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. To find the desired distribution, we'll first use Bayes' rule, then drop unnecessary multiplicative constants, and then match the resulting expression to a known distribution. Let's begin by stating the Bayes' rule:

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

This states that the posterior distribution is proportional to the product of the likelihood of the observed data given a particular μ and the prior distribution on μ. This need not integrate to 1, which is why we need the normalizing term $p(D)$. However, this does not depend on μ, so we can safely ignore it for now. Next, note that by an iid assumption on the data, we have:

$$p(\mu|D) \propto \left(\prod_{i=1}^{n} p(x_i|\mu)\right) p(\mu)$$

Further note that from $x_i = \mu + \epsilon_i$, we have $x_i \sim \mathcal{N}(\mu, \sigma^2)$. Then we can plug in the normal PDF for both the likelihood and the prior (which has mean zero) to get:

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

Again, let's drop multiplicative constants since they get normalized out anyway:

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

Note that the $\sum x_i^2$ term does not depend on μ, so we can take it out of the exponential and forget about it as it's a multiplicative constant. Then we're left with:

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

Next, we'll complete the square. That is, for a quadratic in form of $a\mu^2 + b\mu$, we'll write it as $a(\mu+d)^2 + e$ with $d = \frac{b}{2a}$ and $e = -\frac{b^2}{4a}$. Then setting:

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

We get:

$$d = -\frac{\frac{2}{\sigma^2}\sum_{i=1}^{n} x_i}{2\frac{n\tau^2+\sigma^2}{\sigma^2\tau^2}} = -\frac{\sigma^2\tau^2\sum_{i=1}^{n} x_i}{\sigma^2(n\tau^2+\sigma^2)} = -\frac{\tau^2\sum_{i=1}^{n} x_i}{n\tau^2+\sigma^2}$$

$$e = -\frac{\left(\frac{2}{\sigma^2}\sum_{i=1}^{n} x_i\right)^2}{4\frac{n\tau^2+\sigma^2}{\sigma^2\tau^2}} = -\frac{\sigma^2\tau^2\left(\sum_{i=1}^{n} x_i\right)^2}{\sigma^4(n\tau^2+\sigma^2)} = -\frac{\tau^2\left(\sum_{i=1}^{n} x_i\right)^2}{\sigma^2(n\tau^2+\sigma^2)}$$

So we can complete the square by writing:

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

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

This corresponds to a normal distribution and its PDF, as it's the only distribution where $p(x) \propto \exp(-\frac{1}{2}x^2)$. Then we can take its parameters (mean and variance) out of the expression above to finally proclaim:

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

2. (a) We know that the likelihood of the data is proportional to:

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

from above. Then we can find the MLE by first taking a log:

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

Then differentiating wrt μ and setting to 0:

$$\frac{\partial l}{\partial \mu} = \frac{1}{2\sigma^2}\sum_{i=1}^n 2(x_i-\mu) = \frac{1}{\sigma^2}\sum_{i=1}^n x_i - \frac{n}{\sigma^2}\mu = 0$$

Which implies:

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

The sample mean of the data set!

(b) Now we're instead trying to maximize the posterior distribution $p(\mu|D)$, or in other words, maximizing:

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

We can also maximize the log-probability (the log is monotonous) as:

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

Differentiate and set equal to 0:

$$-\frac{1}{\frac{\sigma^2\tau^2}{n\tau^2+\sigma^2}}\left(\mu - \frac{\tau^2 \sum_{i=1}^n x_i}{n\tau^2+\sigma^2}\right) = 0$$

This characterizes the MAP estimator:

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

(c) From the equation that's given to us (1), we have that the mean of $x|D$ is:

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

So the estimators are actually equal.

(d) We have $x_i = \mu + \epsilon_i$, so for fixed μ^*, we simply get a normal distribution with a mean equal to μ^* (all the randomness is driven by ϵ):

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

Applying this to the estimators above, we get:

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

(e) The variance of $x|D$ is:

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

greater than the variance of $x|\mu_{MLE}$ since n, σ, τ are all positive parameters. This is intuitive: for $x|D$, our uncertainty comes from both the prior distribution of μ (we don't know exactly what μ is and can only estimate it from data) and from the random noise around μ (the normal distribution of ϵ). In comparison, μ_{MLE} is a constant, so randomness comes only from ϵ and the variance is lower. This could be shown formally through Eve's law, but the above suffices for intuition.

