SAPIENZA
UNIVERSITÀ DI ROMA

Advanced Machine Learning - Assignment #1

Group Members
Stefano D'Arrigo 1960500
Paola Antonicoli 1796554
Simone Fiorellino 1960415
Jeremy Sapienza 1960498

Oct 27 2021

1 Preliminary notions

Before proceeding and answering to the questions, we present some of the rules and derivations which will be exploited in the next sections. Let $W \in \mathbb{R}^{n \times m}$ and $A \in \mathbb{R}^{m \times l}$ be two real-value matrices. Recall that $Z = W \cdot A$ is defined as:

$$z_{i,j} = \mathbf{w}_{i,:} \cdot \mathbf{a}_{:,j} = \sum_{k=1}^{m} w_{i,k} \cdot a_{k,j} \tag{1}$$

The derivative of $z_{i,j}$ - which is a scalar - with respect to the vector $\mathbf{w}_{i,:}$ is defined as:

$$\begin{aligned}
&\frac{\partial z_{i,j}}{\partial \mathbf{w}_{i,:}} = \frac{\partial z_{i,j}}{\partial (w_{i,1}, w_{i,2}, ..., w_{i,m})} = \frac{\partial \mathbf{w}_{i,:} \cdot \mathbf{a}_{:,j}}{\partial (w_{i,1}, w_{i,2}, ..., w_{i,m})} \\
&= \frac{\partial (w_{i,1}a_{1,j} + w_{i,2}a_{2,j} + ... + w_{i,m}a_{m,j})}{\partial (w_{i,1}, w_{i,1}, ..., w_{i,m})} \\
&= \frac{\partial \left(\sum_{k=1}^{m} w_{i,k} \cdot a_{k,j}\right)}{\partial (w_{i,1}, w_{i,2}, ..., w_{i,m})} \\
&= \left[\frac{\partial \left(\sum_{k=1}^{m} w_{i,k} \cdot a_{k,j}\right)}{\partial w_{i,1}}, \frac{\partial \left(\sum_{k=1}^{m} w_{i,k} \cdot a_{k,j}\right)}{\partial w_{i,2}}, ..., \frac{\partial \left(\sum_{k=1}^{m} w_{i,k} \cdot a_{k,j}\right)}{\partial w_{i,m}}\right] \\
&= [a_{1,j}, a_{2,j}, ..., a_{m,j}] = \mathbf{a}_{:,j}^T
\end{aligned} \tag{2}$$

The same holds for $\frac{\partial z_{i,j}}{\partial \mathbf{a}_{:j}}$.
Moving forward, the derivative of $\mathbf{z}_{i,:}$ - which is a vector - with respect to the vector $\mathbf{w}_{i,:}$ is defined as:

$$\begin{aligned}
&\frac{\partial \mathbf{z}_{i,:}}{\partial \mathbf{w}_{i,:}} = \frac{\partial \mathbf{z}_{i,:}}{\partial (w_{i,1}, w_{i,2}, ..., w_{i,m})} \\
&= \left[\frac{\partial z_{i,1}}{\partial (w_{i,1}, w_{i,2}, ..., w_{i,m})}, \frac{\partial z_{i,2}}{\partial (w_{i,1}, w_{i,2}, ..., w_{i,m})}, ..., \frac{\partial z_{i,m}}{\partial (w_{i,1}, w_{i,2}, ..., w_{i,m})}\right] \\
&= [[a_{1,1}, a_{2,1}, ..., a_{m,1}], ..., [a_{1,l}, a_{2,l}, ..., a_{m,l}]] \\
&= \left[\mathbf{a}_{:,1}^T, ..., \mathbf{a}_{:,l}^T\right] = A^T
\end{aligned} \tag{3}$$

