Assignment 9

Computer Vision
Zeno Sambugaro 785367

November 13, 2019

Exercise 1

a. If we consider only one sample the formula of the oss will become:

$$E = -t \log y^{(2)}$$

b. Starting from:

$$\frac{\delta E}{\delta z^{(2)}} = \frac{\delta E}{\delta y^{(2)}} \frac{\delta y^{(2)}}{\delta z^{(2)}} \tag{1}$$

The cross entropy loss is defined as follow:

$$E = \frac{1}{m} \sum_{j=1}^{m} -t_j \cdot log(y_j)$$

Thus:

$$\frac{\delta E}{\delta z_i} = -\sum_{j=1}^{m} \frac{\delta t_j \cdot log(y_j)}{\delta z_i} = -\sum_{j=1}^{m} \frac{t_j \cdot \delta log(y_j)}{\delta z_i}$$

substituting (1):

$$-\sum_{j=1}^{m} t_j \frac{1}{y_j} \frac{\delta y_j}{\delta z_i} = -\frac{t_i}{y_i} \frac{\delta y_i}{\delta z_i} - \sum_{j \neq 1}^{m} \frac{t_j}{y_j} \frac{\delta y_j}{\delta z_i}$$

We subdivided the summation in two cases: $i = j$ end $i \neq j$ so in these cases the derivative of the softmax function can be calculated as:
if i = j:

$$\frac{\delta y_i}{\delta z_i} = \frac{\delta \frac{e^{z_i}}{\sum_C}}{\delta z_i} = \frac{e^{z_i} \sum_C - e^{z_i} e^{z_i}}{\sum_C^2} = \frac{e^{z_i}}{\sum_C} \frac{\sum_C - e^{z_i}}{\sum_C} = \frac{e^{z_i}}{\sum_C} (1 - \frac{e^{z_i}}{\sum_C} = y_i(1 - y_i)$$

if i $\neq$ j:

$$\frac{\delta y_j^{(2)}}{\delta z_i^{(2)}} = \frac{0 - e_j^{z^{(2)}} e_i^{z^{(2)}}}{\sum_k^C (e_k z^{(2)}) \sum_k^C (e_k^{z^{(2)}})} = -\frac{e_j^{z^{(2)}}}{\sum_k^C e_i^{k^{(2)}}} \frac{e_i^{z^{(2)}}}{\sum_k^C e_i^{k^{(2)}}} = -y_j y_i$$

Thus:

$$-\frac{t_i}{y_i}y_i(1-y_i)-\sum_{j\neq 1}^{m}\frac{t_j}{y_j}(-y_jy_i) = -t_i+t_iy_i+\sum_{j\neq 1}^{m}t_jy_i = -t_i+\sum_{j=1}^{m}t_jy_i = -t_i+\sum_{j=1}^{m}t_j = y_i-t_i$$

c. Starting from:

$$\frac{\delta E}{\delta y^{(1)}} = \frac{\delta E}{\delta z^{(2)}}\frac{\delta z^{(2)}}{\delta y^{(1)}} \tag{2}$$

Thanks to the derivatives expressed before we know:

$$\frac{\delta E}{\delta z^{(2)}} = y_i - t_i$$

The second term is:

$$\frac{\delta z^{(2)}}{\delta y^{(1)}}$$

Which corresponds to the derivative with respect of the inputs $y^{(1)}$ of the vector containing the weighted inputs $z^{(2)}$. Thus the result is straightforward:

$$\frac{\delta E}{\delta y^{(1)}} = (y_i - t_i)^T\mathbf{W}^{(2)} \tag{3}$$

d. Starting from:

$$\frac{\delta E}{\delta w_{uv}^{(2)}} = \frac{\delta E}{\delta z^{(2)}}\frac{\delta z^{(2)}}{\delta w_{uv}^{(2)}} \tag{4}$$

