Assignment #3
Due: 11:59pm EST, March 11, 2022

Homework 3: Bayesian Methods and Neural Networks

Introduction

This homework is about Bayesian methods and Neural Networks. Section 2.9 in the textbook as well as reviewing MLE and MAP will be useful for Q1. Chapter 4 in the textbook will be useful for Q2.
Please type your solutions after the corresponding problems using this LaTeX template, and start each problem on a new page.
Please submit the **writeup PDF to the Gradescope assignment 'HW3'**. Remember to assign pages for each question. **All plots you submit must be included in your writeup PDF.** We will not be checking your code / source files except in special circumstances.
Please submit your **LaTeXfile and code files to the Gradescope assignment 'HW3 - Supplemental'**.

Problem 1 (Bayesian Methods)

This question helps to build your understanding of making predictions with a maximum-likelihood estimation (MLE), a maximum a posterior estimator (MAP), and a full posterior predictive.
Consider a one-dimensional random variable $x = \mu + \epsilon$, where it is known that $\epsilon \sim N(0, \sigma^2)$. Suppose we have a prior $\mu \sim N(0, \tau^2)$ on the mean. You observe iid data $\{x_i\}_{i=1}^n$ (denote the data as D).
We derive the distribution of $x|D$ for you.
The full posterior predictive is computed using:

$$p(x|D) = \int p(x, \mu|D)d\mu = \int p(x|\mu)p(\mu|D)d\mu$$

One can show that, in this case, the full posterior predictive distribution has a nice analytic form:

$$x|D \sim \mathcal{N}\left(\frac{\sum_{x_i \in D} x_i}{n + \frac{\sigma^2}{\tau^2}}, (\frac{n}{\sigma^2} + \frac{1}{\tau^2})^{-1} + \sigma^2\right) \tag{1}$$

1. Derive the distribution of $\mu|D$.

2. In many problems, it is often difficult to calculate the full posterior because we need to marginalize out the parameters as above (here, the parameter is μ). We can mitigate this problem by plugging in a point estimate of μ^* rather than a distribution.

   a) Derive the MLE estimate μ_{MLE}.

   b) Derive the MAP estimate μ_{MAP}.

   c) What is the relation between μ_{MAP} and the mean of $x|D$?

   d) For a fixed value of $\mu = \mu^*$, what is the distribution of $x|\mu^*$? Thus, what is the distribution of $x|\mu_{MLE}$ and $x|\mu_{MAP}$?

   e) Is the variance of $x|D$ greater or smaller than the variance of $x|\mu_{MLE}$? What is the limit of the variance of $x|D$ as n tends to infinity? Explain why this is intuitive.

3. Let us compare μ_{MLE} and μ_{MAP}. There are three cases to consider:

   a) Assume $\sum_{x_i \in D} x_i = 0$. What are the values of μ_{MLE} and μ_{MAP}?

   b) Assume $\sum_{x_i \in D} x_i > 0$. Is μ_{MLE} greater than μ_{MAP}?

   c) Assume $\sum_{x_i \in D} x_i < 0$. Is μ_{MLE} greater than μ_{MAP}?

4. Compute:

$$\lim_{n \to \infty} \frac{\mu_{MAP}}{\mu_{MLE}}$$

Solution 1.1:

To derive the posterior distribution of $\mu|D$, we apply Bayes rule, giving us:

$$\begin{aligned} p(\mu|D) &\propto p(D|\mu)p(\mu) \\ &= \prod_{i=1}^{n} p(x_i|\mu)p(\mu) \end{aligned}$$

Since $x = \mu + \epsilon$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$, then we have the distribution $x_i|\mu \sim \mathcal{N}(0, \sigma^2)$. As such, we have a normal distribution with a known variance, which is a natrual exponent family (NEF). Accordingly, we have that the sample mean $\bar{x}$ is a sufficient statistic for D, meaning the conditional distribution $\mu|D$ is the same as $\mu|\bar{x}$. This will significantly simplify our calculation as we can update the prior with all data points through the sample mean. Thus, by the central limit theorem, we have $\bar{x}|\mu \sim \mathcal{N}(\mu, \sigma^2/n)$. Accordingly, we have:

