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, $p(\mu|D) \propto p(D|\mu)p(\mu)$. By independence and exponent rules,

$$p(D|\mu) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n exp\Big\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2 \Big\}.$$

   Multiplying by $p(\mu)$ yields

$$\frac{1}{\tau\sqrt{2\pi}} \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n exp\Big\{ -\frac{\mu^2}{2\tau^2} - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2 \Big\}.$$

Expanding yields

$$\frac{1}{\tau\sqrt{2\pi}}\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n exp\Big\{-\frac{\mu^2}{2\tau^2}-\frac{1}{2\sigma^2}\sum_{i=1}^n x_i^2+\frac{1}{2\sigma^2}2\mu\sum_{i=1}^n x_i-\frac{1}{2\sigma^2}n\mu^2\Big\},$$

and regrouping yields

$$\frac{1}{\tau\sqrt{2\pi}}\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n exp\Big\{-\frac{1}{2\sigma^2}\sum_{i=1}^n x_i^2+\Big(\frac{1}{2\sigma^2}\sum_{i=1}^n x_i\Big)2\mu-\frac{1}{2}\Big(\frac{1}{\tau^2}+\frac{n}{\sigma^2}\Big)\mu^2\Big\},$$

Now we'll integrate in order to normalize this value (i.e. in order to divide by $p(D)$).

Because we're integrating with respect to μ, we can pull out many of the terms at will, and even with the messy terms, our form still resembles a normal pdf since there's a μ^2 and μ term in the exponent. We'll guess that the exponent can be written as $exp\{-\frac{(\mu-\mu_{new})^2}{2\gamma^2}\}$ for some value μ_{new} in terms of $\sum_{i=1}^n x_i$, σ, n, and τ and some value $\gamma > 0$ in terms of σ, n, and τ. Our given expansion suggests that $\frac{1}{\gamma^2}$ should be $\frac{1}{\tau^2}+\frac{n}{\sigma^2}$ i.e.

$$\gamma^2 = (\frac{1}{\tau^2}+\frac{n}{\sigma^2})^{-1}.$$

The coefficient on 2μ is $\Big(\frac{1}{2\sigma^2}\sum_{i=1}^n x_i\Big)$, and we would like it to be $\frac{\mu_{new}}{2\gamma^2}$, so μ_{new} must be

$$2\gamma^2\cdot\frac{1}{2\sigma^2}\sum_{i=1}^n x_i = 2\frac{1}{\frac{1}{\tau^2}+\frac{n}{\sigma^2}}\frac{1}{2\sigma^2}\sum_{i=1}^n x_i = \frac{1}{\frac{\sigma^2}{\tau^2}+n}\sum_{i=1}^n x_i.$$

To complete the square, we will multiply by our expression by

$$exp\{\frac{-\mu_{new}^2}{2\gamma^2}+\frac{\mu_{new}^2}{2\gamma^2}\} = 1.$$

Shifting terms in our earlier expression yields

$$\frac{1}{\tau\sqrt{2\pi}}\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n exp\Big\{-\frac{1}{2\sigma^2}\sum_{i=1}^n x_i^2\Big\}\int exp\Big\{\Big(\frac{1}{2\sigma^2}\sum_{i=1}^n x_i\Big)2\mu-\frac{1}{2}\Big(\frac{1}{\tau^2}+\frac{n}{\sigma^2}\Big)\mu^2\Big\}du.$$

And multiplying by 1 to complete the square yields

$$\frac{1}{\tau\sqrt{2\pi}}\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n exp\Big\{-\frac{1}{2\sigma^2}\sum_{i=1}^n x_i^2\Big\}exp\Big\{\frac{\mu_{new}^2}{2\gamma^2}\Big\}\int exp\Big\{\frac{-(\mu-\mu_{new})^2}{2\gamma^2}\Big\}du.$$

