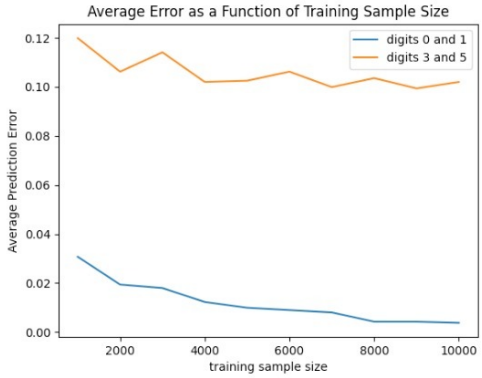
Exercise 3

Introduction to learning and analysis of big data

Nadav Shaked Kim Teldan

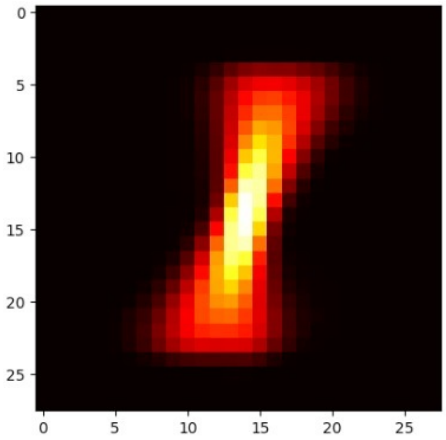
Question 2

a.

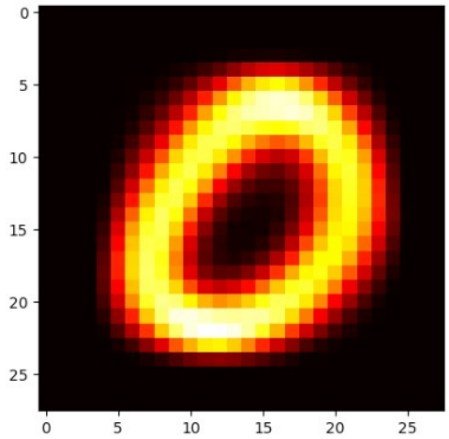


b. When differentiating between 0 and 1, the performance of the method is much better than 3 and 5. For both classification problems, the error decreases as the sample size increases, as we expected. Furthermore, the error decreasing trend is more moderate as the sample size increases.

c. ppos heatmap:



pneg heatmap:



The heatmaps looks like the digits 0 and 1. About ppos heatmap, each heatmap's coordinate indicate how often it is colored when the image's label is 1 - it indicates which coordinate are more reliable indicators of a 1 labeled image. About pneg heatmap, each heatmap's coordinate indicate how often it is colored when the image's label is 0 - it indicates which coordinate are more reliable indicators of a 0 labeled image.
As we learned in class, the prediction function is

$$\log\left(\frac{allpos}{1-allpos}\right)+\sum_{\{i\mid x_i=1\}}\log\left(\frac{ppos(i)}{pneg(i)}\right)-\sum_{\{i\mid x_i=1\}}\log\left(\frac{1-pneg(i)}{1-ppos(i)}\right)$$

This formula implies if the coordinate will indicate:
1 if $\frac{ppos(x_i)}{pneg(x_i)}$ is high and $\frac{1-pneg(x_i)}{1-ppos(x_i)}$ is low, as we can see in the center
0 if $\frac{ppos(x_i)}{pneg(x_i)}$ is low and $\frac{1-pneg(x_i)}{1-ppos(x_i)}$ is high
none of them if $\frac{ppos(x_i)}{pneg(x_i)}$ is low and $\frac{1-pneg(x_i)}{1-ppos(x_i)}$ is low, as we can see in the heatmaps' edges

d. First, we can notice that the samples were chosen randomly and uniformly such that allpos is around 0.5, increasing allpos to 0.75, increases the possibility that the method prediction will be 1, since the change only increased the $\log\left(\frac{allpos}{1-allpos}\right)$ in the classification function

$$sign\left(\log\left(\frac{allpos}{1-allpos}\right)+\sum_{\{i\mid x_i=1\}}\log\left(\frac{ppos(i)}{pneg(i)}\right)-\sum_{\{i\mid x_i=1\}}\log\left(\frac{1-pneg(i)}{1-ppos(i)}\right)\right)$$

Hence, we are expecting that only images that classified as -1 will be classified as 1.

