CSE 512: Homework **Due Wednesday, October 6**

Extra practice problems, ungraded

1. *Gradients.* Compute the gradients of the following functions. Give the exact dimension of the output.

   (a) *Linear regression.* $f(x) = \frac{1}{40}\|Ax - b\|_2^2$, $A \in \mathbb{R}^{20\times 10}$

   (b) *Sigmoid.* $f(x) = \sigma(c^T x)$, $c \in \mathbb{R}^5$, $\sigma(s) = \frac{1}{1+\exp(-x)}$. Hint: Start by showing that $\sigma'(s) = \sigma(s)(1-\sigma(s))$.

2. *Convex or not convex.* Are the following sets convex or not convex? Justify your answer.

   (a) $\mathcal{S} = \mathbf{range}(A) := \{x : Az = x \text{ for some } z\}$

   (b) $\mathcal{S} = \{x : x \leq -1\} \cup \{x : x \geq 1\}$ (Read: either $x \leq -1$ or $x \geq 1$.)

3. *Am I positive semidefinite?* A symmetric matrix X is *positive semidefinite* if for all u, $u^T X u \geq 0$. For each of the following, either prove that the matrix is positive semidefinite, or find a counterexample.

   (a) $X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

   (b) $X = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}$

4. *Convex or not convex.* (**1pt, 0.125 points each.**) From lecture, we know that there are three ways of checking whether a function is convex or not.

   - For any function, we can check if it satisfies the **definition of convexity**:

   $$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y), \quad \forall x, y, \quad \forall 0 \leq \theta \leq 1.$$

   - For any differentiable function, we can check the **first-order condition**

   $$f(x) - f(y) \geq \nabla f(y)^T (x - y).$$

   - For any twice-differentiable function, we can check the **second-order condition**

   $$\nabla^2 f(x) \text{ is positive semidefinite, i.e. } u^T \nabla^2 f(x) u \geq 0 \quad \forall u.$$

   Use any of these ways to determine whether or not each function is convex. (You only need to use one of these rules per function. Pick the one you think gets you to the answer the fastest!)

   (a) $f(x) = \frac{1}{2}(x_{[1]}^2 - 2x_{[1]}x_{[2]} + x_{[2]}^2)$

   (b) $f(x) = |x|$ Hint: Again, remember the triangle inequality.

   (c) $f(x) = \log(\exp(x_{[1]}) + \exp(x_{[2]}))$

   (d) $f(x) = c^T x$

Main assignment, graded

1. *Gradient properties.* (**1 pt, 0.5 pts each.**) Prove the following two properties of gradients:

   (a) *Linearity.* If $h(x) = \alpha f(x) + \beta g(x)$, then $\nabla h(x) = \alpha \nabla f(x) + \beta \nabla g(x)$.

   (b) *Chain rule.* Show that if $g(v) = f(Av)$, then $\nabla g(v) = A^T \nabla f(Av)$.

2. *Gradients.* (**1 pts, 0.5 pts each.**) Compute the gradients of the following functions. Give the exact dimension of the output.

   (a) *Quadratic function.* $f(x) = \frac{1}{2} x^T Q x + p^T x + r$, $Q \in \mathbb{R}^{12 \times 12}$ and Q is symmetric ($Q_{[i,j]} = Q_{[j,i]}$).

   (b) *Softmax function.* $f(x) = \frac{1}{\mu} \log(\sum_{i=1}^{8} \exp(\mu x_{[i]}))$, $x \in \mathbb{R}^8$, μ is a positive scalar

3. *Convex or not convex.* (**1pt**) Are the following sets convex or not convex? Justify your answer.

   (a) (**0.3 pts**) $\mathcal{S} = \{x : \sum_i x_i = 0\}$

   (b) (**0.2 pts**) $\mathcal{S} = \{(x, y) : x^2 + y^2 = 1\}$

   (c) (**0.2 pts**) $\mathcal{S} = \{x : |x| \leq 1\}$. Hint: Remember the triangle inequality.

4. *Am I positive semidefinite?* (**1 pt, 0.5 pts each**) A symmetric matrix X is *positive semidefinite* if for all u, $u^T X u \geq 0$. For each of the following, either prove that the matrix is positive semidefinite, or find a counterexample.

   (a) $X = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 4 \\ 3 & 4 & 1 \end{bmatrix}$

   (b) $X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$

5. *Convex or not convex.* (**1pt, 0.25 points each.**)

   Use any of the three ways (definition, first order condition, or second order condition) to determine whether or not each function is convex. (You only need to use one of these rules per function. Pick the one you think gets you to the answer the fastest!)

