Ben-Gurion University - IEM Department
Introduction to Deep Learning (364-2-1071)
2023 Semester B
Assignment 1

Names: Peleg Eliyahou, Stav Dratwa

Question 1

a) Given a vector $\mathbf{x} \in \mathbb{R}^n$ and square matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$, evaluate $\frac{\partial(\mathbf{x^T B x})}{\partial \mathbf{x}}$
Answer:

- $x^T Bx = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i B_{ij} x_j$
- $\frac{\partial(x^T Bx)}{\partial x_k} = \frac{\partial}{\partial x_k}(\sum_{i=1}^{n} \sum_{j=1}^{n} x_i B_{ij} x_j) = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial x_i}{\partial x_k} B_{ij} x_j + x_i B_{ij} \frac{\partial x_j}{\partial x_k}$
- $\frac{\partial(x^T Bx)}{\partial x_k} = \sum_{i=1}^{n} \sum_{j=1}^{n} \delta_{ik} B_{ij} x_j + x_i B_{ij} \delta_{jk}$
- $\frac{\partial(x^T Bx)}{\partial x_k} = \sum_{i=1}^{n} \sum_{j=1}^{n} B_{kj} x_j + x_i B_{ik} = (Bx^T)_k + (x^T B)_k$
- $(Bx^T)_k + (x^T B)_k = (x^T B^T)_k + (x^T B)_k$
- $\frac{\partial(\mathbf{x^T B x})}{\partial \mathbf{x}} = x^T(B^T + B)$

b) Given matrices $\mathbf{V} \in \mathbb{R}^{n \times m}, \mathbf{X} \in \mathbb{R}^{m \times p}$ and $\mathbf{W} \in \mathbb{R}^{p \times n}$,find an expression for $\frac{\partial tr(\mathbf{VXW})}{\partial \mathbf{X}}$
Answer:

- $tr(VXW) = \sum_{i=1}^{n} (VXW)_{ii}$
- $(VXW)_{ii} = \sum_{k=1}^{m} V_{ik} (XW)_{ki}$
- $(XW)_{ki} = \sum_{l=1}^{p} X_{kl} W_{li}$
- $tr(VXW) = \sum_{i=1}^{n} \sum_{k=1}^{m} \sum_{l=1}^{p} V_{ik} X_{kl} W_{li}$
- $\frac{\partial tr(VXW)}{\partial X_{rs}} = \frac{\partial}{\partial X_{rs}} \sum_{i=1}^{n} \sum_{k=1}^{m} \sum_{l=1}^{p} V_{ik} X_{kl} W_{li}$
- $\frac{\partial}{\partial X_{rs}} \sum_{i=1}^{n} \sum_{k=1}^{m} \sum_{l=1}^{p} V_{ik} X_{kl} W_{li} = \sum_{i=1}^{n} \sum_{k=1}^{m} \sum_{l=1}^{p} \delta_{kr} \delta_{ls} V_{ik} W_{li}$
- $\frac{\partial tr(VXW)}{\partial X_{rs}} = \sum_{i=1}^{n} V_{ir} W_{si}$
- $\frac{\partial tr(VXW)}{\partial X} = \mathbf{VW^T}$

c) For a vector $\mathbf{W} \in \mathbb{R}^n$ and its Euclidean norm $||\mathbf{W}|| := \sqrt{\mathbf{W^T W}}$, calculate $\frac{\partial ||\mathbf{W}||}{\partial \mathbf{W}}$

Answer:

- $\frac{\partial ||W||}{\partial W} = \frac{\partial}{\partial W_k} \sqrt{\sum_{i=1}^{n} W_i^2}$
- $\frac{\partial ||W||}{\partial W} = \frac{1}{2\sqrt{W^T W}} \frac{\partial}{\partial W_k} (W^T W) = \frac{1}{2\sqrt{W^T W}} \frac{\partial}{\partial W_k} (\sum_{i=1}^{n} W_i^2)$
- $\frac{\partial ||W||}{\partial W} = \frac{1}{2\sqrt{W^T W}} 2W_k = \frac{W_k}{W^T W}$
- $\frac{\partial ||W||}{\partial W} = \frac{\mathbf{W}}{\sqrt{\mathbf{W^T W}}} = \frac{\mathbf{W}}{||\mathbf{W}||}$

d) Let $\mathbf{S}$ be a square matrix, find an expression for $\frac{\partial tr(\mathbf{S})}{\partial \mathbf{S}}$