We multiply by $\frac{\gamma\sqrt{2\pi}}{\gamma\sqrt{2\pi}} = 1$ again to get a normal pdf:

$$\frac{\gamma\sqrt{2\pi}}{\tau\sqrt{2\pi}}\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n exp\Big\{-\frac{1}{2\sigma^2}\sum_{i=1}^n x_i^2\Big\}exp\Big\{\frac{\mu_{new}^2}{2\gamma^2}\Big\}\int\frac{1}{\gamma\sqrt{2\pi}}exp\Big\{\frac{-(\mu-\mu_{new})^2}{2\gamma^2}\Big\}du.$$

The integrand is now a normal distribution pdf, and the integral will evaluate to 1. Thus we must divide our earlier expression by

$$\frac{\gamma\sqrt{2\pi}}{\tau\sqrt{2\pi}}\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n exp\Big\{-\frac{1}{2\sigma^2}\sum_{i=1}^n x_i^2\Big\}exp\Big\{\frac{\mu_{new}^2}{2\gamma^2}\Big\}.$$

We can factor our earlier expression like above as

$$\frac{\gamma\sqrt{2\pi}}{\tau\sqrt{2\pi}}\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}x_i^2\right\}exp\left\{\frac{\mu_{new}^2}{2\gamma^2}\right\}\frac{1}{\gamma\sqrt{2\pi}}exp\left\{\frac{-(\mu-\mu_{new})^2}{2\gamma^2}\right\},$$

so dividing yields

$$\frac{\frac{1}{\gamma\sqrt{2\pi}}exp\left\{\frac{-(\mu-\mu_{new})^2}{2\gamma^2}\right\}}{\int\frac{1}{\gamma\sqrt{2\pi}}exp\left\{\frac{-(\mu-\mu_{new})^2}{2\gamma^2}\right\}} = \frac{1}{\gamma\sqrt{2\pi}}exp\left\{\frac{-(\mu-\mu_{new})^2}{2\gamma^2}\right\}$$

since the integral evaluates to 1. Thus we've arrived at a normal pdf, and $\mu|D \sim \mathcal{N}(\mu_{new}, \gamma^2)$. Plugging in our values for μ_{new} and γ yields

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

2. (a) As above,

$$p(D|\mu) = \prod_{i=1}^{n}\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}.$$

Taking the log yields

$$l(\mu; D) = -n\log(\sigma\sqrt{2\pi}) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2.$$

Taking the derivative via the chain rule yields

$$\frac{\partial l}{\partial \mu} = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}-2(x_i-\mu).$$

Setting equal to 0 and manipulating yields

$$\sum_{i=1}^{n}x_i = n\mu.$$

Taking the second derivative yields

$$\frac{\partial^2 l}{\partial \mu^2} = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}2 = \frac{-n}{\sigma^2} < 0,$$

so this value of μ is a maximum. Thus

$$\mu_{MLE} = \frac{1}{n}\sum_{i=1}^{n}x_i.$$

(b) To find the MAP, we will consider the posterior distribution $\mu|D$ from part 1. Since it is normal, the value with the highest probability is simply the mean. Thus

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

(c) The values are equal.

(d) $x = \mu + \epsilon$, so when μ is fixed, x only varies with ϵ. Adding μ^* to ϵ simply shifts the mean of the normal distribution, so $x|\mu^* \sim \mathcal{N}(\mu^*, \sigma^2)$. Thus

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

and

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

(e) The variance of $x|D$ is greater than the variance of $x|\mu_{MLE}$ since it has an extra $(\frac{n}{\sigma^2} + \frac{1}{\tau^2})^{-1}$ term. As n goes to infinity, the $(\frac{n}{\sigma^2} + \frac{1}{\tau^2})^{-1}$ term will go to 0, and the variance of $x|D$ will go to σ^2. Intuitively as the number of data points goes up, the noise becomes less and less of an issue, and conditioning on D gets closer and closer to conditioning on a single value μ^*. Thus as n goes to infinity, the variance for $x|D$ comes mostly from the variance of the noise (σ^2) just like the variance of $x|\mu_{MLE}$.

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

$$\mu_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i = 0 = \frac{1}{\frac{\sigma^2}{\tau^2} + n}\sum_{i=1}^{n} x_i = \mu_{MAP},$$

and both are equal to 0.

