THE HONG KONG UNIVERSITY OF SCIENCE & TECHNOLOGY

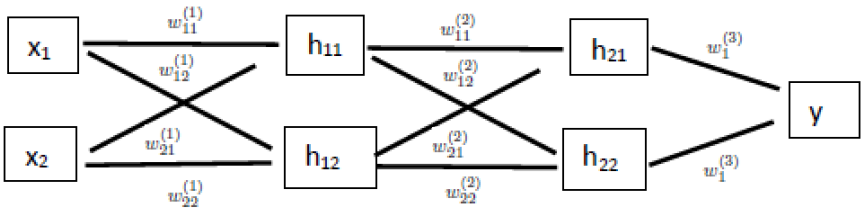
MSBD 5012: Machine Learning

Homework 3 Solutions

Assigned: 20/10/18 **Due Date: 10/11/18**

To submit your work, hand it to the instructor on the due date.

Question 1 Consider the following feedforward neural network with one input layer, two hidden layers, and one output layer. The hidden neurons are ReLU units, while the output neuron is a sigmoid unit.



The weights of the network and their initial values are as follows:

$$\text{Between input and first hidden:} \quad \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$\text{Between two hidden layers:} \quad \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} \end{bmatrix} = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$$

$$\text{Between second hidden and output:} \quad \begin{bmatrix} w_1^{(3)} \\ w_1^{(3)} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

For simplicity, assume the units do not have bias parameters. Let there be only one training example $(x_1, x_2, y) = (1, 2, 0)$.

(a) Consider feeding $(x_1, x_2) = (1, 2)$ to the network. What are the outputs of the hidden units? What is the logit $z = u_{21}w_1^{(3)} + u_{22}w_2^{(3)}$ calculated at the output unit? The output of the output unit is a probability distribution $p(y|x_1 = 1, x_2 = 2, \theta)$. What is the distribution?

(b) Next consider backpropagation. The loss function for the training example is $L = -\log p(y = 0|x_1 = 1, x_2 = 2, \theta)$. What is the error $\frac{\partial L}{\partial z}$ for the output unit? What are the errors for the hidden units? What are $\frac{\partial L}{\partial w_{22}^{(2)}}$ and $\frac{\partial L}{\partial w_{22}^{(1)}}$? If we want to reduce the loss on the example, should we increase or decrease the two parameters?

Solution: (a) Here are the outputs of the hidden units in forward propagation

$$\begin{bmatrix} x_1 = 1 \\ x_2 = 2 \end{bmatrix} \Rightarrow \begin{bmatrix} u_{11} = ReLU(z_{11}) = ReLU(x_1 w_{11}^{(1)} + x_2 w_{21}^{(1)}) = ReLU(1 - 2) = 0 \\ u_{12} = ReLU(z_{12}) = ReLU(x_1 w_{12}^{(1)} + x_2 w_{22}^{(1)}) = ReLU(-1 + 2) = 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} u_{21} = ReLU(z_{21}) = ReLU(u_{11} w_{11}^{(2)} + u_{12} w_{21}^{(2)}) = ReLU(0 + 1) = 1 \\ u_{22} = ReLU(z_{22}) = ReLU(u_{11} w_{12}^{(2)} + u_{12} w_{22}^{(2)}) = ReLU(0 + 1) = 1 \end{bmatrix}$$

Note that z_{11} is the net input of unit h_{11} and its values is -1 in this case. At the output unit y, we first compute the logit:

$$z_y = u_{21}w_1^{(3)} + u_{22}w_2^{(3)} = 2.$$

Hence, the output of the output unit is a the following probability distribution

$$p(y|x_1, x_2) = \sigma((2y - 1)z_y) = \sigma(2(2y - 1))$$

(b) According to page 20, L04, we have

$$\delta_y = \frac{\partial L}{\partial z_y} = -(y - \sigma(z_y)) = \sigma(2) \approx 0.88$$

Through backprop, we get errors for the hidden units:

$$\begin{bmatrix} \delta_{21} = \frac{\partial u_{21}}{\partial z_{21}}[\delta_y w_1^{(3)}] = 1 \times 0.88 = 0.88 \\ \delta_{22} = \frac{\partial u_{22}}{\partial z_{22}}[\delta_y w_2^{(3)}] = 1 \times 0.88 = 0.88 \end{bmatrix} \Leftarrow \delta_y = 0.88$$

$$\begin{bmatrix} \delta_{11} = \frac{\partial u_{11}}{\partial z_{11}}[\delta_{21} w_{11}^{(2)} + \delta_{22} w_{12}^{(2)}] = 0 \times (-1.76) = 0 \\ \delta_{12} = \frac{\partial u_{12}}{\partial z_{12}}[\delta_{21} w_{21}^{(2)} + \delta_{22} w_{22}^{(2)}] = 1 \times 1.76 = 1.76 \end{bmatrix} \Leftarrow$$

Consequently,

$$\begin{aligned} \frac{\partial L}{\partial w_{22}^{(2)}} &= u_{12}\delta_{22} = 1 \times 0.88 = 0.88 \\ \frac{\partial L}{\partial w_{22}^{(1)}} &= x_2\delta_{12} = 2 \times 1.76 = 3.52. \end{aligned}$$

Because the gradients are positive, we should increase the two parameters if we want to reduce the loss on the example.

