Introduction to Machine Learning (0368-3235) - Assignment 3
Lior Erenreich

Homework 3:

Theory Questions:

1. Step-size Perceptron:

1. **(15 points) Step-size Perceptron.** Consider the modification of Perceptron algorithm with the following update rule:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta_t y_t \mathbf{x}_t$$

whenever $\hat{y}_t \neq y_t$ ($\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$ otherwise). Assume that data is separable with margin $\gamma > 0$ and that $\|\mathbf{x}_t\| = 1$ for all t. For simplicity assume that the algorithm makes M mistakes at the first M rounds, afterwhich it has no mistakes. For $\eta_t = \frac{1}{\sqrt{t}}$, show that the number of mistakes step-size Perceptron makes is at most $\frac{4}{\gamma^2}\log(\frac{1}{\gamma})$. (Hint: use the fact that if $x \leq a\log(x)$ then $x \leq 2a\log(a)$). It's okay if you obtain a bound with slightly different constants, but the asymptotic dependence on γ should be tight.

First, reminder – The Perception Algorithm:

- Start with $\boldsymbol{w}_1 = \mathbf{0}$
- Receive example $\boldsymbol{x}_t$
- Predict label $\tilde{y}_t = sign\left(\boldsymbol{w}_t \cdot \boldsymbol{x}_t\right)$
- Receive correct label y_t. Then:
  - If $\tilde{y}_t = y_t$ do nothing.
  - Otherwise update: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + y_t\boldsymbol{x}_t$

Proof:

Let's denote the weight vector at time t by w_t, and let's assume that the algorithm makes M mistakes at the first M rounds, after which it has no mistakes. I need to show that the number of mistakes made by the step-size Perceptron algorithm is at most $\frac{4}{\gamma^2} log\left(\frac{1}{\gamma}\right)$, for $\eta_t = \sqrt{\frac{1}{t}}$. In order to do so, I will follow the proof that was shown on class (of the original form of the algorithm).

After each update (i.e., mistake) we have:

$$w_{t+1} \cdot w^* = (w_t + y_t\eta_t x_t)w^* = w_t \cdot w^* + y_t w^* \cdot x_t\eta_t \geq w_t \cdot w^* + \eta_t\gamma = w_t \cdot w^* + \frac{\gamma}{\sqrt{t}}$$

The last step is due to the assumption that $y_t w^* \cdot x_t \geq \gamma$.

After M mistakes: $w_t \cdot w^* \geq \frac{M\gamma}{\sqrt{M}} = \sqrt{M}\gamma$.

Using

After each mistake:

$$\left||w_{t+1}|\right|_2^2 = \left||w_t + y_t\eta_t x_t|\right|_2^2 = \left||w_t|\right|_2^2 + 2y_t\eta_t x_t + \left||\eta_t x_t|\right|_2^2 \leq \left||w_t|\right|_2^2 + \frac{R^2}{t}$$

Note that $2y_t\eta_t x_t$ is negative since we made a mistake.

After M mistakes: $\left||w_t|\right|_2^2 \leq \frac{MR^2}{t} \approx \log(\sqrt{M})$

Using Cauchy Schwartz:

$$\sqrt{M}\gamma \leq w_M \cdot w^* \leq \left||w_{t+1}|\right|_2^2 \leq \log(\sqrt{M})$$

$$\sqrt{M} \leq \frac{1}{\gamma}\log(\sqrt{M})$$

$$M \leq \frac{4}{\gamma^2}\log^2\left(\frac{1}{\gamma}\right) \blacksquare$$

2. Convex functions:

2. **(15 points) Convex functions.**

(a) Let $f : \mathbb{R}^n \to \mathbb{R}$ a convex function, $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. Show that, $g(\mathbf{x}) = f(A\mathbf{x} + b)$ is convex.

(b) Consider m convex functions $f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})$, where $f_i : \mathbb{R}^d \to \mathbb{R}$. Now define a new function $g(\mathbf{x}) = \max_i f_i(\mathbf{x})$. Prove that $g(\mathbf{x})$ is a convex function. (Note that from (a) and (b) you can conclude that the hinge loss over linear classifiers is convex.)

(c) Let $\ell_{log} : \mathbb{R} \to \mathbb{R}$ be the log loss, defined by

$$\ell_{\log}(z) = \log_2\left(1 + e^{-z}\right)$$

Show that ℓ_{log} is convex, and conclude that the function $f : \mathbb{R}^d \to \mathbb{R}$ defined by $f(\mathbf{w}) = \ell_{log}(y\mathbf{w} \cdot \mathbf{x})$ is convex with respect to $\mathbf{w}$.