   (a) $f(x) = 1/x$, for $x > 0$

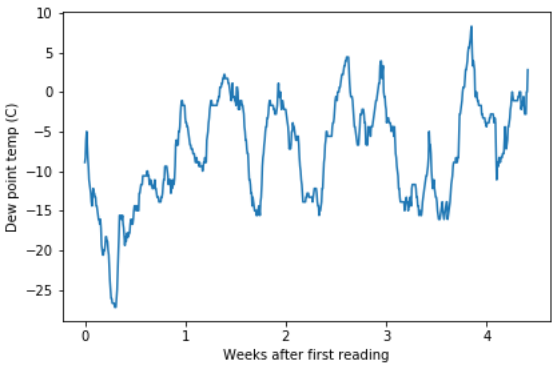
   (b) $f(x) = \|x\|_\infty$

   (c) $f(x) = x_{[3]}^3 + x_{[2]}^2 + x_{[1]}$

   (d) $f(x) = \|Ax - b\|_2^2$

6. *Polyfit via linear regression.* (**3 pts**)

   - Download weatherDewTmp.mat. Plot the data. It should look like the following



- We want to form a polynomial regression of this data. That is, given w = weeks and d = dew readings, we want to find $\theta_1, ..., \theta_p$ as the solution to

$$\underset{\theta \in \mathbb{R}^p}{\text{minimize}} \quad \frac{1}{2}\sum_{i=1}^{m}(\theta_1 + \theta_2 w_i + \theta_3 w_i^2 + \cdots + \theta_p w_i^{p-1} - d_i)^2. \tag{1}$$

Form X and y such that (1) is equivalent to the least squares problem

$$\underset{\theta \in \mathbb{R}^p}{\text{minimize}} \quad \frac{1}{2}\|X\theta - y\|_2^2. \tag{2}$$

That is, for w the vector containing the week number, and y containing the dew data, form

$$X = \begin{bmatrix} 1 & w_1 & w_1^2 & w_1^3 & \cdots & w_1^{p-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 1 & w_m & w_m^2 & w_m^3 & \cdots & w_m^{p-1} \end{bmatrix}.$$

(a) *Linear regression.*

i. Write down the normal equations for problem (2).

ii. Fill in the code to solve the normal equations for θ, and use it to build a predictor. To verify your code is running correctly, the number after `check number` should be 1.341.

iii. Implement a polynomial fit of orders $p = 1, 2, 3, 10, 100$, for the weather data provided. Include a figure that plots the original signal, overlaid with each polynomial fit. Comment on the "goodness of fit" for each value of p.

(b) *Ridge regression.* Oftentimes, it is helpful to add a *regularization term* to (2), to improve stability. In other words, we solve

$$\underset{\theta \in \mathbb{R}^p}{\text{minimize}} \quad \frac{1}{2}\|X\theta - y\|_2^2 + \frac{\rho}{2}\|\theta\|_2^2. \tag{3}$$

for some $\rho > 0$.

i. Again, write down the normal equations for (3). Your equation should be of form $A\theta = b$ for some matrix A and vector b that you specify.

ii. Write the code for solving the ridge regression problem and run it. To verify your code is running correctly, the number after `check number` should be 1.206.

iii. Using $\rho = 1.0$, plot the weather data with overlaying polynomial fits with ridge regression. Provide these plots for $p = 1, 2, 3, 10, 100$. Comment on the "goodness of fit" and the stability of the fit, and also compare with the plots generated without using the extra penalty term.

(c) *Conditioning.*

i. An *unconstrained quadratic problem* is any problem that can be written as

$$\underset{\theta}{\text{minimize}} \quad \frac{1}{2}\theta^T Q\theta + c^T\theta + r \tag{4}$$

for some symmetric positive semidefinite matrix Q, and some vector c and some scalar r. Show that the ridge regression problem (3) is an unconstrained quadratic problem by writing down Q, c, and r in terms of X and y such that (4) is equivalent to (3). Show that the Q you picked is positive semidefinite.

ii. In your code, write a function that takes in X and y, constructs Q as specified in the previous problem, and returns the condition number of Q. Report the condition number $\kappa(Q)$ for varying values of p and ρ, by filling in the following table. Here, $m = 742$ is the total number of data samples. Report at least 2 significant digits. Comment on how much ridge regression is needed to affect conditioning.

p	$\rho = 0$	$\rho = m$	$\rho = 10m$	$\rho = 100m$
1				
2				
5				
10				

iii. Under the *same experimental parameters* as the previous question, run ridge regression for each choice of p and ρ, and fill in the table with the mean squared error of the fit:

$$\textbf{mean squared error} = \frac{1}{m}\sum_{i=1}^{m}(x_i^T\theta - y_{[i]})^2$$

where x_i is the ith row of X. Comment on the tradeoff between using larger ρ to improve conditioning vs its affect on the final performance.

p	$\rho = 0$	$\rho = m$	$\rho = 10m$	$\rho = 100m$
1				
2				
5				
10				

(d) *Forcasting.* Picking your favorite set of hyperparameters (p, ρ), forecast the next week's dew point temperature. Plot the forecasted data over the current observations. Do you believe your forecast? Why?

