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

   Ans. Actually, the best way to do this is to invoke the chain rule, which you will prove in the first graded problem. Write $g(v) = \frac{1}{40}\|v - b\|_2^2$. Then since $b \in \mathbb{R}^{20}$,

   $$\nabla g(v) = \nabla_v \left(\frac{1}{40} \sum_{i=1}^{20} (v_{[i]} - b_{[i]})^2 \right) \overset{\text{linearity}}{=} \frac{1}{40} \sum_{i=1}^{20} \nabla_v \left((v_{[i]} - b_{[i]})^2 \right).$$

   Note that

   $$\nabla_v (v_{[i]} - b_{[i]})^2 = \begin{bmatrix} \frac{\partial}{\partial v_{[1]}} (v_{[i]} - b_{[i]})^2 \\ \frac{\partial}{\partial v_{[2]}} (v_{[i]} - b_{[i]})^2 \\ \vdots \\ \frac{\partial}{\partial v_{[20]}} (v_{[i]} - b_{[i]})^2 \end{bmatrix}$$

   and

   $$\frac{\partial}{\partial v_{[k]}} (v_{[i]} - b)^2 = \begin{cases} 2(v_{[i]} - b) & \text{if } i = k \\ 0 & \text{else.} \end{cases}$$

   So,

   $$\sum_{i=1}^{20} \nabla_v (v_{[i]} - b_{[i]})^2 = 2 \begin{bmatrix} \frac{\partial}{\partial v_{[1]}} (v_{[1]} - b_{[1]}) \\ \frac{\partial}{\partial v_{[2]}} (v_{[2]} - b_{[2]}) \\ \vdots \\ \frac{\partial}{\partial v_{[20]}} (v_{[20]} - b_{[20]}) \end{bmatrix} = 2(v - b).$$

   and $\nabla g(v) = \frac{1}{20}(v - b)$.
   Now, we invoke the chain rule. (Note that f and g are flipped as to their position in 1.(b).) Then

   $$\nabla f(x) = A^T \nabla g(Ax) = A^T (\frac{1}{20}(Ax - b)) = \frac{1}{20} A^T (Ax - b).$$

   To get the dimension, you can do this in two ways. One, you notice that A has 10 columns, so A^T has 10 rows. Two, you notice that the gradient $\nabla f(x)$ should always have the same number of elements as x, which is 10. In either case, $\nabla f(x) \in \mathbb{R}^{10}$.

   (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 $\sigma'(s) = \sigma(s)(1 - \sigma(s))$.

   Ans. We start with the hint, noting that

   $$\sigma'(s) = \frac{-\exp(-x)}{(1 - \exp(-x))^2} = \frac{1}{1 - \exp(-x)} \cdot \left(1 - \frac{1}{1 - \exp(-x)} \right) = \sigma(s)(1 - \sigma(s)).$$

   Then using chain rule, (where $A = c^T$) we can get

   $$\nabla f(x) = \sigma'(c^T x)c = \sigma(c^T x)(1 - \sigma(c^T x))c \in \mathbb{R}^5.$$

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

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

Ans. This set is convex. Again, we check the definition: suppose $x \in \mathbf{range}(A)$ and $y \in \mathbf{range}(A)$. Then there exists some u and v where $Au = x$ and $Av = y$. Then for any $z = \theta x + (1-\theta)y$,

$$z = \theta Au + (1-\theta)Av = A\underbrace{(\theta u + (1-\theta)v)}_{w} = Aw \in \mathbf{range}(A).$$

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

Ans. This set is not convex. We can just take

$$x = 1, \quad y = -1, \quad \theta = 1/2$$

and

$$\theta x + (1-\theta)y = 0 \notin \mathcal{S}.$$

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

Ans. The key observation here is that X can be factorized, e.g.

$$X = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$

Therefore, taking $c = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$,

$$u^T X u = (u^T c)^2 \geq 0 \quad \text{for all } u.$$

Therefore X **is positive semidefinite.**

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

Ans. Any matrix with negative diagonal elements is **not positive semidefinite**. To see this, pick $u = [1, 0, 0]^T$. Then

$$u^T X u = -1 < 0.$$

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

Ans. This function is convex. We can use any of the three conditions to check.

- **By definition.** To make life easier, we first observe that this function can be rewritten as

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

Then

$$\begin{aligned} f(\theta x + (1-\theta)y) &= \frac{1}{2}(\theta x_{[1]} + (1-\theta)y_{[1]} - \theta x_{[2]} - (1-\theta)y_{[2]})^2 \\ &= \frac{\theta^2}{2}(x_{[1]} - x_{[2]})^2 + \frac{(1-\theta)^2}{2}(y_{[1]} - y_{[2]})^2 - \theta(1-\theta)(x_{[1]} - x_{[2]})(y_{[1]} - y_{[2]}) \end{aligned}$$

Note that in general, since $(a+b)^2 \geq 0$, then $a^2 + b^2 \geq -2ab$, for any scalars a and b. Therefore

$$(x_{[1]} - x_{[2]})(y_{[1]} - y_{[2]}) \leq \frac{1}{2}(x_{[1]} - x_{[2]})^2 + \frac{1}{2}(y_{[1]} - y_{[2]})^2.$$

Then

$$\begin{aligned} f(\theta x + (1-\theta)y) &\leq \frac{\theta^2}{2}(x_{[1]} - x_{[2]})^2 + \frac{(1-\theta)^2}{2}(y_{[1]} - y_{[2]})^2 + \frac{1}{2}\theta(1-\theta)((x_{[1]} - x_{[2]})^2 + (y_{[1]} - y_{[2]})^2) \\ &= \frac{\theta^2 + \theta(1-\theta)}{2}(x_{[1]} - x_{[2]})^2 + \frac{(1-\theta)^2 + \theta(1-\theta)}{2}(y_{[1]} - y_{[2]})^2 \\ &= \frac{\theta}{2}(x_{[1]} - x_{[2]})^2 + \frac{1-\theta}{2}(y_{[1]} - y_{[2]})^2 \\ &= \theta f(x) + (1-\theta)f(y) \end{aligned}$$

which satisfies the definition of a convex function.

