Written Assignment 3

Marit Hoefsloot

November 4th, 2016

1 Exercise 3

We have a Neural Network with 1 hidden layer. Input and hidden layer both have 2 nodes. There is 1 output node. The values of theta for the bias nodes (θ_0) are 0.2. The vector $\boldsymbol{\theta^{(1)}}$ is $[0.5, 0.1, 0.5, 0.7]$ and $\boldsymbol{\theta^{(2)}}$ is $[1, 2]$.

1. Calculate by hand the activations of all nodes for $x_1 = 0.5$ and $x_2 = 0.9$.

Figure 1 provides a visualization of the neural network as described above. I choose $\boldsymbol{\theta^{(1)}}$ as can be seen in table 1. I devided the values like this because we want the sigmoid function $g(z) = \frac{1}{1+e^{-z}}$ to be as close to 1, so $1 + e^{-z}$ has to be as close to 1 as possible, so e^{-z} has to be as small as possible and z has to be as big as possible. With these values, we get the largest values as we later compute $z = ... + 0.7 \cdot 0.9 + ...$ instead of, for example, $z = ... + 0.1 \cdot 0.9 + ...$. I want the biggest $\theta_i^{(1)}$ to go to node a_2, because from there the weight to the last node is the biggest, namely 2. And we take $x_0 = 1$ and $a_0 = 1$.

$\boldsymbol{\theta^{(1)}}$	Value
$\theta_1^{(1)}$	0.1
$\theta_2^{(1)}$	0.5
$\theta_3^{(1)}$	0.5
$\theta_4^{(1)}$	0.7

Table 1: The values of $\boldsymbol{\theta^{(1)}}$

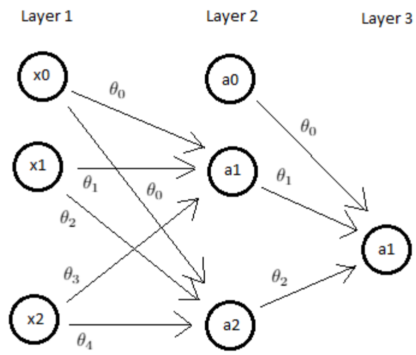

Figure 1: Visualization of the neural network

Now we compute the activations of all nodes.

$$
\begin{aligned}
a_1^{(2)} &= g(\theta_0^{(1)} \cdot x_0 + \theta_1^{(1)} \cdot x_1 + \theta_3^{(1)} \cdot x_2) \\
&= g(0.2 + 0.1 \cdot 0.5 + 0.5 \cdot 0.9) \\
&= \frac{1}{1 + e^{-0.7}} \approx 0.6682 \\
a_2^{(2)} &= g(\theta_0^{(1)} \cdot x_0 + \theta_2^{(1)} \cdot x_1 + \theta_4^{(1)} \cdot x_2) \\
&= g(0.2 + 0.5 \cdot 0.5 + 0.7 \cdot 0.9) \\
&= \frac{1}{1 + e^{-1.08}} \approx 0.7466 \\
a_1^{(3)} &= g(\theta_0^{(2)} \cdot a_0 + \theta_1^{(2)} \cdot a_1 + \theta_2^{(2)} \cdot a_2) \\
&= g(0.2 + 1 \cdot 0.6682 + 2 \cdot 0.7466) \\
&= \frac{1}{1 + e^{-2.3814}} \approx 0.9154
\end{aligned}
$$

2. Suppose the correct output is 1. Calculate the errors for all nodes and the updates of the weights (for 1 iteration).

The error of node j in layer l is $\delta_j^{(l)}$. We look at the errors of the layers from right to left. We calculate the error in the last node with:

$$
\begin{aligned}
\delta_1^{(3)} &= a_1^{(3)} - y \\
&= 0.9154 - 1 = 0.0846
\end{aligned}
$$

