COURSE 02457

Non-Linear Signal Processing: Quiz Exercise 5

C.M. Bishop: Pattern Recognition and Machine Learning, Sections 5.1-5.4

Questions

1. How can we make a weight-initialization range suitable for all data?

   (a) This cannot be done

   (b) By appropriately scaling data sets

   (c) By changing the optimization parameters

   (d) By re-starting the algorithm with random weight initializations several times

2. How many layers in the neural network shown in Figure 1 in Exercise 5 contain weights that are optimized during training?

   (a) 1

   (b) 2

   (c) 3

   (d) 4

3. What is an appropriate cost function for training a neural network if the response variable t is binary? Let t_n denote the response (target) for the nth observation and y_n be the prediction for the nth observation.

   (a) $\sum_{n=1}^{N}(t_n - y_n)^2$

   (b) $\frac{1}{2}\sum_{n=1}^{N}(t_n - y_n)^2$

   (c) $-\sum_{n=1}^{N}(t_n \ln(y_n) + (1 - t_n) \ln(1 - y_n))$

   (d) $-\prod_{n=1}^{N}(t_n \ln(y_n) + (1 - t_n) \ln(1 - y_n))$

4. What is the advantage of diagonal approximation of the Hessian of the error function rather than exact evaluation?

   (a) Diagonal approximation is faster and the method can be altered to give the exact diagonal values without increasing the run time.

   (b) Diagonal approximation is more accurate due to higher numerical stability.

   (c) Diagonal approximation is faster and a close approximation since the true Hessian is often nearly diagonal.

   (d) Diagonal approximation is faster and sometimes the inverse rather than the Hessian itself is needed, making a diagonal approximation desirable.

5. Neural networks can be used for

(a) Continuous response variables (output) only.

(b) Binary response variables only.

(c) Binary and continuous response variable.

(d) Continuous, binary, and multi-category response variables.

6. Consider a fully connected two-layer neural network (one hidden layer) with M hidden units, all with $\tanh(\cdot)$ as activation function. What is the minimum number of weight vectors that give rise to the same output when the same input is given? That is, what is the weight-space symmetry factor of this neural network?

(a) $M! \cdot 2^M$

(b) $M!$

(c) M^2

(d) $M! \cdot M^2$

7. Figure 1 shows a fully connected two-layer neural network with $\tanh(\cdot)$ as activation function. We consider the weight vector $\mathbf{w}^* = (w_1^{(1)}, w_2^{(1)}, w_3^{(1)}, w_4^{(1)}, w_5^{(1)}, w_6^{(1)}, w_1^{(2)}, w_2^{(2)})$. Which of the following weight vectors produce outputs of the neural network identical to $\mathbf{w}^*$?

I) $(-w_1^{(1)}, w_2^{(1)}, -w_3^{(1)}, w_4^{(1)}, -w_5^{(1)}, w_6^{(1)}, -w_1^{(2)}, w_2^{(2)})$

II) $(-w_1^{(1)}, w_2^{(1)}, -w_3^{(1)}, w_4^{(1)}, -w_5^{(1)}, w_6^{(1)}, -w_1^{(2)}, -w_2^{(2)})$

III) $(-w_2^{(1)}, w_1^{(1)}, -w_4^{(1)}, w_3^{(1)}, -w_6^{(1)}, w_5^{(1)}, -w_2^{(2)}, w_1^{(2)})$

IV) $(-w_2^{(1)}, w_1^{(1)}, -w_4^{(1)}, w_3^{(1)}, -w_6^{(1)}, w_5^{(1)}, -w_1^{(2)}, w_2^{(2)})$

(a) I), IV)

(b) II), III), IV)

(c) I), III)

(d) II), IV)

8. Which function of the inputs x_1, x_2, and x_3 does the three-layer neural network in Figure 2 represent?

(a) $\sigma\left(\sum_{k=1}^{2} w_k^{(3)} \sigma\left(\sum_{j=1}^{2} w_{kj}^{(2)} \tanh\left(\sum_{i=1}^{3} w_{ji}^{(1)} x_i\right)\right)\right)$

(b) $\sigma\left(\sum_{k=1}^{2} w_k^{(3)} \tanh\left(\sum_{j=1}^{4} w_{kj}^{(2)} \sigma\left(\sum_{i=1}^{6} w_{ji}^{(1)} x_i\right)\right)\right)$

(c) $\sigma\left(\sum_{k=1}^{2} w_k^{(3)} \tanh\left(\sum_{j=1}^{2} w_{kj}^{(2)} \sigma\left(\sum_{i=1}^{3} w_{ji}^{(1)} x_i\right)\right)\right)$

(d) $\tanh\left(\sum_{k=1}^{2} w_k^{(3)} \sigma\left(\sum_{j=1}^{2} w_{kj}^{(2)} \sigma\left(\sum_{i=1}^{3} w_{ji}^{(1)} x_i\right)\right)\right)$

9. Choose the correct ending of the following sentence: When training a neural network, the training error at the end of training is

(a) sometimes minimized as much as possible

(b) always minimized as much as possible if we have more data points than ten times the number of explanatory variables

(c) never minimized as much as possible

(d) always minimized as much as possible

10. Which of the following statements are true of backpropagation?

I) Backpropagation is only useful when the error function considered is the sum-of-squares

II) Backpropagation can be used only in neural networks

III) Backpropagation can be used to evaluate the Jacobian and Hessian matrices

(a) I)

(b) II)

(c) III)

(d) I), II)

11. Consider the two-layer neural network in Figure 3 with one hidden layer containing one unit with $\tanh(\cdot)$ as activation function. The activation function for the output unit is the identity. Considering this as a regression problem, we use the cost function $E(\mathbf{W}) = (y-t)^2$ for a prediction y with target t. Give the correct two quantities, $\delta_2 = y - t$ and $\delta_1 = \frac{\partial E}{\partial a_1}$.

(a) $\delta_2 = w_{11}^{(2)} \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}) - t$,
$\delta_1 = \left(1 - \tanh^2(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})\right) w_{11}^{(2)} \left(w_{11}^{(2)} \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}) - t\right)$

(b) $\delta_2 = w_{11}^{(2)} \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}) - t$,
$\delta_1 = \left(1 - \tanh^2(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})\right) w_{11}^{(2)} \left(\tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})\right)$

(c) $\delta_2 = \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})$,
$\delta_1 = \left(1 - \tanh^2(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})\right) w_{11}^{(2)} \left(w_{11}^{(2)} \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}) - t\right)$

(d) $\delta_2 = \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})$,
$\delta_1 = \left(1 - \tanh^2(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})\right) w_{11}^{(2)} \left(\tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})\right)$

12. Consider again the two-layer neural network in Figure 3 with one hidden layer containing one unit with $\tanh(\cdot)$ as activation function. The activation function for the output unit is the identity. Considering this as a regression problem, we use the cost function $E(\mathbf{W}) = (y-t)^2$ for a prediction y with true target t. Give the two derivatives $\frac{\partial E(\mathbf{W})}{\partial w_{11}^{(2)}}$ and $\frac{\partial E(\mathbf{W})}{\partial w_{11}^{(1)}}$.

(a) $\frac{\partial E(\mathbf{W})}{\partial w_{11}^{(2)}} = \left(w_{11}^{(2)} \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}) - t\right) \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})$,
$\frac{\partial E(\mathbf{W})}{\partial w_{11}^{(1)}} = \left(1 - \tanh^2(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})\right) x_1$

(b) $\frac{\partial E(\mathbf{W})}{\partial w_{11}^{(2)}} = \left(w_{11}^{(2)} \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}) - t\right) \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}),$
$\frac{\partial E(\mathbf{W})}{\partial w_{11}^{(1)}} = \left(1 - \tanh^2(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})\right) \left(w_{11}^{(2)} \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}) - t\right) x_1$

(c) $\frac{\partial E(\mathbf{W})}{\partial w_{11}^{(2)}} = \left(w_{11}^{(2)} \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})\right) \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}),$
$\frac{\partial E(\mathbf{W})}{\partial w_{11}^{(1)}} = \left(1 - \tanh^2(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})\right) \left(w_{11}^{(2)} \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}) - t\right) x_1$

(d) $\frac{\partial E(\mathbf{W})}{\partial w_{11}^{(2)}} = \left(w_{11}^{(2)} \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})\right) \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}),$
$\frac{\partial E(\mathbf{W})}{\partial w_{11}^{(1)}} = \left(1 - \tanh^2(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})\right) x_1$

13. **Challenge question.** Consider a fully connected two-layer neural network (one hidden layer) with M hidden units, all with the identity as activation function. What is the minimum number of weight vectors that give rise to the same output when the same input is given? That is, what is the weight-space symmetry factor of this neural network?

