Deep Learning - 2022/2023

Homework I – Group 05

95604 João Bagorro , 95617 Juliana Yang

Work Distribution

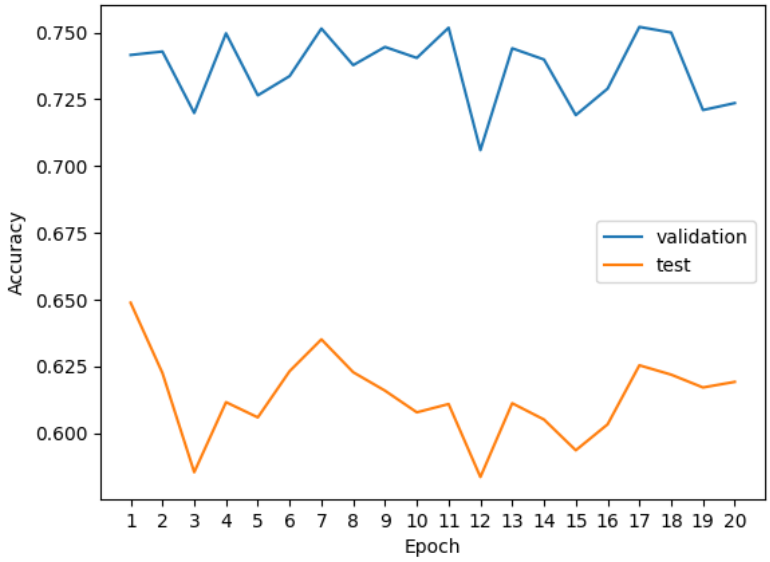
The coding exercises were done by both students which in turn were compared, choosing the best implementation, the testing and bug solving was done as a group. Question number 3 was solved together. Both students contributed equally.

Question 1

1.

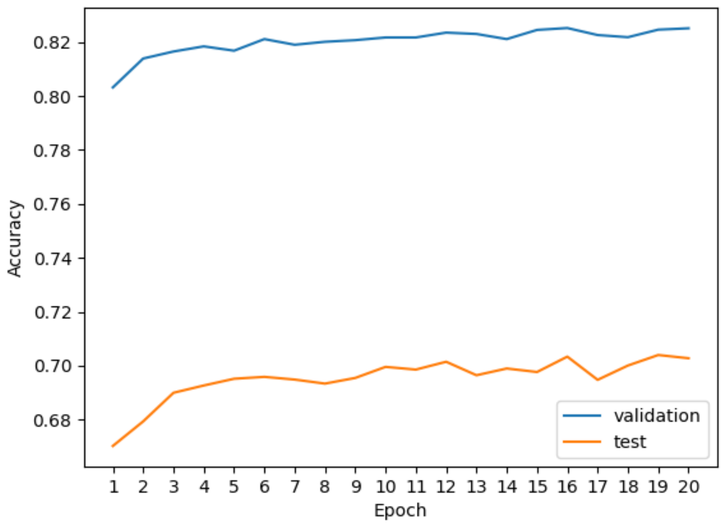
a) Performance:

- **Test set:** 0.6193
- **Validation set:** 0.7236



b) Performance

- **Test set:** 0.7028
- **Validation set:** 0.8251



Deep Learning - 2022/2023

Homework I – Group 05

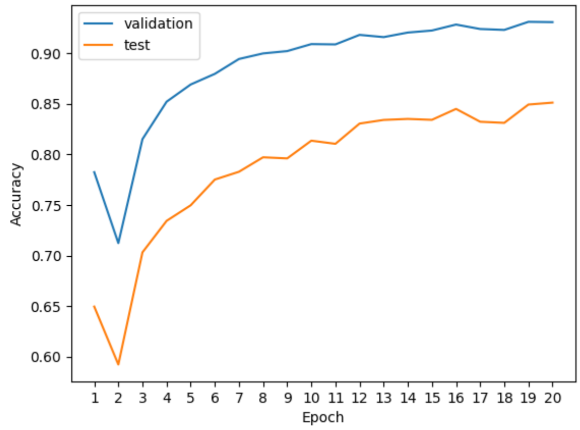
95604 João Bagorro , 95617 Juliana Yang

2.