We can, then, derive a row $\mathbf{z}_{i,:} \in Z$ w.r.t the entire matrix W. Let's write down the product $W \cdot A$ explicitly:

$$Z = W \cdot A = \begin{pmatrix} w_{1,1} & w_{1,2} & ... & w_{1,m} \\ w_{2,1} & w_{2,2} & ... & w_{2,m} \\ ... & ... & ... & ... \\ w_{n,1} & w_{n,2} & ... & w_{n,m} \end{pmatrix} \cdot \begin{pmatrix} a_{1,1} & a_{1,2} & ... & a_{1,l} \\ a_{2,1} & a_{2,2} & ... & a_{2,l} \\ ... & ... & ... & ... \\ a_{m,1} & a_{m,2} & ... & a_{m,l} \end{pmatrix} \tag{4}$$

$$\mathbf{z}_{i,:} = \mathbf{w}_{i,:} \cdot A = \begin{pmatrix} w_{i,1} & w_{i,2} & ... & w_{i,m} \end{pmatrix} \cdot \begin{pmatrix} a_{1,1} & a_{1,2} & ... & a_{1,l} \\ a_{2,1} & a_{2,2} & ... & a_{2,l} \\ ... & ... & ... & ... \\ a_{m,1} & a_{m,2} & ... & a_{m,l} \end{pmatrix} \tag{5}$$

From Eq.(5), it's easy to conclude that $\frac{\partial \mathbf{z}_{i,:}}{\partial W}$ is:

$$\frac{\partial \mathbf{z}_{i,:}}{\partial W} = \frac{\partial \mathbf{z}_{i,:}}{\partial(\mathbf{w}_{1,:}, \mathbf{w}_{2,:}, ..., \mathbf{w}_{n,:})} = \left[O, O, ..., \frac{\partial \mathbf{z}_{i,:}}{\partial \mathbf{w}_{i,:}}, ..., O \right] = \left[O, O, ..., A^T, ..., O \right] \in \mathbb{R}^{(n \times m) \times l} \tag{6}$$

being $O \in \mathbb{R}^{l \times m}$ the zero matrix.

Finally, putting all together, the derivative of the matrix Z w.r.t. the matrix W, i.e. the jacobian, is:

$$\frac{\partial Z}{\partial W} = \left[\frac{\partial \mathbf{z}_{1,:}}{\partial W}, \frac{\partial \mathbf{z}_{2,:}}{\partial W}, ..., \frac{\partial \mathbf{z}_{n,:}}{\partial W} \right] = \left[\left[A^T, O, ..., O \right], \left[O, A^T, ..., O \right], ..., \left[O, ..., A^T \right] \right] \in \mathbb{R}^{(n \times m) \times (m \times l)} \tag{7}$$

It can be noticed that the tensor $\frac{\partial Z}{\partial W}$ is very sparse.
Since it will be useful in the following, we conclude this section by deriving a column $\mathbf{z}_{:,j} \in Z$ w.r.t. the entire matrix W. Again, $\mathbf{z}_{:,j}$ is:

$$\mathbf{z}_{:,j} = W \cdot \mathbf{a}_{:,j} = \begin{pmatrix} w_{1,1} & w_{1,2} & ... & w_{1,m} \\ w_{2,1} & w_{2,2} & ... & w_{2,m} \\ ... & ... & ... & ... \\ w_{n,1} & w_{n,2} & ... & w_{n,m} \end{pmatrix} \cdot \begin{pmatrix} a_{1,j} \\ a_{2,j} \\ ... \\ a_{m,j} \end{pmatrix} \tag{8}$$

The derivative is:

$$\frac{\partial \mathbf{z}_{:,j}}{\partial W} = \frac{\partial \mathbf{z}_{:,j}}{\partial(\mathbf{w}_{1,:}, \mathbf{w}_{2,:}, ..., \mathbf{w}_{n,:})} = \begin{bmatrix} \mathbf{a}_{:,j}^T \\ \mathbf{a}_{:,j}^T \\ ... \\ \mathbf{a}_{:,j}^T \end{bmatrix} \in \mathbb{R}^{n \times l} \tag{9}$$