Answer:

- $tr(S) = \sum_{i=1}^{n} S_{i,i}$
- $\frac{\partial tr(S)}{\partial S_{i,j}} = \frac{\partial S}{\partial S_{i,j}} \sum_{k=1}^{n} S_{k,k} = \frac{\partial S}{\partial S_{i,j}} (S_{1,1} + S_{2,2} + \ldots + S_{n,n})$
- $\frac{\partial \operatorname{tr}(S)}{\partial S_{i,j}} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$
- $\frac{\partial \operatorname{tr}(S)}{\partial S} = \mathbf{I}$

Question 2.a Linear Module

a) Consider a linear module as described above. The input and output features are labeled as $\mathbf{X}$ and $\mathbf{Y}$, respectively. Find closed form expressions for

$$\frac{\partial L}{\partial \mathbf{W}}, \frac{\partial L}{\partial \mathbf{b}}, \frac{\partial L}{\partial \mathbf{X}}$$

Answer:

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

- $\frac{\partial L}{\partial \mathbf{W}} = \frac{\partial L}{\partial \mathbf{Y}} \cdot \frac{\partial \mathbf{Y}}{\partial \mathbf{W}}$
- $[\frac{\partial L}{\partial \mathbf{W}}]_{ij} = \sum_{m,n} \{ \frac{\partial L}{\partial \mathbf{Y_{mn}}} \cdot \frac{\partial}{\partial \mathbf{W_{ij}}} (XW^T + B)_{mn} \}$
- $[\frac{\partial L}{\partial \mathbf{W}}]_{ij} = \sum_{m,n} \{ \frac{\partial L}{\partial \mathbf{Y_{mn}}} \cdot \frac{\partial}{\partial \mathbf{W_{ij}}} [\sum_{k} X_{mk} (W^T)_{kn} + B_{mn}] \}$
- $[\frac{\partial L}{\partial \mathbf{W}}]_{ij} = \sum_{m,n} \{ \frac{\partial L}{\partial \mathbf{Y_{mn}}} [\sum_{k} \frac{\partial}{\partial W_{ij}} (X_{mk} W_{nk}) + \frac{\partial}{\partial W_{ij}} B_{mn}] \}$
- $[\frac{\partial L}{\partial \mathbf{W}}]_{ij} = \sum_{m,n} \{ \frac{\partial L}{\partial \mathbf{Y_{mn}}} [\sum_{k} \frac{\partial W_{nk}}{\partial W_{ij}} X_{mk} + 0] \}$

- $[\frac{\partial L}{\partial \mathbf{W}}]_{ij} = \sum_{m,n}\{\frac{\partial L}{\partial \mathbf{Y_{mn}}}[\sum_k \delta_{ni}\delta_{kj}X_{mk}]\}$
- $[\frac{\partial L}{\partial \mathbf{W}}]_{ij} = \sum_m^s \frac{\partial L}{\partial \mathbf{Y}_{mi}}X_{mj}$
- $[\frac{\partial L}{\partial \mathbf{W}}]_{ij} = \sum_m^s [(\frac{\partial L}{\partial \mathbf{Y}})^T]_{im}X_{mj}$

  The gradient of the loss with respect to the weights $\mathbf{W}$ is given by:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{W}} = (\frac{\partial \mathbf{L}}{\partial \mathbf{Y}})^\top \cdot \mathbf{X}$$

$\frac{\partial L}{\partial \mathbf{b}}$

- $\frac{\partial L}{\partial \mathbf{b}} = \frac{\partial L}{\partial \mathbf{Y_{sk}}}\frac{\partial Y_{sk}}{\partial \mathbf{b_i}} = \frac{\partial L}{\partial \mathbf{Y_{sk}}}\frac{\partial (XW^T+b)_{sk}}{\partial \mathbf{b_i}} = \frac{\partial L}{\partial \mathbf{Y_{sk}}}\frac{\partial \sum_c (X_{sc}W_{kc}+b_k)}{\partial \mathbf{b_i}} = \frac{\partial L}{\partial \mathbf{Y_{sk}}}(0+\delta_{ki}) = \frac{\partial L}{\partial \mathbf{Y_{sk}}}\delta_{ki}$

The gradient of the loss with respect to the biases $\mathbf{b}$ is given by:

$$\frac{\partial L}{\partial \mathbf{b}} = \mathbf{1} \cdot \frac{\partial L}{\partial \mathbf{Y}}$$

$\mathbf{1}$ is a all ones vector.