0 and 1 results:
Changes from 1 to -1: 0%
Changes from -1 to 1: 0%

The update of allpos value to 0.75 have minor impact on the results. In most of the runs there is no change in the prediction at all, and there are some runs that there was changes only from -1 to 1 and for 0.1% of the samples only, as we expected. It can be explained by observation of the heatmaps, the hot areas of each digit are completely different, this fact is probably enough to differ between 1 and 0, even with the change of $\log\left(\frac{allpos}{1-allpos}\right)$.

3 and 5 results:
Changes from 1 to -1: 0%
Changes from -1 to 1: 1.18%

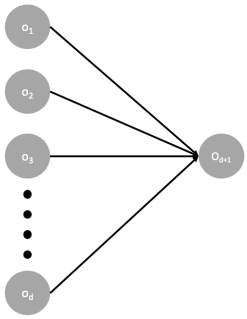
The update of allpos value to 0.75 have much more impact on the results. In every run there is change in the prediction, although the changes are only from -1 to 1. It can be explained by the digits 3 and 5 are sharing hot areas, because the similarity of those digits. So probably there are some samples that were close to zero but negative and the increasing of $\log\left(\frac{allpos}{1-allpos}\right)$ cause the calculation to be positive and mislead the method.

Question 3

a. Define neural network with d input neurons and 1 output neuron, such that:

$$\sigma(x) = sign(x),\ \ G = (V,E)\ \ s.t.\ \ V = \{O_i \mid 1 \le i \le d\} \cup \{O_{d+1}\}, E = \left\{e_{i,d+1} = (o_i, o_{d+1}) \mid 1 \le i \le d\right\}$$

The NN illustration:



The hypothesis class of our neural network is equivalent to the hypothesis class of homogeneous linear predictors. We will proof by definition:

$w_{nn} \in \mathbb{R}^{|E|} = \mathbb{R}^d, w_{hlp} \in \mathbb{R}^d$

$\forall h_w^{nn} \in H_{nn}, h_w^{hlp} \in H_{hlp}$

$$h_w^{nn}(x) = f_w^{G,\sigma}(x) = \sigma(o_{d+1}) = sign\left(\sum_{i=1}^{d} w_{e_{i,d+1}} \bullet o_i\right) = sign\left(\sum_{i=1}^{d} w_{e_{i,d+1}} \bullet o_i\right) = sign\left(\sum_{i=1}^{d} w_i \bullet x_i\right)$$

$$= sign(\langle w, x \rangle) = h_w^{hlp}(x)$$

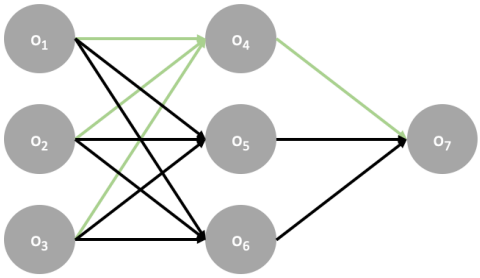
b. Example for NN such that the hypothesis class isn't equivalent to the hypothesis class of homogeneous linear predictor:

Example in $\mathbb{R}^3$:

First, we will show that $H_{hlp} \subseteq H_{nn}$, and then we will show that $\exists h_w^{nn} \in H_{nn}, h_w^{nn} \notin H_{hlp}$,

so $H_{nn} \neq H_{hlp}$.

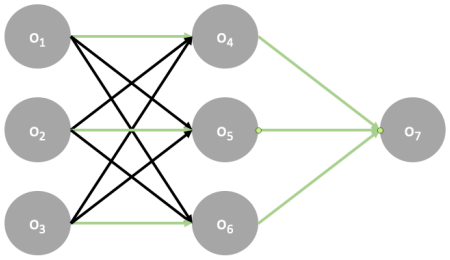
The NN's hypothesis class includes all the homogeneous linear predictors hypothesis class