- **By first order condition** Here, I'm going to again rewrite the problem as

$$f(x) = \frac{1}{2}(x^T c)^2$$

where $c = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$. Then

$$\nabla f(x) = (x^T c)c$$

and

$$\begin{aligned} f(x) - f(y) - \nabla f(y)^T(x-y) &= \frac{1}{2}(x^T c)^2 - \frac{1}{2}(y^T c)^2 - (y^T c)c^T(x-y) \\ &= \frac{1}{2}(x^T c)^2 + \frac{1}{2}(y^T c)^2 - (y^T c)c^T x \\ &= \frac{1}{2}(x^T c - y^T c)^2 \geq 0, \quad \forall x, y. \end{aligned}$$

Thus the first order condition is satisfied!

- **By second order condition.** This is by far the fastest way to check.

$$\nabla^2 f(x) = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

To show this matrix is positive semidefinite, it suffices to use the definition

$$u^T \nabla^2 f(x) u = (u_{[1]} - u_{[2]})^2 \geq 0, \quad \forall u.$$

Alternatively, we can try to "factorize" the Hessian, e.g. noticing that

$$\nabla^2 f(x) = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}^T.$$

Therefore, taking $c = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, then

$$u^T \nabla^2 f(x) u = (c^T u)^2 \geq 0, \quad \forall u.$$

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

Ans. Since this function is not differentiable, our only option is to use the definition of convexity. We check as

$$f(\theta x + (1-\theta)y) = |\theta x + (1-\theta)y| \overset{\text{triangle inequality}}{\leq} |\theta x| + |(1-\theta)y| = \theta|x| + (1-\theta)|y| = \theta f(x) + (1-\theta) f(y).$$

Thus the definition of convex function is satisfied!

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

Ans. This function is convex. For this problem, I really don't recommend trying to use the definition or first order condition, as they are really not straightforward to verify. However, the second order condition works nicely.

First, note we can write the gradient as

$$\nabla f(x) = \begin{bmatrix} \frac{\exp(x_{[1]})}{\exp(x_{[1]})+\exp(x_{[2]})} \\ \frac{\exp(x_{[2]})}{\exp(x_{[1]})+\exp(x_{[2]})} \end{bmatrix} = \begin{bmatrix} 1 - \frac{\exp(x_{[2]})}{\exp(x_{[1]})+\exp(x_{[2]})} \\ 1 - \frac{\exp(x_{[1]})}{\exp(x_{[1]})+\exp(x_{[2]})} \end{bmatrix}$$

Now the Hessian can be written as

$$\nabla^2 f(x) = \frac{\exp(x_{[1]})\exp(x_{[2]})}{(\exp(x_{[1]}) + \exp(x_{[2]}))^2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Since $\frac{\exp(x_{[1]})\exp(x_{[2]})}{(\exp(x_{[1]})+\exp(x_{[2]}))^2} > 0$ for all x, and the matrix

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}^T$$

is positive semidefinite, then the Hessian is positive semidefinite, and the function is convex.

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

Ans. Since this function is linear, it is also convex. Any of the following arguments work:

- Using the definition of convexity,

$$f(\theta x + (1-\theta)y) = c^T(\theta x + (1-\theta)y) = \theta c^T x + (1-\theta)c^T y = \theta f(x) + (1-\theta)c^T y.$$

- Using first order condition,

$$f(x) - f(y) - \nabla f(y)^T (x - y) = c^T x - c^T y - c^T (x - y) = 0 \geq 0.$$

- Technically you could also say that the Hessian is $\nabla^2 f(x) = 0$ which is positive semidefinite, since for any u, $u^T \nabla^2 f(x) u = 0 \geq 0$.

Main assignment, graded

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

Lihan

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

Ans. Actually, this is just a direct consequence that partial derivatives are linear. That is,

$$\frac{\partial h}{\partial x_{[i]}}(x) = \alpha \frac{\partial f}{\partial x_{[i]}}(x) + \beta \frac{\partial g}{\partial x_{[i]}}(x).$$

Therefore,

$$\nabla h(x) = \begin{bmatrix} \frac{\partial h}{\partial x_{[1]}}(x) \\ \frac{\partial h}{\partial x_{[2]}}(x) \\ \vdots \\ \frac{\partial h}{\partial x_{[n]}}(x) \end{bmatrix} = \begin{bmatrix} \alpha \frac{\partial f}{\partial x_{[1]}}(x) + \beta \frac{\partial g}{\partial x_{[1]}}(x) \\ \frac{\partial f}{\partial x_{[2]}}(x) + \beta \frac{\partial g}{\partial x_{[2]}}(x) \\ \vdots \\ \frac{\partial f}{\partial x_{[n]}}(x) + \beta \frac{\partial g}{\partial x_{[n]}}(x) \end{bmatrix} = \alpha \begin{bmatrix} \frac{\partial f}{\partial x_{[1]}}(x) \\ \frac{\partial f}{\partial x_{[2]}}(x) \\ \vdots \\ \frac{\partial f}{\partial x_{[n]}}(x) \end{bmatrix} + \beta \begin{bmatrix} \frac{\partial g}{\partial x_{[1]}}(x) \\ \frac{\partial g}{\partial x_{[2]}}(x) \\ \vdots \\ \frac{\partial g}{\partial x_{[n]}}(x) \end{bmatrix} = \alpha \nabla f(x) + \beta \nabla g(x).$$

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

Ans. The easiest way to do this is to just brute force it. We denote the columns of $A \in \mathbb{R}^{m \times n}$ as

Lihan

$$A = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}, \quad a_i \in \mathbb{R}^m.$$

Then

$$\frac{\partial g}{\partial v_{[i]}}(v) = \sum_k \frac{\partial f}{\partial x_{[k]}}(a_k^T v) \cdot A_{[k,i]} = a_i^T \nabla f(Av).$$

Therefore,

$$\nabla g(v) = \begin{bmatrix} \frac{\partial g}{\partial v_{[1]}}(v) \\ \frac{\partial g}{\partial v_{[2]}}(v) \\ \vdots \\ \frac{\partial g}{\partial v_{[n]}}(v) \end{bmatrix} = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_n^T \end{bmatrix} \nabla f(Av) = A^T \nabla f(Av).$$

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 ($Q_{[i,j]} = Q_{[j,i]}$).