In the second layer, we use:

$$
\begin{aligned}
\delta_1^{(2)} &= \theta_1^{(2)} \cdot \delta_1^{(3)} \cdot g'(z_1^{(2)}) \\
&= \theta_1^{(2)} \cdot \delta_1^{(3)} \cdot a_1^{(2)} \cdot (1 - a_1^{(2)}) \\
&= 1 \cdot 0.0846 \cdot 0.6682 \cdot (1 - 0.6682) \\
&\approx 0.01876 \\
\delta_2^{(2)} &= \theta_2^{(2)} \cdot \delta_1^{(3)} \cdot g'(z_2^{(2)}) \\
&= \theta_2^{(2)} \cdot \delta_1^{(3)} \cdot a_2^{(2)} \cdot (1 - a_2^{(2)}) \\
&= 2 \cdot 0.0846 \cdot 0.7466 \cdot (1 - 0.7466) \\
&\approx 0.03201
\end{aligned}
$$

To update the weights, I added the error to the different theta's. This makes the theta bigger and would mean that the activation for the node in the third layer gets closer to 1. The activation becomes:

$$
\begin{aligned}
a_1^{(3)} &= g(\theta_0^{(2)} \cdot a_0 + (\theta_1^{(2)} + \delta_1^{(2)}) \cdot a_1 + (\theta_2^{(2)} + \delta_2^{(2)}) \cdot a_2) \\
&= g(0.2 + (1 + 0.01876) \cdot 0.6682 + (2 + 0.03201) \cdot 0.7466) \\
&= \frac{1}{1 + e^{-2.3978}} \approx 0.9167
\end{aligned}
$$

2 Exercise 4.1

In this exercise we will use a perceptron (instead of sigmoid function), that is explained in the slides of Mitchell on pages 78 and 79. Please refer to figure 2.

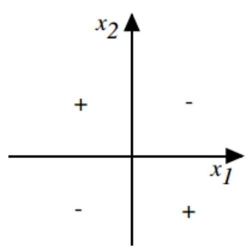

Figure 2: Perceptron from the slides of Mitchell, page 79

1. What are the values of weights w_0, w_1 and w_2 for the perceptron whose decision surface is illustrated in figure 2? Assume the surface crosses the x_1 axis at -1 and the x_2 axis at 2.

The formula for the perceptron is:

$$
o(x_1, ..., x_n) = \left\{ \begin{array}{cc} 1 & \text{if } w_0 + w_1 x_1 + ... + w_n x_n > 0 \\ \text{-1} & \text{otherwise} \end{array} \right\}
$$

We have a two-dimensional line, so we use the equation $w_0+w_1x_1+w_2x_2 = 0$, with the points $(-1, 0)$ and $(0, 2)$ on the line. We rewrite the formula for the line as:

$$\begin{aligned} w_0 + w_1x_1 + w_2x_2 &= 0 \\ w_2x_2 &= -w_1x_1 - w_0 \\ x_2 &= \frac{-w_1x_1}{w_2} - \frac{w_0}{w_2} \end{aligned}$$

Now we calculate the slope of the line $\frac{-w_1}{w_2}$:

$$\begin{aligned} \frac{-w_1}{w_2} &= \frac{(x_2)_2 - (x_2)_1}{(x_1)_2 - (x_1)_1} \\ &= \frac{2-0}{0-(-1)} = \frac{2}{1} = 2 \end{aligned}$$

So $-w_1 = 2w_2$. We now fill in the coordinates for one point on the line and solve for $\frac{w_0}{w_2}$:

$$\begin{aligned} x_2 &= \frac{-w_1x_1}{w_2} - \frac{w_0}{w_2} \\ 0 &= 2 \cdot -1 - \frac{w_0}{w_2} \\ \frac{w_0}{w_2} &= -2 \\ w_0 &= -2w_2 \\ w_0 &= w_1 = -2w_2 \end{aligned}$$

So, one possible combination of weights is: $w_0 = -2$, $w_1 = -2$ and $w_2 = 1$.

3 Exercise 4.2

1. Design a two-input perceptron that implements the Boolean function A AND (NOT B), in other words for: $A \cap \neg B$.

This question is quite similar to the last one, so we go through the same steps.

