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. By Bayes' rule,

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

   We will drop various terms throughout the derivation that are constant with respect to μ, as the final distribution for $\mu|D$ will be renormalized to integrate to unity. Using the definition of the Gaussian

PDF, the likelihood of the data is

$$p(D|\mu) = \prod_{i=1}^{n} \mathcal{N}(x_i|\mu, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\frac{(x_i-\mu)^2}{\sigma^2}\right\} \propto \exp\left\{-\frac{1}{2}\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{\sigma^2}\right\}$$

and the prior on μ is

$$p(\mu) = \mathcal{N}(\mu|0, \tau^2) = \frac{1}{\tau\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\frac{(\mu-0)^2}{\tau^2}\right\} \propto \exp\left\{-\frac{1}{2}\frac{\mu^2}{\tau^2}\right\}.$$

Therefore,

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

The product of two Gaussian PDFs is Gaussian, so we want to re-arrange the above expression to be proportional to a new Gaussian PDF with parameters μ_n and σ_n,

$$\mathcal{N}(\mu|\mu_n, \sigma_n) \propto \exp\left\{-\frac{1}{2}\frac{(\mu-\mu_n)^2}{\sigma_n^2}\right\} = \exp\left\{-\frac{1}{2}\left(\mu^2\frac{1}{\sigma_n^2} - 2\mu\frac{\mu_n}{\sigma_n^2} + \frac{\mu_n^2}{\sigma_n^2}\right)\right\}.$$

Expanding the squared term in our expression for $p(\mu|D)$ and rearranging,

$$\begin{aligned} p(\mu|D) &\propto \exp\left\{-\frac{1}{2}\left(\frac{\sum_{i=1}^{n}x_i^2}{\sigma^2} - 2\mu\frac{\sum_{i=1}^{n}x_i}{\sigma^2} + \frac{n\mu^2}{\sigma^2} + \frac{\mu^2}{\tau^2}\right)\right\} \\ &= \exp\left\{-\frac{1}{2}\left(\mu^2\left(\frac{1}{\tau^2} + \frac{n}{\sigma^2}\right) - 2\mu\frac{\sum_{i=1}^{n}x_i}{\sigma^2} + \frac{\sum_{i=1}^{n}x_i^2}{\sigma^2}\right)\right\}. \end{aligned}$$

Matching up terms in the expression above with the Gaussian PDF expression we want,

$$\mu^2\left(\frac{1}{\tau^2} + \frac{n}{\sigma^2}\right) = \mu^2\frac{1}{\sigma_n^2} \Rightarrow \sigma_n^2 = \left(\frac{1}{\tau^2} + \frac{n}{\sigma^2}\right)^{-1}$$

and

$$-2\mu\frac{\sum_{i=1}^{n}x_i}{\sigma^2} = -2\sigma\frac{\mu_n}{\sigma_n^2} \Rightarrow \mu_n = \left(\frac{1}{\tau^2} + \frac{n}{\sigma^2}\right)^{-1}\frac{\sum_{i=1}^{n}x_i}{\sigma^2}.$$

Therefore,

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

2. (a) To find $\mu_{MLE} \in \arg\max_\mu p(D|\mu)$, take the logarithm of the likelihood, differentiate, and set equal

to zero:

$$\begin{aligned}
p(D|\mu) &= \prod_{i=1}^{n} p(x_i|\mu) = \prod_{i=1}^{n} \mathcal{N}(x_i|\mu, \sigma^2) \\
&= \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\frac{(x_i-\mu)^2}{\sigma^2}\right\} \\
&= \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{\sigma^2}\right\} \\
\log p(D|\mu) &= \frac{1}{\sigma\sqrt{2\pi}} - \frac{1}{2}\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{\sigma^2} \\
\frac{\partial \log p(D|\mu)}{\partial \mu} &= \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu) \\
0 &= \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu_{MLE}) \\
\mu_{MLE} &= \frac{\sum_{i=1}^{n} x_i}{n},
\end{aligned}$$

so μ_{MLE} is the empirical mean.