Lihan

Ans. We can do this piece by piece. First, consider

$$f_1(x) = \frac{1}{2} x^T Q x = \frac{1}{2} \sum_{i=1}^{12} \sum_{j=1}^{12} Q_{[i,j]} x_{[i]} x_{[j]}.$$

Then

$$\frac{\partial f_1(x)}{\partial x_{[k]}} = \frac{1}{2} \sum_{i=1}^{12} \sum_{j=1}^{12} \frac{\partial}{\partial x_{[k]}} (Q_{[i,j]} x_{[i]} x_{[j]})$$

and

$$\frac{\partial}{\partial x_{[k]}} (Q_{[i,j]} x_{[i]} x_{[j]}) = \begin{cases} Q_{[k,j]} x_{[j]} & \text{if } k = i \\ Q_{[i,k]} x_{[i]} & \text{if } k = j \\ 0 & \text{else.} \end{cases}$$

So, we can get to

$$\frac{\partial f_1(x)}{\partial x_{[k]}} = \frac{1}{2} \left(\frac{\partial}{\partial x_{[k]}} Q_{[k,j]} x_{[j]} + \sum_{i=1}^{12} Q_{[i,k]} x_{[i]} \right) = \frac{1}{2} \cdot 2Qx = Qx.$$

Now let's consider $f_2(x) = p^T x = \sum_{k=1}^{12} p_{[k]} x_{[k]}$. Then

$$\nabla f_2(x) = \begin{bmatrix} \frac{\partial}{\partial x_{[1]}} \left(\sum_{k=1}^{12} p_{[k]} x_{[k]} \right) \\ \frac{\partial}{\partial x_{[2]}} \left(\sum_{k=1}^{12} p_{[k]} x_{[k]} \right) \\ \vdots \\ \frac{\partial}{\partial x_{[12]}} \left(\sum_{k=1}^{12} p_{[k]} x_{[k]} \right) \end{bmatrix} = \begin{bmatrix} p_{[1]} \\ p_{[2]} \\ \vdots \\ p_{[12]} \end{bmatrix} = p.$$

Sometimes, we refer to this property

$$\frac{\partial}{\partial x_{[j]}} \left(\sum_{k=1}^{12} p_{[k]} x_{[k]} \right) = p_{[j]}$$

as a "picking property", because we pick out the element of p that we're interested in. Finally, observing that r is a constant, we get

$$\nabla f(x) = \underbrace{Qx}_{\nabla f_1(x)} + \underbrace{p}_{\nabla f_2(x)} \in \mathbb{R}^{12}.$$

(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

Ans. Again, it's useful here to use chain rule. In particular, we decompose

$$f(x) = g(\sum_i h(x_i)), \quad g(s) = \frac{1}{\mu} \log(s), \; h(z) = \exp(\mu z)$$

with derivatives

$$g'(s) = \frac{1}{\mu s}, \quad h'(z) = \mu \exp(\mu z).$$

Then using chain rule,

$$\frac{\partial f(x)}{\partial x_{[k]}} = g'(\sum_i h(x_i)) h'(x_{[k]})$$

and plugging in everything, we get

$$\frac{\partial f(x)}{\partial x_{[k]}} = \frac{1}{\mu \sum_i \exp(\mu x_{[i]})} \cdot \mu \exp(\mu x_{[k]}) = \frac{\exp(\mu x_{[k]})}{\sum_i \exp(\mu x_{[i]})}.$$

In matrix form, we write

$$\nabla f(x) = \frac{1}{\sum_i \exp(\mu x_{[i]})} \begin{bmatrix} \exp(\mu x_{[1]}) \\ \exp(\mu x_{[2]}) \\ \vdots \\ \exp(\mu x_{[8]}) \end{bmatrix} \overset{\text{abuse of notation}}{=} \frac{\exp(\mu x)}{\sum_i \exp(\mu x_{[i]})} \in \mathbb{R}^8.$$

[1]

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

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

Lihan

Ans. This set is convex. We can just test the definition: Suppose $x \in \mathcal{S}$ and $y \in \mathcal{S}$. Then for some $0 \le \theta \le 1$,

$$\sum_i \theta x_i + (1-\theta) y_i = \theta \underbrace{\sum_i x_i}_{=0} + (1-\theta) \underbrace{\sum_i y_i}_{=0} = 0.$$

[1] For those of you who train neural networks for multiclass classification, you probably recognize this as the softmax layer. Well, this is where the name "softmax" comes from! (The function $f(x)$ gives a "soft approximation" of the maximum element of x_i.)

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

Ans. This set is not convex. We can give a counterexample:

$$x = 1/\sqrt{2}, \quad y = -1/\sqrt{2}, \quad \theta = 1/2.$$

Then $x^2 + y^2 = 1$, but $\theta x + (1-\theta)y = 0 \neq 1$.

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

Ans. This set is convex. Assume that $|x| \leq 1$ and $|y| \leq 1$. Then picking any $0 \leq \theta \leq 1$,

$$|\theta x + (1-\theta)y| \overset{\text{triangle inequality}}{\leq} |\theta x| + |(1-\theta)y| = \theta \underbrace{|x|}_{\leq 1} + (1-\theta)|y| \leq \theta + (1-\theta) = 1.$$

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.

Vinh

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

Ans. This one is a tricky one. I intentionally made all the values of X positive to try to throw you off, but actually this matrix is **not positive semidefinite.** To see this, pick $u = [1, 0, -1]^T$. Then

$$u^T X u = \begin{bmatrix} 1-3 & 2-4 & 3-1 \end{bmatrix}^T \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = -2 - 2 = -4 < 0.$$

Usually, such examples can be found when the diagonal of X is not dominant, e.g. the off-diagonal values are too large compared to the diagonal ones.

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

Ans. This matrix **is positive semidefinite.** In fact, this is true for any matrix that is a diagonal of only nonnegative values. To see this, note that if X is diagonal, then

$$u^T X u = \sum_{i=1}^{n} X_{ii} u_i^2 \geq 0 \text{ if all } X_{ii} \geq 0.$$

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 $x > 0$

Ans. This function is twice differentiable, so we can check using any of the three conditions.

- **By definition.**

Rajat

$$\begin{aligned} \theta f(x) + (1-\theta)y &= \frac{\theta}{x} + \frac{(1-\theta)}{y} \\ &= \frac{\frac{\theta}{x}(\theta x + (1-\theta)y) + \frac{(1-\theta)}{y}(\theta x + (1-\theta)y)}{\theta x + (1-\theta)y} \\ &= \frac{(\theta^2 + (1-\theta)^2 + \theta(1-\theta)(\frac{y}{x} + \frac{x}{y}))}{\theta x + (1-\theta)y} \end{aligned}$$

Either $y \geq x$ or $x \geq y$, so it must be that $\frac{y}{x} + \frac{x}{y} \geq 1$. Therefore

$$\begin{aligned} \theta f(x) + (1-\theta)y &\geq \frac{(\theta^2 + (1-\theta)^2 + \theta(1-\theta))}{\theta x + (1-\theta)y} \\ &= \frac{(\theta + 1 - \theta)^2}{\theta x + (1-\theta)y} \\ &= \frac{1}{\theta x + (1-\theta)y} \\ &= f(\theta x + (1-\theta)y) \end{aligned}$$

- **First order condition.**

$$f(x) - f(y) - f'(y)(x-y) = \frac{1}{x} - \frac{1}{y} + \frac{1}{y^2}(x-y) = \frac{1}{x} - \frac{2}{y} + \frac{x}{y^2} = \frac{1}{x}(1 - \frac{2x}{y} + \frac{x^2}{y^2}) = \frac{1}{x}(1 - \frac{x}{y})^2 \geq 0, \ \forall x, y.$$

- **Second order condition.**

$$f''(x) = \frac{2}{x^3} \geq 0 \quad \forall x \geq 0.$$

As I said, some methods are far more straightforward than others.

