TÉCNICO LISBOA

Aprendizagem 2021/22

Homework III

Deadline 5/11/2021 23:59 via Fenix as PDF

- Homework limited to 5 pages (3. 5 −4pp for part I, 1−1.5pp for part II) according to the provided template
- Include your programming code as an Appendix (no page limits)
- Submission Gxxx.PDF in Fenix where xxx is your group number. Please note that it is possible to submit several times on Fenix to prevent last-minute problems. Yet, only the last submission is considered valid
- Exchange of ideas is encouraged. Yet, if copy is detected after automatic/manual clearance, homework is nullified and IST guidelines apply for content sharers and consumers, irrespectively of the underlying intent
- Please consult the FAQ before posting questions to your faculty hosts

I. Pen-and-paper [10v]

1) Consider a MLP classifier characterized by the following weights:

$$\mathbf{W}^{[1]}=\begin{pmatrix}1&1&1&1&1\\0&0&0&0&0\\1&1&1&1&1\end{pmatrix},\mathbf{b}^{[1]}=\begin{pmatrix}1\\1\\1\end{pmatrix},\mathbf{W}^{[2]}=\begin{pmatrix}1&1&1\\1&1&1\end{pmatrix},\mathbf{b}^{[2]}=\begin{pmatrix}1\\1\end{pmatrix},\mathbf{W}^{[3]}=\begin{pmatrix}0&0\\0&0\end{pmatrix},\mathbf{b}^{[3]}=\begin{pmatrix}0\\0\end{pmatrix}$$

a. [5v] Using the hyperbolic tangent activation function $f(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}=tanh(x)$ on all units, and the squared error loss, perform a stochastic gradient descent update (with learning rate $\eta=0.1$) for the training example $\mathbf{x}=(1,1,1,1,1)^T$ with positive target, i.e. $\mathbf{z}=(1,-1)^T$.

We start by writing the initial connection weights and the biases

$$\mathbf{W}^{[1]}=\begin{pmatrix}1&1&1&1&1\\0&0&0&0&0\\1&1&1&1&1\end{pmatrix},\mathbf{W}^{[2]}=\begin{pmatrix}1&1&1\\1&1&1\end{pmatrix},\mathbf{W}^{[3]}=\begin{pmatrix}0&0\\0&0\end{pmatrix}$$

$$\mathbf{b}^{[1]}=\begin{pmatrix}1\\1\\1\end{pmatrix},\mathbf{b}^{[2]}=\begin{pmatrix}1\\1\end{pmatrix},\mathbf{b}^{[3]}=\begin{pmatrix}0\\0\end{pmatrix}$$

We are now ready to do forward propagation

$$\mathbf{net}^{[1]}=\begin{pmatrix}1&1&1&1&1\\0&0&0&0&0\\1&1&1&1&1\end{pmatrix}\begin{pmatrix}1\\1\\1\\1\\1\end{pmatrix}+\begin{pmatrix}1\\1\\1\end{pmatrix}=\begin{pmatrix}6\\1\\6\end{pmatrix}$$

$$\mathbf{x}^{[1]}=\begin{pmatrix}\tanh(6)\\\tanh(1)\\\tanh(6)\end{pmatrix}$$

$$\mathbf{net}^{[2]}=\begin{pmatrix}1&1&1\\1&1&1\end{pmatrix}\begin{pmatrix}\tanh(6)\\\tanh(1)\\\tanh(6)\end{pmatrix}+\begin{pmatrix}1\\1\end{pmatrix}=\begin{pmatrix}2\tanh(6)+\tanh(1)+1\\2\tanh(6)+\tanh(1)+1\end{pmatrix}$$

$$\mathbf{x}^{[2]}=\begin{pmatrix}\tanh(2\tanh(6)+\tanh(1)+1)\\\tanh(2\tanh(6)+\tanh(1)+1)\end{pmatrix}=\begin{pmatrix}0.9989\\0.9989\end{pmatrix}$$

$$\mathbf{net}^{[3]} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \tanh(2\tanh(6) + \tanh(1) + 1) \\ \tanh(2\tanh(6) + \tanh(1) + 1) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\mathbf{x}^{[3]} = \begin{pmatrix} \tanh(0) \\ \tanh(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Recalling the squared error measure:

$$E(\mathbf{x}^{[3]}, \mathbf{z}) = \frac{1}{2} \sum_{i=1}^{2} \left(\mathbf{x}^{[3]}{}_i - \mathbf{z}_i\right)^2$$