Thanks to the derivatives expressed before we know:

$$\frac{\delta E}{\delta z^{(2)}} = y_i - t_i$$

The second term is:

$$\frac{\delta z^{(2)}}{\delta w_{uv}^{(2)}}$$

Which corresponds to the derivative with respect to the weight $w_{uv}^{(2)}$ of the vector containing the weighted inputs $z^{(2)}$. Thus the result is straightforward:

$$(y_u^{(2)} - t_u)y_v^{(1)} \tag{5}$$

This result can be extended to the case of the complete weights matrix since the behaviour is the same. Thus:

$$\frac{\delta E}{\delta W^{(2)}} = (y^{(2)} - t)y^{(1)T} \tag{6}$$

e. Calculating $\delta y^{(1)}/\delta z^{(1)}$ corresponds to calculate the derivative of the sigmoid function, in particular:

$$\frac{\delta\sigma(x)}{\delta x} = \frac{\delta}{\delta x}\frac{1}{1+e^{-x}} = -\frac{1}{(1+e^{-x})^2}\frac{\delta}{\delta x}(1+e^{-x}) = \frac{1}{(1+e^{-x})^2}e^{-x}$$

$$= \frac{1}{(1+e^{-x})}\frac{e^{-x}+1-1}{(1+e^{-x})} = \frac{1}{(1+e^{-x})}\Big(1-\frac{1}{(1+e^{-x})}\Big) = \sigma(x)(1-\sigma(x)) \tag{7}$$

We will use this result to compute the derivative for each element of $z^{(1)}$, taking into account the fact that is 0 when i $\neq$ j is 0:

$$\frac{\delta \mathbf{y}^{(1)}}{\delta \mathbf{z}^{(1)}} = \text{diag}(\mathbf{y}^{(1)}. * (1-\mathbf{y}^{(1)})) \tag{8}$$

f. Starting from:

$$\frac{\delta E}{\delta z^{(1)}} = \frac{\delta E}{\delta y^{(1)}}\frac{\delta y^{(1)}}{\delta z^{(1)}}$$

Thanks from the results showed before in (3) and (7) the result is straightforward:

$$\frac{\delta E}{\delta z^{(1)}} = (y^{(2)}-t)^T W^{(2)} diag(y^{(1)}. \cdot (1-y^{(1)})) \tag{9}$$

g. Applying the chain rule, the derivative $\delta E/\delta W_{uv}^{(1)}$ can be rewritten as:

$$\frac{\delta E}{\delta W_{uv}^{(1)}} = \frac{\delta E}{\delta z^{(1)}}\frac{\delta z^{(1)}}{\delta W_{uv}^{(1)}} \tag{10}$$

The first term was calculated in the step above, while the second corresponds to the derivative with respect to the weights of linear combination of the weights and input values gives as result the input values. So we obtain:

$$\frac{\delta E}{\delta W_{uv}^{(1)}} = ((y^{(2)}-t)^T W^{(2)} \text{diag}(y^{(1)}. \cdot (1-y^{(1)})))^T x_v^T$$

h. In the case we have multiple training sample, each of them will provide a specific gradient descend for each weight. Using only the gradient relative to one sample would over fit the model to a specific sample; but we want to train the model in a manner that minimize the error for every sample. The idea is to average the coefficients retrieved for a specific value of the weight matrices. Averaging over examples will make the variation in the gradient lower than using a single sample, resulting in a more consistent learning.

i. If we make use of the weight decay we will have add the term: $\frac{1}{2}||w||^2$, then its derivative will be λw. Thus we will only need to add the term λw to the gradient formulas.

Exercise 2

The output of the implemented program is the following:

The total loss on the training data is 2.301907
The classification loss (i.e. without weight decay) on the training data is 2.301907
The classification error rate on the training data is 0.889000

The total loss on the validation data is 2.301841
The classification loss (i.e. without weight decay) on the validation data is 2.301841
The classification error rate on the validation data is 0.895000

The total loss on the test data is 2.301865
The classification loss (i.e. without weight decay) on the test data is 2.301865
The classification error rate on the test data is 0.887333