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

Ans. Since this function is not differentiable, we can only use the definition to verify convexity.

$$f(\theta x + (1-\theta)y) = \|\theta x + (1-\theta)y\|_\infty \overset{\text{triangle inequality}}{\leq} \|\theta x\|_\infty + \|(1-\theta)y\|_\infty = \theta\|x\|_\infty + (1-\theta)\|y\|_\infty = \theta f(x) + (1-\theta)f(y).$$

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

Ans. This function is not convex. We can disprove any of the three conditions.

- To disprove the definition, we just need to pick a clever choice of x and y, and show that $f(\theta x + (1-\theta)y) > \theta f(x) + (1-\theta)f(y)$. Since it should be clear that it's the cubic term that's causing the problems, a clever choice may be

$$x = \begin{bmatrix} 0 \\ 0 \\ -2 \end{bmatrix}, \quad y = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Then taking $\theta = 1/2$,

$$f(x) = -8, \quad f(y) = 0$$

and

$$f(\frac{1}{2}x + \frac{1}{2}y) = f(\begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}) = -1.$$

Since

$$-4 = \frac{1}{2}f(x) + \frac{1}{2}f(y) \leq f(\frac{1}{2}x + \frac{1}{2}y) = -1$$

then this function can't be convex.

- **First order condition.** Taking the gradient

$$\nabla f(x) = \begin{bmatrix} 1 \\ 2x_{[2]} \\ 3x_{[3]}^2 \end{bmatrix},$$

then we need to show that

$$f(x) - f(y) - \nabla f(y)^T(x-y) < 0.$$

Again, noting that the problem is going to happen in the third component, we can actually use the same example:

$$x = \begin{bmatrix} 0 \\ 0 \\ -2 \end{bmatrix}, \quad y = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Then

$$f(x) = -8, \quad f(y) = 0, \quad \nabla f(y)^T(x-y) = 0 \cdot (-2-0) = 0$$

and

$$f(x) - f(y) - \nabla f(y)^T(x-y) = -8 < 0.$$

- **Second order condition.** Checking this condition, we see that

$$\nabla^2 f(x) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 6x_{[3]} \end{bmatrix}$$

which, for any x where $x_{[3]} < 0$, will not be positive semidefinite. (No matrix with any negative diagonal element can be positive semidefinite. To see this, just pick u where $u_{[i]} = 1$ whenever $H_{ii} < 0$, and $u_j = 0$ otherwise. Then $u^T H u = H_{ii} < 0$.)

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

Ans. This function is convex.

- Using the definition:

$$\begin{aligned} f(\theta x + (1-\theta)y) &= \|A(\theta x + (1-\theta)y - b\|_2^2 \\ &= \|\theta(Ax-b) + (1-\theta)(Ay-b)\|_2^2 \\ &= \theta^2\|Ax-b\|_2^2 + 2\theta(1-\theta)(Ax-b)^T(Ay-b) + (1-\theta)^2\|Ay-b\|_2^2. \end{aligned}$$

We use the Cauchy Schwartz inequality to say that

$$(Ax-b)^T(Ay-b) \le \|Ax-b\|_2\|Ay-b\|_2$$

and we further use $(a-b)^2 \ge 0 \iff 2ab \le a^2 + b^2$ to say

$$2(Ax-b)^T(Ay-b) \le 2\|Ax-b\|_2\|Ay-b\|_2 \le \|Ax-b\|_2^2 + \|Ay-b\|_2^2.$$

Then

$$\begin{aligned} f(\theta x + (1-\theta)y) &\le \theta^2\|Ax-b\|_2^2 + \theta(1-\theta)(\|Ax-b\|_2^2 + \|Ay-b\|_2^2) + (1-\theta)^2\|Ay-b\|_2^2 \\ &= \theta\|Ax-b\|_2^2 + (1-\theta)\|Ay-b\|_2^2 \\ &= \theta f(x) + (1-\theta)f(y). \end{aligned}$$

As you can see, using the definition here is really not very straightforward.

- First order condition:

$$f(x) - f(y) - \nabla f(y)^T(x-y) = \|Ax-b\|_2^2 - \|Ay-b\|_2^2 - 2(x-y)^T A^T(Ay-b) = \|Ax-Ay\|_2^2 \ge 0$$

- Second order condition

$$\nabla^2 f(x) = A^T A$$

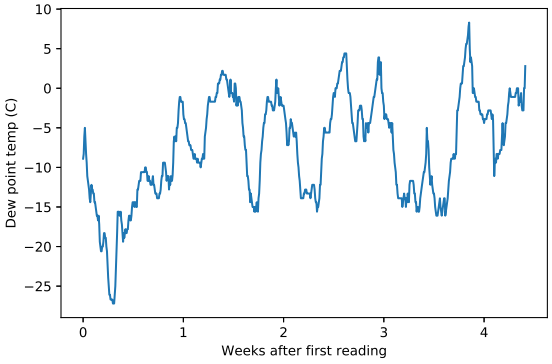
so

$$u^T \nabla^2 f(x) u = u^T A^T A u = \|Au\|_2^2 \ge 0, \quad \forall u.$$

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

Vinh

- 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 (**??**) 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

