[Paper review 2]

Bayesian Learning For Neural Networks (Radford M. Neal, 1995)

[Contents]

2. Priors for Infinite Networks

Infinite network limit provides insight into properties of different priors

Priors

- our belief, information in advance
- (but meaning of weight & bias in Neural Network is obscure... hard to design prior)

Infinite network = "NON-parametric" model

- Network with 1 hidden layers, with INFINTE number of unit can approximate any continuous function
- (in practice) will use only finite number

  (but do not restrict the size based on the "SIZE of TRAINING dataset")

Notation

- I : number of real-valued input ($= x_i$)
- O : number of real-valued output ($= f_k(x)$)
- H : number of hidden units (in one layer) (= $h_j(x)$)

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

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

Relation of the target ditsribution & network output :

- (regression) mean
- (classification) softmax

2-1. Priors converging to Gaussian Processes

Notation :

- hidden-output weight : $v_{jk} \sim N(0, \sigma_v^2)$
- output bias : $b_k \sim N(0, \sigma_b^2)$

As number of hidden units (=H) increases $\rightarrow$ prior over function converges to GAUSSIAN PROCESS!

2-1.1 Limits for Gaussian Process

set 1 input unit(= 1 dimension input) : $x^{(1)}$

we look at the prior distribution of $f_k(x^{(1)})$

- this is implied by the prior distribution for "Weight & Bias"

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

- SUM of bias & weighted contribution of H hidden units

Mean & Variance of $f_k(x^{(1)}) \sim N(0, \sigma_b^2 + H\sigma_v^2 V(x^{(1)}))$

- $E[f_k(x^{(1)})] = E[b_k + \sum_{j=1}^{H} v_{jk} h_j(x^{(1)})] = E[b_k] + E[\sum_{j=1}^{H} v_{jk} h_j(x^{(1)})] = 0 + H \times 0 = 0$
- $V[f_k(x^{(1)})] = \sigma_b^2 + H\sigma_v^2 V(x^{(1)})$

If we set $\sigma_v = \omega_v H^{-1/2}$ for some fixed ω_v,

then $f_k(x^{(1)}) \sim N(0, \sigma_b^2 + \omega_v^2 V(x^{(1)}))$ as $H \to \infty$

Prior Joint distribution

$$E\left[f_k\left(x^{(p)}\right) f_k\left(x^{(q)}\right)\right] = \sigma_b^2 + \sum_j \sigma_v^2 E\left[h_j\left(x^{(p)}\right) h_j\left(x^{(q)}\right)\right]$$
$$= \sigma_b^2 + \omega_v^2 C\left(x^{(p)}, x^{(q)}\right)$$

That is,

$f^k(X) = f^k(x^{(1)}, \ldots, x^{(n)}) = N(0, \sigma_b^2 + \omega_v^2 C\left(x^{(p)}, x^{(q)}\right))$,

where $C\left(x^{(p)}, x^{(q)}\right) = E[h_j(x^{(p)}) h_j(x^{(q)})] - 0^2$ (stands for covariance)

Distribution of this sort (in which joint dist'n of values of function at ANY FINITE NUMBER of points is Gaussian) = 'Gaussian Process'

Given values for $f_k(x^{(1)}, \ldots, x^{(n-1)}) \rightarrow$ we can find the PREDICTIVE distribution for $f_k(x^{(n)})$

- by conditioning on these known values!
- but it requires evaluation of $C\left(x^{(p)}, x^{(q)}\right)$
  - $x^{(p)}$: training set
  - $x^{(q)}$: test set (new data)
  - have to be done by numerical integration

covariance between $f_{k_1}(x^{(q)})$ & $f_{k_2}(x^{(p)})$ is zero, whenever $k_1 \neq k_2$

- different output unit $\rightarrow$ no correlation (independent)
- That is, knowing the values of one output, does not tell about the other outputs

  (train 1 network with 2 output units = train 2 network with 1 output units)
- It seems strange, since those two outputs have shared same hidden units!

Dependencies between outputs can be introduced by making the weights to various outputs from one hidden unit be DEPENDENT!

