Exploring Bayesian Methods to Improve Neural Network Learning

Shruthi Sukumar, Santhanakrishnan Ramani, Shirly Montero, Shane Grigsby

CSCI 5622 - Machine Learning

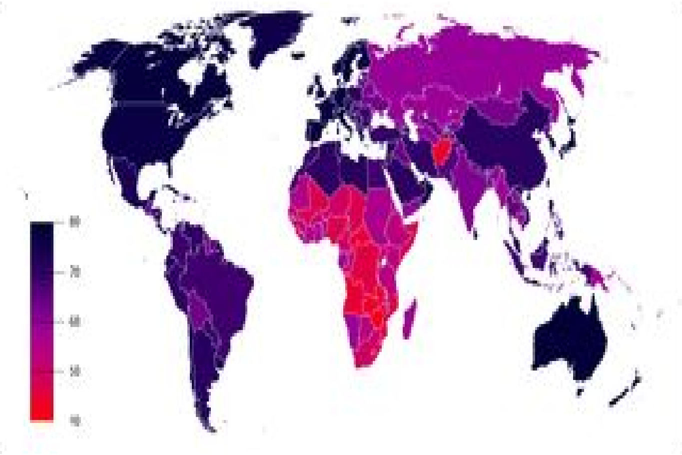
Neural Networks

- Model Selection
- Tuning Hyper-parameters
    - Specific to network architecture
    - Orthogonal to the weight update based on training

Dataset

World Development Indicators

- ➢ Using World Development Indicators, to predict the life expectancy for a given country in a given year.

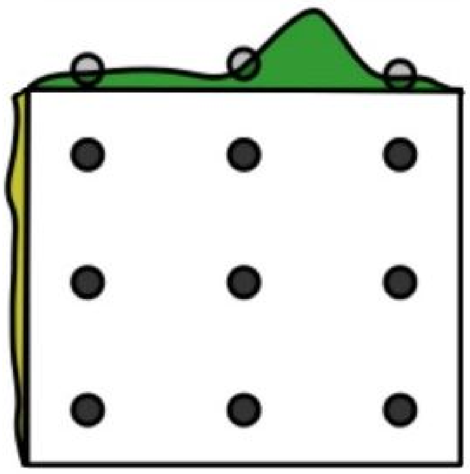


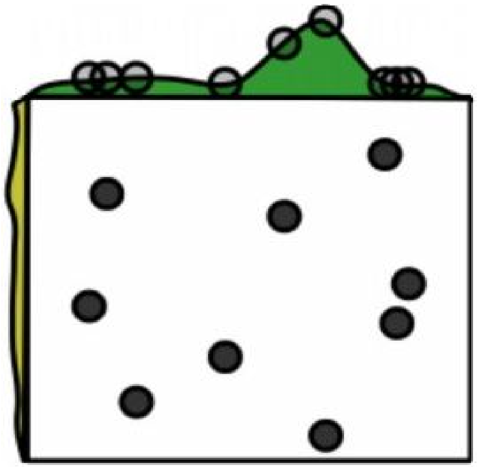
Neural Network Parameters

- ➢ Neurons - INT - range(5, 100)
- ➢ Activation Function for Layer 1 - ENUM - ["sigmoid", "relu"]
- ➢ Activation Function for Layer 2 - ["relu"]
- ➢ Weight Initialization - ENUM - ["glorot_uniform", "glorot_normal"]
- ➢ Weight Decay - FLOAT - range(0, 0.1)
- ➢ Dropout - FLOAT - range(0.25, 0.75)
- ➢ Number of epochs - INT - range(30, 100)

Hyperparameter Tuning Algorithms

- ➢ Grid Search
  - ○ Exhaustively Searching
  - ○ Inefficient
- ➢ Random Search
  - ○ Randomly Searching
  - ○ Better compared to Grid Search