(b) To find $\mu_{MAP} \in \arg\max_{\mu} p(\mu|D)$, use the PDF of the Gaussian distribution derived in part (1), take the logarithm, differentiate, and set equal to zero:

$$\begin{aligned}
p(\mu|D) = \mathcal{N}(\mu|\mu_n, \sigma_n^2) &= \frac{1}{\sigma_n\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\frac{(\mu-\mu_n)^2}{\sigma_n^2}\right\} \\
\log p(\mu|D) &= \log\left(\frac{1}{\sigma_n\sqrt{2\pi}}\right) - \frac{1}{2}\frac{(\mu-\mu_n)^2}{\sigma_n^2} \\
\frac{\partial \log p(\mu|D)}{\partial \mu} &= -\frac{\mu-\mu_n}{\sigma_n^2} \\
0 &= -\frac{\mu_{MAP}-\mu_n}{\sigma_n^2} \\
\mu_{MAP} = \mu_n &= \left(\frac{1}{\tau^2}+\frac{n}{\sigma^2}\right)^{-1}\frac{\sum_{i=1}^{n} x_i}{\sigma^2}.
\end{aligned}$$

In this case, we could also have seen that $\mu_{MAP} = \mu_n$ by recognizing that the most likely value of $\mu|D$ is the mean of its distribution.

(c) We have

$$\mu_{MAP} = \left(\frac{1}{\tau^2}+\frac{n}{\sigma^2}\right)^{-1}\frac{\sum_{i=1}^{n} x_i}{\sigma^2} = \frac{\sum_{i=1}^{n} x_i}{\left(\frac{1}{\tau^2}+\frac{n}{\sigma^2}\right)\sigma^2} = \frac{\sum_{x_i\in D} x_i}{n+\frac{\sigma^2}{\tau^2}},$$

so μ_{MAP} is equal to the mean of $x|D$.

(d) For a fixed value of $\mu = \mu^*$, the distribution of $x|\mu^*$ is $\mathcal{N}(\mu^*, \sigma^2)$, since $x \sim \mu^* + \epsilon \sim \mathcal{N}(\mu^* + 0, 0+\sigma^2)$. Thus,

$$\begin{aligned}
x|\mu_{MLE} &\sim \mathcal{N}\left(\frac{\sum_{x_i\in D} x_i}{n}, \sigma^2\right) \\
x|\mu_{MAP} &\sim \mathcal{N}\left(\frac{\sum_{x_i\in D} x_i}{n+\frac{\sigma^2}{\tau^2}}, \sigma^2\right).
\end{aligned}$$

(e) The variance of $x|D$ is $\left(\frac{n}{\sigma^2}+\frac{1}{\tau^2}\right)^{-1}+\sigma^2$, which is greater than σ^2, the variance of $x|\mu_{MLE}$, because

$$\left(\frac{n}{\sigma^2}+\frac{1}{\tau^2}\right)^{-1} > 0,$$

for all $n > 0$, assuming $\sigma^2, \tau^2 > 0$. The limit of the variance of $x|D$ as n tends to infinity is

$$\lim_{n\to\infty}\left(\left(\frac{n}{\sigma^2}+\frac{1}{\tau^2}\right)^{-1}+\sigma^2\right) = \lim_{n\to\infty}\left(\left(\frac{n}{\sigma^2}+\frac{1}{\tau^2}\right)^{-1}\right)+\sigma^2 = 0+\sigma^2 = \sigma^2.$$

This is intuitive because as the size of the data set increases, we become more certain about what the mean might be, so the variance in $x|D$ due to uncertainty about the mean decreases. This leaves only the predicted variance due to real variance in the data, which converges to σ^2.