We can start the backward phase:

$$\frac{\partial E}{\partial \mathbf{x}^{[3]}} = \mathbf{x}^{[3]} - \mathbf{z} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$\frac{\partial \mathbf{x}^{[3]}}{\partial \mathbf{net}^{[3]}} = 1 - \mathbf{x}^{[3]^2} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\delta^{[3]} = \frac{\partial E}{\partial \mathbf{net}^{[3]}} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$\frac{\partial \mathbf{net}^{[3]}}{\partial \mathbf{x}^{[2]}} = \mathbf{W}^{[3]^T} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\frac{\partial \mathbf{x}^{[2]}}{\partial \mathbf{net}^{[2]}} = 1 - \mathbf{x}^{[2]^2} = \begin{pmatrix} 1 - 0.9989^2 \\ 1 - 0.9989^2 \end{pmatrix}$$

$$\delta^{[2]} = \frac{\partial E}{\partial \mathbf{net}^{[2]}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial \mathbf{net}^{[2]}}{\partial \mathbf{x}^{[1]}} = \mathbf{W}^{[2]^T} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$$\frac{\partial \mathbf{x}^{[1]}}{\partial \mathbf{net}^{[1]}} = 1 - \mathbf{x}^{[1]^2} = \begin{pmatrix} 1 - \tanh(6)^2 \\ 1 - \tanh(1)^2 \\ 1 - \tanh(6)^2 \end{pmatrix}$$

$$\delta^{[1]} = \frac{\partial E}{\partial \mathbf{net}^{[1]}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Having the deltas, we can compute the gradients for every parameter:

$$\frac{\partial E}{\partial \mathbf{W}^{[1]}} = \delta^{[1]} \frac{\partial \mathbf{net}^{[1]}}{\partial \mathbf{W}^{[1]}} = \delta^{[1]} (\mathbf{x}^{[0]})^T = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}^T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \frac{\partial E}{\partial \mathbf{b}^{[1]}} = \delta^{[1]} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial E}{\partial \mathbf{W}^{[2]}} = \delta^{[2]} \frac{\partial \mathbf{net}^{[2]}}{\partial \mathbf{W}^{[2]}} = \delta^{[2]} (\mathbf{x}^{[1]})^T = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} \tanh(6) \\ \tanh(1) \\ \tanh(6) \end{pmatrix}^T = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \frac{\partial E}{\partial \mathbf{b}^{[2]}} = \delta^{[2]} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial E}{\partial \mathbf{W}^{[3]}} = \delta^{[3]} \frac{\partial \mathbf{net}^{[3]}}{\partial \mathbf{W}^{[3]}} = \delta^{[3]} (\mathbf{x}^{[2]})^T = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \begin{pmatrix} 0.9989 \\ 0.9989 \end{pmatrix}^T = \begin{pmatrix} -0.9989 & -0.9989 \\ 0.9989 & 0.9989 \end{pmatrix}, \frac{\partial E}{\partial \mathbf{b}^{[3]}} = \delta^{[3]} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

And now do the SGD updates:

$$\mathbf{W}^{[1]} = \mathbf{W}^{[1]} - \eta\frac{\partial E}{\partial \mathbf{W}^{[1]}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} - 0.1\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{b}^{[1]} = \mathbf{b}^{[1]} - \eta\frac{\partial E}{\partial \mathbf{b}^{[1]}} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - 0.1\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\mathbf{W}^{[2]} = \mathbf{W}^{[2]} - \eta\frac{\partial E}{\partial \mathbf{W}^{[2]}} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} - 0.1\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{b}^{[2]} = \mathbf{b}^{[2]} - \eta\frac{\partial E}{\partial \mathbf{b}^{[2]}} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - 0.1\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\mathbf{W}^{[3]} = \mathbf{W}^{[3]} - \eta\frac{\partial E}{\partial \mathbf{W}^{[3]}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} - 0.1\begin{pmatrix} -0.9989 & -0.9989 \\ 0.9989 & 0.9989 \end{pmatrix} = \begin{pmatrix} 0.09989 & 0.09989 \\ -0.09989 & -0.09989 \end{pmatrix}$$

$$\mathbf{b}^{[3]} = \mathbf{b}^{[3]} - \eta\frac{\partial E}{\partial \mathbf{b}^{[3]}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - 0.1\begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.1 \\ -0.1 \end{pmatrix}$$

b. [5v] Replacing the activation function on the output unit by softmax and the loss function by cross-entropy, perform a stochastic gradient descent update (with learning rate $\eta = 0.1$) for the same example. Note: under softmax, a positive target is defined as $\mathbf{z} = (1{,}0)^T$.