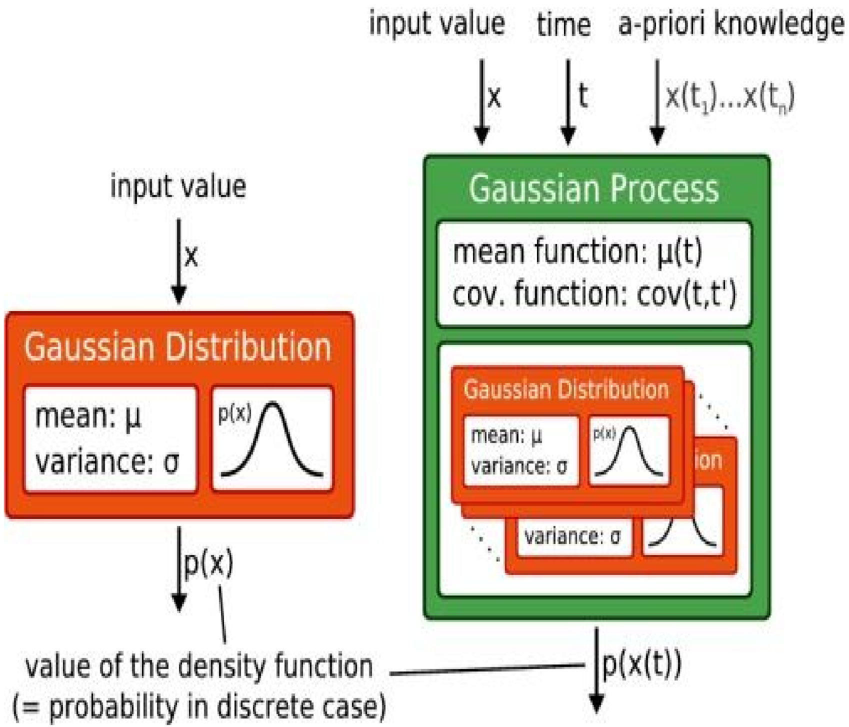
Bayesian Awesomeness

- Constructs a Probabilistic Model for the Objective to be optimized
- Uses the Model to make Decisions about which point to sample next
- Seems like a promising way to go for tuning hyperparameters

Toolbox's For Bayesian Optimization

Gaussian Processes

- ➢ Prior over the functions is assumed to be a Gaussian Process.
- ➢ The advantage of using GP priors
  - ○ Closed form of Marginals and Conditionals can be computed in a convenient form.
- ➢ GP is defined completely by
  - ○ mean function
  - ○ covariance/kernel function.



http://www.cvlibs.net/projects/gausspro/

Acquisition function

- Viewed as the objective that dictates to the GP which points to be sampled next.
- Balances exploration vs exploitation when sampling points from the search space.
- Various Acquisition functions:
    - Expected Improvement
    - Probability of Improvement
    - Upper Confidence Bound
    - Thompsons sampling

Expected Improvement

- Selects the next point to sample from the search space, which returns maximum expected improvement over the target we want to beat.
- Here we look for improvement in validation loss, which is the objective we optimize.

$$
\begin{aligned}
a_{\mathrm{EI}}(x) = \mathbb{E}\big[u(x) \mid x, \mathcal{D}\big] &= \int_{-\infty}^{f'} (f' - f)\,\mathcal{N}\big(f; \mu(x), K(x,x)\big)\,\mathrm{d}f \\
&= \big(f' - \mu(x)\big)\Phi\big(f'; \mu(x), K(x,x)\big) + K(x,x)\mathcal{N}\big(f'; \mu(x), K(x,x)\big).
\end{aligned}
$$

