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 $x \sim \mu + \epsilon$, 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:

Solution 1.1:
Using Bayes' Rule:

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

Given the x_i's are iid, and $x_i \sim \mu + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$, we deduce that $x_i \sim \mathcal{N}(\mu, \sigma^2)$. Additionally, the prior on μ is given in the question:

$$\begin{aligned} p(\mu|D) &\propto \left[\prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(x_i-\mu)^2\right)\right] \frac{1}{\tau\sqrt{2\pi}} \exp\left(-\frac{1}{2\tau^2}(\mu-0)^2\right) \\ &\propto \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \frac{1}{\tau\sqrt{2\pi}} \exp\left(-\frac{1}{2\tau^2}\mu^2 - \sum_{i=1}^n \frac{1}{2\sigma^2}(x_i-\mu)^2\right) \\ &\propto \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \frac{1}{\tau\sqrt{2\pi}} \exp\left(-\frac{1}{2\tau^2}\mu^2 - \sum_{i=1}^n \frac{1}{2\sigma^2}\left(x_i^2 - 2x_i\mu + \mu^2\right)\right) \\ &\propto \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \frac{1}{\tau\sqrt{2\pi}} \exp\left(-\frac{1}{2\tau^2}\mu^2 - \frac{n}{2\sigma^2}\mu^2 + \frac{\mu}{\sigma^2}\sum_{i=1}^n x_i - \frac{1}{2\sigma^2}\sum_{i=1}^n x_i^2\right) \\ &\propto \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \frac{1}{\tau\sqrt{2\pi}} \exp\left(a\mu^2 + b\mu + c\right) \end{aligned}$$

where $a = -\frac{1}{2\tau^2} - \frac{n}{2\sigma^2}$, $b = \frac{1}{\sigma^2}\sum_{i=1}^n x_i$, and $c = -\frac{1}{2\sigma^2}\sum_{i=1}^n x_i^2$.

Completing the square inside the exponent, and dropping constant terms not involving μ, given proportionality (not equality):

$$\begin{aligned} p(\mu|D) &\propto \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \frac{1}{\tau\sqrt{2\pi}} \exp\left(a\left(\mu^2 + \frac{b}{a}\mu + \frac{c}{a}\right)\right) \\ &\propto \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \frac{1}{\tau\sqrt{2\pi}} \exp\left(a\left(\mu + \frac{b}{2a}\right)^2 + c - \frac{b^2}{4a}\right) \\ &\propto \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \frac{1}{\tau\sqrt{2\pi}} \exp\left(c - \frac{b^2}{4a}\right) \exp\left(a\left(\mu + \frac{b}{2a}\right)^2\right) \\ &\propto \exp\left(a\left(\mu + \frac{b}{2a}\right)^2\right) \end{aligned}$$

By normal-normal conjugacy, we know that the posterior on μ will be normal (given μ has a normal prior). Thus, $\mu|D \sim \mathcal{N}(\mu_\mu, \sigma_\mu^2)$, so the PDF of $\mu|D$ is:

$$p(\mu|D) = \frac{1}{\sigma_\mu\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_\mu^2}(\mu-\mu_\mu)^2\right)$$

In order to find μ_μ and σ_μ, we can simply pattern match terms in the exponent of the PDF with the simplified

expression for the posterior in terms of a, b, and c derived above:

$$
\begin{aligned}
-\mu_\mu &= \frac{b}{2a} \\
-\mu_\mu &= \frac{\frac{1}{\sigma^2}\sum_{i=1}^{n} x_i}{2\left(-\frac{1}{2\tau^2} - \frac{n}{2\sigma^2}\right)} \\
\therefore \mu_\mu &= \frac{\tau^2 \sum_{i=1}^{n} x_i}{n\tau^2 + \sigma^2}
\end{aligned}
$$

$$
\begin{aligned}
-\frac{1}{2\sigma_\mu^2} &= a \\
-\frac{1}{2\sigma_\mu^2} &= -\frac{1}{2\tau^2} - \frac{n}{2\sigma^2} \\
\therefore \sigma_\mu^2 &= \frac{\sigma^2\tau^2}{n\tau^2 + \sigma^2}
\end{aligned}
$$

Thus, the distribution of $\mu|D$ is given by:

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

Solution 1.2(a):
To find μ_{MLE}, we find the value of μ that minimizes the negative log likelihood (NLL) of the data:

$$\arg\min_{\mu}\{-\ln p(D|\mu)\}$$

$$\arg\min_{\mu}\left\{K + \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2\right\}$$

where K is a constant term not containing μ.