we set the green edges to be
$w_{(o_1,o_4)} = w_1, w_{(o_2,o_4)} = w_2, w_{(o_3,o_4)} = w_3, w_{(o_4,o_7)} = 1, and\ for\ other\ edges\ w_e = 0$
We will get all the homogeneous linear predictors hypothesis class as described in (a).
$\forall h_w^{hlp} \in H_{hlp}$

$$h_w^{hlp}(x) = sign(\langle w, x\rangle) = sign\left(\sum_{i=1}^{3} w_i \bullet x_i\right) = sign\left(\sum_{i=1}^{3} w_{e_{i,4}} \bullet o_i\right) = sign\left(\left(\sum_{i=1}^{3} w_{e_{i,4}} \bullet o_i\right) \bullet w_{e_{4,7}}\right)$$
$$= \sigma\left(\left(\sum_{i=1}^{3} w_{e_{i,4}} \bullet o_i\right) \bullet w_{e_{4,7}}\right) = f_w^{G,\sigma}(x) = h_w^{nn}(x)$$

There is $h_w^{nn} \in H_{nn}$ such that there is no equivalent homogeneous linear predictor.
The NN over example in $\mathbb{R}^3$ includes function which is positive iff there are more positive coordinates than negative coordinates. The NN will looks like



we set the green edges to be
$w_{(o_1,o_4)} = w_1, w_{(o_2,o_5)} = w_2, w_{(o_3,o_6)} = w_3, w_{(o_4,o_7)} = 1, w_{(o_5,o_7)} = 1, w_{(o_6,o_7)} = 1, and\ for\ other\ edges\ w_e = 0$
Thus, $f_w^{G,\sigma}(x) = sign\big(sign(x_1) + sign(x_2) + sign(x_3)\big)$
Assume in contradict that $\exists h_w^{hlp}\ s.t\ \ h_w^{hlp} = h_w^{nn}$
define $w = (w_1, w_2, w_3)$
So, for $x = (-1, 1, 1)$
$1 = h_w^{hlp}(x) = h_w^{nn}(x) \Rightarrow -w_1 + w_2 + w_3 > 0 \Rightarrow w_2 + w_3 > w_1$(i)

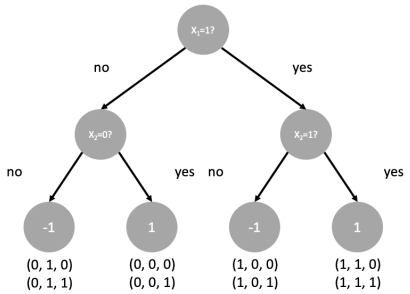
For $x = (1, -3, 1)$
$1 = h_w^{hlp}(x) = h_w^{nn}(x) \Rightarrow w_1 - 3w_2 + w_3 > 0 \Rightarrow w_1 + w_3 > 3w_2$(ii)
Therefore, by (i) and (ii)
$w_2 + 2w_3 > 3w_2 \Rightarrow w_3 > w_2$(iii)

For $x = (1, 1, -3)$
$1 = h_w^{hlp}(x) = h_w^{nn}(x) \Rightarrow w_1 + w_2 - 3w_3 > 0 \Rightarrow w_1 + w_2 > 3w_3$(iv)
Therefore, by (i) and (iv)
$2w_2 + w_3 > 3w_3 \Rightarrow w_2 > w_3$(v)

There is contradiction between (iii) and (v), so $H_{nn} \neq H_{hlp}$.

Question 4

a. First, we will show a decision tree with depth 2 that sufficient the classification problem



As we can see, every example will classify as 1 iff $x_1 = x_2$.

Secondly, we will show that there is not exist decision tree with depth 1 that sufficient the classification problem.
Since $X = \{0,1\}^3$, there are only 6 options for attributes, and every decision tree holds only one attribute, so we will show that each tree will not suffice the classification problem:

- For attributes $x_1 = 0$, $x_1 = 1$, the examples (1, 0, 0) and (1, 1, 0) will classify with same label, while have different labels
- For attributes $x_2 = 0$, $x_2 = 1$, the examples (0, 1, 1) and (1, 1, 1) will classify with same label, while have different labels
- For attributes $x_3 = 0$, $x_3 = 1$, the examples (0, 1, 1) and (1, 1, 1) will classify with same label, while have different labels

