

AML Assignment 2 - report

April 9, 2020

Fabio Montello (1834411), Francesco Russo (1449025), Michele Cernigliaro (1869097)

In this report we proceed to answer as requested the questions 2 (a, b, c), 3 (a,b) and 4 (a,b,c). For each point we are going to write a brief description using figures and formulas whenever needed.

```
[9]: import matplotlib.pyplot as plt
import matplotlib.image as mpimg
```

1 Question 2a

Verify that the loss function defined in Eq. (5) has the gradient w.r.t. $z^{(3)}$ as below:

$$\frac{\partial J}{\partial z^{(3)}}(\{x_i, y_i\}_{i=1}^N) = \frac{1}{N}(\psi(z^{(3)}) - \Delta)$$

We have the loss function

$$J = \frac{1}{N} \left[\sum_{i=1}^N -\log \psi(z_{yi}) \right] = \frac{1}{N} \left[\sum_{i=1}^N -\log \frac{e^{z_{yi}}}{\sum_j e^{z_j}} \right] = \frac{1}{N} \left[\sum_{i=1, j=k_i}^N -\log \frac{e^{z_{i,j}}}{\sum_j e^{z_j}} \right]$$

Where we use the notation $z = z^{(3)}$ and the softmax activation function is

$$\psi(z_{yi}) = \frac{e^{z_{yi}}}{\sum_j e^{z_j}} = \frac{e^{z_{ik_i}}}{\sum_j e^{z_j}}$$

The matrix ∇_J of the derivatives of J with respect to z_{ij} for every $i = 1 \dots N$ and for every $j = 1 \dots K$ has the general element

$$\begin{aligned} \nabla_{J_{i',j'}} &= \frac{\partial J}{\partial z_{i',j'}} = \frac{\partial}{\partial z_{i',j'}} \frac{1}{N} \left[\sum_{i=1}^N -\log \psi(z_{yi}) \right] = -\frac{\partial}{\partial z_{i',j'}} \frac{1}{N} [\log \psi(z_{yi'})] = -\frac{1}{N} \frac{1}{\psi(z_{yi'})} \frac{\partial}{\partial z_{i',j'}} \psi(z_{yi'}) = \\ &= -\frac{1}{N} \frac{1}{\psi(z_{yi'})} \frac{e^{z_{i'j'}} \cdot \delta(j', k_{i'}) \sum_j e^{z_{ij}} - e^{z_{i'k'_i}} e^{z_{i'j'}}}{(\sum_j e^{z_{ij}})^2} = \\ &= -\frac{1}{N} \frac{\sum_j e^{z_{ij}}}{e^{z_{i'k'_i}}} \frac{e^{z_{i'j'}} \cdot \delta(j', k_{i'}) \sum_j e^{z_{ij}} - e^{z_{i'k'_i}} e^{z_{i'j'}}}{(\sum_j e^{z_{ij}})^2} = \end{aligned}$$

$$\frac{1}{N} \left[\frac{e^{z_{i'j'}}}{\sum_j e^{z_{ij}}} - \delta(j', k'_i) \right]$$

where $\delta(j', k'_i)$ is the Kronecker Delta function

In matrix form this is exactly:

$$\nabla_J = \frac{1}{N} [\psi(z) - \Delta_{j,ki}]$$

2 Question 2b

Verify that the partial derivative of the loss w.r.t. $W^{(2)}$ is:

$$\frac{\partial J}{\partial W^{(2)}}(\{x_i, y_i\}_{i=1}^N) = \frac{\partial J}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial W^{(2)}} \quad (1)$$

$$= \frac{1}{N} (\psi(z^{(3)}) - \Delta) a^{(2)} \quad (2)$$

As seen before, we compute the chain rule as follow:

$$\frac{\partial J}{\partial W^{(2)}}(\{x_i, y_i\}_{i=1}^N) = \frac{\partial J}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial W^{(2)}}$$

We already computed:

$$\frac{1}{N} (\psi(z^{(3)}) - \Delta)$$

We know that

$$\frac{\partial z^{(3)}}{\partial W^{(2)}} = a^{(2)}$$

And so we will have that:

$$\frac{\partial J}{\partial W^{(2)}}(\{x_i, y_i\}_{i=1}^N) = \frac{1}{N} (\psi(z^{(3)}) - \Delta) a^{(2)}$$

Similarly, verify that the regularized loss in Eq. (6) has the derivatives

$$\frac{\partial \tilde{J}}{\partial W^{(2)}} = \frac{1}{N} (\psi(z^{(3)}) - \Delta) a^{(2)} + 2\lambda W^{(2)}$$

We can now find the partial derivative of the loss w.r.t the regularization term, which is:

$$\frac{\partial \left[\lambda \left(\|W^{(1)}\|_2^2 + \|W^{(2)}\|_2^2 \right) \right]}{\partial W^{(2)}} = 2\lambda W^{(2)}$$

So we obtain:

$$\frac{dJ}{dW^{(2)}} = \frac{dJ}{dz^{(3)}} \cdot \frac{dz^{(3)}}{dW^{(2)}} = \frac{1}{N} \left[\psi(z^{(3)}) - \Delta \right] \cdot a^{(2)} + 2\lambda W^{(2)}$$

3 Question 2c

We can repeatedly apply chain rule as discussed above to obtain the derivatives of the loss with respect to all the parameters of the model $\theta = (W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)})$. Derive the expressions for the derivatives of the regularized loss in Eq. (6) w.r.t. $W^{(1)}, b^{(1)}, b^{(2)}$ now.