$$\begin{aligned} p(\mu|D) &= p(\mu|\bar{x}) \\ &\propto p(\bar{x}|\mu)p(\mu) \\ &= \prod_{i=1}^{n} p(x_i|\mu)p(\mu) \\ &= \frac{1}{\tau\sqrt{2\pi}} \exp\left(\frac{-(\mu)^2}{2\tau^2}\right) \cdot \frac{1}{(\sigma\sqrt{2\pi})/\sqrt{n}} \exp\left(\frac{-(\bar{x}-\mu)^2}{2\sigma^2/n}\right) \\ &= \frac{\sqrt{n}}{2\tau\sigma\pi} \exp\left(-\frac{1}{2}\left(\frac{\mu^2}{\tau^2} + \frac{(\bar{x}-\mu)^2}{\sigma^2/n}\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{\mu^2\sigma^2/n + (\bar{x}-\mu)^2\tau^2}{\tau^2\sigma^2/n}\right)\right) \\ &= \exp\left(-\frac{1}{2}\left(\frac{\mu^2\sigma^2/n + \bar{x}\tau^2 - 2\bar{x}\mu\tau^2 + \mu^2\tau^2}{\tau^2\sigma^2/n}\right)\right) \\ &= \exp\left(-\frac{1}{2}\left(\frac{\mu^2\sigma^2/n - 2\bar{x}\mu\tau^2 + \mu^2\tau^2}{\tau^2\sigma^2/n} + \frac{\bar{x}\tau^2}{\tau^2\sigma^2/n}\right)\right) \\ &= \exp\left(-\frac{1}{2}\left(\frac{\mu^2(\sigma^2/n + \tau^2) - 2\bar{x}\mu\tau^2}{\tau^2\sigma^2/n} + \frac{\bar{x}\tau^2}{\tau^2\sigma^2/n}\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{\mu^2 - 2\mu(\frac{\bar{x}\tau^2}{\sigma^2/n+\tau^2}) + (\frac{\bar{x}\tau^2}{\sigma^2/n+\tau^2})^2}{(\tau^2\sigma^2/n)/(\sigma^2/n+\tau^2)}\right)\right) \\ &= \exp\left(-\frac{1}{2}\left(\frac{(\mu - \frac{\bar{x}\tau^2}{\sigma^2/n+\tau^2})^2}{(\tau^2\sigma^2)/(\sigma^2+\tau^2 n)}\right)\right) \end{aligned}$$

Note that we have removed all constants that do not depend on μ from the above expression. Next, note that the posterior distribution is Gaussian. Thus, matching the formula of a normal PDF, we have:

$$p(\mu|D) \sim \mathcal{N} \sim \left(\frac{\bar{x}\tau^2}{\sigma^2/n + \tau^2}, \frac{\tau^2\sigma^2}{\sigma^2 + \tau^2 n}\right)$$

Solution 1.2:

(a)

To derive the MLE estimate μ_{MLE}, we maximize the likelihood of $p(D|\mu)$. We can accomplish this by maximizing $\log(p(D|\mu))$ since logarithm is a strictly increasing function which will not change our MLE

estimate μ_{MLE}. Note from problem 1.1 we have that $p(D|\mu) = \prod_i^n p(x_1|\mu)$ and $x_i|\mu \sim \mathcal{N}(\mu, \sigma^2)$ since $x = \mu + \epsilon$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. As such, we have:

$$\begin{aligned}
\log(p(D|\mu)) &= \log(\prod_i^n p(x_1|\mu)) \\
&= \sum_i^n \log(p(x_1|\mu)) \\
&\propto \sum_i^n -\frac{1}{2}(\frac{x_i - \mu}{\sigma^2})^2 \\
&= -\frac{1}{2\sigma^2}\sum_i^n (x_i - \mu)^2
\end{aligned}$$

Differentiating with respect to μ, we have:

$$\begin{aligned}
\frac{\partial}{\partial \mu}\log(p(D|\mu)) &= \frac{\partial}{\partial \mu}\left(-\frac{1}{2\sigma^2}\sum_i^n (x_i - \mu)^2\right) \\
&= -\frac{1}{2\sigma^2}\sum_i^n 2(x_i - \mu) \\
&= -\frac{1}{\sigma^2}\sum_i^n (x_i - \mu)
\end{aligned}$$

Setting to 0 and solving for μ_{MLE}, we have:

$$\begin{aligned}
0 &= -\frac{1}{\sigma^2}\sum_i^n (x_i - \mu_{MLE}) \\
0 &= \sum_i^n (x_i - \mu_{MLE}) \\
n \cdot \mu_{MLE} &= \sum_i^n x_i \\
\mu_{MLE} &= \frac{\sum_i^n x_i}{n} = \bar{x}
\end{aligned}$$

(b)

