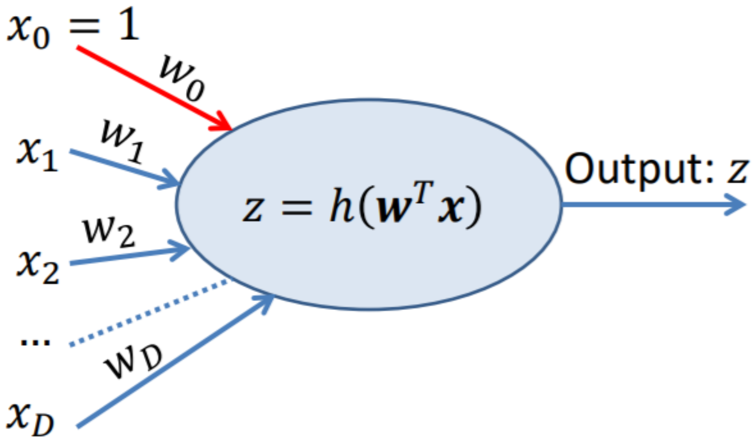
Examples

Perceptrons



- A perceptron computes its output z in two steps:

First step: $a = \boldsymbol{w}^T \boldsymbol{x} = \sum_{i=0}^{D}(w_i x_i)$

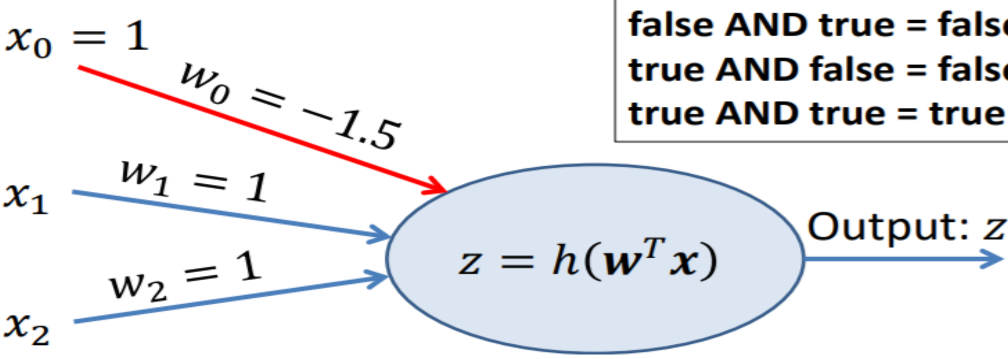
Second step: $z = h(a)$

- h is called an **activation function**.
- For example, h could be the sigmoidal function $\sigma(a) = \frac{1}{1+e^{-a}}$

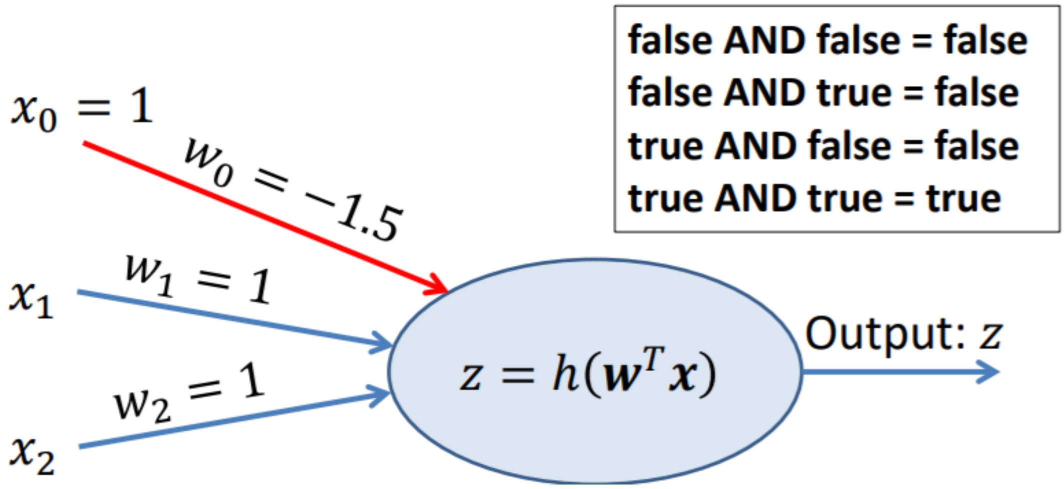
AND Perceptron

- Verification: If $x_1 = 0$ and $x_2 = 0$:
  - $\boldsymbol{w}^T \boldsymbol{x} = -1.5 + 1 * 0 + 1 * 0 = -1.5.$
  - $h(\boldsymbol{w}^T \boldsymbol{x}) = \mathrm{h}(-1.5) = 0.$
- Corresponds to case **false AND false = false**.