We start by writing the initial connection weights and the biases

$$\mathbf{W}^{[1]} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \mathbf{W}^{[2]} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \mathbf{W}^{[3]} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{b}^{[1]} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \mathbf{b}^{[2]} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{b}^{[3]} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

We are now ready to do forward propagation

$$\mathbf{net}^{[1]} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 1 \\ 6 \end{pmatrix}$$

$$\mathbf{x}^{[1]} = \begin{pmatrix} \tanh(6) \\ \tanh(1) \\ \tanh(6) \end{pmatrix}$$

$$\mathbf{net}^{[2]} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} \tanh(6) \\ \tanh(1) \\ \tanh(6) \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2\tanh(6) + \tanh(1) + 1 \\ 2\tanh(6) + \tanh(1) + 1 \end{pmatrix}$$

$$\mathbf{x}^{[2]} = \begin{pmatrix} \tanh(2\tanh(6) + \tanh(1) + 1) \\ \tanh(2\tanh(6) + \tanh(1) + 1) \end{pmatrix} = \begin{pmatrix} 0.9989 \\ 0.9989 \end{pmatrix}$$

$$\mathbf{net}^{[3]} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \tanh(2\tanh(6) + \tanh(1) + 1) \\ \tanh(2\tanh(6) + \tanh(1) + 1) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\mathbf{x}^{[3]} = softmax\begin{pmatrix}0\\0\end{pmatrix} = \begin{pmatrix}0.5\\0.5\end{pmatrix}$$

Recalling the cross entropy error measure:

$$E\left(\mathbf{x}^{[3]}, \boldsymbol{z}\right) = -\sum_{i=1}^{2} \boldsymbol{z}_i \log(\mathbf{x}^{[3]}{}_i)$$

We can start the backward phase:

$$\frac{\partial E}{\partial \mathbf{x}^{[3]}} = -\frac{\mathbf{z}}{\mathbf{x}^{[3]}} = \begin{pmatrix}-\dfrac{1}{0.5}\\ -\dfrac{0}{0.5}\end{pmatrix} = \begin{pmatrix}-2\\0\end{pmatrix}$$

$$\frac{\partial \mathbf{x}^{[3]}}{\partial \mathbf{net}^{[3]}} = \begin{pmatrix}\mathbf{x}^{[3]}{}_1(1-\mathbf{x}^{[3]}{}_1) & -\mathbf{x}^{[3]}{}_1\mathbf{x}^{[3]}{}_2\\ -\mathbf{x}^{[3]}{}_1\mathbf{x}^{[3]}{}_2 & \mathbf{x}^{[3]}{}_2(1-\mathbf{x}^{[3]}{}_2)\end{pmatrix} = \begin{pmatrix}1/4 & -1/4\\ -1/4 & 1/4\end{pmatrix}$$

$$\delta^{[3]} = \frac{\partial E}{\partial \mathbf{net}^{[3]}} = \begin{pmatrix}1/4 & -1/4\\ -1/4 & 1/4\end{pmatrix}\begin{pmatrix}-2\\0\end{pmatrix} = \begin{pmatrix}-1/2\\1/2\end{pmatrix}$$

$$\frac{\partial \mathbf{net}^{[3]}}{\partial \mathbf{x}^{[2]}} = \mathbf{W}^{[3]T} = \begin{pmatrix}0 & 0\\0 & 0\end{pmatrix}$$

$$\frac{\partial \mathbf{x}^{[2]}}{\partial \mathbf{net}^{[2]}} = 1 - \mathbf{x}^{[2]2} = \begin{pmatrix}1-0.9989^2\\1-0.9989^2\end{pmatrix}$$

$$\delta^{[2]} = \frac{\partial E}{\partial \mathbf{net}^{[2]}} = \begin{pmatrix}0\\0\end{pmatrix}$$

$$\frac{\partial \mathbf{net}^{[2]}}{\partial \mathbf{x}^{[1]}} = \mathbf{W}^{[2]T} = \begin{pmatrix}1 & 1\\1 & 1\\1 & 1\end{pmatrix}$$

$$\frac{\partial \mathbf{x}^{[1]}}{\partial \mathbf{net}^{[1]}} = 1 - \mathbf{x}^{[1]2} = \begin{pmatrix}1-\tanh(6)^2\\1-\tanh(1)^2\\1-\tanh(6)^2\end{pmatrix}$$

$$\delta^{[1]} = \frac{\partial E}{\partial \mathbf{net}^{[1]}} = \begin{pmatrix}0\\0\\0\end{pmatrix}$$

