Theoretical questions

1.

a.

We look on the following training set:

$$S = \begin{Bmatrix} S_1 = ((1,1,1),1), \\ S_2 = ((1,0,0),1), \\ S_3 = ((1,1,0),0), \\ S_4 = ((0,0,1),0) \end{Bmatrix} \subset \{0,1\}^3 \times \{0,1\}$$

We look on the first iteration of ID3 with the information gain:
Therefore $C(\alpha) = -(\alpha \log(\alpha) + (1-\alpha)\log(1-\alpha))$.

We want to calculate which split the algorithm will choose,
To do that we calculate
$Gain(S,i) =$
$C(P[Y=1]) -$
$\left(P[X_i = 1]C(P[\,Y = 1 \mid X_i = 1\,]) + P[X_i = 0]C(P[\,Y = 1 \mid X_i = 0\,])\right)$
for $i = 1,2,3$.

Let us calculate the pre-split error:
$C(P[Y=1]) = -P[Y=1]\log(P[Y=1]) - P[Y=0]\log(P[Y=0])$
$= -\left(\frac{1}{2}\log\left(\frac{1}{2}\right) + \frac{1}{2}\log\left(\frac{1}{2}\right)\right) = 1$

$\boldsymbol{i = 1}$:
$P[X_1 = 1]C(P[\,Y = 1 \mid X_1 = 1\,]) + P[X_1 = 0]C(P[\,Y = 1 \mid X_1 = 0\,])$
$= \frac{3}{4}C\left(\frac{2}{3}\right) + \frac{1}{4}C(0) = -\frac{3}{4}\left(\frac{2}{3}\log\left(\frac{2}{3}\right) + \frac{1}{3}\log\left(\frac{1}{3}\right)\right) \approx 0.688$

$Gain(S,1) \approx 1 - 0.688 = 0.311$

$\boldsymbol{i = 2}$:
$P[X_2 = 1]C(P[\,Y = 1 \mid X_2 = 1\,]) + P[X_2 = 0]C(P[\,Y = 1 \mid X_2 = 0\,])$
$= \frac{1}{2}C\left(\frac{1}{2}\right) + \frac{1}{2}C\left(\frac{1}{2}\right) = C\left(\frac{1}{2}\right) = -\left(\frac{1}{2}\log\left(\frac{1}{2}\right) + \frac{1}{2}\log\left(\frac{1}{2}\right)\right) = 1$

$Gain(S,2) = 1 - 1 = 0$

$\boldsymbol{i = 3}$
$P[X_3 = 1]C(P[\,Y = 1 \mid X_3 = 1\,]) + P[X_3 = 0]C(P[\,Y = 1 \mid X_3 = 0\,])$
$= \frac{1}{2}C\left(\frac{1}{2}\right) + \frac{1}{2}C\left(\frac{1}{2}\right) = C\left(\frac{1}{2}\right) = -\left(\frac{1}{2}\log\left(\frac{1}{2}\right) + \frac{1}{2}\log\left(\frac{1}{2}\right)\right) = 1$

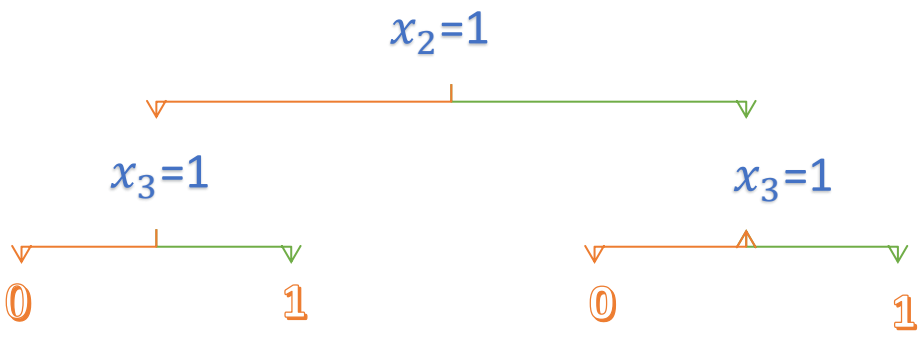
$Gain(S,3) = 1 - 1 = 0$

This means that algorithm will choose to split by $\mathbb{I}_{\{X_1=1\}}$.
This means that during the classification
$S_1 = ((1,1,1),1), S_2 = ((1,0,0),1), S_3 = ((1,1,0),0)$ will go to the same branch.

