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

$$p(\mu|D) = \frac{p(\mu)}{p(D)}p(D|\mu)$$

$$p(\mu|D) \propto p(\mu)p(D|\mu)$$

$$p(\mu|D) \propto \exp(-\mu^2/2\tau^2)p(D|\mu)$$

$$p(\mu|D) \propto \exp(-\mu^2/2\tau^2) \times \Pi_{i=1}^{n} \exp(-\frac{(x_i-\mu)^2}{2\sigma^2})$$

$$p(\mu|D) \propto \exp(-\mu^2/2\tau^2 - \Sigma_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^2})$$

$$p(\mu|D) \propto \exp(-\frac{1}{2}(\mu^2/\tau^2 + \Sigma_{i=1}^{n}\frac{(x_i-\mu)^2}{\sigma^2}))$$

$$p(\mu|D) \propto \exp\left(-\frac{1}{2}(\frac{\mu^2}{\tau^2} + \Sigma_{i=1}^{n}\frac{x_i^2 - 2x_i\mu + \mu^2}{\sigma^2})\right)$$

$$p(\mu|D) \propto \exp\left(-\frac{1}{2}(\frac{\mu^2}{\tau^2} + \frac{n\mu^2}{\sigma^2} - \frac{2\mu}{\sigma^2}\Sigma_{i=1}^{n}x_i)\right)$$

$$p(\mu|D) \propto \exp\left(-\frac{\mu^2}{2}(\frac{1}{\tau^2} + \frac{n}{\sigma^2}) + \frac{\mu}{\sigma^2}\Sigma_{i=1}^{n}x_i\right)$$

Let $a = \frac{-1}{2}(\frac{1}{\tau^2} + \frac{n}{\sigma^2})$ and $b = \frac{1}{\sigma^2}\Sigma_{i=1}^{n}x_i$. Then:

$$p(\mu|D) \propto \exp\left(a\mu^2 + b\mu\right)$$

$$p(\mu|D) \propto \exp\left(a\mu^2 + b\mu + \frac{b^2}{4a}\right)$$

$$p(\mu|D) \propto \exp\left(a(\mu^2 + \frac{b}{a}\mu + \frac{b^2}{4a^2})\right)$$

$$p(\mu|D) \propto \exp\left(a(\mu + \frac{b}{2a})^2\right)$$

$$p(\mu|D) \propto \exp\left(\frac{1}{2}\frac{2a}{1}(\mu - \frac{-b}{2a})^2\right)$$

$$p(\mu|D) \propto \exp\left(\frac{1}{2}\frac{(\mu - \frac{-b}{2a})^2}{\sqrt{1/2a}^2}\right)$$

$$p(\mu|D) \propto \exp\left(-\frac{1}{2}\frac{(\mu - \frac{b}{2a})^2}{\sqrt{1/2a}^2}\right)$$

$$p(\mu|D) \sim \mathcal{N}\left(b/2a, \sqrt{1/2a}^2\right)$$

$$p(\mu|D) \sim \mathcal{N}\left(\frac{\Sigma_{i=1}^n x_i}{\sigma^2\left(\frac{1}{\tau^2} + \frac{n}{\sigma^2}\right)}, \sqrt{1/\left(\frac{1}{\tau^2} + \frac{n}{\sigma^2}\right)}^2\right)$$

$$p(\mu|D) \sim \mathcal{N}\left(\frac{\Sigma_{i=1}^n x_i}{\frac{\sigma^2}{\tau^2} + n}, \sqrt{1/\left(\frac{1}{\tau^2} + \frac{n}{\sigma^2}\right)}^2\right)$$

