Deep Learning

Homework 1

Group 71 members:

- Luis Jose De Macedo Guevara (95621)
- Vincent Jakl (108529)

Contributions of each member

- Luis Jose De Macedo Guevara (95621):
- Vincent Jakl (108529):

Question 1

1. a)

The single perceptron was not the best choice for this task. As can be seen in the figure below, the perceptron was not able to fit to the high dimensional data also with no sign of improvement. With the 20 epochs, it ended up with a 0.3422 test accuracy which is slightly higher than 0.25 that would happen with random selection for 4 classes. Also the learning rate was left at the value of 1 so this might cause the problem as well.

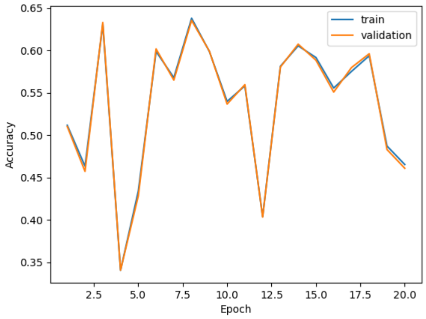


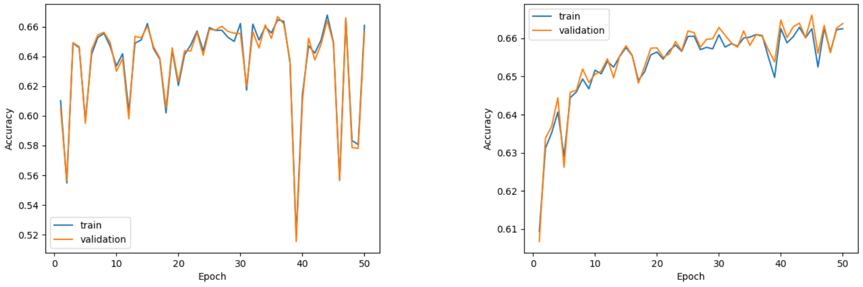
Figure 1: Single perceptron

1. b)

The logistic regression performed better than the single perceptron. As can be seen in the figure below, the logistic regression was able to fit to the data better. There was however a big difference between the two learning rates. Where the model with the learning rate of 0.01 jumped faster to the higher accuracy, it did not improve on a regular basis. The 0.001 lr model was a bit slower to get up to higher accuracy, but it was able to consistently improve. In the end the 0.01 lr model ended up with 0.5784 test accuracy and the 0.001 lr model ended up with 0.5936 test accuracy. Running these models for more epochs would possibly improve the accuracy since the training was still showing signs of improvement even with the evaluation metric.

2. a)

We do believe that the statement is correct. The multiplayer network can be more expressive than single layer logistic regression. The logistic regression is



(a) lr 0.01

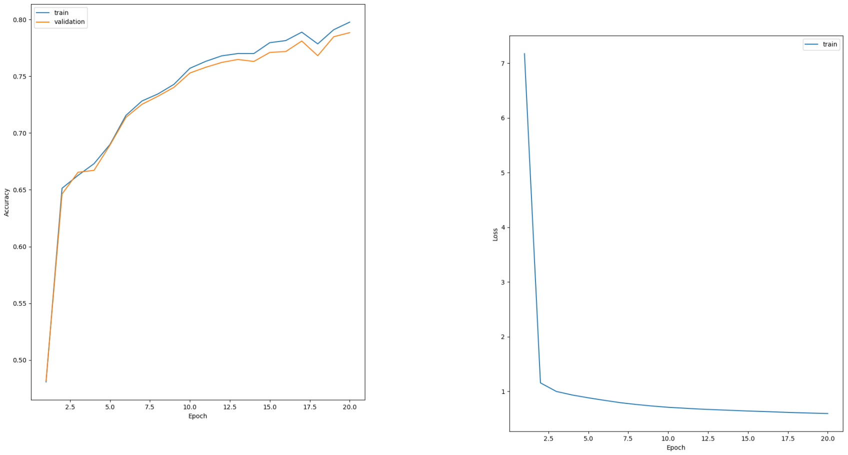
(b) lr 0.001

Figure 2: Logistic regression

a linear model and therefore it can only learn linear decision boundaries. The multi layer network can learn non-linear decision boundaries. However with it being non-linear there can be a problem with local minima for example, which we will not have with the linear model.