3. (a) Assuming $\sum_{x_i\in D} x_i = 0$,

$$\mu_{MLE} = \frac{\sum_{x_i\in D} x_i}{n} = 0$$

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

(b) The denominator in μ_{MLE} is smaller than that in μ_{MAP}, so assuming $\sum_{x_i\in D} x_i > 0$,

$$\mu_{MLE} > \mu_{MAP}.$$

(c) Again, the denominator in μ_{MLE} is smaller than that in μ_{MAP}, so assuming $\sum_{x_i\in D} x_i < 0$,

$$\mu_{MLE} < \mu_{MAP}.$$

4. Computing,

$$\lim_{n\to\infty}\frac{\mu_{MAP}}{\mu_{MLE}} = \lim_{n\to\infty}\frac{\left(\frac{\sum_{x_i\in D} x_i}{n+\frac{\sigma^2}{\tau^2}}\right)}{\left(\frac{\sum_{x_i\in D} x_i}{n}\right)} = \lim_{n\to\infty}\frac{n}{n+\frac{\sigma^2}{\tau^2}} = 1.$$

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

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

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.[a]

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

Solution:

1. By Bayes' rule,

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w}).$$

The likelihood expands to

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

and the prior is

$$p(\mathbf{w}) = \frac{1}{2\tau}\exp\left\{-\frac{\|\mathbf{w}-\mu\|_1}{\tau}\right\} \propto \exp\left\{-\frac{\|\mathbf{w}-\mu\|_1}{\tau}\right\},$$

so the posterior distribution has PDF

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

2. The MAP estimate can be obtained by solving

$$\begin{aligned}
\mathbf{w}_{\text{MAP}} &\in \arg\max_{\mathbf{w}} p(\mathbf{w}|\mathbf{X},\mathbf{y}) \\
&\in \arg\max_{\mathbf{w}} \log p(\mathbf{w}|\mathbf{X},\mathbf{y}) \\
&\in \arg\min_{\mathbf{w}} (-\log p(\mathbf{w}|\mathbf{X},\mathbf{y})) \\
&\in \arg\min_{\mathbf{w}} \left(\frac{1}{2}\frac{\sum_{i=1}^{N}(y_i-\mathbf{w}^T\mathbf{x}_i)^2}{\sigma^2} + \frac{1}{\tau}\|\mathbf{w}-\mu\|_1\right) \\
&\in \arg\min_{\mathbf{w}} \left(\frac{1}{2}\sum_{i=1}^{N}(y_i-\mathbf{w}^T\mathbf{x}_i)^2 + \frac{1}{\tau}\|\mathbf{w}-\mu\|_1\right).
\end{aligned}$$

From this equation, we can see that solving for $\mathbf{w}_{\text{MAP}}$ corresponds to optimizing $\mathbf{w}$ using least squares loss and lasso regression if $\mu = 0$ and where $\lambda = \frac{1}{\tau}$. The scale factor τ is the reciprocal of the corresponding regularization parameter λ. Intuition for this connection is that imposing a prior on $\mathbf{w}$ corresponds to restricting weights in the frequentist approach such that they are not too far from some pre-determined value, in particular 0 when $\mu = 0$, or $\mathbf{v}$ when $\mu = \mathbf{v}$.

3. We could incorporate such prior expert knowledge by letting $\mu = \mathbf{v}$ and setting $\lambda = 1/\tau$ to a moderately high value so that in the MAP estimate, the regularization term has non-negligible weight relative to the least squares loss term. This makes sense from the Bayesian viewpoint because setting the mean of our prior to $\mathbf{v}$ and letting τ be relatively small reflects that we think $\mathbf{w}$ is probably close to $\mathbf{v}$. From the frequentist viewpoint, including a location parameter μ inside the lasso regression term and letting λ be relatively large enforces that the model will converge to a solution whose weights are relatively close to those in $\mathbf{v}$.

