Introduction to Machine Learning-Exercise 3

Odai Agbaria

Theory Questions

1. **(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.

Proof:

f *is a convex function, then for every* $w_1, w_2 \in R^n, \lambda\epsilon[0,1]$ *it holds that*:

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

Let's prove that $g = f(Ax + b)$ is also convex.

Let $w_1, w_2 \in R^n, \lambda\epsilon[0,1]$ then by definition of g:

$$g(\lambda w_1 + (1-\lambda)w_2) = f(A(\lambda w_1 + (1-\lambda)w_2) + b) = f(\lambda Aw_1 + (1-\lambda)Aw_2 + b)$$

$$= f(\lambda Aw_1 + (1-\lambda)Aw_2 + \lambda b + (1-\lambda)b) = f(\lambda(Aw_1 + b) + (1-\lambda)(Aw_2 + b))$$

Then by convexity of f(consider that the vectors are: $(Aw_1 + b), (Aw_2 + b) \in R^n$, *and with the same* λ):

$$g(\lambda w_1 + (1-\lambda)w_2) = f(\lambda(Aw_1 + b) + (1-\lambda)(Aw_2 + b)) \leq \lambda f(Aw_1 + b) + (1-\lambda)f(Aw_2 + b)$$
$$= \lambda g(w_1) + (1-\lambda)g(w_2)$$

Thus, we conclude that g also 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.)

Proof:

Let $w_1, w_2 \in R^n, \lambda\epsilon[0,1]$ then by definition of g:

$$g(\lambda w_1 + (1-\lambda)w_2) = max_i f_i(\lambda w_1 + (1-\lambda)w_2)$$

Denote $j = argmax_j f_j(\lambda w_1 + (1-\lambda)w_2)$. Then:

$$g(\lambda w_1 + (1-\lambda)w_2) = max_i f_i(\lambda w_1 + (1-\lambda)w_2) = f_j(\lambda w_1 + (1-\lambda)w_2)$$
$$\leq \lambda f_j(w_1) + (1-\lambda)f_j(w_2) \leq \lambda max_i f_i(w_1) + (1-\lambda)max_i f_i(w_2)$$
$$= \lambda g(w_1) + (1-\lambda)g(w_2)$$

By convexity of fj

Where the inequality in red is immediately by definition of max because:

$f_j(w_1) \leq max_i f_i(w_1)\ and f_j(w_2) \leq max_i f_i(w_2)$.

Notice that i which maximizes $f_i(w_1)$ doesn't have to be the same as i which maximizes $f_i(w_2)$, but the above inequality holds independently for each i.

Thus, we conclude that g 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 $\mathbf{w}$.

Proof:

Let's prove that the second derivative of $l_{log}(z)$ is positive then we will conclude that $l_{log}(z)$ is convex, denote: $f(z) = l_{log}(z) = log_2(1 + e^{-z})$

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

$$f''(z) = \frac{-1}{\ln(2)} * \left(\frac{-e^{-z} * (1 + e^{-z}) - e^{-z} * -e^{-z}}{(1 + e^{-z})^2}\right) = \frac{-1}{\ln(2)} * \left(\frac{-e^{-z} - \cancel{e^{-2z}} + \cancel{e^{-2z}}}{(1 + e^{-z})^2}\right)$$

$$= \frac{e^{-z}}{\ln(2)(1 + e^{-z})^2} = \frac{(+)}{(+)} > 0$$

Thus, we conclude that f is convex.

Now, define $g(w) = l_{log}(y\boldsymbol{wx})$.

Notice that: $g(w) = l_{log}(y\boldsymbol{wx}) = l_{log}(\boldsymbol{yxw})$.

Let $w_1, w_2 \in R^n, \lambda \epsilon [0,1]$, then by definition of g and the note above:

$$g(\lambda w_1 + (1 - \lambda)w_2) = l_{log}(\boldsymbol{yx}(\lambda w_1 + (1 - \lambda)w_2)) = l_{log}(\lambda \boldsymbol{yxw_1} + (1 - \lambda) y\boldsymbol{xw_2})$$

Then by convexity of $f(z) = l_{log}(z)$ (consider that the vectors now are: $(\boldsymbol{yxw_1}), (\boldsymbol{yxw_2}) \in R^d, and\ with\ the\ same\ \lambda$):

$$g(\lambda w_1 + (1 - \lambda)w_2) = l_{log}(\lambda \boldsymbol{yxw_1} + (1 - \lambda) y\boldsymbol{xw_2}) \leq \lambda l_{log}(\boldsymbol{yxw_1}) + (1 - \lambda) l_{log}(\boldsymbol{yxw_2})$$
$$= \lambda g(\boldsymbol{w_1}) + (1 - \lambda) g(\boldsymbol{w_2})$$

thus, we conclude that $g(w) = l_{log}(y\boldsymbol{wx})$ is convex with respect to w.

