Homework Assignment # 4

Due: Friday, December 8, 2022, 11:59 p.m.
Total marks: 100

Question 1. [10 MARKS]

In this question we will reason about the convergence rate of logistic regression. Recall that the loss for logistic regression is

$$c(\mathbf{w}) \doteq \frac{1}{n}\sum_{i=1}^{n} c_i(\mathbf{w}) \qquad \text{for} \quad c_i(\mathbf{w}) \doteq -y_i \ln \sigma(\mathbf{x}_i^\top \mathbf{w}) - (1-y_i)\ln(1-\sigma(\mathbf{x}_i^\top \mathbf{w}))$$

Our convergence rate bounds included the Lipschitz constant for our gradients. In particular, we assumed that there is an $L > 0$ such that for any $\mathbf{w}_1, \mathbf{w}_2$,

$$\|\nabla c(\mathbf{w}_1) - \nabla c(\mathbf{w}_2)\| \leq L\|\mathbf{w}_1 - \mathbf{w}_2\|$$

where throughout this question the norm $\|\mathbf{v}\|$ for a vector $\mathbf{v}$ means the ℓ_2 norm $\|\mathbf{v}\|_2$. In this question, you will reason about the Lipschitz constant for logistic regression. Assume that someone has kindly normalized the inputs for you such that $\|\mathbf{x}_i\| \leq 1$.

Let us start some of the reasoning for you. First recall that $\nabla c_i(\mathbf{w}) = (\sigma(\mathbf{x}_i^\top \mathbf{w}) - y_i)\mathbf{x}_i$. So we get that

$$\begin{aligned}
&\|\nabla c(\mathbf{w}_1) - \nabla c(\mathbf{w}_2)\| \\
&= \frac{1}{n}\left\|\sum_{i=1}^{n}(\sigma(\mathbf{x}_i^\top \mathbf{w}_1) - y_i)\mathbf{x}_i - \sum_{i=1}^{n}(\sigma(\mathbf{x}_i^\top \mathbf{w}_2) - y_i)\mathbf{x}_i\right\| && \text{constant } \tfrac{1}{n} \text{ comes out } \|\tfrac{1}{n}\mathbf{v}\| = \tfrac{1}{n}\|\mathbf{v}\| \\
&= \frac{1}{n}\left\|\sum_{i=1}^{n}(\sigma(\mathbf{x}_i^\top \mathbf{w}_1)\mathbf{x}_i - \sum_{i=1}^{n}(\sigma(\mathbf{x}_i^\top \mathbf{w}_2)\mathbf{x}_i\right\| && \text{shared term } y_i\mathbf{x}_i \text{ cancels} \\
&\leq \frac{1}{n}\sum_{i=1}^{n}|\sigma(\mathbf{x}_i^\top \mathbf{w}_1) - \sigma(\mathbf{x}_i^\top \mathbf{w}_2)|\|\mathbf{x}_i\| && \left\|\sum_i d_i\mathbf{v}_i\right\| \leq \sum_i |d_i|\|\mathbf{v}\| \\
&\leq \frac{1}{n}\sum_{i=1}^{n}|\sigma(\mathbf{x}_i^\top \mathbf{w}_1) - \sigma(\mathbf{x}_i^\top \mathbf{w}_2)| && \|\mathbf{x}_i\| \leq 1 \\
&\leq \frac{1}{n}\sum_{i=1}^{n} L\|\mathbf{w}_1 - \mathbf{w}_2\| = L\|\mathbf{w}_1 - \mathbf{w}_2\|
\end{aligned}$$

where the last step follows from using the Lipschitz constant L for $\sigma(\mathbf{x}_i^\top \mathbf{w})$.

(a) [5 MARKS] Show that the Lipschitz constant L for $\sigma(\mathbf{x}_i^\top \mathbf{w})$ is $L = 1$. **Hint**: Use the fact that for a function $g(\mathbf{w})$ it's Lipschitz constant L satisfies $\|\nabla g(\mathbf{w})\| \leq L$. In other words, find an upper bound on the gradient of $\sigma(\mathbf{x}_i^\top \mathbf{w})$.

(b) [3 MARKS] Now imagine we decide to add ℓ_2 regularization to our logistic regression objective, giving new objective $\tilde{c}(\mathbf{w}) = c(\mathbf{w}) + \frac{\lambda}{2n}\|\mathbf{w}\|^2$. What is the Lipschitz constant $\tilde{L}$ for the gradient of $\tilde{c}$? **Hint**: Recall that $\|\mathbf{a} + \mathbf{b}\| \leq \|\mathbf{a}\| + \|\mathbf{b}\|$ and use the fact that you know the Lipschitz constant for the gradient of c.