7. (**2 pts**) *Logistic regression for Binary MNIST*

(a) (**0.5 pt**) What is the gradient of the logistic loss function

$$f(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\log(\sigma(y_i x_i^T\theta)), \quad \sigma(s) = \frac{1}{1+e^{-s}}$$

where $y_i \in \{-1, 1\}$? (Hint: Check out problem 1 again.)

(b) **Coding gradient descent.** Do **not** use `scikit-learn` or other built in tools for this exercise. Please only use the packages that are already imported in the notebook. [1]

- Open `mnist_logreg.ipynb`. We will use logistic regression to diffrentiate 4's from 9's, a notoriously tricky problem. Run the first box to see what the data looks like.
- In the second box I have set up the problem for you by pulling out the train and test set, selecting out only the data samples related to 4's and 9's. I have not altered the data in any other way. While other normalization tricks will help you improve the accuracy, for the purposes of this exercise we will forgo them, so that it's easy to compare everyone's solutions.
- Fill in the next box by providing code that will return the loss function value and the gradient. Make sure that everything is normalized, e.g. don't forget the $1/m$ term in the front of our loss function. Run the script. If done correctly, you should see

```
45.192, 12343.177
```

- Write a script that returns the classification accuracy given θ.
- Use gradient descent to minimize the logistic loss for this classification problem. Use a step size of 10^{-6}.
- (**1 pt**) Run for 1500 iterations. In your report, give the plot of the train loss and train/test misclassification rate, plotted as a function of *iterations*. Report the final train and test accuracy values.

(c) **Coding stochastic gradient descent.** Do **not** use `scikit-learn` or other built in tools for this exercise. Please only use the packages that are already imported in the notebook. Now, fill in the next box a function that takes in θ and a minibatch $\mathcal{B}$ as either a list of numbers or as an np.array, and returns the *minibatch gradient*

$$\nabla_{\mathcal{B}} f(\theta) = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\nabla f_i(\theta)$$

where $f_i(\theta)$ is the contribution to the gradient from datapoint i:

$$f_i = -\log(\sigma(y_i x_i^T\theta)).$$

Run the script. If done correctly, you should see the number 5803.5 printed out.

[1] This is to test your understanding of the basic machine learning concepts, like calculating accuracy and logistic loss; in the future you can use whatever tools you'd like.

(d) Write a script to run stochastic gradient descent over logistic regression. When coding up the minibatching, make sure you cycle through an entire training set once before moving on to the next epoch. Additionally, use `time()` to record the runtime, and compare the performance of gradient descent and stochastic gradient descent, using a minibatch size of 50 data samples and running for 50000 iterations. Return a plot that compares the objective loss, train accuracy, and test accuracy between the two optimization methods, as a function of *runtime*. Comment on the pros and cons of the two methods.

Important Remember that calculating the loss function and train/test accuracy requires making *full passes* through the data. If you do this at each iteration, you will not see any runtime benefit between stochastic gradient descent and gradient descent. Therefore I recommend you log these values every 10 iterations for gradient descent, and every 100 iterations for stochastic gradient descent.

Challenge!

1. (**1 pt.**) *Gradient descent for ridge regression.* Consider the problem

$$\underset{x}{\text{minimize}} \quad \overbrace{\underbrace{\frac{1}{2}\|Ax-b\|_2^2}_{=:f(x)}+\frac{\rho}{2}\|x\|_2^2}^{F(x)} \tag{5}$$

where $A \in \mathbb{R}^{m\times n}$ and $n > m$. Justify all answers.

(a) Define $C = A^TA$. Recall that an eigenvalue of a symmetric matrix C is λ where $Cv = \lambda v$ for some vector v. Show that if $\lambda_{\max}^2$ is the largest eigenvalue of C and $\lambda_{\min}^2$ the minimum eigenvalue of C, then for any vector u,

$$\lambda_{\min}^2\|u\|_2 \le \|Cu\|_2 \le \lambda_{\max}^2\|u\|_2.$$

Hint: The *eigenvalue decomposition* of a symmetric matrix can be written as $C = V\Lambda V^T$ where $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$ contain the eigenvalues v_i of C, and $\Lambda = \mathbf{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$ contain the corresponding eigenvalues λ_i. (That is, $Cv_i = \lambda_i v_i$.) Under certain conditions which we will just assume [2] then V is an orthonormal matrix, e.g. $V^TV = VV^T = I$. Then, we can use this to form projections, e.g. pick any vector u. Then $VV^Tu = V^TVu = u$.

(b) Show that since $n > m$, then if $C = A^TA$ then $\lambda_{\min}(C) = 0$. Hint: The nonzero eigenvalues of AA^T and of A^TA are the same.