    (a) $M! \cdot 2^M$

    (b) $M!$

    (c) M^2

    (d) $M! \cdot M^2$

14. **Challenge question.** Consider a fully connected two-layer neural network (one hidden layer) with M hidden units, all with $\exp(\cdot)$ as activation function. What is the minimum number of weight vectors that give rise to the same output when the same input is given? That is, what is the weight-space symmetry factor of this neural network?

    (a) $M! \cdot 2^M$

    (b) $M!$

    (c) M^2

    (d) $M! \cdot M^2$

15. **Challenge question.** Consider a fully connected three-layer neural network (two hidden layers). In the first hidden layer, there are M hidden units, all with $\exp(\cdot)$ as activation function. In the second hidden layer, there are N hidden units, all with $\tanh(\cdot)$ as activation function. What is the minimum number of weight vectors that give rise to the same output when the same input is given? That is, what is the weight-space symmetry factor of this neural network?

    (a) $M! \cdot 2^M \cdot N! \cdot 2^N$

    (b) $(M+N)! \cdot 2^{M+N}$

    (c) $2^M \cdot N! \cdot 2^N$

    (d) $M! \cdot N! \cdot 2^N$

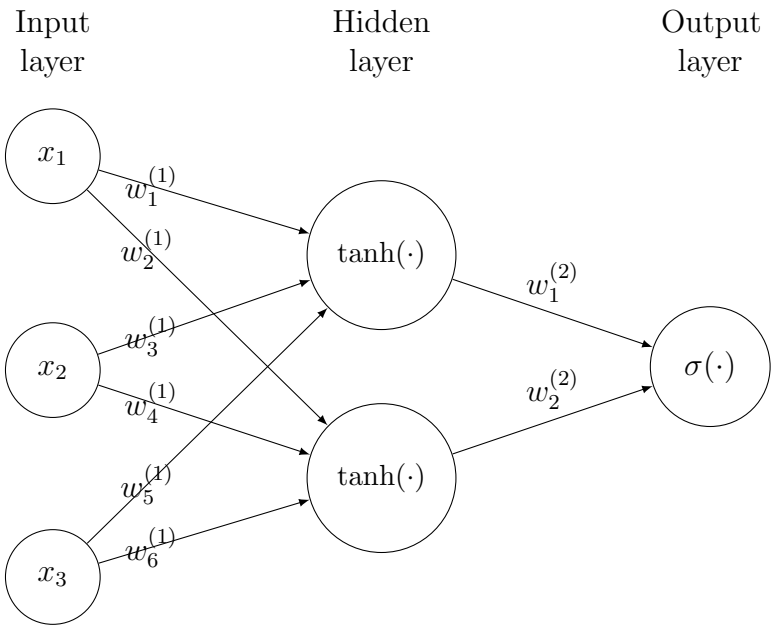

Figure 1: Two-layer neural network.