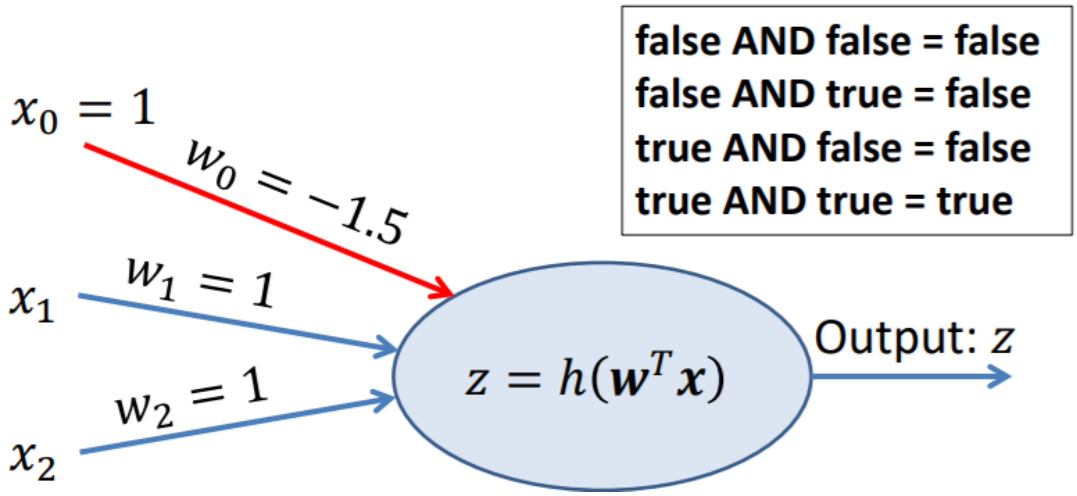
false AND false = false
false AND true = false
true AND false = false
true AND true = true



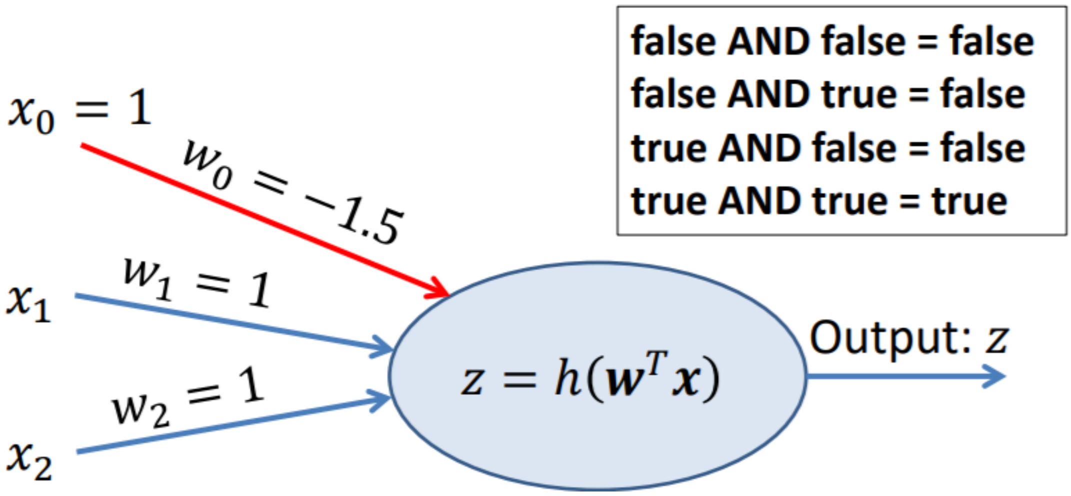
- Verification: If $x_1 = 0$ and $x_2 = 1$:
  - $\boldsymbol{w}^T\boldsymbol{x} = -1.5 + 1 * 0 + 1 * 1 = -0.5.$
  - $h(\boldsymbol{w}^T\boldsymbol{x}) = \mathrm{h}(-0.5) = 0.$
- Corresponds to case **false AND true = false**.



