Deep Learning Assignment 1

Omri Zeevy 313327041, Uriel Zaed 319435608

May 2023

Q1

1.a.

$$\left[\frac{\partial \mathbf{x}^T\mathbf{B}\mathbf{x}}{\partial \mathbf{x}}\right]_i = \frac{\partial \mathbf{x}^T\mathbf{B}\mathbf{x}}{\partial \mathbf{x}_i} = \frac{\partial}{\partial \mathbf{x}_i} * \sum_{p,q} \mathbf{x}_p * \mathbf{B}_{p,q} * \mathbf{x}_q =$$

$$= \sum_{p,q} \frac{\partial \mathbf{x}_p}{\partial \mathbf{x}_i} * \mathbf{B}_{p,q} * \mathbf{x}_q + \sum_{p,q} \mathbf{x}_p * \mathbf{B}_{p,q} * \frac{\partial \mathbf{x}_q}{\partial \mathbf{x}_i} =$$

$$= \sum_{p,q} \delta_{p,i} * \mathbf{B}_{p,q} * \mathbf{x}_q + \sum_{p,q} \mathbf{x}_p * \mathbf{B}_{p,q} * \delta_{q,i} =$$

$$= \sum_{q} \mathbf{B}_{i,q} * \mathbf{x}_q + \sum_{p} \mathbf{x}_p * \mathbf{B}_{p,i} =$$

The final closed-form expression is:

$$\frac{\partial \mathbf{x}^T\mathbf{B}\mathbf{x}}{\partial \mathbf{x}} = \mathbf{B}^T * \mathbf{x} + \mathbf{B} * \mathbf{x}$$

1.b.

$$\left[\frac{\partial tr(\mathbf{VXW})}{\partial \mathbf{X}}\right]_{i,j} = \frac{\partial tr(\mathbf{VXW})}{\partial \mathbf{X}_{i,j}} = \frac{\partial}{\partial \mathbf{X}_{i,j}} * \sum_{n}\sum_{p} \mathbf{W}_{p,n} \sum_{k} \mathbf{V}_{n,k} * \mathbf{X}_{k,p} =$$

$$= \sum_{n}\sum_{p} \mathbf{W}_{p,n} \sum_{k} \mathbf{V}_{n,k} * \frac{\partial}{\partial \mathbf{X}_{i,j}} * \mathbf{X}_{k,p} = \sum_{n}\sum_{p} \mathbf{W}_{p,n} \sum_{k} \mathbf{V}_{n,k} * \delta_{k,i} * \delta_{p,j} =$$

$$= \sum_{n}\sum_{p} \mathbf{W}_{p,n}\mathbf{V}_{n,i} * \delta_{p,j} = \sum_{n} \mathbf{W}_{j,n} * \mathbf{V}_{n,i}$$

The final closed-form expression is:

$$\frac{\partial tr(\mathbf{VXW})}{\partial \mathbf{X}} = [\mathbf{WV}]^{\mathbf{T}}$$

1.c.

$$\left[\frac{\partial \mathbf{w}}{\partial \mathbf{w}}\right]_i = \frac{\partial \mathbf{w}}{\partial \mathbf{w}_i} = \frac{\partial}{\partial \mathbf{w}_i} * \sqrt{\mathbf{w^T} * \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}_i} * \sqrt{\sum_k \mathbf{w_k^2}} =$$

$$\frac{1}{2 * \sqrt{\sum_k \mathbf{w_k^2}}} * \sum_k \frac{\partial}{\partial \mathbf{w_k}} * \mathbf{w_k^2} = \frac{1}{2 * \sqrt{\sum_k \mathbf{w_k^2}}} * \sum_k 2 * \mathbf{w_k} * \delta_{k,i} =$$

$$= \frac{2 * \mathbf{w_i}}{2 * \sqrt{\sum_k \mathbf{w_k^2}}} = \frac{\mathbf{w_i}}{\sqrt{\sum_k \mathbf{w_k^2}}}$$

The final closed-form expression is:

$$\frac{\partial ||\mathbf{w}||}{\partial \mathbf{w}} = \frac{\mathbf{w}}{||\mathbf{w}||}$$

1.d.

$$\left[\frac{\partial \mathbf{tr(S)}}{\partial \mathbf{S}}\right]_{i,j} = \frac{\partial}{\partial \mathbf{S}_{i,j}} * \sum_k \mathbf{S}k, k = \sum_k \frac{\partial \mathbf{S}k, k}{\partial \mathbf{S}_{i,j}} =$$

$$= \sum_k \delta_{k,i} * \delta_{k,j} = \delta_{i,i} * \delta_{i,j} = \delta_{i,j} = [I]_{i,j}$$

The final closed-form expression is:

$$\frac{\partial \mathbf{tr(S)}}{\partial \mathbf{S}} = I$$

Q2

2.a.

