# Infinite Models

Many real phenomena are of essentially unlimited complexity:

> Suppose we model the growth rate of trees as a function of climate, soil type, genetic characteristics, disease suppression measures taken, etc. There is no reason to think any simple functional form (eg, linear, low-order polynomial) will capture the many ways these factors interact to determine tree growth.

How can we build a model that accommodates such complexity? One approach:

- Define models that can have any finite amount of complexity (eg, any finite number of hidden units in an MLP network).
- Define priors for these models that make sense.
- See if the limit as the complexity goes to infinity is sensible.

If the limit makes sense, we can use a model that is as large as we can handle computationally. (And sometimes, we can figure out how to actually implement the infinite model on a finite computer!)



# The Meaning of a MLP Network Model & Prior

A MLP network with only a few hidden units may not seem like a reasonable model — seldom is the real function we need a sum of tanh functions.

But a network with many hidden units is a reasonable "non-parametric" model, since it can approximate any function arbitrarily closely.

So we should use a network with many hidden units. But how does a prior over the network parameters translate into a prior for the function, $f(x)$, computed by the network, and hence for the conditional distribution for $y$ given $x$?

Suppose we give independent Gaussian priors to all the network parameters:

$$\begin{aligned} \beta_0 &\sim \text{Gaussian}(0, \sigma_{\beta 0}^2) \\ \beta_k &\sim \text{Gaussian}(0, \sigma_{\beta}^2) \\ \gamma_{k,0} &\sim \text{Gaussian}(0, \sigma_{\gamma 0}^2) \\ \gamma_{k,j} &\sim \text{Gaussian}(0, \sigma_{\gamma}^2) \end{aligned}$$

This produces a prior distribution for the output of the network at any selection of input points — i.e. for $f(x^{(1)})$, $f(x^{(2)})$, ... It is this distribution that really matters.



# The Limit of the Prior Over Functions

Consider the prior over functions as the number of hidden units, $q$, goes to infinity.

Look at the network output function evaluated at a single point, say $x^{(1)}$:

$$f(x^{(1)}) \;=\; \beta_0 + \sum_{k=1}^{q} \beta_k h_k(x^{(1)})$$

The first term above is Gaussian.

By the Central Limit Theorem, the second term will become Gaussian as $q \to \infty$, as long as each term has finite variance. If we use the tanh activation function, $h_k(x^{(1)})$ will be bounded, so this must be the case.

Hence $f(x^{(1)})$ becomes Gaussian for large $q$. Its distribution will reach a limit if we make $\sigma_\beta$ scale as $q^{-1/2}$.

Similarly, the joint distribution of the function at any number of input points converges to a multivariate Gaussian. A distribution over functions that has this property is called a *Gaussian process*.



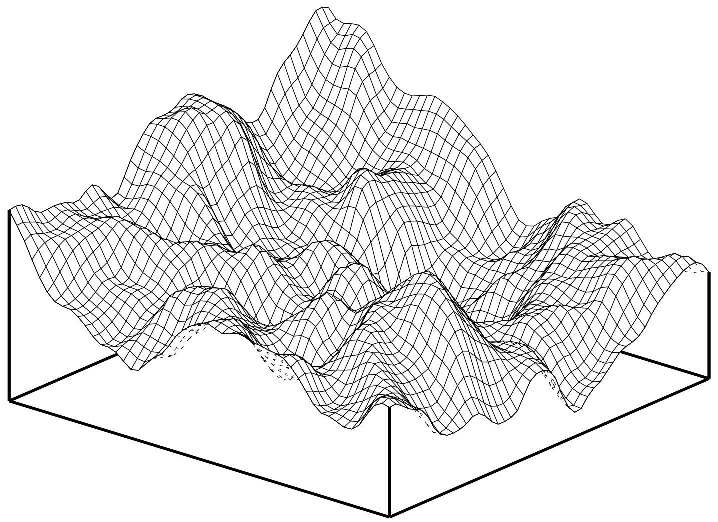
# Some Properties of the Gaussian Network Prior

- The $\beta_k$ parameters go to zero as the number of hidden units goes to infinity. So hidden units are individually of no importance.
- With a smooth hidden unit activation function, the functions are smooth. In a central region around $x = 0$,

$$\text{Corr}\left[f(x^{(1)}),\, f(x^{(2)})\right] \;\approx\; 1 \,-\, c\,||x^{(1)} - x^{(2)}||^2, \qquad \text{for some } c > 0$$

- Even with more hidden layers, we still get a Gaussian process prior (perhaps with a different covariance function).

## A Function From the Gaussian Network Prior



The network had two inputs, one output, and 10000 tanh hidden units.