a) A simple perceptron such as the one implemented in the previous exercise is a model that cannot solve non-linearly separable problems since the VC- dimension of a line is 3 and a single line in 2 dimensions cannot discriminate it.
As opposed to this the multi layered perceptron can solve non-linearly separable problems like XOR.
This is possible because each hidden layer of the perceptron computes a representation of the input and propagates it forward. This in turn will increase the expressive power of the network allowing for more complex, non-linear models.
In case the activation function is linear the multi-layer perceptron will have the same results as the simple perceptron. A multi-layer perceptron with linear activations can be replaced with a simple perceptron by composing those linear activation functions into one linear function.

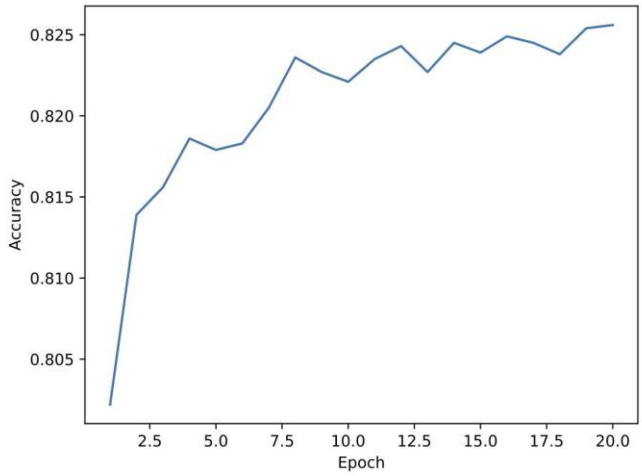
b) Performance:

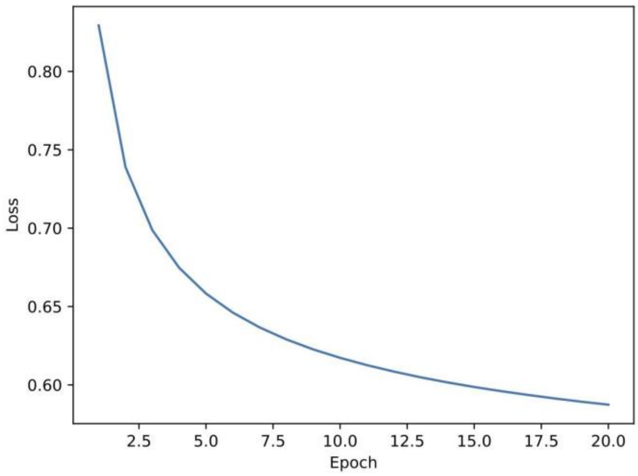
- **Test set:** 0.8512
- **Validation set:** 0.9307



Question 2

1. **Best configuration with learning rate** 0.001
   Final Accuracy of test set: 0.7019





