Problem 1: (20 pnts) Consider gradient descent with fixed step size $\eta > 0$:

$$x_{k+1} = x_k - \eta \nabla f(x_k)$$

Given $a > 0$, make a convex function $f_a(x)$ so that, from *any* initial point, the above converges linearly for all step sizes $\eta < a$, and diverges for all $\eta > a$. Here linear convergence means that there exists some $c < 1$ such that $\|x_k - x_*\| \leq c^k$; of course this c will depend on η and a.

For full credit, you would need to prove both statements: that for $\eta > a$ it diverges from *any* initial point that is not already optimal, and for $\eta < a$ it converges linearly from any initial point.

<u>Problem 1</u>

Take $f(x) = \frac{1}{a} \cdot x^2$

then $x_{k+1} = x_k - \eta \cdot \frac{2}{a} \cdot x_k$

$= \left(1 - \frac{2\eta}{a}\right) x_k$

Note $x^* = 0$

$x_k \to 0$ if $\left|1 - \frac{2\eta}{a}\right| < 1$

$|x_k| \to \infty$ if $\left|1 - \frac{2\eta}{a}\right| > 1$

As $a, \eta > 0$ so

$\left|1 - \frac{2\eta}{a}\right| > 1 \iff 1 - \frac{2\eta}{a} < -1$

$\iff \eta > a$

Similarly $\left|1 - \frac{2\eta}{a}\right| < 1 \iff 1 - \frac{2\eta}{a} > -1$

(as $1 - \frac{2\eta}{a} > 0 \ \forall \eta, a > 0$)

$\iff \eta < a$

In many real-world scenarios our data has millions of dimensions, but a given example has only hundreds of non-zero features. For example, in document analysis with word counts for features, our dictionary may have millions of words, but a given document has only hundreds of unique words. In this question we will make l_2 regularized SGD efficient when our input data is sparse. Recall that in l_2 regularized logistic regression, we want to maximize the following objective (**in this problem we have excluded w_0 for simplicity**):

$$F(\mathbf{w}) = \frac{1}{N}\sum_{j=1}^{N} l(\mathbf{x}^{(j)}, y^{(j)}, \mathbf{w}) - \frac{\lambda}{2}\sum_{i=1}^{d}\mathbf{w}_i^2$$

where $l(\mathbf{x}^{(j)}, y^{(j)}, \mathbf{w})$ is the logistic objective function

$$l(\mathbf{x}^{(j)}, y^{(j)}, \mathbf{w}) = y^{(j)}(\sum_{i=1}^{d}\mathbf{w}_i\mathbf{x}_i^{(j)}) - \ln(1 + \exp(\sum_{i=1}^{d}\mathbf{w}_i\mathbf{x}_i^{(j)}))$$

and the remaining sum is our regularization penalty.

When we do stochastic gradient descent on point $(\mathbf{x}^{(j)}, y^{(j)})$, we are approximating the objective function as

$$F(\mathbf{w}) \approx l(\mathbf{x}^{(j)}, y^{(j)}, \mathbf{w}) - \frac{\lambda}{2}\sum_{i=1}^{d}\mathbf{w}_i^2$$

Definition of sparsity: Assume that our input data has d features, i.e. $\mathbf{x}^{(j)} \in \mathbb{R}^d$. In this problem, we will consider the scenario where $\mathbf{x}^{(j)}$ is sparse. Formally, let s be average number of nonzero elements in each example. We say the data is sparse when $s << d$. In the following questions, **your answer should take the sparsity of $\mathbf{x}^{(j)}$ into consideration when possible**. **Note:** When we use a sparse data structure, we can iterate over the non-zero elements in $O(s)$ time, whereas a dense data structure requires $O(d)$ time.

1. [2 points] Let us first consider the case when $\lambda = 0$. Write down the SGD update rule for $\mathbf{w}_i$ when $\lambda = 0$, using step size η, given the example $(\mathbf{x}^{(j)}, y^{(j)})$.

★ **ANSWER:** The update rule can be written as

$$\mathbf{w}_i^{(t+1)} \leftarrow \mathbf{w}_i^{(t)} + \eta\mathbf{x}_i^{(j)}\left(y^{(j)} - \frac{1}{1 + \exp(-\sum_k \mathbf{w}_k x_k^{(j)})}\right)$$

2. [4 points] If we use a dense data structure, what is the average time complexity to update $\mathbf{w}_i$ when $\lambda = 0$? What if we use a sparse data structure? Justify your answer in one or two sentences.

★ **ANSWER:** The time complexity to calculate $\sum_k \mathbf{w}_k x_k^{(j)}$ is $O(d)$ when the data structure is dense, and $O(s)$ when the data structure is sparse. Note that even if we update $\mathbf{w}_i$ for all i, we only need to calculate $\sum_k \mathbf{w}_k x_k^{(j)}$ once, and then update the $\mathbf{w}_i$ such that $\mathbf{x}_i^{(j)} \neq 0$. So the answer is $\boxed{O(d)}$ for the dense case, and $\boxed{O(s)}$ for the sparse case.

3. [2 points] Now let us consider the general case when $\lambda > 0$. Write down the SGD update rule for $\mathbf{w}_i$ when $\lambda > 0$, using step size η, given the example $(\mathbf{x}^{(j)}, y^{(j)})$.