■

2. **(10 points) Hinge loss with linearly separable data.** Consider a setup of learning linear classifiers over a dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, +1\}$ for all i. Assume the data is linearly separable, i.e., given a training set there exists $\mathbf{w}^* \in \mathbb{R}^d$ such that $y_i \mathbf{w}^* \cdot \mathbf{x}_i > 0$ for all i (note the strict inequality; assume no bias for simplicity). Recall the definition of the *hinge loss* given in lecture #6:

$$\ell_{hinge}(r) = \max\{0, 1 - r\}.$$

We would like to show that in the linearly separable case, minimizing the hinge loss over the training data will yield a classifier with optimal zero-one loss. Formally, let

$$\mathbf{w}^*_{hinge} = \text{argmin}_{\mathbf{w} \in \mathbb{R}^d} \left\{ \sum_{i=1}^{n} \ell_{hinge}(y_i \mathbf{w} \cdot \mathbf{x}_i) \right\}.$$

Show that $sign(\mathbf{w}^*_{hinge} \cdot \mathbf{x}_i) = y_i$ for all i, meaning $\mathbf{w}^*_{hinge}$ achieves optimal zero-one loss.
(**Hint:** First show that the hinge loss upper bounds the zero-one loss. Then consider the hinge loss of $c\mathbf{w}^*$ for some constant $c > 0$. What happens to the hinge loss as $c \to \infty$?)

Proof:

We will prove the desired claim using the hint.

First, let's show that the hinge loss upper bounds the zero-one loss:

Reminder: the zero-one loss as defined in class:

$$l_{zo}(r) = \begin{cases} 1 \; if \; r \leq 0 \\ 0 \; if \; r > 0 \end{cases}$$

Let $w \in R^d$. We want to show that $l_{hinge}(y_i w x_i) \geq l_{zo}(y_i w x_i)$.

if $y_i w x_i > 0$ then by definition $l_{zo}(y_i w x_i) = 0$ and then immediately by the definition of the hinge loss it holds that $l_{hinge}(y_i w x_i) = \max(0, 1 - y_i w x_i) \geq 0 = l_{zo}(y_i w x_i)$.

Otherwise, $y_i w x_i \leq 0$ then $l_{hinge}(y_i w x_i) = \max(0, 1 - y_i w x_i) = \max\big(0, 1 - (negative \; or \; 0)\big) = 1 - 1 - y_i w x_i \geq 1 = l_{zo}(y_i w x_i)$.

So, we got that $l_{hinge}(y_i w x_i) \geq l_{zo}(y_i w x_i)$.

Now, we want to show that $sign\big(w^*{}_{hinge} x_i\big) = y_i$, for that let's consider $\boldsymbol{w}^{\wedge} = \boldsymbol{cw}^* \; for \; some \; constant \; c > 0, where \; w^* is \; the \; vector \; which \; holds \; that \; y_i w^* x_i > 0 \; for \; all \; i$, which existence is guaranteed in the question as we are in the realizable case.

Then it holds that:

$$l_{hinge}\big(y_i w^{\wedge} x_i\big) = \max\{0, 1 - y_i w^{\wedge} x_i\} = \max\{0, 1 - y_i c w^* x_i\} = \max\{0, 1 - c y_i w^* x_i\}$$

Since $y_i w^* x_i > 0$ for every i, and c>0 then $c y_i w^* x_i > 0$, and thus if $c \to \infty$ then $c y_i w^* x_i \to \infty$ and then $l_{hinge}\big(y_i w^{\wedge} x_i\big) = \max\{0, 1 - c y_i w^* x_i\} = \max\{0, 1 - \infty\} = \max\{0, -\infty\} = 0$, which holds for every i.

Thus, we have proved that there exists a $\boldsymbol{w}^{\wedge} = \boldsymbol{cw}^{*}$ where $l_{hinge}(y_i w^{\wedge} x_i) = 0$ for every i, and since we proved that hinge loss upper bounds the zero-one loss for every w, in particular this holds for $\boldsymbol{w}^{\wedge}$, thus we conclude that the zero-one loss: $l_{zo}(y_i w^{\wedge} x_i) = 0$ as desired(this means that $sign(w^{*}{}_{hinge} x_i) = y_i \; where \; w^{*}{}_{hinge} = w^{\wedge}$).

■

3. **(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!)