Hence there is not decision tree with depth 1 that suffice the classification problem, so the decision tree that suffice the classification problem with the smallest depth is 2.

b. We will show that greedy algorithm that using Gini's gain function scenario that get error of 50% after getting to a depth of 2.
First, we will set the sample set $S = X = \{0,1\}^3$
Notice that $\forall i,\ \ p_i^1 = \frac{4}{8} = \frac{1}{2},\ \ p_i^1 = \frac{4}{8} = \frac{1}{2}$
and $Gain(S,i) = err_{before}(S) - err_{after}(S,i)$

Iteration 1

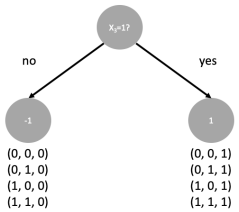
The algorithm chooses the root, so $err_{before}(S) = 0$, the gain is $\frac{1}{2}$ for any feature i, since

$q_i^0 = \frac{4}{8} = \frac{1}{2}, \quad q_i^1 = \frac{4}{8} = \frac{1}{2}$

$err_{after}(S,i) = p_i^0 \bullet Gini(q_i^0) + (1 - p_i^0) \bullet Gini(q_i^1) = \frac{1}{2} \bullet \frac{1}{2} + \frac{1}{2} \bullet \frac{1}{2} = \frac{1}{2}$

so $Gain = err_{before}(S) - err_{after}(S,i) = 0 - \frac{1}{2} = -\frac{1}{2}$

Assume the algorithm chose the feature $x_3 = 1?$ as root.



The decision tree error is $\frac{1}{2}$

Iteration 2 - left child

Notice that $err_{before} = \frac{1}{2}$, and feature $x_3 = 1?$ is not in the feature's store

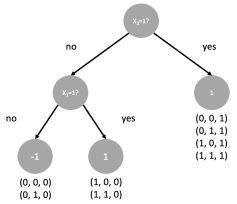
The algorithm chooses the second junction, the gain is 0 for any feature i, since

$q_i^0 = \frac{2}{4} = \frac{1}{2}, \quad q_i^1 = \frac{2}{4} = \frac{1}{2}$

$err_{after}(S,i) = p_i^0 \bullet Gini(q_i^0) + (1 - p_i^0) \bullet Gini(q_i^1) = \frac{1}{2} \bullet \frac{1}{2} + \frac{1}{2} \bullet \frac{1}{2} = \frac{1}{2}$

so $Gain = err_{before}(S) - err_{after}(S,i) = \frac{1}{2} - \frac{1}{2} = 0$

Assume the algorithm chose the feature $x_1 = 1?$ as left side.



The decision tree error is $\frac{1}{2}$

Iteration 3 - right child

Notice that $err_{before} = \frac{1}{2}$, and feature $x_3 = 1?$ and $x_1 = 1?$ is not in the feature's store

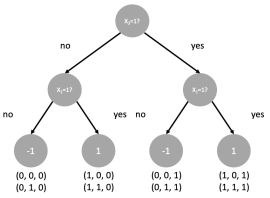
The algorithm chooses the third junction, the gain is 0 for any feature i, since

$q_i^0 = \frac{2}{4} = \frac{1}{2}, \quad q_i^1 = \frac{2}{4} = \frac{1}{2}$

$err_{after}(S,i) = p_i^0 \bullet Gini(q_i^0) + (1 - p_i^0) \bullet Gini(q_i^1) = \frac{1}{2} \bullet \frac{1}{2} + \frac{1}{2} \bullet \frac{1}{2} = \frac{1}{2}$

so $Gain = err_{before}(S) - err_{after}(S,i) = \frac{1}{2} - \frac{1}{2} = 0$

Assume the algorithm chose the feature $x_1 = 1?$ as left side.