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

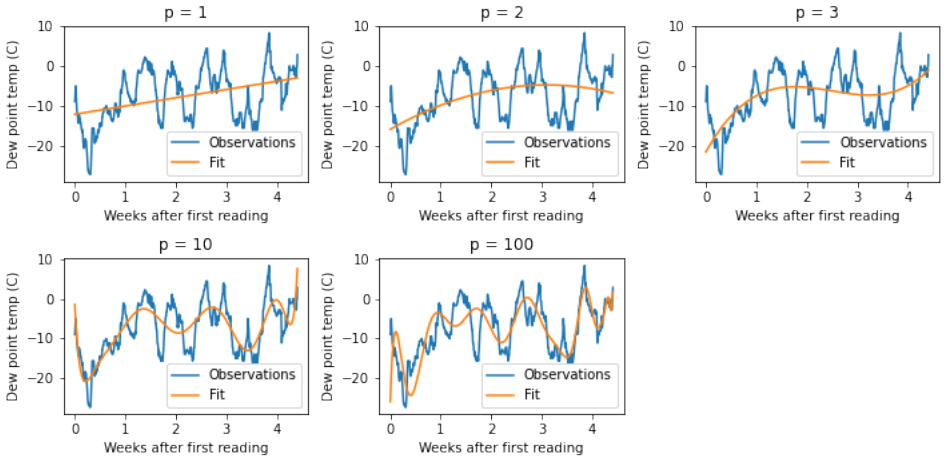
(a) *Linear regression.*

i. Write down the normal equations for problem (**??**). **Ans.** The normal equations are characterized by the linear system that emerges from setting the gradient of (**??**) to 0:

$$X^T X\theta = X^T y.$$

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 1.759 (implemented correctly) or 1.341 (also accepted).

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. **Ans.**



The goodness of fit definitely improves with larger p, with no obvious defects. However, it is possible that as p gets really large, some overfitting may be occurring. (Graders, give full credit for any relevant observation that is not false.)

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

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

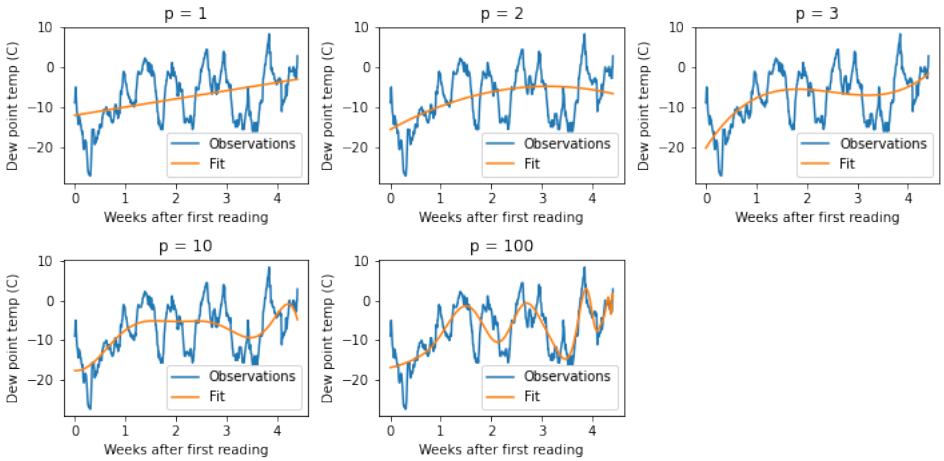
for some $\rho > 0$.

i. Again, write down the normal equations for (**??**). Your equation should be of form $A\theta = b$ for some matrix A and vector b that you specify.
Ans. The normal equations here just need to also include the regularization term in the gradient of (**??**):

$$(X^T X + \rho I)\theta = X^T y.$$

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.
Ans.



With such a large value of ρ, you do see more smoothing in the resulting figure. Whether or not that's a good thing is hard to decide; less overfitting may be more stable, but more overfitting may be more accurate.
(Graders, give full credit for any relevant observation that is not false.)

(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 (**??**) is an unconstrained quadratic problem by writing down Q, c, and r in terms of X and y such that (**??**) is equivalent to (**??**). Show that the Q you picked is positive semidefinite.

Ans. Expanding the ridge regression objective function gives Q, c, and r:

$$\frac{1}{2}\|X\theta - y\|_2^2 + \frac{\rho}{2}\|\theta\|_2^2 = \frac{1}{2}\theta^T X^T X\theta - y^T X\theta + \frac{1}{2}y^T y + \frac{\rho}{2}\theta^T\theta = \frac{1}{2}\theta^T \underbrace{(X^T X + \rho I)}_{Q}\theta \underbrace{-y^T X}_{c}\theta + \underbrace{\frac{1}{2}y^T y}_{r}.$$

To see that Q is positive semidefinite, pick any vector u. Then

$$u^T Qu = u^T X^T Xu + \rho u^T u = \|Xu\|_2^2 + \rho\|u\|_2^2 \geq 0, \quad \forall u.$$

Therefore, Q 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				

Ans.

p	$\rho = 0$	$\rho = m$	$\rho = 10m$	$\rho = 100m$
1	3.25e+01	6.75e+00	1.69e+00	1.07e+00
2	1.48e+03	7.91e+01	9.20e+00	1.82e+00
5	7.31e+08	2.72e+05	2.72e+04	2.72e+03
10	7.16e+18	3.97e+11	3.97e+10	3.97e+09

In general, bigger p results in higher condition numbers. However, bigger ρ can reduce the condition numbers.

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				

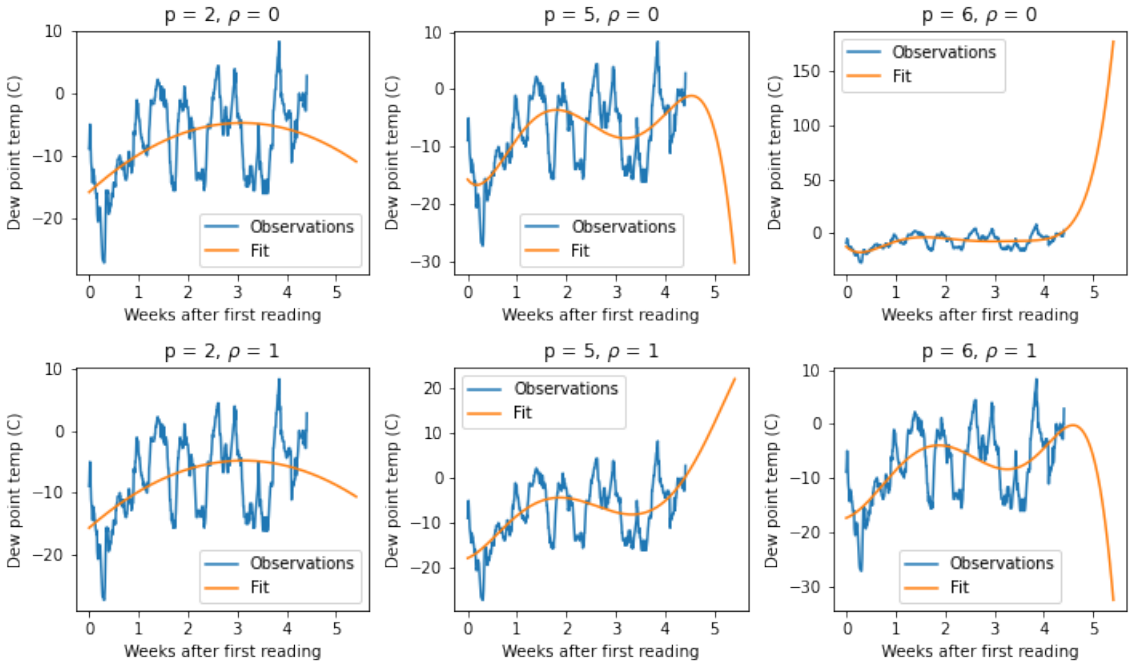
Ans.