Deep Learning - 2022/2023

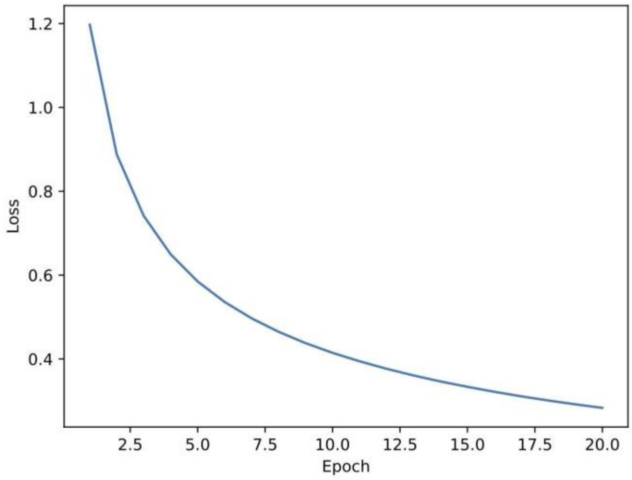
Homework I – Group 05

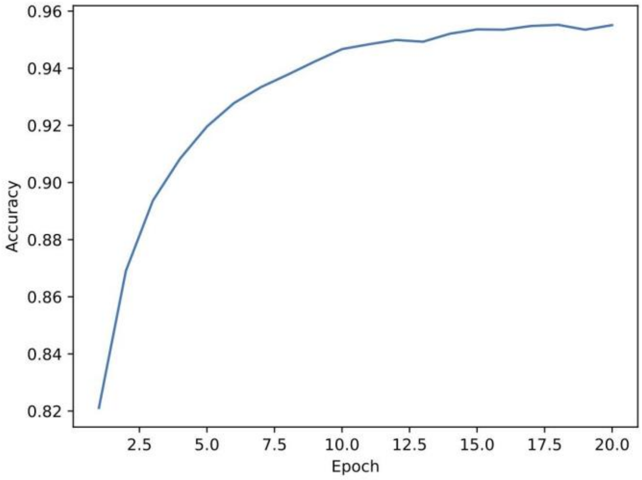
95604 João Bagorro , 95617 Juliana Yang

2.

Best Hyperparameters and Design Choices	
Learning Rate	0.01
Hidden Size	200
Dropout Probability	0.3
Activation Function	Relu

Final Accuracy of test set: 0.8953



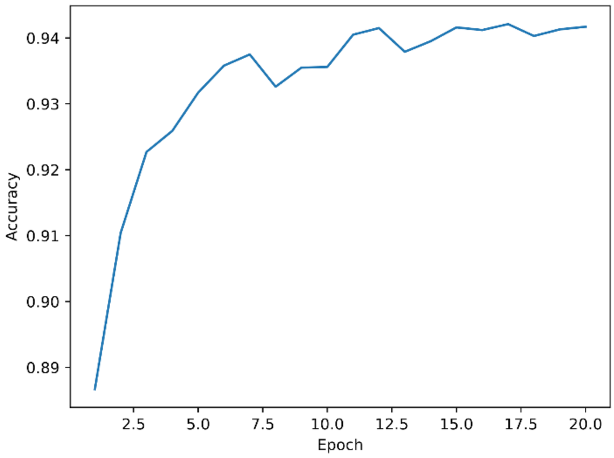


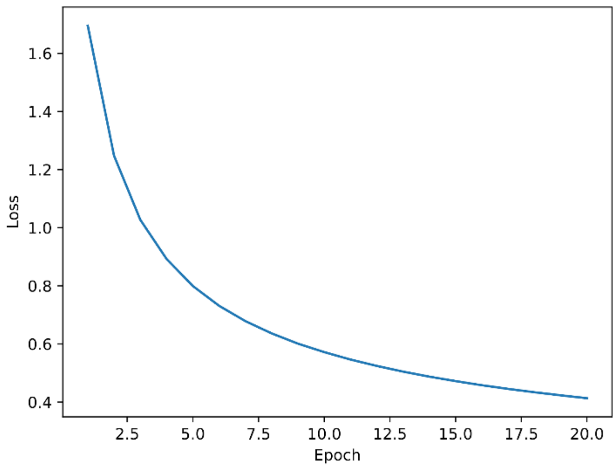
3.

- **With 2 Layers**

Original Parameters (Shown in the provided table):