All possible outputs for the perceptron are given in table 2. The figure that goes with such a perceptron can be found in figure 3. The line in the figure goes through $(1, 0)$ and $(0, -1)$.

A	B	$o(A \cap \neg B)$
-1	-1	-1
-1	1	-1
1	-1	1
1	1	-1

Table 2: The truth table of the perceptron for $A \cap \neg B$

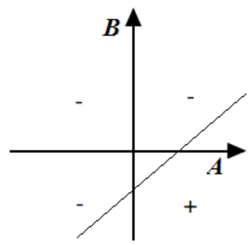

Figure 3: Perceptron for $A \cap \neg B$

To find the formula for the line in figure 3, we set up $B = mA + c$, and try to find m and c. We first calculate the slope by solving for m:

$$\begin{aligned} m &= \frac{B_2 - B_1}{A_2 - A_1} \\ &= \frac{0 - (-1)}{1 - 0} = \frac{1}{1} = 1 \end{aligned}$$

Now we fill in one point on the line, namely $(1, 0)$:

$$\begin{aligned} B &= mA + c \\ 0 &= 1 \cdot 1 + c \\ c &= -1 \end{aligned}$$

So the formula for the line is $B = A - 1$, which can be written as: $1 - A + B = 0$. Comparing this to $w_0 + w_1x_1 + w_2x_2 = 0$, we can see that a possible combination for the weights of this perceptron is: $w_0 = -1$, $w_1 = 1$ and $w_2 = -1$.

2. Design a two-layer network of perceptrons that that implements A XOR B.

The truth table for the exclusive OR function can be found in table 3.

A	B	o(A XOR B)
-1	-1	1
-1	1	-1
1	-1	1
1	1	-1

Table 3: The truth table of the perceptron for the exclusive OR

We can rewrite A XOR B as: $(A \cap \neg B) \cup (\neg A \cap B)$. From this we can design a two-layer perceptron using the perceptron we described in Exercise 4.2 twice. This would look like figure 4 in the situation where $A = 1$ and $B = -1$.

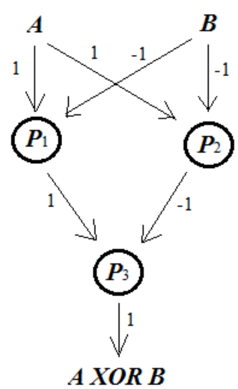

Figure 4: Two-layered perceptron

Where P_1 is the perceptron as described in Exercise 4.2. P_2 is the opposite of P_1 with truth table 4 and weights $w_0 = -1$, $w_1 = -1$ and $w_2 = 1$. And

A	B	$o(\neg A \cap B)$
-1	-1	-1
-1	1	1
1	-1	-1
1	1	-1

Table 4: The truth table for perceptron P_2

A	B	$o(P_1 \cup P_2)$
-1	-1	-1
-1	1	1
1	-1	1
1	1	1

Table 5: The truth table for perceptron P_3

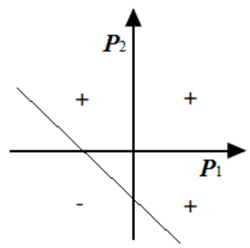

Figure 5: Perceptron P_3

P_3 is the perceptron with truth table 5 and depicted in figure 5. The line in the figure goes through $(-1, 0)$ and $(0, -1)$. Let's calculate the weights for this perceptron:

$$
\begin{aligned}
m &= \frac{(P_2)_2 - (P_2)_1}{(P_1)_2 - (P_1)_1} = \frac{-1-0}{0-(-1)} = \frac{-1}{1} = -1 \\
B &= mA + c \\
B &= -1 \cdot A + c \\
0 &= -1 \cdot -1 + c \\
c &= -1 \\
B &= -A - 1 \\
&\Rightarrow 1 + A + B = 0
\end{aligned}
$$

If we compare this with $w_0 + w_1 x_1 + w_2 x_2 = 0$, we can see that a possible combination for the weights of the P_3 perceptron is: $w_0 = 1$, $w_1 = 1$ and $w_2 = 1$.