To derive the MAP estimate μ_{MAP}, we maximize the posterior $p(\mu|D)$ from problem 1.1. We can accomplish this by maximizing $\log(p(\mu|D))$ since logarithm is a strictly increasing function which will not change our MAP estimate μ_{MAP}. Thus, getting rid of additive constants, we have:

$$\log(p(\mu|D)) = \frac{-(\mu - \frac{\bar{x}\tau^2}{\sigma^2/n + \tau^2})^2}{2(\frac{\tau^2\sigma^2}{\sigma^2 + n\tau^2})^2}$$

Differentiating with respect to μ, we have:

$$\begin{aligned}\frac{\partial}{\partial \mu}\log(p(\mu|D)) &= \frac{\partial}{\partial \mu}\left(\frac{-(\mu - \frac{\bar{x}\tau^2}{\sigma^2/n+\tau^2})^2}{2(\frac{\tau^2\sigma^2}{\sigma^2+n\tau^2})^2}\right) \\ &= \frac{-(\mu - \frac{\bar{x}\tau^2}{\sigma^2/n+\tau^2})}{(\frac{\tau^2\sigma^2}{\sigma^2+n\tau^2})^2}\end{aligned}$$

Setting to 0 and solving for μ_{MAP}, we have:

$$\begin{aligned}0 &= \frac{-(\mu_{MAP} - \frac{\bar{x}\tau^2}{\sigma^2/n+\tau^2})}{(\frac{\tau^2\sigma^2}{\sigma^2+n\tau^2})^2} \\ 0 &= -(\mu_{MAP} - \frac{\bar{x}\tau^2}{\sigma^2/n+\tau^2}) \\ \mu_{MAP} &= \frac{\bar{x}\tau^2}{\sigma^2/n+\tau^2}\end{aligned}$$

(c)

From part (b), we have $\mu_{MAP} = \frac{\bar{x}\tau^2}{\sigma^2/n+\tau^2}$. The mean of the posterior predictive distribution $x|D$ can be given by:

$$\begin{aligned}E(x|D) &= \frac{\sum_{x\in D} x_i}{n + \frac{\sigma^2}{\tau^2}} \\ &= \frac{\bar{x}}{1 + \frac{\sigma^2}{(n\tau^2)}} \\ &= \frac{\bar{x}\tau^2}{\sigma^2/n + \tau^2} \\ &= \mu_{MAP}\end{aligned}$$

Thus, we have $\mu_{MAP} = E(x|D)$.

(d)

For a fixed $\mu = \mu^*$, since $x = \mu + \epsilon$ and $\epsilon \sim N(0, \sigma^2)$, we have $x|\mu^* = \mu^* + \epsilon$ so the distribution of $x|\mu^* \sim \mathcal{N}(\mu^*, \sigma^2)$. Thus, we have:

$$\begin{aligned}x|\mu_{MLE} &\sim \mathcal{N}(\mu_{MLE}, \sigma^2) \\ x|\mu_{MAP} &\sim \mathcal{N}(\mu_{MAP}, \sigma^2)\end{aligned}$$

(e)

From the problem statement, we have that the variance of $x|D$ is $(\frac{n}{\sigma^2} + \frac{1}{\tau^2})^{-1} + \sigma^2$ while the variance of $x|\mu_{MLE}$ is σ^2. Since $(\frac{n}{\sigma^2} + \frac{1}{\tau^2})^{-1} > 0$, the variance of $x|D$ is greating than the variance of $x|\mu_{MLE}$. This makes sense intuitively as when conditioning on D, we have higher uncertainty associated with μ as we are accounting for all models whereas when we condition on μ_{MLE}, we are accounting soley for the point estimate μ_{MLE}. We are in essence treating μ_{MLE} as an ideal estimator, so we would expect $x|\mu_{MLE}$ to have a smaller variance. We can also note that for $x|D$, we have uncertainty with respect to both μ and ϵ

whereas in $x|\mu_{MLE}$, we have fixed μ and our uncertainty comes from ϵ. Taking the limit of variance of $x|D$ as n goes to infinity we have:

$$\lim_{n\to\infty}(x|D) = \lim_{n\to\infty}(\frac{n}{\sigma^2}+\frac{1}{\tau^2})^{-1}+\sigma^2 = \sigma^2$$

This makes sense intuitively because as the number n of data points observed increases, we are more certain about the distribution of μ, leading the contribution to variance of μ to become smaller and smaller.