(but if these weights have Gaussian priors $\rightarrow$ can be dependent only if they are correlated)

2-1-2. Priors that lead to smooth & Brownian functions

Setting :

- input-hidden weight : $u_{1j} \sim N(0, \sigma_u^2)$
- hidden bias : $a_j \sim N(0, \sigma_a^2)$

hidden unit = "step-function" $\rightarrow$ (1) Brownian Priors

hidden unit = "tanh" $\rightarrow$ (2) Priors over smooth functions

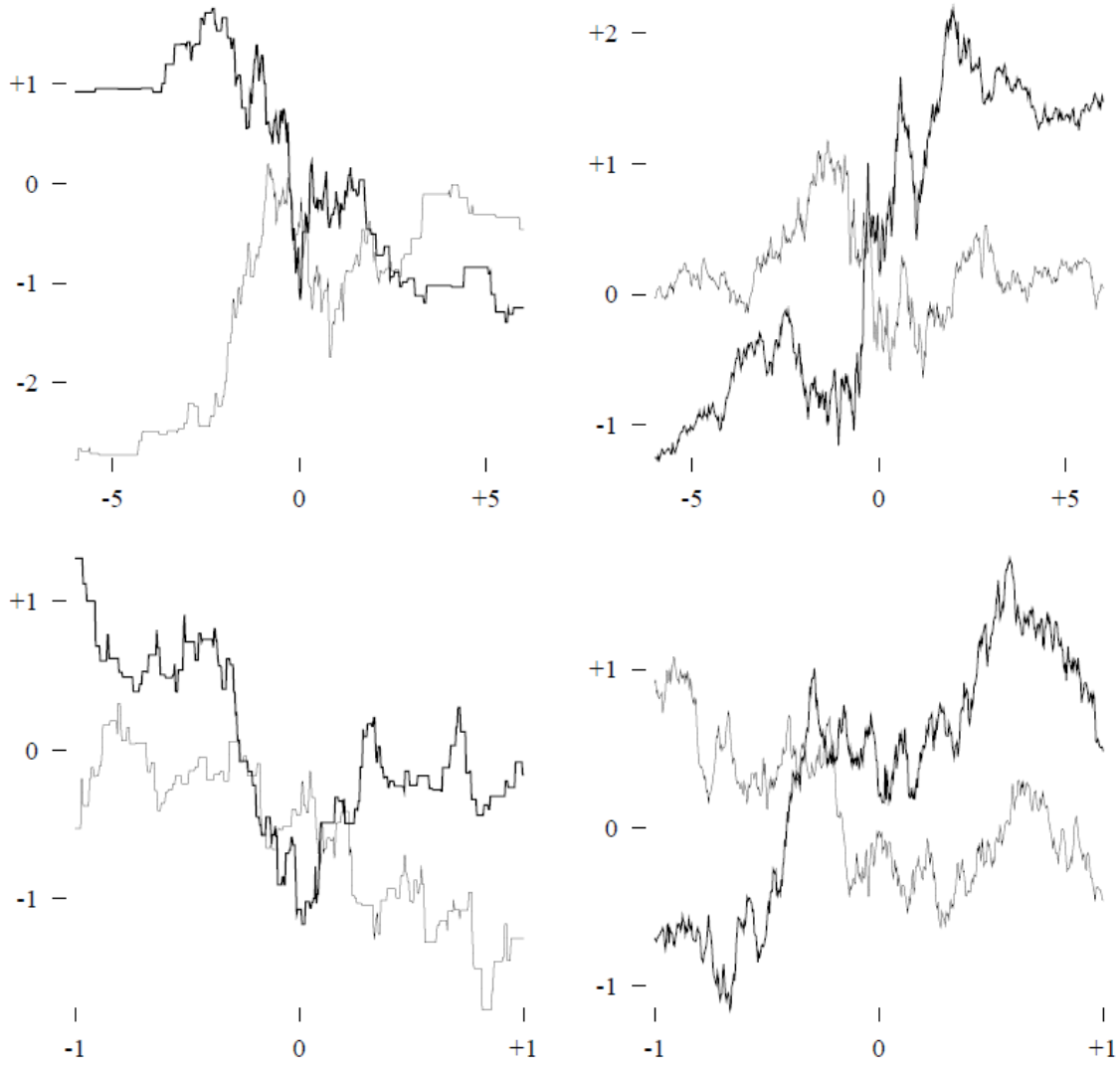
(1) Brownian Priors

- hidden unit = STEP FUNCTION (changing from -1 to +1 at zero)

  (determine the point in the input space where that hidden unit's step occurs)
- Prior dist'n of this step point : "Cauchy (width= σ_a / σ_u)"

  (= variation in this function is concentrated $x = 0$, with width of roughly σ_a / σ_u)

