Deep Learning for NLP 2020
Exercise 03

May 4, 2020

1 Pingo

Try to find the right answer(s) to each question on your own or in a group with your colleagues. The interactive survey will be conducted near the end of the practice class.

- How many hidden layers does a one-layer MLP (multi-layer-perceptron) have?
  - ☐ 0
  - ☐ 1
  - ☐ 2
- Which statements about gradient descent are correct?
  - ☐ Gradient descent always finds the global minimum of a loss function.
  - ☐ When using mini-batch GD, training with large mini-batches leads to smoother training (less jumpy gradient).
  - ☐ When using mini-batch GD, the learning rate should be chosen independently from the mini-batch size.
  - ☐ Regularization has the purpose of reducing the variance in weight vectors/matrices.
- Which statements about backpropagation are correct?
  - ☐ Backpropagation is a supervised learning paradigm.
  - ☐ Backpropagation computes a gradient for every hidden layer weight.
  - ☐ Backpropagation changes (updates) the hidden layer weights.

2 Weight Matrix Initialization

There are many reasons why a neural network "refuses to learn". Improper weight matrix initialization is one such issue which has strong consequences for the overall training convergence.

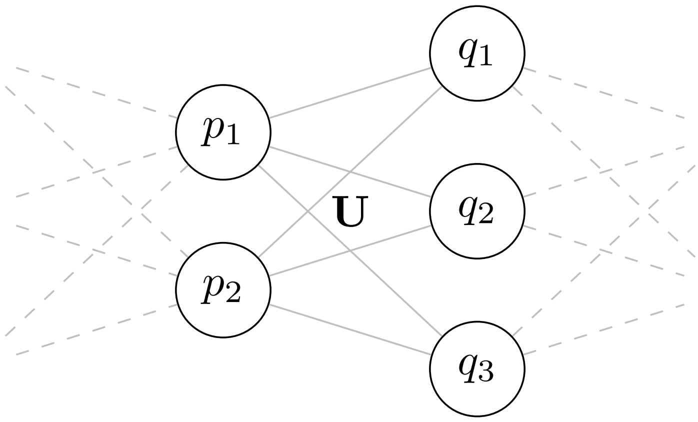

Figure 1: Two hidden layers of an MLP

1. Take a look at the MLP in figure 1. Imagine initializing all weights in matrix $\mathbf{U}$ to the same non-zero value (for example, 1). Why is this a suboptimal choice?
2. Now, imagine initializing all weights in matrix $\mathbf{U}$ being zero. How does this affect the gradients at the hidden units in layer p during backpropagation?

Hint: You can simulate both scenarios in the TensorFlow Playground[1].

3 Backpropagation by Hand

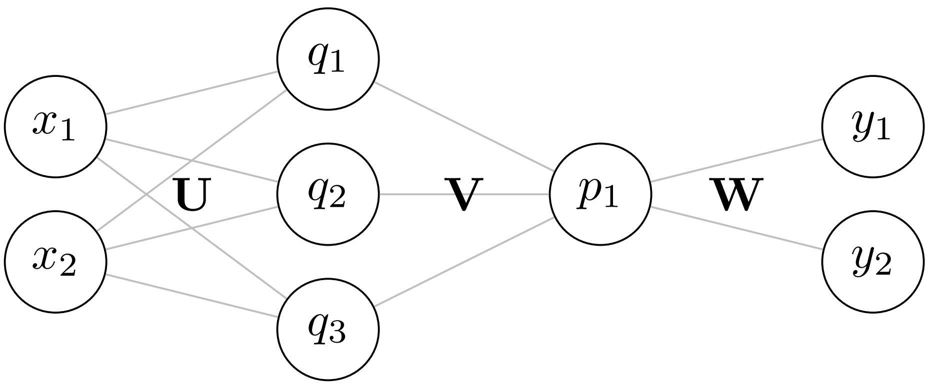

Figure 2: MLP for Backpropagation

[1] `http://playground.tensorflow.org`

Network Details Figure 2 shows an MLP without bias neurons. The hidden layers and the output layer use the sigmoid activation function:

$$\text{sig}(x) = \frac{1}{1 + \exp(-x)} \qquad \text{sig}'(x) = \text{sig}(x) \cdot (1 - \text{sig}(x))$$

For the loss function, square loss is used. t_j denotes a true label, y_j denotes a network output:

$$\ell(t_j, y_j) = (t_j - y_j)^2 \qquad \frac{\partial \ell(t_j, y_j)}{\partial y_j} = 2(y_j - t_j)$$

The weight matrices are initialized as follows:

$$\mathbf{U} = \begin{pmatrix} 1.20 & -1.20 & -0.11 \\ 0.30 & 1.10 & 0.65 \end{pmatrix} \quad \mathbf{V} = \begin{pmatrix} -0.25 \\ -1.10 \\ -0.09 \end{pmatrix} \quad \mathbf{W} = \begin{pmatrix} -2.00 & 0.43 \end{pmatrix}$$

Task Using pen and paper, perform backpropagation to compute:

$$\frac{\partial E}{\partial p_1} \quad \text{and} \quad \frac{\partial E}{\partial w_{1,1}}$$

Use $\mathbf{x} = (0, 1)$ as the input and $\mathbf{t} = (1, 1)$ as the truth label.

Round your result to two decimal points after each calculation. The necessary formulas can be found in the `tu03` slides (see Moodle).