Final Accuracy of test set: 0.8633





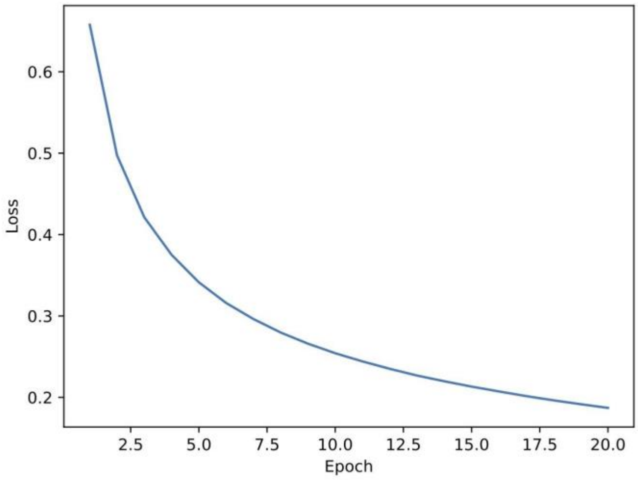
Deep Learning - 2022/2023

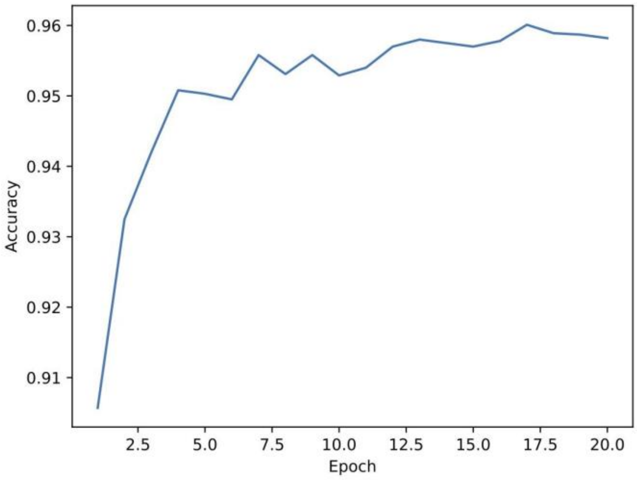
Homework I – Group 05

95604 João Bagorro , 95617 Juliana Yang

Best Hyperparameters and Design Choices	
Learning Rate	0.1
Hidden Size	200
Dropout Probability	0.3
Activation Function	Relu

Final Accuracy of test set: 0.9011

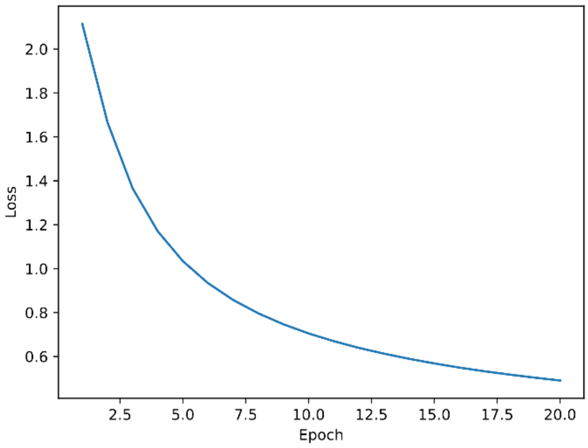


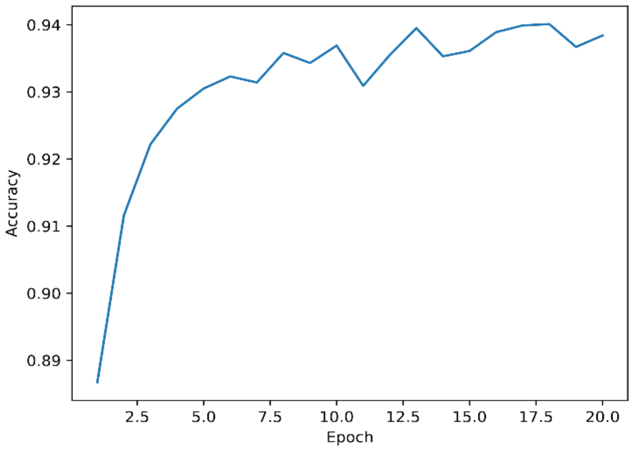


- **With 3 Layers**

Original Parameters (Shown in the provided table):

Final Accuracy of test set: 0.8600





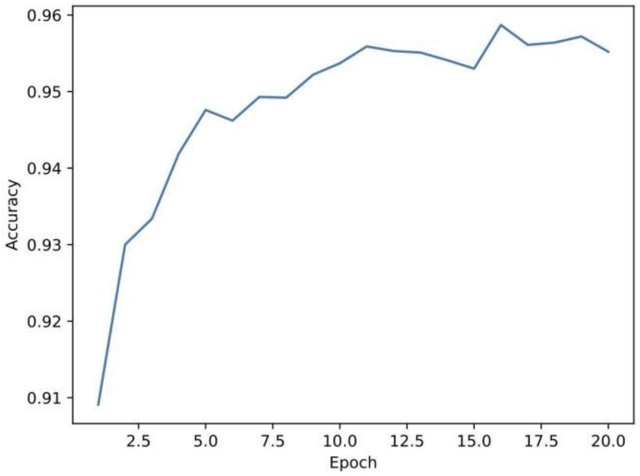
Deep Learning - 2022/2023

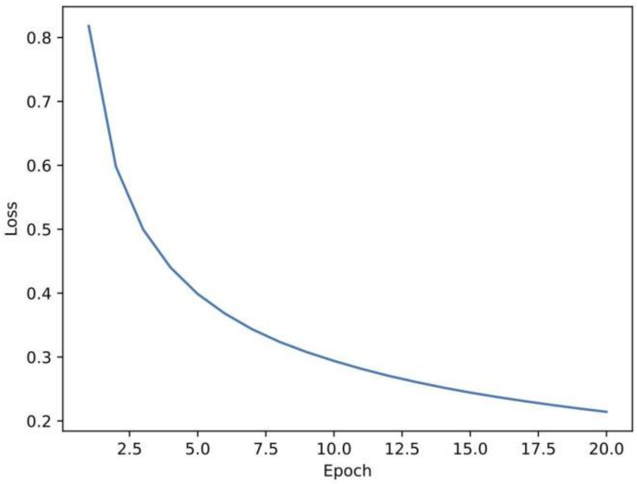
Homework I – Group 05

95604 João Bagorro , 95617 Juliana Yang

Best Hyperparameters and Design Choices	
Learning Rate	0.1
Hidden Size	200
Dropout Probability	0.3
Activation Function	Relu

Final Accuracy of test set: 0.8990





Question 3

1.

$$h = (Wx)^2 = \left(\begin{bmatrix} w_{11} & \cdots & w_{1D} \\ \vdots & \ddots & \vdots \\ w_{k1} & \cdots & w_{kD} \end{bmatrix}\begin{bmatrix} x_1 \\ \vdots \\ x_D \end{bmatrix}\right)^2 = \left(\begin{bmatrix} \sum_{i=1}^{D} w_{1i}x_i \\ \vdots \\ \sum_{i=1}^{D} w_{ki}x_i \end{bmatrix}\right)^2 = \left(\begin{bmatrix} \sum_{i=1}^{D} (w_{1i}x_i)^2 + 2\sum_{j=1}^{D}\sum_{i=1}^{j-1} (w_{1i}x_i)(w_{1j}x_j) \\ \vdots \\ \sum_{i=1}^{D} (w_{ki}x_i)^2 + 2\sum_{j=1}^{D}\sum_{i=1}^{j-1} (w_{ki}x_i)(w_{kj}x_j) \end{bmatrix}\right)$$

Considering $d = \frac{D(D+1)}{2}$, we can say that:

$$h = A_\theta \phi(x) \Leftrightarrow \begin{bmatrix} a_{11}b_1 + \cdots + a_{1d}b_d \\ \vdots \\ a_{k1}b_1 + \cdots + a_{kd}b_d \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kd} \end{bmatrix}\begin{bmatrix} b_1 \\ \vdots \\ b_D \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{d} a_{1i}b_i \\ \vdots \\ \sum_{i=1}^{d} a_{ki}b_i \end{bmatrix}$$

Considering the below equation from the first row of the expressions above, we want to know what is a_{1i} and b_i

$$\sum_{i=1}^{d} a_{1i}b_i = \sum_{i=1}^{D} (w_{1i}x_i)^2 + 2\sum_{j=1}^{D}\sum_{i=1}^{j-1} (w_{1i}x_i)(w_{1j}x_j) = \sum_{i=1}^{D} (w_{1i}x_i)^2 + 2\sum_{j=1}^{D}\sum_{i=1}^{j-1} (w_{1i}w_{1j})(x_i x_j)$$

Therefore, we could consider that

$$\sum_{i=1}^{d} a_{1i} = \sum_{i=1}^{D} (w_{1i})^2 + 2\sum_{j=1}^{D}\sum_{i=1}^{j-1} (w_{1i}w_{1j}) \text{ , } \sum_{i=1}^{d} b_i = \sum_{i=1}^{D} (x_i)^2 + 2\sum_{j=1}^{D}\sum_{i=1}^{j-1} (x_i x_j)$$

