Stanford CS 229, Winter 2024 Midterm Solutions

The midterm is open-book (no internet), closed-collaboration, and subject to the Honor Code. Please write your name and SID on the first page below and **write your name initials on the top of every page**. **Please take a look at the instructions below before you start.** There are **26 pages** in total. You **may**:

- You have an additional 15 minutes (totaling 3 hours and 15 minutes) to scan and upload your answers to Gradescope. **Do not use this time to work on the exam.**
- You can print and handwrite or write directly onto the exam using your digital device at your convenience. You can also write your answer by hand on blank pages, take photos, copy your photos to the exam booklet, and then upload the complete booklet to Gradscope. **In all cases, please make sure your answers are on the blank exam booklet instead of completely blank pages. Please upload the exact number of pages (including the last blank pages) given in the exam booklet.** You don't need to select pages.
- Access any materials or resources, including the course notes and reference material you may have downloaded or printed previously. You can use electronic devices, but **cannot connect to the internet**.
- Cite without proof any result from lecture slides, homework, or lecture notes, unless otherwise stated.
- If you encounter a question that needs clarification, please write your assumptions at the beginning of your solution for the teaching staff to consider when grading. For example, "I wasn't sure if this question was asking X or Y. I assumed X and answered the question accordingly."

You **may not**:

- Talk to, consult, or collaborate with anyone about the exam, and you may not consult any human or artificial intelligence about the exam problems. Any such collaboration is a violation of the Honor Code.
- **Access any resources from the internet during the midterm.** Note: you are allowed to download materials from the internet ahead of the midterm and access them offline during the midterm.
- You are not allowed to ask questions on Ed during the exam and we will not answer any clarification questions.

Good luck! We know you've been working hard, and we all want you to succeed!

Name of Student: _____

SUNetID: _____@stanford.edu

Exam Duration: 3 hours

Question	Points
1 True or False	/12
2 Multiple Choice	/12
3 Decision Tree Construction	/14
4 Binomial GLM	/15
5 Tikhonov Regression	/10
6 Quadratic Loss Descent	/14
7 Multi-task Networks	/23
Total	/100

The Stanford University Honor Code:

I attest that I have not given or received aid in this examination, and that I have done my share and taken an active part in seeing to it that others as well as myself uphold the spirit and letter of the Honor Code. In addition, I have not accessed any online resources during the exam.

Signed: _____

1. **[12 points] True or False**

   For each statement, just indicate whether it is TRUE (always completely correct) or FALSE (at least some aspect is sometimes wrong). Make sure to fully fill out the checkboxes, i.e. ■. Answers which do not fully fill out the checkboxes, i.e. ☑ or ☒ may be marked incorrect. **No need to provide an explanation.**

   (a) [2 points] Consider the ordinary least squares solution θ^* in $X^TX\theta^* = X^Ty$, where $X \in \mathbb{R}^{m\times d}$ consists of m examples with d features each. So long as $m \gg d$, i.e. the number of examples is way more than the number of features, there exists a unique least squares solution.

   ☐ True

   ☐ False

   Answer: False. Specifically, least squares requires that the columns are unique (i.e. no repeated features, features are not linear combinations of each other, etc.)

   (b) [2 points] Consider a classification problem with the provided training dataset:

x_1	x_2	y
-1	2	0
0	3	0
1	4	0
-1	5	1
0	6	1
1	7	1

   We perform logistic regression on the dataset, using the hypothesis form $h_\theta(x) = g(\theta^Tx) = \frac{1}{1+e^{-\theta^Tx}}$, where $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ and $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$. Using this parameterization, we can achieve 100% accuracy on the training dataset.

   ☐ True

   ☐ False

   Answer: False; the data appears linearly separable but without the intercept term, cannot express it.

   (c) [2 points] Given a unique global optimum, gradient ascent applied to the log-likelihood function of any Generalized Linear Model (GLM) is guaranteed to converge to this global optimum, assuming the step size is sufficiently small.

   ☐ True

☐ False

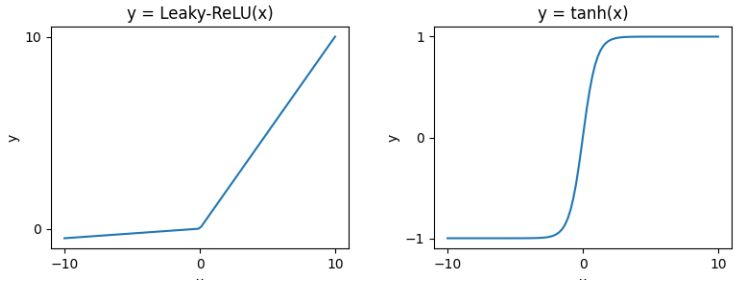
Answer: True. This is due to the concavity of GLM log-likelihood function (as shown in PSet 2). Thus, gradient ascent will converge to the unique global optimum (under mild assumptions on the step size).

