Deep Learning

Theoretical Exercises – Week 9 – Chapter 7

Exercises on the book "Deep Learning" written by Ian Goodfellow,
Yoshua Bengio, and Aaron Courville.
Exercises and solutions by T. Méndez and G. Schuster

FS 2024

1 Exercises on Regularization for Deep Learning

1. Describe the following regularization methods in your own words by explaining the idea behind the method and describing how it acts as a regularizer.

   (a) Early Stopping

   (b) Bagging

   (c) Dropout

   Solution:

   (a) **Early Stopping**: The idea behind early stopping is to stop training at the time when the generalization error is lowest. In order to find this point in time, the generalization error has to be estimated with a validation set. Hence, some labeled data can not be used for training. Also the model (parameters of the network) that generated the lowest generalization error has to be saved.

   (b) **Bagging**: With bagging the generalization error is reduced by training different models separately and then averaging their outputs of the test examples. The main reason why this works is, that different models will make different, and ideally uncorrelated, errors. Hence, averaging model responses will reduce the variance of the errors. However, for deep neural networks this is a very expensive (both in terms of time and computing resources) regularization method, since different deep models need to be trained.

   (c) **Dropout**: Dropout is a very clever and efficient regularizer that combines the two methods of *noise robustness* and *bagging*. By randomly dropping hidden units, different networks are generated and trained, which all share parameters. This approximately leads to the averaging of exponentially many networks. In addition, each hidden unit must be able to perform well, regardless of which other hidden units are in the model. Dropout thus regularizes each hidden unit to be not merely a good feature but a feature that is good in many contexts (it is robust against noise).

2. Given is the network in Figure 1, which has the following weights and biases:

$$\boldsymbol{W}^{(1)} = \begin{bmatrix} -0.4 & 0.1 & 0.9 \\ 0.8 & -0.2 & -0.7 \end{bmatrix}, \qquad \boldsymbol{b}^{(1)} = \begin{bmatrix} 0.6 & -0.4 & -0.7 \end{bmatrix}$$

and

$$\boldsymbol{w}^{(2)} = \begin{bmatrix} -0.9 \\ -0.8 \\ 0.6 \end{bmatrix}, \qquad b^{(2)} = 0.4.$$

In addition, an 4×2 matrix

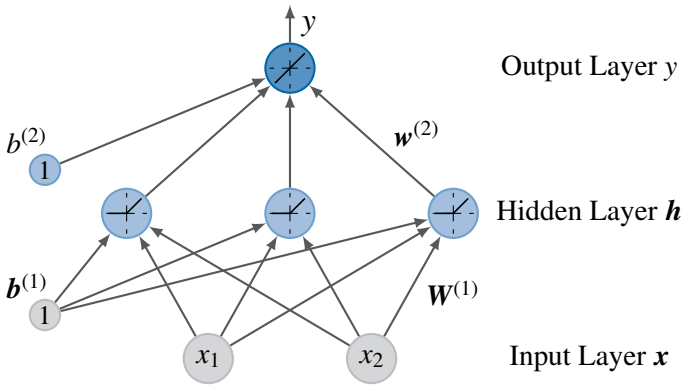

Figure 1: A neural network with two input units, three hidden units and one output. The activation function in the output layer is linear and in the hidden layer a rectified linear unit (ReLu) is used.

$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

is given, where line i contains the input example $\boldsymbol{x}^{(i)}$.

(a) Calculate the mean value of the output y by using the given training set $\boldsymbol{X}$ as input.

(b) The hidden units are now dropped with the probability of $p = 0.6$. Recalculate the mean value of the output y by using the given training set $\boldsymbol{X}$ as input and the networks modified by dropout. Calculate the mean value for each modification given in the matrix

$$\boldsymbol{D} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix},$$

where each line corresponds to one network. 0 means that the corresponding hidden unit is dropped and 1 means that it is kept. Then average all calculated mean values and compare the result with the one of task (a).

Hint:
Remember to adjust the weights accordingly when using dropout.