2 Question 2

$$\underset{(4,1)}{\mathbf{a}_i^{(1)}} = \mathbf{x}_i \tag{10}$$

$$\underset{(10,1)}{\mathbf{z}_i^{(2)}} = \underset{(10,4)}{W^{(1)}} \underset{(4,1)}{\mathbf{a}_i^{(1)}} + \underset{(10,1)}{\mathbf{b}^{(1)}} \tag{11}$$

$$\underset{(10,1)}{\mathbf{a}_i^{(2)}} = \phi(\underset{(10,1)}{\mathbf{z}_i^{(2)}}) \tag{12}$$

$$\underset{(3,1)}{\mathbf{z}_i^{(3)}} = \underset{(3,10)}{W^{(2)}} \underset{(10,1)}{\mathbf{a}_i^{(2)}} + \underset{(3,1)}{\mathbf{b}^{(2)}} \tag{13}$$

$$\underset{(3,1)}{\mathbf{a}_i^{(3)}} = \psi(\underset{(3,1)}{\mathbf{z}_i^{(3)}}) \tag{14}$$

2.1 a)

First, we were required to demonstrate the equation for the gradient of the cross-entropy loss:

$$
\begin{aligned}
&\frac{\partial J}{\partial z_i^{(3)}}\left(\theta,\ \{x_i, y_i\}_{i=1}^N\right) = \frac{\partial}{\partial z_i^{(3)}}\ \frac{1}{N}\sum_{i=1}^{N} -\log\left(\frac{\exp\left(z_i^{(3)}\right)_{y_i}}{\sum_{j=1}^K \exp\left(z_i^{(3)}\right)_j}\right) \\
&= \frac{1}{N}\frac{\partial}{\partial z_i^{(3)}}\ \left[-\log\left(\frac{\exp\left(z_i^{(3)}\right)_{y_i}}{\sum_{j=1}^K \exp\left(z_i^{(3)}\right)_j}\right)\right] \\
&= \frac{1}{N}\frac{\partial}{\partial z_i^{(3)}}\left[\log\left(\sum_{j=1}^K \exp\left(z_i^{(3)}\right)_j\right) - \log\left(\exp\left(z_i^{(3)}\right)_{y_i}\right)\right] \\
&= \frac{1}{N}\left\{\frac{\partial}{\partial z_i^{(3)}}\left[\log\left(\sum_{j=1}^K \exp\left(z_i^{(3)}\right)_j\right)\right] - \frac{\partial}{\partial z_i^{(3)}}\left[\log\left(\exp\left(z_i^{(3)}\right)_{y_i}\right)\right]\right\} \\
&= \frac{1}{N}\left\{\frac{\partial}{\partial z_i^{(3)}}\left[\log\left(\sum_{j=1}^K \exp\left(z_i^{(3)}\right)_j\right)\right] - \frac{\partial}{\partial z_i^{(3)}}\left(z_i^{(3)}\right)_{y_i}\right\} \\
&\overset{\star}{=} \frac{1}{N}\left(\frac{\exp\left(z_i^{(3)}\right)}{\sum_{j=1}^K \exp\left(z_i^{(3)}\right)_j} - \Delta_i\right) = \frac{1}{N}\left(\psi\left(z_i^{(3)}\right) - \Delta_i\right)
\end{aligned}
\tag{15}
$$

Where we considered in the $(\star)$ equation that:

$$
\frac{\partial}{\partial z_i^{(3)}}\ \left(z_i^{(3)}\right)_{y_i} = \begin{pmatrix} \frac{\partial}{\left(\partial z_i^{(3)}\right)_1}\left(z_i^{(3)}\right)_{y_i} \\ \vdots \\ \frac{\partial}{\left(\partial z_i^{(3)}\right)_K}\left(z_i^{(3)}\right)_{y_i} \\ \vdots \\ \frac{\partial}{\left(\partial z_i^{(3)}\right)_{y_1}}\left(z_i^{(3)}\right)_{y_i} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \left(\Delta_i\right)_j
\tag{16}
$$

Where

$$
\left(\Delta_i\right)_j = \begin{cases} 1, & \text{if } j = y_i \\ 0, & \text{otherwise} \end{cases}
\tag{17}
$$

