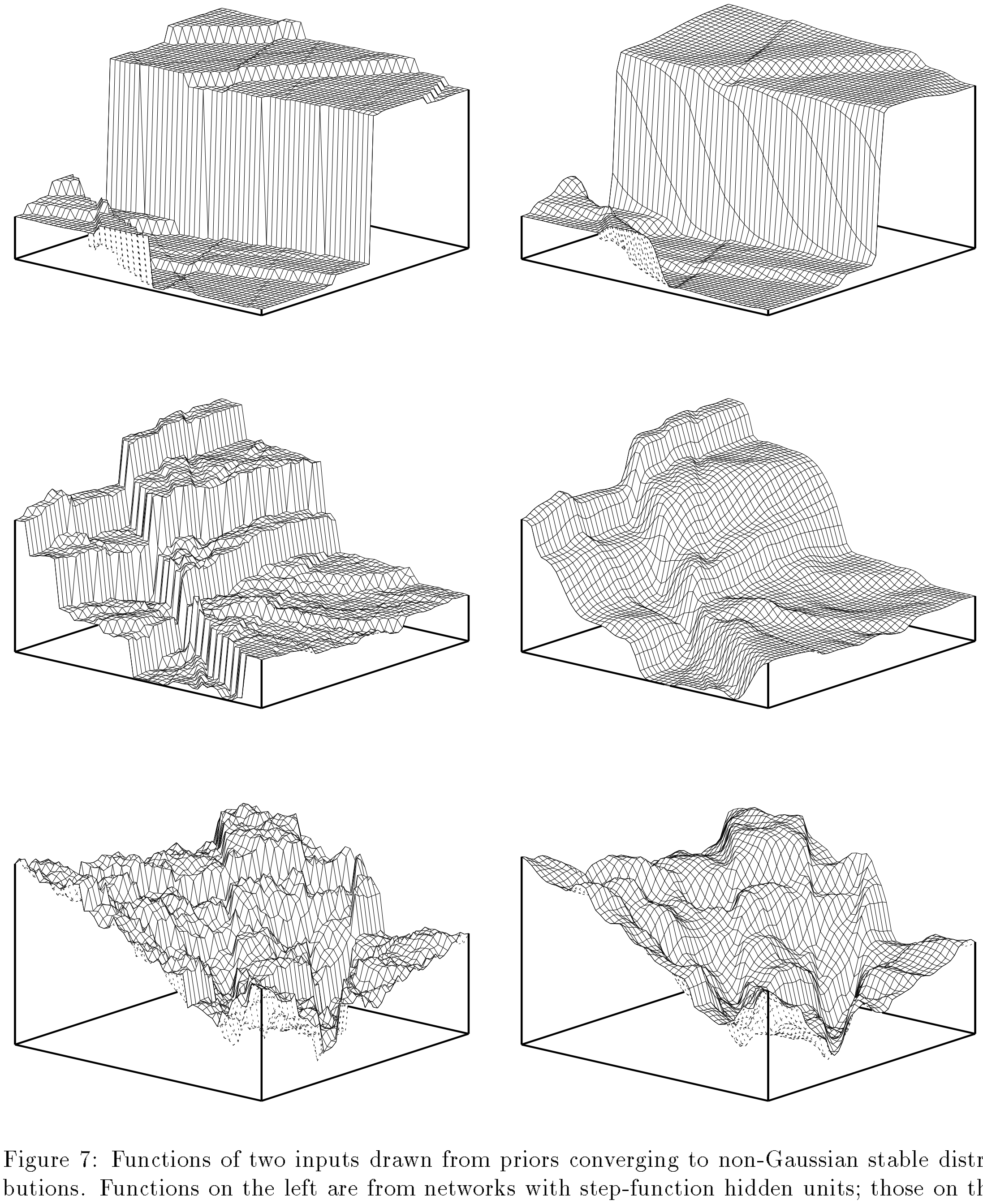

Figure 7: Functions of two inputs drawn from priors converging to non-Gaussian stable distributions. Functions on the left are from networks with step-function hidden units; those on the right are the corresponding functions from networks with tanh hidden units, with $\sigma_u = 20$. For the functions at the top, the prior on hidden-to-output weights was a t-distribution with $\alpha = 0.5$; in the middle, the prior was Cauchy (a t-distribution with $\alpha = 1$); on the bottom the prior was a t-distribution with $\alpha = 1.5$. All the networks had 1000 hidden units. In all cases, priors with $\sigma_a/\sigma_u = 1$ were used; the plots extend from -1 to $+1$ for both inputs, within the corresponding central region.