RESULTS

Random Search

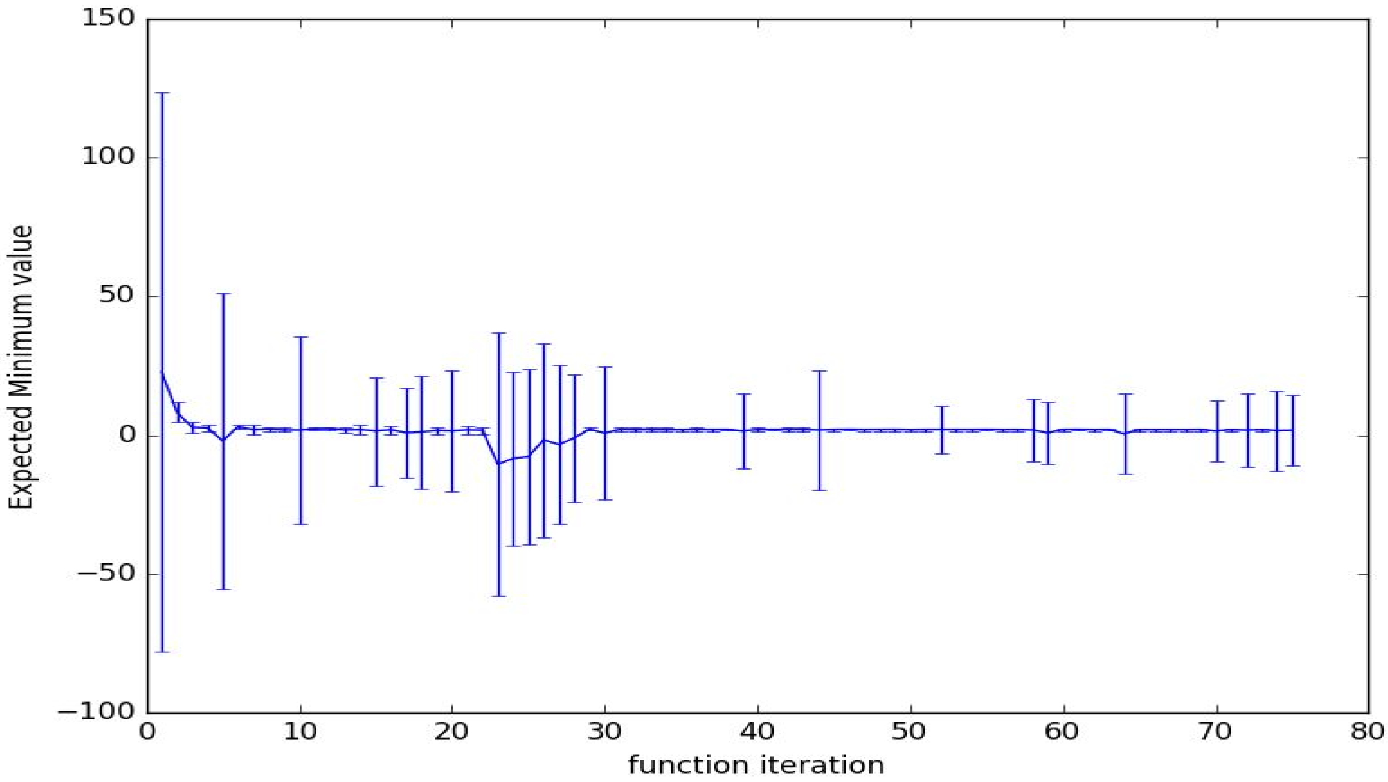
- 'wd': 0.0001,
- 'errFunc': 'mse',
- 'nb_epoch': 100,
- 'activation': ['sigmoid', ‘relu’],
- 'neurons': 70,
- 'w_init': 'glorot_normal',
- 'dropout': 0.35

Validation Loss: **1.7967**

Spearmint

- 'wd': 0.015206,
- 'errFunc': 'mse',
- 'nb_epoch': 100,
- 'activation': ['sigmoid', ‘relu’],
- 'neurons': 100,
- 'w_init': 'glorot_uniform',
- 'dropout': 0.25

Validation Loss: **1.669**



Expected Minimum value
function iteration
150
100
50
0
−50
−100
0
10
20
30
40
50
60
70
80

Bayesian Neural Networks

- ➢ Improve generalization by inferring posterior over weights of a neural network.
- ➢ Alternative to backpropagation algorithm which uses local information like derivative or gradient of error.
- ➢ Could possibly eliminate the use of hold-out set for validation as it attempts to estimate true posterior distribution given the training data seen so far.

Approximate Bayesian Inference

- ➢ To identify the closed-form expression for the posterior distribution.
- ➢ Relies on approximate inference techniques like MCMC and Variational Inference.
- ➢ Here, we have attempted this method on a simple multi-layer perceptron network with a single hidden layer, and hence have used MCMC.

Markov Chain Monte Carlo

- One of approximate inference methods that samples from a posterior distribution.
- Utilizes the property of markov chains' ability to arrive at an equilibrium distribution after sampling multiple times, depending on the transition function used.

$$p(x^{(i)}) = \sum_{x^{(i-1)}} p(x^{(i-1)}) T(x^{(i)} \mid x^{(i-1)}).$$

Continued...

- ➢ MCMC methods are not ideal for today’s scale of deep learning models because of the huge number of weights.
- ➢ However with the advent of Variational Inference, there has been a revival of neural networks in the paradigm of probabilistic modeling.

Gibbs Sampling

1. set t = 0
2. generate an initial state $x^{(0)} \sim \pi^{(0)}$
3. repeat until t = M

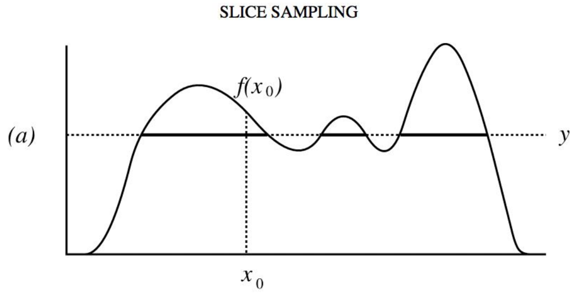
    set t = t+1

    for each dimension i = 1..D

    draw x_i from $p(x_i | x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_D)$

- ➢ Requires a sequential update of each one of the weights separately from an individual conditional distribution.
- ➢ Even for our simple network the number of weights are of the order ~16,000.
- ➢ Reason why gibbs sampling is infeasible.