And that:

$$
\begin{aligned}
&\frac{\partial}{\partial z_i^{(3)}} \quad \left[\log \sum_{j=1}^{K} \exp\left(z_i^{(3)}\right)_j\right] = \\
&= \frac{1}{\sum_{j=1}^{K} \exp\left(z_i^{(3)}\right)_j} \cdot \ \frac{\partial}{\partial z_i^{(3)}} \sum_{j=1}^{K} \exp\left(z_i^{(3)}\right)_j \\
&= \frac{1}{\sum_{j=1}^{K} \exp\left(z_i^{(3)}\right)_j} \cdot \left[\begin{pmatrix} \exp\left(z_i^{3}\right)_1 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \cdots + \begin{pmatrix} 0 \\ \vdots \\ \exp\left(z_i^{3}\right)_j \\ \vdots \\ 0 \end{pmatrix} + \cdots + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ \exp\left(z_i^{3}\right)_K \end{pmatrix}\right] \\
&= \frac{1}{\sum_{j=1}^{K} \exp\left(z_i^{(3)}\right)_j} \cdot \exp\left(z_i^{(3)}\right)
\end{aligned}
\tag{18}
$$

2.2 b)

Then we computed the partial derivative of the loss function with respect to $W^{(2)}$.

$$
\begin{aligned}
&\frac{\partial J}{\partial W^{(2)}}\left(\theta, \{x_i, y_i\}_{i=1}^{N}\right) = \frac{\partial}{\partial W^{(2)}} \left\{ \frac{1}{N} \sum_{i=1}^{N} - \log \left[\frac{\exp\left(z_i^{(3)}\right)_{y_i}}{\sum_{j=1}^{K} \left(\exp\left(z_i^{(3)}\right)\right)_j} \right] \right\} \\
&\overset{\star}{=} \frac{\partial z^{(3)}}{\partial W^{(2)}} \cdot \frac{\partial}{\partial z^{(3)}} \left\{ \frac{1}{N} \sum_{i=1}^{N} - \log \left[\frac{\exp\left(z_i^{(3)}\right)_{y_i}}{\sum_{j=1}^{K} \left(\exp\left(z_i^{(3)}\right)\right)_j} \right] \right\} \\
&= \sum_{i=1}^{N} \frac{\partial z_i^{(3)}}{\partial W^{(2)}} \cdot \frac{\partial}{\partial z_i^{(3)}} \left\{ -\frac{1}{N} \log \left[\frac{\exp\left(z_i^{(3)}\right)_{y_i}}{\sum_{j=1}^{K} \left(\exp\left(z_i^{(3)}\right)\right)_j} \right] \right\} \\
&\overset{\star\star}{=} \sum_{i=1}^{N} \frac{\partial z_i^{(3)}}{\partial W^{(2)}} \cdot \frac{\partial J}{\partial z_i^{(3)}} = \sum_{i=1}^{N} \frac{\partial}{\partial W^{(2)}} \left(W^{(2)} a_i^{(2)} + b^{(2)}\right) \cdot \frac{\partial J}{\partial z_i^{(3)}} = \sum_{i=1}^{N} \frac{1}{N} \left(\psi\left(z_i^{(3)}\right) - \Delta_i\right) \left(a_i^{(2)}\right)^T
\end{aligned}
\tag{19}
$$

having ($\star$) by the chain rule and ($\star\star$) by the Equation of point *a)*.

The derivative $\frac{\partial \widetilde{J}}{\partial W^{(2)}}$ is trivial, since:

$$\begin{aligned}\frac{\partial \widetilde{J}}{\partial W^{(2)}}\left(\theta, \{x_i, y_i\}_{i=1}^{N}\right) &= \frac{\partial J}{\partial W^{(2)}} + \frac{\partial R}{\partial W^{(2)}} = \frac{\partial J}{\partial W^{(2)}} + \frac{\partial}{\partial W^{(2)}}\left[\lambda\Big(\left\|W^{(1)}\right\|_2^2 + \left\|W^{(2)}\right\|_2^2\Big)\right] \\ &= \frac{\partial J}{\partial W^{(2)}} + \frac{\partial}{\partial W^{(2)}}\left[\lambda\left(\sum_{i=1}^{3}\sum_{j=1}^{10}\left(w_{ij}^{(2)}\right)^2\right)\right] = \sum_{i=1}^{N}\frac{1}{N}\left(\psi\left(z_i^{(3)}\right) - \Delta_i\right)\left(a_i^{(2)}\right)^T + 2\lambda W^{(2)}\end{aligned} \tag{20}$$

2.3 c)

Here we consider the partial derivative of the Loss with respect to $W^{(1)}$:

$$\begin{aligned}\frac{\partial \widetilde{J}}{\partial W^{(1)}}\left(\theta, \{x_i, y_i\}_{i=1}^{N}\right) &= \frac{\partial J}{\partial W^{(1)}} + \frac{\partial R}{\partial W^{(1)}} = \frac{\partial J}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial W^{(1)}} + \frac{\partial R}{\partial W^{(1)}} \\ &= \frac{\partial J}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a^{(2)}} \cdot \frac{\partial a^{(2)}}{\partial W^{(1)}} + \frac{\partial R}{\partial W^{(1)}} = \frac{\partial J}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a^{(2)}} \cdot \frac{\partial a^{(2)}}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial W^{(1)}} + \frac{\partial R}{\partial W^{(1)}} \\ &= \sum_{i=1}^{N}\left[\frac{1}{N}\left(\psi\left(z_i^{(3)}\right) - \Delta_i\right)\right] \cdot \frac{\partial z^{(3)}}{\partial a^{(2)}} \cdot \frac{\partial a^{(2)}}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial W^{(1)}} + \frac{\partial R}{\partial W^{(1)}} \\ &= \sum_{i=1}^{N}\left(W^{(2)}\right)^T \cdot \left[\frac{1}{N}\left(\psi\left(z_i^{(3)}\right) - \Delta_i\right)\right] \cdot \frac{\partial a^{(2)}}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial W^{(1)}} + \frac{\partial R}{\partial W^{(1)}} \\ &= \sum_{i=1}^{N}\left\{\left[\left(W^{(2)}\right)^T\left(\frac{1}{N}\left(\psi\left(z_i^{(3)}\right) - \Delta_i\right)\right)\right] \odot \mathbb{I}\left\{z_i^{(3)} > 0\right\}\right\} \cdot \frac{\partial z^{(2)}}{\partial W^{(1)}} + \frac{\partial R}{\partial W^{(1)}} \\ &= \sum_{i=1}^{N}\left\{\left[\left(W^{(2)}\right)^T \cdot \left(\frac{1}{N}\left(\psi\left(z_i^{(3)}\right) - \Delta_i\right)\right)\right] \odot \mathbb{I}\left\{z_i^{(3)} > 0\right\}\right\}\left(a_i^{(1)}\right)^T + \frac{\partial R}{\partial W^{(1)}} \\ &= \sum_{i=1}^{N}\left\{\left[\left(W^{(2)}\right)^T \cdot \left(\frac{1}{N}\left(\psi\left(z_i^{(3)}\right) - \Delta_i\right)\right)\right] \odot \mathbb{I}\left\{z_i^{(3)} > 0\right\}\right\}\left(a_i^{(1)}\right)^T + 2\lambda W^{(1)}\end{aligned} \tag{21}$$

Here we consider the partial derivative of the Loss with respect to $b^{(1)}$:

$$\begin{aligned}\frac{\partial \widetilde{J}}{\partial b^{(1)}}\left(\theta, \{x_i, y_i\}_{i=1}^{N}\right) &= \frac{\partial J}{\partial b^{(1)}} + \frac{\partial R}{\partial b^{(1)}} = \frac{\partial J}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial b^{(1)}} + \frac{\partial R}{\partial b^{(1)}} \\ &= \frac{\partial J}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a^{(2)}} \cdot \frac{\partial a^{(2)}}{\partial b^{(1)}} + \frac{\partial R}{\partial b^{(1)}} = \frac{\partial J}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a^{(2)}} \cdot \frac{\partial a^{(2)}}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial b^{(1)}} + \frac{\partial R}{\partial b^{(1)}} \\ &= \sum_{i=1}^{N}\left(W^{(2)}\right)^T\left[\left(\frac{1}{N}\left(\psi\left(z_i^{(3)}\right) - \Delta_i\right)\right)\right] \odot \mathbb{I}\left\{z_i^{(3)} > 0\right\} + \frac{\partial R}{\partial b^{(1)}} \\ &= \sum_{i=1}^{N}\left(W^{(2)}\right)^T\left[\left(\frac{1}{N}\left(\psi\left(z_i^{(3)}\right) - \Delta_i\right)\right)\right] \odot \mathbb{I}\left\{z_i^{(3)} > 0\right\}\end{aligned} \tag{22}$$