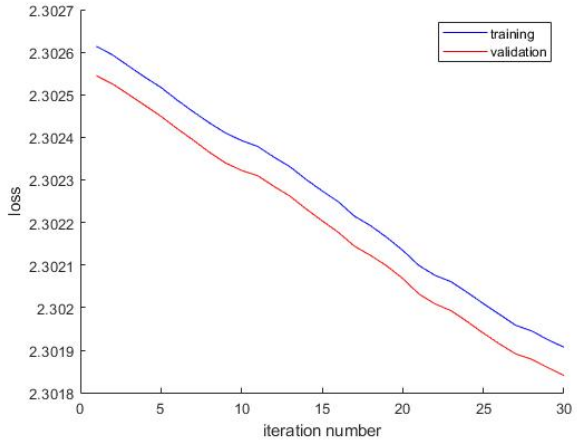
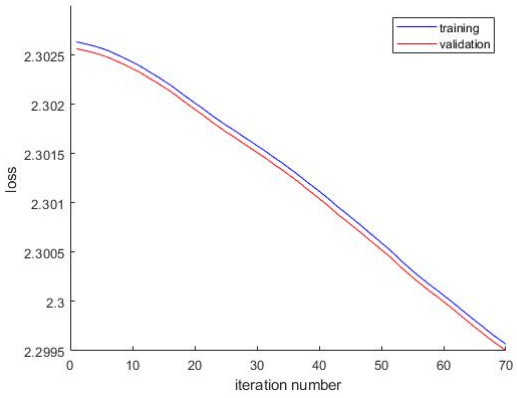


Figure 1: Optimization results

Exercise 3



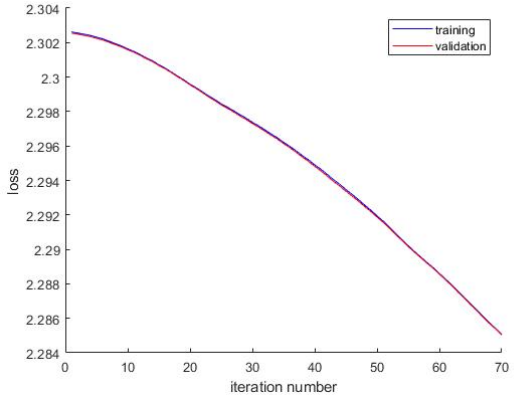
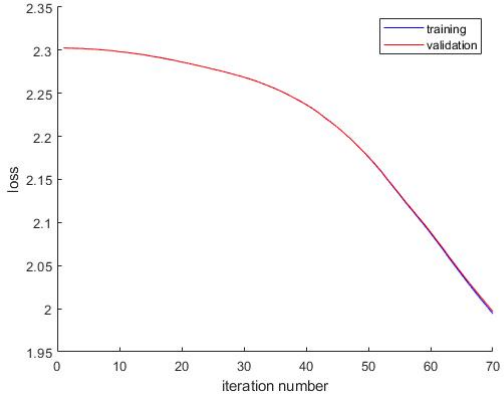


Figure 2: Optimization with momentum, learning rate 0.002 and 0.01



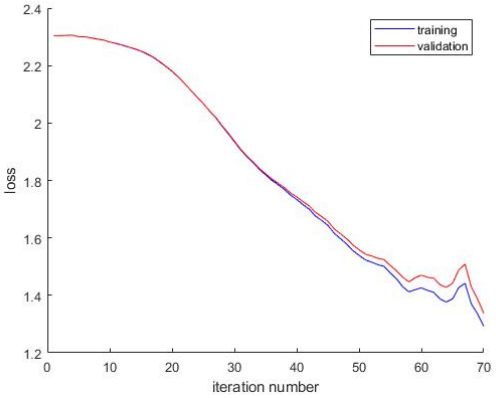
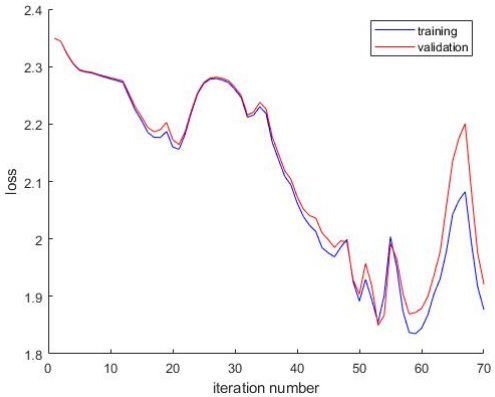