  (= far from $x = 0$, functions become almost constant, since few hidden units have their steps far out)



(L) 300 hidden units

(R) 10000 hidden units

This illustrate that the prior over function is reaching a limiting distribution as $H \to \infty$

(2) Priors over smooth functions

- hidden unit = 'TANH'

2-1-3. Covariance Functions of Gaussian Priors

Gaussian process

- 1) mean value (of the function at each point)
- 2) covariance (of the function value at any two points)

difference in 2-1-2 (different hidden unit $\to$ different prior)

- it is reflected in the local behavior of their "covariance function"

  (= covariance of the values of a hidden unit)

$$C\left(x^{(p)}, x^{(q)}\right) = \frac{1}{2}\left(V\left(x^{(p)}\right) + V\left(x^{(q)}\right) - E\left[\left(h\left(x^{(p)}\right) - h\left(x^{(q)}\right)\right)^2\right]\right)$$
$$= V - \frac{1}{2}D\left(x^{(p)}, x^{(q)}\right)$$

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

  ($D\left(x^{(p)}, x^{(q)}\right) = E\left[\left(h\left(x^{(p)}\right) - h\left(x^{(q)}\right)\right)^2\right]$)

(1) STEP-FUNCTION hidden unit

- Brownian motion
- $\left(h\left(x^{(p)}\right) - h\left(x^{(q)}\right)\right)^2$ will be either 0 or 4

  (depending on whether the values of the hidden unit's bias & incoming weight result in the step being located between $x^{(p)}$ and $x^{(q)}$)
- $D\left(x^{(p)}, x^{(q)}\right) \sim \left|x^{(p)} - x^{(q)}\right|$

  ($\sim$ indicates proportionality for neqrby points)

(2) TANH hidden unit

- Smooth function
- $D\left(x^{(p)}, x^{(q)}\right) \sim \left|x^{(p)} - x^{(q)}\right|^2$

$$E_a\left[(h(x-s/2)-h(x+s/2))^2\right] \approx \begin{cases} c_1\left(|u|/\sigma_a\right)\cdot s & \text{if } |u| \geq 1/s \\ c_2\left(|u|^2/\sigma_a\right)s^2 & \text{if } |u| \lesssim 1/s \end{cases}$$

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

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

2-1-4. Fractional Brownian Priors

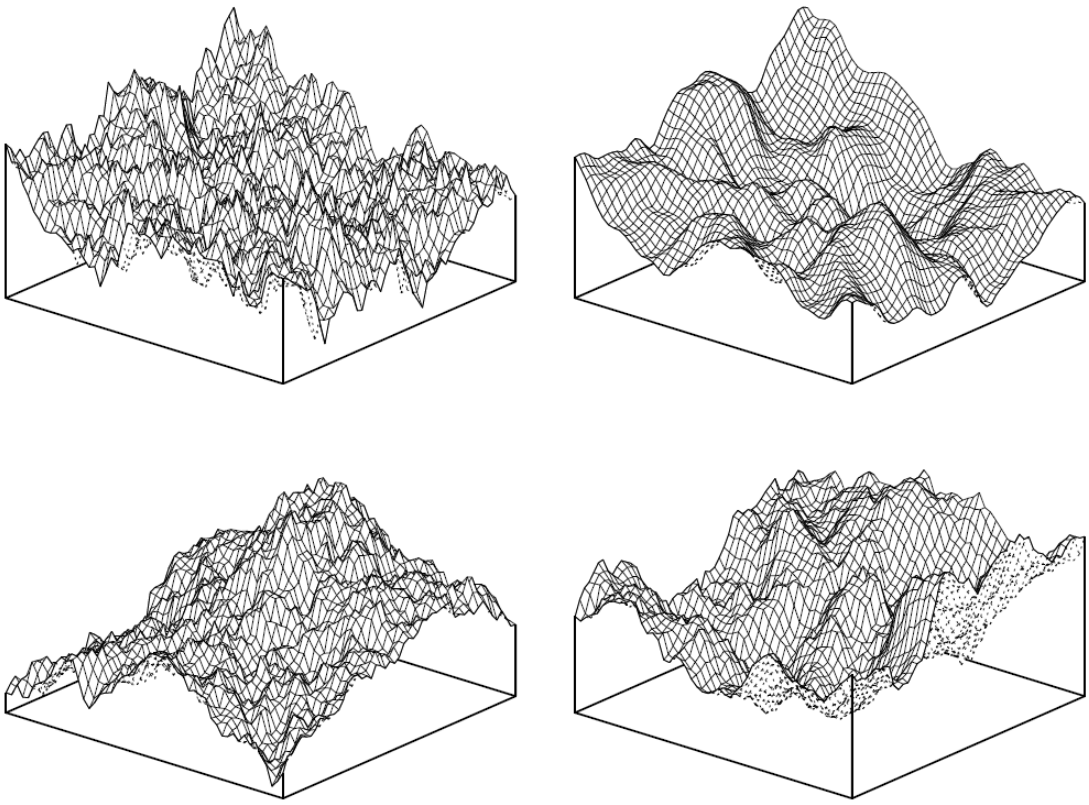
$D\left(x^{(p)}, x^{(q)}\right) \sim \left|x^{(p)} - x^{(q)}\right|^{\eta}$