(c) We say a function $f : \mathbb{R}^n \to \mathbb{R}$ is L-smooth if its gradient is L-Lipschitz, e.g. for some $L > 0$,

$$\|\nabla f(x) - \nabla f(y)\|_2 \le L\|x-y\|_2, \quad \forall x, y.$$

Is $f(x)$ as defined in (5) L-smooth? What about $F(x)$?

(d) We say a function $f : \mathbb{R}^n \to \mathbb{R}$ is μ-strongly convex if it is tangent to a quadratic function that is strictly under it; that is, for some $\mu > 0$,

$$f(x) - f(y) \ge \nabla f(y)^T(x-y) + \frac{\mu}{2}\|x-y\|_2^2, \quad \forall x, y.$$

If this is only true for $\mu = 0$, we say f is convex, but not strongly convex.

Is the function $f(x)$ in (5) μ-strongly convex? What about $F(x)$? Hint: This problem requires less work if you first answer for $F(x)$, and then take $\rho = 0$ for $f(x)$.

(e) *Linear convergence.* We will now show that gradient descent on minimizing $F(x)$ converges *linearly.* First, recall that gradient descent iterates as

$$x^{(t+1)} = x^{(t)} - \alpha\nabla F(x^{(t)})$$

for some step size $\alpha > 0$.

i. Show that for any point x^* where $\nabla F(x^*) = 0$,

$$\|x^{(t+1)} - x^*\|_2 \le c\|x^{(t)} - x^*\|_2$$

for some $c < 1$. What is c?

ii. Use this to argue that gradient descent on (5) converges with *linear complexity*, e.g. the error $f(x^{(t)}) - f(x^*) = O(c^t)$. [3]

[2] eigenvalues must all have algebraic multiplicity = geometric multiplicity

[3] It's a weird convention, but we say $O(c^t)$ is a linear rate because the loglog graph looks linear. I don't make the rules, I just share them with you.

2. (**1pt**) *Linear regression without strong convexity still gets linear convergence.* Now consider linear regression with $A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, and $\rho = 0$ (no ridge). That is, we consider only

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2.$$

(a) *Linear algebra.* For a matrix A, the **nullspace** of A is the set

$$\mathbf{null}(A) = \{x : Ax = 0\},$$

and the **range** of A is the set

$$\mathbf{range}(A) = \{Ax \text{ for any } x\}.$$

Show that for

$$u = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \qquad v = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

then $u \in \mathbf{null}(A)$ and $v \in \mathbf{range}(A)$.

(b) *Linear decomposition theorem part 1.* Show that for *any* vector $u \in \mathbf{null}(A)$ and $v \in \mathbf{range}(A^T)$, it must be that $u^T v = 0$.

(c) *Linear decomposition theorem part 2.* Argue also that for *any* vector x, there exist some $u \in \mathbf{null}(A)$ and $v \in \mathbf{range}(A^T)$ where $x = u + v$. Do this by providing two matrices P and Q where $Px = u$ and $Qx = v$, using $A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$. (This matrix will be unique.)

(d) *Linear regression doesn't pick up nullspace components.* Suppose $x^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Now we run the gradient descent method

$$x^{(t+1)} = x^{(t)} - \alpha \nabla f(x^{(t)}) \tag{6}$$

to get $x^{(1)}$, $x^{(2)}$, ...
Show that using this initial point, $Qx^{(t)} = 0$ for *all* t, where Q is the matrix computed in the previous question.

(e) *Linear regression doesn't pick up nullspace components, another example.* Now suppose that for some $x^{(0)}$, $Qx^{(0)} = r \neq 0$. Again, we run the gradient descent method. Show that $Qx^{(t)} = r$ for all t. (That is, r does not depend on t!)

(f) Now consider a *reduced gradient descent problem*, where we minimize over a scalar variable v

$$\underset{v \in \mathbb{R}}{\text{minimize}}\ g(v) = \frac{1}{2}\|ASv - Pb\|_2^2, \quad S = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Argue that, using gradient descent

$$v^{(t+1)} = v^{(t)} - \alpha \nabla g(v^{(t)}),$$

the iterates $x^{(t)} = Sv^{(t)}$ are exactly those outputted by (6) where $x^{(0)} = Sv^{(0)}$.
Hint: Start by showing that $SS^T \nabla f(x) = \nabla f(x)$.

(g) Finally, show that $g(v)$ is strongly convex in u.

(h) Show that for this specific choice of A, gradient descent converges linearly, e.g. $f(x^{(t)}) - f(x^*) = O(c^t)$, and carefully state what c is.

(i) Now consider *any* matrix A. Argue that this entire problem basically shows that gradient descent, minimizing linear regression, will *always* converge at a linear rate, and describe what c is.