★ **ANSWER:**

$$\mathbf{w}_i^{(t+1)} \leftarrow \mathbf{w}_i^{(t)} - \eta\lambda\mathbf{w}_i^{(t)} + \eta\mathbf{x}_i^{(j)}\left(y^{(j)} - \frac{1}{1+\exp(-\sum_k \mathbf{w}_k x_k^{(j)})}\right)$$

4. [2 points] If we use a dense data structure, what is the average time complexity to update $\mathbf{w}_i$ when $\lambda > 0$?

★ **ANSWER:** The time complexity is $O(d)$

5. [4 points] Let $\mathbf{w}_i^{(t)}$ be the weight vector after t-th update. Now imagine that we perform k SGD updates on $\mathbf{w}$ using examples $(\mathbf{x}^{(t+1)}, y^{(t+1)}), \cdots, (\mathbf{x}^{(t+k)}, y^{(t+k)})$, where $\mathbf{x}_i^{(j)} = 0$ for every example in the sequence. (i.e. the i-th feature is zero for all of the examples in the sequence). Express the new weight, $\mathbf{w}_i^{(t+k)}$ in terms of $\mathbf{w}_i^{(t)}$, k, η, and λ.

★ **ANSWER:** When $\mathbf{x}_i^{(j)} = 0$,

$$\mathbf{w}_i^{(t+1)} = \mathbf{w}_i^{(t)} - \eta\lambda\mathbf{w}_i^{(t)} = \mathbf{w}_i^{(t)}(1-\eta\lambda)$$

so the answer is

$$\mathbf{w}_i^{(t+k)} = \mathbf{w}_i^{(t)}(1-\eta\lambda)^k$$

6. [6 points] Using your answer in the previous part, come up with an efficient algorithm for regularized SGD when we use a sparse data structure. What is the average time complexity per example? (Hint: when do you need to update $\mathbf{w}_i$?)

Initialize $c_i \leftarrow 0$ for $i \in \{1, 2, \cdots, d\}$
for $j \in \{1, 2, \cdots n\}$ **do**
  $\hat{p} \leftarrow \frac{1}{1+\exp(-\sum_k \mathbf{w}_k x_k^{(j)})}$
  for i *such that* $\mathbf{x}_i^{(j)} \neq 0$ **do**
    $k \leftarrow j - c_i$; auxiliary variable c_i holds the index of last time we see $\mathbf{x}_i^{(j)} \neq 0$
    $\mathbf{w}_i \leftarrow \mathbf{w}_i(1 - \eta\lambda)^k$; apply all the regularization updates
    $\mathbf{w}_i \leftarrow \mathbf{w}_i + \eta \mathbf{x}_i^{(j)} \left(y^{(j)} - \hat{p}\right)$; regularization is done in previous step
    $c_i \leftarrow j$; remember last time we see $\mathbf{x}_i^{(j)} \neq 0$
  end
end

★ **ANSWER:** The idea is to only update $\mathbf{w}_i$ when $\mathbf{x}_i^{(j)} \neq 0$. Before we do the update, we apply all the regularization updates we skipped before, using the answer from previous question. You can checkout Algorithm 1 for details. Using this trick, each update takes $\boxed{O(s)}$ time. (Note: we can use the same trick applies for SGD with l_1 regularization)

Which of the following functions is convex in $x \in \mathbb{R}^n$?

(a) $||x||_{1/2}$

(b) $\sqrt{||x||_2}$

(c) $\max_j \sqrt{x_j}$

(d) $\min_i a_i^T x$

[e] $\log \sum_j \exp(x_j)$

Consider the five functions x, x^2, x^3, x^4, x^5. Which of these functions are convex on $\mathbb{R}$? Which are strictly convex on $\mathbb{R}$? Which are strongly convex on $\mathbb{R}$? Which are strongly convex on $[0.5, 4.5]$? NO EXPLANATIONS REQUIRED! [12 points]

Convex: [3 pts] x, x^2, x^4

Strictly convex: [3 pts] x^2, x^4

Strongly convex: [3 pts] x^2

Strongly convex on $[0.5, 4.5]$**: [3 pts]** x^2, x^3, x^4, x^5

(a) Given a vector $a \in \mathbb{R}^n$ and a scalar $t > 0$, give a closed form equation for the optimum of the following optimization problem

$$\begin{aligned} \min_u \quad & \|u - a\|_2^2 \\ s.t. \quad & \|u\|_\infty \leq t \end{aligned}$$

Solution: This is equivalent to $\forall_i \min_{u_i} |u_i - a_i|^2$ s.t. $u_i \leq t$.
If $a_i > 0$, $u_i = \min(a_i, t)$. If $a_i < 0$, $u_i = -\min(-a_i, t)$.

(c) Let $f(x)$, where $x \in \mathbb{R}^n$ be a convex function (not necessarily smooth). Define $g(x, y) = f(x + y)$ for all $x, y \in \mathbb{R}^n$. Is g a convex function on $\mathbb{R}^{2n}$? If yes, prove it. If no, give a simple counter-example.

Solution: $g(\lambda x_1 + (1 - \lambda)x_2, \lambda y_1 + (1 - \lambda)y_2)$
$= f(\lambda(x_1 + y_1) + (1 - \lambda)(x_2 + y_2))$
$\leq \lambda f(x_1 + y_1) + (1 - \lambda)f(x_2 + y_2) \quad f$ is convex.
$= \lambda g(x_1, y_1) + (1 - \lambda)g(x_2, y_2)$.