4. As τ decreases, the entries of the estimate $\mathbf{w}_{\text{MAP}}$ become more similar to those in the location parameter μ, since the lasso regularization term starts to dominate the least squares error term in the minimization. As $\tau \to 0$, $\frac{1}{\tau} \to \infty$, and the only way to minimize the regularization term is to let $\mathbf{w} = \mu$, so

$$\mathbf{w} \to \mu.$$

5. The expected MSE for an estimator $\hat{\mathbf{w}}$ on a test set of size T is

$$\sum_{i=1}^{T} E_{\mathbf{x},y}[(y_i - \hat{y}_i)^2] = T E_{\mathbf{x},y}[(y - \hat{y})^2].$$

We will drop the T and subsequent terms that are constant with respect to $\hat{\mathbf{w}}$, since we are only interested in comparing estimators. Expanding, and adding and subtracting the same term as in the bias/variance derivation,

$$\begin{aligned}
E_{\mathbf{x},y}[(y - \hat{y})^2] &= E_{\mathbf{x},y}[(y - \hat{\mathbf{w}}^T\mathbf{x})^2] \\
&= E_{\mathbf{x},y}[(y - \mathbf{w}^T\mathbf{x} + \mathbf{w}^T\mathbf{x} - \hat{\mathbf{w}}^T\mathbf{x})^2] \\
&= E_{\mathbf{x},y}[(y - \mathbf{w}^T\mathbf{x})^2] + E_{\mathbf{x},y}[(\mathbf{w}^T\mathbf{x} - \hat{\mathbf{w}}^T\mathbf{x})^2] + 2E_{\mathbf{x},y}[(y - \mathbf{w}^T\mathbf{x})(\mathbf{w}^T\mathbf{x} - \hat{\mathbf{w}}^T\mathbf{x})].
\end{aligned}$$

The first term does not contain $\hat{\mathbf{w}}$, so we discard it. In the third term, $(y - \mathbf{w}^T\mathbf{x})$ is independent of $(\mathbf{w}^T\mathbf{x} - \hat{\mathbf{w}}^T\mathbf{x})$ and goes to zero since the expected deviation of y from its mean is zero. We are left with, letting $\mathbf{v} = \mathbf{w} - \hat{\mathbf{w}}$,

$$\begin{aligned}
E_{\mathbf{x},y}[(\mathbf{w}^T\mathbf{x} - \hat{\mathbf{w}}^T\mathbf{x})^2] &= E_{\mathbf{x},y}[(\mathbf{v}^T\mathbf{x})^2] \\
&= E_{\mathbf{x},y}\left[\sum_{i=1}^{p} v_i^2 x_i^2 + 2 \sum_{1 \le i < j \le p}^{p} v_i v_j x_i x_j\right].
\end{aligned}$$

Applying linearity, the independence of $\mathbf{v}$ and $\mathbf{x}$, and the independence of distinct features, this becomes

$$E_{\mathbf{x},y}[\mathbf{x}^T\mathbf{x}]E_{\mathbf{x},y}[\mathbf{v}^T\mathbf{v}] + 2 \sum_{1 \le i < j \le p}^{p} E_{\mathbf{x},y}[v_i v_j] E_{\mathbf{x},y}[x_i] E_{\mathbf{x},y}[x_j].$$

Recalling that $\mathbf{x}$ has mean zero and variance one, we have simply

$$E_{\mathbf{x},y}[\mathbf{v}^T\mathbf{v}] = E_{\mathbf{x},y}[(\mathbf{w} - \hat{\mathbf{w}})^T(\mathbf{w} - \hat{\mathbf{w}})].$$

This quantity is minimized when $\hat{\mathbf{w}} = \mathbf{w}_{\text{mean}}$, since by assumption $\mathbf{w}$ is sampled from $\mathbf{w}|\mathbf{X}, \mathbf{y}$. Therefore the model with $\mathbf{w}_{\text{mean}}$ has a lower expected test MSE than the model with $\mathbf{w}_{\text{MAP}}$.