2. a)

$$\mu_{\text{MLE}} = \text{argmax}_\mu \{p(D|\mu)\}$$

$$\mu_{\text{MLE}} = \text{argmax}_\mu \{\Pi_{x_i \in D} p(x_i|\mu)\}$$

$$\mu_{\text{MLE}} = \text{argmax}_\mu \left\{\Pi_{x_i \in D} \exp(-\frac{(x_i - \mu)^2}{2\sigma^2})\right\}$$

$$\mu_{\text{MLE}} = \text{argmax}_\mu \log\left(\Pi_{x_i \in D} \exp(-\frac{(x_i - \mu)^2}{2\sigma^2})\right)$$

$$\mu_{\text{MLE}} = \text{argmax}_\mu \Sigma_{x_i \in D} \log\left(\exp(-\frac{(x_i - \mu)^2}{2\sigma^2})\right)$$

$$\mu_{\text{MLE}} = \text{argmax}_\mu \Sigma_{x_i \in D} - \frac{(x_i - \mu)^2}{2\sigma^2}$$

$$\mu_{\text{MLE}} = \text{argmin}_\mu \Sigma_{x_i \in D} \frac{(x_i - \mu)^2}{2\sigma^2}$$

$$\mu_{\text{MLE}} = \text{argmin}_\mu \Sigma_{x_i \in D} (x_i - \mu)^2$$

$$\mu_{\text{MLE}} = \text{argmin}_\mu \left(\Sigma_{i=1}^n (x_i - \mu)^2\right)$$

$$\mu_{\text{MLE}} = \frac{1}{n}\Sigma_{i=1}^n x_i$$

b) μ_{MAP} is the mode of the posterior distribution (derived in 1.1)

$$\mu_{\text{MAP}} = \text{mode}\left\{\mathcal{N}\left(\frac{\Sigma_{i=1}^{n} x_i}{\frac{\sigma^2}{\tau^2}+n}, \sqrt{1/\left(\frac{1}{\tau^2}+\frac{n}{\sigma^2}\right)^2}\right)\right\} = \frac{\Sigma_{i=1}^{n} x_i}{\frac{\sigma^2}{\tau^2}+n}$$

c) μ_{MAP} is equal to the mean of the posterior predictive distribution for $x|D$.

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?

$x = \mu + \epsilon$. Therefore, for any fixed mean dataset value $\mu = \mu^*$, the distribution of data points is:

$$x|\mu^* \sim \mathcal{N}(\mu^*, \sigma^2)$$

Thus,

$$x|\mu_{\text{MAP}} \sim \mathcal{N}(\mu_{\text{MAP}}, \sigma^2) = \mathcal{N}\left(\frac{\Sigma_{i=1}^{n} x_i}{\frac{\sigma^2}{\tau^2}+n}, \sigma^2\right)$$

and

$$x|\mu_{\text{MLE}} \sim \mathcal{N}(\mu_{\text{MLE}}, \sigma^2) = \mathcal{N}\left(\frac{1}{n}\Sigma_{i=1}^{n} x_i, \sigma^2\right)$$

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

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

$$\text{Var}\left[x|D\right] = (\frac{n}{\sigma^2}+\frac{1}{\tau^2})^{-1}+\sigma^2$$

The variance of $x|D$ is greater than the variance of $x|\mu_{\text{MLE}}$.

$$\lim_{n\to\infty} \text{Var}\left[x|D\right] = \lim_{n\to\infty} n^{-1}+\sigma^2 = \sigma^2$$

As the number of data points approaches infinity, the variance in $x|D$ approaches the variance of the noise component of x. This is because the other term in the variance of x comes from the prior on the mean, but if we have an infinitely large data set, the prior is completely outweighed by the information learned from the rest of the data.

3. 1.3 Compare μ_{MAP} and μ_{MLE}.

$$\mu_{\text{MLE}} = \frac{1}{n}\Sigma_{i=1}^{n} x_i$$

$$\mu_{\text{MAP}} = \frac{\Sigma_{i=1}^{n} x_i}{\frac{\sigma^2}{\tau^2}+n}$$

a)

If $\sum_{x_i \in D} x_i = 0$, then:

$\mu_{\text{MLE}} = \frac{1}{n}\Sigma_{i=1}^{n} x_i = \frac{1}{n} * 0 = 0$

$\mu_{\text{MAP}} = \frac{\Sigma_{i=1}^{n} x_i}{\frac{\sigma^2}{\tau^2}+n} = \frac{0}{\frac{\sigma^2}{\tau^2}+n} = 0$

So when the data sum to zero, the maximum likelihood and maximum a posteriori means of the dataset are both zero.

b)

If $\sum_{x_i \in D} x_i > 0$, let $\sum_{x_i \in D} = c > 0$.