If the algorithm will choose to split by $\mathbb{I}_{\{X_2=1\}}$then S_1 and S_3 will go to the same branch.
If the algorithm will choose to further classify by $\mathbb{I}_{\{X_3=1\}}$then S_2 and S_3 will go to the same branch.

Because we stop the algorithm in the next iteration this means we will get false classification with at least one sample, this means that the training error of the resulting decision tree is at least 1/4.

b.
Let us look at the following tree:



Where at each node we ask if $x_i = 1$, the left branch marks false result, and the right branch marks positive results.

We can see that this tree is with depth 2, and that on the training set we get that:
On $((1,1,1),1)$ the result is 1.
On $((1,0,0),1)$ the result is 1.
On $((1,1,0),0)$ the result is 0.
On $((0,0,1),1)$ the result is 1.

Thus, this tree achieves zero training error.

2.

a.

Let there be p, q distributions over $\mathcal{X}$.
Let us consider the concave function $\log(x)$.

From Jensen's inequality, we obtain that

$$-D_{KL}(p\|q) = -\sum_x p(x)\log\left(\frac{p(x)}{q(x)}\right) = \sum_x p(x)\log\left(\frac{q(x)}{p(x)}\right) =$$

$$\mathbb{E}_p\left[\log\left(\frac{q}{p}\right)\right] \leq \log\left(\mathbb{E}_p\left[\frac{q}{p}\right]\right) = \log\left(\sum_x p(x)\frac{q(x)}{p(x)}\right) =$$

$$\log\left(\sum_x q(x)\right) = \log(1) = 0$$

Therefore, $D_{KL}(p\|q) \geq 0$.

b.

Let there be distributions p_1, p_2, q_1, q_2 over $\mathcal{X}$ s.t p_1 independent from p_2 and q_1 independent from q_2.

$$D_{KL}(p\|q) =$$

$$\sum_{x_1,x_2} p(x_1,x_2)\log\left(\frac{p(x_1,x_2)}{q(x_1,x_2)}\right) =$$

$$\sum_{x_1,x_2} p_1(x_1)p_2(x_2)\log\left(\frac{p_1(x_1)p_2(x_2)}{q_1(x_1)q_2(x_2)}\right)$$

$$= \sum_{x_1,x_2} p_1(x_1)p_2(x_2)\log\left(\frac{p_1(x_1)}{q_1(x_1)}\right) + \sum_{x_1,x_2} p_1(x_1)p_2(x_2)\log\left(\frac{p_2(x_2)}{q_2(x_2)}\right)$$

$$= \sum_{x_1} p_1(x_1)\log\left(\frac{p_1(x_1)}{q_1(x_1)}\right)\sum_{x_2} p_2(x_2)$$

$$+ \sum_{x_1} p_1(x_1)\sum_{x_2} p_2(x_1)\log\left(\frac{p_2(x_1)}{q_2(x_1)}\right) =$$

$$\sum_x p_1(x)\log\left(\frac{p_1(x)}{q_1(x)}\right) + \sum_x p_2(x)\log\left(\frac{p_2(x)}{q_2(x)}\right) =$$

$$D_{KL}(p_1\|q_1) + D_{KL}(p_2\|q_2)$$

Where we used that $\sum_x p(x) = 1$.

3.

a.

Let us look on the t iteration of AdaBoost.

We assume the $\epsilon_t > 0$.