Next, as $n \to \infty$, we get $\frac{n}{\sigma^2} \to \infty$ for any fixed σ^2 and therefore $(\frac{n}{\sigma^2} + \frac{1}{\tau^2})^{-1} \to 0$. Then what we get is that the limit of the variance of $x|D$ as $n \to \infty$ is simply:

$$\sigma^2$$

equal to the variance of $x|\mu_{MLE}$. This makes perfect sense: with infinitely many observations, we can infer the true μ arbitrarily precisely, so we get rid of all variance stemming from uncertainty about μ (from its prior distribution). What is left is just the randomness of x around μ, specifically ϵ with variance σ^2.

3. (a) Assuming $\sum_{i=1}^{n} x_i = 0$, we get that:

$$\mu_{MLE} = \bar{x} = 0$$
$$\mu_{MAP} = \frac{\tau^2}{n\tau^2 + \sigma^2} 0 = 0$$

So then $\mu_{MLE} = \mu_{MAP} = 0$, the data set coerces both estimators of mean to be zero.

(b) Now assuming $\sum_{i=1}^{n} x_i > 0$, note that:

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

Which then means:

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

The map estimator is smaller than the MLE.

(c) Finally, if $\sum_{i=1}^{n} x_i < 0$, we can repeat the analysis above and get:

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

So the MAP estimator is larger. In fact, it seems as though the MAP estimator is like the MLE but with "attenuation bias" that forces it to be closer to zero.

(d) First, let:

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

Now taking the limit as $n \to \infty$, I am lazy to actually reason about what will happen, so let's just use l'Hopital's:

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

So the estimators are equal in the limit (with infinite data points).

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. We will be working with multiplicative constants throughout this problem, since we're only asked to derive the posterior up to a constant (that we could then find by integration. Then by Bayes' rule:

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

This is a classic 'posterior is proportional to likelihood times prior' expression. Now note that by an iid assumption on the errors as well as the MVN PDF, for a single data point $\mathbf{x}_i$, we have:

$$\begin{aligned} p(x_i|\mathbf{w}, y_i) &= \frac{1}{\sqrt{\det(2\pi\sigma^2 I_p)}} \exp\left(-\frac{1}{2}(y_i - \mathbf{w}^T x_i)^T(\sigma^2 I_p)^{-1}(y_i - \mathbf{w}^T x_i)\right) \\ &= c_0 \exp\left(-\frac{1}{2}(y_i - \mathbf{w}^T x_i)^T(\sigma^2 I_p)^{-1}(y_i - \mathbf{w}^T x_i)\right) \end{aligned}$$

Since $\Sigma = \sigma^2 I_p$ where I_p is the identity and c_0 is a constant wrt $\mathbf{w}$. Then using an iid assumption for the data set:

$$\begin{aligned} p(\mathbf{X}|\mathbf{y}, \mathbf{w}) &= \prod_{i=1}^{n} p(x_i|\mathbf{w}, y_i) = \prod_{i=1}^{n} c_0 \exp\left(-\frac{1}{2}(y_i - \mathbf{w}^T x_i)^T(\sigma^2 I_p)^{-1}(y_i - \mathbf{w}^T x_i)\right) \\ &= c_1 \exp\left(-\frac{1}{2}\sum_{i=1}^{n}(y_i - \mathbf{w}^T x_i)^T(\sigma^2 I_p)^{-1}(y_i - \mathbf{w}^T x_i)\right) \end{aligned}$$

