Deep Learning for NLP 2020
Exercise 03 Solution

May 14, 2020

1 Pingo

- 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.

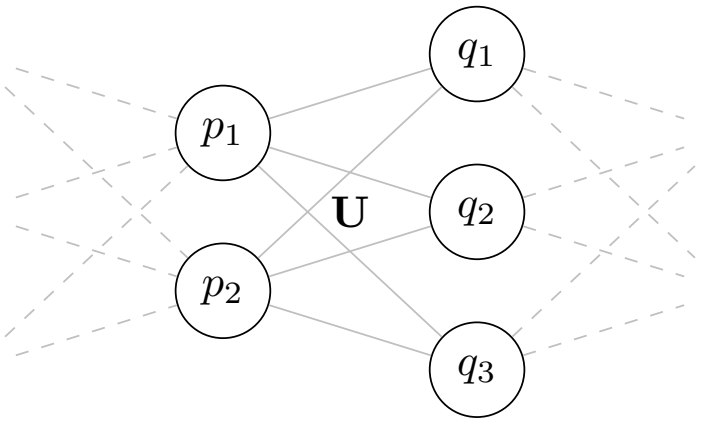

Figure 1: Two hidden layers of an MLP

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?

   Answer: If all $u_{i,j}$ are the same value, the pre-activations and activations will be the same for all hidden units in layer q. Essentially, every hidden unit in layer q will behave the same which hurts the expressiveness of the network. Layer q will become useful to the network once the variance in the weights of $\mathbf{U}$ has increased enough (via the gradients $\frac{\partial E}{\partial q_i}$ and varied activations from layer p) to cause different activations in the hidden units of layer q.

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?

   Answer: We have

   $$\frac{\partial E}{\partial p_i} = \sum_j \frac{\partial E}{\partial q_j} \cdot \sigma'(q_{j,\text{in}}) \cdot u_{i,j}$$

   and $u_{i,j} = 0$ for all i, j. The gradient $\frac{\partial E}{\partial p_i}$ would therefore always be zero. Without a gradient at layer p, none of the hidden layer weight matrices to the left of layer p (in direction of the input layer) will receive gradients during this backpropagation pass. Similar to the case in a), the network will eventually recover from the suboptimal initialization.

3 Backpropagation by Hand

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:

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

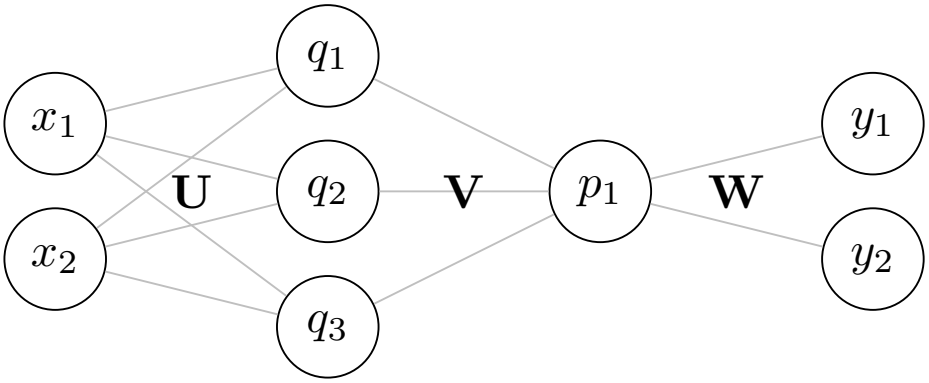

Figure 2: MLP for Backpropagation

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).

Forward propagation

$$q_1 = \text{sig}\left(\mathbf{x} \cdot \mathbf{U}_1\right) = \text{sig}\left(\begin{pmatrix} 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1.20 \\ 0.30 \end{pmatrix}\right) = \text{sig}(\underbrace{0.30}_{\text{pre-activation}}) = \underbrace{0.57}_{\text{activation}}$$

Et cetera.

Backpropagation Step 1: Error Derivative at Neurons

$$\frac{\partial E}{\partial y_1} = 2 \cdot (y_1 - t_1) = 2 \cdot (0.37 - 1) = -1.26$$

$$\frac{\partial E}{\partial y_2} = 2 \cdot (y_2 - t_2) = 2 \cdot (0.53 - 1) = -0.94$$

$$\begin{aligned}
\frac{\partial E}{\partial p_1} &= \sum_j \frac{\partial E}{\partial y_j} \cdot \sigma'(y_{j,\text{in}}) \cdot w_{1,j} \\
&= \left(\frac{\partial E}{\partial y_1} \cdot \sigma'(y_{1,\text{in}}) \cdot w_{1,1}\right) + \left(\frac{\partial E}{\partial y_2} \cdot \sigma'(y_{2,\text{in}}) \cdot w_{1,2}\right) \\
&= \left(-1.26 \cdot \text{sig}'(-0.52) \cdot (-2)\right) + \left(-0.94 \cdot \text{sig}'(0.11) \cdot 0.43\right) \\
&= 0.5874 - 0.1 \\
&\approx 0.49
\end{aligned}$$

$$\begin{aligned}
\frac{\partial E}{\partial q_1} &= \sum_j \frac{\partial E}{\partial p_j} \cdot \sigma'(p_{j,\text{in}}) \cdot v_{1,j} \\
&= \frac{\partial E}{\partial p_1} \cdot \sigma'(p_{1,\text{in}}) \cdot v_{1,1} \\
&= 0.49 \cdot \text{sig}'(-1.03) \cdot (-0.25) \\
&\approx -0.02
\end{aligned}$$

Et cetera.