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:

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

   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:

1. The dimensionalities are:

$$\mathbf{W}_1 : H \times M, \qquad \mathbf{b}_1 : H \times 1$$
$$\mathbf{W}_2 : 1 \times H, \qquad \mathbf{b}_2 : 1 \times 1$$
$$\mathbf{a}_1 : H \times 1, \qquad \mathbf{z}_1 : H \times 1$$
$$a_2 : 1 \times 1, \qquad \hat{y} = z_2 : 1 \times 1$$

Given that $\mathbf{x} : M \times 1$, substituting these dimensions into the mathematical operations described above gives final dimension

$$\begin{aligned} \hat{y} &: (1 \times H)\left[(H \times M)(M \times 1) + (H \times 1)\right] + (1 \times 1) \\ &: (1 \times H)(H \times 1) + (1 \times 1) \\ &: 1 \times 1, \end{aligned}$$

which is correct since we are given $\hat{y}$ is a scalar.

2. (a) By the chain rule,

$$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial a_2}\frac{\partial a_2}{\partial b_2}.$$

For the second term, $\frac{\partial a_2}{\partial b_2} = 1$. For the first term,

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

so

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

(b) By the chain rule, using the expression for $\frac{\partial L}{\partial a_2}$ found in the previous part,

$$\begin{aligned} \frac{\partial L}{\partial W_2^h} &= \frac{\partial L}{\partial a_2}\frac{\partial a_2}{\partial W_2^h} \\ &= (\hat{y} - y)\frac{\partial \sum_{h=1}^{H} W_2^h z_1^h + b_2}{\partial W_2^h} \\ &= (\hat{y} - y) z_1^h, \end{aligned}$$

where z_1^h represents the hth element of $\mathbf{z}_1$.

(c) By the chain rule, recognizing that that only the hth element of $\mathbf{a}_1$ depends on b_1^h,

$$\begin{aligned} \frac{\partial L}{\partial b_1^h} &= \frac{\partial L}{\partial a_1^h}\frac{\partial a_1^h}{\partial b_1^h} \\ &= \frac{\partial L}{\partial a_2}\frac{\partial a_2}{\partial a_1^h}\frac{\partial a_1^h}{\partial b_1^h}, \end{aligned}$$

where a_1^h represents the hth element of $\mathbf{a}_1$. We already know $\frac{\partial L}{\partial a_2} = \hat{y} - y$, and we find $\frac{\partial a_1^h}{\partial b_1^h} = 1$. To find $\frac{\partial a_2}{\partial a_1^h}$, expand and use the chain rule:

$$\begin{aligned}
\frac{\partial a_2}{\partial a_1^h} &= \frac{\partial \mathbf{W}_2 \sigma(\mathbf{a}_1) + \mathbf{b}_2}{\partial a_1^h} \\
&= W_2^h \frac{\partial \sigma(a_1^h)}{\partial a_1^h} \\
&= W_2^h \sigma(a_1^h)(1 - \sigma(a_1^h)) \\
&= W_2^h z_1^h (1 - z_1^h),
\end{aligned}$$

where z_1^h represents the hth element of $\mathbf{z}_1$. Putting it all together,

$$\frac{\partial L}{\partial b_1^h} = (\hat{y} - y) W_2^h z_1^h (1 - z_1^h).$$

(d) Noting that $W_1^{h,m}$ only influences the hth element of $\mathbf{a}_1$,

$$\frac{\partial L}{\partial W_1^{h,m}} = \frac{\partial L}{\partial a_1^h} \frac{\partial a_1^h}{\partial W_1^{h,m}}$$

For the second term,

$$\frac{\partial a_1^h}{\partial W_1^{h,m}} = \frac{\partial \mathbf{W}_1^h x + b_1^h}{\partial W_1^{h,m}} = \frac{\partial W_1^{h,m} x^m}{\partial W_1^{h,m}} = x^m,$$

where x^m represents the mth element in $\mathbf{x}$. Using the expression for $\frac{\partial L}{\partial a_1^h}$ from part (c),

$$\frac{\partial L}{\partial W_1^{h,m}} = (\hat{y} - y) W_2^h z_1^h (1 - z_1^h) x^m.$$