Now multiplying by the prior:

$$\begin{aligned} p(\mathbf{w}|\mathbf{X}, \mathbf{y}) &\propto c_1 \exp\left(-\frac{1}{2}\sum_{i=1}^{n}(y_i - \mathbf{w}^T x_i)^T(\sigma^2 I_p)^{-1}(y_i - \mathbf{w}^T x_i)\right) \frac{1}{2\tau} \exp\left(-\frac{||\mathbf{w} - \mu||_1}{\tau}\right) \\ &= c_2 \exp\left(-\frac{1}{2}\sum_{i=1}^{n}(y_i - \mathbf{w}^T x_i)^T(\sigma^2 I_p)^{-1}(y_i - \mathbf{w}^T x_i)\right) \exp\left(-\frac{1}{\tau}\sum_{j=1}^{p}|w_i - \mu_i|\right) \\ &= c_2 \exp\left(-\frac{1}{2}\sum_{i=1}^{n}(y_i - \mathbf{w}^T x_i)^T(\sigma^2 I_p)^{-1}(y_i - \mathbf{w}^T x_i) - \frac{1}{\tau}\sum_{j=1}^{p}|w_i - \mu_i|\right) \end{aligned}$$

2. Next, to find $\mathbf{w}_{MAP}$, we want to maximize the posterior. Notice that all the constants hidden away in c_2 are positive, so this is the same as maximizing the log posterior:

$$\log p(\mathbf{w}|\mathbf{X}, \mathbf{y}) \propto -\frac{1}{2}\sum_{i=1}^{n}(y_i - \mathbf{w}^T x_i)^T(\sigma^2 I_p)^{-1}(y_i - \mathbf{w}^T x_i) - \frac{1}{\tau}\sum_{j=1}^{p}|w_i - \mu_i|$$

We can maximize this by taking a derivative and setting it to 0, but instead, I'll just leave an unsimplified form:

$$\mathbf{w}_{MAP} = \arg\max_{\mathbf{w}}\left[-\frac{1}{2}\sum_{i=1}^{n}(y_i - \mathbf{w}^T x_i)^T(\sigma^2 I_p)^{-1}(y_i - \mathbf{w}^T x_i) - \frac{1}{\tau}\sum_{j=1}^{p}|w_i - \mu_i|\right]$$

This is very similar to regularization in the frequentist perspective. The first term here governs how well the parameter "fits" the data, specifically a normal distribution. The second term expresses how far the given set of weights is from a prior mean vector μ and penalizes for high distances. The solution $\mathbf{w}_{MAP}$ will therefore be a compromise between a very close MVN fit and similarity to μ. The parameter τ governs how much weight is placed on the regularization term: if τ is small, then the second term will dominate, and the log-posterior will be maximized when $\mathbf{w}$ closely matches μ. Therefore, we can relate it to the frequentist regularization term by stating $\tau \approx \frac{1}{\lambda}$: large λ means more weight on regularization than on the likelihood.

3. We would incorporate expert knowledge by setting up a prior based on our knowledge about the distribution of $\mathbf{w}$. That is, if we had high confidence about some vector $\mathbf{v}$, we'd set $\mu = \mathbf{v}$ and choose a small $\tau 0$. This would force the MAP estimator $\mathbf{w}_{MAP}$ to be close to $\mathbf{v}$ while still incorporating some information learned form observed data. This makes sense from a Bayesian viewpoint, since a well-known "usual" value means that we have a tight prior distribution around $\mathbf{v}$ that is only somewhat updated by data, and from a frequentist viewpoint because we're imposing a heavy regularization penalty for weight vectors that fit the data well bust stray far from $\mathbf{v}$.

4. As τ decreases, the second term in the arg max expression becomes much larger (in absolute value) than the first term. This means that the maximum will be reached by keeping the second term at or very close to 0, which means picking $\mathbf{v} \approx \mu$. This is taken to the extreme in the limit $\tau \to 0$: we can in fact make the second expression larger in magnitude than the first expression, for any number of observed data points n and any specific values $\mathbf{X}, \mathbf{y}$, in effect bringing $\mathbf{w}_{MAP}$ arbitrarily close to $\mathbf{v}$. This makes sense: if our prior is being absolutely certain about $\mathbf{w}$ being a specific vector, then observed data won't change our estimate at all, even if they appear inconsistent with it.

5. The MLE mininmizes mean squared error because it's an efficient estimator that reaches the Cramer-Rao bound. Therefore, it's MSE must be lower or equal than that of $\mathbf{w}_{MAP}$.

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:

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

   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:

1. Starting with the given information, we want $\mathbf{a_1}$ to be a $H \times 1$ vector since the sigmoid function is applied element-wise and does not change dimensionality. For parameters, this means that the additive bias term $\mathbf{b_1}$ is also $H \times 1$, and the matrix $\mathbf{W_1}$ is $H \times M$, which ensures that $\mathbf{W_1 x}$ is a valid matrix operation that outputs a $H \times 1$ vector for x with dimensionality $M \times 1$. The sigmoid does not change

dimensions, so z_1 is $H \times 1$ as well.

Next, we want the output to be a 1×1 scalar, so z_2 is 1×1. The sigmoid does not change dimension, so a_2 is also a scalar. Then we need $\mathbf{b_2}$ to be a 1×1 scalar so that addition is well-defined, and $\mathbf{W_2}$ to be $1 \times H$ such that $\mathbf{W_2 z_1}$ is valid and 1×1.

2. A couple of notes before we start. The model parameters enter the loss function only through the estimator $\hat{y}$, so we will proceed by computing $\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \theta}$ for a parameter θ. Further, note that:

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

Also note that for $\sigma(x) = \frac{1}{1+e^{-x}}$, we have $\sigma'(x) = -\frac{1}{(1+e^{-x})^2}(-e^{-x}) = \frac{e^{-x}}{1+e^{-x}} \frac{1}{1+e^{-x}} = (1-\sigma(x))\sigma(x)$. I will cary the sigmoid notation forward for simplicity instead of writing it out fully. Now we can proceed:

(a) Chain rule:

$$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_2} \frac{\partial a_2}{\partial b_2} = \left(\frac{1-y}{1-\hat{y}} - \frac{y}{\hat{y}} \right) (1-\sigma(a_2))\sigma(a_2)$$

With a_2 defined as above, because the third term $\frac{\partial a_2}{\partial b_2}$ is just 1.

(b) Again, let's use the chain rule:

$$\frac{\partial L}{\partial W_2^h} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_2} \frac{\partial a_2}{\partial W_2^h} = \left(\frac{1-y}{1-\hat{y}} - \frac{y}{\hat{y}} \right) (1-\sigma(a_2))\sigma(a_2) z_1^h$$

The first two terms are the same, but $\frac{\partial a_2}{\partial W_2^h}$ is equal to the h-th element of $\mathbf{z_1}$, which I call z_1^h, since the parameter $\mathbf{W_2}$ enters into an inner product with $\mathbf{z_1}$.

(c) We will start with the same partials, but then need to string together a couple more. In particular, from the hint, we know that b_1^h only enters the h-th element z_1^h which equals $\sigma(a_1^h)$. This means that the last two partials are derivatives of scalars. Also, the element z_1^h is multiplied by the corresponding W_2^h in the definition of a_2. Then:

$$\frac{\partial L}{\partial b_1^h} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_2} \frac{\partial a_2}{\partial z_1^h} \frac{\partial z_1^h}{\partial a_1^h} \frac{\partial a_1^h}{\partial b_1^h} = \left(\frac{1-y}{1-\hat{y}} - \frac{y}{\hat{y}} \right) (1-\sigma(a_2))\sigma(a_2) W_2^h (1-\sigma(a_1^h))\sigma(a_1^h)$$

Since z_1^h is multiplied by W_2^h, the derivative of the sigmoid is defined above, and the final term $\frac{\partial a_1^h}{\partial b_1^h}$ is just 1.

(d) Similar to the previous part, except for the last step. In the last step, notice that $W_1^{h,m}$ multiplies the m-th element of $\mathbf{x}$ which I'll call x^m, and then the result feeds into the h-th element of $\mathbf{a_1}$, and therefore also the h-th element of z_1. Proceeding:

$$\frac{\partial L}{\partial W_1^{h,m}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_2} \frac{\partial a_2}{\partial z_1^h} \frac{\partial z_1^h}{\partial a_1^h} \frac{\partial a_1^h}{\partial W_1^{h,m}} = \left(\frac{1-y}{1-\hat{y}} - \frac{y}{\hat{y}} \right) (1-\sigma(a_2))\sigma(a_2) W_2^h (1-\sigma(a_1^h))\sigma(a_1^h) x^m$$

Problem 4 (Modern Deep Learning Tools: PyTorch)

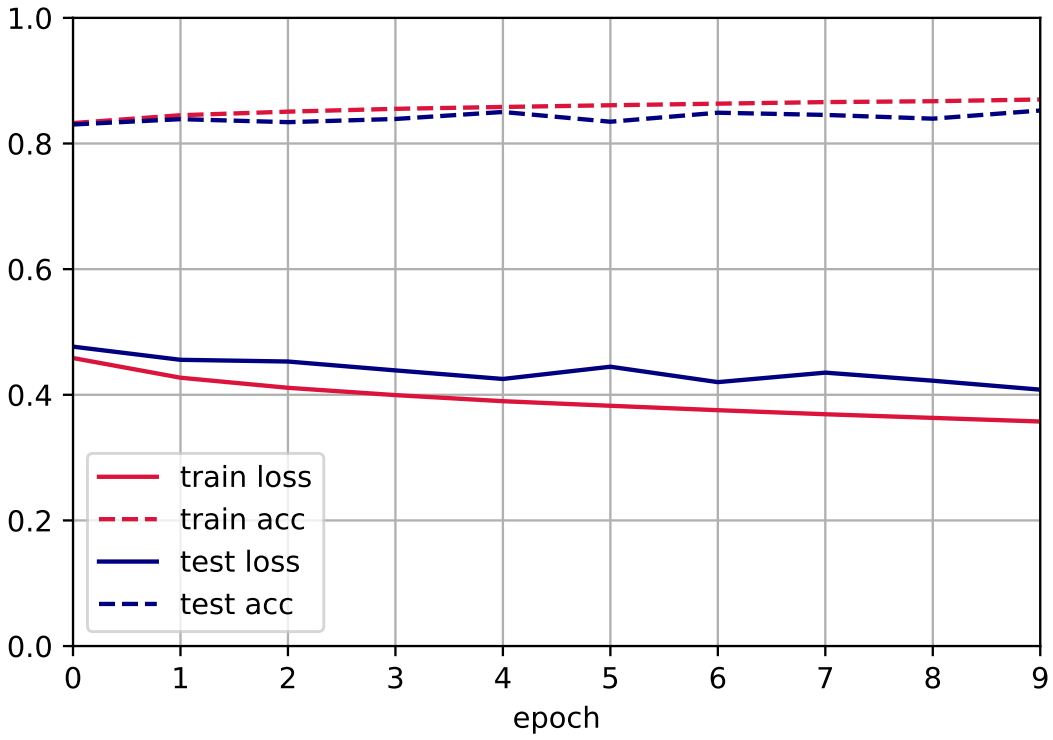
In this problem, you will learn how to use PyTorch. This machine learning library is massively popular and used heavily throughout industry and research. In `T3_P3.ipynb` you will implement an MLP for image classification from scratch. Copy and paste code solutions below and include a final graph of 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**2
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)))

params = [W1, b1, W2, b2]
```

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

def softmax(X):
  expx = torch.exp(X)
  divs = 1/(torch.sum(expx, 1))
  return expx * divs[:,None]



def net(X):
  Xf = X.flatten(start_dim=1)
  H = relu(torch.matmul(Xf, params[0]) + params[1])
  O = softmax(torch.matmul(H, params[2]) + params[3])
  return O



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


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



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

      loss.backward()
      updater(params)
```

Name

Matej Cerman

Collaborators and Resources

No collaborators. For resources, I used https://www.mathsisfun.com/algebra/completing-square.html, https://people.eecs.berkeley.edu/ jordan/courses/260-spring10/lectures/lecture5.pdf, and https://en.wikipedia.org/wiki/Normal$_d istribution$.

Calibration

About 12 hours, most spent debugging the NN.