- Verification: If $x_1 = 1$ and $x_2 = 0$:
  - $\boldsymbol{w}^T\boldsymbol{x} = -1.5 + 1 * 1 + 1 * 0 = -0.5.$
  - $h(\boldsymbol{w}^T\boldsymbol{x}) = \mathrm{h}(-0.5) = 0.$
- Corresponds to case **true AND false = false**.

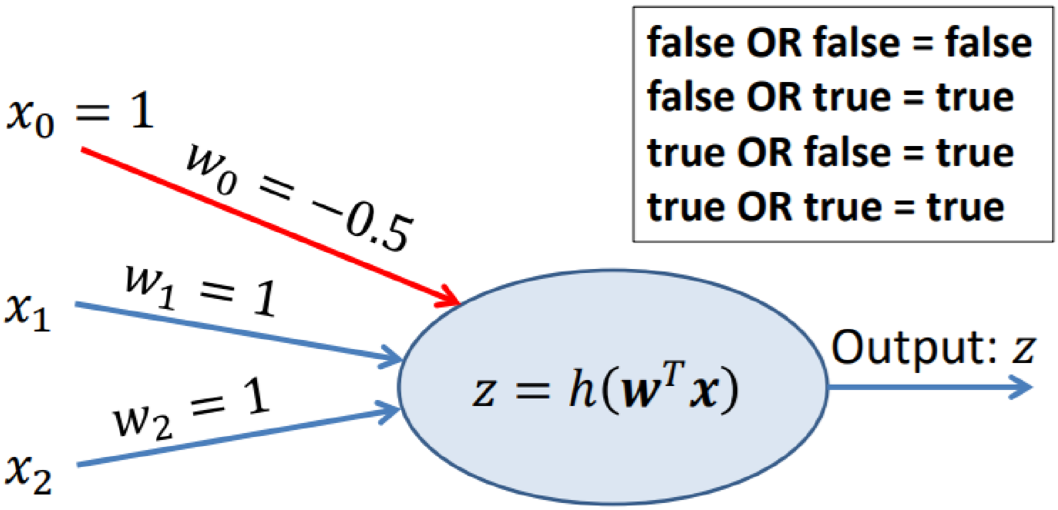


- Verification: If $x_1 = 1$ and $x_2 = 1$:
  - $\boldsymbol{w}^T\boldsymbol{x} = -1.5 + 1 * 1 + 1 * 1 = 0.5.$
  - $h(\boldsymbol{w}^T\boldsymbol{x}) = \mathrm{h}(0.5) = 1.$
- Corresponds to case **true AND true = true**.



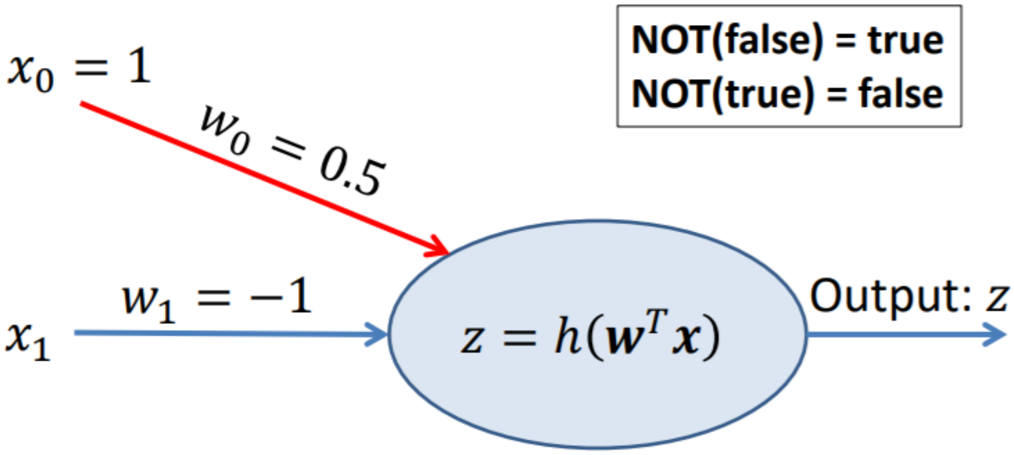
<u>OR Perceptron:</u>

- Verification: If $x_1 = 0$ and $x_2 = 0$:
  - $\boldsymbol{w}^T\boldsymbol{x} = -0.5 + 1 * 0 + 1 * 0 = -0.5.$
  - $h(\boldsymbol{w}^T\boldsymbol{x}) = \mathrm{h}(-0.5) = 0.$
- Corresponds to case **false OR false = false**.