Then $\mu_{\text{MLE}} = c/n$ and $\mu_{\text{MAP}} = c/\left(\frac{\sigma^2}{\tau^2} + n\right)$.

The values n, σ^2, τ^2 are all positive, so $\left(\frac{\sigma^2}{\tau^2} + n\right) > n$; therefore $c/n > c/\left(\frac{\sigma^2}{\tau^2} + n\right)$ and $\mu_{\text{MLE}} > \mu_{\text{MAP}}$.

c) If $\sum_{x_i \in D} x_i < 0$, let $\sum_{x_i \in D} = c < 0$.

Then $\mu_{\text{MLE}} = c/n$ and $\mu_{\text{MAP}} = c/\left(\frac{\sigma^2}{\tau^2} + n\right)$; now $c/n < c/\left(\frac{\sigma^2}{\tau^2} + n\right)$ so $\mu_{\text{MLE}} < \mu_{\text{MAP}}$.

4. 1.4

$$\lim_{n\to\infty} \frac{\mu_{MAP}}{\mu_{MLE}} = \lim_{n\to\infty} \frac{\frac{1}{\frac{\sigma^2}{\tau^2}+n}\Sigma_{i=1}^{n} x_i}{\frac{1}{n}\Sigma_{i=1}^{n} x_i} = \lim_{n\to\infty} \frac{1/\left(\frac{\sigma^2}{\tau^2} + n\right)}{1/n} = \lim_{n\to\infty} \frac{n}{\frac{\sigma^2}{\tau^2} + n} = 1$$

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 D-dimensional labelled dataset $\mathcal{D} = \{(y_i, \mathbf{x}_i)\}_{i=1}^N$. 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

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

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.

Solution:

1. 2.1

$$p(w|X, y) = \frac{p(w|X)}{p(y|X)} p(y|X, w)$$

$$p(w|X, y) \propto p(w|X) p(y|w, X)$$

$$p(w|X,y) \propto \exp\left(-\frac{\|\mathbf{w}-\mu\|_1}{\tau}\right) \times \Pi_{x_i \in \mathcal{D}} \exp\left(-\frac{1}{2}\left(\frac{y_i - w^\top x_i}{\sigma}\right)^2\right)$$

$$p(w|X,y) \propto \exp\left(-\frac{\|\mathbf{w}-\mu\|_1}{\tau} - \frac{1}{2}\Sigma_{x_i \in \mathcal{D}}\left(\frac{y_i - w^\top x_i}{\sigma}\right)^2\right)$$

$$p(w|X,y) \propto \exp\left(-\frac{\|\mathbf{w}-\mu\|_1}{\tau} - \frac{1}{2}\left(\frac{(y-Xw)^\top (y-Xw)}{\sigma^2}\right)\right)$$

2. 2.2

   $w_{\text{MAP}} = \underset{w}{\text{argmax}} \{p(w|X,y)\}$

   $w_{\text{MAP}} = \underset{w}{\text{argmin}} \left\{-\frac{1}{2\sigma^2}(y-Xw)^\top (y-Xw) - \frac{1}{\tau}||w-\mu||_1\right\}$

   $w_{\text{MAP}} = \underset{w}{\text{argmin}} \left\{(y-Xw)^\top (y-Xw) + \frac{2\sigma^2}{\tau}||w-\mu||_1\right\}$

3. 2.3

   The Bayesian approach is to use a prior with a distribution of w with most of the probability mass near v. A frequentist would perform regularization of the distance that w is from v by incorporating the term $+\lambda||v-w||_2$ into the equation, which depresses w towards v. The regularization parameter lambda can be set based on the equation in 2.2.

4. 2.4 $w_{\text{MAP}} = \underset{w}{\text{argmin}} \left\{(y-Xw)^\top (y-Xw) + \frac{2\sigma}{\tau}||w-\mu||_1\right\}$

   Very small values of tau cause the second term to become large and dominate, forcing $w = \mu$.