Solution 1.3:

From problem 1.2, we have that:

$$\mu_{MLE} = \bar{x} = \frac{\sum_{x_i\in D} x_i}{n}$$
$$\mu_{MAP} = \frac{\bar{x}\tau^2}{\sigma^2/n+\tau^2} = \frac{\sum_{x_i\in D} x_i}{n}\cdot\frac{\tau^2}{\sigma^2/n+\tau^2}$$

(a)

In the case that $\sum_{x_i\in D} x_i = 0$, we have:

$$\mu_{MLE} = \frac{\sum_{x_i\in D} x_i}{n} = 0$$
$$\mu_{MAP} = \frac{\sum_{x_i\in D} x_i}{n}\cdot\frac{\tau^2}{\sigma^2/n+\tau^2} = 0$$

So $\mu_{MLE} = \mu_{MAP}$.

(b)

In the case that $\sum_{x_i\in D} x_i > 0$, let us note that:

$$\frac{\tau^2}{\sigma^2/n+\tau^2} < 1$$

Thus in this case, since $\mu_{MLE} = \frac{\sum_{x_i\in D} x_i}{n}$ and $\mu_{MAP} = \frac{\sum_{x_i\in D} x_i}{n}\cdot\frac{\tau^2}{\sigma^2/n+\tau^2}$, we have $\mu_{MAP} < \mu_{MLE}$.

(c)

In the case that $\sum_{x_i\in D} x_i < 0$, let us once again note that:

$$\frac{\tau^2}{\sigma^2/n+\tau^2} < 1$$

Thus in this case, since $\mu_{MLE} = \frac{\sum_{x_i\in D} x_i}{n}$ and $\mu_{MAP} = \frac{\sum_{x_i\in D} x_i}{n}\cdot\frac{\tau^2}{\sigma^2/n+\tau^2}$, we have $\mu_{MAP} > \mu_{MLE}$.

Solution 1.4:

We can compute the limit as follows:

$$\lim_{n\to\infty}\frac{\mu_{MAP}}{\mu_{MLE}} = \lim_{n\to\infty}\frac{\frac{\bar{x}\tau^2}{\sigma^2/n+\tau^2}}{\bar{x}} = \lim_{n\to\infty}\frac{\tau^2}{\sigma^2/n+\tau^2} = 1$$

Problem 2 (Bayesian Frequentist Reconciliation)

In this question, we connect the Bayesian version of regression with the frequentist view we have seen in the first week of class by showing how appropriate priors could correspond to regularization penalities in the frequentist world, and how the models can be different.

Suppose we have a $(p+1)$-dimensional labelled dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. We can assume that y_i is generated by the following random process:

$$y_i = \mathbf{w}^\top \mathbf{x}_i + \epsilon_i$$

where all $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ are iid. Using matrix notation, we denote

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_N \end{bmatrix}^\top \in \mathbb{R}^{N \times p}$$
$$\mathbf{y} = \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}^\top \in \mathbb{R}^N$$
$$\epsilon = \begin{bmatrix} \epsilon_1 & \dots & \epsilon_N \end{bmatrix}^\top \in \mathbb{R}^N.$$

Then we can write have $\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon$. Now, we will suppose that $\mathbf{w}$ is random as well as our labels! We choose to impose the Laplacian prior $p(\mathbf{w}) = \frac{1}{2\tau} \exp\left(-\frac{\|\mathbf{w}-\mu\|_1}{\tau}\right)$, where $\|\mathbf{w}\|_1 = \sum_{i=1}^p |w_i|$ denotes the L^1 norm of $\mathbf{w}$, μ the location parameter, and τ is the scale factor.

1. Compute the posterior distribution $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$ of $\mathbf{w}$ given the observed data $\mathbf{X}, \mathbf{y}$, up to a normalizing constant. You **do not** need to simplify the posterior to match a known distribution.

2. Determine the MAP estimate $\mathbf{w}_{\text{MAP}}$ of $\mathbf{w}$. You may leave the answer as the solution to an equation. How does this relate to regularization in the frequentist perspective? How does the scale factor τ relate to the corresponding regularization parameter λ? Provide intuition on the connection to regularization, using the prior imposed on $\mathbf{w}$.

3. Based on the previous question, how might we incorporate prior expert knowledge we may have for the problem? For instance, suppose we knew beforehand that $\mathbf{w}$ should be close to some vector $\mathbf{v}$ in value. How might we incorporate this in the model, and explain why this makes sense in both the Bayesian and frequentist viewpoints.

4. As τ decreases, what happens to the entries of the estimate $\mathbf{w}_{\text{MAP}}$? What happens in the limit as $\tau \to 0$?