Reminder: Definition – Convex Function:

Definition 5.2.2 (convex function)**.** *Let $C \subseteq \mathbb{R}^d$ be a convex set. $f : C \to \mathbb{R}$ is a* convex function *if for any $w_1, w_2 \in C$ and any $\lambda \in [0,1]$ we have that*

$$f(\lambda w_1 + (1-\lambda)w_2) \leq \lambda f(w_1) + (1-\lambda)f(w_2).^1$$

(a) Solution:

To show that $g(x)$ is convex, we need to show that for any two points $x_1, x_2 \in \mathbb{R}^n$ and any $\lambda \in [0,1]$, we have:

$$g(\lambda x_1 + (1-\lambda)x_2) \leq \lambda g(x_1) + (1-\lambda)g(x_2)$$

We start by using the definition of convexity of f to get:

$$f(\lambda(Ax_1 + b) + (1-\lambda)(Ax_2 + b)) \leq \lambda f(Ax_1 + b) + (1-\lambda)f(Ax_2 + b)$$

Simplifying the left side of the equation, we get:

$$f(A(\lambda x_1 + (1-\lambda)x_2) + b) \leq \lambda f(Ax1 + b) + (1-\lambda)f(Ax2 + b)$$

Now, substituting $g(x) = f(Ax + b)$, we get:

$$g(\lambda x_1 + (1-\lambda)x_2) \leq \lambda g(x_1) + (1-\lambda)g(x_2)$$

which is what I needed to show. Therefore, $g(x)$ is convex.■

(b) Solution:
To show that $g(x)$ is convex, I need to show that for any two points $x_1, x_2 \in \mathbb{R}^d$ and any $\lambda \in [0,1]$, we have:

$$g(\lambda x_1 + (1-\lambda)x_2) \leq \lambda g(x_1) + (1-\lambda)g(x_2)$$

First, the definition of $g(x)$:

$$g(\lambda x_1 + (1-\lambda)x_2) \;=\; max\{f_1(\lambda x_1 + (1-\lambda)x_2), \ldots, f_m(\lambda x_1 + (1-\lambda)x_2)\}$$

Next, we can use the fact that for any function $h(x)$, the inequality
$$h(\max\{a,b\}) \leq \max\{h(a), h(b)\}$$
holds. Therefore, we can write:

$$g(\lambda x_1 + (1-\lambda)x_2) \leq max\{f_1(\lambda x_1 + (1-\lambda)x_2), \ldots, f_m(\lambda x_1 + (1-\lambda)x_2)\}$$

$$\leq \max\{\lambda f_1(x_1) + (1-\lambda)f_1(x_2), \ldots, \lambda f_m(x_1) + (1-\lambda)f_m(x_2)\}$$

$$\leq \lambda\, max\{f_1(x_1), \ldots, f_m(x_1)\} + \;(1-\lambda)\, max\{f_1(x_2), \ldots, f_m(x_2)\}$$

$$=\; \lambda g(x_1) + (1-\lambda)g(x_2)$$

where we have used the fact that each $f_i(x)$ is convex, and therefore satisfies the inequality:
$$f_i(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f_i(x_1) + (1-\lambda)f_i(x_2)$$

(c) Solution:
To show that $\ell_{\log}(z)$ is convex, we need to show that for any two points $x_1, x_2 \in \mathbb{R}$ and any $\lambda \in [0,1]$, we have:

$$\ell_{\log}(\lambda x_1 + (1-\lambda)x_2) \;\leq\; \lambda \ell_{\log}(x_1) + (1-\lambda)\ell_{\log}(x_2)$$

We start by defining a function $g(z) \;=\; log_2(1 \;+\; e^z)$.
Let's derive this function twice:

$$g'(z) = \frac{e^{-z}}{\ln(2)\,(e^{-z}+1)} = -\frac{1}{\ln(2)\,(e^{-z}+1)}$$

$$g''(z) = \frac{e^z}{ln(2) * (1 + e^z)^2} \geq 0$$

So I have got that this function is convex because its second derivative is non-negative.

To show that f is convex with respect to w:

$$\ell_{\log}\left(\lambda(yxw_1) + (1-\lambda)(yxw_2)\right) \leq \lambda\ell_{\log}\left(yxw_1\right) + (1-\lambda)\ell_{\log}\left(yxw_2\right)$$

Therefore, we can conclude that $f(w) = \ell_{log}(yw \cdot x)$ is convex with respect to w.

3. Ranking:

3. **(20 points) Ranking**. In this question, we consider a new learning task in which the objective is to rank items. Assume items are elements of $\mathcal{X} \subseteq \mathbb{R}^d$, and you are given a training set of n lists of k items each, and for each list you receive a "label" vector corresponding to the correct ranking of its items. More formally, you receive a training set

$$S = \left\{\left((\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_k^i), \mathbf{y}^i\right)\right\}_{i=1}^n$$

Introduction to Machine Learning (0368-3235) - Assignment 3
Lior Erenreich

such that for all $1 \leq i \leq n$, $\mathbf{y}^i \in \mathbb{R}^k$ assigns a value for each item in $\bar{\mathbf{x}}^i = (\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_k^i)$, interpreted as a ranking of the items. Your goal is to learn a ranking function $h : \mathcal{X}^k \to \mathbb{R}^k$ which correctly ranks the lists of items from S. The *Kendall-Tau* loss between two rankings $\mathbf{y}', \mathbf{y}$ is defined as follows:

$$\Delta(\mathbf{y}', \mathbf{y}) = \frac{2}{k(k-1)} \sum_{j=1}^{k-1} \sum_{r=j+1}^{k} \mathbf{1}\left\{sgn(y'_j - y'_r) \neq sgn(y_j - y_r)\right\}$$