Here we consider the partial derivative of the Loss with respect to $b^{(2)}$:

$$\begin{aligned}\frac{\partial \widetilde{J}}{\partial b^{(2)}}\left(\theta, \{x_i, y_i\}_{i=1}^{N}\right) &= \frac{\partial J}{\partial b^{(2)}} + \frac{\partial R}{\partial b^{(2)}} = \frac{\partial J}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial b^{(2)}} + \frac{\partial R}{\partial b^{(2)}} \\ &= \sum_{i=1}^{N} \frac{1}{N}\left(\psi\left(z_i^{(3)}\right) - \Delta_i\right) + \frac{\partial R}{\partial b^{(2)}} = \sum_{i=1}^{N} \frac{1}{N}\left(\psi\left(z_i^{(3)}\right) - \Delta_i\right)\end{aligned} \tag{23}$$

where:

- $\odot$ indicates the element-wise product (Hadamard product)
- $\mathbb{I}\{u > 0\}$ is the derivative of the ReLU function:

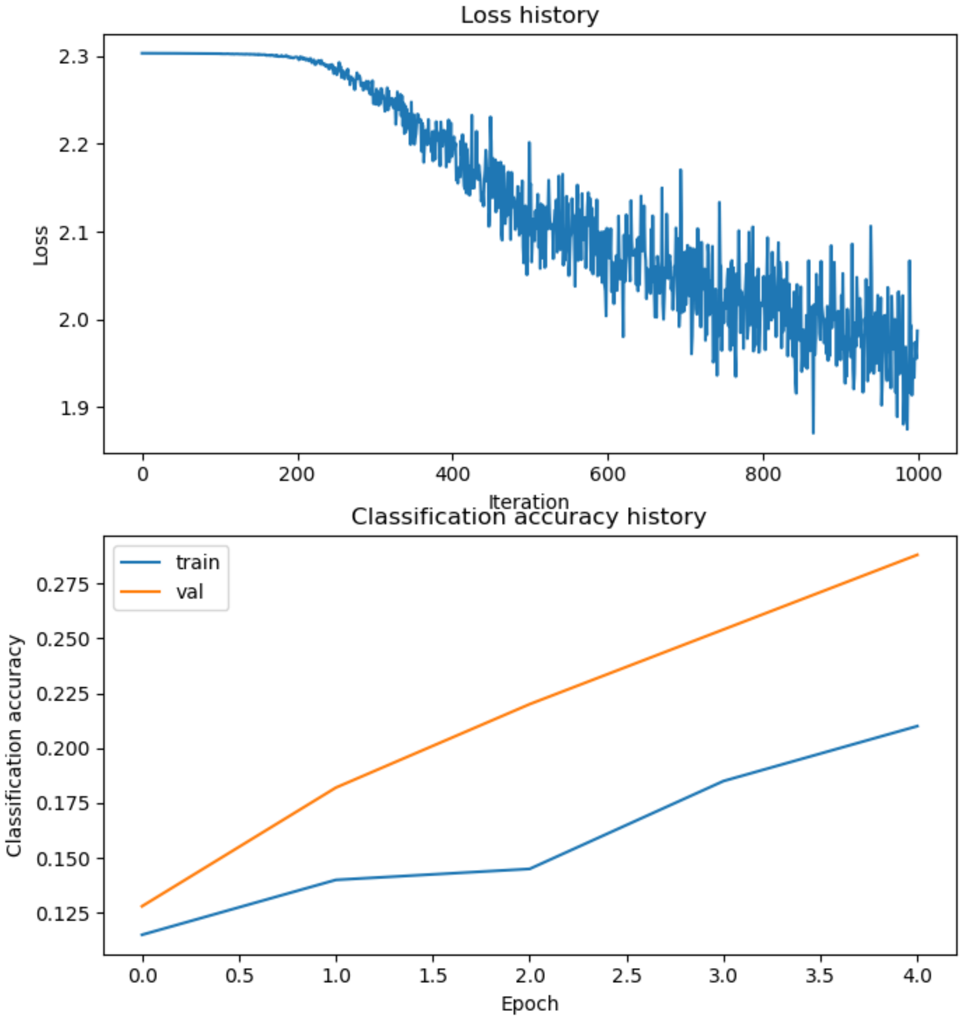
$$\mathbb{I}\{u > 0\} = \begin{cases} 1, & \text{if } \ u > 0 \\ \text{indefinite}, & \text{if } \ u = 0 \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

- $\frac{\partial z^{(k+1)}}{\partial b^{(k)}} = \mathbb{1}_{b^{(k)}}$ a vector with the same dimensionality of $b^{(k)}$ filled with all ones.