p	$\rho = 0$	$\rho = m$	$\rho = 10m$	$\rho = 100m$
1	36.4	58.8	78.8	96.3
2	33.5	57.6	77.3	86.8
5	27.1	57.2	72.0	77.5
10	16.7	55.4	71.4	76.7

While larger ρ and smaller p give better condition numbers, it is clear that it comes as a hit in performance, as the MSE grows as well. Based on this table, I would pick $\rho = 0$ and $p = 2$ as a best fit choice, but there might be other reasons to choose otherwise.

(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?

Ans. There are a number of possible solutions here. Here are some examples. (In general, for useful values of p, I did not notice much impact of ρ.)



There are definitely many cases where I don't believe the numbers, especially if $\rho = 0$ since the forcasting seems pretty wild. But, when $\rho = 1$ and larger values of p, the curves look plausible–but if I were a betting person I still wouldn't put down my mortgage!

You could reach two conclusions here: either polynomial fitting is not a great tool because there's no reason why dew temperature should follow a polynomial structure, or we could say that dew temperature is not that predictable, period, using only historical data.

Graders, any discussion here that is reasonable should get full credit.

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

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

Rajat

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

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

Ans. To reduce notation, we write $z_i = y_i x_i$, and the matrix $Z = [z_1, z_2, ...z_m]^T$ has each vector z_i as a row. We can work out the derivatives of the sigmoid function, and show that

$$\sigma'(s) = \sigma(s)(1 - \sigma(s)).$$

After that, standard calculus rules gets us to

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = -\frac{1}{m}\sum_{i=1}^{m} \frac{\sigma(z_i^T\theta)(1-\sigma(z_i^T\theta))}{\sigma(z_i^T\theta)} z_i.$$

This yields

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{1}{m}\sum_{i=1}^{m}(\sigma(z_i^T\theta) - 1)z_i.$$

which we can write succinctly as

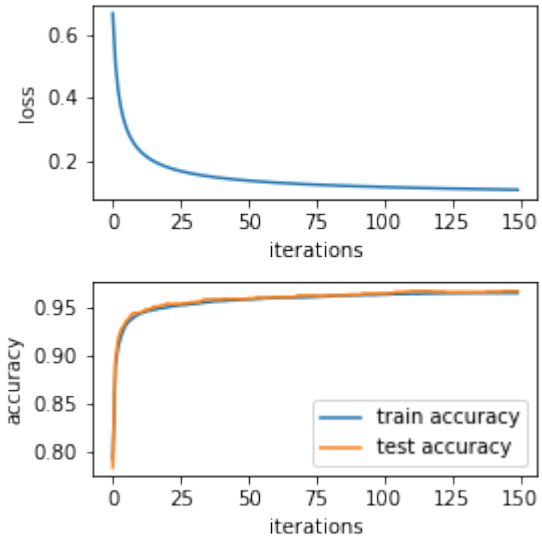
$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{1}{m}Z^T(d - \mathbf{1})$$

for $d_i = \sigma(z_i^T\theta)$, $i = 1, ..., m$.

(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. [2]

- 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.
  Ans. Train accuracy: 96.58%, test accuracy: 96.83%. Plot:



(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*

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

[2] 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.

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

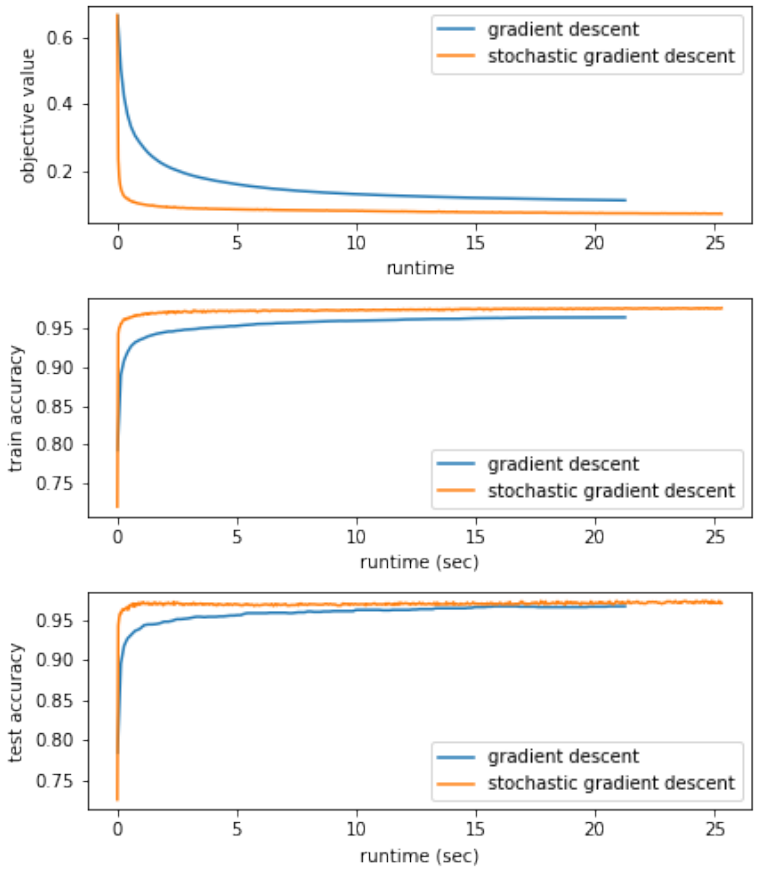
$$f_i = -\log(\sigma(y_i x_i^T \theta)).$$

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

(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.