Note that this function averages the total number of pairs of items which are in different order in $\mathbf{y}'$ compared to $\mathbf{y}$. Assume you are trying to learn a linear ranking function, i.e. a function of the form

$$h_{\mathbf{w}}((\mathbf{x}_1, \ldots, \mathbf{x}_k)) = (\mathbf{w} \cdot \mathbf{x}_1, \ldots, \mathbf{w} \cdot \mathbf{x}_k)$$

for some $\mathbf{w} \in \mathbb{R}^d$, and your goal is to minimize the Kendall-Tau loss over S: $\sum_{i=1}^n \Delta(h_{\mathbf{w}}(\bar{\mathbf{x}}^i), \mathbf{y}^i)$. Since this function is hard to optimize, you instead optimize the surrogate "hinge" loss $\sum_{i=1}^n \ell(h_{\mathbf{w}}(\bar{\mathbf{x}}^i), \mathbf{y}^i)$ where:

$$\ell(h_{\mathbf{w}}(\bar{\mathbf{x}}), \mathbf{y}) = \frac{2}{k(k-1)} \sum_{j=1}^{k-1} \sum_{r=j+1}^{k} \max\{0, 1 - sgn(y_j - y_r)\mathbf{w} \cdot (\mathbf{x}_j - \mathbf{x}_r)\}$$

(a) Prove that the hinge loss described above for the ranking objective is convex in $\mathbf{w}$.

(b) Prove that the hinge loss upper-bounds the Kendall-Tau loss, i.e. that $\Delta(h_{\mathbf{w}}(\bar{\mathbf{x}}), \mathbf{y}) \leq \ell(h_{\mathbf{w}}(\bar{\mathbf{x}}), \mathbf{y})$ for all $\mathbf{w} \in \mathbb{R}^d, \bar{\mathbf{x}} \in \mathcal{X}^k, \mathbf{y} \in \mathbb{R}^k$.

(c) Prove that if the data is separable with a margin $\gamma > 0$ (i.e. when there exists $\mathbf{w}^\star \in \mathbb{R}^d$ and $\gamma > 0$ such that $sgn(y_j^i - y_r^i)\mathbf{w}^\star \cdot (\mathbf{x}_j^i - \mathbf{x}_r^i) \geq \gamma$ for all $1 \leq i \leq n$ and all $1 \leq j < r \leq k$), minimizing the hinge loss will result in a ranking function which minimizes the Kendall-Tau loss.

(a) Proof:

First let's define the goal:

Proving $\ell(y', y) = \frac{2}{k(k-1)} \sum_{j=1}^{k-1} \sum_{r=j+1}^{k} \max\{0, 1 - sign(y_j - y_r)w(x_j - x_r)\}$ is convex.

Step 1: I'll use the fact that multiplying a convex function with a scalar, results with a convex function, there for I can remove the multiplication of $\frac{2}{k(k-1)}$.

At this point I am left with:

$$\sum_{j=1}^{k-1} \sum_{r=j+1}^{k} \max\{0, 1 - sign(y_j - y_r)w(x_j - x_r)\}$$

Step 2: I'll use the fact that a sum of convex functions is convex, therefor I can remove the two sums.

At this point I am left with:

$$\max\{0, 1 - sign(y_j - y_r)w(x_j - x_r)\}$$

Step 3: I'll use the fact that a max of convex functions is convex, therefor I can remove the max and since 0 is a constant and therefor is a convex function, at this point I am left with:

$$f(w) = 1 - sign(y_j - y_r)w(x_j - x_r)$$
$$f''(w) = 0 \blacksquare$$

(b) Proof:
I need to show that for all $y \in \mathbb{R}^k, \tilde{x} \in \mathbb{R}^d$ the following holds:

$$\Delta(\mathrm{y}', \mathrm{y}) \leq \ell(y', y)$$

Which means:

$$\frac{2}{k(k-1)} \sum_{j=1}^{k-1} \sum_{r=j+1}^{k} 1\{sign(y'_j - y'_r) \neq sign(y_j - y_r)\}$$
$$\leq \frac{2}{k(k-1)} \sum_{j=1}^{k-1} \sum_{r=j+1}^{k} \max\{0, 1 - sign(y_j - y_r)w(x_j - x_r)\}$$

Note that $w(x_j - x_r) = (y'_j - y'_r)$:

$$\frac{2}{k(k-1)} \sum_{j=1}^{k-1} \sum_{r=j+1}^{k} 1\{sign(y'_j - y'_r) \neq sign(y_j - y_r)\}$$
$$\leq \frac{2}{k(k-1)} \sum_{j=1}^{k-1} \sum_{r=j+1}^{k} \max\{0, 1 - sign(y_j - y_r)(y'_j - y'_r)\}$$