Problem 4 (Modern Deep Learning Tools: PyTorch)

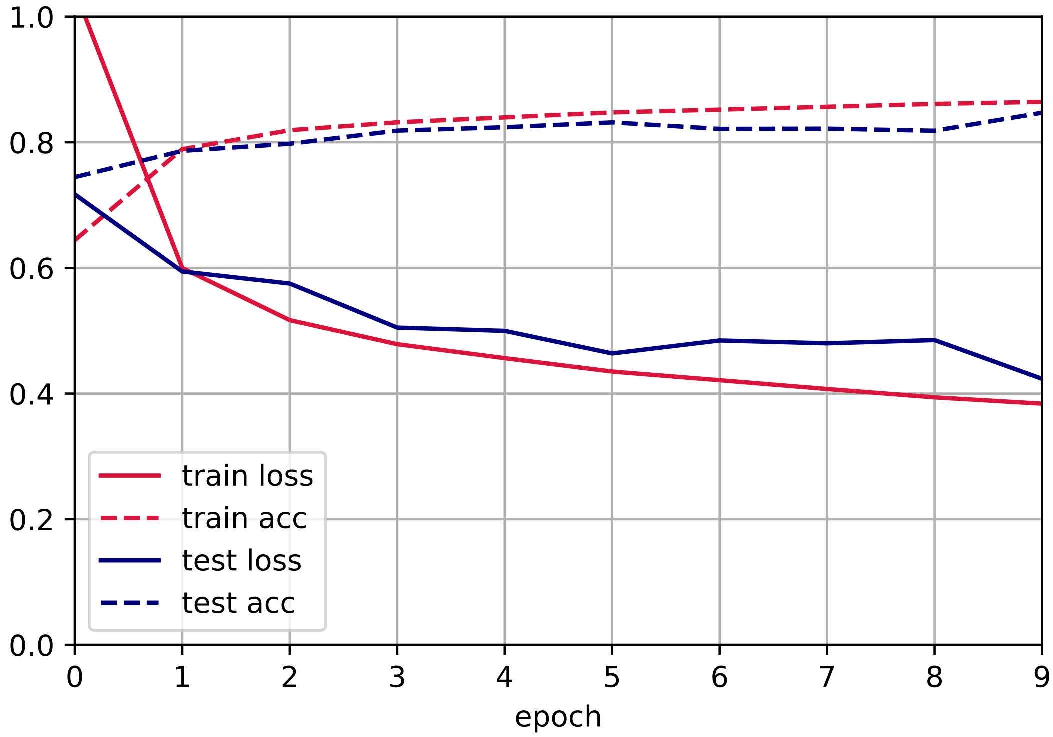
In this problem, you will learn how to use PyTorch. This machine learning library is massively popular and used heavily throughout industry and research. In `T3_P3.ipynb` you will implement an MLP for image classification from scratch. Copy and paste code solutions below and include a final graph of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



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(size=(1, n_hiddens)))
W2 = torch.nn.Parameter(0.01 * torch.randn(size=(n_hiddens, n_outputs)))
b2 = torch.nn.Parameter(torch.zeros(size=(1, n_outputs)))



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

```python
def softmax(x):
  exped = X.exp()
  return exped / exped.sum(1).reshape(-1, 1)



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):
  y_hat_sub_y = y_hat.gather(1, y.unsqueeze(1)).squeeze()
  return -torch.log(y_hat_sub_y)



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



def train(net, params, train_iter, loss_func=cross_entropy, updater=sgd):
  epochs = 10
  for _ in range(epochs):
    for X, y in train_iter:
      y_hat = net(X)
      losses = loss_func(y_hat, y)
      loss = losses.mean()
      loss.backward()
      updater(params)
```

Name

Collaborators and Resources

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

No one. PyTorch documentation.

Calibration

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

10