$\frac{\partial \mathbf{L}}{\partial \mathbf{W}}$:

$$\mathbf{Y} = \mathbf{X}\mathbf{W^T} + \mathbf{B}$$

assuming L is a scalar loss function of Y. Using the chain rule:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{W}} = \frac{\partial \mathbf{L}}{\partial \mathbf{Y}} * \frac{\partial \mathbf{Y}}{\partial \mathbf{W}}$$

$$\frac{\partial \mathbf{Y_{ij}}}{\partial \mathbf{W_{pq}}} = \frac{\partial (\mathbf{XW^T} + \mathbf{b})_{ij}}{\partial \mathbf{W_{pq}}} = \frac{\partial (\mathbf{XW^T})_{ij}}{\partial \mathbf{W_{pq}}} = \frac{\partial (\sum_k \mathbf{X_{ik}} \mathbf{W_{kj}^T})}{\partial \mathbf{W_{pq}}} = \frac{\partial (\sum_k \mathbf{X_{ik}} \mathbf{W_{jk}})}{\partial \mathbf{W_{pq}}} =$$

$$= \sum_k \mathbf{X_{ik}} * \delta_{jp} * \delta_{kq} = \mathbf{X_{iq}} * \delta_{jp}$$

$$\frac{\partial \mathbf{L}}{\partial \mathbf{W_{pq}}} = \sum_{i,j} \frac{\partial \mathbf{L}}{\partial \mathbf{Y_{ij}}} * \mathbf{X_{iq}} * \delta_{jp} = \sum_i (\frac{\partial \mathbf{L}}{\partial \mathbf{Y}})_{pj}^T * \mathbf{X_{iq}}$$

Therefore, the closed-form expression is:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{W}} = (\frac{\partial \mathbf{L}}{\partial \mathbf{Y}})^T * X$$

$\frac{\partial \mathbf{L}}{\partial \mathbf{b}}$:
using the chain rule:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{b}} = \frac{\partial \mathbf{L}}{\partial \mathbf{Y}} * \frac{\partial \mathbf{Y}}{\partial \mathbf{b}}$$

Similar to the previous case:

$$\frac{\partial \mathbf{Y_{ij}}}{\partial \mathbf{b_p}} = \frac{\partial (\mathbf{XW^T} + \mathbf{b})_{ij}}{\partial \mathbf{b_p}} = \frac{\partial \mathbf{b_{ij}}}{\partial \mathbf{b_p}} = \delta_{jp}$$

$$\frac{\partial \mathbf{L}}{\partial \mathbf{b}} = \sum_{i,j} \frac{\partial \mathbf{L}}{\partial \mathbf{Y_{ij}}} * \frac{\partial \mathbf{Y_{ij}}}{\partial \mathbf{b_p}} = \sum_{i,j} \frac{\partial \mathbf{L}}{\partial \mathbf{Y_{ij}}} * \delta_{jp} = \sum_{i} (\frac{\partial \mathbf{L}}{\partial \mathbf{Y_{qi}}})^T * (1) = ((\frac{\partial \mathbf{L}}{\partial \mathbf{Y}})^T * (1))_p$$

Therefore, the closed-form expression is:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{b}} = (1)^T * \frac{\partial \mathbf{L}}{\partial \mathbf{Y}}$$