Taking the derivative of the NLL with respect to μ and setting it equal to 0, yields the MLE for μ:

$$
\begin{aligned}
\frac{\partial[NLL]}{\partial\mu} &= \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu) = 0 \\
\implies \mu_{MLE} &= \frac{1}{n}\sum_{i=1}^{n} x_i
\end{aligned}
$$

Solution 1.2(b):
To find μ_{MAP}, we find the value of μ that maximizes the posterior probability, $p(\mu|D)$. We calculated that this posterior distribution follows a normal distribution which is maximized at its mean:

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

Solution 1.2(c):
Re-writing μ_{MAP}, we discover that:

$$\mu_{MAP} = \tau^2\left(\frac{\sum_{x_i \in D} x_i}{n + \frac{\sigma^2}{\tau^2}}\right)$$

$$\mu_{MAP} = \tau^2\mu_{x|D} \text{ where } \mu_{x|D} \text{ is the mean of } x|D$$

Solution 1.2(d):
Using the same logic as in 1.1, given $x \sim \mu + \epsilon$ where $\epsilon \sim N(0, \sigma^2)$, we deduce that:

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

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

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

Solution 1.2(e):
The variance of $x|D$ is greater than the variance of $x|\mu_{MLE}$. At the limit:

$$\lim_{n\to\infty}[VAR_{x|D}] \to VAR_{x|\mu_{MLE}}$$

This is intuitive because the variance of $x|D$ is calculated by marginalizing out over all the possible values of μ, as oppose to using a single point estimate, μ_{MLE}, to derive the distribution of $x|\mu_{MLE}$. Thus, the distribution of $x|D$ will have a higher variance than the distribution of $x|\mu_{MLE}$, and will only converge at the limit when the MLE is the mean of all n data points.

Solution 1.3(a):
When $\sum_{x_i \in D} x_i = 0$, $\mu_{MLE} = \mu_{MAP} = 0$.

Solution 1.3(b):
For this question it helps to rewrite μ_{MLE} and μ_{MAP} as:

$$\mu_{MLE} = \frac{\sum_{i=1}^{n} x_i}{n} \qquad \mu_{MAP} = \frac{\sum_{i=1}^{n} x_i}{n + \frac{\sigma^2}{\tau^2}}$$

We can see that the denominator of μ_{MLE} is less than the denominator of μ_{MAP}, given σ and τ are positive. Thus, when $\sum_{x_i \in D} x_i > 0$, $\mu_{MLE} > \mu_{MAP}$.

Solution 1.3(c):
Using the same logic as above, when $\sum_{x_i \in D} x_i < 0$, $\mu_{MLE} < \mu_{MAP}$.

Solution 1.4:

$$\begin{aligned}
\lim_{n\to\infty} \frac{\mu_{MAP}}{\mu_{MLE}} &= \lim_{n\to\infty} \frac{\left(\frac{\sum_{i=1}^{n} x_i}{n + \frac{\sigma^2}{\tau^2}}\right)}{\left(\frac{\sum_{i=1}^{n} x_i}{n}\right)} \\
&= \lim_{n\to\infty} \frac{n}{n + \frac{\sigma^2}{\tau^2}} \\
&= 1
\end{aligned}$$

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 providing the point estimate $\mathbf{w}_{\text{mean}}$ is based on the mean of the posterior $\mathbf{w}|\mathbf{X}, \mathbf{y}$. Provide an expression for the estimate $\mathbf{w}_{\text{mean}}$, up to a normalizing constant. Based on this expression, which model (original or Bayesian) would we expect to take longer to train? Further, **if** the model assumptions are correct (i.e. there indeed is a linear relationship and $\mathbf{w}$ is indeed sample from a Laplace distribution), which model would we expect to have a lower test MSE?

Solution:

Solution 2.1:
Using Bayes' Rule:

$$\begin{aligned}
p(\mathbf{w}|\mathbf{X}, \mathbf{y}) &= \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y})} \\
&\propto p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})
\end{aligned}$$

Given $\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon$ where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, we deduce that $\mathbf{y}|\mathbf{X}, \mathbf{w} \sim \mathcal{N}(\mathbf{X}\mathbf{w}, \sigma^2)$ and $y_i|\mathbf{x_i}, \mathbf{w} \sim \mathcal{N}(\mathbf{x}_i^\top \mathbf{w}, \sigma^2)$. Additionally, the Laplacian prior on $\mathbf{w}$ is given in the question:

$$\begin{aligned} p(\mathbf{w}|\mathbf{X}, \mathbf{y}) &\propto \left[\prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}\left(y_i - \mathbf{x}_i^\top \mathbf{w}\right)^2\right)\right] \frac{1}{2\tau} \exp\left(-\frac{\|\mathbf{w} - \mu\|_1}{\tau}\right) \\ &\propto \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \frac{1}{2\tau} \exp\left(-\frac{\|\mathbf{w} - \mu\|_1}{\tau} - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \mathbf{x}_i^\top \mathbf{w}\right)^2\right) \end{aligned}$$