Proof:

First, by the smoothness of the function f with points x_t, x_{t+1} it holds that:

$$f(x_{t+1}) \leq f(x_t) + \nabla f(x_t)^T(x_{t+1} - x_t) + \frac{\beta}{2}||x_t - x_{t+1}||^2 \quad ★★$$

by the way gradient descent works we get that:

$$x_t - x_{t+1} = \eta \nabla f(x_t) \quad , \quad x_{t+1} - x_t = -\eta \nabla f(x_t) \; ★$$

Compensating ★ in ★★ we get:

$$f(x_{t+1}) \leq f(x_t) + \nabla f(x_t)^T - \eta \nabla f(x_t) + \frac{\beta}{2}||\eta \nabla f(x_t)||^2$$

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

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

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

Since $\eta < \frac{2}{\beta}$ then: $\left(1 - \eta \frac{\beta}{2}\right) > \left(1 - \frac{2}{\beta} * \frac{\beta}{2}\right) = 0$, then we get that:

$\eta - \eta^2\frac{\beta}{2} = \eta\left(1 - \eta\frac{\beta}{2}\right) = (+) * (+) > 0$,thus if we divide the above inequality by $\left(\eta - \eta^2\frac{\beta}{2}\right) > 0$ we get:

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

Now for every T>0 it holds that:

$$\sum_{i=0}^{T}\left||\nabla f(x_t)|\right|^2 \leq \sum_{i=0}^{T}\frac{f(x_t) - f(x_{t+1})}{\left(\eta - \eta^2\frac{\beta}{2}\right)} = \frac{1}{\left(\eta - \eta^2\frac{\beta}{2}\right)} * \sum_{i=0}^{T} f(x_t) - f(x_{t+1})$$

$$= \frac{1}{\left(\eta - \eta^2\frac{\beta}{2}\right)} * \left(f(x_0) - f(x_{t+1})\right) \leq \frac{f(x_0)}{\left(\eta - \eta^2\frac{\beta}{2}\right)}$$

Since f is non-negative

In particular, it holds that:

$$\sum_{i=0}^{\infty}\left||\nabla f(x_t)|\right|^2 \leq \frac{f(x_0)}{\left(\eta - \eta^2\frac{\beta}{2}\right)} < \infty$$

Note that $\left||\nabla f(x_t)|\right|^2 \geq 0$ by its definition, thus we conclude that $lim_{t\to\infty}\left||\nabla f(x_t)|\right|^2 = 0$. ■

4. **(10 points) Solving hard SVM.** Consider two distinct points $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$ with labels $y_1 = 1$ and $y_2 = -1$. Compute the hyperplane that Hard SVM will return on this data, i.e., give explicit expressions for $\boldsymbol{w}$ and b as functions of $\mathbf{x}_1$, $\mathbf{x}_2$.
(Hint: Solve the dual problem by transforming it to an optimization problem in a single variable. Use your solution to the dual to obtain the primal solution).

Solution:

The primal problem is: (we compensate $y_2 = -1\ and\ y_1 = 1$ in the constraints below)

$$min_{w,b}0.5||w||^2$$

$$s.t\ wx_1 + b \geq 1\ and\ wx_2 + b \geq 1$$

We have seen in lecture 7 that given α_1, α_2 as a solution of the dual problem then w which is the solution of the primal problem above is given by: (also here we compensate $y_2 = -1\ and\ y_1 = 1$)

$$w = \alpha_1 x_1 - \alpha_2 x_2 \quad ★$$

So, first lets solve the dual problem, which is given as follows:

$$max_{\alpha_1,\alpha_2}(\alpha_1 + \alpha_2 - 0.5\sum_{i=1}^{2}\sum_{j=1}^{2}\alpha_i\alpha_j y_i y_j x_i x_j)$$

$$s.t\ \alpha_1, \alpha_2 \geq 0\ and\ \alpha_1 - \alpha_2 = 0$$

Since $\alpha_1 - \alpha_2 = 0 \rightarrow\ \alpha_1 = \alpha_2$ then the dual problem can be simplified to single variable optimization problem, denote $\alpha = \alpha_1 = \alpha_2$ then the dual optimization problem is as follows:

$$max_{\alpha}(2\alpha - 0.5(\alpha^2 {x_1}^2 + \alpha^2 {x_2}^2 - 2\alpha^2 x_2 x_1))$$

$$s.t\ \alpha_1, \alpha_2 \geq 0$$

Above we used $\alpha = \alpha_1 = \alpha_2$ and the fact that $y_2 = -1\ and\ y_1 = 1$. So now we get:

$$max_{\alpha}(2\alpha - 0.5 * \alpha^2 \left|\left|x_1 - x_2\right|\right|^2)$$

$$s.t\ \alpha \geq 0$$

To find α which maximizes the above, we differentiate w.r.t to α and then equals to zero:

$$2 - 0.5 * 2\alpha * \left|\left|x_1 - x_2\right|\right|^2 = 0$$

$$\alpha = \frac{2}{\left|\left|x_1 - x_2\right|\right|^2}$$

Thus, due to ★ we get the next:

$$w = \alpha_1 x_1 - \alpha_2 x_2 = \alpha(x_1 - x_2) = \frac{2(x_1 - x_2)}{\left|\left|x_1 - x_2\right|\right|^2}$$