I can remove the multiplication with the constant:

$$\sum_{j=1}^{k-1} \sum_{r=j+1}^{k} 1\{sign(y'_j - y'_r) \neq sign(y_j - y_r)\}$$
$$\leq \sum_{j=1}^{k-1} \sum_{r=j+1}^{k} \max\{0, 1 - sign(y_j - y_r)w(x_j - x_r)\}$$

Let's divide this inequation to different cases:

Case 1: $sign(y_j - y_r), sign(y'_j - y'_r) > 0$:

$$1\{sign(y'_j - y'_r) \neq sign(y_j - y_r)\} = 0 \leq \max\{0, 1 - sign(y_j - y_r)(y'_j - y'_r)\} < 1$$

Case 2: $sign(y_j - y_r) > 0, sign(y'_j - y'_r) < 0$:

$$1\{sign(y'_j - y'_r) \neq sign(y_j - y_r)\} = 1$$

$$\max\{0, 1 - sign(y_j - y_r)(y'_j - y'_r)\} = 1 - (y'_j - y'_r) > 1$$

So overall I got for this case that the following holds:

$$1\{sign(y'_j - y'_r) \neq sign(y_j - y_r)\} \leq \max\{0, 1 - sign(y_j - y_r)(y'_j - y'_r)\}$$

Case 3: $sign(y_j - y_r) < 0, sign(y'_j - y'_r) > 0$:

$$1\{sign(y'_j - y'_r) \neq sign(y_j - y_r)\} = 1$$

$$\max\{0, 1 - sign(y_j - y_r)(y'_j - y'_r)\} = \max\{0, 1 + (y'_j - y'_r)\} \geq 1$$

So overall I got for this case that the following holds:

$$1\{sign(y'_j - y'_r) \neq sign(y_j - y_r)\} \leq \max\{0, 1 - sign(y_j - y_r)(y'_j - y'_r)\}$$

Case 4: $sign(y_j - y_r), sign(y'_j - y'_r) < 0$:

$$1\{sign(y'_j - y'_r) \neq sign(y_j - y_r)\} = 0 \leq \max\{0, 1 - sign(y_j - y_r)(y'_j - y'_r)\} < 1 \blacksquare$$

(c) Proof:
We are given that:

$$sign(y_j - y_r)w^*(x_j - x_r) \geq \gamma > 0$$

Note that $w(x_j - x_r) = (y'_j - y'_r)$:

$$sign(y_j - y_r)(y'_j - y'_r) \geq \gamma > 0$$

Kendall Tau loss:

$$\Delta(y', y) = \frac{2}{k(k-1)} \sum_{j=1}^{k-1} \sum_{r=j+1}^{k} 1\{sign(y'_j - y'_r) \neq sign(y_j - y_r)\}$$

So let's divide into cases again:

Case 1: $sign(y_j - y_r) < 0$:

In order for $sign(y_j - y_r)(y'_j - y'_r) \geq \gamma > 0$ to hold, the following must hold:

$$(y'_j - y'_r) < 0$$

So:

$$1\{sign(y'_j - y'_r) \neq sign(y_j - y_r)\} = 0$$

Case 2: $sign(y_j - y_r) > 0$:

In order for $sign(y_j - y_r)(y_j' - y_r') \geq \gamma > 0$ to hold, the following must hold:

$$(y_j' - y_r') > 0$$

So, again:

$$1\{sign(y_j' - y_r') \neq sign(y_j - y_r)\} = 0 \blacksquare$$

4. Gradient Descent on Smooth Functions:

4. **(15 points) Gradient Descent on Smooth Functions.** We say that a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is β-smooth if for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{\beta}{2}\|\mathbf{x} - \mathbf{y}\|^2$$

In words, β-smoothness of a function f means that at every point $\mathbf{x}$, f is upper bounded by a qaudratic function which coincides with f at $\mathbf{x}$.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a β-smooth and non-negative function (i.e., $f(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$). Consider the (non-stochastic) gradient descent algorithm applied on f with constant step size $\eta > 0$:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t)$$

Assume that gradient descent is initialized at some point $\mathbf{x}_0$. Show that if $\eta < \frac{2}{\beta}$ then

$$\lim_{t \to \infty} \|\nabla f(\mathbf{x}_t)\| = 0$$

(Hint: Use the smoothness definition with points $\mathbf{x}_{t+1}$ and $\mathbf{x}_t$ to show that $\sum_{t=0}^{\infty} \|\nabla f(\mathbf{x}_t)\|^2 < \infty$ and recall that for a sequence $a_n \geq 0$, $\sum_{n=1}^{\infty} a_n < \infty$ implies $\lim_{n \to \infty} a_n = 0$. Note that f is not assumed to be convex!)