(c) [2 MARKS] Finally, imagine that we now first expand $\mathbf{x}_i \in \mathbb{R}^d$ into a new feature space using

RBF networks, to get $p > d$ features $\boldsymbol{\phi}(\mathbf{x}_i) \in \mathbb{R}^p$. For simplicity, let's write these new inputs as $\boldsymbol{\phi}_i \in \mathbb{R}^p$, where the bolding emphasizes that this is the feature *vector* for sample i (rather than an index into the feature vector). These features $\boldsymbol{\phi}_i$ are not normalized and so we cannot say that $\|\boldsymbol{\phi}_i\| \leq 1$. But, since we are using RBF features, we know that each element in the vector is between 0 and 1 and so $\|\boldsymbol{\phi}_i\| \leq \sqrt{p}$. In this setting, what is this Lipschitz constant for our logistic regression loss,

$$c(\mathbf{w}) \doteq \frac{1}{n}\sum_{i=1}^{n} c_i(\mathbf{w}) \qquad \text{for} \quad c_i(\mathbf{w}) \doteq -y_i \ln \sigma(\boldsymbol{\phi}_i^\top \mathbf{w}) - (1-y_i)\ln(1-\sigma(\boldsymbol{\phi}_i^\top \mathbf{w})).$$

Hint: here you can write the answer in terms of L, where L is now the Lipschitz constant for $\sigma(\boldsymbol{\phi}_i^\top \mathbf{w})$ (instead of $\sigma(\mathbf{x}_i^\top \mathbf{w})$)

Question 2. [10 MARKS]

In machine learning, we sometimes want to use an expectation as a loss $L(\theta)$ and compute

$$\nabla_\theta L(\theta) = \nabla_\theta \mathbb{E}_{\boldsymbol{X}\sim P_\theta(\cdot)}[f_\theta(\boldsymbol{X})].$$

We used just such a loss for generative models, where the goal was to learn a parameterized generative model P_θ that generates plausible $\boldsymbol{X}$. However, sampling this gradient is less straightforward than it was for our predictive models (our generalized linear models and extensions use fixed basis and neural networks). You will reason about this issue in this question.

You can use the fact that

$$\nabla_\theta \mathbb{E}_{\boldsymbol{X}\sim P_\theta(\cdot)}[f_\theta(\boldsymbol{X})] = \mathbb{E}_{\boldsymbol{X}\sim P_\theta(\cdot)}[f_\theta(\boldsymbol{X})\nabla_\theta \log P_\theta(\boldsymbol{X})] + \mathbb{E}_{\boldsymbol{X}\sim P_\theta(\cdot)}[\nabla_\theta f_\theta(\boldsymbol{X})] \tag{1}$$

(a) [3 MARKS] Intuitively, it feels like we should be able to sample $\boldsymbol{X}_i \sim P_\theta$, n times, and then use $\frac{1}{n}\sum_{i=1}^n \nabla_\theta f_\theta(\boldsymbol{X}_i)$ to get an estimate of the gradient in (1). Explain why $\frac{1}{n}\sum_{i=1}^n \nabla_\theta f_\theta(\boldsymbol{X}_i)$ is not an unbiased estimate of $\nabla_\theta L(\theta)$. *Hint:* This question is meant to be straightforward. Reason about what $\frac{1}{n}\sum_{i=1}^n \nabla_\theta f_\theta(\boldsymbol{X}_i)$ estimates (its true expectation) and compare that with Equation (1).

(b) [7 MARKS] Some $P_\theta(\cdot)$ allow us to rewrite the expectation in a convenient way. For example, if $\theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ and $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, we have $\boldsymbol{X} = X_\theta(\boldsymbol{\epsilon}) = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then,

$$\mathbb{E}_{\boldsymbol{X}\sim P_\theta(\cdot)}[f_\theta(\boldsymbol{X})] = \mathbb{E}_{\boldsymbol{X}\sim \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}[f_\theta(\boldsymbol{X})] = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[f_\theta(X_\theta(\boldsymbol{\epsilon}))] = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[f_\theta(\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon})]$$

Now, equivalently, we can get an unbiased estimate of $\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[f_\theta(\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon})]$ using a sample average over n sampled $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Explain the procedure to do so.

Question 3. [25 MARKS]

This question uses `A4.jl`, specifically the components that help you visualize the hidden layers in neural networks. You do not have to implement anything for this question. Instead, you will be using the code given and reporting the behavior that you see. In some cases, you will need to save plots in this section to submit them in your answers; the notebook tells you how to save these plots. Even if a question does not require you to include a plot, if you feel it helps you answer the question, then do include one!

(a) [5 MARKS]

In the section titled **Visualizing the Decision Surface** you should see a number of drop-down menus that you can play with and a plot that shows the output of the learned network.

1. (3 marks) Look at the decision surface for the three activation functions, and notice that the one for the `identity` activation is quite different. Explain why we get this decision surface when using the `identity` activation.

2. (2 marks) If you set the activation function to `relu` and play with the number of layers and hidden units per layer, what do you notice about the decision boundary?

   Note: you might also find (disconcertingly) that sometimes the model fails and is not able to separate the classes. For example, you might find that `relu` with 1 hidden layer and 2 hidden units performs well and giving more capacity to `relu` with 2 hidden layer and 2 hidden units results in failure! This can occur simply due to different random initializations. For this question, you do not have to report on failures. Instead, focus on trends in the networks with `relu` in terms of increasing hidden layers and hidden units.

(b) [5 MARKS]

For this question, we will use the visualization in the section titled **Visualizing Hidden Unit Activations**. This plot shows a single hidden unit's activation across the set of inputs to the network. Go through and look at the different activations for the different hidden units.

Describe what one of the activations seems to be representing and why that might be useful for classifying. Include a plot to make it more concrete.

(c) [8 MARKS]

This question goes through **Data Transformations**. We will be looking at the transformations the data goes through so the ANN can accurately classify the data. We will be comparing a successful model and a failed model, both learned using two hidden layers and sigmoid activations.

1. (4 marks) What do you notice about the failed model vs the successful model? Based on the plot, explain why the successful model is able to classify accurately, but the failed model cannot.

2. (4 marks) Now unhide the code snippets that create the successful and failed model. There is only one code difference between them: what is that code difference? Explain why this one difference could result in failure in one case and success in another.

(d) [7 MARKS] In the section called **Epoch by Epoch**, we visualize the decision surface (in terms of each hidden layer) after each epoch. When you step through, you will notice the heatmap and lines for the data generating functions change. Based only on this plot, at which epoch do you expect the classifier to start accurately being able to classify between the two classes? Save a plot and justify your answer using it.

Question 4. [25 MARKS]

In this question, you will be implementing the necessary details for a VAE. We will use these implementations to reconstruct the MNIST data set.

1. (5 marks) Complete the `reconstruct` function

2. (10 marks) Complete the `elbo` loss (2 marks), including the KL-divergence (4 marks) and binary cross entropy (4 marks) used to compute it.

3. (10 marks) Complete the `train_mnist_vae` function

Question 5. [30 MARKS]

In this question, you will be looking at what changes occur in a VAE when we condition on the class of the input. First, we will derive the conditional ELBO (CELBO) objective, and then you will implement the necessary details in the code.

(a) [8 MARKS]

In a conditional VAE (CVAE), we condition the distributions we want to learn on the label of the data's class (see the preamble for the **Conditional VAE** section in the julia notebook for more information). Show that

$$-\ln p(\mathbf{x}|y,\mathbf{W}) + D_{KL}(q(\cdot|\mathbf{x},y)||p(\cdot|\mathbf{x},y,\mathbf{W})) = D_{KL}(q(\cdot|\mathbf{x},y)||p) - \mathbb{E}_{\mathbf{h}\sim q(\cdot|\mathbf{x},y)}[\ln(p(\mathbf{x}|\mathbf{h},y,\mathbf{W}))] \quad (2)$$

where $\mathbf{x}$ are the inputs, y is the class label of $\mathbf{x}$, $\mathbf{W}$ are the weights, $q(\mathbf{h}|\mathbf{x},y)$ is the *variational* distribution (learned by the encoder), and $p(\mathbf{x}|\mathbf{h},y,\mathbf{W})$ is the data distribution (learned by the decoder).

(b) [2 MARKS] Given the formula in Equation (2), what is the conditional evidence lower bound objective (CELBO)?

(c) [15 MARKS]

Complete the implementation for the **CVAE**.

1. (5 marks) Complete the `reconstruct` function
2. (5 marks) Complete the `celbo` loss
3. (5 marks) Complete the `train_mnist_cvae` function

(d) [5 MARKS]

In the final section, we have several visualizations. After the reconstruction visualization, we use the two models to generate data without having new data to reconstruct from. Use the visualization for the CVAE and change the class we are generating. Which MNIST digits are the hardest to generate in your opinion? Why? Save some plots and use them to justify your answer.

Note that many answers are possible here, there is no one digit that is the hardest to generate. The main point of this question is to get you to look at the sampled images and report outcomes.

Homework policies:

Your assignment should be submitted on eClass as a single pdf document and a zip file containing: the code (a .jl file), a .html file of the pluto notebook with all the cells run. The answers must be written legibly and scanned or must be typed (e.g., Latex). All code should be turned in when you submit your assignment. This means submitting the completed Pluto notebook, where you took the Pluto notebook with `todos` and completed them with your implementation. You are not allowed to change any of the imports in the notebook.

Because assignments are more for learning, and less for evaluation, grading will be based on coarse bins. **The grading is atypical**. For grades between (1) 80-100, we round-up to 100; (2) 60-80, we round-up to 80; (3) 40-60, we round-up to 60; and (4) **0-40, we round down to 0**. The last bin is to discourage quickly throwing together some answers to get some marks. The goal for the assignments is to help you learn the material, and completing less than 50% of the assignment is ineffective for learning.

We will not accept late assignments. Plan for this and aim to submit at least a day early. If you know you will have a problem submitting by the deadline, due to a personal issue that arises, please contact the instructor as early as possible to make a plan. If you have an emergency that prevents submission near the deadline, please contact the instructor right away. Retroactive reasons for delays are much harder to deal with in a fair way.

All assignments are individual. All the sources used for the problem solution must be acknowledged, e.g. web sites, books, research papers, personal communication with people, etc. Academic honesty is taken seriously; for detailed information see the University of Alberta Code of Student Behaviour.

Good luck!