$\frac{\partial \mathbf{L}}{\partial \mathbf{X}}$:
Again, using the chain rule:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X}} = \frac{\partial \mathbf{L}}{\partial \mathbf{Y}} * \frac{\partial \mathbf{Y}}{\partial \mathbf{X}}$$

$$\frac{\partial \mathbf{Y_{ij}}}{\partial \mathbf{X_{pq}}} = \frac{\partial (\mathbf{XW^T} + \mathbf{b})_{\mathbf{ij}}}{\partial \mathbf{X_{pq}}} = \frac{\partial (\mathbf{XW^T})_{\mathbf{ij}}}{\partial \mathbf{X_{pq}}} = \frac{\partial (\sum_k \mathbf{X_{ik}} \mathbf{W^T_{kj}})}{\partial \mathbf{X_{pq}}} = \frac{\partial (\sum_k \mathbf{X_{ik}} \mathbf{W_{jk}})}{\partial \mathbf{X_{pq}}}$$

$$= \sum_k \delta_{ip} * \delta_{kq} * \mathbf{W_{jk}} = \delta_{ip} * \mathbf{W_{jq}}$$

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X_{pq}}} = \sum_{i,j} \frac{\partial \mathbf{L}}{\partial \mathbf{Y_{ij}}} * \frac{\partial \mathbf{Y_{ij}}}{\partial \mathbf{X_{pq}}} = \sum_{i,j} \frac{\partial \mathbf{L}}{\partial \mathbf{Y_{ij}}} * \delta_{ip} * \mathbf{W_{jq}} = \sum_{j} \frac{\partial \mathbf{L}}{\partial \mathbf{Y_{pj}}} * \mathbf{W_{jq}}$$

Therefore, the closed-form expression is:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X}} = \frac{\partial \mathbf{L}}{\partial \mathbf{Y}} * W$$

2.b.

i) Generic activation function h: Since Y = h(X), we can express the relationship between $\frac{\partial \mathbf{L}}{\partial \mathbf{X}}$ and $\frac{\partial \mathbf{L}}{\partial \mathbf{Y}}$ as follows:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X}_{ij}} = \frac{\partial \mathbf{L}}{\partial \mathbf{Y_{ij}}} * \frac{\partial \mathbf{Y_{ij}}}{\partial \mathbf{X_{ij}}} = \frac{\partial \mathbf{L}}{\partial \mathbf{h_{ij}}} * \frac{\partial \mathbf{h_{ij}}}{\partial \mathbf{X_{ij}}}$$

$$\frac{\partial \mathbf{Y_{ij}}}{\partial \mathbf{X_{ij}}} = \frac{\partial \mathbf{h}(\mathbf{X_{ij}})}{\partial \mathbf{X_{ij}}} = \mathbf{h}'(\mathbf{X_{ij}})$$

Therefore, the closed-form expression for $\frac{\partial \mathbf{L}}{\partial \mathbf{X}}$ in terms of $\frac{\partial \mathbf{L}}{\partial \mathbf{Y}}$ is:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X}} = \frac{\partial \mathbf{L}}{\partial \mathbf{Y}} * \mathbf{h}'(\mathbf{X})$$

ii) ReLU activation function is defined by: h(x) = max(0, x). The ReLU function has a specific derivative depending on the input value.
h'(x) is defined:

$$h'(x) = \{\ 1, if$$

x 0
0, if $x \leq 0$
Using this derivative, we can write the closed-form expression for the ReLU activation function.
In general:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X}} = \frac{\partial \mathbf{L}}{\partial \mathbf{Y}} * \mathbf{h}'(\mathbf{X})$$

If $\mathbf{X_ij} > \mathbf{0}$:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X_ij}} = \frac{\partial \mathbf{L}}{\partial \mathbf{Y_ij}}$$

else:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X_ij}} = 0$$

In both cases, the expression involves the element-wise product (Hadamard product) between $\frac{\partial \mathbf{L}}{\partial \mathbf{Y}}$ and h'(X), which means each element of $\frac{\partial \mathbf{L}}{\partial \mathbf{X}}$ is obtained by multiplying the corresponding elements of $\frac{\partial \mathbf{L}}{\partial \mathbf{Y}}$ and h'(X).