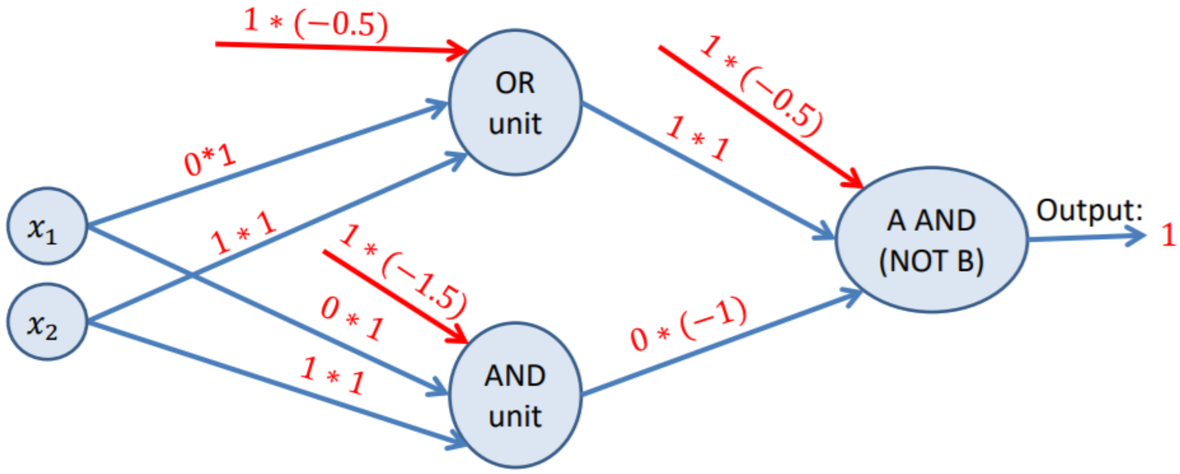
NOT Perceptron

- Verification: If $x_1 = 0$:
  - $\boldsymbol{w}^T \boldsymbol{x} = 0.5 - 1 * 0 = 0.5.$
  - $h(\boldsymbol{w}^T \boldsymbol{x}) = \mathrm{h}(0.5) = 1.$
- Corresponds to case **NOT(false) = true**.

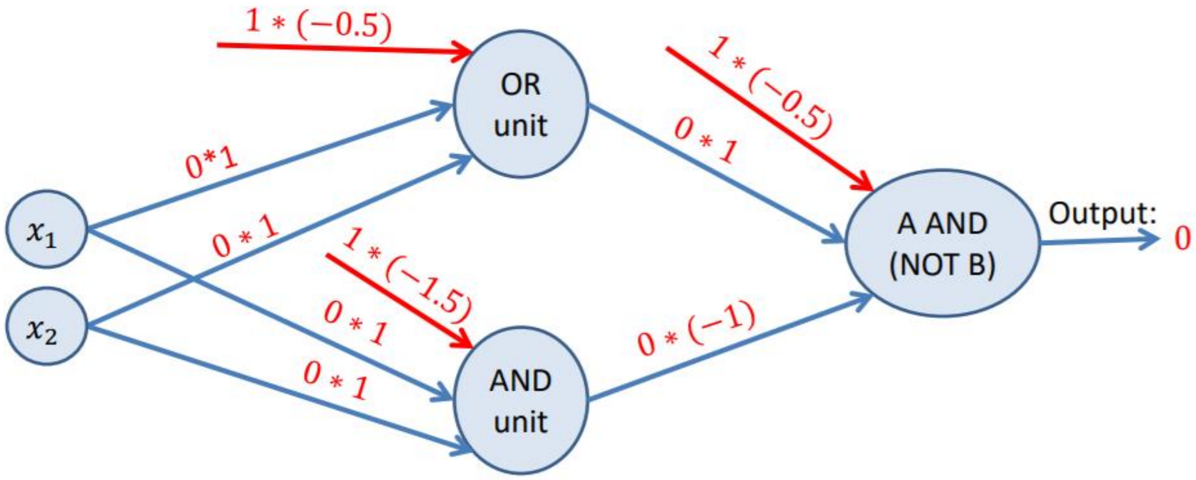


XOR Perceptron

- Suppose that $x_1 = 0, x_2 = 1$ (corresponding to **false** XOR **true**).
- For the output unit (computing the A AND (NOT B) function):
  - The dot product is: $1 * (-0.5) * 1 + 1 * 1 + 1 * 0 = 0.5.$
  - The activation function (assuming a step function) outputs 1.