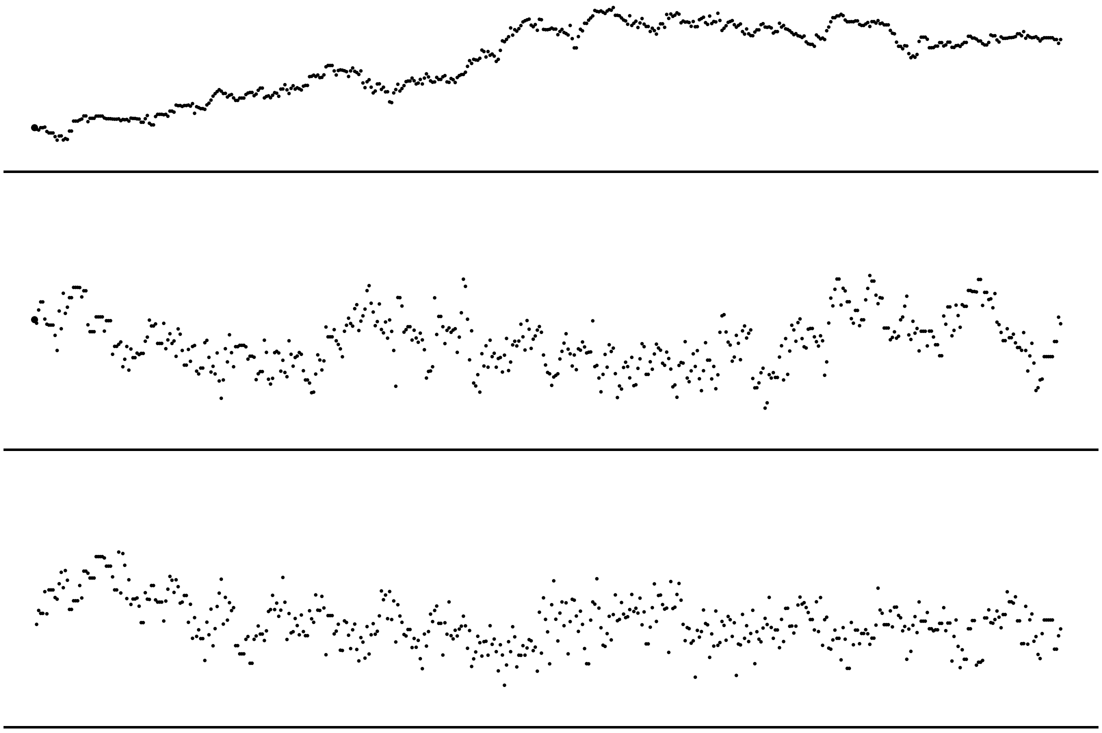

Figure 8: Functions computed by networks with one (top), two (middle), and three (bottom) layers of step-function hidden units, with Gaussian priors. All networks had 2000 units in each hidden layer. The value of each function is shown at 500 grid points along the horizontal axis.

of the old network. For a unit in the new hidden layer, $(h(x^{(p)}) - h(x^{(q)}))^2$ will be 0 or 4 depending on whether the unit's total input changes sign between $x^{(p)}$ and $x^{(q)}$. The probability of this occurring will be directly proportional to the difference in value between the total input to the unit at $x^{(p)}$ and the total input at $x^{(q)}$. By hypothesis, this difference is Gaussian with a variance proportional to $|x^{(p)} - x^{(q)}|^\eta$, giving an expected absolute magnitude for the difference that is proportional to $|x^{(p)} - x^{(q)}|^{\eta/2}$. From this it follows that $D(x^{(p)}, x^{(q)}) = E[(h(x^{(p)}) - h(x^{(q)}))^2]$ is also proportional to $|x^{(p)} - x^{(q)}|^{\eta/2}$.

