Homework 2 report

Advanced Machine Learning course, teached by Prof. Fabio Galasso
AY2020/21

Riccardo Ceccaroni	Simone Ercolino
1884368	1587229
Romeo Lanzino	Dario Ruggeri
1753403	1741637

Sapienza Università di Roma

November 2020

1 Question 2

1.1 Point a

$$
\begin{aligned}
J(\theta) &= \frac{1}{N}\sum_{i=1}^{N} -\log\left(a^{(3)}\right) \\
&= \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{1}{a^{(3)}}\right)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial J}{\partial z_i^{(3)}} &= \frac{1}{N}\sum_{i=1}^{N} \frac{\partial}{\partial z_i^{(3)}} \log\left(\frac{1}{a^{(3)}}\right) \\
&= \frac{1}{N}\sum_{i=1}^{N} a^{(3)}\left[\frac{-\frac{\partial}{\partial z_i^{(3)}} a^{(3)}}{(a^{(3)^2})}\right] \\
&= \frac{1}{N}\sum_{i=1}^{N} \frac{1}{a^{(3)}}\left[-\frac{\partial}{\partial z_i^{(3)}} a^{(3)}\right] \\
&= \sum_{i=1}^{N} -\frac{1}{N}\frac{1}{a^{(3)}}\psi'(z_i^{(3)})\mathbf{1}_i \\
&= \sum_{i=1}^{N} -\frac{1}{N}\frac{1}{a^{(3)}} a^{(3)}(\delta_{i,y_i} - a^{(3)})\mathbf{1}_i \\
&= \frac{1}{N}(a^{(3)} - \delta_{i,y_i}) \\
&= \frac{1}{N}(\psi(z_i^{(3)} - \delta_{i,y_i})
\end{aligned}
$$

where the derivative of the *softmax* function is given by:

$$
\begin{aligned}
\psi'(u_i) &= \frac{\partial}{\partial u_i} \frac{e^{u_i}}{\sum_{j=1} e^{u_j}} \\
&= \frac{e^{u_i} \sum_{j=1} (e^{u_j} - e^{u_i})}{\left[\sum_{j=1} e^{u_j}\right]^2} \\
&= \frac{e^{u_i}}{\sum_{j=1} e^{u_j}} \frac{\sum_{j=1} e^{u_j} - e^{u_i}}{\sum_{j=1} e^{u_j}} \\
&= \frac{e^{u_i}}{\sum_{j=1} e^{u_j}} \left(\frac{\sum_{j=1} e^{u_j}}{\sum_{j=1} e^{u_j}} - \frac{e^{u_i}}{\sum_{j=1} e^{u_j}} \right) \\
&= \frac{e^{u_i}}{\sum_{j=1} e^{u_j}} \left(\delta_{i,j} - \frac{e^{u_i}}{\sum_{j=1} e^{u_j}} \right) \\
&= \psi(u_i)(\delta_{i,j} - \psi(u_i))
\end{aligned}
$$

1.1.1 Point b

$$
\begin{aligned}
a^{(1)} &= x \\
z^{(2)} &= W^{(1)} \cdot a^{(1)} + b^{(1)} \\
a^{(2)} &= \phi(z^{(2)}) \\
z^{(3)} &= W^{(2)} \cdot a^{(2)} + b^{(2)} \\
a^{(3)} &= \psi(z^{(3)})
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial J}{\partial W^{(2)}_{i,j}} &= \sum_{i=1}^{N} \frac{\partial J}{\partial z^{(3)}_i} \frac{\partial z^{(3)}_i}{\partial W^{(2)}_{i,j}} \\
&= \sum_{i=1}^{N} \frac{\partial J}{\partial z^{(3)}_i} a^{(2)}_{1,k} \\
&= \sum_{i=1}^{N} \frac{1}{N} \big(\psi(z^{(3)}_i) - \delta_{i,y_i}\big) a^{(2)}_{1,k}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial J}{\partial W^{(2)}} &= \left[\frac{1}{N}\big(\psi(z^{(3)}) - \delta\big)^T \times a^{(2)}\right]^T \\
\frac{\partial \tilde{J}}{\partial W^{(2)}} &= \frac{\partial J}{\partial W^{(2)}} + 2\lambda W^{(2)}
\end{aligned}
$$