The decision tree error is $\frac{1}{2}$ and the depth is 2

Question 5

For training sample $S = \big((x_1, y_1), \dots, (x_m, y_m)\big)$

We want to solve the LASSO regression problem:

$$Minimize_{w \in \mathbb{R}^d} \lambda \|w\|_1 + \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2$$

As we learned in class:

$$\sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2 = \|X^T w - y\|^2 = (X^T w - y)^T (X^T w - y) = (w^T X - y^T)(X^T w - y) = w^T X X^T w - 2y^T X^T w + y^t y$$

Thus, the LASSO regression problem equivalent to:

$$Minimize_{w \in \mathbb{R}^d} w^T X X^T w - 2y^T X^T w + y^t y + \lambda \|w\|_1$$

A minimalization problem that is equivalent to the problem above is:

$$Minimize_{w \in \mathbb{R}^d, \overline{w} \in \mathbb{R}^d} w^T X X^T w - 2y^T X^T w + \lambda \sum_{i=1}^{d} \overline{w}_i$$

$$s.t \quad \forall 0 < i \leq d, \quad \overline{w}_i \geq w_i$$
$$\forall 0 < i \leq d, \quad \overline{w}_i \geq -w_i$$

And return only w from the algorithm solution.

Explanation:

If $w_i \geq 0$, the constraint $\overline{w}_i \geq w_i$ is equivalent to $\overline{w}_i = w_i$, this is because the optimizer will drive the value of y_i as low as possible all the way to equality, note the in this case $\overline{w}_i \geq -w_i$ is also sufficient.

If $w_i < 0$, the constraint $\overline{w}_i \geq -w_i$ is equivalent to $\overline{w}_i = -w_i$, this is because the optimizer will drive the value of y_i as low as possible all the way to equality, note that in this case $\overline{w}_i \geq w_i$ is also sufficient.

Thus for every i, $\overline{w}_i = |w_i|$, so $\sum_{i=1}^{d} \overline{w}_i = \sum_{i=1}^{d} |w_i|$ and represent $\|w\|_1$

To solve the LASSO regression problem by quadratic problem solver

$$Minimize_{z\in\mathbb{R}^n}\frac{1}{2}z^T \bullet H \bullet z + \langle u, z\rangle$$
$$s.t.\ \ Az \geq v$$

We have to set:

u: $\left(-2\sum_{i=1}^{m} X_{1,i}y_i, -2\sum_{i=1}^{m} X_{2,i}y_i, \dots, -2\sum_{i=1}^{m} X_{d,i}y_i, \underbrace{\lambda, \dots, \lambda}_{d\ times}\right) \in \mathbb{R}^{2d}$

v: $\left(\underbrace{0, \dots, 0}_{2d\ times}\right) \in \mathbb{R}^{2d}$

H: $\begin{bmatrix} 2XX^T & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{(2d)x(2d)}$

A: $\begin{bmatrix} -I_d & I_d \\ I_d & I_d \end{bmatrix} \in \mathbb{R}^{(2d)x(2d)}$

z: $(w_1, \dots, w_d, \overline{w}_1, \dots, \overline{w}_d) \in \mathbb{R}^{2d}$

Explanation about matrix A, and the constraints $Az \geq v$:
First d rows of A and v, will represent the constraints
$-w_i + \overline{w}_i \geq 0 \Longrightarrow \overline{w}_i \geq w_i$
Second d rows of A and v, will represent the constraints
$w_i + \overline{w}_i \geq 0 \Longrightarrow \overline{w}_i \geq -w_i$

The equation $\frac{1}{2}z^T \bullet H \bullet z$ will be equivalent to $w^T XX^T w$, and $\langle u, z\rangle$ will be equivalent to $-2y^T X^T w + \lambda\|w\|_1$

Question 6

a. Assume matrix X is a matrix of the examples from the sample set of size $d \times m$, and assume that there is a unique solution w to solve the problem of linear regression with the squared loss on the given sample.
Since there is a unique w that solve the optimization problem of SGD with square loss, we know that XX^T is invertible.
Since $XX^T \in \mathbb{R}^{dxd}$ and XX^T invertible, the rank of XX^T is d.
Multiplication of matrices holds $rank(AB) \leq \min\big(rank(A), rank(B)\big)$, so the rank of X at least d.
The dimension of X is $d \times m$ so the rank is at most d, therefore the rank of X is d.