3 Question 3

b) As expected, the model defined in the previous questions shows a very poor performance on the CIFAR-10 data set with the hyper-parameters' configuration provided by default. Let us recall that the default hyper-parameters' configuration for the model is:

- `hidden_size = 50`
- `learning_rate = 1e-4`
- `num_iters = 1000`
- `reg = 0.25`
- `learning_rate_decay = 0.95`



The plots above show the trend of the loss on the training set and accuracy curves during the training and validation's steps. The accuracy score on the validation set is 29%.

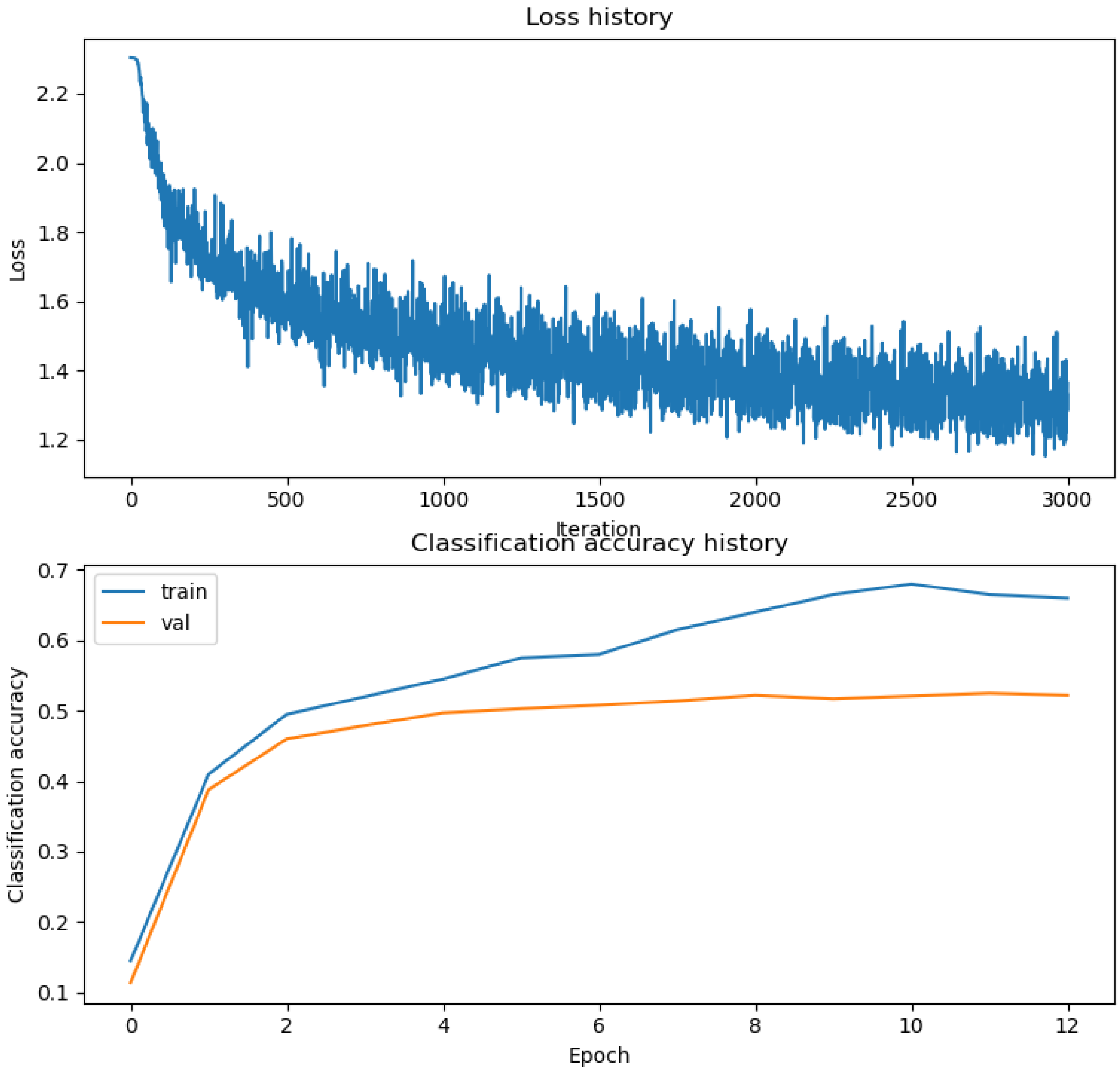
In order to improve the performance of the model, a hyper-parameters' tuning is required. We decided to perform first a *random search* algorithm, starting from the values above, in order to find candidate ranges in which our best parameters may lie.
We identified the following ranges:

```
param_grid = {
    'hidden_size': [60, 80, 100, 120],
    'num_iters': [1000, 2000, 3000],
    'batch_size': [100, 200],
    'learning_rate': [0.01, 0.001, 0.0001],
    'learning_rate_decay': [0.95],
    'reg': [0.10, 0.15, 0.25, 0.35]
}
```