Having the deltas, we can compute the gradients for every parameter:

$$\frac{\partial E}{\partial \mathbf{W}^{[1]}} = \delta^{[1]}\frac{\partial \mathbf{net}^{[1]}}{\partial \mathbf{W}^{[1]}} = \delta^{[1]}(\mathbf{x}^{[0]})^T = \begin{pmatrix}0\\0\\0\end{pmatrix}\begin{pmatrix}1\\1\\1\\1\\1\end{pmatrix}^T = \begin{pmatrix}0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\end{pmatrix}, \frac{\partial E}{\partial \mathbf{b}^{[1]}} = \delta^{[1]} = \begin{pmatrix}0\\0\\0\end{pmatrix}$$

$$\frac{\partial E}{\partial \mathbf{W}^{[2]}} = \delta^{[2]}\frac{\partial \mathbf{net}^{[2]}}{\partial \mathbf{W}^{[2]}} = \delta^{[2]}(\mathbf{x}^{[1]})^T = \begin{pmatrix}0\\0\end{pmatrix}\begin{pmatrix}\tanh(6)\\\tanh(1)\\\tanh(6)\end{pmatrix}^T = \begin{pmatrix}0 & 0 & 0\\0 & 0 & 0\end{pmatrix}, \frac{\partial E}{\partial \mathbf{b}^{[2]}} = \delta^{[2]} = \begin{pmatrix}0\\0\end{pmatrix}$$

$$\frac{\partial E}{\partial \mathbf{W}^{[3]}} = \delta^{[3]}\frac{\partial \mathbf{net}^{[3]}}{\partial \mathbf{W}^{[3]}} = \delta^{[3]}(\mathbf{x}^{[2]})^T = \begin{pmatrix}-1/2\\1/2\end{pmatrix}\begin{pmatrix}0.9989\\0.9989\end{pmatrix}^T = \begin{pmatrix}-0.4995 & -0.4995\\0.4995 & 0.4995\end{pmatrix}, \frac{\partial E}{\partial \mathbf{b}^{[3]}} = \delta^{[3]}$$

$$= \begin{pmatrix}-1/2\\1/2\end{pmatrix}$$

And now do the SGD updates:

$$\mathbf{W}^{[1]} = \mathbf{W}^{[1]} - \eta \frac{\partial E}{\partial \mathbf{W}^{[1]}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} - 0.1 \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{b}^{[1]} = \mathbf{b}^{[1]} - \eta \frac{\partial E}{\partial \mathbf{b}^{[1]}} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - 0.1 \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\mathbf{W}^{[2]} = \mathbf{W}^{[2]} - \eta \frac{\partial E}{\partial \mathbf{W}^{[2]}} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} - 0.1 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{b}^{[2]} = \mathbf{b}^{[2]} - \eta \frac{\partial E}{\partial \mathbf{b}^{[2]}} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - 0.1 \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

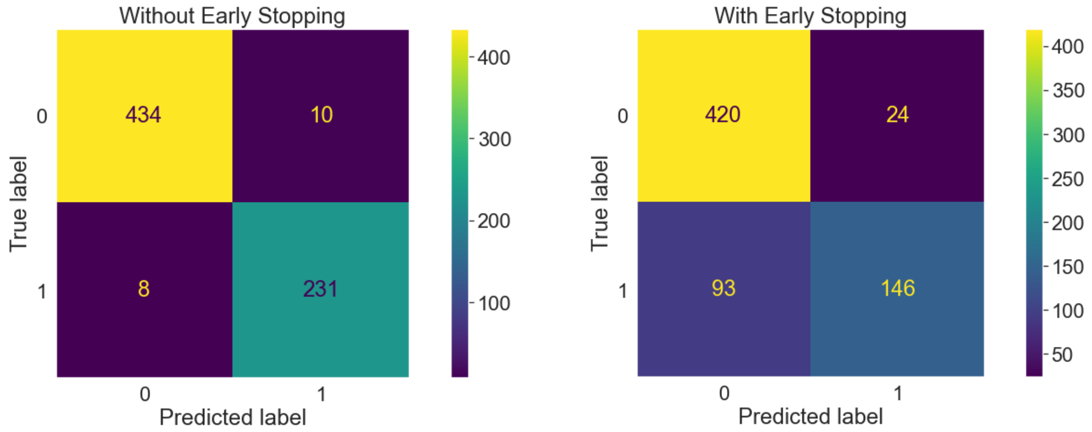
$$\mathbf{W}^{[3]} = \mathbf{W}^{[3]} - \eta \frac{\partial E}{\partial \mathbf{W}^{[3]}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} - 0.1 \begin{pmatrix} -0.4995 & -0.4995 \\ 0.4995 & 0.4995 \end{pmatrix} = \begin{pmatrix} 0.04995 & 0.04995 \\ -0.04995 & -0.04995 \end{pmatrix}$$

$$\mathbf{b}^{[3]} = \mathbf{b}^{[3]} - \eta \frac{\partial E}{\partial \mathbf{b}^{[3]}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - 0.1 \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix} = \begin{pmatrix} 0.05 \\ -0.05 \end{pmatrix}$$