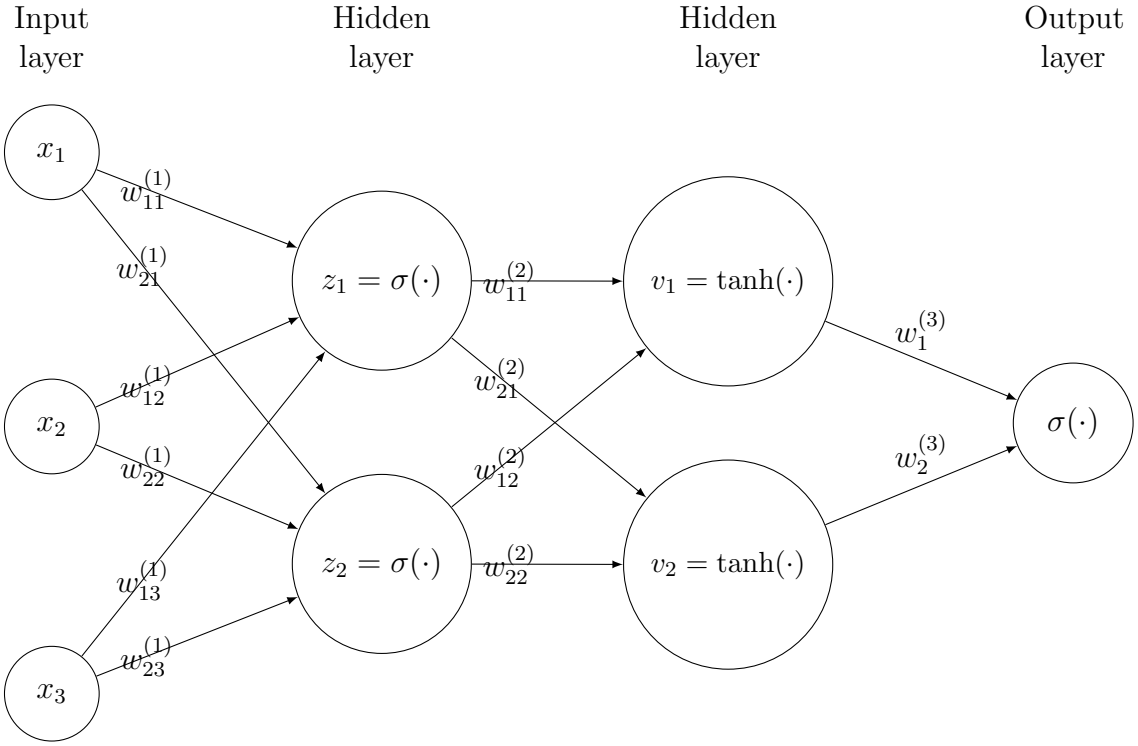

Figure 2: Three-layer neural network.

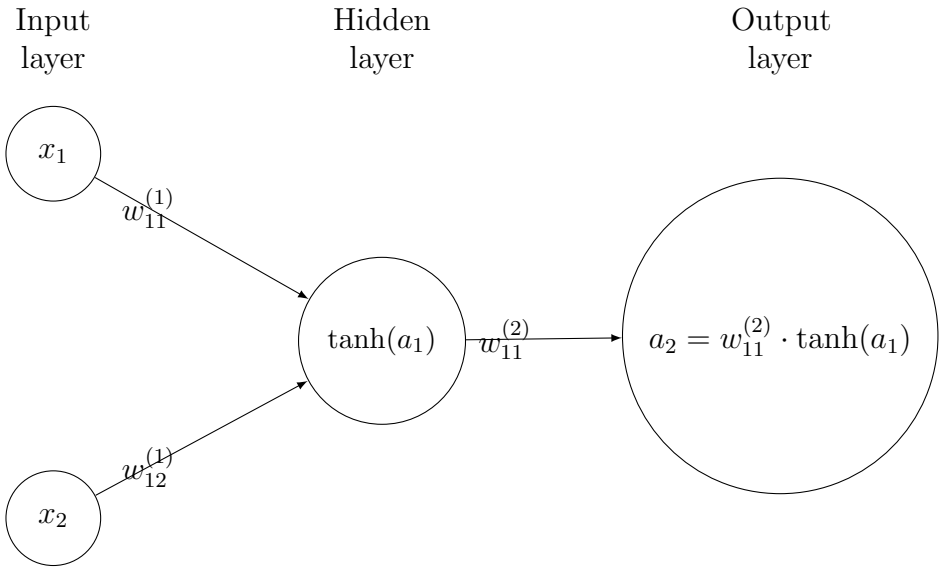

Figure 3: Two-layer neural network.

Hint - reading for each question

1. Exercise check point 5.1
2. Section 5.1 in Bishop
3. Section 5.2
4. Section 5.4
5. Section 5.2
6. Section 5.1.1
7. Section 5.1.1
8. Section 5.1
9. Section 5.2
10. Section 5.3
11. Section 5.3
12. Section 5.3
13. Section 5.1.1
14. Section 5.1.1
15. Section 5.1.1

Correct answers

1. (b) By scaling the explanatory variables in the data and controlling the range for initial weights, we can ensure that we stay in the non-linear part of the activation functions initially.
2. (b) The weights for the linear combinations of the explanatory variables in the hidden layer and the weights for the linear combinations of hidden unit outputs in the output layer are optimized during training. Hence two layers contain weights that are optimized during training.
3. (c) To derive the appropriate cost function, we note that each observation is assumed independent from all others. This allows us to write $p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} p(t_n|\mathbf{x}_n, \mathbf{w})$. Since the target variable is binary, it is appropriate to use the Bernouilli distribution to model it. Hence we have $p(t_n|\mathbf{x}_n, \mathbf{w}) = y_n^{t_n}(1 - y_n)^{(1-t_n)}$, where y_n is the probability that t_n is from class one ($y_n = p(t_n = 1) = \sigma(\mathbf{w}^T\mathbf{x}_n)$), and we make t_n take on the values zero and

one. To get an appropriate loss function, we take the negative logarithm of the likelihood function and get:

$$
\begin{aligned}
-\ln\left(p(\mathbf{t}|\mathbf{X},\mathbf{w})\right) &= -\ln\left(\prod_{n=1}^{N} p(t_n|\mathbf{x}_n,\mathbf{w})\right) = -\sum_{n=1}^{N}\ln(p(t_n|\mathbf{x}_n,\mathbf{w})) \\
&= -\sum_{n=1}^{N}\ln(p(t_n|\mathbf{x}_n,\mathbf{w})) = -\sum_{n=1}^{N}\ln(y_n^{t_n}(1-y_n)^{(1-t_n)}) \\
&= -\sum_{n=1}^{N}(t_n\ln(y_n)+(1-t_n)\ln(1-y_n)))
\end{aligned}
$$

4. (d) It is faster to invert a diagonal matrix than an arbitrary matrix since the inverse of a diagonal matrix is just another diagonal matrix with the diagonal elements inverted.

5. (d) By choosing appropriate activation functions for the output units, neural networks can be used for both continous, binary, and multi-class response variables.

6. (a) Since $\tanh(\cdot)$ is an odd function, the signs of all weights coming in to a unit can be flipped if the sign of the weight coming out of the unit is also flipped. This makes 2^M different weight vectors result in the same output. Likewise, the order of the hidden layer units is arbitrary. Since there are $M!$ permutations of M units, at least $M!2^M$ weight vectors give the same output upon the same input. The number can be higher for some weight vectors due to coincidental symmetries caued by particular weight values.

7. (c) The weight vector in I) has the signs of the weights coming in to and going out from the upper hidden unit flipped. Hence this weight vector does not change the neural network. The weight vector in II) has the same signs flipped as that in I), but also the sign of the weight going out from the lower hidden unit. Hence this weight vector does change the neural network. In the weight vector in III), the order of the hidden units has been switched along with the appropriate weights, so that the linear combination represented by the network is unchanged. Also, the signs of the weights coming in to and going out from the upper hidden unit (after the switched order) have been flipped. Hence this weight vector does not change the neural network. In the weight vector in IV), the weights from the input layer to the hidden layer are the same as in III). However, the weights from the hidden layer to the output layer are not switched, and the sign of the weight from the upper hidden unit is flipped. Hence this weight vector does change the neural network.

8. (c) The first hidden layer has the activation function $\sigma(\cdot)$, the second hidden layer has the activation function $\tanh(\cdot)$, and the output is transformed using $\sigma(\cdot)$. Hence the function represented by the neural network must be of the form $\sigma(\tanh(\sigma(\cdot)))$. This leaves only b) and c) as options. By just inspecting the indices over which the sums are taken, we see that c) must be correct since there are only three, not six, inputs, and only two, not four, hidden units.

9. (a) Since the cost function of a neural network is non-convex, the training error at the end of training may be at a local minimum instead of a global minimum. However, the training error could also be at the global minimum, but we have no way to know this. Hence a) is the correct answer.

10. (c) Backpropagation can be used to evaluate derivatives for many different cost functions, and in other settings than neural networks. The principle of backpropagation can also be used to evaluate Jacobian and Hessian matrices. Hence c) is correct.

11. (a) We use equations (5.54) and (5.56) in the textbook. Since the prediction from the network is $y = w_{11}^{(2)} \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)})$, we have that $\delta_2 = w_{11}^{(2)} \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}) - t$ from equation (5.54). Using equation (5.56), we find $\delta_1 = (1 - \tanh^2(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}))w_{11}^{(2)}(w_{11}^{(2)} \tanh(x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)}) - t)$.

12. (b) The two requested derivatives can be found using equation (5.53) in the textbook and the two quantities from the previous question, δ_1 and δ_2. Using these, we arrive at the answer.

13. (a) Since the identity function is also an odd function, the answer is the same as for question 6.

14. (b) When $\exp(\cdot)$ is used as the activation function, symmetry only arises due to permutations of the order of the hidden units since $\exp(\cdot)$ is an even funcion such that the signs of weights cannot be flipped. Thus the weight-space symmetry is "only" $M!$ in this case.

15. (d) For the first hidden layer, the weight-space symmetry is $M!$, as argued in the above answer. In the second hidden layer, the weight-space symmetry is $N!2^N$ as argued in the answer for question 6. Hence the total weight-space symetry is $M!N!2^N$.

DTU, September 2013,

Laura Frølich and Ole Winther