Though it is interesting that fractional Brownian priors with $\eta < 1$ can be obtained in this manner, I suspect that such priors will have few applications. For small values of η, the covariances between the function values at different points drop off rapidly with distance, making prediction for test points that are different from training points very difficult. Hence, although problems for which such a prior would be appropriate may well exist, we will usually not even try to solve them, due to the poor prospects for success. (It might, however, be useful to include such a prior as one of several possibilities to be selected from at a higher level, thereby allowing the data itself to indicate whether the function is of this unlearnable type.)

More interesting effects can be obtained using a combination of Gaussian and non-Gaussian priors, in a network with two hidden layers of the following structure. The first hidden layer contains H_1 tanh or step-function units, with priors for the biases and the weights on the input connections that are Gaussian, or of the fractional Brownian type described in Section 2.4. The second hidden layer contains H_2 tanh or step-function units, with Gaussian priors for the biases and for the weights on the connections from the first hidden layer (with the standard deviation for these weights scaled as $H_1^{-1/2}$). There are no direct connections from the inputs to the second hidden layer. Finally, the outputs are connected only to the last hidden layer, with a prior for the hidden-to-output weights that converges to a non-Gaussian stable distribution of index α (for which the width of the prior will scale as $H_2^{-1/\alpha}$).

With this setup, the function giving the total input into a unit in the second hidden layer has the same prior distribution as the output function for a network of one hidden layer with Gaussian priors, which may, for example, have the forms seen in Figures 1, 2, 3, or 5. The step-function or tanh hidden units will convert such a function into one bounded between -1 and $+1$. These hidden units may be seen as "feature detectors" that indicate whether the input lies in one of the regions where their total input is significantly greater than zero. The use of non-Gaussian priors for the weights from these hidden units to the outputs allows these feature detectors to have a significant affect on the output.

Functions drawn from such a prior are illustrated in Figure 9. Such functions have low probability under the priors for networks with one hidden layer that have been discussed, suggesting that two-layer networks will be advantageous in some applications.

Finally, we can consider the limiting behaviour of the prior over functions as the number of hidden layers increases. If the priors on hidden-to-hidden weights, hidden unit biases, and input-to-hidden weights (if present) are the same for all hidden layers, the prior over the functions computed by the units in the hidden layers of such a network will have the form of a homogeneous Markov chain — that is, under the prior, the distribution of functions computed by hidden units in layer $\ell + 1$ is influenced by the functions computed by earlier layers only through the functions computed by layer ℓ, and furthermore, the conditional distribution of functions computed by layer $\ell + 1$ given those computed by layer ℓ is the same for all ℓ. We can now ask whether this Markov chain converges to some invariant distribution as the number of layers goes to infinity, given the starting point established by the prior on weights into the first hidden layer. If the chain does