II. **Programming and critical analysis** [10v]

Use **sklearn** to answer the following questions. Consider a MLP with l_2 regularization, RELU activation functions in the hidden layers, and an architecture resembling that described in Part I, i.e. two hidden layers of size 3 and 2. Consider all the remaining MLP parameters as the defaults in sklearn (e.g. cross-entropy loss for classification and squared error loss for regression).

2) [5v] Using the `breast.w.arff` data from previous homeworks, show the confusion matrix of the aforementioned MLP in the presence and absence of early stopping. Briefly enumerate two reasons for the observed differences.



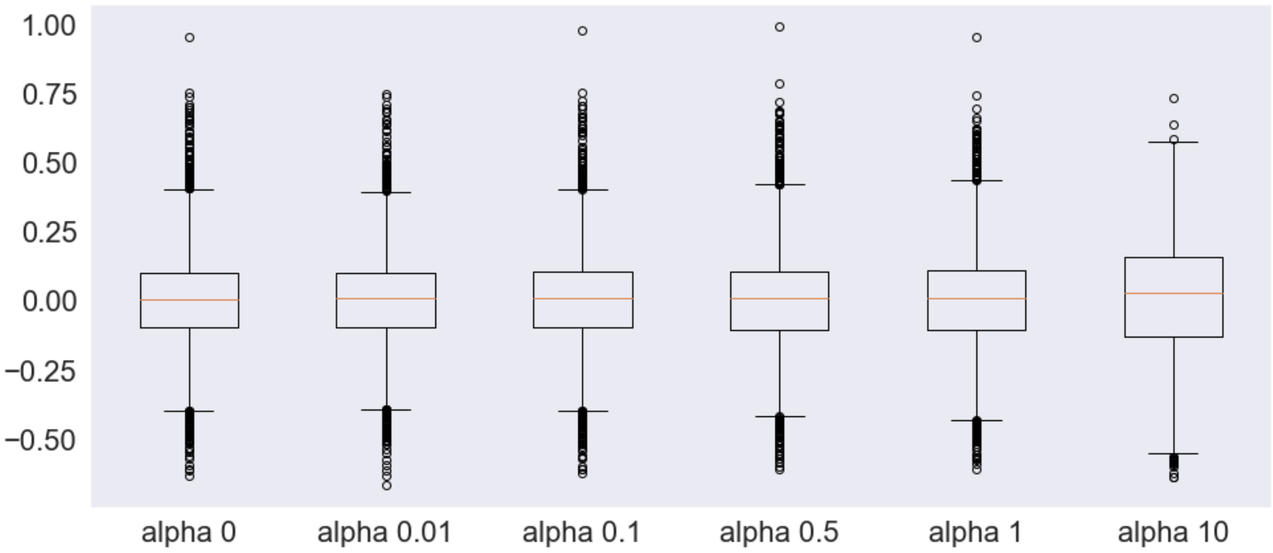
Performance improves without early stopping.

Possible reasons are:

- The relatively small dataset size and it's further subdivision by both early stopping and cross validation means the amount of available data for the model to learn from is low, resulting in underfitting;
- Early stopping may be stuck in a local minimum
- Early stopping in SKLearn stops training when the error doesn't change for a certain number of iterations, this amount may be overly optimistic
- The used MLP is very simple, and thus has no propension towards overfitting

3) [5v] Using the `kin8nm.arff`, plot the distribution of the residues using boxplots in the presence and absence of regularization. Identify 4 strategies to minimize the observed error of the MLP regressor.

Note: consider a 5-CV with a fixed zero seed to answer (3) and (4).
`Kin8nm.arff` available at https://fenix.tecnico.ulisboa.pt/downloadFile/845043405555949/kin8nm.arff



We can see that regularization seems to reduce the number of outliers, although it also seems to increase the Inter-Quartile Range (IQR). Differences overall are relatively small.

Some strategies we could take to improve performance are:

- Increase the number of layers in the network
- Increase the number of units in each layer
- Increasing regulation seems to reduce outliers (and could prove a good strategy especially when combined with regulation)
- Change the activation function
- Change the learning rate
- Change the loss function

END