2.c.

i) For a softmax module where $\mathbf{Y_ij} = [\mathbf{softmax}(\mathbf{X})]_\mathbf{ij}$ we can find a closed-form expression for $\frac{\partial \mathbf{L}}{\partial \mathbf{X}}$ in terms of $\frac{\partial \mathbf{L}}{\partial \mathbf{Y}}$. We know that the softmax function is defined as follows for each element (i,j):

$$SoftMax(\mathbf{X_ij}) = \frac{\mathbf{e}^{\mathbf{X_ij}}}{\sum_k \mathbf{e}^{\mathbf{X_ik}}}$$

By the chain rule:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X_pq}} = \frac{\partial \mathbf{L}}{\partial \mathbf{Y_pq}} * \frac{\partial \mathbf{Y}_{pq}}{\partial \mathbf{X}_{pq}}$$

Let's proceed with the derivation:

$$\frac{\partial \mathbf{Y}_{cd}}{\partial \mathbf{X}_{pq}} = \frac{\partial}{\partial \mathbf{X}_{pq}} * (\frac{\mathbf{e}^{\mathbf{X_cd}}}{\sum_k \mathbf{e}^{\mathbf{X_ck}}}) = \frac{\frac{\partial \mathbf{e}^{\mathbf{X_cd}}}{\partial \mathbf{X}_{pq}} * \sum_k \mathbf{e}^{\mathbf{X_ck}} - \frac{\partial \sum_k \mathbf{e}^{\mathbf{X_ck}}}{\partial \mathbf{X}_{pq}} * \mathbf{e}^{\mathbf{X_cd}}}{(\sum_k \mathbf{e}^{\mathbf{X_ck}})^2}$$

$$= \frac{\delta_{cp} * \delta_{pq} * \mathbf{e^{X_{cd}}} * \sum_k \mathbf{e^{X_{ck}}} - \sum_k \delta_{cp} * \delta_{kq} * \mathbf{e^{X_{ck}}} * \mathbf{e^{X_{cd}}}}{(\sum_k \mathbf{e^{X_{ck}}})^2}$$

$$= \frac{\delta_{cp} * \delta_{pq} * \mathbf{e^{X_{cd}}} * \sum_k \mathbf{e^{X_{ck}}} - \delta_{cp} * \mathbf{e^{X_{cq}}} * \mathbf{e^{X_{cd}}}}{(\sum_k \mathbf{e^{X_{ck}}})^2}$$

Now, to find the closed-form expression for $\frac{\partial \mathbf{L}}{\partial \mathbf{X}}$, we can use the chain rule. We have:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X_{pq}}} = \sum_{c,d} \frac{\partial \mathbf{L}}{\partial \mathbf{Y_{cd}}} * \frac{\delta_{cp} * \delta_{pq} * \mathbf{e^{X_{cd}}} * \sum_k \mathbf{e^{X_{ck}}} - \delta_{cp} * \mathbf{e^{X_{cq}}} * \mathbf{e^{X_{cd}}}}{(\sum_k \mathbf{e^{X_{ck}}})^2} =$$

$$\sum_d \frac{\partial \mathbf{L}}{\partial \mathbf{Y_{pd}}} * \frac{\delta_{dq} * \mathbf{e^{X_{pd}}} * \sum_k \mathbf{e^{X_{pk}}} - \mathbf{e^{X_{pq}}} * \mathbf{e^{X_{pd}}}}{(\sum_k \mathbf{e^{X_{pk}}})^2} =$$

$$\sum_d \frac{\partial \mathbf{L}}{\partial \mathbf{Y_{pd}}} * \frac{\delta_{dq} * \mathbf{e^{X_{pd}}} * \sum_k \mathbf{e^{X_{pk}}}}{(\sum_k \mathbf{e^{X_{pk}}})^2} - \sum_d \frac{\partial \mathbf{L}}{\partial \mathbf{Y_{pd}}} * \frac{\mathbf{e^{X_{pq}}} * \mathbf{e^{X_{pd}}}}{(\sum_k \mathbf{e^{X_{pk}}})^2} =$$

$$\frac{\partial \mathbf{L}}{\partial \mathbf{Y_{pd}}} * \frac{\mathbf{e^{X_{pq}}}}{\sum_k \mathbf{e^{X_{pk}}}} - \sum_d \frac{\partial \mathbf{L}}{\partial \mathbf{Y_{pd}}} * \frac{\mathbf{e^{X_{pq}}} * \mathbf{e^{X_{pd}}}}{(\sum_k \mathbf{e^{X_{pk}}})^2} =$$