- if $\eta = 2$: Smooth Function
- if $\eta = 1$: Brownian Functions

2-1-5. Networks with more than one input

$h_j(x) = \tanh\left(u_{1j}x_1 + a_j + \sum_{i=2}^{I} u_{ij}x_i\right)$

- when $x_2, \ldots x_I$ are fixed $\rightarrow$ they cat simply to increase the "variance of the effective bias" (just spreads the variation in the function over a larger range of values for x_1)



Functions of two inputs drawn from Gaussian priors

- [U-L] : 10000 step function
- [U-R] : tanh
- [L-L] : $\eta = 1.3$

- [L-R] : $\eta = 1.7$

2.2 Priors converging to non-Gaussian stable distribution

have seen various interesting priors (above) $\rightarrow$ still disappointing

Reason?

- 1) can do the same thing without need for a network!

  (need to look at different priors, if BNN are to significantly extend the range of models available)
- 2) with Gaussian priors, the contributions of individual hidden units are all negligible

  (each unit does not capture important aspect of data)

2-2-1. Limits for priors, with infinite variance

Stable distributions (Wikipedia)

"In probability theory, a distribution is said to be **stable** if a linear combination of two independent random variables with this distribution has the same distribution, up to location and scale parameters. " (Wikipedia)

Let *X*1 and *X*2 be independent copies of a random variable *X*. Then *X* is said to be **stable** if for any constants *a* > 0 and *b* > 0 the random variable *aX*1 + *bX*2 has the same distribution as *cX* + *d* for some constants *c* > 0 and *d*. The distribution is said to be *strictly stable* if this holds with *d* = 0.[

Stable distribution

- convergence of priors, where hidden-to-output weights have infinite variance
- if Z_is are independent & have the same symmetric stable distribution of index α,

  then $(Z_1 + \ldots + Z_n)/n^{1/\alpha}$ has the same distribution as Z_i (can make n go to infinity!)
- if $\alpha = 2$: Gaussians (of varying standard deviation)

  if $\alpha = 1$: Cauchy distributions (of varying widths)
- Z_i don't have to be Gaussian!

non-Gaussian priors

- can introduce dependence between outputs! (by putting correlation)
- ex) t-distribution

  common hyperparameter $\rightarrow$ weights out of a hidden unit have independent Gaussian distribution of variance $sigma_{v,j}^2$

  (they are likely to have similar magnitudes)
- allows single hidden units to compute COMMON features that "affect many outputs"

2-2-2. Properties of non-Gaussian stable priors

- no simple way to characterize

  (in case of Gaussian Process priors $\rightarrow$ could express it with "mean & covariance" function)
- stable priors with $1 < \alpha < 2$, effects intermediate between Cauchy & Gaussian priors

  (these prior may of course be used in conjunction with tanh hidden units)

2-3. Priors for networks with more than one hidden layer

Outputs are connected ONLY to the last hidden layer

Interested in the limiting distribution over functions, as the number of hidden units (H) goes to infinity

Example with 3 layers

case 1) all the units follows Gaussian Prior

- standard deviation of the weights : scaled down by $H^{-1/2}$
- (with 1 hidden layer : Brownian)

  (with 2 hidden layer : covariance between nearby points falls off more rapidly)

  (with 3 hidden layer : More ~)