5. Consider the point estimate $\mathbf{w}_{\text{mean}}$, the mean of the posterior $\mathbf{w}|\mathbf{X}, \mathbf{y}$. Further, assume that the model assumptions are correct. That is, $\mathbf{w}$ is indeed sampled from the posterior provided in subproblem 1, and that $y|\mathbf{x}, \mathbf{w} \sim \mathcal{N}(\mathbf{w}^T\mathbf{x}, \sigma^2)$. Suppose as well that the data generating processes for $\mathbf{x}, \mathbf{w}, y$ are all independent (note that $\mathbf{w}$ is random!). Between the models with estimates $\mathbf{w}_{\text{MAP}}$ and $\mathbf{w}_{\text{mean}}$, which model would have a lower expected test MSE, and why? Assume that the data generating distribution for $\mathbf{x}$ has mean zero, and that distinct features are independent and each have variance 1.[a]

[a] The unit variance assumption simplifies computation, and is also commonly used in practical applications.

Solution 2.1:

We compute the posterior distribution $p(\mathbf{w}|\mathbf{X},\mathbf{y})$ of $\mathbf{w}$ given the observed data $\mathbf{X},\mathbf{y}$, up to a normalizing constant in the following manner. Note that we have $y_i|\mathbf{x_i},\mathbf{w} \sim \mathcal{N}(\mathbf{w}^T\mathbf{x_i},\sigma^2)$ since $y_i = \mathbf{w}^T\mathbf{x_i} + \epsilon_i$, and $\epsilon \sim \mathcal{N}(0,\sigma^2)$. Thus, by Naive Bayes assumption and Bayes rule we have:

$$\begin{aligned}
p(\mathbf{w}|\mathbf{X},\mathbf{y}) &\propto p(\mathbf{y}|\mathbf{X},\mathbf{w})p(\mathbf{w}) \\
&= \prod_{i=1}^{N} p(y_i|\mathbf{x_i},\mathbf{w})p(\mathbf{w}) \\
&= \prod_{i=1}^{N} \mathcal{N}(\mathbf{w}^T\mathbf{x_i},\sigma^2)p(\mathbf{w}) \\
&= \prod_{i=1}^{N} \left(\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{(y_i-\mathbf{w}^T\mathbf{x_i})^2}{\sigma^2}\right)\right)\left(\frac{1}{2\tau}\exp\left(\frac{-\|\mathbf{w}-\mu\|_1}{\tau}\right)\right) \\
&\propto \exp\left(\frac{-\|\mathbf{w}-\mu\|_1}{\tau}\right)\prod_{i=1}^{N}\exp\left(-\frac{1}{2\sigma^2}(y_i-\mathbf{w}^T\mathbf{x_i})^2\right) \\
&= \exp\left(\frac{-\|\mathbf{w}-\mu\|_1}{\tau} - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i-\mathbf{w}^T\mathbf{x_i})^2\right)
\end{aligned}$$

Solution 2.2:

To find the MAP estimate $\mathbf{w_{MAP}}$ of $\mathbf{w}$, we must find the $\mathbf{w}$ which maximizes the posterior distribution $p(\mathbf{w}|\mathbf{X},\mathbf{y})$ which we derived up to a normalizing constant in problem 2.1. We can accomplish this by maximizing $\log p(\mathbf{w}|\mathbf{X},\mathbf{y})$ since logarithm is a strictly increasing function which will not change our MAP estimate. Thus, since $\mathbf{w_{MAP}} = argmax_w \log p(\mathbf{w}|\mathbf{X},\mathbf{y})$, we have:

$$\begin{aligned}
\mathbf{w_{MAP}} &= argmax_w \log p(\mathbf{w}|\mathbf{X},\mathbf{y}) \\
&= argmax_w \left(\frac{-\|\mathbf{w}-\mu\|_1}{\tau} - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i-\mathbf{w}^T\mathbf{x_i})^2\right) \\
&= argmin_w \left(\frac{1}{2}\sum_{i=1}^{N}(y_i-\mathbf{w}^T\mathbf{x_i})^2 + \frac{\sigma^2}{\tau}\|\mathbf{w}-\mu\|_1\right)
\end{aligned}$$

Taking the gradient with respect to $\mathbf{w}$ of the expression within the parenthesis and setting to 0, we have:

$$-\sum_{i=1}^{N}(y_i-\mathbf{w}^T\mathbf{x_i})\mathbf{x_i} + \frac{\sigma^2}{\tau}\nabla_w\|\mathbf{w}-\mu\|_1 = 0$$

$\mathbf{w_{MAP}}$ is the solution to this equation. From this equation, we may note that maximizing the posterior distribution is the same as minimizing the sum of squared errors and adding an L_1 regularization term $\|\mathbf{w}-\mu\|_1$ following a frequentist perspective. Specifically, adding a Laplacian prior over $\mathbf{w}$ and maximizing the posterior distribution is equivalent to Lasso regression (using the L_1 norm), where our regularization parameter λ is equal to our variance divided by the scale factor τ, or $\lambda = \frac{\sigma^2}{\tau}$. As such, for larger values of τ we impose less regularization as λ is smaller in Lasso regression.