(d) [2 points] The "vanishing gradient" problem is an issue in deeper neural networks: earlier layers in the network may receive near-zero gradients, due to the chain rule in backpropagation.

Consider the *Leaky ReLU* function, which is defined as follows for some $\alpha \ll 1$:

$$\text{Leaky-ReLU(x)} = \begin{cases} \alpha x & x \leq 0 \\ x & x \geq 0 \end{cases}$$

An 18 layer network, where each layer takes the form $h^{t+1} = \sigma(W_{t+1}h^T + b_{t+1})$, using *tanh* activation functions is less likely to suffer from the vanishing gradient issue than one using the *Leaky-ReLU* activation (both pictured below).



☐ True

☐ False

Answer: False; tanh tends to "saturate", as the gradients when it is near -1 or 1 approach 0. One advantage of ReLU is that it has a constant gradient if positive.

(e) [2 points] As the value of λ in LASSO regression increases, the optimization process is more likely to result in a solution with a smaller number of features having non-zero coefficients. This means it has the capacity to reduce certain coefficients to zero exactly.

☐ True

☐ False

Answer: True.

(f) [2 points] The set of multi-layer neural networks with the identity activation function is strictly more expressive than the set of linear classifiers because they (multi-layer neural networks) are deep and can capture complex patterns and relationships within data.

☐ True

☐ False

Answer: False. They are no different than linear models with identity activation functions.

2. **[12 points] Multiple Choice**

   For each question, choose all of the correct answers (there may be one or more correct answers). Make sure to fully fill out the checkboxes, i.e. ■. Answers which do not fully fill out the checkboxes, i.e. ☑ or ☒ may be marked incorrect.

   No need to provide an explanation. There will be partial credits.

   (a) [3 points] Suppose we train a logistic-regression model (with *bounded* parameters θ) to convergence to perform binary classification. We use the cross-entropy loss function, where $\sigma(x) = \frac{1}{1+\exp(-x)}$:

   $$\mathcal{L}(\theta, (x^{(i)}, y^{(i)})_{i=1}^{N}) = -\sum_{i=1}^{N} \left((y^{(i)}) \log \sigma(\theta^\top x^{(i)}) + (1 - y^{(i)}) \log \sigma(\theta^\top x^{(i)}) \right).$$

   Which **one or more** of the following changes are guaranteed to not increase the **training** loss of the model?

   ☐ Adding a new training datapoint $(x^{(N+1)}, y^{(N+1)})$ and evaluating $\mathcal{L}(\theta, (x^{(i)}, y^{(i)})_{i=1}^{N+1})$

   ☐ Appending a new feature to each $x^{(i)}$ and minimizing loss of new model with an extra parameter.

   ☐ Removing the final datapoint $(x^{(N)}, y^{(N)})$ and evaluate $\mathcal{L}(\theta, (x^{(i)}, y^{(i)})_{i=1}^{N-1})$.

   ☐ Adding L2 regularization to the loss function.

   Answer: (2)*, (3)

   (1) will increase the training loss since cross-entropy loss is non-negative.

   (2) **We will accept both true and false**. Assuming that optimization is deterministic (i.e. will always converge to the global optimum), then at worst, this will keep the training loss constant if the features are superfluous i.e. zero in all dimensions. However, if optimization is stochastic, the optimizer may settle in a local optima.

   (3) since the parameters are bounded, the cross-entropy error of each data point is > 0. As a result, the training error will strictly decrease if we remove a data point from the set.

   (4) L2 regularization is non-negative so will increase the training loss.

(b) [3 points] Consider an example in which the data is generated by the following linear random process. For all $n \in \mathbb{Z}_{++}$,

$$X_n \sim \mathcal{N}(0, I_d), \quad Y_n = \theta^\top X_n + Z_n, \quad Z_n \sim \mathcal{N}(0, \sigma^2),$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance / covariance matrix $\sigma^2 > 0$, I_d denotes the identity matrix in d dimensions, and $\theta \in \mathbb{R}^d$ is *unknown*.

In machine learning, we are interested in the expected out-of-sample (test) error of our learned model. For all n, let $\hat{\theta}_n$ denote the ordinary least squares estimate of θ produced by a dataset of n points:$(X_i, Y_i)_{i=0}^{n-1}$.

Question: Which **one or more** of the following expected out-of-sample errors converge to 0 as $n \to \infty$?