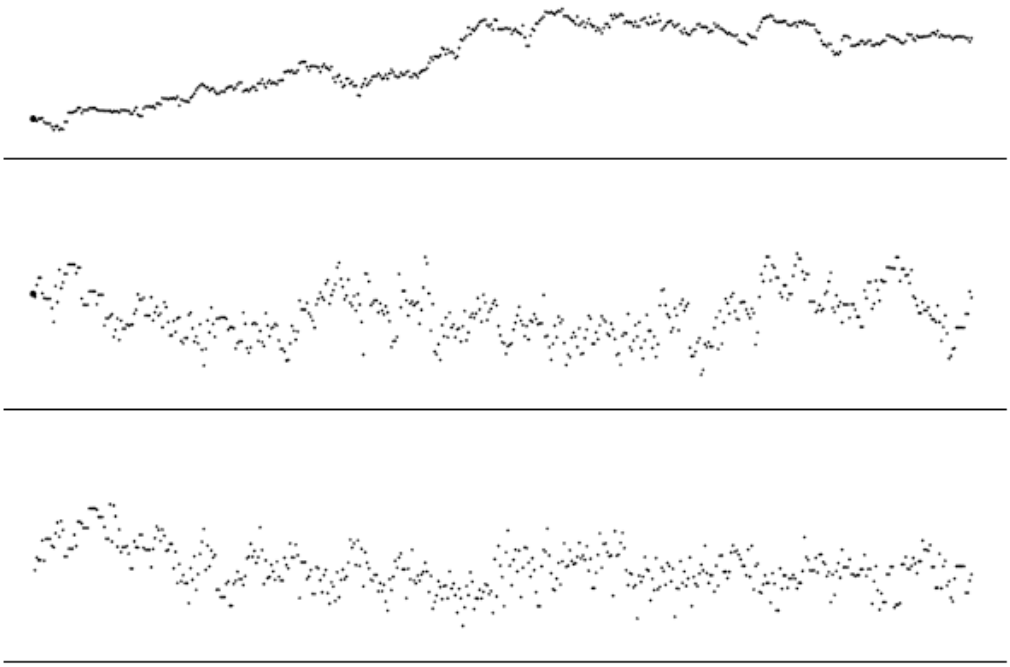


Figure 2.8: Functions computed by networks with one (top), two (middle), and three (bottom) layers of step-function hidden units, with Gaussian priors. All networks had 2000 units in each hidden layer. The value of each function is shown at 500 grid points along the horizontal axis.

- Network with 1 hidden layer : $\eta = 1$

  Network with 2 hidden layer : $\eta = 1/2$

  Network with 3 hidden layer : $\eta = 1/4$

With multiple hidden layers, fractional Brownian priors with $\eta < 1$ can be obtained!

But.... have few applications.

- for small values of η, covariance between function values at different points, drop off rapidly with distance!

  (thus, introduces unavoidable uncertainty in predictions for test points that are even slightly different from training points!)
- How to solve?

Combination of Gaussian & non-Gaussian priors (with 2 hidden layers)

- 1st hidden layer (H_1) : tanh/step function + Gaussian/fractional Brownian type prior

  2nd hidden layer (H_2): tanh/step function + Gaussian prior

  output : non-Gaussian stable distribution of index α (with width scaled as $H_2^{1/\alpha}$)
- tanh/step-function : bounded between -1~1 $\rightarrow$ "feature detector"

  non-Gaussian prior : allows individual features to have a significant effect on the output!

Combination of Gaussian & non-Gaussian have form of "homogeneous Markov Chain"

- layer $l+1$ is only affect by layer l
- does it converge to some invariant(stationary) distribution? (as $H \rightarrow \infty$)

  (if chain converges, the prior over functions (computed by the output units) should also converge! since it is only affected by the last hidden layer)
- if direct connections from the "input to ALL the hidden layers" are included, it appears that convergence to a sensible distribution may occur!
- Bayesian Inference does not require limiting the complexity of the model!

2-4. Hierarchical Models

Our prior knowledge will be to unspecific to fix values for $\sigma_b, \omega_v, \sigma_a, \sigma_u$

Wish to treat these values as UNknown hyperparameters (treat it as broad)

Benefits of hierarchical models : "Regularization" (automatically!)

- by allowing σ_u : let the data determine the scale (above which the function takes on a Brownian character)
- ex) Brownian motion : also allow hyperparameter η

  ex) t-distribution : also allow hyperparameter α

Problems

- when these hierarchical models are used with networks having "large number of hidden units"
- large $H \rightarrow$ pressure for them to all be used to explain the training data will be much less
- lead to bad predictive performance
- how to reformulate?