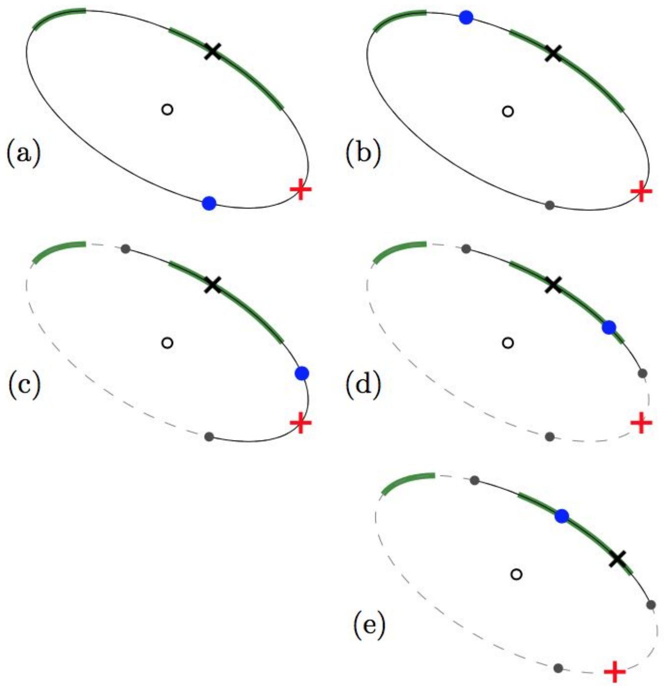
Slice Sampling



- Analogous to Metropolis-Hastings algorithm for MCMC.
- Automatically tunes the step size according to the local shape of the density function.
- Similar to Gibbs and Metropolis-Hastings sampling with respect to sequential update.

Elliptical Slice sampling

- Makes use of the update procedure in Slice sampling with the adaptive step size.
- Helpful because it updates multiple variables in one update step as opposed to Gibbs and slice sampling.



Probabilistic model

Define the observation model :

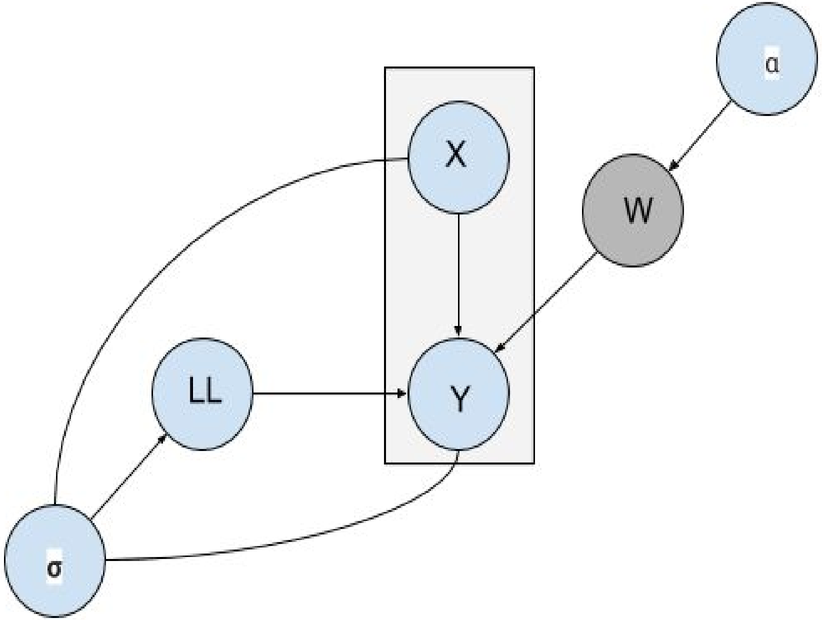
$$y|x \sim lnN\ (f(x), \sigma^2)$$

Where σ^2 is the standard deviation which defines the observation noise,

$$\tau \sim Gamma(a, b)$$

$$\tau|D, W \sim Gamma(a + \frac{n}{2}, b + \sum_{i=1}^{n} \frac{\left(y - f(x)\right)^2}{2})$$

$$f(x) - Neural\ network$$



References

[1] Bayesian Learning for Neural Networks; Radford Neal (1995)

[2] Practical Bayesian Optimization for Machine Learning Algorithms; Snoek. J, Adams, R. P, MacKay, J. C (2012)

[3] Bayesian Methods for Adaptive Models; David J C Mackay (1991)

[4] Practical Variational Inference for Neural Networks; Alex Graves (2011) NIPS

[5] On Modern Deep Learning and Variational Inference; Yarin Gal & Zoubin Gharamani (2015)