$\frac{\partial L}{\partial \mathbf{X}}$

- $[\frac{\partial L}{\partial \mathbf{X}}]_{ij} = \frac{\partial L}{\partial \mathbf{X_{ij}}} = \sum_{m,n}\{\frac{\partial L}{\partial \mathbf{Y_{mn}}}\frac{\partial Y_{mn}}{\partial \mathbf{X_{ij}}}\}$
- $[\frac{\partial L}{\partial \mathbf{X}}]_{ij} = \sum_{m,n}\{\frac{\partial L}{\partial \mathbf{Y_{mn}}} \cdot \frac{\partial}{\partial \mathbf{X_{ij}}}[(\sum_k X_{mk}W_{nk}) + B_{mn}]\}$
- $[\frac{\partial L}{\partial \mathbf{X}}]_{ij} = \sum_{m,n}\{\frac{\partial L}{\partial \mathbf{Y_{mn}}}[\sum_k \frac{\partial X_{mk}}{\partial X_{ij}}W_{nk} + 0]\}$
- $[\frac{\partial L}{\partial \mathbf{X}}]_{ij} = \sum_{m,n}\{\frac{\partial L}{\partial \mathbf{Y_{mn}}}[\sum_k \delta_{mi}\delta_{kj}W_{nk}]\}$
- $[\frac{\partial L}{\partial \mathbf{X}}]_{ij} = \sum_n^s \frac{\partial L}{\partial \mathbf{Y}_{in}}W_{nj}$

  The gradient of the loss with respect to the inputs $\mathbf{X}$ is given by:

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

Question 2.b Activation Module

b) Consider an element-wise activation function h. The activation module has input and output features labeled by $\mathbf{X}$ and $\mathbf{Y}$, respectively. I.e. $\mathbf{Y} = \mathrm{h}(\mathbf{X}) \Rightarrow \mathbf{Y_{ij}} = \mathrm{h}(\mathbf{X_{ij}})$. Find a closed form expression for

$$\frac{\partial L}{\partial \mathbf{X}}$$

in terms of the gradient of the loss with respect to the output features $\frac{\partial L}{\partial \mathbf{Y}}$ provided by the next module. Assume the gradient has the same shape as $\mathbf{X}$. Provide answers both for i) a generic activation function h and ii) the ReLU case h(x) = max(0, x). [Hint: You might need to write your answer in terms of a Hadamard product.]

Answer:

i) A generic activation function h

- $[\frac{\partial L}{\partial \mathbf{X}}]_{ij} = \frac{\partial L}{\partial \mathbf{X_{ij}}} = \sum_{m,n} \frac{\partial L}{\partial \mathbf{Y_{mn}}} \frac{\partial Y_{mn}}{\partial \mathbf{X_{ij}}} = \sum_{m,n} \frac{\partial L}{\partial \mathbf{Y_{mn}}} \frac{\partial}{\partial \mathbf{X_{ij}}} [h(X_{mn})]$
- $[\frac{\partial L}{\partial \mathbf{X}}]_{ij} = \frac{\partial L}{\partial \mathbf{X_{ij}}} = \sum_{m,n} \frac{\partial L}{\partial \mathbf{Y_{mn}}} [\frac{\partial h(X_{mn})}{\partial X_{ij}}]$

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

where $\odot$ represents the Hadamard (element-wise) product, and $h'(\mathbf{X})$ is a matrix of the same shape as $\mathbf{X}$ with the element-wise derivative of the activation function h with respect to its argument evaluated at $\mathbf{X}$.