Let us calculate the error the h_t by the distribution D_{t+1}:

$$\Pr_{x\sim D_{t+1}}[h_t(x) \neq y] = \sum_{i:y_i \neq h_t(x_i)} D_{t+1}(i) = \sum_{i:y_i \neq h_t(x_i)} \frac{D_t(i)e^{-w_t y_i h_t(x_i)}}{Z_t}$$

$$= \frac{1}{Z_t} \sum_{i:y_i \neq h_t(x_i)} D_t(i)e^{w_t} = \frac{\epsilon_t e^{w_t}}{2\sqrt{\epsilon_t(1-\epsilon_t)}} = \frac{\epsilon_t e^{0.5\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)}}{2\sqrt{\epsilon_t(1-\epsilon_t)}}$$

$$= \frac{\epsilon_t \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{2\sqrt{\epsilon_t(1-\epsilon_t)}} = \frac{\sqrt{\epsilon_t(1-\epsilon_t)}}{2\sqrt{\epsilon_t(1-\epsilon_t)}} = \frac{1}{2}$$

Where we used that $z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$, and $y_i h_t(x_i) = -1$ when $y_i \neq h_t(x_i)$.

b. S

We assume by contradiction that $h_t = t_{t+1}$.

Using article a., we notice that $e_{D_{t+1},p}(h_t) = e_{D_{t+1},p}(h_{t+1}) = \frac{1}{2}$.

But this contradicts our assumptions on the weakly learner.

4.

a.

Let $S = \{(x_i, y_i)\}_{i=1}^n$ be the training set, $\mathcal{H}$ a hypothesis class.
We assume that there is $\gamma > 0$, $h_1, \dots, h_k \in \mathcal{H}$ hypotheses and coefficients $a_1, .., a_k \geq 0$ s.t $\sum_i a_i = 1$ for which the following holds:

$$y_i \sum_{j=1}^{k} a_j h_j(x_i) \geq \gamma$$

For all $(x_i, y_i) \in S$.

Let there be distribution D over S.
We take the expectation of both sides:

$$\gamma \leq \mathop{\mathbb{E}}_{i\sim D}\left[y_i \sum_{j=1}^{k} a_j h_j(x_i)\right] = \sum_{j=1}^{k} a_j \mathop{\mathbb{E}}_{i\sim D}\left[y_i h_j(x_i)\right]$$

Because $a_1, .., a_k \geq 0, \sum_i a_i = 1$ this means that there is $\ell \in [k]$ s.t

$$\mathop{\mathbb{E}}_{i\sim D}[y_i h_\ell(x_i)] \geq \gamma$$

We now compute the expectation w.r.t D of $y_i h_\ell(x_i)$:

$$\mathop{\mathbb{E}}_{i\sim D}[y_i h_\ell(x_i)] = \sum_{i=1}^{n} D(i) y_i h_\ell(x_i)$$

$$= \sum_{i:\, y_i = h_\ell(x_i)} D(i) y_i h_\ell(x_i) + \sum_{i:\, y_i \neq h_\ell(x_i)} D(i) y_i h_\ell(x_i)$$

$$= \sum_{i:\, y_i = h_\ell(x_i)} D(i) - \sum_{i:\, y_i \neq h_\ell(x_i)} D(i)$$

$$= \sum_{i:\, y_i = h_\ell(x_i)} D(i) + \sum_{i:\, y_i \neq h_\ell(x_i)} D(i) - \sum_{i:\, y_i \neq h_\ell(x_i)} D(i) - \sum_{i:\, y_i \neq h_\ell(x_i)} D(i)$$

$$= \sum_{i=1}^{n} D(i) - 2 \sum_{i:\, y_i \neq h_\ell(x_i)} D(i) = 1 - 2 \Pr_{i\sim D}[h_\ell(x_i) \neq y_i]$$

This means that $1 - 2 \Pr_{i\sim D}[h_\ell(x_i) \neq y_i] \geq \gamma \Leftrightarrow \Pr_{i\sim D}[h_\ell(x_i) \neq y_i] \leq \frac{1}{2} - \frac{\gamma}{2}$

b. D

Let there be $\{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\} \subset \mathbb{R}^d \times \{-1, 1\}$ training set that is realized by a d dimensional rectangle $[b_1, c_1] \times \dots \times [b_d, c_d]$.

Using the hint, we set $k = 4d - 1, a_i = \frac{1}{4d-1}$.

We will compose $4d - 1$ hypotheses:

For $i \in [d]$ we define:

$$h_{b_i} = \begin{cases} 1 & x_i \geq b_i, \\ -1 & x_i < b_i \end{cases}, \qquad \mathcal{H}_b = \bigcup_{i=1}^{d} \{h_{b_i}\}$$

$$h_{c_i} = \begin{cases} 1 & x_i \leq c_i, \\ -1 & x_i > c_i \end{cases}, \qquad \mathcal{H}_c = \bigcup_{i=1}^{d} \{h_{c_i}\}$$

And we use $2d - 1$ hypotheses that is constant -1, i.e., $h \equiv -1$.