2. b)

As seen Figure 3, we can see that the multilayer perceptron, as specified, is a better fit for the data, not only its accuracy on both training and validation data largely improves with each epoch, so does its training loss. It also shows a good test accuracy of 0.7580 after 20 epochs. Looking at the trend in both graphs, we can expect a better accuracy on the training/validation data with more epochs and possibly, taking care to avoid overfitting, a better testing accuracy.



(a) Training/Validation accuracy per epoch

(b) Loss as a function of epoch number

Figure 3: MLP w/o Pytorch

Question 2

1.

2. a)

2. b)

2. c)

Question 3

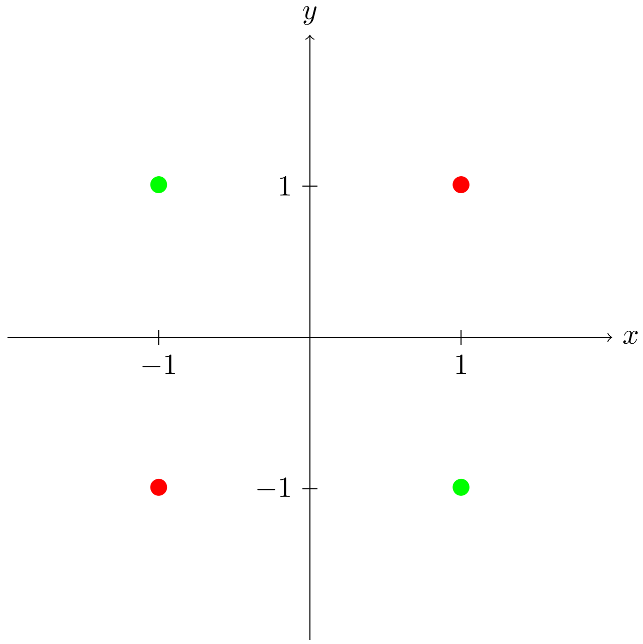
1.

a)

Let

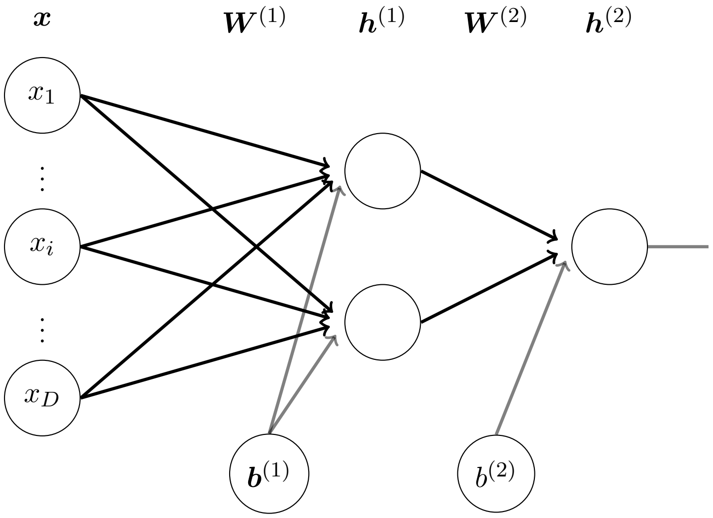
- $D = 2$
- $A = B = 0$
- $\boldsymbol{x}^{(1)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\boldsymbol{x}^{(2)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, $\boldsymbol{x}^{(3)} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ and $\boldsymbol{x}^{(4)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$,

We have $f(\boldsymbol{x}^{(1)}) = -1$, $f(\boldsymbol{x}^{(2)}) = +1$, $f(\boldsymbol{x}^{(3)}) = +1$ and $f(\boldsymbol{x}^{(4)}) = -1$



Which we can see is not linearly separable and therefore a perceptron cannot learn a separating hyperplane.

b)