ii) The ReLU case h

- $[\frac{\partial L}{\partial \mathbf{X}}]_{ij} = \frac{\partial L}{\partial \mathbf{X_{ij}}} = \sum_{m,n} \frac{\partial L}{\partial \mathbf{Y_{mn}}} \frac{\partial Y_{mn}}{\partial \mathbf{X_{ij}}} = \sum_{m,n} \frac{\partial L}{\partial \mathbf{Y_{mn}}} \frac{\partial}{\partial \mathbf{X_{ij}}} [h(X_{mn})]$
- $[\frac{\partial L}{\partial \mathbf{X}}]_{ij} = \frac{\partial L}{\partial \mathbf{X_{ij}}} = \sum_{m,n} \frac{\partial L}{\partial \mathbf{Y_{mn}}} [\frac{\partial h(X_{mn})}{\partial X_{ij}}]$

$$\frac{\partial L}{\partial \mathbf{X}} = \frac{\partial L}{\partial \mathbf{Y}} \cdot \frac{\partial \mathbf{Y}}{\partial \mathbf{X}} = \frac{\partial L}{\partial \mathbf{Y}} \cdot \frac{\partial (max(0, x))}{\partial \mathbf{X}} = \frac{\partial L}{\partial \mathbf{Y}} \odot \begin{cases} 0, & \text{if } x \geq 0 \\ 1, & \text{if } x > 0 \end{cases}$$

Question 2.c Softmax and Loss Modules

i) Consider a softmax module such that $Y_{ij} = [\text{softmax}(\mathbf{X})]\text{ij}$, where $\mathbf{X}$ is the input and $\mathbf{Y}$ is the output of the module. Find a closed-form expression for $\frac{\partial L}{\partial \mathbf{X}}$ in terms of $\frac{\partial L}{\partial \mathbf{Y}}$ [Hint: The answer might require using an all-ones matrix.]

Answer :

- $\frac{\partial L}{\partial X_{pq}} = \sum_{c,d} \frac{\partial L}{\partial Y_{cd}} \frac{\partial Y_{cd}}{\partial X_{pq}}$
- $\frac{\partial Y_{cd}}{\partial X_{pq}} = \frac{\partial}{\partial X_{pq}} (\frac{e^{X_{cd}}}{\sum_k e^{X_{ck}}})$
- $\frac{\partial}{\partial X_{pq}} \left(\frac{e^{X_{cd}}}{\sum_k e^{X_{ck}}} \right) = \frac{\frac{\partial e^{X_{cd}}}{\partial X_{pq}} \sum_k e^{X_{ck}} - \frac{\partial (\sum_k e^{X_{ck}})}{\partial X_{pq}} e^{X_{cd}}}{\left(\sum_k e^{X_{ck}}\right)^2}$
- $\frac{\partial L}{\partial X_{pq}} = \sum_{c,d} \frac{\partial L}{\partial Y_{cd}} (\frac{\delta_{cp}\delta_{dq} e^{X_{cd}} \sum\limits_k e^{X_{ck}} - \sum\limits_k \delta_{cp}\delta_{kq} e^{X_{ck}} e^{X_{cd}}}{(\sum e^{X_{ck}})^2})$
- $\frac{\partial L}{\partial X_{pq}} = \frac{\partial L}{\partial Y_{pq}} \frac{e^{X_{pq}}}{\sum e^{X_{pk}}} - \sum_d \frac{\partial L}{\partial Y_{pd}} \frac{e^{X_{pq}} e^{X_{pd}}}{(\sum e^{X_{pk}})^2}$
- $\frac{\partial L}{\partial X_{pq}} = \frac{\partial L}{\partial Y_{pq}} Y_{pq} - Y_{pq} \sum\limits_d \frac{\partial L}{\partial Y_{pd}} Y_{pd}$

$$\frac{\partial L}{\partial \mathbf{X}} = [\frac{\partial L}{\partial Y} \odot Y] - [\frac{\partial L}{\partial Y} \cdot 1 \odot Y] \odot [Y]$$

ii) The gradient that kicks the whole backpropagation algorithm off is the one for the loss module itself. The loss module for the categorical cross entropy takes as input $\mathbf{X}$ and returns $L = \frac{1}{S} \sum_i L_i = -\frac{1}{S} \sum_{ik} T_{ik} log(X_{ik})$ Find a closed form expression for $\frac{\partial L}{\partial \mathbf{X}}$. [Hint: You may use element-wise operations.]