   Very large tau causes the regression to behave as if there is no regularization / no weight on the prior; i.e. the new evidence dominates the process of setting the weights.

5. 2.5

   w_{mean} is the mean of the generating distribution of w, whereas w_{MAP} is the mode. The error when predicting y from X is $(y-Xw)$, so the mean squared error using an estimate $\hat{w}$ of the weights is:

$$E_{X,y}\left[(y-X\hat{w})^2\right]$$

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 $\mathbf{W}_1$.

$\sigma(x) = \frac{1}{1+\exp(-x)}$

$\frac{d}{dx}\sigma(x) = \sigma(x)(1 - \sigma(x))$

Solution:

1. 3.1

   x is an input vector with dimension $M \times 1$.

   W_1 is a matrix with dimension $H \times M$.

   a_1, b_1, and z_1 are vectors with dimension $H \times 1$.

   W_2 is a row vector with dimension $1 \times H$.

   a_2 and b_2 are scalars with dimension 1.

   $\hat{y}$ is a scalar with dimension 1.

2. $$L = -(y \log(\hat{y}) + (1-y)\log(1-\hat{y}))$$

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

   a)

   $$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial b_2}$$

   $$\frac{\partial L}{\partial \hat{y}} = \frac{y - \hat{y}}{(1-\hat{y})\hat{y}}$$

   $$\frac{\partial \hat{y}}{\partial b_2} = \sigma(\mathbf{W}_2 [\sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1)] + \mathbf{b}_2) \times (1 - \sigma(\mathbf{W}_2 [\sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1)] + \mathbf{b}_2))$$

   $$\frac{\partial L}{\partial b_2} = \frac{y - \hat{y}}{(1-\hat{y})\hat{y}} \sigma(\mathbf{W}_2 [\sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1)] + \mathbf{b}_2) \times (1 - \sigma(\mathbf{W}_2 [\sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1)] + \mathbf{b}_2))$$

   $$\frac{\partial L}{\partial b_2} = y - \hat{y}$$

   b)

   Find $\frac{\partial L}{\partial W_2^h}$, where W_2^h represents the hth element of $\mathbf{W}_2$.

   $$\frac{\partial L}{\partial W_2^h} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial W_2^h}$$

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

   $$\frac{\partial L}{\partial \hat{y}} = \frac{y - \hat{y}}{(1-\hat{y})\hat{y}}$$

   $$\frac{\partial \hat{y}}{\partial W_2^h} = \frac{\partial \hat{y}}{\partial W_2} \frac{\partial W_2}{\partial W_2^h}$$

$$\frac{\partial \hat{y}}{\partial W_2} = \sigma(\mathbf{W}_2 \left[\sigma\left(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1\right)\right] + \mathbf{b}_2) \left(\sigma(\mathbf{W}_2 \left[\sigma\left(1 - \mathbf{W}_1\mathbf{x} + \mathbf{b}_1\right)\right] + \mathbf{b}_2)\right) \left[\sigma\left(1 - \mathbf{W}_1\mathbf{x} + \mathbf{b}_1\right)\right]$$

$\frac{\partial W_2}{\partial W_2^h}$ is a vector of dimension H where all entries are 0 except the hth element is 1, i.e. $\begin{bmatrix}0_1 & 0_2 & \cdots & 1_h & \cdots 0_H\end{bmatrix}$

$$\frac{\partial L}{\partial W_2^h} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial W_2} \frac{\partial W_2}{\partial W_2^h}$$

(the individual terms are shown above)

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.)

$$\frac{\partial L}{\partial b_1^h} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial b_1^h}$$

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

$$\frac{\partial L}{\partial \hat{y}} = \frac{y - \hat{y}}{(1 - \hat{y})\hat{y}}$$

$$\frac{\partial \hat{y}}{\partial b_1^h} = \frac{\partial \hat{y}}{\partial b_1} \frac{\partial b_1}{\partial b_1^h}$$

$$\frac{\partial \hat{y}}{\partial b_1} = \sigma(\mathbf{W}_2 \left[\sigma\left(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1\right)\right] + \mathbf{b}_2) \left(1 - \sigma(\mathbf{W}_2 \left[\sigma\left(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1\right)\right] + \mathbf{b}_2)\right) \mathbf{W}_2 \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \left(1 - \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)\right)$$

$$\frac{\partial b_1}{\partial b_1^h} = \begin{bmatrix}0_1 & 0_2 & \cdots & 1_h & \cdots 0_H\end{bmatrix}^\top$$

$$\frac{\partial L}{\partial b_1^h} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial b_1} \frac{\partial b_1}{\partial b_1^h}$$

d)