☐ **(Expected Squared Error)**

$$\lim_{n\to\infty} \mathbb{E}\left[\left(Y_n - \hat{\theta}_n^\top X_n\right)^2\right]$$

☐ **(Expected Absolute Error)**

$$\lim_{n\to\infty} \mathbb{E}\left[\left|\theta^\top X_n - \hat{\theta}_n^\top X_n\right|\right]$$

☐ **(Expected Log Loss)**

$$\lim_{n\to\infty} \mathbb{E}\left[-\log\left(\sqrt{2\pi\sigma^2}\exp\left\{-\frac{(Y_n - \hat{\theta}_n^\top X_n)^2}{2\sigma^2}\right\}\right)\right]$$

☐ **(Expected Distance to θ)**

$$\lim_{n\to\infty} \mathbb{E}\left[\|\theta - \hat{\theta}_n\|_2^2\right]$$

Answer: (2), (4)

(1) Y_n has irreducible noise from Z_n so the limit will be σ^2 and not 0.

(2) Note that if the data is produced by a linear process, the OLS solution will converge to the true linear parameters as $n \to \infty$ despite the independent 0-mean noise. Therefore, by continuity, the absolute error goes to 0.

(3) This limit converges to $-\frac{1}{2}\log(2\pi e\sigma^2) \neq 0$.

(4) If the data is produced by a linear process, the OLS solution will converge to the true linear parameters as $n \to \infty$ despite the independent 0-mean noise.

(c) [3 points] Select **all** of the following that are true statements pertaining to logistic regression.

☐ When the number of datapoints is less than the number of parameters, the minimum value of the logistic loss is 0.

☐ If the data are linearly separable, logistic regression will perfectly separate the data.

☐ A logistic regression model with bounded (i.e. finite) parameter values can assign probability exactly 0 to a class.

☐ The logistic loss function is bounded above by some constant.

Answer: (2)

(1) Consider 2 data points which have identical features but different labels. The minimum value of logistic loss is not 0 in such an instance.

(2) This fact is true and the weights will tend towards ∞ in such a case.

(3) In order to assign probability 0 to a class, the parameter values must have norm ∞. This is not possible if the parameter values are bounded.

(4) The logistic loss function is unbounded when θ or x are unbounded.

(d) [3 points] Consider fitting a neural network to the MNIST dataset. You decide to first fit a 3 layer MLP, and witness low training error and high validation error. Which **one or more** of the following is likely to reduce validation error?

☐ Increasing the number of layers of the MLP to 4.

☐ Adding an appropriately weighted L2 regularization term to the loss function.

☐ Replacing the first layer of the MLP with an RNN.

☐ Replacing the first layer of the MLP with a convolutional layer.

Answer: The neural network is likely overfit to the training dataset. So we want to reduce the variance of the estimator, i.e. decrease network capacity, or choose an estimator better suited for the task. (2,3,4)

3. **[14 points] Decision Tree Construction**

You are the Chief Data Scientist for a public library system. You have collected data on some visitors to your library's website, including whether or not they live in the library's service area, the time they have spent on the library's website, whether or not they attended a library event, and whether or not they signed up for a library membership. The table below is your training set. You want to build a decision tree to predict whether future visitors to the website will sign up for a library membership. The rightmost column of the table is the label you wish to predict. The other columns are the features, except for the visitor number, which you should use solely to illustrate which visitors are in which nodes of your decision tree.

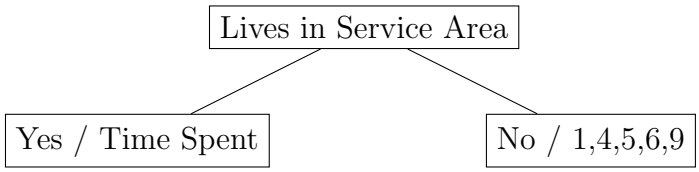
Visitor	Lives in Service Area	Time Spent (min)	Event Attendance	Library Membership
1	No	9	Yes	Yes
2	Yes	16	Yes	Yes
3	Yes	9	Yes	No
4	No	16	No	No
5	No	9	Yes	Yes
6	No	16	Yes	Yes
7	Yes	9	No	No
8	Yes	9	No	No
9	No	9	No	No

(a) [4 points] Instead of using misclassification as the cost function from the lecture, we will introduce the weighted sum of entropies as the cost function. The formula for entropy, given a set S with class labels, is: $\text{Entropy}(S) = -\sum_{i=1}^{n} p_i \log_2(p_i)$. where n is the number of classes (in this problem $n = 2$), p_i is the proportion of examples in S that belong to class i, and $0 \log 0 = 0$. The entropy of a dataset decreases when the proportion of samples belonging to a single class increases, reaching zero when all samples belong to one class (pure leaf). Finally, the cost function is $L(S, A) = \sum_{j=1}^{m} \left(\frac{|S_j|}{|S|}\right) \text{Entropy}(S_j)$ where $|S_j|$ is the number of instances in subset S_j (in this problem $m = 2$) after splitting S by feature A.