Solution 2.3:

If we knew beforehand that $\mathbf{w}$ would be close to some vector $\mathbf{v}$ in value, from a Bayesian viewpoint, we could specify μ, the location parameter of our Laplacian prior over $\mathbf{w}$ to $\mathbf{v}$. This would maximize the density of

our prior at the vector v, which would reflect our beforehand knowledge that $\mathbf{w}$ is close to the vector $\mathbf{v}$. Once we have taken additional data into account, we can use the posterior distribution to reassess. Moreover, depending on how confident we are in our prior knowledge that $\mathbf{w}$ is close to the vector $\mathbf{v}$, we could use differing values of the scale factor τ. For example, if we were very confident, we would use smaller τ values.

From a frequentist perspective, if we knew beforehand that $\mathbf{w}$ would be close to some vector $\mathbf{v}$ in value, we could similarly add an L_1 regularization term using $\|\mathbf{w} - \mu\|_1$, to reflect our prior knowledge. Moreover, depending on how confident we are in our prior knowledge that $\mathbf{w}$ is close to the vector $\mathbf{v}$, we could use differing values of our regularization coefficient λ, as per our discussion of the relationship between λ and τ in the problem 2.2.

Solution 2.4:

As τ decreases, the entries of $\mathbf{w}_{MAP}$ will converge towards μ, similar to using a high regularization coefficient in regression. This is because deviations from μ will be penalized more and more for smaller and smaller values of τ. As such, the posterior will start to resemble the more prior as we are more confident in our prior knowledge. As $\tau \to 0$, $\mathbf{w}_{MAP}$ will become equal to μ and the posterior will become almost identical to the prior.

Problem 3 (Neural Net Optimization)

In this problem, we will take a closer look at how gradients are calculated for backprop with a simple multi-layer perceptron (MLP). The MLP will consist of a first fully connected layer with a sigmoid activation, followed by a one-dimensional, second fully connected layer with a sigmoid activation to get a prediction for a binary classification problem. Assume bias has not been merged. Let:

- $\mathbf{W}_1$ be the weights of the first layer, $\mathbf{b}_1$ be the bias of the first layer.
- $\mathbf{W}_2$ be the weights of the second layer, $\mathbf{b}_2$ be the bias of the second layer.

The described architecture can be written mathematically as:

$$\hat{y} = \sigma(\mathbf{W}_2 \left[\sigma \left(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1\right)\right] + \mathbf{b}_2)$$

where $\hat{y}$ is a scalar output of the net when passing in the single datapoint $\mathbf{x}$ (represented as a column vector), the additions are element-wise additions, and the sigmoid is an element-wise sigmoid.

1. Let:
   - N be the number of datapoints we have
   - M be the dimensionality of the data
   - H be the size of the hidden dimension of the first layer. Here, hidden dimension is used to describe the dimension of the resulting value after going through the layer. Based on the problem description, the hidden dimension of the second layer is 1.

   Write out the dimensionality of each of the parameters, and of the intermediate variables:

   $$\mathbf{a}_1 = \mathbf{W}_1 \mathbf{x} + \mathbf{b}_1, \qquad \mathbf{z}_1 = \sigma(\mathbf{a}_1)$$
   $$a_2 = \mathbf{W}_2 \mathbf{z}_1 + \mathbf{b}_2, \qquad \hat{y} = z_2 = \sigma(a_2)$$

   and make sure they work with the mathematical operations described above.

2. We will derive the gradients for each of the parameters. The gradients can be used in gradient descent to find weights that improve our model's performance. For this question, assume there is only one datapoint $\mathbf{x}$, and that our loss is $L = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$. For all questions, the chain rule will be useful.
   (a) Find $\frac{\partial L}{\partial b_2}$.
   (b) Find $\frac{\partial L}{\partial W_2^h}$, where W_2^h represents the hth element of $\mathbf{W}_2$.
   (c) Find $\frac{\partial L}{\partial b_1^h}$, where b_1^h represents the hth element of $\mathbf{b}_1$. (*Hint: Note that only the hth element of $\mathbf{a}_1$ and $\mathbf{z}_1$ depend on b_1^h - this should help you with how to use the chain rule.)
   (d) Find $\frac{\partial L}{\partial W_1^{h,m}}$, where $W_1^{h,m}$ represents the element in row h, column m in $\mathbf{W}_1$.