Find $\frac{\partial L}{\partial W_1^{h,m}}$, where $W_1^{h,m}$ represents the element in row h, column m in $\mathbf{W}_1$.

$$\frac{\partial L}{\partial W_1^{h,m}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial W_1} \frac{\partial W_1}{\partial W_1^{h,m}}$$

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.

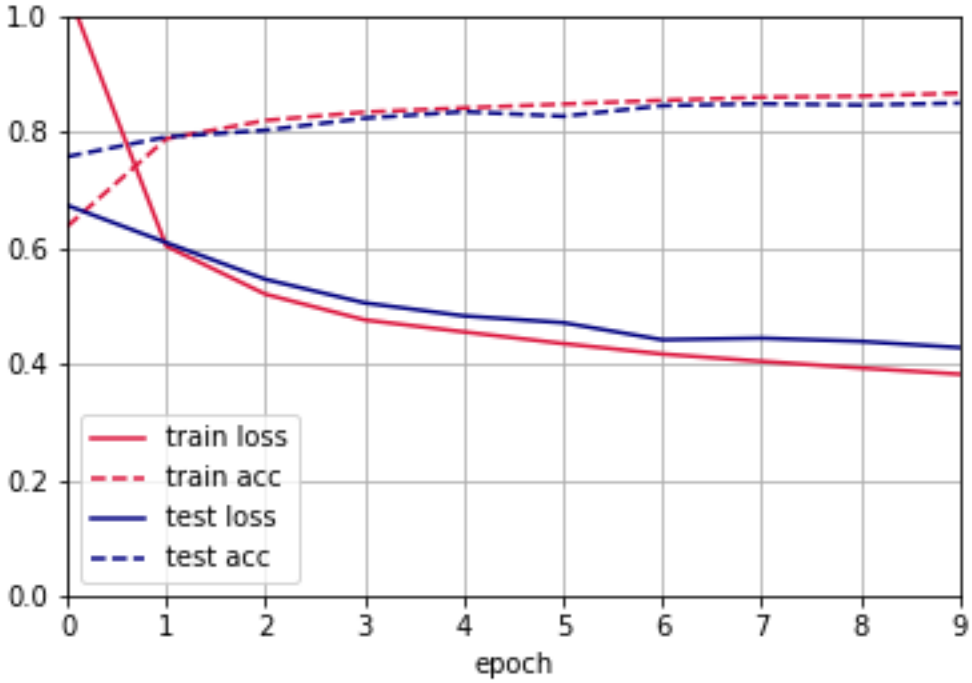
You will recieve no points for code using built-in APIs from the `torch.nn` library.

Solution:

Plot: see below

Code:

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

W1 = 0.01 * torch.randn(size=(784, 256), requires_grad=True)
W1.retain_grad()
b1 = torch.zeros(size=[256], requires_grad=True)
b1.retain_grad()
W2 = 0.01 * torch.randn(size=(256, 10), requires_grad=True)
W2.retain_grad()
b2 = torch.zeros(size=[10], requires_grad=True)
b2.retain_grad()


def relu(x):
    return torch.clamp(x, min=0)


def softmax(x):
    return torch.div(torch.exp(X), torch.sum(torch.exp(X), axis=-1, keepdim=True))


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


def cross_entropy(y_hat, y):
    return -torch.log(y_hat[torch.arange(y_hat.shape[0]), y])


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


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


1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
7
8
9
epoch
train loss
train acc
test loss
test acc

Name

Collaborators and Resources

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

Martin, Ethan, Josh Michaels, Nikola Jurkovic, Tanner Marsh

No other resources.

Calibration

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

20 hours