Modelling Uncertainty with Bayesian Neural Netwowkrs

Ellis Brown, Melanie Manko, Ethan Matlin

Columbia University

EECSE6699

September 21, 2020

Motivation

Neural Networks are really good at predicting stuff

But what about when they're wrong?

We want to know something about the certainty of a NN's prediction.

A natural framework for modelling uncertainty is the Bayesian paradigm.

Experiment 1: Why Uncertainty Matters

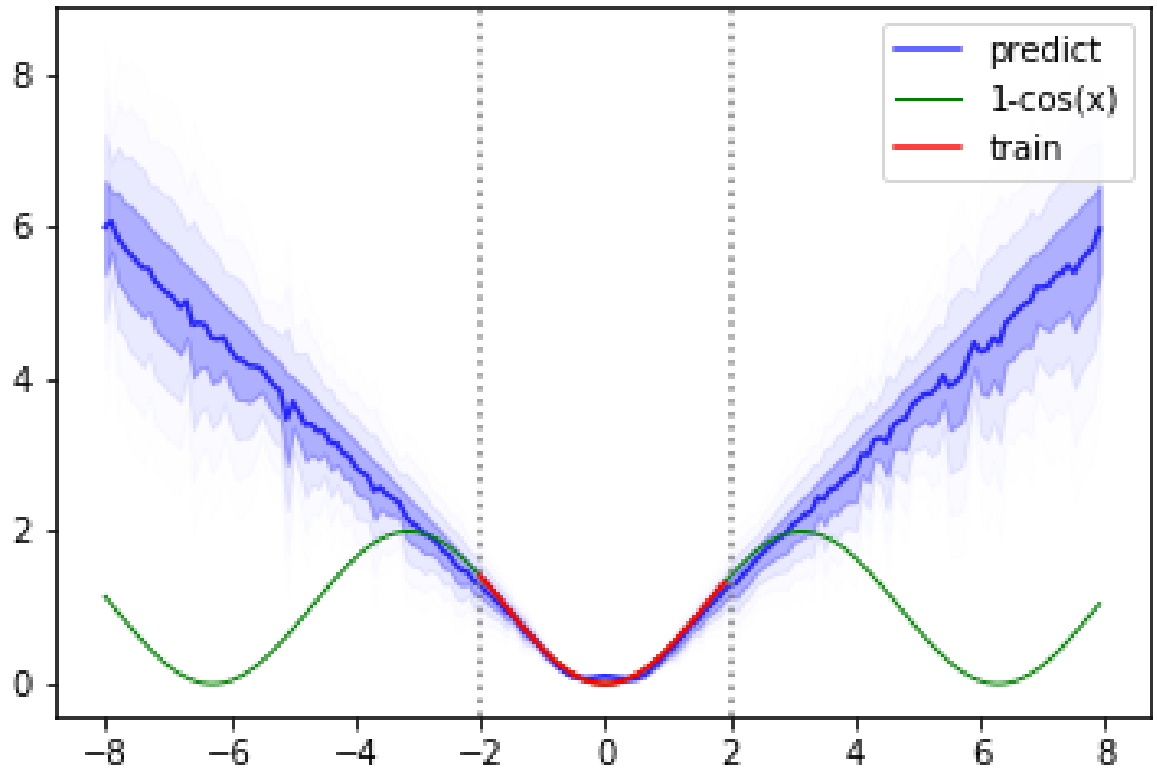

Figure: ReLU Network with 5 Hidden Layers and 1024 neurons per layer

But you just fabricated an example to t your story...

Uncertainty following the Great Recession: Gross Domestic Product

Figure: ReLU Network with 5 Hidden Layers and 1024 neurons per layer

Experiment 2: But what about Softmax? Doesn't that give uncertainty?

f (x)

(f (x))

Experiment 2: But what about Softmax? Doesn't that give uncertainty?

$\tilde{f}(x)$

$(\tilde{f}(x))$

Outline

- Neural Networks Need Better Notions of Uncertainty
- Methods of Reasoning about Uncertainty
  - In nite Bayesian NN () Gaussian Process
  - Finite Bayesian NN: Use Numerical Techniques (MCMC, Variational Inference,Dropout)
- How do priors on the weights translate to priors on functions?
- Activation Functions and the Posterior Distribution
- Shortfalls of the Approach

Bayesian Neural Networks and Gaussian Processes

In nite neural network () Gaussian Process (Neal [1995], William [1998])

- 1 hidden layer
- Bounded nonlinearites
- Weights (u_{ij} and $_{jk}$) are i.i.d with zero mean and nite variance

$$f_k(x) = \sum_{j=1}^{H} {}_{jk} \left(\sum_{i=1}^{I} u_{ij} x_i \right) \qquad (1)$$

Central Limit Theorem $\Rightarrow$ $f_k(x)$ N 0; ^2_b + H $^2V(x)$

$V(x)$ is the covariance function of the Gaussian Process corresponding to the network.