Solution 3.1:

The dimensionality of each of the parameters and intermediate variables is as follows:

- $x - M$ x 1
- $a_1 - H$ x 1
- $W_1 - H$ x M
- $b_1 - H$ x 1
- $z_1 - H$ x 1
- $a_2 - 1$ x 1
- $W_2 - 1$ x H
- $b_2 - 1$ x 1
- $z_2 - 1$ x 1
- $\hat{y} - 1$ x 1

Solution 3.2:

(a)

$$\begin{aligned}
\frac{\partial L}{\partial b_2} &= \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot \frac{\partial a_2}{\partial b_2} \\
&= \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot 1 \\
&= \frac{\partial L}{\partial \hat{y}} \cdot \sigma(a_2) \cdot (1 - \sigma(a_2)) \\
&= -(\frac{y}{\hat{y}} + \frac{1-y}{\hat{y}-1}) \cdot \sigma(a_2) \cdot (1 - \sigma(a_2))
\end{aligned}$$

(b)

$$\begin{aligned}
\frac{\partial L}{\partial W_2^h} &= \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot \frac{\partial a_2}{\partial W_2^h} \\
&= \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot z_1^h \\
&= \frac{\partial L}{\partial \hat{y}} \cdot \sigma(a_2) \cdot (1 - \sigma(a_2)) \cdot z_1^h \\
&= -(\frac{y}{\hat{y}} + \frac{1-y}{\hat{y}-1}) \cdot \sigma(a_2) \cdot (1 - \sigma(a_2)) \cdot z_1^h
\end{aligned}$$

(c)

$$
\begin{aligned}
\frac{\partial L}{\partial b_1^h} &= \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_1^h} \cdot \frac{\partial z_1^h}{\partial a_1^h} \cdot \frac{\partial a_1^h}{\partial b_1^h} \\
&= \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2^h} \cdot \frac{\partial a_2}{\partial z_1^h} \cdot \frac{\partial z_1^h}{\partial a_1^h} \cdot 1 \\
&= \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2^h} \cdot \frac{\partial a_2}{\partial z_1^h} \cdot \sigma(a_1^h) \cdot (1 - \sigma(a_1^h)) \\
&= \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot W_2^h \cdot \sigma(a_1^h) \cdot (1 - \sigma(a_1^h)) \\
&= \frac{\partial L}{\partial \hat{y}} \cdot \sigma(a_2) \cdot (1 - \sigma(a_2)) \cdot W_2^h \cdot \sigma(a_1^h) \cdot (1 - \sigma(a_1^h)) \\
&= -(\frac{y}{\hat{y}} + \frac{1-y}{\hat{y}-1}) \cdot \sigma(a_2) \cdot (1 - \sigma(a_2)) \cdot W_2^h \cdot \sigma(a_1^h) \cdot (1 - \sigma(a_1^h))
\end{aligned}
$$

(d)