(b) If $\sum_{x_i \in D} x_i > 0$, then μ_{MLE} is greater than μ_{MAP} since $\frac{1}{n} > \frac{1}{\frac{\sigma^2}{\tau^2}+n}$, and both multipliers are positive.

(c) On the other hand, if $\sum_{x_i \in D} x_i < 0$, then μ_{MLE} is less than μ_{MAP} since $\frac{1}{n} > \frac{1}{\frac{\sigma^2}{\tau^2}+n}$, and both multipliers are positive.

4.

$$\lim_{n\to\infty} \frac{\mu_{MAP}}{\mu_{MLE}} = \lim_{n\to\infty} \frac{\frac{1}{n}\sum_{i=1}^{n} x_i}{\frac{1}{\frac{\sigma^2}{\tau^2}+n}\sum_{i=1}^{n} x_i} = \lim_{n\to\infty} \frac{\frac{\sigma^2}{\tau^2} + n}{n} = 1.$$

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 penalties 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 $\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon$. 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 with extra conditioning,

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

Since $\mathbf{w}$ does not depend on $\mathbf{X}$, $p(\mathbf{w}|\mathbf{X}) = p(\mathbf{w})$. Plugging in the PDFs yields

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

2. Taking the log is a monotonic transformation, so maximizing the above quantity is equivalent to maximizing the log of the quantity. Thus we take the log to get

$$-\log(2\tau) - \frac{\|\mathbf{w} - \mu\|_1}{\tau} - nlog(\sigma\sqrt{2\pi}) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i)^2.$$

Since we're maximizing over $\mathbf{w}$, we can drop the terms that don't involve $\mathbf{w}$. Additionally, we can multiply by $2\sigma^2 > 0$ since this is also a monotonic transformation. Thus we have

$$\mathbf{w}_{MAP} = \arg\max_{\mathbf{w}} -\frac{2\sigma^2}{\tau}\|\mathbf{w} - \mu\|_1 - \sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i)^2.$$

Finally, note that maximizing this quantity is equivalent to minimizing its negation. This fact yields

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

This form is equivalent to least squares loss with lasso regularization (when $\mu = \mathbf{0}$ i.e. the origin). By examining the form, we can see that $\lambda = \frac{2\sigma^2}{\tau}$, so increasing τ decreases λ and vice versa. Intuitively, the Laplacian prior says that values near the location parameter μ are more likely than values farther away from μ and that this distance should be measured with the L_1 norm. Similarly lasso regularization says that weights near the origin are preferred with the distance again being measured with the L_1 norm.

3. We could incorporate expert knowledge by setting $\mu = \mathbf{v}$ since the Laplacian prior says that values near the location parameter are more likely. In the Bayesian model, setting this parameter represents our belief about the distribution being centered at μ. This also makes sense in the frequentist view since we can assume that the expert knowledge is based on some previously collected data and so setting this parameter reflects the previously collected data.

4. As τ decreases, the coefficient of the L_1 norm term increases, and the estimate $\mathbf{W}_{MAP}$ gets closer and closer to the location parameter μ. This is because the model cares more and more about how close $\mathbf{w}$ is to μ as this coefficient increases. Thus as $\tau \to 0$, $\mathbf{w}_{MAP} \to \mu$.

5. Note that every time I use $\mathbf{w}$ below, I am referring to $\mathbf{w}|\mathbf{X}, \mathbf{y}$ i.e. $\mathbf{w}$ drawn from the posterior distribution. Let $\widehat{\mathbf{w}}$ denote an estimate of $\mathbf{w}$. Then by definition, the expected test MSE is

$$E[(\widehat{\mathbf{w}}^T\mathbf{x}_{test} - y_{test})^2] = E[(\widehat{\mathbf{w}}^T\mathbf{x}_{test} - \mathbf{w}^T\mathbf{x}_{test} - \epsilon)^2] = E[((\widehat{\mathbf{w}}^T - \mathbf{w}^T)\mathbf{x}_{test} - \epsilon)^2].$$

By linearity of expectation, we can write this as

$$E[((\widehat{\mathbf{w}}^T - \mathbf{w}^T)\mathbf{x}_{test})^2] + E[\epsilon^2] - 2E[(\widehat{\mathbf{w}}^T - \mathbf{w}^T)\mathbf{x}_{test}\epsilon].$$

