Exercise Set 1: Solutions

## 

## 1. OR

|  |  |
| --- | --- |
| **Weight1** | 1 |
| **Weight2** | 1 |
| **Bias** | 0 |

|  |  |  |
| --- | --- | --- |
| Input 1 | Input 2 | Output |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

## 2. AND

|  |  |
| --- | --- |
| **Weight1** | 1 |
| **Weight2** | 1 |
| **Bias** | -1 |

|  |  |  |
| --- | --- | --- |
| Input 1 | Input 2 | Output |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## 3.NOT

|  |  |
| --- | --- |
| **Weight** | -1 |
| **Bias** | 1 |

|  |  |
| --- | --- |
| Input | Output |
| 0 | 1 |
| 1 | 0 |

# 4. XOR

Here are a set of values that satisfy the XOR truth table.

|  |  |
| --- | --- |
| **Input1Hidden1Weight** | 1 |
| **Input2Hidden1Weight** | -1 |
| **Hidden1Bias** | 0 |
| **Input1Hidden2Weight** | -1 |
| **Input2Hidden2Weight** | 1 |
| **Hidden2Bias** | 0 |
| **Hidden1OutputWeight** | 1 |
| **Hidden2OutputWeight** | 1 |
| **OutputBias** | 0 |

Using the weights to simplify those equations, we get:

The reason this works is:

* Hidden1 is 1 if only Input1 is 1. This gate is “A and not B”
* Hidden2 is 1 if only Input2 is 1. This gate is “B and not A”
* Output is Hidden1 OR Hidden2. This gate is “A or B”

That is “De Morgan’s Law,” and is described more concisely in logic symbols as:

Perceptrons are not limited to digital logic gates and digital logic circuits, but they are capable of simulating them. A single layer network is not able to perform all possible actions however, as shown with XOR, and hidden layers are needed to create more complex behavior.