Answer :

- $[\frac{\partial \mathbf{L}}{\partial \mathbf{X}}]_{ij} = \frac{\partial}{\partial \mathbf{X}_{ij}} \{-\frac{1}{S} \sum\limits_{i,k} T_{ik} log(X_{ik})\}$
- $[\frac{\partial \mathbf{L}}{\partial \mathbf{X}}]_{ij} = -\frac{1}{S} \sum\limits_{i,k} \frac{\partial}{\partial \mathbf{X}_{ij}} T_{ik} log(X_{ik})$
- $[\frac{\partial \mathbf{L}}{\partial \mathbf{X}}]_{ij} = -\frac{1}{S} \sum\limits_{i,k} T_{ik} \frac{\partial}{\partial \mathbf{X}_{ij}} log(X_{ik})$
- $[\frac{\partial \mathbf{L}}{\partial \mathbf{X}}]_{ij} = -\frac{1}{S} T_{ij} \cdot \frac{\mathbf{1}}{\mathbf{X}_{ij}}$

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

- $\oslash$ is element-wise division between two matrices

Question 3 NumPy implementation

Test accuracy : 0.4733

loss curve :

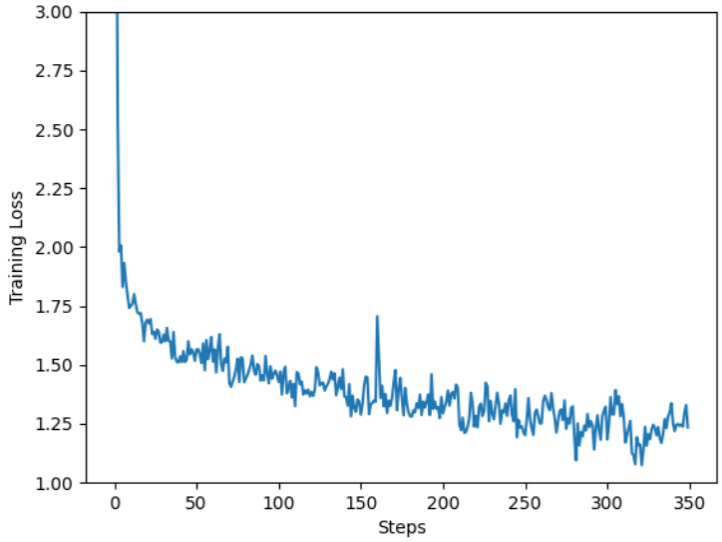

Figure 1: Loss curve

Question 4.a PyTorch MLP

Test accuracy : 0.4917

loss curve :

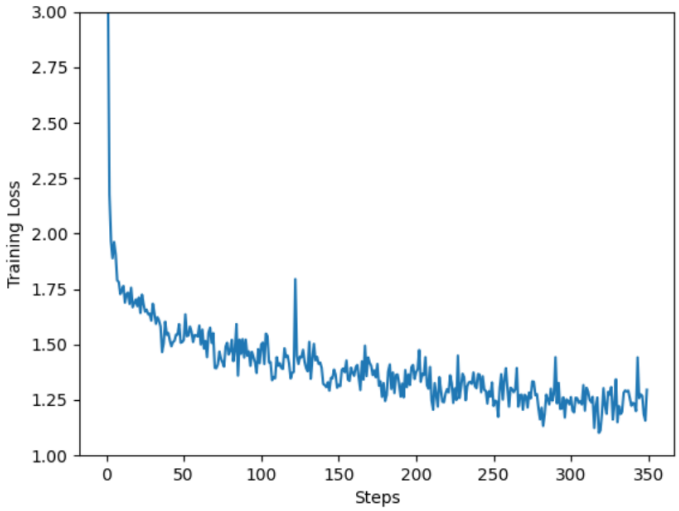

Figure 2: Loss curve

Question 4.b Learning rate

i) Why the learning rate is the most important hyperparameter in your deep neural network? What happens(loss, accuracy, training time) if we set the learning rate too high or too low?

Answer :

The hyperparameter refers to the proportion of change to the partial derivative of the error with respect to the weights in a single iteration. As this value increases, the change in weights will be faster and vice versa. A large value of this parameter can lead to missing the points of minimum error in the learning process, while a value that is too low may require many iterations and may get stuck in a local minimum and be inefficient.

If the learning rate is set too high:

- Loss: The model may fail to converge or even diverge. The loss function may oscillate or increase significantly, resulting in unstable training.
- Accuracy: The model's accuracy may suffer as it fails to find the optimal solution or gets trapped in a sub-optimal region, and might also fail to generalize on new data.
- Training time: The training process may be faster initially due to larger weight updates, but it may take longer to converge due to the instability and the need for multiple iterations to correct the overshooting.

If the learning rate is set too low:

- Loss: The model may converge very slowly or not at all. The loss function may decrease very gradually, requiring a large number of epochs for convergence. This might cause the training process to be much longer.
- Accuracy: The model's accuracy may not reach its potential as it takes longer to learn and adjust to the training data.
- Training time: The training process may take significantly longer as the small updates to the weights require more iterations to reach convergence.

ii) What is the appropriate schedule to adjust the learning rate during training? A figure (epoch for the x-axis and learning rate for the y-axis) could be useful to show the schedule. Please make an explanation for your schedule

Answer :

A schedule is to make the learning rate adaptive to the gradient descent optimization procedure. The model is updated by an optimizer. This plot shows the changes in the learning rate over the different epochs. During training, after each epoch the appropriate learning rate was updated

according to the scheduler. We got the following learning rates:

$$[0.10000, 0.09833, 0.09667, 0.09500, 0.09333, 0.09167, 0.09000, 0.08833, 0.08667, 0.08500]$$

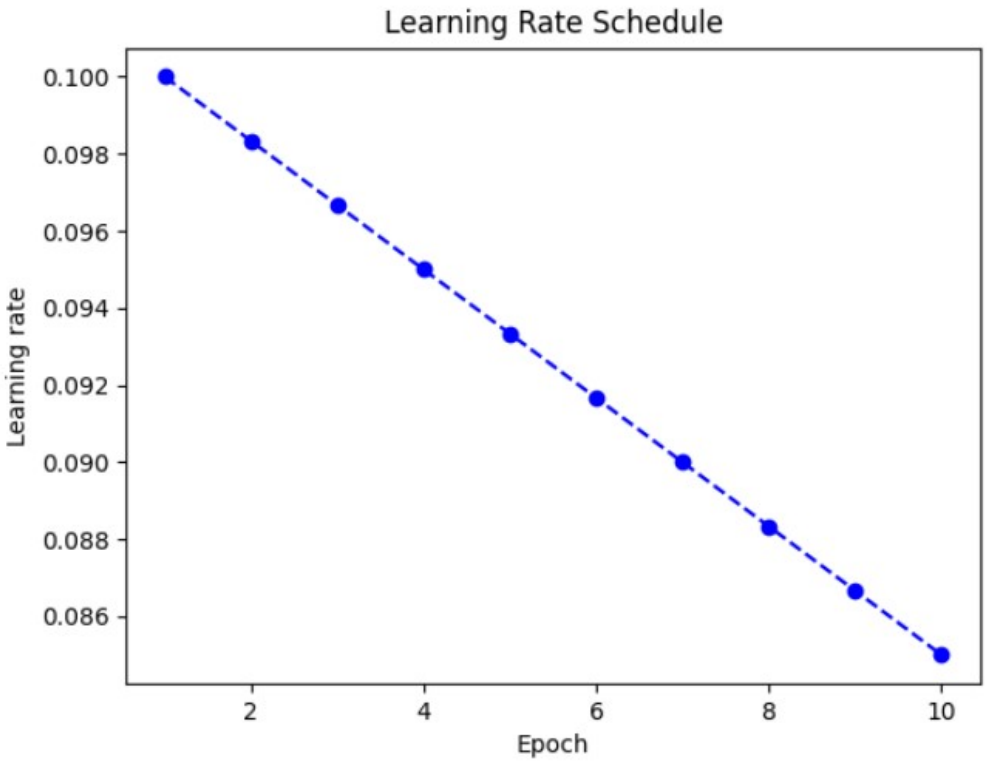

Figure 3: Learning rate schedule

iii) To investigate the effect of the learning rate for deep neural networks, train the model with different learning rates using your built MLP network with stochastic gradient descent. Use 9 different learning rates, from 0.000001 to 100 at equal logarithmic intervals. Find the best learning rate for your model and explain why it is better than other options.

Answer :

We trained the model with different learning rates from 0.000001 to 100 at equal logarithmic intervals. The appropriate plot appears in question 4.iv. Too-small learning rate resulted in a model that didn't converge enough, causing the loss function to remain almost the same even when the iterations keep growing. Too-high learning rate caused the loss function to diverge with infinite values. This leads to our conclusion that the learning rate should be not too-small and not too big, such as around 0.01. This learning rate is better than other options because it gets a small training loss that decays over the iterations, in a relatively stable process.

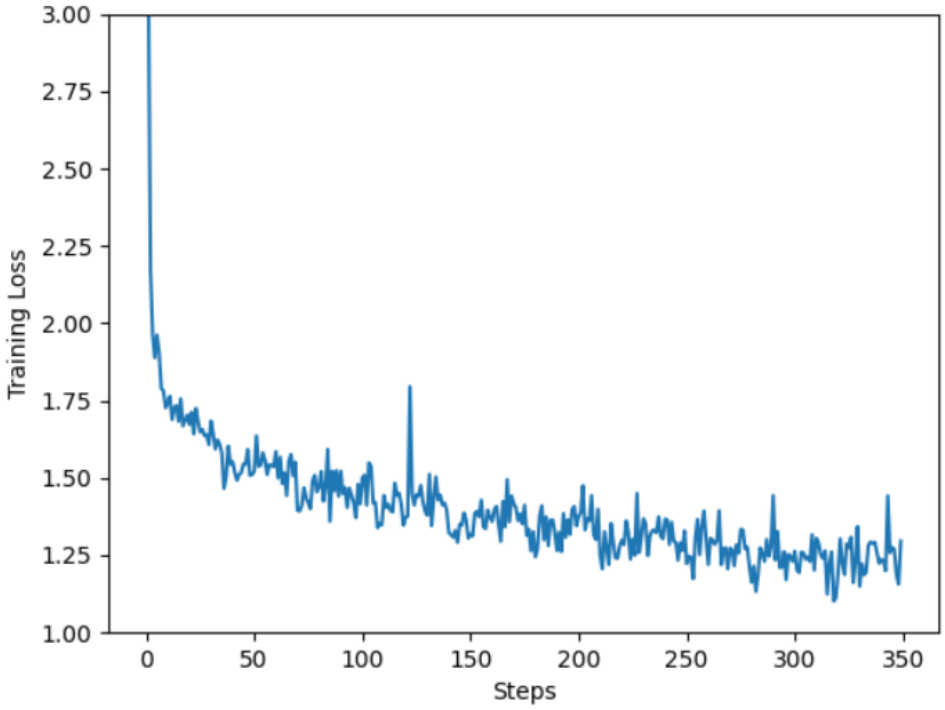

Figure 4: Training Loss vs steps of the best learning rate

iv) Plot two figures:

- Best validation accuracy as function of the learning rates
- Loss curves with different learning rates over time (iteration or epoch)

Answer :