Solution:

$$f(x_{t+1}) - f(x_t) \leq \Delta f(x_t)(x_{t+1} - x_t) + \frac{\beta}{2}||x_{t+1} - x_t||^2 = \left(\frac{\beta\eta^2}{2} - \eta\right)||\Delta f(x_t)||^2$$

Note that $\left(\frac{\beta\eta^2}{2} - \eta\right) < 0$ because:

$$\eta < \frac{2}{\beta}$$

$$\frac{\beta\eta^2}{2} - \eta < \frac{\eta^2}{\eta} - \eta = 0$$

So I get that the following holds:

$$\sum_{t=1}^{T}||\Delta f(x_i)||^2 \leq \left(\eta - \frac{\beta\eta^2}{2}\right)^{-1}\sum_{t=1}^{T} f(x_i) - f(x_{t+1}) = \left(\eta - \frac{\beta\eta^2}{2}\right)^{-1} f(x_0) - f(x_{T+1})$$

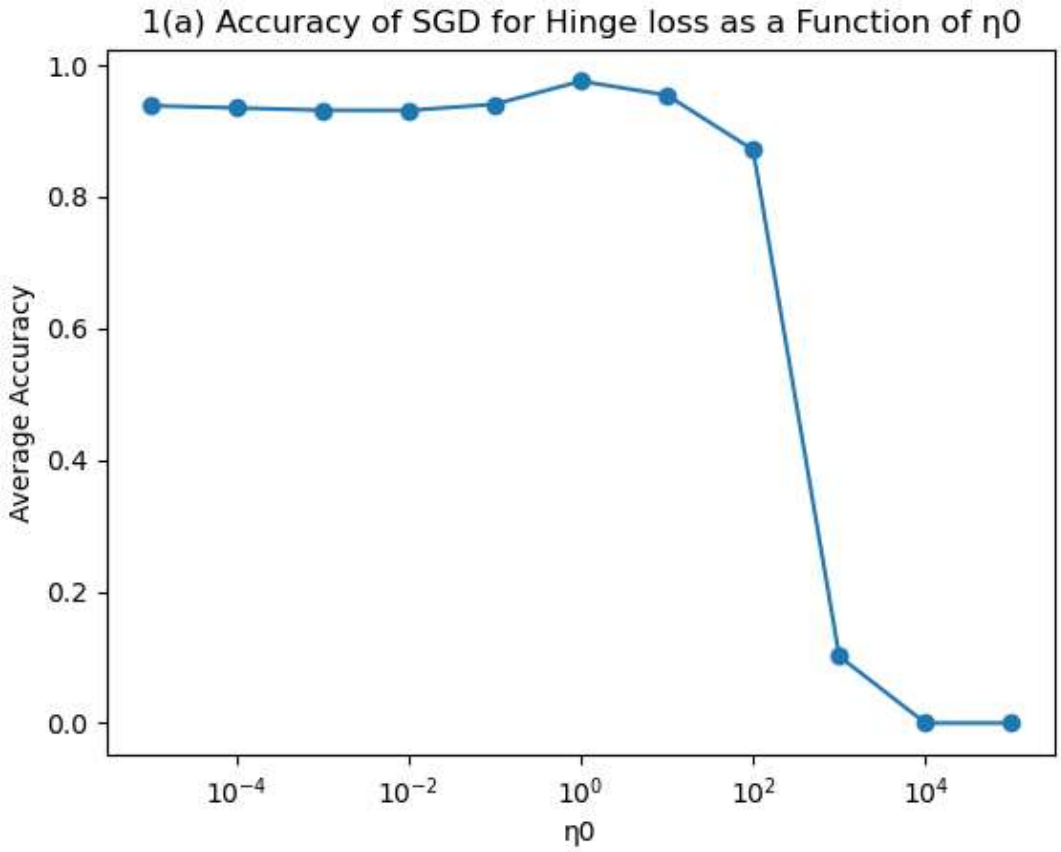
$$\leq \left(\eta - \frac{\beta\eta^2}{2}\right)^{-1} f(x_0) < \infty$$