1). **Which feature** should you split on **at the root** of the decision tree to minimize the weighted sum of entropies? 2). Write a numerical expression (no variables) for that cost function of the best split. (Your expression can include logarithms and fractions. We define $0 \log 0 = 0$). **Answer:** Split on "Event Attendance". The cost function is

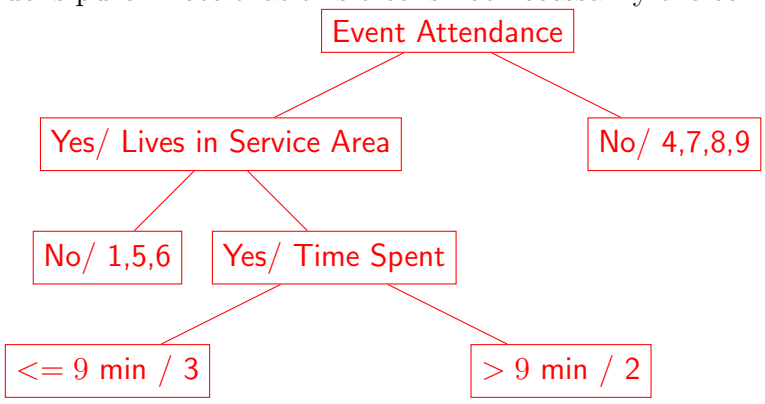
$$-\frac{4}{9}(0) - \frac{5}{9}(\frac{4}{5}log_2\frac{4}{5} + \frac{1}{5}log_2\frac{1}{5}) = -\frac{4}{9}log_2\frac{4}{5} - \frac{1}{9}log_2\frac{1}{5}$$

(b) [8 points] **Draw the decision tree that minimizes the weighted sum of entropies at each split**. The leaves of your tree should be drawn; write inside each leaf the splitting value and training points (indicated by the visitor number) it stores. In the internal tree nodes, write the **splitting features and splitting values**. There is no need to write any entropy. Split until all leaves are pure (i.e. all the data points are from the same class). An example can be:



where "Yes"/"No" indicates whether they live in the service area, "Time Spent" indicates the second split feature, and "1,4,5,6,9" indicates the visitor number since this node is pure. Note that this tree is not necessarily the correct solution.

Answer:



(c) [2 points] For the decision tree you constructed (not the example tree provided) in part (b), suppose we limit the tree to a depth of two (where the root counts as depth zero). That is, the root node can have grandchildren but no great-grandchildren. What will be the training error rate (misclassified training points divided by all training points) of the impure tree? **Answer:** Only visitor 2 or 3 is misclassified. The training error rate is $\frac{1}{9}$.

4. **[15 points] Binomial GLM**

   In this problem, we study a GLM under the binomial distribution. Recall that a distribution belongs to the exponential family if its probability density/mass function can be expressed as:

   $$p(y;\eta) = b(y)\exp\left(\eta^\top T(y) - a(\eta)\right),$$

   where η is the natural parameter, $T(y)$ is the sufficient statistic and $a(\eta)$ is the log-partition function.

   For a fixed positive integer n, the binomial random variable takes values in the set $\{0, 1, \ldots, n\}$. Its probability mass function is provided below:

   $$p(y;\phi) = \binom{n}{y}\phi^y(1-\phi)^{n-y}.$$

   where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, and $0 < \phi < 1$ is the *success probability* of an independent trial. The binomial distribution describes the likelihood that y out of the n trials are successful given that each trial is independent with success probability ϕ.

   (a) [6 points] Demonstrate that the binomial distribution is part of the exponential family. Provide the appropriate mathematical expressions for $b(y), \eta, a(\eta)$. Note that η should be a *scalar*, and $a(\eta)$ should be expressed in terms of η. Let $T(y) = y$. **You don't need to show any intermediate steps.**

   $b(y) =$ [ ]

   $\eta =$ [ ]

   $a(\eta) =$ [ ]

Answer:

$$b(y) = \boxed{\binom{n}{y}}$$
$$T(y) = \boxed{y}$$
$$\eta = \boxed{\log \frac{\phi}{1-\phi}}$$
$$a(\eta) = \boxed{n \log(1 + e^{\eta})}$$

(b) [3 points] Consider performing regression using a GLM model with a binomial response variable. What is the canonical response function for the family ($\mathbb{E}[T(y); \eta]$)? **Write intermediate steps to get full credit.** **Answer:**

$$\begin{aligned}\mathbb{E}[T(y); \eta] &= \mathbb{E}[y; \eta] \\ &= n\psi \\ &= \frac{ne^{\eta}}{1 + e^{\eta}}.\end{aligned}$$

(c) [6 points] Let $(x^{(1)} \in \mathbb{R}^d, y^{(1)} \in \{0, 1, \ldots, n\})$ represent a training point. To run regression on this point, we resort to a GLM formulation with learnable parameters $\theta \in \mathbb{R}^d$, and update θ via stochastic gradient ascent. The objective is the log-likelihood of the data under a binomial response variable. Derive an expression for θ_1 (the result of the first gradient update) in terms of the initial θ_0 and learning rate α.