And then as we saw in the same lecture, using the complementary slackness we can recover b, using that for every support vector (x_i, y_i) which $\alpha_i \neq 0$ then it holds that:

$$1 = y_i(wx_i + b)$$

Using $x_1, y_1 = 1$ we get:

$$1 = wx_1 + b \rightarrow b = 1 - wx_1 = 1 - \frac{2x_1(x_1 - x_2)}{\left|\left|x_1 - x_2\right|\right|^2}$$

To sum up, the final answer is:

$$\boldsymbol{w = \frac{2(x_1 - x_2)}{\left|\left|x_1 - x_2\right|\right|^2}, b = 1 - \frac{2x_1(x_1 - x_2)}{\left|\left|x_1 - x_2\right|\right|^2}}$$

5. **(15 points) ℓ^2 penalty.** Consider the following problem:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2$$
$$\text{s.t. } y_i(\boldsymbol{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i = 1, \ldots, n$$

(a) Show that a constraint of the form $\xi_i \geq 0$ will not change the problem. Meaning, show that these non-negativity constraints can be removed. That is, show that the optimal value of the objective will be the same whether or not these constraints are present.

Proof:

Assume by contradiction that a constraint of the form $\xi_i \geq 0$ will change the problem, which means that the optimal value of the objective won't be the same.

Notice that this means that the optimal value of the objective without the non-negativity constraints is smaller than the optimal value with them, since adding constraints only makes the problem harder, so it cant give us a better optimal value(a smaller value in this case).

Since the optimal value of the objective without the non-negativity constraints is smaller than the optimal value with them, then there exists at least one $\xi_j < 0$ in the case without the constraints which optimizes the value - makes it smaller.

But let's notice that in this case:

The meaning of this transition is that the R.H.S is the optimal value of the case without the non-negativity constraints.

$$min_{w,b,\xi}\left(0.5||w||^2 + 0.5C\sum_{i=1}^{n}\xi_i^2\right) = 0.5||w||^2 + 0.5C\sum_{i=1}^{n}\xi_i^2$$
$$= 0.5||w||^2 + 0.5C\xi_j^2 + 0.5C\sum_{i=1,i\neq j}^{n}\xi_i^2 > 0.5||w||^2 + 0.5C * 0^2 + 0.5C\sum_{i=1,i\neq j}^{n}\xi_i^2$$

Meaning that replacing $\xi_j < 0$ with $\xi_j = 0$ will give us a smaller value, and we can do that since:

$$y_j(w^T x_j + b) \geq 1 - \xi_j = 1 - (-) \geq 1 = 1 - 0$$

Meaning that if we replace $\xi_j < 0$ with $\xi_j = 0$ we will be holding the rest of the constraints of the problem. And that is a contradiction to the fact that the value of the objective is optimal-minimal. ■

(b) What is the Lagrangian of this problem?

Answer:

We want to solve the problem which our function can be defined as:

$$f(w,b,\xi) = 0.5||w||^2 + 0.5C\sum_{i=1}^{n}\xi_i^2$$

with the constraints:

$$y_i(w^T x_i + b) \geq 1 - \xi_i \rightarrow -y_i(w^T x_i + b) + 1 - \xi_i \leq 0$$

So, we want to minimize $f(w,b,\xi)$ with constraints $r(w,b,\xi) = 1-y_i(w^T x_i + b) - \xi_i \leq 0$

Then the Lagrangian of the problem is:

$$L(w,b,\xi,\alpha) = f(w,b,\xi) + \sum_{i=1}^{n} \alpha_i r(w,b,\xi)$$

$$= 0.5\left|\left|w\right|\right|^2 + 0.5C\sum_{i=1}^{n} {\xi_i}^2 + \sum_{i=1}^{n} \alpha_i(1-y_i(w^T x_i + b) - \xi_i)$$

$$= 0.5\left|\left|w\right|\right|^2 + 0.5C\left|\left|\xi\right|\right|^2 - \alpha\xi + \sum_{i=1}^{n} \alpha_i(1-y_i(w^T x_i + b))$$

(c) Minimize the Lagrangian with respect to $\boldsymbol{w}, b, \boldsymbol{\xi}$ by setting the derivative with respect to these variables to 0.

Solution:

We want to solve the following problem:

$$min_{w,b,\xi}\, L(w,b,\xi,\alpha) = min_{w,b,\xi}\, 0.5\left|\left|w\right|\right|^2 + 0.5C\left|\left|\xi\right|\right|^2 - \alpha\xi + \sum_{i=1}^{n} \alpha_i(1-y_i(w^T x_i + b))$$

Let's compute the partial derivatives and equate them to zero:

$$\nabla_w L(w,b,\xi,\alpha) = 0 \rightarrow w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\nabla_b L(w,b,\xi,\alpha) = 0 \rightarrow -\sum_{i=1}^{n} \alpha_i y_i = 0 \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\nabla_\xi L(w,b,\xi,\alpha) = 0 \rightarrow C\xi - \alpha = 0 \rightarrow \xi = \frac{\alpha}{C}$$