Finally, simplifying this expression to be an equality, up to a normalizing constant called Z, we get the posterior of $\mathbf{w}$:

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) = Z \exp\left(-\frac{\|\mathbf{w} - \mu\|_1}{\tau} - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \mathbf{x}_i^\top \mathbf{w}\right)^2\right)$$

Solution 2.2:
To find $\mathbf{w}_{MAP}$, we find the value of $\mathbf{w}$ that maximizes the posterior probability, $p(\mathbf{w}|D)$; in other words, the value of $\mathbf{w}$ that minimizes the negative log-likelihood, $-\ln p(\mathbf{w}|D)$. (Note that we can exclude constant terms not including $\mathbf{w}$ in the arg min expression):

$$\begin{aligned} \mathbf{w}_{MAP} &= \arg\min_{\mathbf{w}} \{-\ln p(\mathbf{w}|D)\} \\ &= \arg\min_{\mathbf{w}} \left\{\frac{\|\mathbf{w} - \mu\|_1}{\tau} + \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \mathbf{x}_i^\top \mathbf{w}\right)^2\right\} \end{aligned}$$

Finally, taking out a factor of $\frac{1}{2\sigma^2}$, and noting that $\arg\min_x kf(x) = \arg\min_x f(x)$, we get:

$$\mathbf{w}_{MAP} = \arg\min_{\mathbf{w}} \left\{\frac{2\sigma^2}{\tau}\|\mathbf{w} - \mu\|_1 + \sum_{i=1}^{n}\left(y_i - \mathbf{x}_i^\top \mathbf{w}\right)^2\right\}$$

The interpretation of this Bayesian approach is that adding a Laplacian prior over the distribution of our weight parameters and then maximizing the resulting posterior distribution is equivalent to the frequentist approach of adding an L1 requlariza tion term to least squares regression with $\lambda = \frac{2\sigma^2}{\tau}$. A larger value of τ suggests we are less confident in the prior on $\mathbf{w}$ (Bayesian approach), which results in a smaller value for λ and less regularization (frequentist approach).

Solution 2.3:
Given some prior knowledge that $\mathbf{w} \approx \mathbf{v}$, we can set the location parameter, μ, in the prior on $\mathbf{w}$ equal to $\mathbf{v}$. This makes sense from a Bayesian perspective, since this incorporates prior knowledge into our prior distribution on $\mathbf{w}$, and the posterior is calculated using this information. Likewise, from a frequentist perspective, this makes sense because when we set $\mu = \mathbf{v}$, we apply more regularization when the components of $\mathbf{w}$ are far from the components of $\mathbf{v}$, which is what we want to be penalizing since we know the components of $\mathbf{w}$ should be close to the components of $\mathbf{v}$.

Solution 2.4:
τ decreasing is equivalent to increasing the strength of regularization. Thus, as τ decreases, the entries of $\mathbf{w}_{MAP}$ should get closer to the entries of μ. As $\tau \to 0$, the entries of $\mathbf{w}_{MAP}$ should end up being equal to μ.

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:

Solution 3.1:

Parameter/Variable	Dimensions
$\mathbf{x}$	N x 1
$\mathbf{W}_1$	H x N
$\mathbf{b}_1$	H x 1
$\mathbf{a}_1$	H x 1
$\mathbf{z}_1$	H x 1
$\mathbf{W}_2$	1 x H
$\mathbf{b}_2$	1 x 1
a_2	1 x 1
$\hat{y} = z_2$	1 x 1

Solution 3.2(a):
Applying the chain rule:

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

Solution 3.2(b):
Applying the chain rule:

$$\frac{\partial L}{\partial W_2^h} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot \frac{\partial a_2}{\partial W_2^h}$$

Using $\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} = \hat{y} - y$ from part (a):

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

Solution 3.2(c):
Applying the chain rule:

$$\frac{\partial L}{\partial b_1^h} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_1^h} \cdot \frac{\partial z_1^h}{\partial a_1^h} \cdot \frac{\partial a_1^h}{\partial b_1^h}$$