We start deriving the loss J w.r.t $W^{(1)}$, by applying the chain rule. We'll had then the regularization term. Applying the chain rule we get for $W^{(1)}$ the following:

$$\frac{\partial J}{\partial W^{(1)}} = \frac{\partial J}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial W^{(1)}} \quad (3)$$

$$(4)$$

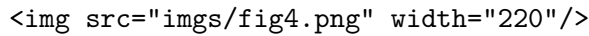
For $\frac{\partial J}{\partial z^{(2)}}$ we apply iteratively the chain rule obtaining:

$$\frac{\partial J}{\partial z^{(2)}} = \frac{\partial J}{\partial a^{(2)}} \odot \frac{\partial a^{(2)}}{\partial z^{(2)}} \quad (5)$$

$$= \left(\frac{\partial J}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a^{(2)}} \right) \odot \frac{\partial a^{(2)}}{\partial z^{(2)}} \quad (6)$$

$$(7)$$

Where \odot indicates the element-wise product (Hadamard product) and $\frac{\partial a^{(2)}}{\partial z^{(2)}} \in \mathbb{R}^{10 \times 5}$ is the the derivative of the ReLu activation w.r.t. $z^{(2)}$, which is also the heaviside step function:



We do already know $\frac{\partial J}{\partial z^{(3)}}$, now the calculation of the other partial derivatives is straightforward:

$$\frac{\partial z^{(3)}}{\partial a^{(2)}} = \frac{\partial (W^{(2)} a^{(2)} + b^{(2)})}{\partial a^{(2)}} = W^{(2)} \quad (8)$$

$$\frac{\partial z^{(2)}}{\partial W^{(1)}} = \frac{\partial (W^{(1)} a^{(1)} + b^{(1)})}{\partial W^{(1)}} = a^{(1)} \quad (9)$$

Now we finally obtain the loss derivatives w.r.t. $W^{(1)}$ as it follows:

$$\frac{\partial J}{\partial W^{(1)}} = \frac{\partial J}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial W^{(1)}} \quad (10)$$

$$= \left[\left(\frac{\partial J}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a^{(2)}} \right) \odot \frac{\partial a^{(2)}}{\partial z^{(2)}} \right] \cdot \frac{\partial z^{(2)}}{\partial W^{(1)}} \quad (11)$$

$$= \frac{1}{N} \cdot \left\{ \left[W^{(2)T} \cdot (\psi(z^{(3)}) - \Delta) \right] \odot \frac{\partial a^{(2)}}{\partial z^{(2)}} \right\} \cdot a^{(1)} \quad (12)$$

We can now find the gradient of $W^{(1)}$ w.r.t the regularization term, which is:

$$\frac{\partial \left[\lambda \left(\|W^{(1)}\|_2^2 + \|W^{(2)}\|_2^2 \right) \right]}{\partial W^{(1)}} = 2\lambda W^{(1)}$$

Hence we end up having:

$$\frac{\partial \tilde{J}}{\partial W^{(1)}} = \frac{1}{N} \cdot \left\{ \left[W^{(2)T} \cdot (\psi(z^{(3)}) - \Delta) \right] \odot \frac{\partial a^{(2)}}{\partial z^{(2)}} \right\} \cdot a^{(1)} + 2\lambda W^{(1)}$$

Which can be also implemented in vectorized form using numpy.

We now derive the loss J w.r.t $b^{(1)}$. Applying the chain rule we obtain similarly:

$$\frac{\partial J}{\partial b^{(1)}} = \frac{\partial J}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial b^{(1)}} \quad (13)$$

$$= \left[\left(\frac{\partial J}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a^{(2)}} \right) \odot \frac{\partial a^{(2)}}{\partial z^{(2)}} \right] \cdot \frac{\partial z^{(2)}}{\partial b^{(1)}} \quad (14)$$

$$= \left[\left(\frac{\partial J}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a^{(2)}} \right) \odot \frac{\partial a^{(2)}}{\partial z^{(2)}} \right] \cdot \mathbb{1} \quad (15)$$

$$(16)$$

For the loss J w.r.t. $b^{(2)}$ we have the following:

$$\frac{dJ}{db^{(2)}} = \frac{dJ}{dz^{(3)}} \cdot \frac{dz^{(3)}}{db^{(2)}} = \frac{1}{N} \left[\psi(z^{(3)}) - \Delta \right] \cdot \mathbb{1}$$

4 Question 3a

Implement the stochastic gradient descent algorithm in `two_layernet.py` and run the training on the toy data. Your model should be able to obtain loss = 0.02 on the training set and the training curve should look similar to the one shown in Fig. 2.

Copy-pasta del codice:

And the loss curve we obtained is the following:

5 Question 3b

[...] Your task is to debug the model training and come up with better hyper-parameters to improve the performance on the validation set. Visualize the training and validation performance curves to help with this analysis. [...] Once you have tuned your hyper-parameters, and get validation accuracy greater than 48% run your best model on the test set once and report the performance. (report, 5 points)

[]:

6 Question 4a

Complete the code to implement a multi-layer perceptron network in the class `MultiLayerPerceptron` in `ex2.pytorch.py`. This includes instantiating the required layers from `torch.nn` and writing the code for forward pass. Initially you should write the code for the same two-layer network we have seen before.

7 Question 4c

Now that you can train the two layer network to achieve reasonable performance, try increasing the network depth to see if you can improve the performance. Experiment with networks of at least 2, 3, 4, and 5 layers, of your chosen configuration. Report the training and validation accuracies for these models and discuss your observations. Run the evaluation on the test set with your best model and report the test accuracy.

[]: