LABWORK 1

CAMA Hector, OUADAHI Fares

Task 1 :

The goal of the first task was to predict correctly the output of the AND logic operator with the given code. The target being [0,0,0,1], with a low number of iterations, the target is not reached correctly. For example, with 10 iterations, we could get [0.32,0.29,0.28,0.27]. The result is not the same if the code is run several times because of the random initialization of the weights for the first layer.

Iterations	**10**	**50**	**100**	**500**	**2000**
Loss value	0.8	0.6	0.4	0.2	~0

Table 1 : Loss value according to the number of iterations

With an increasing number of iterations (table 1), we can observe that the loss value decreases faster and therefore we get more accurate predictions for the outputs.

With a fixed value of iterations (1000), when the n_unit parameter is increased (number of neurons in the hidden layer), the SSD value is also low and the results obtained are close to the target. With a n_unit = 10, the decreasing is faster: it could be possible to reduce the number of iterations to get the expected result.

Task 2 :

For n_unit = 1, the AND operator has lower loss values. For the XOR operator, the target is not the same, and so as you can see on figure 1 below you need two lines to classify the 0 and the 1 instead of only one for the AND operator. Therefore, it is easier for the AND model to reach the target with only one neuron in the hidden layer. A number of n_units = 1 is not enough to reach the target for the XOR model. Increasing the number of neurons to 2, 5, 10, 50 gives a better result for the loss value because it gives a better classification. (courbe)

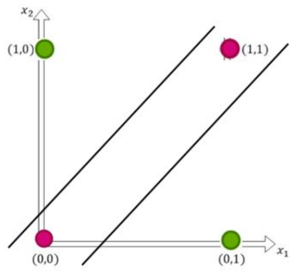

Figure 2 : Scheme for the classification of XOR operator

Task 3 :

In the given code, changing the parameter n_unit doesn't really change the results : the number of iterations has to be also increased to get better results. Improving the learning rate could also improve the outputs.

Task 4 :

The observed loss value corresponds to the difference between the label found in the training of the model and the target. When these values are close, the loss value is low and the accuracy value is high. On the plot of the learning curve with the given parameters (n_epochs = 50, Batch_Size = 16, base_dense = 64, LR = 0.0001), the loss values are not converging but they are still very close (between 0.6 and 0.7). The number of epochs is therefore not sufficient and has to be increased.

For the same number of epochs, when the LR is reduced to 0.1, the loss values stay high (around 0.7). This learning rate is too high to correctly predict the target. With epochs = 150 and LR = 0.0001, the model is still not converging (figure 3), we can see that the validation curve is not following the training curve and moves away from it : this is a case of overfitting.

Task 5 :

A. For the defined parameters, the validation and training loss values are still around 0.7 and the curves are decreasing and not converging. The accuracy is around 0.6 There is an overfitting because the validation curve does not fit the training curve. So it is not a good idea to rely on this set-up. The LR is too small.
B. When we change the n_epochs=200, the two curves fit well and the accuracy is a little better.
C. When we modify the LR, the loss value is again improved (0.63) but until a certain number of epochs because when it reaches 100 epochs the model is overfitting. We should not use more than 100 epochs. The accuracy is also improved to 0.69.
D. The two convolution layers are used to extract features from the images directly from the input for the first one and then from the pooling layer. These layers are locally connected
E. The two last dense layers are fully connected and used to modelize mathematical functions.
F. The major difference between LeNet and MLP is the fact that the input is not flatten and in MLP there are no convolutions.
G. The number of neurons in the last layer, the output should correspond to the number of classes. For binary classification only one neuron can be used for the output layer. The activation function has to be chosen according to the task asked (regression, classification)