Neural Networks: Perceptrons

Perceptrons

Suppose that our training data is a set of n inputs $x_0, x_1, \ldots, x_n$, where each input is 0 or 1 and each output is 0 or 1. (Note: You can equivalently define the logic levels to be –1 or 1, or any other values.)

Define the activation function to be the step function for some threshold θ, as follows:

$$s_\theta(x) = \begin{cases} 1 & : x > \theta \\ 0 & : x \leq \theta \end{cases}$$

The output of a single neuron with threshold θ, incoming inputs $\mathbf{x}$, and corresponding weights $\mathbf{w}$ is defined as:

$$output = s_\theta(\sum_{i=1}^{n} w_i x_i)$$

Let's design a few single perceptrons:

OR

Our desired training data is as follows, where we present two inputs a and b to a single neuron and want $OR(a, b)$ to be its output:

a	b	OR(a,b)
0	0	0
0	1	1
1	0	1
1	1	1

To realize this, we have two weights w_a and w_b along with a threshold θ.

Let $w_a = 1$ and $w_b = 1$ and $\theta = 0.5$. Then, the output of the neuron is:

$$s_\theta(x) = \begin{cases} 1 & : aw_a + bw_b > 0.5 \\ 0 & : aw_a + bw_b \leq 0.5 \end{cases}$$

For each of the values in the training data, we see that it implements $OR(a, b)$. For example, for (0,1) we have 0*1 + 1*1 > 0.5, so the output is 1 as desired.

AND

By changing the threshold to 1.5, we realize AND:

a	b	AND(a,b)
0	0	0
0	1	0
1	0	0
1	1	1

NAND

NAND is NOT AND (the opposite outputs of AND).

a	b	NAND(a,b)
0	0	1
0	1	1
1	0	1
1	1	0

Here, suppose we have $w_a = w_b = -1$ and $\theta = -1.5$. See if you can verify it works.

XOR

XOR shows us that neural networks are more powerful than linear regression. Plotting these four points, we see they are not linearly separable. This implies we will need more than one perceptron to realize this function.

a	b	XOR(a,b)
0	0	0
0	1	1
1	0	1
1	1	0

Let's first prove that a single perceptron cannot realize XOR. We will do a proof by contradiction. In this proof form, we suppose that something is true, e.g. "The sky is always blue." Then, we find a fact that contradicts this – e.g. we walk outside and notice "The sky is pink." Because this fact contradicts what we assume to be true, we know our original assumption must be false – the sky cannot always be blue.

Claim. XOR can be realized using a single perceptron.

Proof. Suppose XOR can be realized using a single perceptron. Then, from the definition of a perceptron and the definition of XOR above, there must exist some threshold θ and weights w_a and w_b such that the following four inequalities hold:

$$\begin{cases} 0w_a + 0w_b \leq \theta \\ 1w_a + 0w_b > \theta \\ 0w_a + 1w_b > \theta \\ 1w_a + 1w_b \leq \theta \end{cases}$$

By multiplying by 0 and 1, this simplifies to:

$$\begin{cases} 0 \leq \theta \\ w_a > \theta \\ w_b > \theta \\ w_a + w_b \leq \theta \end{cases}$$

We see that $w_a > \theta$ and $w_b > \theta$, and so $w_a + w_b > \theta$. This contradicts the final statement, which claims the opposite – that $w_a + w_b \leq \theta$.

Both inequalities cannot possibly be true, yet they must both be true given the original assumption – a contradiction! Hence, the assumption "XOR can be realized using a single perceptron" must be false. So, at least two perceptrons must be required to implement XOR. ■

XOR Implementation

Two Perceptrons

Suppose there are two perceptrons with thresholds $\theta_1 = 1.5$ and $\theta_2 = 0.5$, where perceptron 2 is the output.

Let the first perceptron have inputs $w_a = w_b = 1$. Let the second have inputs $w_{a2} = w_{b2} = 1$, as well as an input from the first perceptron's output, weighted by $w_o = -100$.

Show that this realizes XOR. (The first perceptron realizes AND on the two inputs. The second realizes OR, except it will not fire in the (1,1) case due to the strong negative weight from the first perceptron.)

Three Perceptrons

Here, we have a two-perceptron hidden layer with a single output perceptron. This is a classic feedforward network – the two inputs both feed into each of the two hidden layer nodes. Both hidden layer outputs feed into the single output layer perceptron.

Let one hidden layer perceptron fire ONLY if the inputs are (1,0). Let the second hidden layer fire ONLY if the inputs are (0,1). Now, let the output layer fire ONLY if at least one of the two hidden layer perceptrons fire.

Show that this realizes XOR.

Challenge Problems

1. Can you realize AND, OR, NAND, NOR using –1 and 1 as logic values?
2. Can you find alternative weights and thresholds that implement XOR using two and three perceptrons? (For the three perceptron case, try fixing all of the input weights to 1.)
3. Show a procedure for adjusting the weights and/or threshold of a perceptron so that it realizes the same truth table but with inputs {–1, 1} instead of {0, 1}. e.g. you have a working perceptron for inputs {0,1} – how would you adjust the weights/threshold to make it realize the same function with inputs {–1,1} instead?