Now, we compensate $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ and $\xi = \frac{\alpha}{C}$ in the Lagrangian and we get:

$$0.5\left|\left|\sum_{i=1}^{n} \alpha_i y_i x_i\right|\right|^2 + 0.5C\left|\left|\frac{\alpha}{C}\right|\right|^2 - \alpha\frac{\alpha}{C} + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i x_i \left(\sum_{i=1}^{n} \alpha_i y_i x_i\right)^T - \sum_{i=1}^{n} \alpha_i y_i b$$

$$= 0.5(\sum_{i=1}^{n} \alpha_i y_i x_i)^2 + \alpha^2(\frac{1}{2c} - \frac{1}{C}) + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i x_i \left(\sum_{i=1}^{n} \alpha_i y_i x_i\right)^T - b\sum_{i=1}^{n} \alpha_i y_i$$

Notice that $\sum_{i=1}^{n} \alpha_i y_i = 0$ then $b\sum_{i=1}^{n} \alpha_i y_i = 0$ and thus:

$$= 0.5(\sum_{i=1}^{n} \alpha_i y_i x_i)^2 - \sum_{i=1}^{n} \alpha_i y_i x_i \left(\sum_{i=1}^{n} \alpha_i y_i x_i\right)^T - \frac{1}{2C}\alpha^2 + \sum_{i=1}^{n} \alpha_i$$

$$= -0.5(\sum_{i=1}^{n} \alpha_i y_i x_i)^2 - \frac{1}{2C}\alpha^2 + \sum_{i=1}^{n} \alpha_i$$

So, we conclude that the above is the minimization of the Lagrangian.

Note: this is in case that the Lagrangian is differentiable and has an optimum, because then it holds that $\sum_{i=1}^{n} \alpha_i y_i = 0$, which may not hold as alpha is not a variable of the optimization problem.

(d) What is the dual problem?

Answer:

We can define the dual function as $g(\alpha) = min_{w,b,\xi}\, L(w, b, \xi, \alpha)$ and then the dual problem is:

$$max_\alpha g(\alpha) = max_\alpha min_{w,b,\xi}\, L(w, b, \xi, \alpha)$$

From the previous question we get that the dual problem is:

$$max_\alpha min_{w,b,\xi} -\frac{1}{2}(\sum_{i=1}^{n} \alpha_i y_i x_i)^2 - \frac{1}{2C}\alpha^2 + \sum_{i=1}^{n} \alpha_i$$

$$s.t\ \alpha_i \geq 0$$

$$and \sum_{i=1}^{n} \alpha_i y_i = 0$$

Programming Assignment:

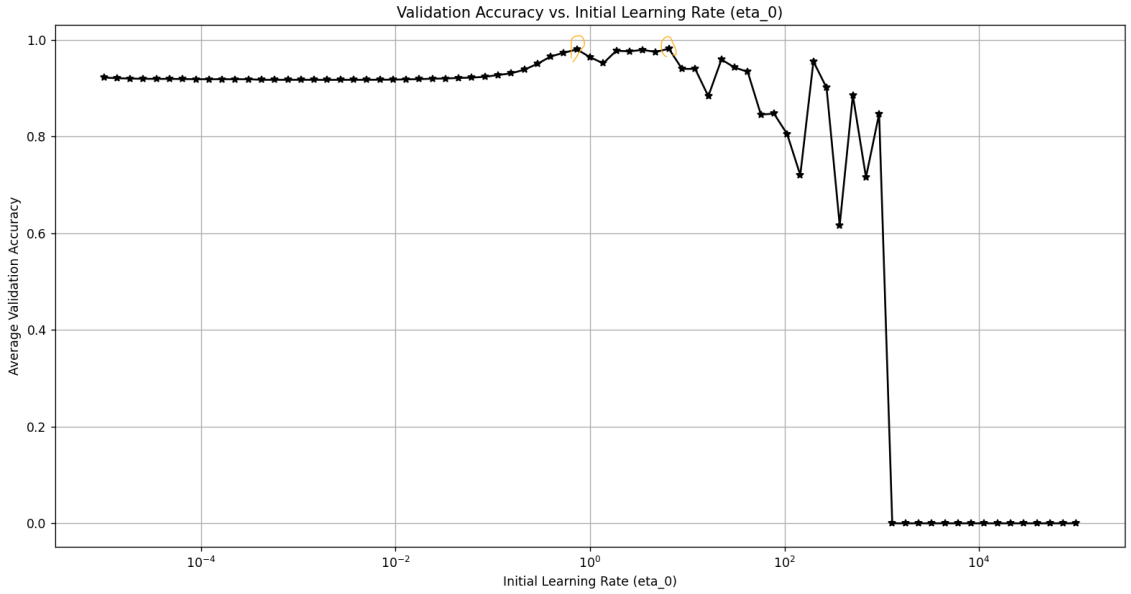
1. **(20 points) SGD for Hinge loss.** We will continue working with the MNIST data set. The file template (`skeleton_sgd.py`), contains the code to load the training, validation and test sets for the digits 0 and 8 from the MNIST data. In this exercise we will optimize the Hinge loss with $L2$-regularization ($\ell(\mathbf{w}, \mathbf{x}, y) = C \cdot \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}\rangle\} + 0.5\|\mathbf{w}\|^2$), using the stochastic gradient descent implementation discussed in class. Namely, we initialize $\mathbf{w}_1 = 0$, and at each iteration $t = 1, \ldots$ we sample i uniformly; and if $y_i \mathbf{w}_t \cdot x_i < 1$, we update:

$$\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t C y_i \mathbf{x}_i$$

and $\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t$ otherwise, where $\eta_t = \eta_0/t$, and η_0 is a constant. Implement an SGD function that accepts the samples and their labels, C, η_0 and T, and runs T gradient updates as specified above. In the questions that follow, make sure your graphs are meaningful. Consider using `set_xlim` or `set_ylim` to concentrate only on a relevant range of values.

   (a) **(10 points)** Train the classifier on the training set. Use cross-validation on the validation set to find the best η_0, assuming $T = 1000$ and $C = 1$. For each possible η_0 (for example, you can search on the log scale $\eta_0 = 10^{-5}, 10^{-4}, \ldots, 10^4, 10^5$ and increase resolution if needed), assess the performance of η_0 by averaging the accuracy on the validation set across 10 runs. Plot the average accuracy on the validation set, as a function of η_0.

Answer:



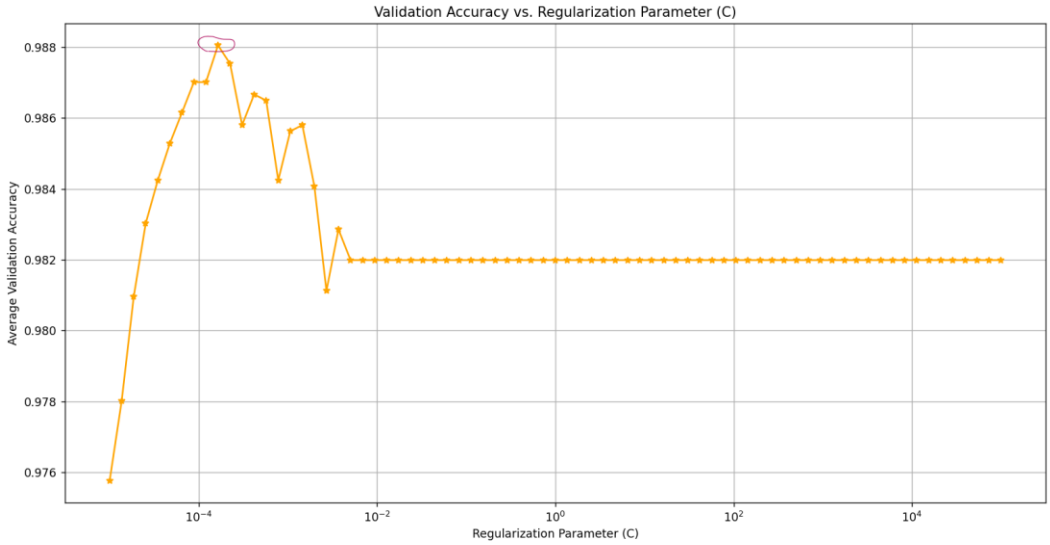
As we can see from the plot above there are 2 best initial learning rate:

$$\eta_0 = 0.723 \; with \; accuracy = 0.983$$

$$\eta_0 = 6.394 \; with \; accuracy = 0.983$$

(b) **(5 points)** Now, cross-validate on the validation set to find the best C given the best η_0 you found above. For each possible C (again, you can search on the log scale as in section (a)), average the accuracy on the validation set across 10 runs. Plot the average accuracy on the validation set, as a function of C.

Answer:

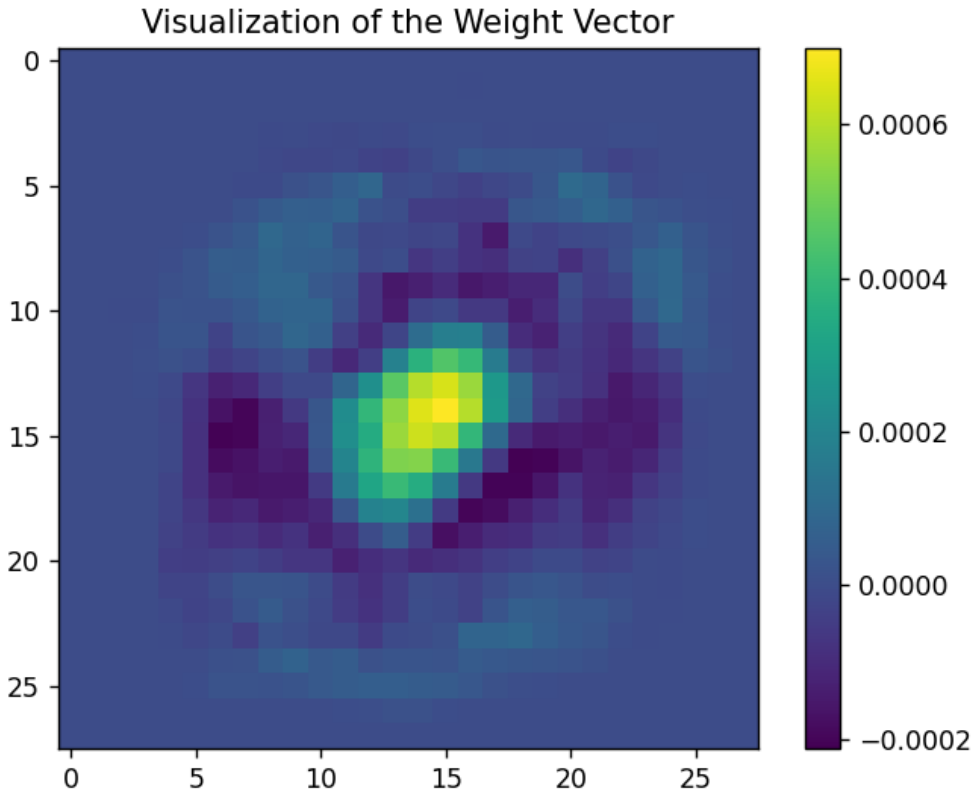


The best accuracy we get is when C=0.000161, so that is the best C(as the best η_0 has been set to be 0.723 according to the previous question).

(c) **(5 points)** Using the best C, η_0 you found, train the classifier, but for $T = 20000$. Show the resulting $\mathbf{w}$ as an image, e.g. using the following `matplotlib.pyplot` function: `imshow(reshape(image, (28, 28)), interpolation='nearest')`. Give an intuitive interpretation of the image you obtain.

Answer:

This is the resulting w as an image:



Bright Yellow Areas: These suggest that the model has learned that the presence of pixel intensity (darkness) in these specific regions is a strong indicator of the image being of the digit 8. In the MNIST dataset, the digit 8 typically has a darker center where the two loops meet.
Dark Blue Areas: These imply that the model assigns a strong negative weight to these pixels. It suggests that high pixel intensity in these areas is less common or less indicative of the digit 8 and possibly more indicative of the digit 0. Since 0 is typically darker around the edges and lighter in the center, these areas could correspond to the central part of 0.

Intuitively we can see that the Bright Yellow Areas tend to look more like 8, but the Dark Blue Areas tend to look more like 0.

(d) **(5 points)** What is the accuracy of the best classifier on the test set?

Answer:

The accuracy of the best classifier on the test set is: 0.9923234390992836

2. **(15 points) SGD for log-loss.** In this exercise we will optimize the log loss defined as follows:

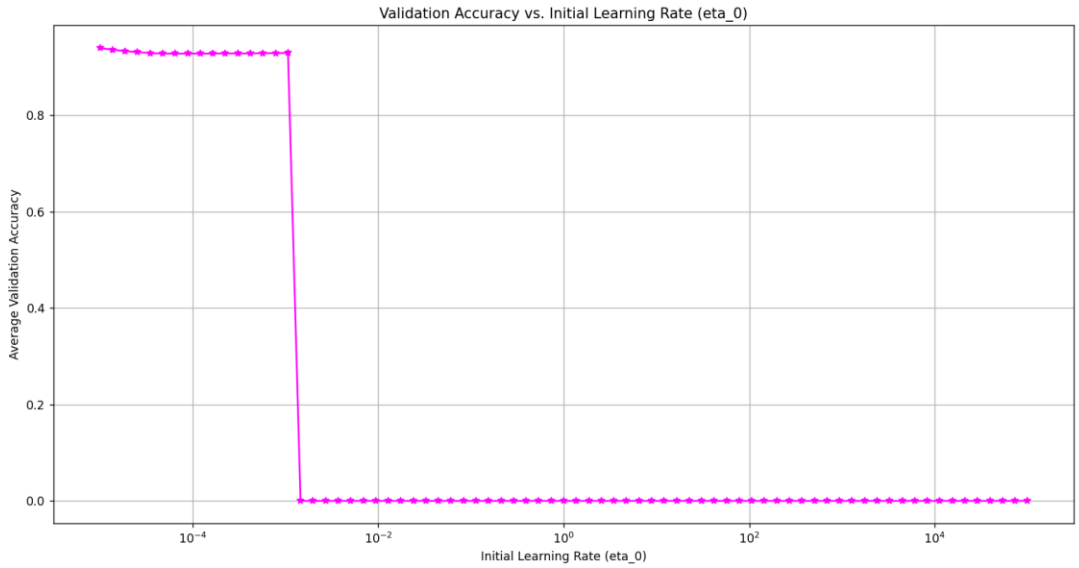
$$\ell_{log}(\mathbf{w}, \mathbf{x}, y) = \log\left(1 + e^{-y\mathbf{w}\cdot\mathbf{x}}\right)$$

(in the lecture you defined the loss with $\log_2(\cdot)$, but for optimization purposes the logarithm base doesn't matter). Derive the gradient update for this case, and implement the appropriate SGD function.

