Numerical Optimization
Assignment 3

s.guda@student.tugraz.at
patrick.gaisberger@student.tugraz.at

January 2023

1 Neural Networks

1. For the given dataset, what is the size of the matrices W(0), W(1), and the bias terms b(0) and b(1)? Specify the number of learnable parameters of the neural network as a function of N_H

   a Function h is our softplus function.

   $$z^{(1)} = W^{(0)}x + b^{(0)}$$
   $$a^{(1)} = h(z^{(1)}) \Longleftrightarrow a_i^{(1)} = \frac{1}{1 + exp(-z_i^{(1)})}$$

   $$W^{(0)} \in R^{N_H \times 2}$$
   $$b^{(0)} \in R^{N_H}$$

   b Function g is our softmax function.

   $$z^{(2)} = W^{(1)}a^{(1)} + b^{(1)}$$
   $$\tilde{y} = g(z^{(2)}) \Longleftrightarrow \tilde{y}_i = \frac{exp(z_i^{(2)})}{\sum_{j=1}^{N_O} exp(z_j^{(2)})}$$

   $$W^{(1)} \in R^{3 \times N_H}$$
   $$b^{(1)} \in R^3$$

   c Number of learnable parameters function $f(N_H)$
   are the learnable parameters in $W^{(0)} + b^{(0)} + W^{(1)} + b^{(1)}$.

   $$f(N_H) = 2N_H + N_H + 3N_H + 3 = 6N_H + 3$$

2. Compute the derivative of the total loss w.r.t. to all model parameters using the results from the practical exercise session.
   Derivation over the second weight $W^{(1)}$.

$$\frac{\partial l}{\partial W_{ki}^{(1)}} = \frac{\partial l}{\partial \tilde{y}_k} \frac{\partial \tilde{y}_k}{\partial z_k^{(2)}} \frac{\partial z_k^{(2)}}{\partial W_{ki}^{(0)}}$$

$$\frac{\partial l}{\partial \tilde{y}_k} = \frac{y_k}{\tilde{y}_k}$$

$$\frac{\partial \tilde{y}_k}{\partial z_k^{(2)}} = \frac{\partial}{\partial z_k^{(2)}} \frac{1}{1 + e^{-z_k^{(2)}}} = \frac{-e^{z_k^{(2)}}}{(e^{z_k^{(2)}} + 1)^2}$$

$$\frac{\partial z_k^{(2)}}{\partial W_{ki}^{(0)}} = a_i^{(1)}$$

$$\implies \frac{\partial l}{\partial W_{ki}^{(1)}} = \frac{y_k}{\tilde{y}_k} \frac{-e^{z_k^{(2)}}}{(e^{z_k^{(2)}} + 1)^2} a_i^{(1)}$$

Derivation over the second bias term $b^{(1)}$

$$\frac{\partial l}{\partial b_k^{(1)}} = \frac{\partial l}{\partial \tilde{y}_k} \frac{\partial \tilde{y}_k}{\partial z_k^{(2)}} \frac{\partial z_k^{(2)}}{\partial b_k^{(1)}}$$
$$\implies \frac{\partial l}{\partial b^{(1)}} = \frac{y_k}{\tilde{y}_k} \frac{-e^{z_k^{(2)}}}{(e^{z_k^{(2)}}+1)^2} \; 1$$

Derivation over the first weight $W^{(0)}$.