Speci c Covariance Functions (Williams [1998])

Gaussian Nonlinearity in NN() Scaled Squared Exponential in GP

$$V_G(x; x^0) = \frac{e}{u}^{d} \exp\left(\frac{x^T x}{2\ \frac{2}{m}}\right) \exp\left(\frac{(x \quad x^0)^T (x \quad x^0)}{2\ \frac{2}{s}}\right) \exp\left(\frac{x^{0T} x^0}{2\ \frac{2}{m}}\right)!$$

Sigmoid Nonlinearity in NN() ArcSin Covariance Function in GP

$$V_{erf}(x; x^0) = \frac{2}{}\sin^{1}\left(\frac{2x^T \quad x^0}{\sqrt{(1 + 2x^T \quad x)(1 + 2x^{0T} \quad x^0)}}\right)$$

Experiment 3: In practice though we have only nite Networks

(a) NN (Gaussian Nonlinearity, 1 Hidden Layer);
GP (Squared Exponential Covariance)

(b) NN (Gaussian Nonlinearity, 5 Hidden Layers);
GP (Squared Exponential Covariance)

Dropout

How to evaluate a nite NN?

- Monte Carlo techniques: HMC, RWMH, SMC, etc. Neal [1995]
- Variational Inference: Barber & Bishop [1998], Graves [2011], Blunde et al. [2015]

Dropout is equivalent to Variational Inference in Gaussian Processes(Gal [2016], Gal & Ghahramani [2015b], Gal & Ghahramani [2016a], Gal & Ghahramani [2016b], Gal & Ghahraman [2015a])

- Used as a regularization technique [Hinton et al. [2012], Srivastava et al. [2014]]
- Intuitively, makes sense as a way to get a distribution with uncertainty

NN Variational Inference Objective Function:

$$\hat{L} = \frac{N}{M} \sum_{i \in S} \log p(y_i | f^{g(\ ;\ )}) + KL(q(\ )||p(\ )) \quad (2)$$

NN Loss Function:

$$L = \log p(y|f^{g(\ ;\ _i)}) + \text{const.} + \ _1||W_1|| + \ _2||W_2|| + \ _3||b|| \quad (3)$$

KL condition:

$$\frac{@}{@} KL(Q(\ )||p(\ )) = \frac{@}{@} \ _1||W_1|| + \ _2||W_2 + \ _3||b|| \quad (4)$$

=) the two are equivalent.

Experiment 4: What prior should I choose? How do we priors correspond to function priors?

$$W \quad N \quad (0; 0{:}05) \qquad\qquad W \quad \text{Unif}\,[\ 0{:}05; 0{:}05]$$

$$W \quad N \quad 0; \sqrt{\frac{2}{N_{in} + N_{out}}} \qquad\qquad W \quad \text{Unif} \left[\sqrt{\frac{6}{N_{in}}}; \sqrt{\frac{6}{N_{in}}} \right]$$