And $\lim_{T\to\infty}\left||\Delta f(x_i)|\right|^2 = 0$

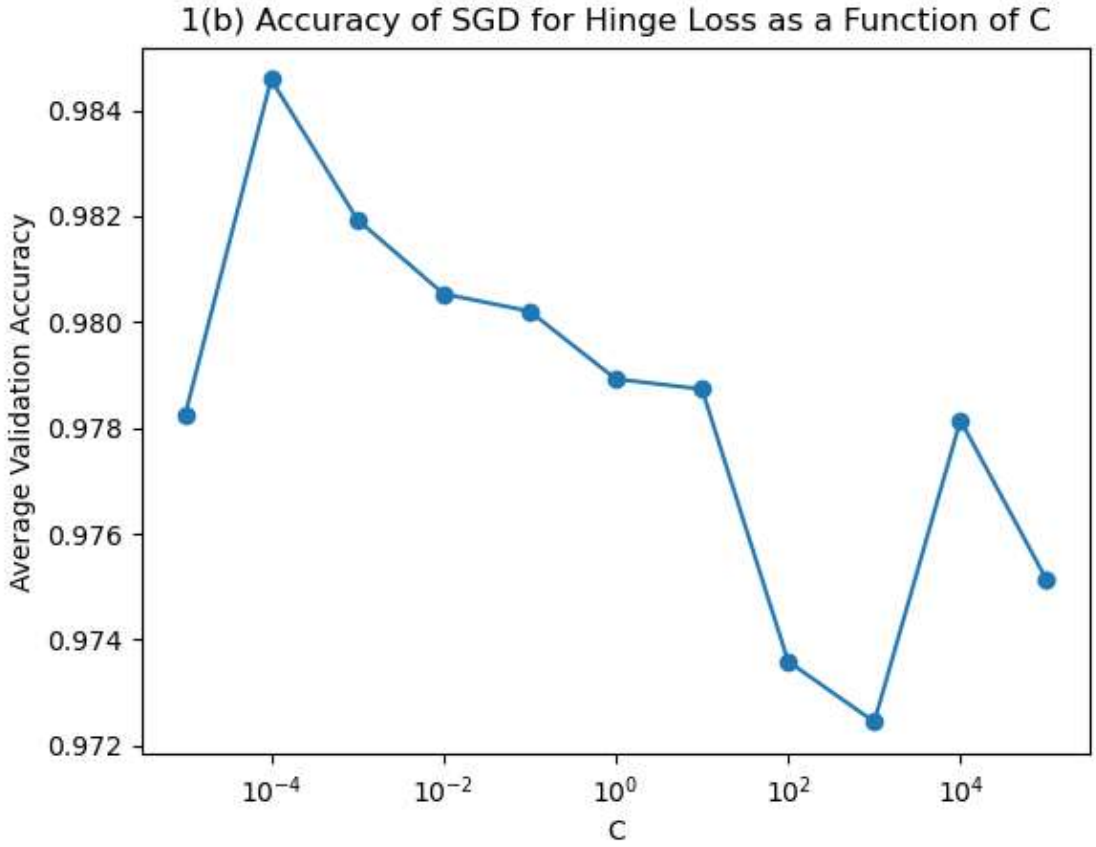
Introduction to Machine Learning (0368-3235) - Assignment 3
Lior Erenreich

Programming Assignment:

1. (a)

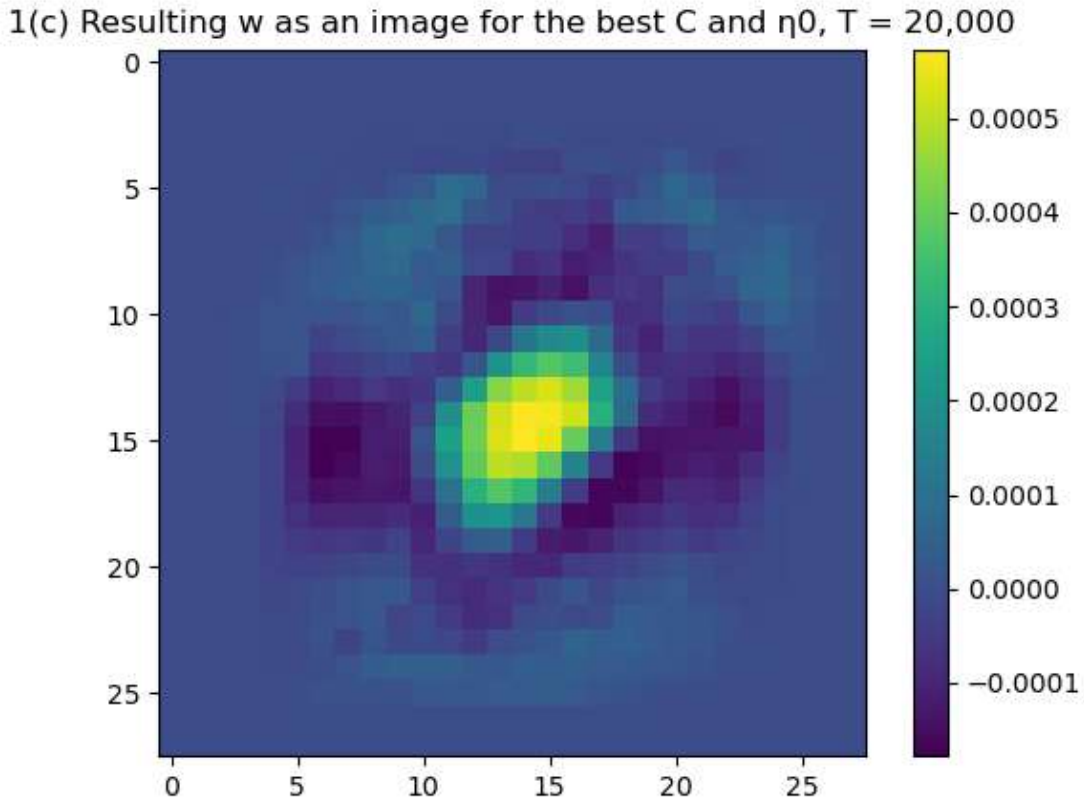


(b)



(c) The resulting image:

Introduction to Machine Learning (0368-3235) - Assignment 3
Lior Erenreich

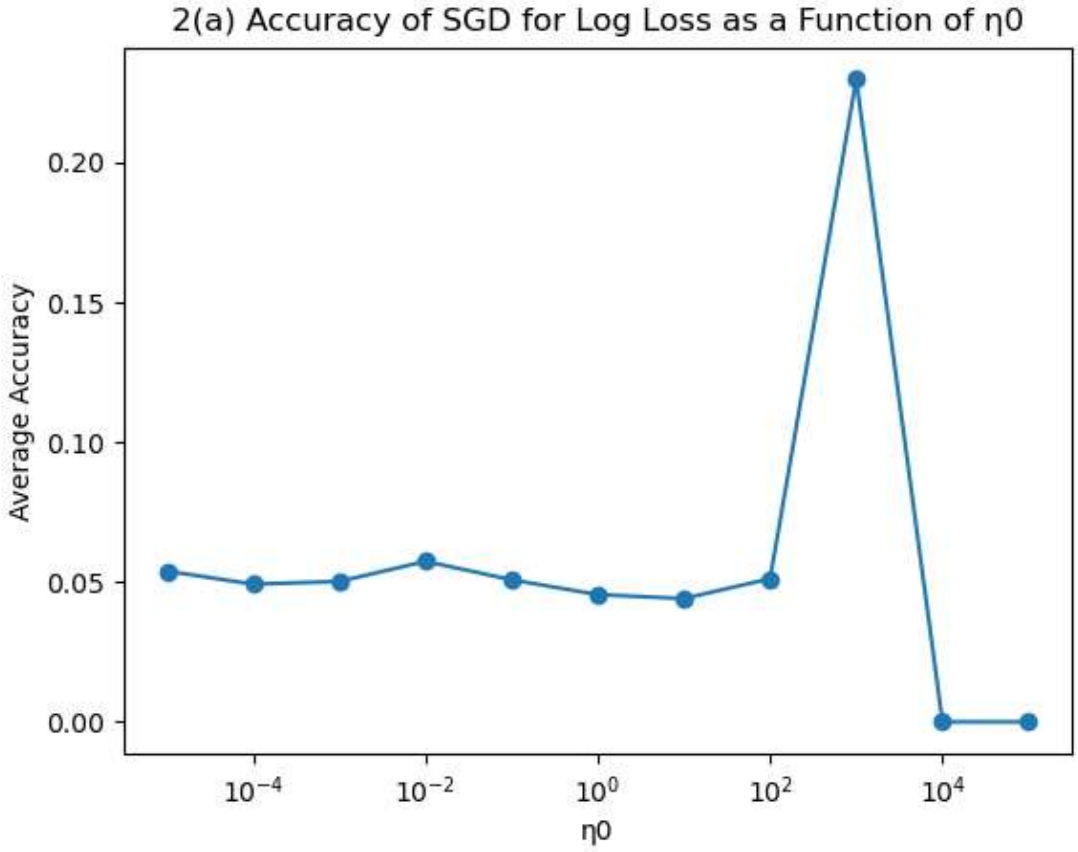


An intuitive interpretation: the image obtained from the resulting w shows a pattern of pixel values that form a circle-like shape in the center of the image, with pixel values of around 0.0006. Around this central circle, there is a gradual decrease in pixel values forming other concentric circles of pixel values of 0.0005, 0.0004, and so on, until it reaches values close to 0. The outermost pixels have values of around -0.0002.
The image appears to be a tilted ellipse with the major axis oriented from the bottom left to the top right. The pixel values represent the weights assigned to each pixel in the image, indicating their contribution to the classification task. The ellipse-like shape in the center of the image indicates that the pixels in that region are highly informative for the classification task, while the gradually decreasing values in the surrounding circles indicate a decreasing contribution to the classification. The ellipse's outer pixels with negative values could indicate a negative correlation with the classification task.

(d) As printed by running my code, the accuracy of the best classifier on the test is 99.23234390992836%.

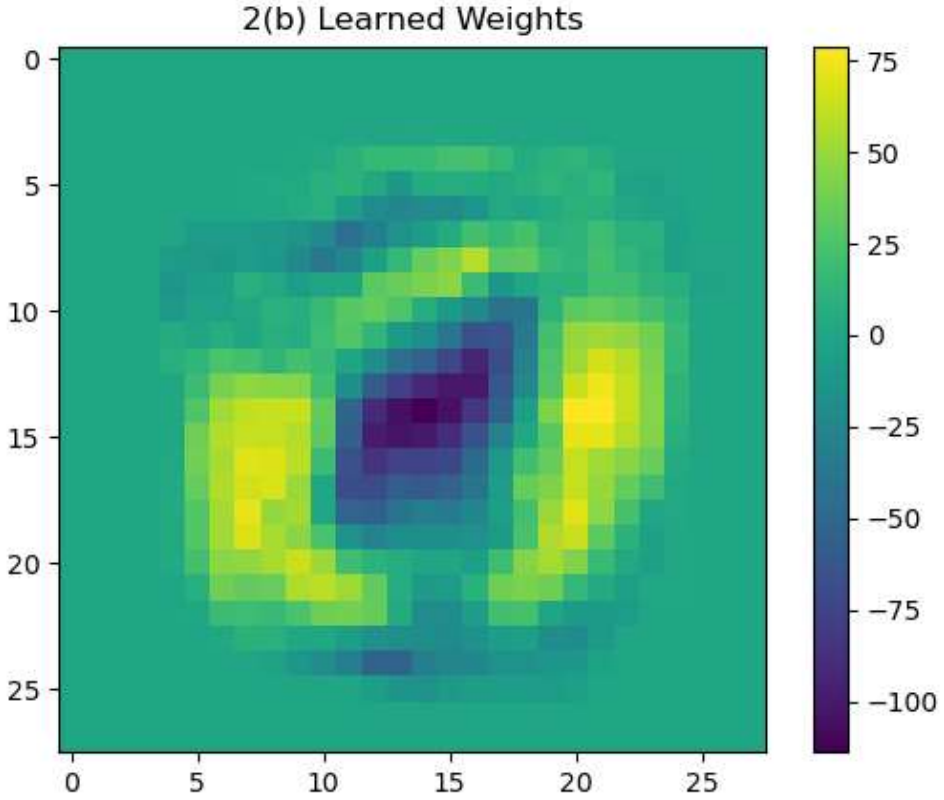
2. (a)