By independence,

$$E[(\widehat{\mathbf{w}}^T - \mathbf{w}^T)\mathbf{x}_{test}\epsilon] = E[(\widehat{\mathbf{w}}^T - \mathbf{w}^T)\mathbf{x}_{test}]E[\epsilon] = 0,$$

since $E[\epsilon] = 0$. Thus we drop the final term, and $E[\epsilon^2] = \sigma^2$ is constant regardless of $\widehat{\mathbf{w}}$, so we can also drop this term. Next we expand as follows

$$E[((\widehat{\mathbf{w}}^T - \mathbf{w}^T)\mathbf{x}_{test})^2] = E[(\widehat{\mathbf{w}}^T - \mathbf{w}^T)\mathbf{x}_{test}]^2 + \mathrm{Var}\big((\widehat{\mathbf{w}}^T - \mathbf{w}^T)\mathbf{x}_{test}\big).$$

By independence $E[(\widehat{\mathbf{w}}^T - \mathbf{w}^T)\mathbf{x}_{test}] = E[\widehat{\mathbf{w}}^T - \mathbf{w}^T]E[\mathbf{x}_{test}] = 0$ since $E[\mathbf{x}_{test}] = 0$. Thus we are left with

$$\mathrm{Var}\big((\widehat{\mathbf{w}}^T - \mathbf{w}^T)\mathbf{x}_{test}\big).$$

Now we'll take a break to show that for independent X and Y,

$$\mathrm{Var}(XY) = \mathrm{Var}(X)\mathrm{Var}(Y) + \mathrm{Var}(X)E[Y]^2 + \mathrm{Var}(Y)E[X]^2.$$

This is a result I found on Stats Stack Exchange by damla (see resources section for link), but I came up with the proof on my own. We will write all terms as products of $E[Y^2]$ and $E[Y]^2$. By definition,

$$\mathrm{Var}(X)\mathrm{Var}(Y) = (E[X^2] - E[X]^2)(E[Y^2] - E[Y]^2) = (E[X^2] - E[X]^2)E[Y^2] + (E[X]^2 - E[X^2])E[Y]^2.$$

Additionally,

$$\mathrm{Var}(X)E[Y]^2 = (E[X^2] - E[X]^2)E[Y]^2,$$

and

$$\mathrm{Var}(Y)E[X]^2 = E[X]^2E[Y^2] - E[X]^2E[Y]^2.$$

Adding up coefficients yields

$$(E[X^2] - E[X]^2 + E[X]^2)E[Y^2] + (E[X]^2 - E[X^2] + E[X^2] - E[X]^2 - E[X]^2)E[Y]^2.$$

Cancelling yields

$$(E[X^2])E[Y^2] + (-E[X]^2)E[Y]^2 = E[X^2]E[Y^2] - E[X]^2E[Y]^2.$$

By independence,

$$E[X^2]E[Y^2] - E[X]^2E[Y]^2 = E[X^2Y^2] - E[XY]^2 = E[(XY)^2] - E[XY]^2 = \mathrm{Var}(XY).$$

Thus

$$\mathrm{Var}\big((\widehat{\mathbf{w}}^T - \mathbf{w}^T)\mathbf{x}_{test}\big) = \mathrm{Var}\big(\widehat{\mathbf{w}}^T - \mathbf{w}^T\big)\mathrm{Var}\big(\mathbf{x}_{test}\big) + \mathrm{Var}\big(\mathbf{x}_{test}\big)E[\widehat{\mathbf{w}}^T - \mathbf{w}^T]^2 + \mathrm{Var}\big(\widehat{\mathbf{w}}^T - \mathbf{w}^T\big)E[\mathbf{x}_{test}]^2.$$

By the given, $E[\mathbf{x}_{test}] = 0$, and $\mathrm{Var}(\mathbf{x}_{test}) = 1$. Plugging in these values leaves us with

$$\mathrm{Var}\big(\widehat{\mathbf{w}}^T - \mathbf{w}^T\big) + E[\widehat{\mathbf{w}}^T - \mathbf{w}^T]^2.$$

Note that $\widehat{\mathbf{w}}$ is a constant once we've observed the training data $\mathbf{X}$ and $\mathbf{y}$. Thus

$$\mathrm{Var}\big(\widehat{\mathbf{w}}^T - \mathbf{w}^T\big) = \mathrm{Var}\big(\mathbf{w}^T\big),$$

and this term is the same regardless of which estimate we use.