Totally we have $4d - 1$ hypotheses from $\mathcal{H}$.

Let there be $(x^{(i)}, y^{(i)})$ positive sample in S:

We know that for all $j \in [d], h_{b_j}(x^{(i)}) = 1 \vee h_{c_j}(x^{(i)}) = 1$.

Therefore,

$$y^{(i)} \sum_{j=1}^{k} a_j h_j(x^{(i)}) = \sum_{j: h_j \in \mathcal{H}_b \cup \mathcal{H}_c} a_j h_j(x^{(i)}) + \sum_{j: h_j \notin \mathcal{H}_b \cup \mathcal{H}_c} a_j h_j(x^{(i)})$$
$$= \frac{2d}{4d-1} - \frac{2d-1}{4d-1} = \frac{1}{4d-1}$$

Let there be $(x^{(i)}, y^{(i)})$ negative sample in S.

This means that $(x^{(i)}, y^{(i)})$ is not in the d dimensional rectangle.

This means that there is $j \in [d]$ s.t $x_j^{(i)} \notin [b_j, c_j]$, this means that $h_{b_j}(x^{(i)}) + h_{c_j}(x^{(i)}) = 0$.

If $x_j^{(i)} > c_j$, then $h_{b_j}(x^{(i)}) = 1, h_{c_j}(x^{(i)}) = -1$

And if $x_j^{(i)} < b_j$, then $h_{b_j}(x^{(i)}) = -1, h_{c_j}(x^{(i)}) = 1$

This means that

$$\sum_{j:h_j \in \mathcal{H}_b \cup \mathcal{H}_c} a_j h_j(x^{(i)}) \leq \frac{2d-2}{4d-1}$$

Because we left only with $2d-2$ literals $\left(a_j h_{b_j}(x^{(i)}) + a_j h_{c_j}(x^{(i)}) = 0\right)$.

This means that

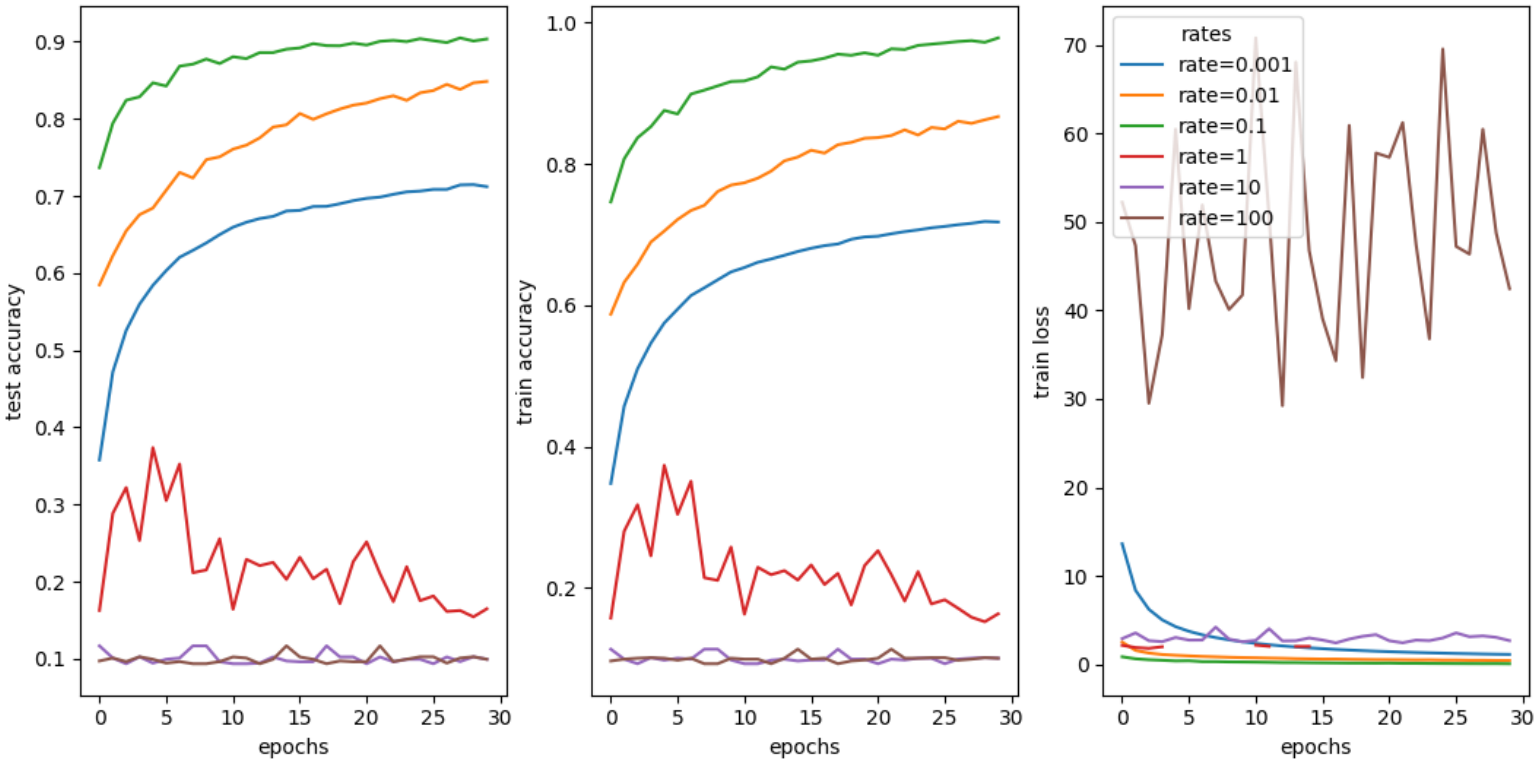
$$y^{(i)} \sum_{j=1}^{k} a_j h_j(x^{(i)}) = -\sum_{j:h_j \notin \mathcal{H}_b \cup \mathcal{H}_c} a_j h_j(x^{(i)}) - \sum_{j:h_j \in \mathcal{H}_b \cup \mathcal{H}_c} a_j h_j(x^{(i)})$$
$$= \frac{2d-1}{4d-1} - \sum_{j:h_j \in \mathcal{H}_b \cup \mathcal{H}_c} a_j h_j(x^{(i)}) \geq \frac{2d-1}{4d-1} - \frac{2d-2}{4d-1} = \frac{1}{4d-1}$$

So, if we denote $\gamma = \frac{1}{4d-1}$ we get what we wanted.

Programming questions:

1.

b.



We can see that in some cases our learning was good, this happened when the learning rate was relatively small (smaller than 0.1).

When the learning rate was large, we can see that we haven't converged to an optimum, perhaps this happened because the leaning step in SGD was too large and caused us to miss a critical point.

When the learning rate was too small, we haven't managed to converge to an optimum, but it looks like with more iterations SGD will finally converge with this learning rate.

c. Using the entire dataset, 30 iterations, learning rate of 0.1, batch size 10, and architecture of [784, 40, 10] the test accuracy in the final epoch is 94.41%.

```
Initial test accuracy: 0.0809
Epoch 0 test accuracy: 0.8361
Epoch 1 test accuracy: 0.891
Epoch 2 test accuracy: 0.8974
Epoch 3 test accuracy: 0.9063
Epoch 4 test accuracy: 0.9075
Epoch 5 test accuracy: 0.9238
Epoch 6 test accuracy: 0.9231
Epoch 17 test accuracy: 0.9413
Epoch 18 test accuracy: 0.9413
Epoch 19 test accuracy: 0.942
Epoch 20 test accuracy: 0.9425
Epoch 21 test accuracy: 0.935
Epoch 22 test accuracy: 0.9415
Epoch 23 test accuracy: 0.9435
Epoch 24 test accuracy: 0.9452
Epoch 25 test accuracy: 0.9417
Epoch 26 test accuracy: 0.9433
Epoch 27 test accuracy: 0.9432
Epoch 28 test accuracy: 0.9427
Epoch 29 test accuracy: 0.9441
```