Starting from such intervals, we performed a *grid search* algorithm, i.e. an exhaustive search on those sets of values.
Throughout this approach, we obtained the following best parameters:

```
'hidden_size': 100
'num_iters': 3000
'batch_size': 200
'learning_rate': 0.001
'learning_rate_decay': 0.95
'reg': 0.15
```

Using these best values, we managed to achieve an accuracy value in $[0.52, 0.54]$, with obvious fluctuations due to the random initialization of the parameters. As the depicted in the plot, the model is not able to improve significantly on the validation set after the 8^{th} epoch, while the accuracy on the training set keeps increasing. Actually, the divergence of the two curves after the 4^{th} epoch can be a symptom of overfitting arising.



4 Question 4

c) We decided to experiment settings with different numbers of layers and different numbers of neurons per layer. Specifically, we considered from 2 up to 5 layers, trying also some other techniques like dropout, etc. We kept the default hyper-parameters' configuration:

```
'num_epochs':10
'batch_size':200
'learning_rate':1e-3
'learning_rate_decay':0.95
'reg':0.001
```

In the following, we report the loss value and the validation accuracy for each setting coming from a number of tried configurations:

- **5 Layers** with [50, 50, 50, 50] as units of each hidden layer with loss approximately 2.30 and validation accuracy approximately 7.8%
- **4 Layers** with [75, 75, 75] as units of each hidden layer with loss approximately 1.43 and validation accuracy approximately 51.0%
- **3 Layers** with [75, 150] as units of each hidden layer with loss approximately 1.22 and validation accuracy approximately 54.8%
- **2 Layers** with [50] as units of each hidden layer with loss approximately 1.27 and validation accuracy approximately 51.9%

In order to interpret the results, let's stress out that the regularization strength is quite low (10^{-3}); thus, if the number of parameters is low enough, their absolute values can increase without being penalized that much; on the contrary, the greater the number of parameters, the less their values must be to avoid penalization, the more we somehow enforce sparsity. In other words, the trade off to be chased regards the balance between the capacity of the model, i.e. few parameters fail in addressing complexity or lead to overfitting, and the penalization of the solution.
The experiments indeed confirmed these informal considerations: with 2 layers the network is allowed to learn a more complex function, still maintaining a relatively low number of neurons; with 3 layers, again, complexity is "spread" across layers and neurons, giving us the best model; in the case of 4 layers chasing good performance has been harder, since only keeping an architecture somehow similar to the 3-layers case for the first layer has given satisfying results; finally, the 5-layers network wasn't able to address the problem properly.
Furthermore, by adding some kind of regularization such as dropout, the network does not improve in terms of accuracy, in fact, overfitting is never achieved.

Ultimately, then, the best model seemed to be the 3-layers network with an increasing number of neurons going forward through the layers. Its final test accuracy was approximately 56.0%.