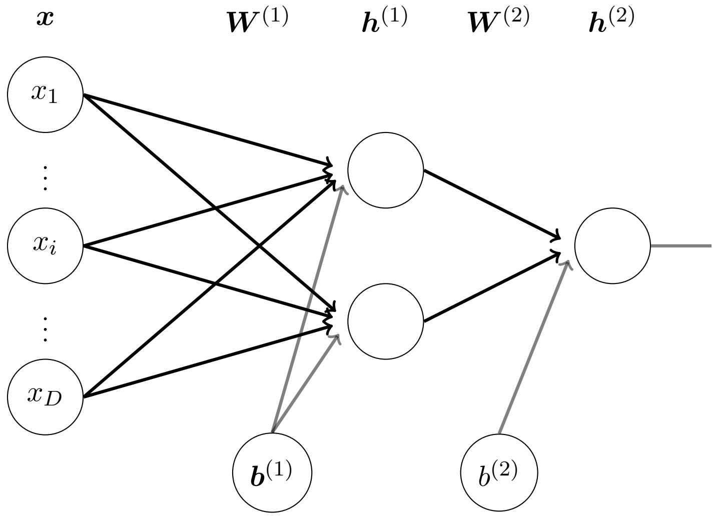
$$\boldsymbol{W}^{(1)} = \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ -1 & -1 & \cdots & -1 \end{bmatrix}}_{2\times D}, \qquad \boldsymbol{b}^{(1)} = \begin{bmatrix} -A \\ B \end{bmatrix}$$

$$\boldsymbol{W}^{(2)} = \begin{bmatrix} 1 & 1 \end{bmatrix}, \qquad b^{(2)} = -1$$

We have for $\boldsymbol{x}$:

$$\begin{aligned}
\boldsymbol{h}^{(2)} &= \operatorname{sign}\left(\boldsymbol{W}^{(2)}\boldsymbol{h}^{(1)} + b^{(2)}\right) \\
&= \operatorname{sign}\left(\boldsymbol{W}^{(2)}\left(\operatorname{sign}\left(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}\right)\right) + b^{(2)}\right) \\
&= \operatorname{sign}\left(\begin{bmatrix} 1 & 1 \end{bmatrix}\left(\operatorname{sign}\left(\begin{bmatrix} 1 & 1 & \cdots & 1 \\ -1 & -1 & \cdots & -1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} + \begin{bmatrix} -A \\ B \end{bmatrix}\right)\right) - 1\right) \\
&= \operatorname{sign}\left(\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} \operatorname{sign}\left(\sum_{i=1}^{D} x_i - A\right) \\ \operatorname{sign}\left(B - \sum_{i=1}^{D} x_i\right) \end{bmatrix} - 1\right) \\
&= \operatorname{sign}\left(\operatorname{sign}\left(\sum_{i=1}^{D} x_i - A\right) + \operatorname{sign}\left(B - \sum_{i=1}^{D} x_i\right) - 1\right)
\end{aligned}$$

c)



$$\boldsymbol{W}^{(1)} = \underbrace{\begin{bmatrix} -1 & -1 & \cdots & -1 \\ 1 & 1 & \cdots & 1 \end{bmatrix}}_{2\times D}, \qquad \boldsymbol{b}^{(1)} = \begin{bmatrix} A \\ -B \end{bmatrix}$$

$$\boldsymbol{W}^{(2)} = \begin{bmatrix} -1 & -1 \end{bmatrix}, \qquad b^{(2)} = 0$$

We have for $\boldsymbol{x}$:

$$\begin{aligned}
\boldsymbol{h}^{(2)} &= \operatorname{sign}\left(\boldsymbol{W}^{(2)}\boldsymbol{h}^{(1)} + b^{(2)}\right) \\
&= \operatorname{sign}\left(\boldsymbol{W}^{(2)}\left(\operatorname{ReLU}\left(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}\right)\right) + b^{(2)}\right) \\
&= \operatorname{sign}\left(\begin{bmatrix} -1 & -1 \end{bmatrix}\left(\operatorname{ReLU}\left(\begin{bmatrix} -1 & -1 & \cdots & -1 \\ 1 & 1 & \cdots & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} + \begin{bmatrix} A \\ -B \end{bmatrix}\right)\right)\right) \\
&= \operatorname{sign}\left(\begin{bmatrix} -1 & -1 \end{bmatrix}\begin{bmatrix} \operatorname{ReLU}\left(A - \sum_{i=1}^{D} x_i\right) \\ \operatorname{ReLU}\left(\sum_{i=1}^{D} x_i - B\right) \end{bmatrix}\right) \\
&= \operatorname{sign}\left(-\operatorname{ReLU}\left(A - \sum_{i=1}^{D} x_i\right) - \operatorname{ReLU}\left(\sum_{i=1}^{D} x_i - B\right)\right)
\end{aligned}$$