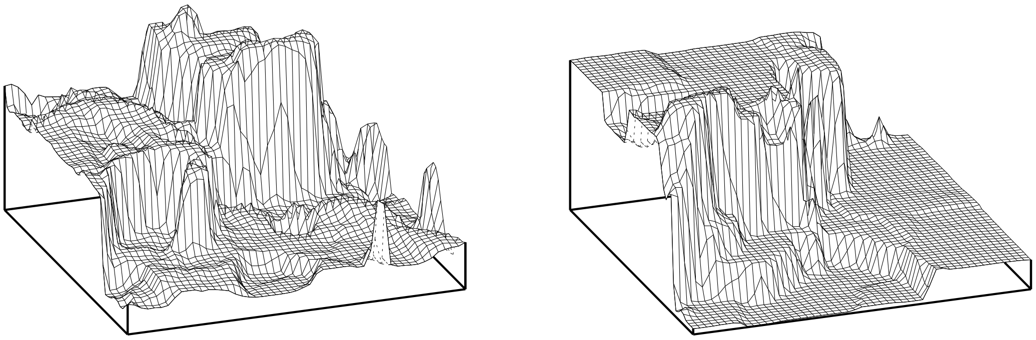

Figure 9: Two functions drawn from a combined Gaussian and non-Gaussian prior for a network with two layers of tanh hidden units. The first hidden layer contained $H_1 = 500$ units; the second contained $H_2 = 300$ units. The priors for weights and biases into the first hidden layer were Gaussian with standard deviation 10. The priors for weights and biases into the second hidden layer were also Gaussian, with the biases having standard deviation 20 and the weights from the first hidden layer having standard deviation $20H_1^{-1/2}$. The weights from the second hidden layer to the output were drawn from a t-distribution with $\alpha = 0.6$ and a width parameter of $H_2^{-1/0.6}$, which converges to the corresponding stable distribution. The central regions of the functions are shown, where the inputs vary from -1 to $+1$.

converge, then the prior over functions computed by the output units should also converge, since the outputs are computed solely from the hidden units in the last layer.

This question of convergence appears difficult to answer. Indeed, when each hidden layer contains an infinite number of hidden units, it is not even obvious how convergence should be defined. Nevertheless, from the discussion above, it is clear that a Gaussian-based prior for a network with many layers of step-function hidden units, with no direct connections from inputs to hidden layers after the first, either does not converge as the number of layers goes to infinity, or if it can be regarded as converging, it is to an uninteresting distribution concentrated on completely unlearnable functions. However, if direct connections from the inputs to all the hidden layers are included, it appears that convergence to a sensible distribution may occur, and of course there are also many possibilities involving non-Gaussian stable priors and hidden units that compute a smooth function such as tanh rather than a step function.

Finding a prior with sensible properties for a network with an infinite number of hidden layers, each with an infinite number of units, would perhaps be the ultimate demonstration that Bayesian inference does not require limiting the complexity of the model. Whether such a result would be of any practical significance would of course depend on whether such networks have any significant advantage over networks with one or two layers, and on whether a prior close to the limit is obtained with a manageable number of layers (say less than ten) and a manageable number of hidden units per layer (perhaps in the hundreds).

Acknowledgements

I thank David MacKay, Chris Williams, and Carl Rasmussen for helpful discussions and comments on the manuscript. This work was supported by the Natural Sciences and Engineering Research Council of Canada, and by the Information Technology Research Centre.

References

Peitgen, H.-O. and Saupe, D., editors (1988) *The Science of Fractal Images*, New York: Springer-Verlag.

Buntine, W. L. and Weigend, A. S. (1991) "Bayesian back-propagation", *Complex Systems*, vol. 5, pp. 603-643.

Falconer, K. (1990) *Fractal Geometry: Mathematical Foundations and Applications*, Chichester: John Wiley.

Feller, W. (1966) *An Introduction to Probability Theory and its Applications, Volume II*, New York: John Wiley.

Funahashi, K. (1989) "On the approximate realization of continuous mappings by neural networks", *Neural Networks*, vol. 2, pp. 183-192.

MacKay, D. J. C. (1991) *Bayesian Methods for Adaptive Models*, Ph.D thesis, California Institute of Technology.

MacKay, D. J. C. (1992) "A practical Bayesian framework for backpropagation networks", *Neural Computation*, vol. 4, pp. 448-472.

Neal, R. M. (1992) "Bayesian training of backpropagation networks by the hybrid Monte Carlo method", Technical Report CRG-TR-92-1, Dept. of Computer Science, University of Toronto.

Neal, R. M. (1993) "Bayesian learning via stochastic dynamics", in C. L. Giles, S. J. Hanson, and J. D. Cowan (editors), *Advances in Neural Information Processing Systems 5*, pp. 475-482, San Mateo, California: Morgan Kaufmann.

Ripley, B. D. (1981) *Spatial Statistics*, New York: John Wiley.

Szeliski, R. (1989) *Bayesian Modeling of Uncertainty in Low-level Vision*, Boston: Kluwer.