Introduction to Machine Learning (0368-3235) - Assignment 3
Lior Erenreich



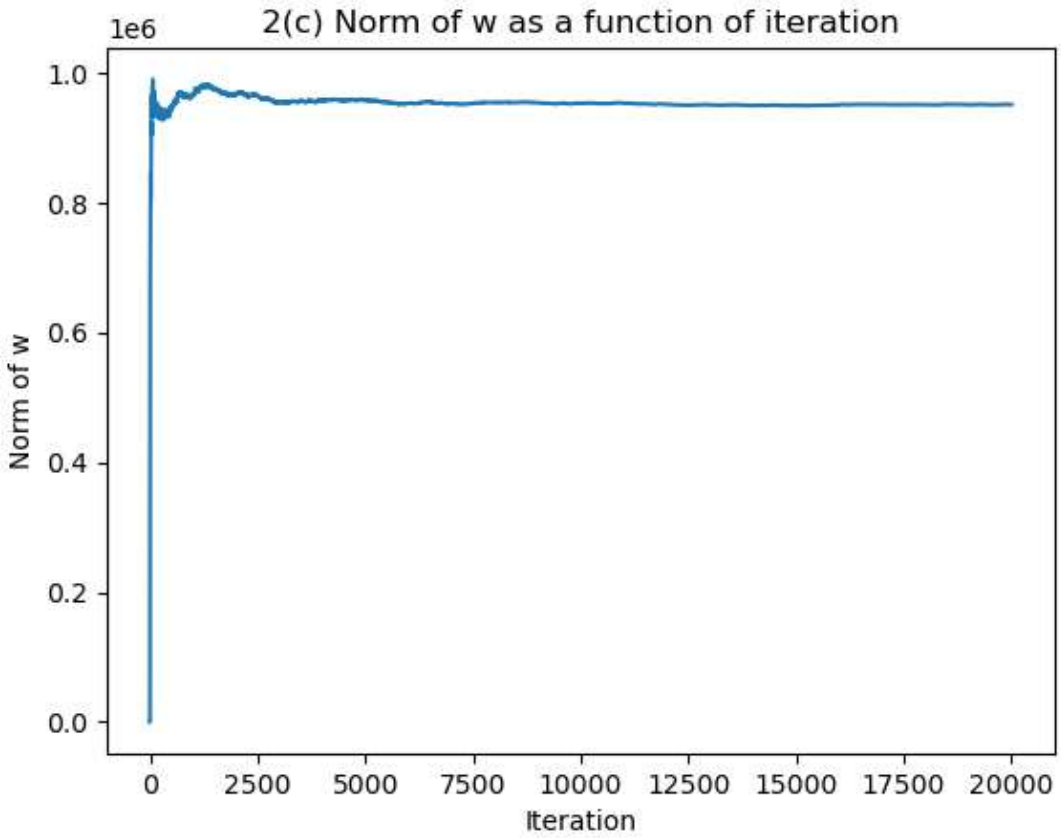
(b) The resulting *w* as an image:

Introduction to Machine Learning (0368-3235) - Assignment 3
Lior Erenreich



As printed by running my code, the accuracy on the test set: 3.50%
(c) Plot of the norm of w as a function of the iteration:

Introduction to Machine Learning (0368-3235) - Assignment 3
Lior Erenreich



Explanation:
The norm of the weight vector *w* is a measure of the complexity of the model. As the model is trained, the weights are updated in order to minimize the loss function, which leads to a decrease in the norm of the weight vector. In the plot, we can observe an initial rapid increase in the norm of the weight vector, followed by a decrease and then a more gradual oscillation around a certain value.
The rapid increase in the norm of the weight vector at the beginning of training is due to the random initialization of the weights, which are far from their optimal values. As training progresses, the weights are updated and the model becomes better at fitting the training data. This leads to a decrease in the norm of the weight vector.
The weight vector converges to a stable value because as the algorithm iterates, the updates to the weight vector become smaller and smaller. This happens because the learning rate is set to decrease over time, which means that the step sizes taken towards the optimal solution become progressively smaller. As the updates get smaller, the weight vector gets closer to the optimal solution and eventually stabilizes around it.