b. Let $S' = \big((x'_1, y_1), (x'_2, y_2), \dots, (x'_m, y_m)\big)$ as described in the question, denote $w \in \mathbb{R}^d$
We will prove that w' can be any solution from $W' \in \{(w_1, w_2, \dots, w_d, \alpha) \mid \alpha \in \mathbb{R}\}$ by show that
$\forall w' \in W'\ and\ x' \in X, \langle w, x\rangle = \langle w', x'\rangle$
$\langle w, x\rangle = \sum_{i=1}^{d} w_i x_i = \sum_{i=1}^{d} w_i x_i + \alpha \bullet 0 = \sum_{i=1}^{d+1} w'_i x'_i = \langle w', x'\rangle$

Question 7

The code for this question is in the zip, in question7.py

a. The distortion for the date set in the question is 4.151633772242233.

b. $U^T = \begin{bmatrix} 0.18467286 & -0.37083788 & 0.6614683 & 0.62516789 \\ -0.77243642 & 0.00944118 & -0.30530838 & 0.55681203 \end{bmatrix}$
And we checked that U^T is orthonormal

c. The restored X is:
$\begin{bmatrix} 1.31364626 & -2.49821189 & 4.48223153 & 4.15965253 \\ 3.67197174 & 0.93260544 & -0.10929356 & -4.65795229 \\ -9.70324553 & 0.5286199 & -4.48988343 & 6.15105425 \end{bmatrix}$

We calculate the distortion by the formula $\sum_{i=1}^{m} \|x_i - UU^T x_i\|_2^2$
The distortion we received is 4.151633772242253.
The distortion we received is very close to the distortion in 7a.
We calculated the restored vectors using U, so we received the optimal solution.
Furthermore, we learned in class that the optimal objective value is

$$\sum_{i=1}^{m} \|x_i - restored_i\|_2^2 = \sum_{i=k+1}^{d} \lambda_i$$

where λ is a vector of eigenvalues of $A = X^T X$, in descending order.
The sum of the lowest two eigenvalues is 4.151633772242233, which is what we received.

Question 8

a. No, it does not suffice the scale-invariance axiom.
Let $S \subseteq X$, denote $x_1, x_2 = argmin_{x_1,x_2 \in S} \rho(x_1, x_2)$ and $\rho(x_1, x_2) = d$, such that $d < r$,
by definition of F, there is C_i such that $x_1, x_2 \in C_i$.
Denote $\alpha = \frac{r+1}{d}$, hence for $F(S, \alpha\rho)$ we got
$argmin_{x_1,x_2 \in S} \rho(x_1, x_2) = argmin_{x_1,x_2} \alpha\rho(x_1, x_2) \Longrightarrow \alpha\rho(x_1, x_2) = \frac{r+1}{d} \bullet d = r + 1 \not< r$
So, there are no two examples that will be linked, also x_1, x_2 will not share the same cluster C_i,
hence $F(S, \rho) \neq F(S, \alpha\rho)$.

b. Yes, it does suffice the richness axiom.
Let F to be a single-linkage clustering algorithm as described in the question, let $S \subseteq X$, data points and C a valid partition of S.
For $F(S, \rho) = C$ define ρ such that

$$\rho(x, y) = \begin{cases} 0 & x = y \\ \frac{3}{4} r & x \neq y \text{ and exist } i \text{ s.t. } x, y \in C_i \\ 2r & else \end{cases}$$

ρ is a metric that satisfy the three conditions:
$\forall x, y, z \in S$
1. $\rho(x, y) = 0, \ iff \ x = y$
2. $\rho(x, y) = \rho(y, x)$

3. $\rho(x,y) + \rho(y,z) \geq \rho(x,z)$

Proof that $F(S,\rho) = C$:
Let $C' = F(S,\rho)$, we will show that $C = C'$. Let $C_i \in C$ a cluster we will prove that $C_i \in C'$.
Let $x \in C_i$, denote $C'_j \in C'$ the cluster that suffice $x \in C'_j$.
Let $y \in S$, and split to cases
If $y \in C_i$, by definition $\rho(x,y) \leq \frac{3}{4}r$ so by definition of F, $y \in C'_j$.
If $y \notin C_i$, by definition $\rho(x,y) = 2r$ so by definition of F, $y \notin C'_j$.
Hence $C_i = C'_j$, and therefore $C \subseteq C'$. Because they are both partitions of S, we can conclude that $C = C'$.