So, we can conclude that:

Deep Learning - 2022/2023

Homework I – Group 05

95604 João Bagorro , 95617 Juliana Yang

$$A_\theta = \begin{bmatrix} (w_{11})^2 & \cdots & (w_{1d})^2 & 2w_{11}w_{12} & \cdots & 2w_{kD-1}w_{1D} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ (w_{k1})^2 & \cdots & (w_{kd})^2 & 2w_{k1}w_{k2} & \cdots & 2w_{kD-1}w_{1D} \end{bmatrix} \quad \text{and} \quad \phi(x) = \begin{bmatrix} (x_i)^2 \\ \vdots \\ (x_D)^2 \\ 2x_1x_2 \\ \vdots \\ 2x_{D-1}x_D \end{bmatrix}$$

2. We can say that $\hat{y} = (x; c_\theta) = c_\theta^T \phi(x) = v^T h = v^T A_\theta \phi(x)$ since we previously showed that $h = A_\theta \phi(x)$ for a certain $\phi(x)$.
Therefore, we can conclude that $c_\theta^T = v^T A_\theta$.
Even though c_θ is a linear combination of the rows in A_θ, the entries in this matrix are quadratic, as exemplified above, so c_θ is not a linear function of Θ = (W , v). Because of this, the resulting model is **not linear** in terms of the original parameters Θ, but **quadratic** in terms of **W**.

3. From the previous exercises we have:
$\hat{y} = v^T h = \sum_{i=1}^{K} v_i h_i$ and $h_i = (w_i{}^T x)^2$

Let $\ll\gg$ denote de Frobenius inner product $\ll A, B \gg = vec(A)^T vec(B) = TR(A^T B)$.
We also know that $TR(ABC) = TR(CBA)$.

With this and inner product properties we can keep manipulating the expression:

$\sum_{i=1}^{K} v_i h_i = \sum_{i=1}^{K} v_i \ll WiWi^T, xx^T \gg$ also knowing that v is a scalar we can obtain:
$\ll \sum_{i=1}^{K} v_i WiWi^T, xx^T \gg \; = \; \ll W^T vW, xx^T \gg$ which will be the product between 2 matrixes.

- $W^T vW$ will be a symmetric matrix since v is a diagonal matrix and W^T, W are symmetric.
- $W \in R^{K \times D}, v \in R^K$
- $W^T vW$ will belong to $R^{D \times D}$ for a $rank \le K$

What we pretend to do now is collect the initial parameters Θ = (W , v) expressed in the matrix $W^T vW$ and put them in a vector which will be equal to c_θ.
This vector will be obtained with $c_\theta \in R^{\frac{D(D+1)}{2}}$ since the question states that $K \ge D$ and we will be able fit all the parameters in our new model in terms of c_θ. This together with question 3.2 shows us that it will be a linear model.

Lastly in case $K < D$ we get a matrix $W^T vW$ with $rank > K$ and because of this we might not be able to capture all of the original parameters Θ = (W , v) in terms of c_θ.

4. Since $\hat{y} = c_\theta^T \phi(x)$ is linear we will be able to find a closed form solution to $\hat{C}_\theta$.
Let X be a matrix $\in R^{N \times \frac{D(D+1)}{2}}$ with $\phi(x)$ as rows, and $Y = (y1, y2, \ldots, yN)$. D = training data with $N > \frac{D(D+1)}{2}$.

We then have $L(c_\theta; D) = \frac{1}{2}\sum_{n=1}^{N}(\hat{y}_n(x_n; c_\theta) - y_n)^2$ which we can write as $\frac{1}{2}\|Zc_\theta - y\|^2$ which is the L2 norm of R^N.
We now want to minimize this expression regarding c_θ. This is trivial because it is the known Least Squares problem.
The result will be $\hat{C}_\theta = (X^T X)^{-1} X^T y$.
This is a global minimum which is usually hard to find for activation function such as tanh or Relu but in this case we have a polynomial activation function. The main reason why it is possible is because Least Squares is a convex problem with a closed form solution for the optimal parameters. This means that there will be a global minimum that can be found by using mathematical techniques.