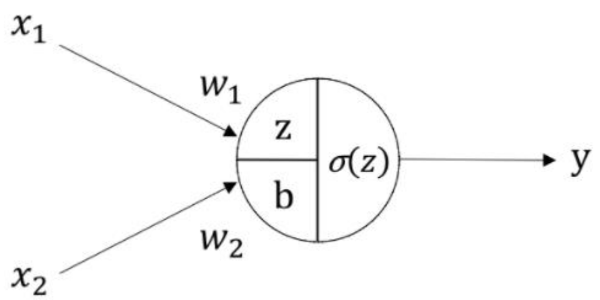


- We can follow the same process to compute the output of this network for the other three cases.
  - Here we consider the case where $x_1 = 0, x_2 = 0$ (corresponding to **false** XOR **false**).
  - The output is 0, as it should be.



Example 2:

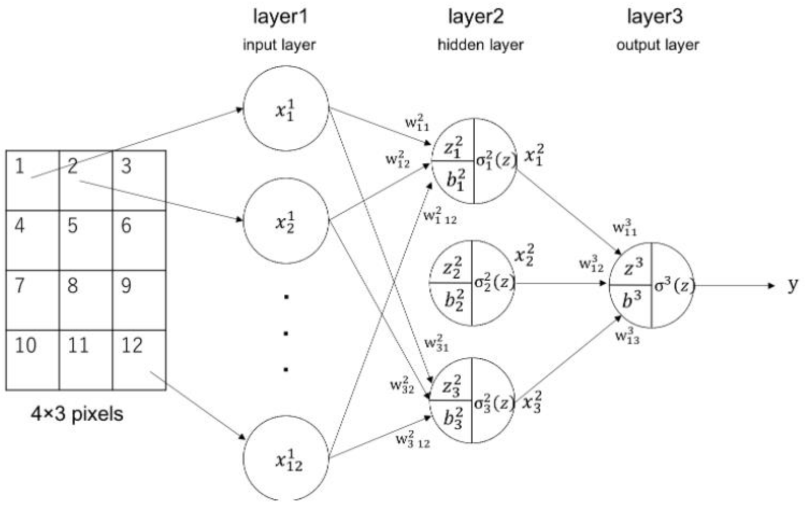
1.Calculation with a Single Unit



The following are the computation examples of using random input and weight values ranged in 0~1, bias as 0:

x1	x2	w1	w2	weighted input z	sigmoid	y
0.1	0.3	0.6	0.2	$z = 0.1 * 0.6 + 0.3 * 0.2 + 0$	$\sigma(z) = \frac{1}{1+\exp(-0.12)} = 0.530$	0.530
0.7	0.2	0.8	0.1	$z = 0.7 * 0.8 + 0.2 * 0.1 + 0$	$\sigma(z) = \frac{1}{1+\exp(-0.58)} = 0.641$	0.641
0.2	0.4	0.9	0.5	$z = 0.2 * 0.9 + 0.4 * 0.5 + 0$	$\sigma(z) = \frac{1}{1+\exp(-0.38)} = 0.594$	0.594

Example 3:



The figure above describes the neural network classifying 4×3 pixel picture.

Here, we will introduce variables and parameters which are commonly used in thesis or interpretations.
The figure would help you grasp more deeply.

x_i^{l-1} : The output from unit i in layer l-1. It becomes the inputs of the units in layer l.

x_j^l : The output from unit j in layer l. It becomes the inputs of the units in layer l+1.

w_{ji}^l : The weight bears on the signal which is outputted from unit i in layer l-1 and goes in to unit j in layer l.

b_j^l : The bias bears on unit j in layer l.

z_j^l : The weighted input of unit j in layer l.

$\sigma_j^l(z_j^l)$: The activation function of unit j in layer l

The following are examples:

x_2^1 : The output from unit 2 in layer 1. It becomes the inputs of the units in layer 2.

x_3^2 : The output from unit 3 in layer 2. It becomes the inputs of the units in layer 3.

w_{31}^2 : The weight bears on the signal which is outputted from the unit No. 1 in layer 1 and goes in to unit 3 in layer 2.

b_3^2 : The bias bears on unit 3 in layer 2.