$$\frac{\partial \mathbf{L}}{\partial \mathbf{Y_{pq}}} * softmax(X_{pq}) - \sum_d \frac{\partial \mathbf{L}}{\partial \mathbf{Y_{pd}}} * softmax(X_{pd}) * softmax(X_{pq}) =$$

$$[\frac{\partial \mathbf{L}}{\partial \mathbf{Y}} \circ softmax(X)]_{pq} - [[\frac{\partial \mathbf{L}}{\partial \mathbf{Y}} \circ softmax(X)] * 1]_{pq} \circ softmax(X_{pq}) =$$

$$[\frac{\partial \mathbf{L}}{\partial \mathbf{Y}} \circ softmax(X)] - [[\frac{\partial \mathbf{L}}{\partial \mathbf{Y}} \circ softmax(X)] * 1] \circ softmax(X)$$

ii) To find the closed form expression for $\frac{\partial \mathbf{L}}{\partial \mathbf{X}}$ in the case of the categorical cross-entropy loss, we can break it down step by step.
Let's consider the categorical cross-entropy loss function for a single sample:

$$\mathbf{L_i} = \sum_k \mathbf{T}_{ik} * \mathbf{log}(\mathbf{X_i k})$$

where:

- $\mathbf{L}$ is the loss
- $\mathbf{T}_{ik}$ is the target for class k (1 for the true class, 0 for others)
- $\mathbf{X_i k}$ is the predicted probability for class k.

The general expression of L is:

$$\mathbf{L} = \frac{1}{S} * \sum_i \mathbf{L}_i$$

To calculate the derivative $\frac{\partial \mathbf{L}}{\partial \mathbf{X}}$, we need to take the derivative of L with respect to each element of X.
Let's consider the derivative of L where $\mathbf{j} = \mathbf{k}$ with respect to $\mathbf{X_i k}$:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X_i k}} = \frac{\mathbf{T}_{ik}}{\mathbf{X_i k}}$$

And for all other elements $\mathbf{X_i j}$ where $\mathbf{j} \neq \mathbf{k}$:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X_i k}} = 0$$

Therefore, the closed-form expression for $\frac{\partial \mathbf{L}}{\partial \mathbf{X}}$ can be represented as:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X_{pq}}} = -\frac{1}{S} * \sum_{i,k} \frac{\mathbf{T_{ik}}}{\mathbf{X_{ik}}} = -\frac{1}{S} * \sum_{i,k} \delta_{kq}\delta_{ip} \frac{\mathbf{T_{ik}}}{\mathbf{X_{ik}}} =$$

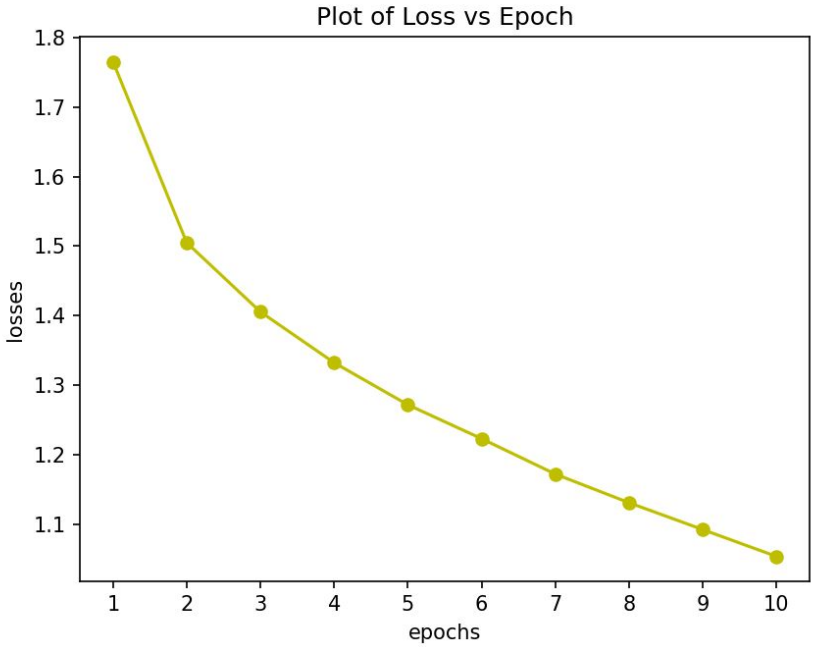
$$-\frac{1}{S} * \sum_{k} \delta_{kq} \frac{\mathbf{T_{pk}}}{\mathbf{X_{pk}}} = -\frac{1}{S} * \frac{T_{pk}}{X_{pk}}$$