Thus we simply consider $E[\widehat{\mathbf{w}}^T - \mathbf{w}^T]^2$ for both estimates. Since $\mathbf{w}_{mean}$ is defined to be $E[\mathbf{w}]$, this term is equal to 0 for the estimate $\mathbf{w}_{mean}$. Since the quantity has a square, it is nonnegative for $\mathbf{w}_{MAP}$. Thus $\mathbf{w}_{mean}$ always has lower or equal expected test MSE.

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:

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

where $\widehat{y}$ 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:

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

   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 $L = -(y \log(\widehat{y}) + (1-y)\log(1-\widehat{y}))$. 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 dimensions of $\mathbf{W}_1$ are $H \times M$ since we matrix multiply $\mathbf{W}_1$ and $\mathbf{x}$ (which is a $M \times 1$ column vector). The dimensions of $\mathbf{b}_1$ are $H \times 1$ since we add it to $\mathbf{W}_1\mathbf{x}$, and $\mathbf{a_1}$ is $H \times 1$ since it is the sum of $\mathbf{W}_1\mathbf{x}$ and $\mathbf{b}_1$. $\mathbf{z}_1$ is also $H \times 1$ since the sigmoid is applied element-wise, and $\mathbf{a}_1$ is $H \times 1$.

   The dimensions of $\mathbf{W}_2$ are $1 \times H$ since we matrix multiply $\mathbf{W}_2$ and $\mathbf{z}_1$ (which is a $H \times 1$ column vector). The dimension of $\mathbf{b}_2$ are 1×1 since we add it to $\mathbf{W}_2\mathbf{z}_1$, and a_2 is a scalar since we can get it by adding 1×1 values. $\widehat{y} = z_2$ is also a scalar since it is the sigmoid function applied to a_2.

2. (a) By the chain rule,

$$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial \widehat{y}} \frac{\partial \widehat{y}}{\partial a_2} \frac{\partial a_2}{\partial b_2} = -\big(\frac{y}{\widehat{y}} - \frac{1-y}{1-\widehat{y}}\big) \cdot (\widehat{y}(1-\widehat{y})) \cdot 1 = -y(1-\widehat{y}) + (1-y)\widehat{y} = \widehat{y} - y.$$

(b) By the chain rule,

$$\frac{\partial L}{\partial W_2^h} = \frac{\partial L}{\partial \widehat{y}} \frac{\partial \widehat{y}}{\partial a_2} \frac{\partial a_2}{\partial W_2^h} = -\big(\frac{y}{\widehat{y}} - \frac{1-y}{1-\widehat{y}}\big) \cdot (\widehat{y}(1-\widehat{y})) \cdot z_1^h = z_1^h(\widehat{y} - y).$$

(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.) By the chain rule,

$$\frac{\partial L}{\partial b_1^h} = \frac{\partial L}{\partial \widehat{y}} \frac{\partial \widehat{y}}{\partial a_2} \frac{\partial a_2}{\partial z_1^h} \frac{\partial z_1^h}{\partial a_1^h} \frac{\partial a_1^h}{\partial b_1^h} = -\big(\frac{y}{\widehat{y}} - \frac{1-y}{1-\widehat{y}}\big) \cdot (\widehat{y}(1-\widehat{y})) \cdot W_2^h \cdot (z_1^h(1-z_1^h)) \cdot 1 = (\widehat{y}-y) W_2^h z_1^h (1-z_1^h).$$