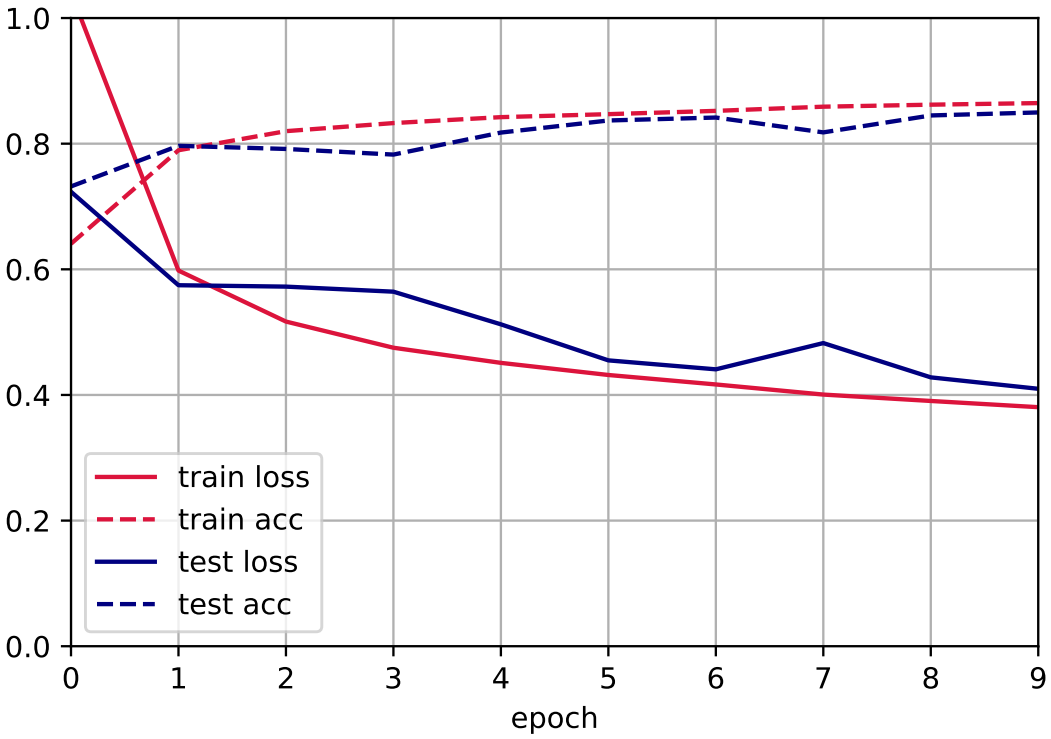
$$
\begin{aligned}
\frac{\partial L}{\partial W_1^{h,m}} &= \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_1^h} \cdot \frac{\partial z_1^h}{\partial a_1^h} \cdot \frac{\partial a_1^h}{\partial W_1^{h,m}} \\
&= \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_1^h} \cdot \frac{\partial z_1^h}{\partial a_1^h} \cdot x^m \\
&= \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_1^h} \cdot \sigma(a_1^h) \cdot (1 - \sigma(a_1^h)) \cdot x^m \\
&= \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot W_2^h \cdot \sigma(a_1^h) \cdot (1 - \sigma(a_1^h)) \cdot x^m \\
&= \frac{\partial L}{\partial \hat{y}} \cdot \sigma(a_2) \cdot (1 - \sigma(a_2)) \cdot W_2^h \cdot \sigma(a_1^h) \cdot (1 - \sigma(a_1^h)) \cdot x^m \\
&= -(\frac{y}{\hat{y}} + \frac{1-y}{\hat{y}-1}) \cdot \sigma(a_2) \cdot (1 - \sigma(a_2)) \cdot W_2^h \cdot \sigma(a_1^h) \cdot (1 - \sigma(a_1^h)) \cdot x^m
\end{aligned}
$$

Problem 4 (Modern Deep Learning Tools: PyTorch)

In this problem, you will learn how to use PyTorch. This machine learning library is massively popular and used heavily throughout industry and research. In `T3_P3.ipynb` you will implement an MLP for image classification from scratch. Copy and paste code solutions below and include a final graph of your training progress. Also submit your completed `T3_P3.ipynb` file.
You will recieve no points for code not included below.
You will recieve no points for code using built-in APIs from the `torch.nn` library.

Solution:

Plot:



Code:

```
n_inputs = 784
n_hiddens = 256
n_outputs = 10

W1 = torch.nn.Parameter(torch.normal(mean = 0, std = 0.01, size=(n_inputs, n_hiddens),
                                     requires_grad=True))
b1 = torch.nn.Parameter(torch.zeros(size= (1, n_hiddens), requires_grad=True))
W2 = torch.nn.Parameter(torch.normal(mean = 0, std = 0.01, size=(n_hiddens, n_outputs),
                                     requires_grad=True))
b2 = torch.nn.Parameter(torch.zeros(size= (1, n_outputs), requires_grad=True))
params = [W1, b1, W2, b2]



def relu(x):
```

```
  return torch.clamp(x, min = 0)



def softmax(X):
  num = torch.exp(X)
  den = torch.sum(num, dim = 1, keepdim = True)
  return num / den



def net(X):
  X_flat = X.flatten(start_dim = 1)
  H = relu((X_flat @ W1) + b1)
  O = softmax((H @ W2) + b2)
  return O



def cross_entropy(y_hat, y):
  return -torch.log(y_hat[range(len(y_hat)), y])



def sgd(params, lr=0.1):
  with torch.no_grad():
    for w in params:
      w -= lr * w.grad
      w.grad.zero_()



def train(net, params, train_iter, loss_func=cross_entropy, updater=sgd):
  for _ in range(epochs):
    for X, y in train_iter:
      y_hat = net(X)
      loss = loss_func(y_hat, y).mean()
      loss.backward()
      updater(params)
```

Name

Jamin Liu

Collaborators and Resources

Whom did you work with, and did you use any resources beyond cs181-textbook and your notes?

Calibration

Approximately how long did this homework take you to complete (in hours)?

15 hours