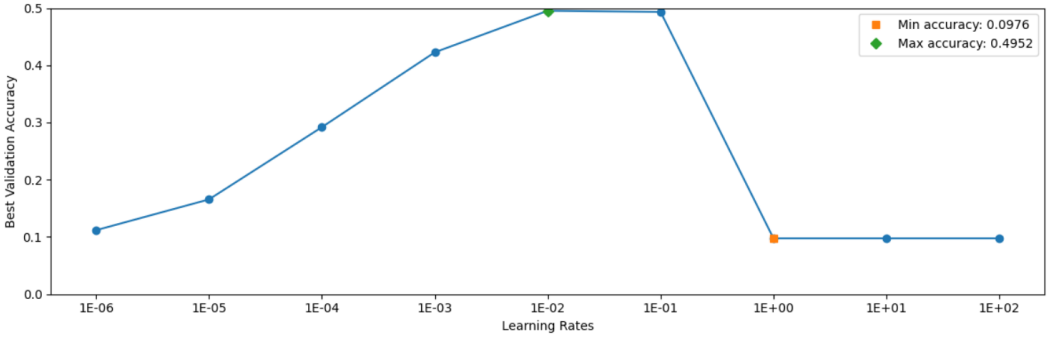

Figure 5: Validation accuracy as function of learning rates

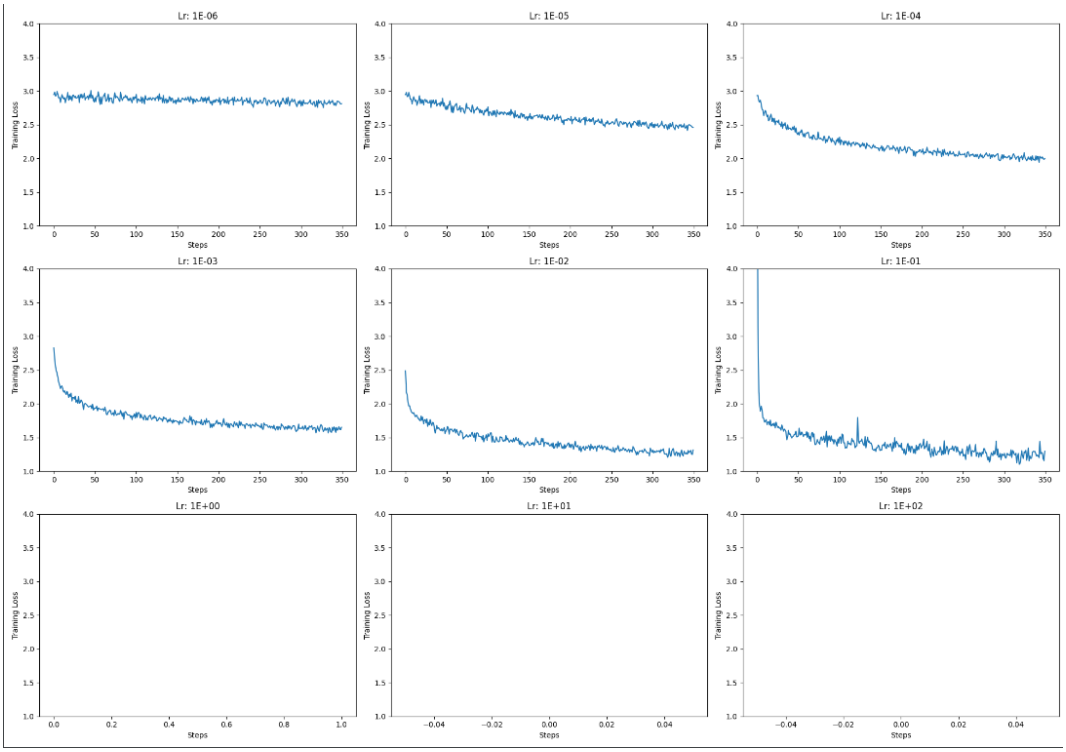

Figure 6: Loss curves with different learning rates over time

Question 5 Initialization

Answer :

$$b = \begin{cases} f & \text{if } f \geq 0 \\ 0 & \text{if } f < 0 \end{cases}$$

- $\mathrm{E}[b^2] = \int_{-\infty}^{\infty} b^2 p(b) db$
- $\mathrm{E}[b^2] = \int_{-\infty}^{\infty} (max(f, 0))^2 p(f) df$
- Case 1: $f \geq 0$
  $\mathrm{E}[b^2] = \int_{-\infty}^{0} 0 \cdot p(f) df + \int_{0}^{\infty} f^2 p(f) df$
  Since the probability density is symmetric around 0, i.e., $p(f) = p(-f)$:

$$E[f^2] = 2 \cdot \int_{0}^{\infty} f^2 p(f) df$$

- Case 2: $f < 0$
  When $f < 0$, $\max(f, 0) = 0$, f < 0 does not contribute to the Integral

$$\int_{-\infty}^{0} f^2 p(f) = \int_{-\infty}^{0} 0^2 p(f)$$

- $Var[f] = \sigma^2$
- $Var[f] = \int_{\infty}^{-\infty}(f)^2 p(f)df + \mu^2 = \sigma^2$
- Since $\mu = 0, Var[f] = \int_{\infty}^{-\infty}(f)^2 p(f)df = \sigma^2$
- Combine 2 cases:

$$E[b^2] = \int_0^{\infty} f^2 p(f)df$$

$$E[b^2] = \int_0^{\infty} f^2 p(f)df = \frac{1}{2} \cdot \int_{\infty}^{-\infty}(f)^2 p(f)df = \frac{\sigma^2}{2}$$

- In conclusion, the second moment of the transformed variable b (ReLU$[f] = \max(f, 0)$) is E$[b^2] = \frac{\sigma^2}{2}$.