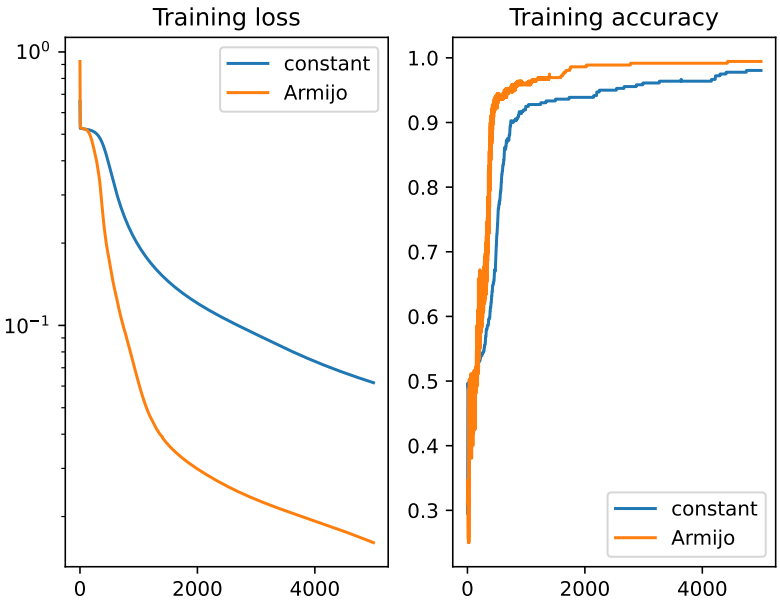
$$\frac{\partial l}{\partial W_{k,i}^{(0)}} = \frac{\partial l}{\partial a_j^{(1)}} \frac{\partial a_j^{(1)}}{\partial z_k^{(1)}} \frac{\partial z_k^{(1)}}{\partial W_{k,i}^{(0)}}$$
$$\frac{\partial l}{\partial a_j^1} = \sum_{k=1}^{N} \frac{\partial l}{\partial \tilde{y}_k} \frac{\partial \tilde{y}_k}{\partial z_k^{(2)}} \frac{\partial z_k^{(2)}}{\partial a_j^1} = \sum_{k=1}^{N} \frac{y_k}{\tilde{y}_k} \frac{-e^{z_k^{(2)}}}{(e^{z_k^{(2)}}+1)^2} \; W_{k,j}^{(0)}$$
$$\frac{\partial a_j^{(1)}}{\partial z_k^{(1)}} = \frac{\partial}{\partial z_k^{(1)}} \frac{1}{1+exp(-z_j^{(1)})} = \frac{e^{z_k^{(1)}}}{e^{z_k^{(1)}}+1}$$
$$\frac{\partial z_k^{(1)}}{\partial W_{k,i}^{(0)}} = x_i$$
$$\implies \frac{\partial l}{\partial W_{k,i}^{(0)}} = \left(\sum_{k=1}^{N} \frac{y_k}{\tilde{y}_k} \frac{-e^{z_k^{(2)}}}{(e^{z_k^{(2)}}+1)^2} \; W_{k,j}^{(0)} \right) \frac{e^{z_k^{(1)}}}{e^{z_k^{(1)}}+1} x_i$$

Derviation over first bias term $b^{(0)}$

$$\frac{\partial l}{\partial b_k^{(0)}} = \frac{\partial l}{\partial a_j^{(1)}} \frac{\partial a_j^{(1)}}{\partial z_k^{(1)}} \frac{\partial z_k^{(1)}}{\partial b_k^{(0)}}$$
$$\implies \frac{\partial l}{\partial b_k^{(1)}} = \left(\sum_{k=1}^{N} \frac{y_k}{\tilde{y}_k} \frac{-e^{z_k^{(2)}}}{(e^{z_k^{(2)}}+1)^2} \; W_{k,j}^{(0)} \right) \frac{e^{z_k^{(1)}}}{e^{z_k^{(1)}}+1} \; 1$$

2 Python Part

2.1 Training



2.2 Testing

Constant step size of 1:

1. Test loss: 6.369
2. Accuracy: 0.311

Armijo($\alpha = 10, \sigma = 10^{-4}, \beta = 0.5$)

1. Test loss: 14.317
2. Accuracy: 0.3

As we can see the test loss of Armijo is higher and the accuracy is lower than the respective results of the constant version. This is due to the fact that Armijo was reaching the minima of the function faster than the other approach (see plots). Hence, Armijo was overfitting much faster, but both approaches over fitted the training data (high test loss and low train loss).

2.3 Decision Boundaries