CSCI-GA.2565-001 Machine Learning: Homework 2

Due April 9, 2019

1. **Backpropagation in Recurrent Neural Network** Suppose we have a recurrent neural network (RNN). The recursive function is:

$$\begin{aligned} \mathbf{z}_{t-1} &= \mathbf{W}\mathbf{x}_{t-1} + \mathbf{U}\mathbf{h}_{t-1}, \\ \mathbf{h}_t &= g(\mathbf{z}_{t-1}), \end{aligned}$$

where $\mathbf{h}_t$ is the hidden state and $\mathbf{x}_t$ is the input at time step t. $\mathbf{W}$ and $\mathbf{U}$ are the weighted matrix. g is an element-wise activation function. And $\mathbf{h}_0$ is a given fixed initial hidden state and $\mathbf{x}_0$ is the "start of sequence" token. We model the conditional distribution with the help of the recurrent neural network:

$$\log p(\mathbf{x}_t|\mathbf{x}_{<t}) = \log p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{h}_t),$$

where $p_{\boldsymbol{\theta}}$ is a parameterized conditional probability model. Our loss function for one data point is the negative log-likelihood $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$:

$$\mathcal{L}(\mathbf{U}, \mathbf{W}, \boldsymbol{\theta}) = -\sum_{t=1}^{T} \log p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{h}_t).$$

   (a) Calculate the gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{U}}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{W}}$. You can use the symbol $\mathcal{L}_t = -\log p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{h}_t)$ in your calculation and you do not need to specify $\frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t}$.

   (b) Suppose g' is always greater than λ and the smallest eigenvalue of U is larger than $1/\lambda$. What will happen to the gradient in (a).

   (c) Suppose g' is always smaller than λ and the largest eigenvalue of U is smaller than $1/\lambda$. What will happen to the gradient in (a).

   (d) How do you fix the problem in (b) and (c)?

2. **Variational Inference**

In variational inference, we want to minimize the ELBO with respect to λ:

$$\text{ELBO}(\lambda) = \mathbb{E}_{q_\lambda}(z|x)[\log p(x|z)] - KL(q_\lambda(z|x)\|p(z)).$$

(a) Variational autoencoder uses a parameterized function $p_\theta(x|z)$ and aims to minimize the ELBO over λ and θ. We can also use mean-field variational inference for $q(z)$. Comparing variational distribution in variational autoencoder and mean field variational inference, which one is more expressive?

(b) Variational inference with normalizing flows uses the help of normalizing flow to construct the variational distribution $q(z)$. By normalizing flow, we obtain z_K with density q_K by transforming z_0 with distribution q_0 through a series of transformations $f_1, \ldots, f_K$:

$$z_K = f_K \circ \cdots \circ f_2 \circ f_1(z_0),$$
$$\ln q_K(z_K) = \ln q_0(z_0) - \sum_{k=1}^{K} \ln |\det \frac{\partial f_k}{\partial z_{k-1}}|.$$

How can a normalizing flow help move beyond the mean field? Normalizing flows can be combined with inference networks? Compute the gradient.

(c) Suppose the random variable z is one dimensional, and we have a new variational distribution that transforms a univariate standard normal with $t(z) = \sum_{i=1}^{K} v_i \exp(-\|z - \mu_i\|^2/\sigma_i^2)$. And the variational parameters here are $v_i, \mu_i, \sigma_i,\ i = 1, \ldots, K$. Calculate the gradient of the ELBO with respect to the variational parameters. What densities can this transformation approximate?

3. **Gaussian Process**

Let's define a Gaussian Process $y|\mathbf{x} \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$ where

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\frac{1}{2}\left\|\mathbf{x} - \mathbf{x}'\right\|^2),$$

such that for any finite collection of data $\mathcal{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$,

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \Bigg| \mathbf{x}_1, ..., \mathbf{x}_n \sim \mathcal{N}(\vec{0}, K)$$

where K is the covariance matrix and $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

(a) What happens if we use the above Gaussian Process to predict a test data point $\mathbf{x}^*$ far away from the training set $\mathcal{D}$?

(b) What happens if we use a neural network to predict a test data point $\mathbf{x}^*$ far away from the training set $\mathcal{D}$?

(c) Compare the results in part (a) and part (b). Explain the difference. If one has a problem, how to solve it?

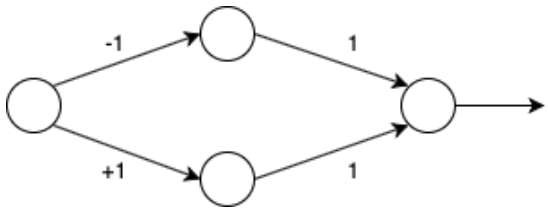
4. **Prediction using Bayesian Neural Networks**

Consider a true data generating distribution:

$$P(\mathbf{x}) \sim F$$

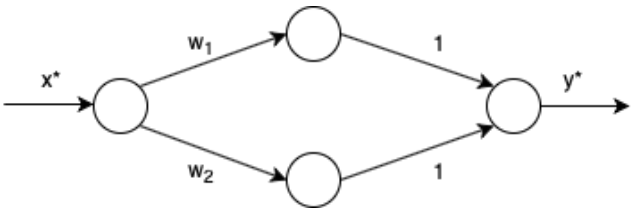
$$P(\boldsymbol{y}|\mathbf{x}) \sim \text{Normal}(f(x), 1)$$

where $f_w(x)$ is computed using the given neural network:



We observe $\mathcal{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_2, y_2)\}$ as our training data with a very large n (think 10^{12})

Now, Consider the given Bayesian neural network:



$$P(\mathbf{w}_1) = \text{Uniform}[-10, 10]$$

$$P(\mathbf{w}_2) = \text{Uniform}[-10, 10]$$

$$P(\boldsymbol{y}|\mathbf{x}) \sim \text{Normal}(f_w(\mathbf{x}), 1)$$

(a) What would be the prediction of a test point $\mathbf{x}^*$ under the posterior predictive distribution?

(b) What would be the prediction of a test point $\mathbf{x}^*$ under the likelihood at the posterior expected weights?

(c) Explain the difference between (a) and (b)