Hint: consider the gradient of the log-likelihood, i.e. $\log(p(y; \eta))$.

Answer:

$$
\begin{aligned}
\theta_1 &= \theta_0 + \alpha \frac{\partial \ell(\theta_0)}{\partial \theta} \\
&= \theta_0 + \alpha \frac{\partial \log \binom{n}{y^{(1)}} e^{y^{(1)}\eta - n\log(1+e^{\eta})}}{\partial \theta} \\
&= \theta_0 + \alpha \frac{\partial \log \binom{n}{y^{(1)}} e^{y^{(1)}\theta_0^\top x^{(1)} - n\log(1+e^{\theta_0^\top x^{(1)}})}}{\partial \theta} \\
&= \theta_0 + \alpha \frac{\partial \log e^{y^{(1)}\theta_0^\top x^{(1)} - n\log(1+e^{\theta_0^\top x^{(1)}})}}{\partial \theta} \\
&= \theta_0 + \alpha \frac{\partial \; y^{(1)}\theta_0^\top x^{(1)} - n\log(1+e^{\theta_0^\top x^{(1)}})}{\partial \theta} \\
&= \theta_0 + \alpha \left(x^{(1)} y^{(1)} - n \frac{e^{\theta_0^\top x^{(1)}}}{1+e^{\theta_0^\top x^{(1)}}} x^{(1)} \right)
\end{aligned}
$$

5. **[10 points] Tikhonov Regression**

   To reduce the variance of the least squares solution, we can add a regularization term to the objective function. In this problem, we will draw connections between different regularization methods.

   For all parts of this question, we are given a design matrix $X \in \mathbb{R}^{m \times n}$ and a vector of labels $y \in \mathbb{R}^m$. The goal is to fit a vector $\theta \in \mathbb{R}^n$ such that the corresponding cost function is minimized.

   (a) [6 points] The *Tikhonov regularized* least-squares cost function is as follows:

   $$J_T(\theta) = \|X\theta - y\|_2^2 + \|\Gamma\theta\|_2^2$$

   where Γ is a symmetric matrix. Derive a closed form solution for θ_T^*, the minimizer of J_T. **Answer:** Proceeding with matrix calc:

   $$\begin{aligned}
   J_T(\theta) &= \|X\theta - y\|_2^2 + \|\Gamma\theta\|_2^2 \\
   &= (X\theta - y)^T(X\theta - y) + (\Gamma\theta)^T(\Gamma\theta) \\
   &= \theta^T X^T X\theta - y^T X\theta - \theta^T X^T y + y^T y + \theta^T \Gamma^T \Gamma\theta \\
   &= \theta^T X^T X\theta - y^T X\theta - \theta^T X^T y + y^T y + \theta^T \Gamma^T \Gamma\theta \\
   &= \theta^T (X^T X + \Gamma^T \Gamma)\theta - 2y^T X\theta + C \\
   \nabla_\theta J &= 2(X^T X + \Gamma^T \Gamma)\theta - 2X^T y \\
   0 &= 2(X^T X + \Gamma^T \Gamma)\theta - 2X^T y \\
   \theta^* &= (X^T X + \Gamma^T \Gamma)^{-1} X^T y
   \end{aligned}$$

(b) [2 points] What are the necessary conditions on X and Γ for θ_T^* to be unique and a minimizer? **Answer:** Necessary conditions: $X^TX + \Gamma^T\Gamma$ must be PD; either of the matrices must be PD, and the other can be PSD.

(c) [2 points] Now, consider the *ridge regularized* least-squares cost function:

$$J_R(\theta) = \|X\theta - y\|_2^2 + \lambda\|\theta\|_2^2$$

1) For what value of Γ is Tikhonov regression equivalent to ridge regression? 2) Using this, provide the closed form solution for θ_R^*, the minimizer of J_R. **Answer:** When $\Gamma = \sqrt{\lambda}I$.

6. **[14 points] Quadratic Loss Descent**

   Consider an arbitrary loss function, $J(x) = \frac{1}{2}x^T Ax + bx + c$, where $A \in \mathbb{R}^{d \times d}$ is an arbitrary positive definite matrix with eigenvalues $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$, and $x \in \mathbb{R}^d$.

   (a) [2 points] Calculate the gradient with respect to x, $\nabla_x J$. Then, write the update rule for gradient descent on J, assuming a learning rate of α, in terms of the iterates $x^{(k)}$ and $x^{(k+1)}$. **Answer:** The gradient is $Ax + b$.

   Using this, the gradient descent update is $x^{(k+1)} = x^{(k)} - \alpha(Ax^{(k)} + b) = (I - \alpha A)x^{(k)} - \alpha b$.