Figure: ReLU (Untrained), 5 Hidden Layers, 1024 neurons per layer

Experiment 5: Activation Functions and Uncertainty Estimates

Figure: Linear; 5 Hidden Layers, 1024 Neurons per Layer

Experiment 5: Activation Functions and Uncertainty Estimates

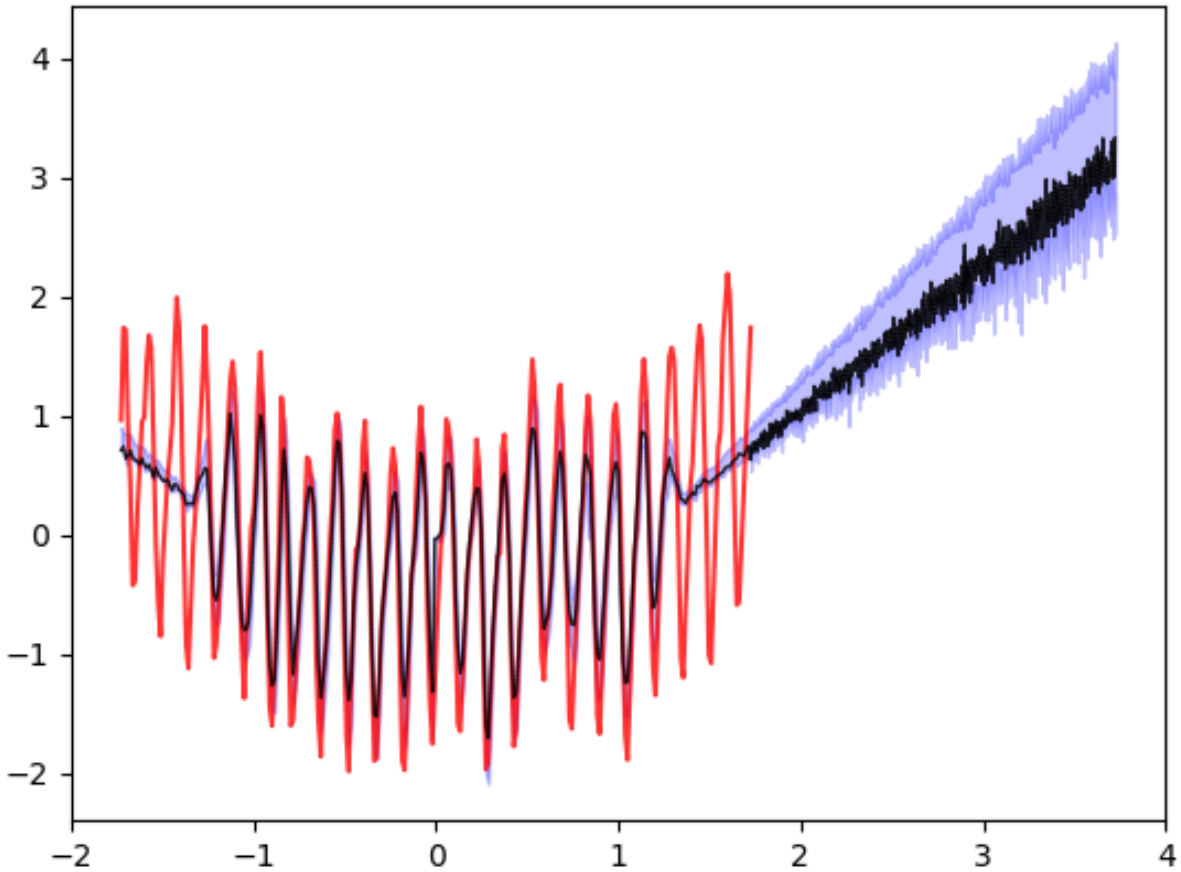

Figure: ReLu; 5 Hidden Layers; 1024 Neurons per Layer

Experiment 5: What Activation Function?