Figure 3: Optimization with momentum, learning rate 0.05 and 0.2



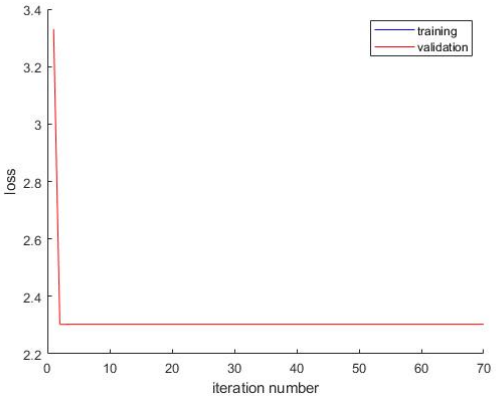


Figure 4: Optimization with momentum, learning rate 1.0 and 5.0

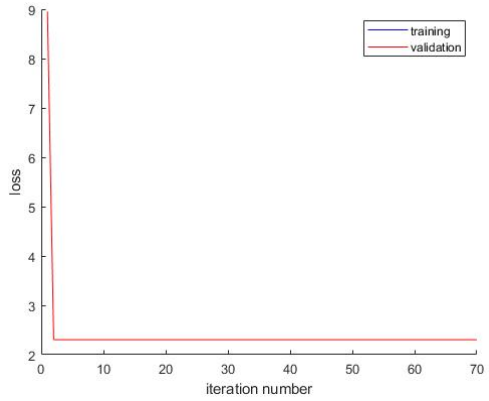
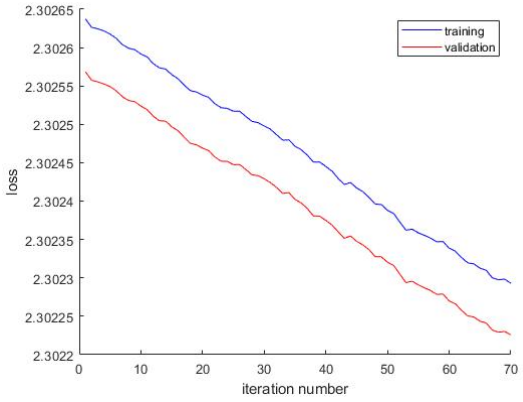


Figure 5: Optimization with momentum, learning rate 20.0



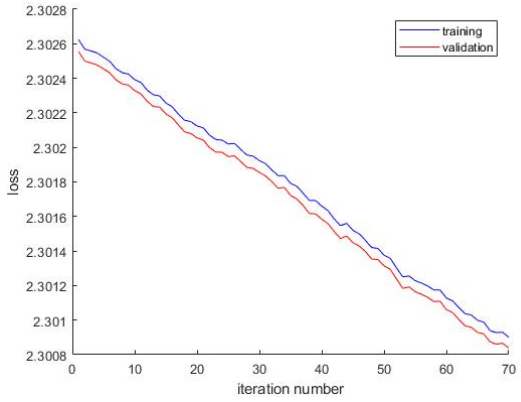
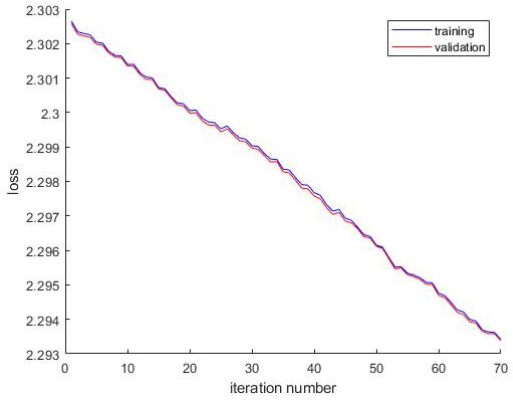