(b) [5 points] Using the gradient update rule you derived in Part 1 of this question, express the current iteration $x^{(k)}$ in terms of the initial vector $x^{(0)}$. **Answer:** We unroll a few iterations:

$$
\begin{aligned}
x^{(1)} &= (I - \alpha A)x^{(0)} - \alpha b \\
x^{(2)} &= (I - \alpha A)x^{(1)} - \alpha b \\
&= (I - \alpha A)((I - \alpha A)x^{(0)} - \alpha b) - \alpha b \\
&= (I - \alpha A)^2 x^{(0)} - (I - \alpha A)\alpha b - \alpha b \\
x^{(3)} &= (I - \alpha A)x^{(2)} - \alpha b \\
&= (I - \alpha A)((I - \alpha A)^2 x^{(0)} - (I - \alpha A)\alpha b - \alpha b) - \alpha b \\
&= (I - \alpha A)^3 x^{(0)} - (I - \alpha A)^2 \alpha b - (I - \alpha A)\alpha b - \alpha b \\
x^{(k)} &= (I - \alpha A)^k x^{(0)} - \sum_{i=0}^{k-1} (I - \alpha A)^i \alpha b
\end{aligned}
$$

(c) [7 points] Consider the error between the current iteration $x^{(k)}$ and the optimizer x^*. If for all eigenvalues $\sigma_i, \sigma_i \in [0, \frac{2}{\alpha}]$, prove that gradient descent converges to x^* when run for infinite iterations:

$$\lim_{k\to\infty} x^{(k)} \to x^*$$

You may use without proof the following: given a diagonal matrix A such that $0 \leq a_{ii} \leq 1$, $\lim_{k\to\infty} \sum_{i=0}^{k} (I - A)^i = A^{-1}$.

Note: this problem is challenging, so it's advisable to tackle other problems first if you are stuck.

Answer: Because the matrix is PD, we can derive the closed form optimizer to be $x^* = -A^{-1}b$. Now, consider the limit of the error:

$$\begin{aligned}\lim_{k\to\infty} x^k &= \lim_{k\to\infty} (I - \alpha A)^k x^{(0)} - \sum_{i=0}^{k-1} (I - \alpha A)^i \alpha b \\ &= \lim_{k\to\infty} (I - \alpha A)^k x^{(0)} - \lim_{k\to\infty} \sum_{i=0}^{k-1} (I - \alpha A)^i \alpha b\end{aligned}$$

Recall that we know that all eigenvalues must be less than $\frac{1}{\alpha}$. Under these conditions, $(I - \alpha A)$ has eigenvalues bounded by $[-1, 1]$, and the first term approaches 0.

For the second term, note that $(I - \alpha A)$ may not necessarily be a diagonal matrix. Thus, we consider the spectral decomposition of $(I - \alpha A)$. Note that $(I - \alpha A)$ has eigenvalues $\lambda = (1 - \alpha\lambda_A)$. Thus, we can write the spectral decomposition as $(I - \alpha A) = V(I - \alpha\Lambda)V^{-1}$, and rewrite the second term:

$$\begin{aligned}\lim_{k\to\infty} \sum_{i=0}^{k-1} (I - \alpha A)^i \alpha b &= \lim_{k\to\infty} \sum_{i=0}^{k-1} V(I - \alpha\Lambda)^i V^{-1} \alpha b \\ &= \lim_{k\to\infty} V(\sum_{i=0}^{k-1} (I - \alpha\Lambda)^i) V^{-1} \alpha b\end{aligned}$$

Using the provided theorem, we know that we can rewrite the limit term as $\frac{1}{\alpha} A^{-1}$.

$$\lim_{k\to\infty} (\sum_{i=0}^{k-1} (I - \alpha\Lambda)^i) = \frac{1}{\alpha}\Lambda^{-1}$$

Lastly:

$$\begin{aligned}\lim_{k\to\infty} V(\sum_{i=0}^{k-1} (I - \alpha\Lambda)^i) V^{-1} \alpha b &= \frac{1}{\alpha} V(\Lambda)^{-1} V^{-1} \alpha b \\ &= A^{-1} b\end{aligned}$$

Thus, x^k approaches $-A^{-1}b$, which is exactly the form of the closed form optimizer. Thus, gradient descent converges to x^*.

7. **[23 points] Multi-task Networks**

A popular idea in modern machine learning is to perform *multi-task* training – that is, to use the same network for multiple different prediction tasks. At a high level, it is believed that training on a similar task may help improve performance on a different task. For example, in natural language, training on the task of predicting the next character in a sentence may improve performance on a validation dataset for predicting if one sentence follows another.

You have been contracted by a startup to build a predictor that, given a set of image features $x \in \mathbb{R}^d$, outputs both a binary classification if there is an animal in the image y_1, and the height of the animal y_2, where $y_1 \in \{0, 1\}$ and $y_2 \in \mathbb{R}$; you may assume that if $y_1 = 0, y_2 = -1$.

To generate predictions, you use the following neural network architecture:

$$\begin{aligned} z_1 &= W_1^T x + b_1 \\ h &= ReLU(z_1) \\ z_{21} &= W_{21}^T h + b_{21} \\ z_{22} &= W_{22}^T h + b_{22} \\ L_1 &= \text{Cross-Entropy}(y_1, \sigma(z_{21})) \\ &= y_1 \log(\sigma(z_{21})) + (1 - y_1) \log(1 - \sigma(z_{21})) \\ L_2 &= \text{MSE}(y_2, z_{22}) \\ &= \|y_2 - z_{22}\|_2^2 \\ L &= L_1 + L_2 \end{aligned}$$

where $W_1 \in \mathbb{R}^{d \times h}$, $b_1 \in \mathbb{R}^h, W_{21} \in \mathbb{R}^h, W_{22} \in \mathbb{R}^h$, $b_{21} \in \mathbb{R}$, and $b_{22} \in \mathbb{R}$. $\sigma(x) = \frac{1}{1+\exp(-x)}$ denotes the sigmoid function. You perform backpropogation using L, i.e. the sum of the two losses.

In this problem, you will calculate the gradients necessary for back-propogation through this network. Whenever necessary, you may refer to the gradient of CrossEntropy$(y, \hat{y})$ as $\frac{\partial \text{CE}}{\partial \hat{y}}$, the gradient of MSE$(y, \hat{y})$ as $\frac{\partial \text{MSE}}{\partial \hat{y}}$, and the gradient of *ReLU(x)* as $\frac{\partial ReLU}{\partial \hat{x}}$.

For all of the following questions, whenever applicable, please write the answer (1) as an expression using the chain rule in terms of any previously derived gradients, and (2) analytically in closed form in terms of the variables given (i.e. x, h, W, etc., including $\frac{\partial \text{CE}}{\partial \hat{y}}$, $\frac{\partial \text{MSE}}{\partial \hat{y}}$, and $\frac{\partial \text{ReLU}}{\partial x}$).

Hint: keep in mind the expected shapes of the gradients.

(a) [5 points] **Calculate $\frac{\partial L_1}{\partial W_{21}}$ and $\frac{\partial L_1}{\partial b_{21}}$.** The network architecture is copied here for convenience:

$$\begin{aligned}
z_1 &= W_1^T x + b_1 \\
h &= ReLU(z_1) \\
z_{21} &= W_{21}^T h + b_{21} \\
z_{22} &= W_{22}^T h + b_{22} \\
L_1 &= \text{Cross-Entropy}(y_1, \sigma(z_{21})) \\
&= y_1 \log(\sigma(z_{21})) + (1 - y_1) \log(1 - \sigma(z_{21})) \\
L_2 &= \text{MSE}(y_2, z_{22}) \\
&= \|y_2 - z_{22}\|_2^2 \\
L &= L_1 + L_2
\end{aligned}$$

Answer:

$$\begin{aligned}
\frac{\partial L_1}{\partial W_{21}} &= \frac{\partial L_1}{\partial \sigma(z_{21})} \frac{\partial \sigma(z_{21})}{\partial z_{21}} \frac{\partial z_{21}}{\partial W_{21}} \\
&= \frac{\partial \text{CE}}{\partial \sigma(z_{21})} (\sigma(z_{21})(1 - \sigma(z_{21}))h
\end{aligned}$$

Note that h is shape h and the other two terms are scalar.

$$\begin{aligned}
\frac{\partial L_1}{\partial b_{21}} &= \frac{\partial L_1}{\partial \sigma(z_{21})} \frac{\partial \sigma(z_{21})}{\partial z_{21}} \frac{\partial z_{21}}{\partial b_{21}} \\
&= \frac{\partial \text{CE}}{\partial \sigma(z_{21})} (\sigma(z_{21})(1 - \sigma(z_{21}))
\end{aligned}$$

Note that this is scalar.

(b) [5 points] **Calculate $\frac{\partial L_2}{\partial b_{22}}$ and $\frac{\partial L_2}{\partial W_{22}}$.** The network architecture is copied here for convenience:

$$
\begin{aligned}
z_1 &= W_1^T x + b_1 \\
h &= ReLU(z_1) \\
z_{21} &= W_{21}^T h + b_{21} \\
z_{22} &= W_{22}^T h + b_{22} \\
L_1 &= \text{Cross-Entropy}(y_1, \sigma(z_{21})) \\
&= y_1 \log(\sigma(z_{21})) + (1 - y_1) \log(1 - \sigma(z_{21})) \\
L_2 &= \text{MSE}(y_2, z_{22}) \\
&= \|y_2 - z_{22}\|_2^2 \\
L &= L_1 + L_2
\end{aligned}
$$

Answer:

$$
\begin{aligned}
\frac{\partial L_2}{\partial W_{22}} &= \frac{\partial L_2}{\partial z_{22}} \frac{\partial z_{22}}{\partial W_{22}} \\
&= \frac{\partial \text{MSE}}{\partial z_{22}} h
\end{aligned}
$$

Note that h is shape h and the other term is scalar.

$$
\begin{aligned}
\frac{\partial L_2}{\partial b_{22}} &= \frac{\partial L_2}{\partial z_{22}} \frac{\partial z_{22}}{\partial b_{22}} \\
&= \frac{\partial \text{MSE}}{\partial z_{22}}
\end{aligned}
$$

Note that this is scalar.