$$\frac{\partial \mathbf{L}}{\partial \mathbf{X}} = -\frac{1}{S} * \mathbf{T} \oslash X$$

This expression indicates that the derivative of the loss with respect to each element of X is simply the negative target probability divided by the predicted probability for the true class - and zero for all other classes.

1 Q3

Plot for the Numpy model with default values of parameters (one hidden layer, 128 hidden units, 10 epochs, learning rate 0.1):
Loss Curve:

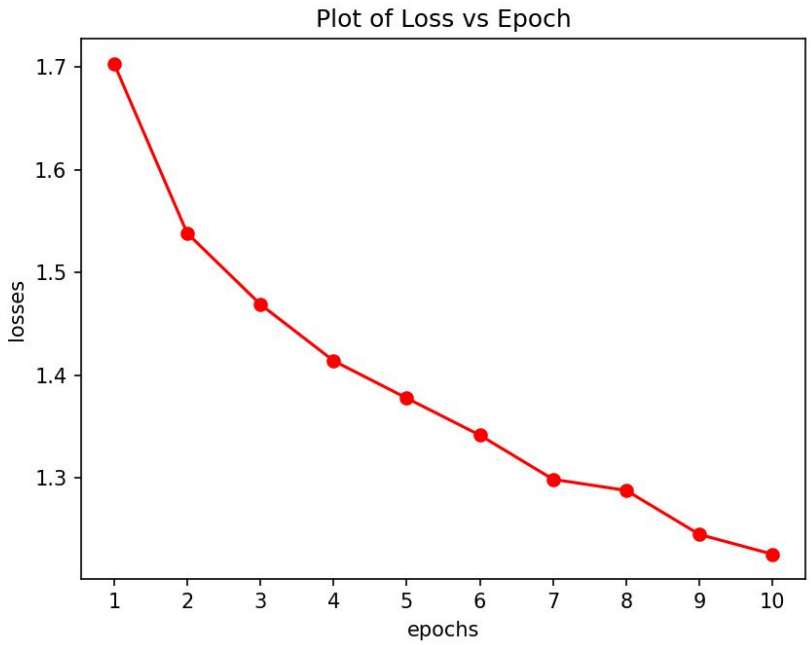


Test Accuracy: 0.4884

Q4

1.1 4.a.

Plot for the for the PyTorch model with default values of parameters (one hidden layer, 128 hidden units, 10 epochs, learning rate 0.1):
Loss Curve:



Test Accuracy: 0.4966

4.b.1

The learning rate determines the step size at which the model parameters are updated during the training process. The choice of an appropriate learning rate is essential because it can significantly impact the convergence of the model.

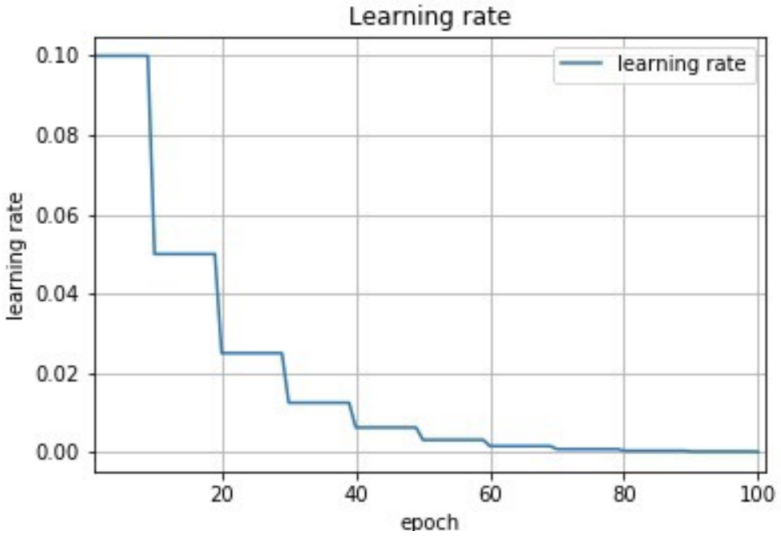
On one hand, if the learning rate is set too high, it can cause the loss to fluctuate or even diverge. The large updates in parameters' values can lead to overshooting the optimal solution, resulting in oscillations and instability during training. The model may struggle to converge or settle into local minima, leading to sub-optimal performance. As a result, the network will suffer from poor generalization capability.