Using $\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} = \hat{y} - y$ from part (a):

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

Solution 3.2(d):
Applying the chain rule:

$$\frac{\partial L}{\partial W_1^{h,m}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_1^h} \cdot \frac{\partial z_1^h}{\partial a_1^h} \cdot \frac{\partial a_1^h}{\partial W_1^{h,m}}$$

Using $\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_1^h} \cdot \frac{\partial z_1^h}{\partial a_1^h} = (\hat{y} - y) W_2^h \sigma(a_1^h)(1 - \sigma(a_1^h))$ from part (c):

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

Problem 4 (Modern Deep Learning Tools: PyTorch)

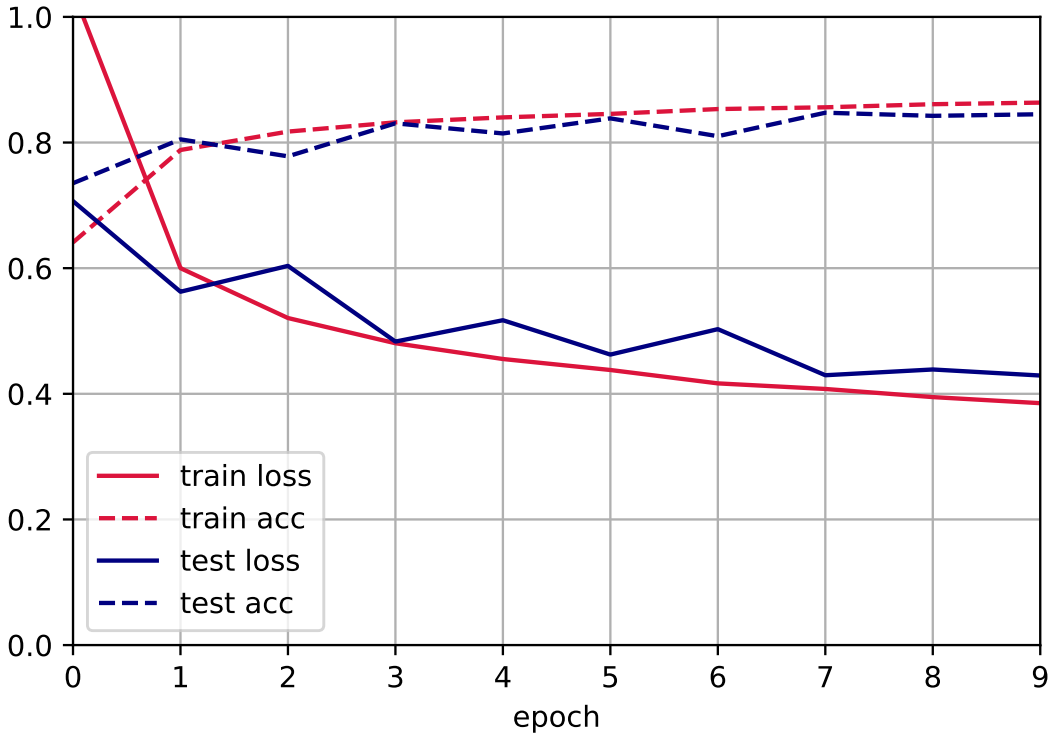
In this problem, you will learn how to use PyTorch. This machine learning library is massively popular and used heavily throughout industry and research. In `T3_P3.ipynb` you will implement an MLP for image classification from scratch. Copy and paste code solutions below and include a final graph of 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 = 28 * 28
n_hiddens = 256
n_outputs = 10

W1 = torch.nn.Parameter(0.01 * torch.randn(size=(n_inputs, n_hiddens)))
b1 = torch.nn.Parameter(torch.zeros(n_hiddens))
W2 = torch.nn.Parameter(0.01 * torch.randn(size=(n_hiddens, n_outputs)))
b2 = torch.nn.Parameter(torch.zeros(n_outputs))



def relu(x):
    return torch.clamp(x, 0, None)
```

```
def softmax(X):
    numerator = torch.exp(X)
    denominator = torch.sum(numerator, axis = 1, keepdim=True)
    return torch.div(numerator, denominator)



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



def cross_entropy(y_hat, y):
    loss = 0
    loss -= torch.log(y_hat[range(len(y_hat)), y])
    return loss



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

Ben Ray

Collaborators and Resources

Whom did you work with, and did you use any resources beyond cs181-textbook and your notes?
Worked with Ty Geri, Elias Nuwara, and Angela Li
CS 181 Office hours

Calibration

Approximately how long did this homework take you to complete (in hours)?
15