# 1. OR

Out = Sign(Input1 \* Weight1 + Input2 \* Weight2 + Bias)

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

Out = Sign(Input1 + Input2)

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

# 2. AND

Out = Sign(Input1 \* Weight1 + Input2 \* Weight2 + Bias)

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

Out = Sign(Input1 + Input2 – 1)

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

# 3.NOT

Out = Sign(Input \* Weight + Bias)

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

Out = Sign(Input \* -1 + 1)

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

# 4. XOR

This was a trick question! it is not possible to make XOR with a single layer perceptron.

It is possible if you add another layer though – a hidden layer, between input and output.

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

Hidden1 = Sign(Input1 \* Input1Hidden1Weight + Input2 \* Input2Hidden1Weight + Hidden1Bias)

Hidden2 = Sign(Input1 \* Input1Hidden2Weight + Input2 \* Input2Hidden2Weight + Hidden2Bias)

Output = Sign(Hidden1 \* Hidden1OutputWeight + Hidden2 \* Hidden2OutputWeight + OutputBias)

Using the weights to simplify those equations, we get:

Hidden1 = Sign(Input1 – Input2)

Hidden2 = Sign(Input2 – Input1)

Output = Sign(Hidden1 + Hidden2)

The intuition of why this works is:

* Hidden1 is 1 if only Input1 is 1.
* Hidden2 is 1 if only Input2 is 1.
* Output is Hidden1 OR Hidden2.

We need an extra layer to do a second “if statement”

1

1

-1

-1

1