- In your computations, it is recommended to use various built in functions (`scipy.special.softmax` might be helpful) in order to avoid numerical issues which arise from exponentiating very large numbers.

(a) **(5 points)** Train the classifier on the training set using SGD. Use cross-validation on the validation set to find the best η_0, assuming $T = 1000$. For each possible η_0 (for example, you can search on the log scale $\eta_0 = 10^{-5}, 10^{-4}, \ldots, 10^4, 10^5$ and increase resolution if needed), assess the performance of η_0 by averaging the accuracy on the validation set across 10 runs. Plot the average accuracy on the validation set, as a function of η_0.

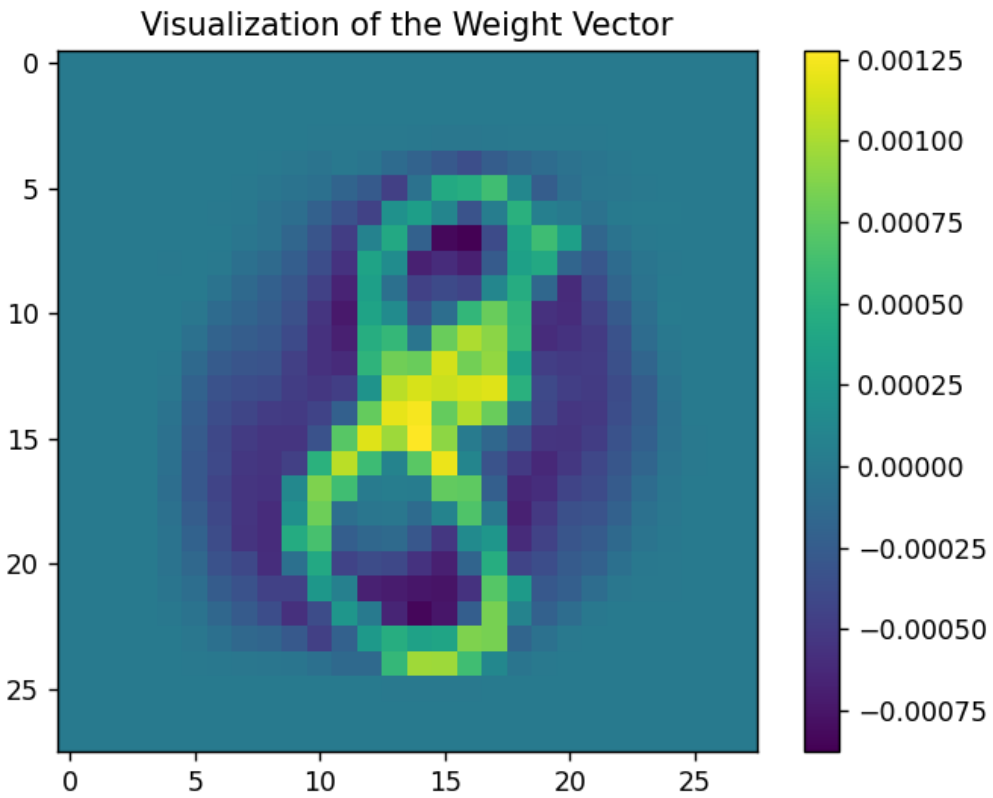
Answer:



We can see that the best eta0 is the first one which is 1e-05=10^{-5} = 0.00001.

(b) **(5 points)** Using the best η_0 you found, train the classifier, but for $T = 20000$. Show the resulting $\mathbf{w}$ as an image. What is the accuracy of the best classifier on the test set?

Answer:



Also, here we can see that the bright yellow areas look like 8, and the dark blue areas look like 0(and I think here it is more obvious).

The accuracy of the best classifier on the test set is: 0.9646878198567042

(c) **(5 points)** Train the classifier for $T = 20000$ iterations, and plot the norm of $\mathbf{w}$ as a function of the iteration. How does the norm change as SGD progresses? Explain the phenomenon you observe.

Answer:

we can see below the norm of w as a function of the iteration, we can say that in general as the iteration increases the norm increases too. Also we can see that the biggest and the fastest increase

happens at first but then it is slow and small, that's due to that we do big steps at first but at last we do small steps.