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

$$\frac{\partial L}{\partial W_1^{h,m}} = \frac{\partial L}{\partial \widehat{y}} \frac{\partial \widehat{y}}{\partial a_2} \frac{\partial a_2}{\partial z_1^h} \frac{\partial z_1^h}{\partial a_1^h} \frac{\partial a_1^h}{\partial W_1^{h,m}} = -\big(\frac{y}{\widehat{y}} - \frac{1-y}{1-\widehat{y}}\big) \cdot (\widehat{y}(1-\widehat{y})) \cdot W_2^h \cdot (z_1^h(1-z_1^h)) \cdot x_m = (\widehat{y}-y) W_2^h z_1^h (1-z_1^h) x_m.$$

Problem 4 (Modern Deep Learning Tools: PyTorch)

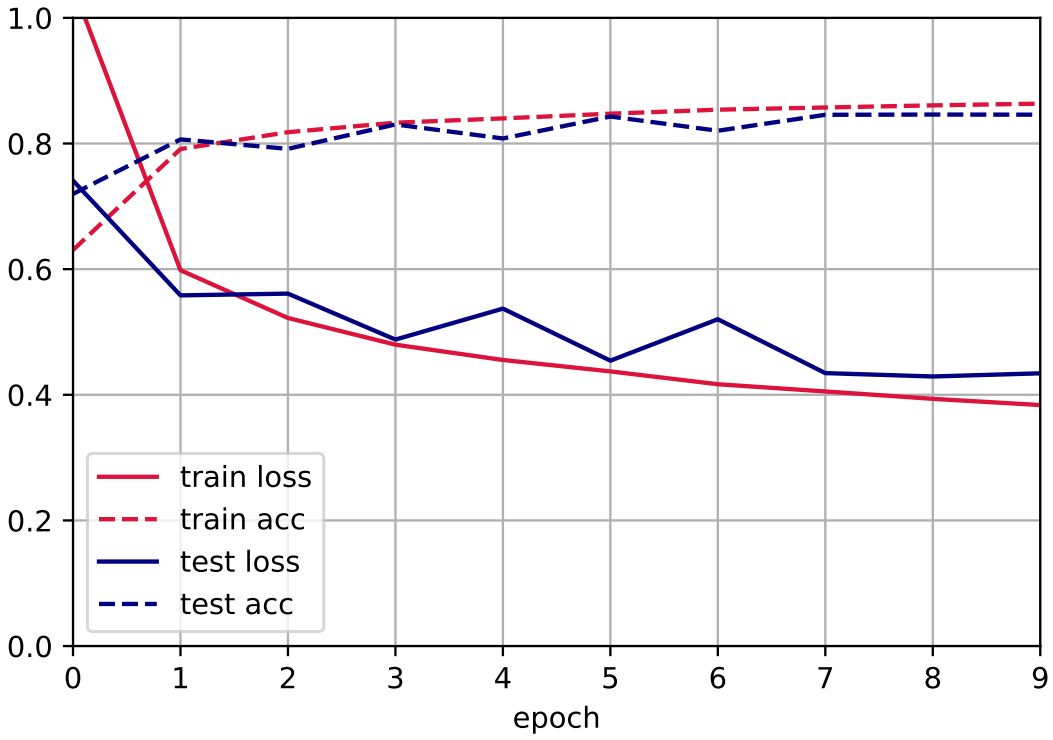
In this problem, you will learn how to use PyTorch. This machine learning library is massively popular and used heavily throughout industry and research. In `T3_P3.ipynb` you will implement an MLP for image classification from scratch. Copy and paste code solutions below and include a final graph of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

You will receive no points for code not included below.

You will receive 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(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 torch.clamp(x, min=0)
```

```python
def softmax(x):
    exp = torch.exp(X)
    return exp / torch.sum(exp, axis=1, keepdims=True)



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



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



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



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

Name

Christy Jestin

Collaborators and Resources

Whom did you work with, and did you use any resources beyond cs181-textbook and your notes?
I worked with David Qian and used Google for syntax. I used https://stats.stackexchange.com/questions/52646/variance-of-product-of-multiple-independent-random-variables to get a result for the variance of the product of independent random variables, but I derived the proof myself.

Calibration

Approximately how long did this homework take you to complete (in hours)?
The pset took about 16 hours.