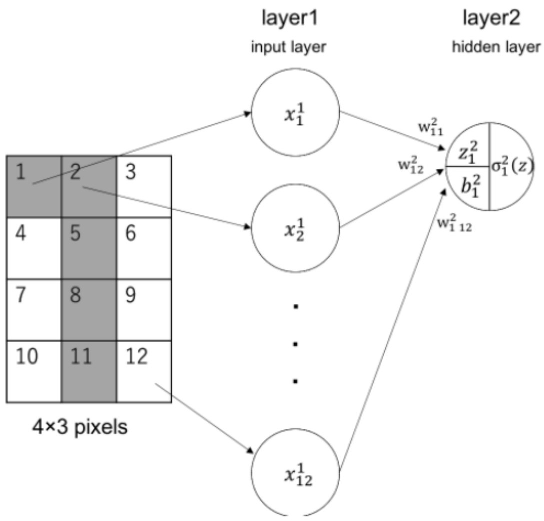
z_3^2 : The weighted input of unit 3 in layer 2.

$\sigma_3^2(z_3^2)$: The activation function of unit 3 in layer 2

Binary Classification of a Picture Using 1 Unit

$$
\begin{aligned}
x_1^1 = 1,\quad x_2^1 = 1,\quad x_3^1 = 0\\
x_4^1 = 0,\quad x_5^1 = 1,\quad x_6^1 = 0\\
x_7^1 = 0,\quad x_8^1 = 1,\quad x_9^1 = 0\\
x_{10}^1 = 0,\quad x_{11}^1 = 1,\quad x_{12}^1 = 0
\end{aligned}
$$

Let's see how the calculation will go on in unit 1 of layer 2.



Just like we did in section 1, the weights are set as random (0~1 value) and the bias is set as 0.

$$
\begin{aligned}
&w_{11}^2 = 0.48,\ w_{12}^2 = 0.29,\ w_{13}^2 = 0.58,\ w_{14}^2 = 0.61,\ w_{15}^2 = 0.47,\ w_{16}^2 = 0.18,\\
&w_{17}^2 = 0.02,\ w_{18}^2 = 0.49,\ w_{19}^2 = 0.67,\ w_{1\,10}^2 = 0.62,\ w_{1\,11}^2 = 0.017,\ w_{1\,12}^2 = 0.28\\
&b_1^2 = 0
\end{aligned}
$$

The weighted input would be as follows:

$$
\begin{aligned}
z_1^2 &= x_1^1 w_{11}^2 + x_2^1 w_{12}^2 + x_3^1 w_{13}^2 + x_4^1 w_{14}^2 + x_5^1 w_{15}^2 + x_6^1 w_{16}^2 + x_7^1 w_{17}^2\\
&\quad + x_8^1 w_{18}^2 + x_9^1 w_{19}^2 + x_{10}^1 w_{1\,10}^2 + x_{11}^1 w_{1\,11}^2 + x_{12}^1 w_{1\,12}^2 + b_1^2\\
&= 1 \times w_{11}^2 + 1 \times w_{12}^2 + 0 \times w_{13}^2 + 0 \times w_{14}^2 + 1 \times w_{15}^2 + 0 \times w_{16}^2 + 0 \times w_{17}^2\\
&\quad + 1 \times w_{18}^2 + 0 \times w_{19}^2 + 0 \times w_{1\,10}^2 + 1 \times w_{1\,11}^2 + 0 \times w_{1\,12}^2 + b_1^2\\
&= 1 \times 0.48 + 1 \times 0.29 + 0 \times 0.58 + 0 \times 0.61 + 1 \times 0.47 + 0 \times 0.18 + 0 \times 0.02\\
&\quad + 1 \times 0.49 + 0 \times 0.67 + 0 \times 0.62 + 1 \times 0.017 + 0 \times 0.28 + 0\\
&= 1 \times 0.48 + 1 \times 0.29 + 1 \times 0.47 + 1 \times 0.49 + 1 \times 0.017 + 0\\
&= 1.277
\end{aligned}
$$

We insert the value into an activation function. We will use sigmoid function for the same reason in section 1. The output of the unit will be,

$$
\begin{aligned}
x_1^2 = \sigma_1^2(z_1^2) &= \tfrac{1}{1+\exp(-z_1^2)}\\
&= \tfrac{1}{1+\exp(-(1.277))}\\
&= 0.782
\end{aligned}
$$

The same computations are conducted in other units of the neural network.