(c) [5 points] **Calculate** $\frac{\partial L}{\partial h}$. The network architecture is copied here for convenience:

$$\begin{aligned}
z_1 &= W_1^T x + b_1 \\
h &= ReLU(z_1) \\
z_{21} &= W_{21}^T h + b_{21} \\
z_{22} &= W_{22}^T h + b_{22} \\
L_1 &= \text{Cross-Entropy}(y_1, \sigma(z_{21})) \\
&= y_1 \log(\sigma(z_{21})) + (1 - y_1) \log(1 - \sigma(z_{21})) \\
L_2 &= \text{MSE}(y_2, z_{22}) \\
&= \|y_2 - z_{22}\|_2^2 \\
L &= L_1 + L_2
\end{aligned}$$

Answer:

$$\begin{aligned}
\frac{\partial L}{\partial h} &= \frac{\partial L}{\partial L_1}\frac{\partial L_1}{\partial z_{21}}\frac{\partial z_{21}}{\partial h} + \frac{\partial L}{\partial L_2}\frac{\partial L_2}{\partial \sigma(z_{22})}\frac{\partial \sigma(z_{22})}{\partial z_{22}}\frac{\partial z_{22}}{\partial h} \\
&= \frac{\partial \text{CE}}{\partial \sigma(z_{21})}(\sigma(z_{21})(1 - \sigma(z_{21}))W_{21} + \frac{\partial \text{MSE}}{\partial z_{22}}W_{22}
\end{aligned}$$

Note that this is of shape h.

(d) [5 points] **Calculate $\frac{\partial L}{\partial W_1}$ and $\frac{\partial L}{\partial b_1}$.** The network architecture is copied here for convenience:

$$\begin{aligned}
z_1 &= W_1^T x + b_1 \\
h &= ReLU(z_1) \\
z_{21} &= W_{21}^T h + b_{21} \\
z_{22} &= W_{22}^T h + b_{22} \\
L_1 &= \text{Cross-Entropy}(y_1, \sigma(z_{21})) \\
&= y_1 \log(\sigma(z_{21})) + (1 - y_1) \log(1 - \sigma(z_{21})) \\
L_2 &= \text{MSE}(y_2, z_{22}) \\
&= \|y_2 - z_{22}\|_2^2 \\
L &= L_1 + L_2
\end{aligned}$$

Answer:

$$\begin{aligned}
\frac{\partial L}{\partial W_1} &= \frac{\partial L}{\partial h} \frac{\partial h}{\partial z_1} \frac{\partial z_1}{\partial W_1} \\
&= x((\frac{\partial \text{CE}}{\partial \sigma(z_{21})}(\sigma(z_{21})(1 - \sigma(z_{21}))W_{21} + \frac{\partial \text{MSE}}{\partial z_{22}} W_{22}) \odot \frac{\partial ReLU}{\partial z_1})^T
\end{aligned}$$

Note that this needs to be of shape $d \times h$. We know that x is shape d, and the upstream gradient is shape h, so the outer product suffices.

$$\begin{aligned}
\frac{\partial L}{\partial b_1} &= \frac{\partial L}{\partial h} \frac{\partial h}{\partial z_1} \frac{\partial z_1}{\partial b_1} \\
&= (\frac{\partial \text{CE}}{\partial \sigma(z_{21})}(\sigma(z_{21})(1 - \sigma(z_{21}))W_{21} + \frac{\partial \text{MSE}}{\partial z_{22}} W_{22}) \odot \frac{\partial ReLU}{\partial z_1}
\end{aligned}$$

Note that this is of shape h.

(e) [3 points] Instead of solving this using a multi-task network, your colleague suggests training one network for each task. In terms of the bias-variance tradeoff, which network architecture between single-task and multi-task is more biased for the task of predicting if an animal is present? For the sake of comparison, assume that the networks have equal capacities, i.e. each network contains a separate version of h. Please **explain** your answer.

Hint: Consider parallels between ridge regression and the additional task's loss.
Answer: The multi-task network is more biased; the addditional loss serves as an inductive bias / a regularizer.

That's all! Congratulations on completing the midterm exam!
If you need more space, feel free to utilize the final empty pages. Ensure that the problems are indexed clearly.