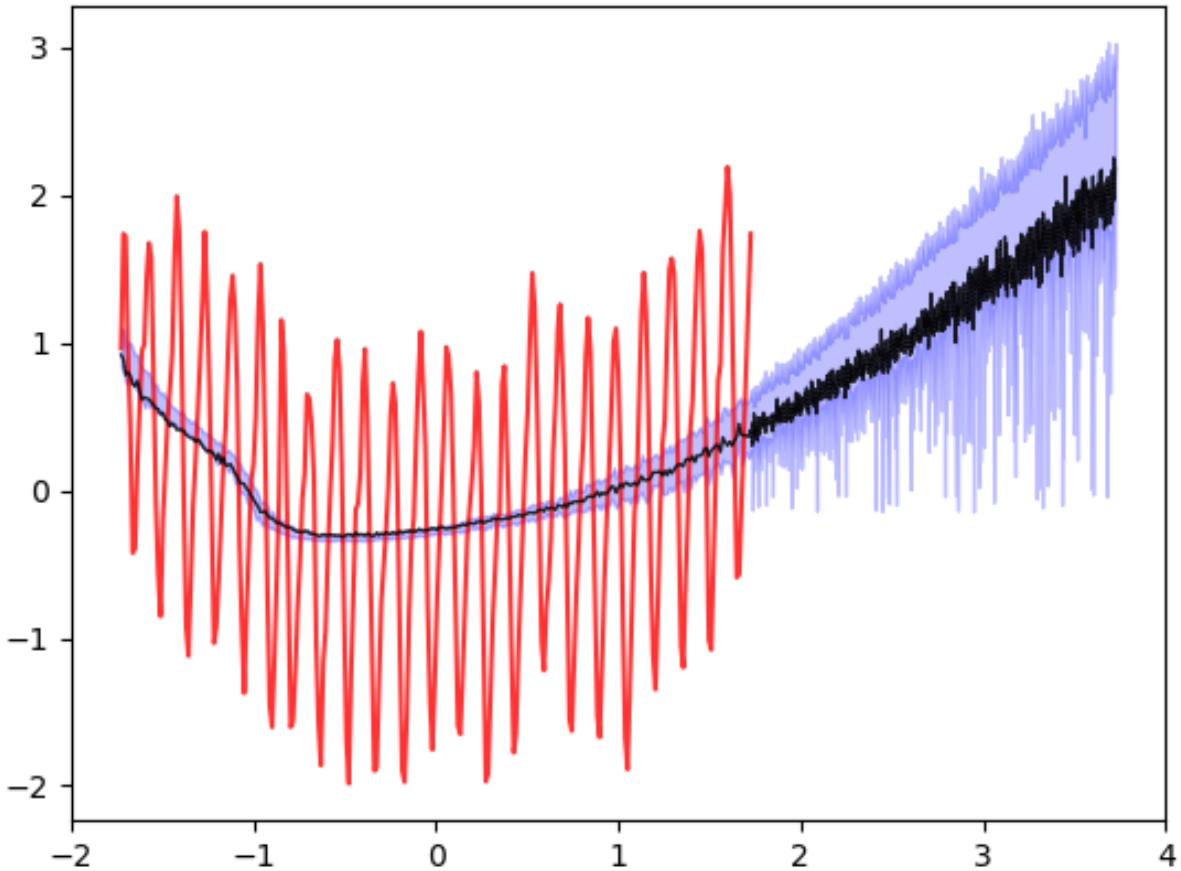

Figure: Softplus; 5 Hidden Layers; 1024 Neurons per Layer

Experiment 5: What Activation Function?

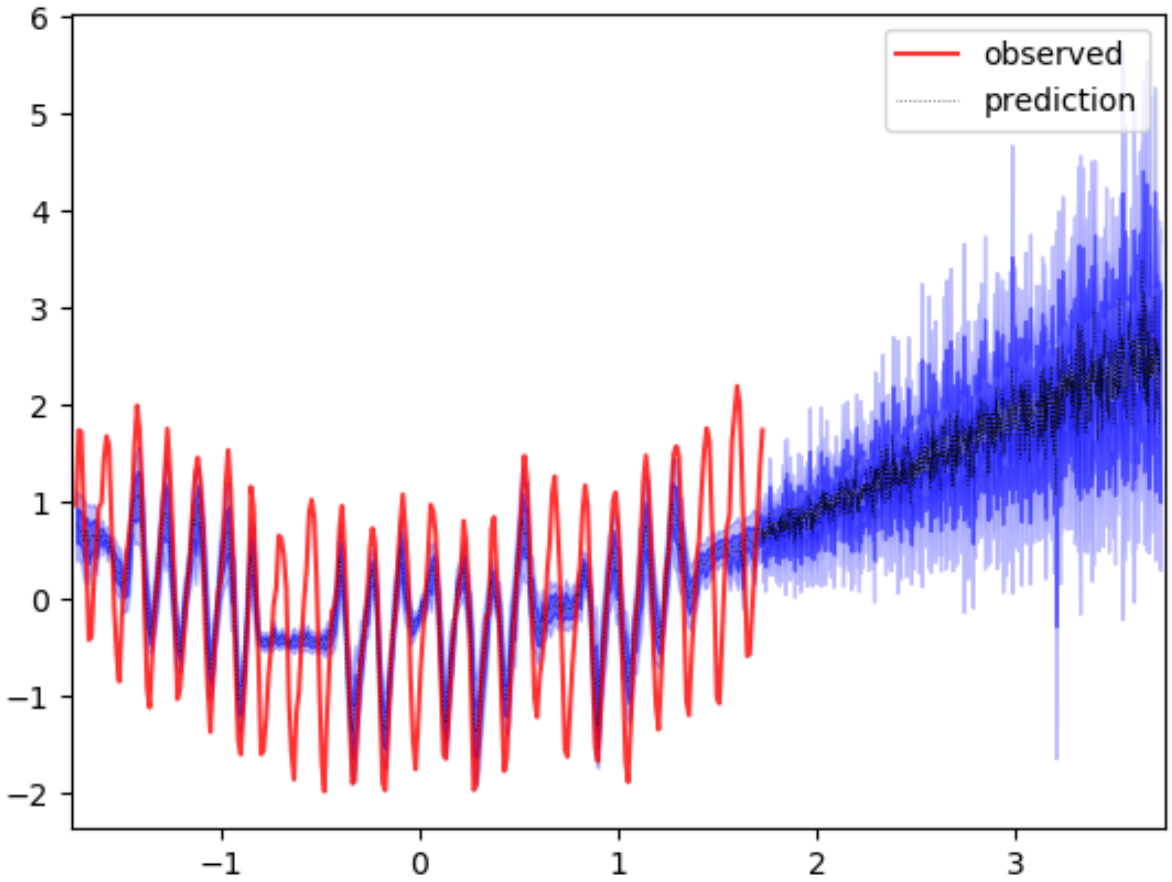

Figure: Softplus; 5 Hidden Layers; 1024 Neurons per Layer