1.1.2 Point c

$$
\begin{aligned}
\frac{\partial J}{\partial W^{(1)}_{k,j}} &= \sum_{i=1}^{N} \frac{\partial J}{\partial z^{(3)}_i} \frac{\partial z^{(3)}_i}{\partial W^{(1)}_{k,j}} \\
&= \sum_{i=1}^{N} \frac{\partial J}{\partial z^{(3)}_i} W^{(2)}_{j,i} \frac{\partial a^{(2)}_j}{\partial W^{(1)}_{k,j}} \\
&= \sum_{i=1}^{N} \frac{\partial J}{\partial z^{(3)}_i} W^{(2)}_{j,i} \phi'(z^{(2)}_j) \frac{\partial z^{(2)}_j}{\partial W^{(1)}_{k,j}} \\
&= \sum_{i=1}^{N} \frac{\partial J}{\partial z^{(3)}_i} W^{(2)}_{j,i} \phi'(\sum_{k=1}^{m} W^{(1)}_{k,j} a^{(1)}_{1,k} + b^{1}_{1,k})\, a^{(1)}_{1,k} \\
&= \sum_{i=1}^{N} \frac{1}{N} \big(\psi(z^{(3)}_i) - \delta_{i,y_i}\big) W^{(2)}_{j,i} \phi'(\sum_{k=1}^{m} W^{(1)}_{k,j} a^{(1)}_{1,k} + b^{1}_{1,k})\, a^{(1)}_{1,k}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial J}{\partial W^{(1)}} &= \left[\Big[\Big(\frac{1}{N}\big(\psi(z^{(3)}) - \delta\big)^T \times W^{(2)}\Big)\phi'(z^{(2)})\Big]^T \times a^{(1)}\right]^T \\
\frac{\partial \tilde{J}}{\partial W^{(1)}} &= \frac{\partial J}{\partial W^{(1)}} + 2\lambda W^{(1)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial J}{\partial b_{1,j}^{(1)}} &= \sum_{i=1}^{N} \frac{\partial J}{\partial z_i^{(3)}} \frac{\partial z_i^{(3)}}{\partial b_{1,j}^{(1)}} \\
&= \sum_{i=1}^{N} \frac{\partial J}{\partial z_i^{(3)}} W_{j,i}^{(2)} \frac{\partial a_j^{(2)}}{\partial b_{1,j}^{(1)}} \\
&= \sum_{i=1}^{N} \frac{\partial J}{\partial z_i^{(3)}} W_{j,i}^{(2)} \phi'(z_j^{(2)}) \frac{\partial z_j^{(2)}}{\partial b_{1,j}^{(1)}} \\
&= \sum_{i=1}^{N} \frac{\partial J}{\partial z_i^{(3)}} W_{j,i}^{(2)} \phi'(\sum_{k=1}^{m} W_{k,j}^{(1)} a_{1,k}^{(1)} + b_{1,k}^{1})\, \mathbf{1}_j \\
&= \sum_{i=1}^{N} \frac{1}{N} \big(\psi(z_i^{(3)}) - \delta_{i,y_i}\big) W_{j,i}^{(2)} \phi'(\sum_{k=1}^{m} W_{k,j}^{(1)} a_{1,k}^{(1)} + b_{1,k}^{1})\, \mathbf{1}_j
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial J}{\partial b^{(1)}} &= \Big[\Big(\frac{1}{N}\big(\psi(z^{(3)}) - \delta\big)^T \times W^{(2)}\Big)\phi'(z^{(2)})\Big]^T \\
\frac{\partial \tilde{J}}{\partial b^{(1)}} &= \frac{\partial J}{\partial b^{(1)}}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial J}{\partial b_{1,i}^{(2)}} &= \sum_{i=1}^{N} \frac{\partial J}{\partial z_i^{(3)}} \frac{\partial z_i^{(3)}}{\partial b_{1,i}^{(2)}} \\
&= \sum_{i=1}^{N} \frac{1}{N} \big(\psi(z_i^{(3)}) - \delta_{i,y_i}\big) \mathbf{1}_i
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial J}{\partial b^{(2)}} &= \sum_{i=1}^{N} \frac{1}{N} \big(\psi(z_i^{(3)}) - \delta_{i,y_i}\big) \\
\frac{\partial \tilde{J}}{\partial b^{(2)}} &= \frac{\partial J}{\partial b^{(2)}}
\end{aligned}
$$

2 Question 3

2.1 Point b

By doing a random search repeated 100 times on the hyperparameters with discrete uniform distributions over the following supports:

Parameter	Range tested	Optimal value
hidden size	$[90, 150]$	135
learning rate	$[3.5e-3, 5e-3]$	0.004786
learning rate decay	$\{0.98, 0.99\}$	0.99
regularization strength	$[1e-5, 2e-3]$	0.001847
epochs	$[700, 1300]$	1100
batch size	$[15000, 25000]$	22000

The model with optimal values achieves 0.555 of accuracy on the train set, 0.491 on the validation set and 0.527 on the test set.