Figure 6: Optimization with momentum, learning rate 0.002 and 0.05



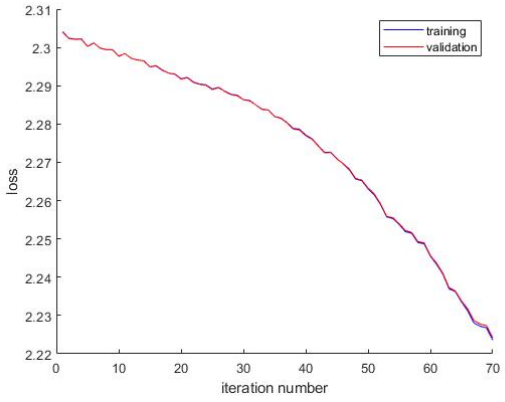


Figure 7: Optimization with momentum, learning rate 0.05 and 0.2

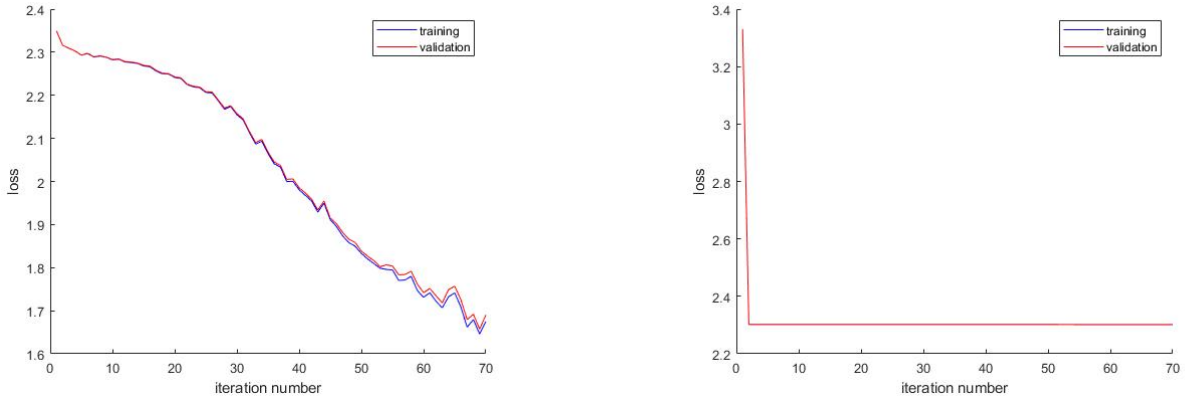


Figure 8: Optimization with momentum, learning rate 1.0 and 5.0

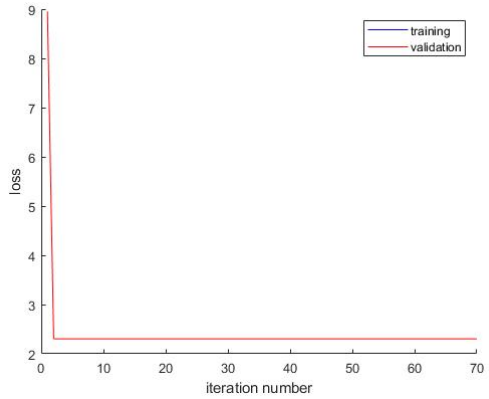


Figure 9: Optimization without momentum, learning rate 20.0

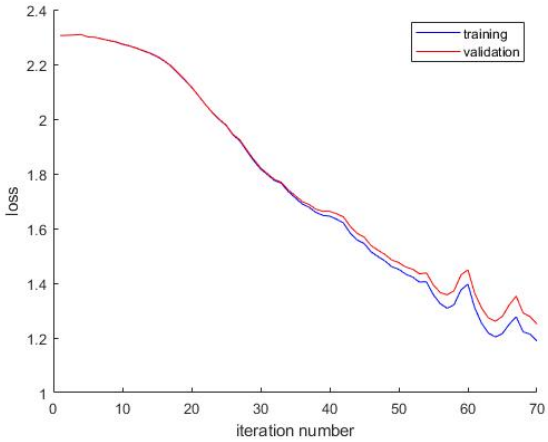


Figure 10: Optimization without momentum and learning rate 0.26

- The best result has been found, by fine tuning, with a learning rate of 0.26. The optimization ended with a loss on the training data of 1.188545.