On the other hand, if the learning rate is set too low, it can lead to a significantly slow convergence process. The model may require a lot of training iterations to reach an acceptable level of performance, increasing the overall training time. like in a high learning rate, the model may settle into local minima.

Therefore, it is important to determine a balanced learning rate in order to avoid all the above-mentioned problems.

4.b.2

One commonly used schedule is the "Step Decay" schedule. In this schedule, the learning rate is reduced by a certain factor at specific epochs. Here's an example of a step decay schedule with a visual representation:



The step decay schedule is often used to gradually reduce the learning rate during training. It helps the model to make more precise adjustments to the parameters as it approaches convergence. By reducing the learning rate over time, the schedule helps prevent overshooting and fine-tunes the model's parameters for better convergence and performance.

1.2 4.b.3

After conducting an investigation to examine the effect of different learning rates , it was found that among the nine different learning rates - [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 0.5, 1, 5, 10], a learning rate of 0.1 exhibited the best performance for the MLP. Each point in the graph appears in 4.b.4 has a larger learning rate than the previous one, and the 6^{th} point represents learning rate of 0.1.

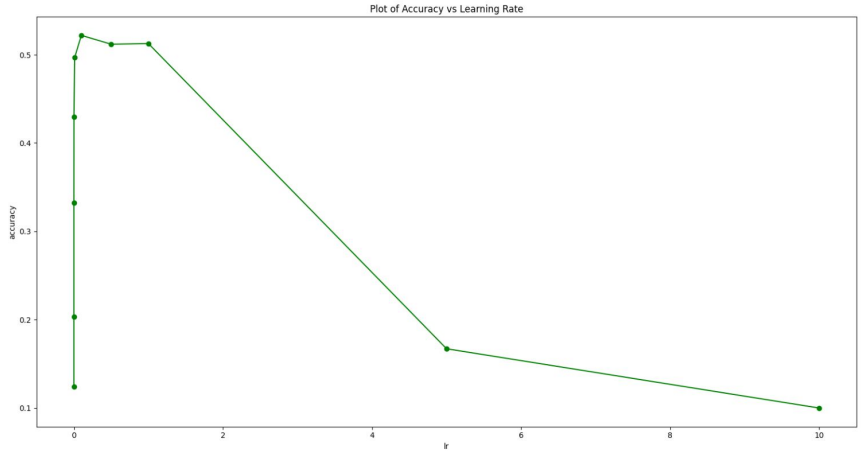
By evaluating the model's performance on a test set, it was observed that a learning rate of 0.1 resulted in faster convergence and superior generalization compared to the other tested options. This learning rate allowed the model to effectively navigate the parameter space and find an optimal solution while avoiding the pitfalls of overly slow or aggressive weight updates.

It is important to note that the choice of the best learning rate can vary depending on the specific dataset and model architecture. The selection of 0.1 as the optimal learning rate for this investigation signifies its compatibility with the given MLP architecture and the characteristics of the problem at hand. Other datasets or model architectures might require a different learning