## Priors That Don't Produce Gaussian Processes

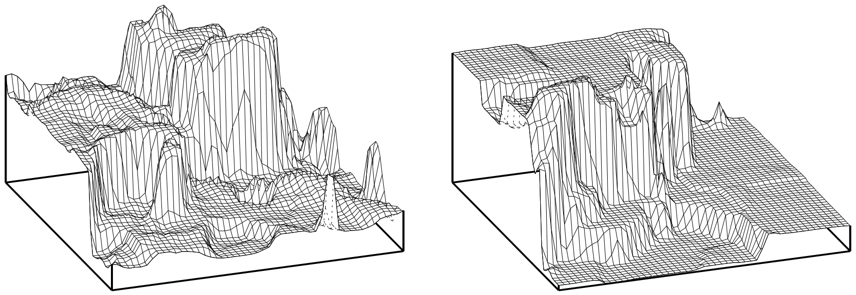
If we give the $\beta_k$ parameters a prior with an *infinite* variance, such as a $t$ distribution, and scale the width of the prior suitably, we can get a well-defined $q \to \infty$ limit that isn't a Gaussian process.

In this case:

- The $\beta_k$ parameters don't all go to zero — some individual hidden units do have a big effect.
- Networks with more than one hidden layer can produce interesting new effects.



## Functions From a Network With Two Hidden Layers



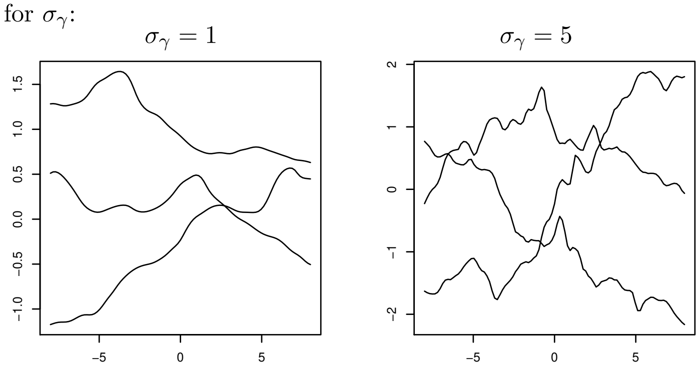
The networks had two inputs. Gaussian priors were used for parameters on connections into the first hidden layer (with 500 tanh units) and second hidden layer (300 tanh units), but the prior for parameters on connections from the second hidden layer to the output had a $t$ distribution with 0.6 df.

The two functions differ only in the random number seed used when generating parameters from their priors.



## Hyperparameters for Neural Network Models

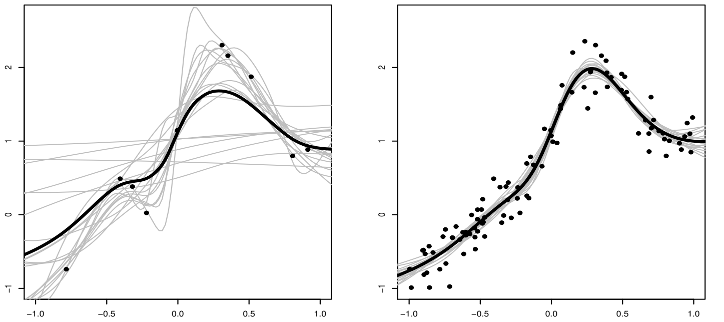
The priors for network parameters (the $\beta$s and $\gamma$s) affect the nature of functions drawn from the network prior. Here are samples of three functions (of one input) drawn from Gaussian priors for networks of 1000 hidden units, using two different values for $\sigma_\gamma$:




A larger $\sigma_\gamma$ produces "wigglier" functions. Usually, we won't know exactly how wiggly the function should be. So we make $\sigma_\gamma$ a variable *hyperparameter*, and give it a prior distribution that spans a few orders of magnitude. We usually make $\sigma_{\gamma 0}$ and $\sigma_\beta$ hyperparameters as well ($\sigma_{\beta 0}$ might be fixed).

## Complexity of Functions Versus Complexity of Predictions

Here's an illustration of how a complex model — a neural network with 100 hidden units — can produce simple predictions when there is little data, despite being capable of representing more complex functions that fit the data better. The data is from the function $x^3 + 2\exp(-6(x-0.3)^2)$, with Gaussian noise with standard deviation 0.2 added. The plots show show 20 functions from the posterior distributions given 10 data points (left) and 100 data points (right):




The bold line is the average of these functions, the Bayesian prediction for minimizing squared error, which is smoother than some individual functions.



## Hierarchical Hyperpriors: Automatic Relevance Determination

More elaborate priors for hyperparameters can be used to allow for various possible high-level characteristics of the data. One very useful example is the *Automatic Relevance Determination* prior.

When we have many inputs, we are often uncertain how relevant each is to predicting the target variable. We can express this uncertainty by using a prior for the $\gamma$ parameters such as the following:

$$\begin{aligned} \gamma_{k,j} \mid \sigma_{\gamma,j} &\sim \text{Gaussian}(0, \sigma^2_{\gamma,j}), \quad \text{for } j = 1, \ldots, p \text{ and } k = 1, \ldots, q \\ \log(\sigma_{k,j}) &\sim \text{Gaussian}(\ldots) \end{aligned}$$

Here, $\sigma_{\gamma,j}$ controls the magnitude of coefficients on connections from input $j$ to the hidden units. If it is small, this input will largely be ignored.