Furthermore, *PCA* was fit on the train and applied on the dataset, in order to reduce the dimensions to 1200, and so speed up the computations and get rid of some noise; on the other hand, for speedup convergence was used a *momentum gradient descent* with a strength of the previous iteration gradient of 20%.

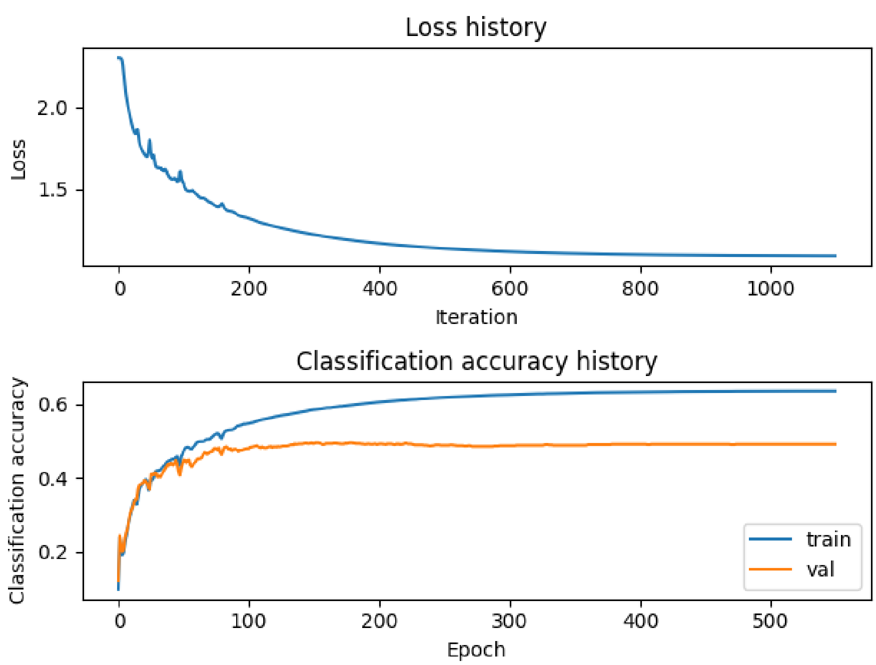


Figure 1: Loss and accuracy of the Numpy model

3 Question 4

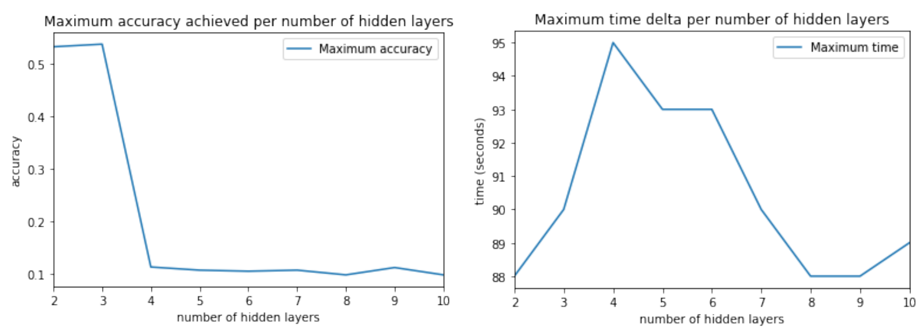
3.1 Point c

We've tried several models with different numbers of hidden layers $|l| \in [2, 10]$, and different numbers of neurons in each layer (using a discrete uniform $[20, 100]$); among all the possible combinations we've took just 3 for each number of layers, in a random search fashion.

We'll use a_m to denote the maximum accuracy obtained during the evaluation of a particular model m.

Besides the different number of hidden layers and neurons there are two main variations with respect to the model presented, corresponding to two new layers: a batch normalization layer at the beginning of the chain which improves a_m from 1 to 3 percentage points and dropout, which drops a_m by 2 to 4 points and as such it was discarded.

We can empirically see that a_m drastically drops to even 9% when l rise above 3.



(a) Accuracy

(b) Time

Figure 2: Charts for the PyTorch model