Ans.



The computational runtime benefit of using minibatched gradients is clear here, even though our original full batch gradient descent is not that computationally burdensome. A possible con of using the SGD vs the full batch GD approach is that picking the minibatch size might not be obvious. Also, although it is not obvious in these plots, running SGD with a constant step size cannot actually get to the exact minimum loss, but rather wanders in a small region of error. However, in this case, the small region of error is so tiny that it is imperceptible, and certainly does not affect the test accuracy.

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 [3] 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 $VV^Tu = V^TVu = u$.

Ans. Pick any u. Then since $CV = VV^TCV = V\Lambda$,

$$Cu = CVV^Tu = V\Lambda V^Tu.$$

Then

$$\|Cu\|_2^2 = u^TC^TCu = u^TV\Lambda V^TV\Lambda V^Tu = u^TV^T\Lambda\Lambda V^Tu = \sum_{i=1}^n \lambda_i^2(v_i^Tu)^2.$$

Well, in general,

$$\lambda_{\min}^2\sum_{i=1}^n (v_i^Tu)^2 \le \sum_{i=1}^n \lambda_i^2 v_i^Tu \le \lambda_{\max}^2\sum_{i=1}^n (v_i^Tu)^2$$

and $\sum_{i=1}^n (v_i^Tu)^2 = u^TV^TVu = u^Tu = \|u\|_2^2$.
Therefore,

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

Taking the square roots everywhere gives us the result we want.

(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.

Ans. Without using more powerful tools like SVD and eigenvalue decompositions, the easiest way to argue this is that $AA^T \in \mathbb{R}^{m\times m}$ and has m eigenvalues, and $A^TA \in \mathbb{R}^{n\times n}$ and has n eigenvalues. Since $n > m$ and the nonzero eigenvalues of AA^T and A^TA are the same, it must be that at least one of the eigenvalues of A^TA is 0. To see that this is also the minimum eigenvalue, note that

$$A^TAu = \lambda u \iff u^TA^TAu = \lambda u^Tu \iff \lambda = \frac{\|Au\|_2^2}{\|u\|_2^2} \ge 0, \quad \forall \text{ eigenvectors } u.$$

In other words, *all eigenvalues of A^TA must be nonnegative.* Therefore, A^TA has a 0 eigenvalue and it is also the minimum eigenvalue.

(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 (**??**) L-smooth? What about $F(x)$?

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

Ans. First, we need to compute some gradients:

$$\nabla f(x) = A^T(Ax - b), \quad \nabla F(x) = A^T(Ax - b) + \rho I.$$

Then

$$\|\nabla f(x) - \nabla f(y)\|_2 = \|A^TA(x-y)\|_2 \overset{\text{from part (a)}}{\leq} \lambda_{\max}(A^TA)\|x-y\|_2$$

which shows that f is L-smooth, with $L = \lambda_{\max}(A^TA)$.
Note also that

$$\|\nabla F(x) - \nabla F(y)\|_2 = \|A^TA(x-y)\|_2 \leq \lambda_{\max}(A^TA)\|x-y\|_2$$

which shows that F is also L-smooth, with $L = \lambda_{\max}(A^TA)$.

(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 $\mu > 0$,

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

If this is only true for $\mu = 0$, we say f is convex, but not strongly convex.
Is the function $f(x)$ in (**??**) 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 $f(x)$. **Ans.** First we do some messy calculations:

$$\begin{aligned}
F(x) - F(y) - \nabla F(y)^T(x-y) &= \frac{1}{2}\|Ax-b\|_2^2 + \frac{\rho}{2}\|x\|_2^2 - \frac{1}{2}\|Ay-b\|_2^2 + \frac{\rho}{2}\|y\|_2^2 - (A^T(Ay-b) + \rho y)^T(x-y) \\
&= \frac{1}{2}x^TA^TAx - b^TAx + \frac{1}{2}b^Tb + \frac{\rho}{2}x^Tx - \frac{1}{2}y^TA^TAy - b^TAy + \frac{1}{2}b^Tb - \frac{\rho}{2}y^Ty \\
&\quad -x^TA^TAy + x^TA^Tb - \rho y^Tx + y^TA^TAy - y^TA^Tb + \rho y^Ty \\
&\overset{\text{simplify}}{=} \frac{1}{2}x^TA^TAx + \frac{1}{2}y^TA^TAy - x^TA^TAy + \frac{\rho}{2}(x^Tx - 2x^Ty + y^Ty) \\
&= \frac{1}{2}\|Ax-Ay\|_2^2 + \frac{\rho}{2}\|x-y\|_2^2 \\
&\leq \frac{\lambda_{\min}^2(A) + \rho}{2}\|x-y\|_2^2.
\end{aligned}$$

So, $F(x)$ is μ-strongly convex with $\mu = \lambda_{\min}^2(A) + \rho$. But, from part (b), we know that $\lambda_{\min} = 0$, so more simply, $F(x)$ is just ρ-strongly convex, and $f(x)$ is not strongly convex.

(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 $\nabla F(x^*) = 0$,

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

for some $c < 1$. What is c?
Ans. First, note that

$$\begin{aligned}
x^{(t+1)} &= x^{(t)} - \alpha A^T(Ax^{(t)} - b) - \alpha\rho x^{(t)} \\
&= (I - \alpha A^TA - \alpha\rho I)x^{(t)} + \alpha A^Tb.
\end{aligned}$$

Then, since

$$\nabla F(x^*) = 0 \iff A^TAx^* + \rho x^* = A^Tb$$

then

$$\begin{aligned} x^{(t+1)} - x^* &= (I - \alpha A^T A - \alpha\rho I)x^{(t)} + \alpha A^T b - x^* \\ &\overset{A^TAx^* + \rho x^* = A^T b}{=} (I - \alpha A^T A - \alpha\rho I)x^{(t)} + \alpha(A^T A x^* + \rho x^*) - x^* \\ &= (I - \alpha A^T A - \alpha\rho I)(x^{(t)} - x^*). \end{aligned}$$

This means $\|x^{(t+1)} - x^*\|_2 \leq c\|x^{(t)} - x^*\|_2$ for

$$c = \lambda_{\max}(I - \alpha A^T A - \alpha\rho I) = 1 - \alpha\rho - \alpha\lambda_{\min}(A^T A) = 1 - \alpha\rho.$$

ii. Use this to argue that gradient descent on (**??**) converges with *linear complexity*, e.g. the error $f(x^{(t)}) - f(x^*) = O(c^t)$. [4]

Ans. Now we can set up a recursion:

$$\|x^{(t)} - x^*\|_2 \leq c\|x^{(t-1)} - x^*\|_2 \leq c^2\|x^{(t-2)} - x^*\|_2 \leq \cdots \leq c^t\|x^{(0)} - x^*\|_2 = O(c^t).$$

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 $v \in \mathbf{range}(A)$.