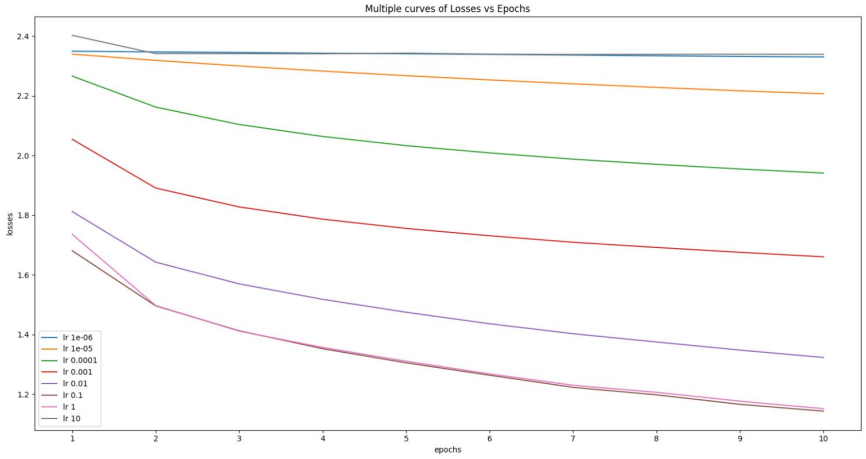
rate to achieve optimal performance.

1.3 4.b.4

Best validation accuracy as function of the learning rates:



Loss curves with different learning rates over epochs:



Q5

To prove that the second moment of the transformed variable, $\mathbf{E}[\mathbf{b^2}] = \frac{\sigma^2}{2}$, we can use the law of total expectation and the definition of the ReLU function. First, let's start with the definition of the ReLU function:

$$\mathbf{b} = \mathbf{ReLU}[\mathbf{f}] = \mathbf{max}(\mathbf{f}, \mathbf{0})$$

The ReLU function takes the maximum value between f and 0. Therefore, b will be equal to f if f is positive, and b will be equal to 0 if f is negative.
Now, let's calculate the second moment of the transformed variable, $\mathbf{E}[\mathbf{b^2}]$:

$$\mathbf{E}[\mathbf{b^2}] = \mathbf{E}[\mathbf{max^2}(\mathbf{f}, \mathbf{0})]$$

Using the law of total expectation, we can expand this expression as:

$$\mathbf{E}[\mathbf{b^2}] = \mathbf{E}[\mathbf{max^2}(\mathbf{f}, \mathbf{0})] = \mathbf{E}[\mathbf{max^2}(\mathbf{f}, \mathbf{0})|\mathbf{f} \geq \mathbf{0}]*\mathbf{P}(\mathbf{f} \geq \mathbf{0})+\mathbf{E}[\mathbf{max^2}(\mathbf{f}, \mathbf{0})|\mathbf{f} < \mathbf{0}]*\mathbf{P}(\mathbf{f} < \mathbf{0})$$

we know that if $f < 0$ then $\mathbf{E}[\mathbf{max^2}(\mathbf{f}, \mathbf{0})|\mathbf{f} < \mathbf{0}] = 0$ so:

$$\mathbf{E}[\mathbf{b^2}] = \mathbf{E}[\mathbf{max^2}(\mathbf{f}, \mathbf{0})] = \mathbf{E}[\mathbf{max^2}(\mathbf{f}, \mathbf{0})|\mathbf{f} \geq \mathbf{0}] * \mathbf{P}(\mathbf{f} < \mathbf{0})$$

In this case, the ReLU function will be equal to f:

$$\mathbf{E}[\mathbf{b^2}] = \mathbf{E}[\mathbf{f^2}|\mathbf{f} \geq \mathbf{0}] * \mathbf{P}(\mathbf{f} \geq \mathbf{0})$$

Now, let's consider the probability $P(f \geq 0)$. Since $\mathbf{E}[\mathbf{f}] = 0$, and assuming the distribution of f is symmetric around 0. Therefore,
P$P(f \geq 0) = P(f < 0) = 0.5$.

$$\mathbf{E}[\mathbf{b^2}] = \mathbf{E}[\mathbf{f^2}|\mathbf{f} \geq \mathbf{0}] * 0.5$$

Now, let's look at $\mathbf{E}[\mathbf{f^2}]$:

$$\mathbf{E}[\mathbf{f^2}] = \mathbf{Var}[\mathbf{f}] + \mathbf{E}[\mathbf{f}]^2 = \sigma^2 + 0^2 = \sigma^2$$

Substituting this back into the previous equation, we have:

$$\mathbf{E}[\mathbf{b^2}] = \sigma^2 * 0.5$$

Therefore, we have proved that the second moment of the transformed variable,$\mathbf{E}[\mathbf{b^2}]$, is equal to σ^2.