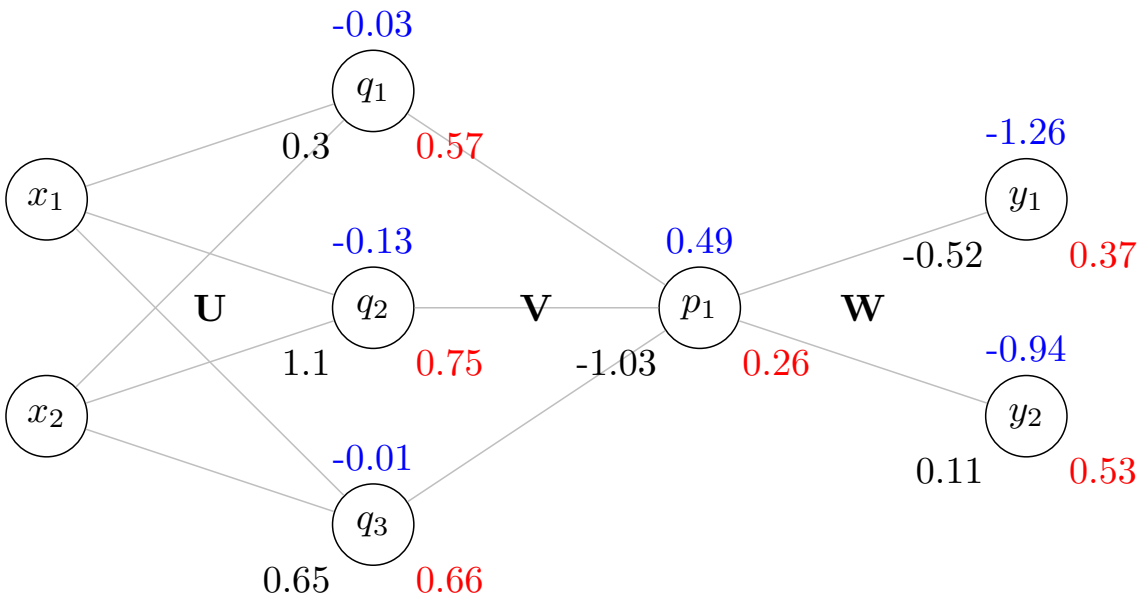

Figure 3: MLP after the forward propagation and the calculation of errors at neurons. Color coding: **pre-activation**, **activation**, **error derivative at neurons**

Backpropagation Step 2: Error Derivative at Weights

$$\begin{aligned}\frac{\partial E}{\partial w_{1,1}} &= \frac{\partial E}{\partial y_1} \cdot \sigma'(y_{1,\text{in}}) \cdot p_1 \\ &= -1.26 \cdot \text{sig}'(-0.52) \cdot 0.26 \\ &\approx -0.02\end{aligned}$$

$$\begin{aligned}\frac{\partial E}{\partial w_{1,2}} &= \frac{\partial E}{\partial y_2} \cdot \sigma'(y_{2,\text{in}}) \cdot p_1 \\ &= -0.94 \cdot \text{sig}'(0.11) \cdot 0.26 \\ &\approx -0.06\end{aligned}$$

$$\begin{aligned}\frac{\partial E}{\partial v_{1,1}} &= \frac{\partial E}{\partial p_1} \cdot \sigma'(p_{1,\text{in}}) \cdot q_1 \\ &= 0.49 \cdot \text{sig}'(-1.03) \cdot 0.57 \\ &\approx 0.05\end{aligned}$$

$$\frac{\partial E}{\partial v_{2,1}} \approx 0.07 \qquad \frac{\partial E}{\partial v_{3,1}} \approx 0.06$$

$$\begin{aligned}\frac{\partial E}{\partial u_{1,1}} &= \frac{\partial E}{\partial q_1} \cdot \sigma'(q_{1,\text{in}}) \cdot r_1 \\ &= -0.03 \cdot \text{sig}'(0.3) \cdot 0 \\ &= 0\end{aligned}$$

$$\frac{\partial E}{\partial u_{1,2}} = 0 \qquad \frac{\partial E}{\partial u_{1,3}} = 0 \qquad \frac{\partial E}{\partial u_{2,1}} \approx 0 \qquad \frac{\partial E}{\partial u_{2,2}} \approx -0.02 \qquad \frac{\partial E}{\partial u_{2,3}} \approx 0$$