Assignment 1

To be solved in groups of at most four elements

Submit by November 11, 2023, 11:59 PM at Moodle. Only a member of the group submits the work. The other member(s) of the group only submit a txt file stating "joint submission with [Colleague Name]".

1. **Feedforward neural networks (50%)**

   (a) (5%) To optimize deep neural networks (DNN) backpropagation and gradient descent are often used. Distinguish between these two algorithms, if there is any distinction, and explain how they can be used for DNN optimization.

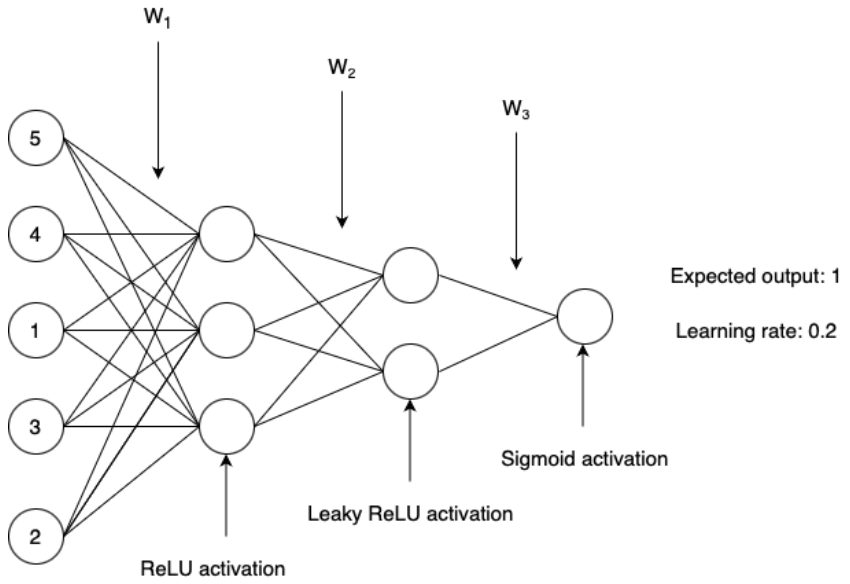

Figure 1: Neural Network for exercise (b). Represented by 2 hidden layers, 1 input layer (with input values) and 1 output layer.

   (b) (15%) Initialize all the weights values of the matrices W_1, W_2 and W_3 with the value 2. You can ignore the bias term. Calculate a forward pass and the output, and a backward pass using Binary Cross-Entropy as your loss for the architecture seen in Figure 1. As you can see the input values are already filled in the input nodes and the expected output is 1. Use a Learning Rate of 0.2. After updating your weights what do you observe? Explain why.

   (c) (10%) Train a two hidden layer architecture (similar to Figure 1) on PyTorch's Fashion-MNIST dataset (PyTorch Class). Feel free to change the number of nodes in each layer of the network. Moreover, there is a need for an adaptation of the network to the MNIST problem, ensure that you do it. State the architecture of your network, the accuracy it achieved and the Cross-Entropy on the training and test sets.

   (d) (10%) Repeat the previous exercise with a more complex feedforward neural network, different regularization techniques, or activations. Explain the changes you have made and how they contributed to a change in the results. State the accuracy it achieved and the Cross-Entropy on the training and test sets.

   (e) (10%) Given an image of 1080x1080 pixels state advantages of Network A (Figure 2) when compared to Network B (Figure 2).

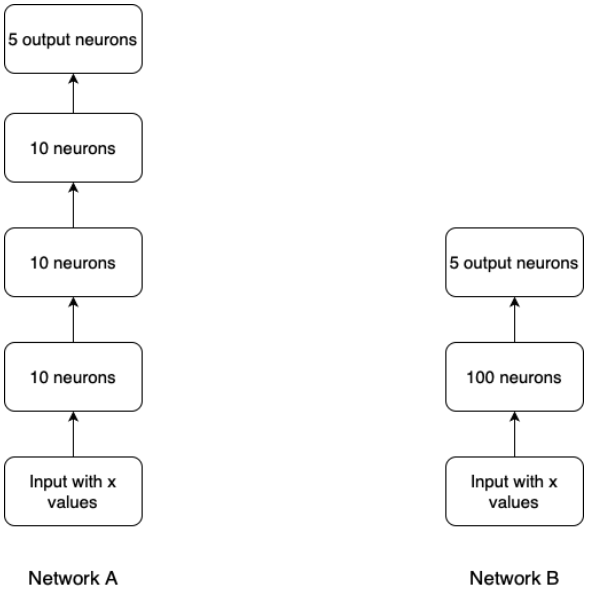

Figure 2: Two feedforward neural networks with different depth and width.

2. **Recurrent neural networks (50%)**

(a) (10%) Consider a binary classification problem in which we aim to classify a two-dimensional input $\boldsymbol{x}^{(t)}$ at each time step $t = 1, 2, \ldots$ The model is designed with a simple recurrent neural network structure that produces a scalar output $\hat{y}^{(t)}$:

$$h^{(t)} = \sigma(\boldsymbol{w}^\top \boldsymbol{x}^{(t)} + v h^{(t-1)} + b), \quad \hat{y}^{(t)} = h^{(t)},$$

where σ is the sigmoid function. The training objective is the binary cross-entropy loss.

Suppose you are training this model using stochastic gradient descent (SGD) with a unit batch size and a learning rate of 0.1. At a given training step, the input sequence, target sequence, and model parameters are:

$$\text{input: } \boldsymbol{x}^{(1)} = \begin{pmatrix} 0.3 \\ -0.2 \end{pmatrix},\ \boldsymbol{x}^{(2)} = \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}; \quad \text{target: } y^{(1)} = 1,\ y^{(2)} = 0; \tag{1}$$

$$\text{parameters: } \boldsymbol{w} = \begin{pmatrix} 1 \\ -1 \end{pmatrix},\ v = -1,\ b = 0. \tag{2}$$

Assuming the RNN initial state is zero, compute:

i. (5%) The loss value for the current example.

ii. (5%) The value of $\boldsymbol{w}$ after the current step of SGD.

(b) (40%) When training RNN-based autoregressive models, one has the option of using or not *teacher forcing*.

i. (5%) Investigate what teacher forcing is and explain it in your own words.

ii. (30%) The model implemented in `time_series_lstm_SOLVED.ipynb` does not use teacher forcing. Implement it and train it using teacher forcing. Submit the produced code / notebook together with the rest of the assignment.

iii. (5%) Which of the strategies do you expect to yield a model with better generalization? Explain your answer.

DPC, PDNC