Ans. This can basically be done mechanically, by showing

$$Au = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1-1 \\ 1-1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

and picking $x = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$,

$$Ax = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = v.$$

(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 $u^T v = 0$.

Ans. Since $v \in \mathbf{range}(A^T)$, it must be that $v = Ax$ for some x. Then

$$u^T v = u^T A^T x = (Au)^T x = 0.$$

(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.)

[4] 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.

Ans. This is in fact true for *any* A, but for this problem we can see this by breaking down every $x = u + v$ as

$$v_{[1]} = v_{[2]} = \frac{1}{2}(x_{[1]} + x_{[2]})$$

and $u = x - v$. To show that $u \in \mathbf{null}(A)$,

$$Au = Ax - Av = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2}(x_{[1]} - x_{[2]}) \\ \frac{1}{2}(x_{[2]} - x_{[1]}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

To show that $v \in \mathbf{range}(A^T)$, it suffices to say that the range of A are just vectors whose first and second component are the same. But, if we want to be super pedantic, we can just say

$$v = A^T \begin{bmatrix} \beta \\ 0 \end{bmatrix}$$

where $\beta = \frac{1}{2}(x_{[1]} + x_{[2]})$. In matrix form, we can write this as

$$P = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \qquad Q = I - P = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

(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.
Ans. We use the hint to first argue that

$$\nabla f(x) = A^T \underbrace{(Ax - b)}_{z} = A^T z \in \mathbf{range}(A^T).$$

Therefore, $\nabla f(x) = P\hat{u}$ for some $\hat{u}$, and

$$Q\nabla f(x) = \underbrace{QP}_{=\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}} \hat{u} = 0.$$

Therefore, for any t, if $Qx^{(t)} = 0$ then

$$Qx^{(t+1)} = Qx^{(t)} - \alpha Q \nabla f(x) = 0 - 0 = 0.$$

Since $Qx^{(0)} = 0$, by induction we have that $Qx^{(t)} = 0$ for all t.

(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!)
Ans. Following the same scheme as above, we can make the inductive argument that, if $Qx^{(t)} = r$, then

$$Qx^{(t+1)} = \underbrace{Qx^{(t)}}_{=r} - \alpha \underbrace{Q\nabla f(x^{(t)})}_{=0} = r.$$

Since $Qx^{(0)} = 0$, this completes the proof.

(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 (**??**) where $x^{(0)} = Sv^{(0)}$.
Hint: Start by showing that $SS^T\nabla f(x) = \nabla f(x)$.

Ans. Starting with the hint, we have that $\nabla f(x) \in \textbf{range}(A)$, and therefore it must be that the two elements in $\nabla f(x)$ are the same. That is, $z = \nabla f(x)$, $z_{[1]} = z_{[2]}$. Then

$$SS^Tz = \frac{1}{2}\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}z_{[1]}\\z_{[2]}\end{bmatrix} = \frac{1}{2}\begin{bmatrix}z_{[1]}+z_{[2]}\\z_{[1]}+z_{[2]}\end{bmatrix} = z.$$

Now suppose that $x^{(t)} = Sv^{(t)}$. Then, since by chain rule $\nabla g(v) = S^T\nabla f(Sv)$,

$$Sv^{(t+1)} = \underbrace{Sv^{(t)}}_{x^{(t)}} - \alpha S\ \underbrace{\nabla g(v^{(t)})}_{S^T\nabla f(x^{(t)})} = x^{(t)} - \alpha SS^T\nabla f(x^{(t)}) \overset{SS^T\nabla f(x)=\nabla f(x)}{=} x^{(t)} - \alpha\nabla f(x^{(t)}) = x^{(t+1)}.$$

That is, minimizing the reduced objective function $g(v)$ is *equivalent* to minimizing the original function $f(x)$!

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

Ans. This is easier if we start with $AS = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$, which means that

$$\begin{aligned} g(v) &= \frac{1}{2}\left((\frac{1}{\sqrt{2}}v - b_{[1]})^2) + (\frac{1}{\sqrt{2}}v - b_{[2]})^2)\right)\\ &= \frac{1}{2}\left(v^2 - 2v\cdot\underbrace{\frac{b_{[1]}+b_{[2]}}{\sqrt{2}}}_{\beta} + \underbrace{(b_{[1]}^2 + b_{[2]}^2)}_{\gamma}\right)\\ &= \frac{1}{2}v^2 - v\beta + 2\gamma. \end{aligned}$$

for some constant β. Then,

$$\begin{aligned} g(u) - g(v) - g'(v)(u-v) &= \frac{1}{2}(u^2-v^2) - (u-v)\beta - (v-\beta)(u-v)\\ &\overset{\text{simplify}}{=} \frac{1}{2}(u-v)^2 \end{aligned}$$

which is μ-strongly convex with $\mu = 1$.

(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.

Ans. In part (e), we showed that if F is ρ-strongly convex, then gradient descent converges linearly with $c = 1-\alpha\rho$. Here, g is 1-strongly convex, so gradient descent on $g(v)$ converges at a linear rate with $c = 1-\alpha$. To show this, we can simply write the recursion

$$(v^{(t+1)} - v^*) = (v^{(t)} - v^*) - \alpha g'(v^{(t)}) = (v^{(t)} - v^*) - \alpha(v^{(t)} - \beta) \overset{v^*-\beta=0}{=} (1-\alpha)(v^{(t)} - v^*).$$

Since gradient descent on g is equivalent to gradient descent on f, this shows that gradient descent on f is also converging at a linear rate, with the same c.

(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.

Ans. This problem is a bit tough, and really requires some understanding of the big picture. What we have shown in this question is that gradient descent over linear regression doesn't seem to care about directions in the null space of A. So, we can always construct some function $g(v) = f(Sv)$ where $x = Sv + r$, $r \in \textbf{null}(A)$ and $Sv \in \textbf{range}(A^T)$. Then, running gradient descent over g will be equivalent to running gradient descent over f, but where g is strongly convex, with $\mu =$ the smallest nonzero eigenvalue of A^TA. For this value of μ, gradient descent will have convergence rate $O(c^t)$ for $c = 1 - \mu\alpha$.