Solution:

(a) The output vector $\boldsymbol{y}$ is calculated as

$$\begin{aligned}\boldsymbol{y} &= \boldsymbol{H} \cdot \boldsymbol{w}^{(2)} + b^{(2)} \\ &= \begin{bmatrix} 0.6 & 0.0 & 0.0 \\ 1.4 & 0.0 & 0.0 \\ 0.2 & 0.0 & 0.2 \\ 1.0 & 0.0 & 0.0 \end{bmatrix} \cdot \begin{bmatrix} -0.9 \\ -0.8 \\ 0.6 \end{bmatrix} + 0.4 = \begin{bmatrix} -0.14 \\ -0.86 \\ 0.34 \\ -0.50 \end{bmatrix},\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{H} &= \max\left\{0, \boldsymbol{X} \cdot \boldsymbol{W}^{(1)} + \boldsymbol{b}^{(1)}\right\} \\ &= \max\left\{0, \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} -0.4 & 0.1 & 0.9 \\ 0.8 & -0.2 & -0.7 \end{bmatrix} + \begin{bmatrix} 0.6 & -0.4 & -0.7 \end{bmatrix}\right\} \\ &= \max\left\{0, \begin{bmatrix} 0.6 & -0.4 & -0.7 \\ 1.4 & -0.6 & -1.4 \\ 0.2 & -0.3 & 0.2 \\ 1.0 & -0.5 & -0.5 \end{bmatrix}\right\} = \begin{bmatrix} 0.6 & 0.0 & 0.0 \\ 1.4 & 0.0 & 0.0 \\ 0.2 & 0.0 & 0.2 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}.\end{aligned}$$

Thus, the mean of the output vector y is

$$\mu = \frac{-0.14 - 0.86 + 0.34 - 0.5}{4} = -0.29.$$

(b) For the modified networks, the first layer does not change, thus, $\boldsymbol{H}$ remains the same. For the calculation of the output vector $\boldsymbol{y}$, however, the weights $\boldsymbol{w}^{(2)}$ have to be adjusted according to the weight scaling inference rule to

$$\tilde{\boldsymbol{w}}^{(2)} = \frac{1}{1-p} \cdot \boldsymbol{w}^{(2)} = \frac{1}{1-0.6} \begin{bmatrix} -0.9 \\ -0.8 \\ 0.6 \end{bmatrix} = \begin{bmatrix} -2.25 \\ -2.00 \\ 1.50 \end{bmatrix}$$

and the outputs of the hidden units have to be pointwise multiplied by the dropout-mask $\boldsymbol{D}_{i,:}$ (ith row of matrix $\boldsymbol{D}$). Thus, the output vector of the ith network is calculated as

$$\boldsymbol{y}_i = (\boldsymbol{H} \odot \boldsymbol{D}_{i,:}) \cdot \tilde{\boldsymbol{w}}^{(2)} + b^{(2)}.$$

If this calculation is performed for all five networks, the following output vectors

result

$$
\begin{aligned}
\boldsymbol{y}_1 &= \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.2 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \cdot \begin{bmatrix} -2.25 \\ -2.00 \\ 1.50 \end{bmatrix} + 0.4 = \begin{bmatrix} 0.4 \\ 0.4 \\ 0.7 \\ 0.4 \end{bmatrix}, \\
\boldsymbol{y}_2 &= \begin{bmatrix} 0.6 & 0.0 & 0.0 \\ 1.4 & 0.0 & 0.0 \\ 0.2 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \end{bmatrix} \cdot \begin{bmatrix} -2.25 \\ -2.00 \\ 1.50 \end{bmatrix} + 0.4 = \begin{bmatrix} -0.95 \\ -2.75 \\ -0.05 \\ -1.85 \end{bmatrix}, \\
\boldsymbol{y}_3 &= \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \cdot \begin{bmatrix} -2.25 \\ -2.00 \\ 1.50 \end{bmatrix} + 0.4 = \begin{bmatrix} 0.4 \\ 0.4 \\ 0.4 \\ 0.4 \end{bmatrix}, \\
\boldsymbol{y}_4 &= \boldsymbol{y}_2, \\
\boldsymbol{y}_5 &= \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.2 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \cdot \begin{bmatrix} -2.25 \\ -2.00 \\ 1.50 \end{bmatrix} + 0.4 = \begin{bmatrix} 0.4 \\ 0.4 \\ 0.7 \\ 0.4 \end{bmatrix},
\end{aligned}
\tag{1.1}
$$

and the corresponding mean values are

$$
\begin{aligned}
\mu_1 &= 0.475, \\
\mu_2 &= -1.4, \\
\mu_3 &= 0.4, \\
\mu_4 &= -1.4, \\
\mu_5 &= 0.475.
\end{aligned}
$$

If these mean values are averaged again, the result is the same as in task (a)

$$
\mu = \frac{0.475 - 1.4 + 0.4 - 1.4 + 0.475}{5} = -0.29.
$$

This result was to be expected due to the applied weight scaling inference rule, the linear output activation function and the shallow network with only one hidden layer.