Question 2: Why is the sigmoid activation function not recommended for hidden units, but it is fine for an output unit.

Solution: The sigmoid activation function $\sigma(z) = \frac{1}{1+exp(-z)}$ is not recommended for hidden units because it saturates across most of its domain. In other words, its derivative is usually small, preventing errors to back-propagate to units at previous layers.

It is fine for an output unit because of the use of negative log-likelihood (i.e., cross entropy) as the loss function. As such, we back-propagate the gradient of the **logarithm of a sigmoid function**, i.e., $-\log \sigma((2y-1)z)$, instead of the gradient of a sigmoid function itself. The function $-\log \sigma((2y-1)z)$ saturates only when the training example is classified correctly by the model.

Question 3: What is dropout used for in deep learning? How does it work? Why does it work? Answer briefly.

Solution: Dropout is regularization technique used in deep learning to avoid overfitting. It associates a binary mask variable with some of the units. During training, values for the mask variables are randomly sampled for each minibatch of data, and only parameters for the units with mask variable taking value 1 are oupdated on the minibatch. It reduces overfitting by preventing complex co-adaptation of parameters.

Question 4: What are the key ideas behind the Adam algorithm for training deep neural networks? Answer briefly.

Solution: There are three key ideas: The use of momentum to accelerate learning; Adaption of learning rate to slow down changes on parameters that have changed a lot before; The correction of bias in moment estimates.

Question 5: The input of a convolutional layer has shape $27 \times 27 \times 256$ (width, height, depth). The layer uses 384 3×3 filters applied at stride 1 with no zero padding. What is the shape of the output of the layer? How many parameters are there? How many float multiplication operations it will take to compute the net inputs of the all the output units?

Solution: The shape of the output layer is $W_2 \times H_2 \times D_2$ where

$$\begin{aligned} W_2 &= (W_1 - F + 2P)/S + 1 = (27 - 3 + 0)/1 + 1 = 25 \\ H_2 &= (H_1 - F + 2P)/S + 1 = (27 - 3 + 0)/1 + 1 = 25 \\ D_2 &= 384. \end{aligned}$$

The number of parameters is

$$(FFD_1 + 1)K = (3 \times 3 \times 256 + 1)384 = 885,120.$$

The number of float multiplication ops it takes to compute the net input of each output unit is FFD_1. The number of output units is $W_2H_2D_2$. Hence, the total number of ops is:

$$(3 \times 3 \times 256) \times (25 \times 25 \times 384) = 552,960,000$$

Question 6: (a) Is it a good idea to apply dropout to a convolutional neural network? If so, which part of the model should we apply dropout to? Answer briefly.

(b) Is it a good idea to apply dropout to a recurrent neural network? If so, which part of the model should we apply dropout to? Answer briefly.

Solution: (a) Most of the parameters of a CNN are in the FC layers. The excessive amount of parameters can lead to overfitting. So, it is a good idea to apply dropout there.

(b) It is not a good idea to dropout to RNN. The reason is that dropping any of the hidden units (the h's) would render the network disconnected, which makes forward and back propagations impossible.

Question 7 In an LSMT cell, $\mathbf{h}^{(t)}$ is computed from $\mathbf{h}^{(t-1)}$ and $\mathbf{x}^{(t)}$ using the following formulae:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_f\mathbf{x}^{(t)} + \mathbf{U}_f\mathbf{h}^{(t-1)} + \mathbf{b}_f) \\
\mathbf{i}_t &= \sigma(\mathbf{W}_i\mathbf{x}^{(t)} + \mathbf{U}_i\mathbf{h}^{(t-1)} + \mathbf{b}_i) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_q\mathbf{x}^{(t)} + \mathbf{U}_q\mathbf{h}^{(t-1)} + \mathbf{b}_q) \\
\mathbf{c}_t &= \mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes \tanh(\mathbf{U}\mathbf{x}^{(t)} + \mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{b}) \\
\mathbf{h}^{(t)} &= \mathbf{o}_t \otimes \tanh(\mathbf{c}_t)
\end{aligned}$$

(a) Intuitively, what are the functions of the forget gate $\mathbf{f}_t$ and the input gate $\mathbf{i}_t$ do? Answer briefly.

(b) Why do we use the sigmoid function for $\mathbf{f}_t$ and $\mathbf{i}_t$, but tanh for the memory cell $\mathbf{c}_t$ and the output $\mathbf{h}^{(t)}$? Answer briefly.

Solution: (a) The forget gate $\mathbf{f}_t$ determines which components of the previous state $\mathbf{c}_{t-1}$ and how much of them to remember/forget. The input gate $\mathbf{i}_t$ determines which components of the input from $\mathbf{h}^{(t-1)}$ and $\mathbf{x}^{(t)}$ and how much of them should go into the current state.

(b) The sigmoid function is used for $\mathbf{f}_t$ and $\mathbf{i}_t$ so that their values are likely to be close to 0 or 1, and hence mimicking the close and open of gates. The tanh function is used for $\mathbf{c}_t$ and $\mathbf{h}^{(t)}$ so that strong gradient signals can be backpropagated from $\mathbf{h}^{(t)}$ to $\mathbf{c}_t$, and then to $\mathbf{h}^{(t-1)}$.