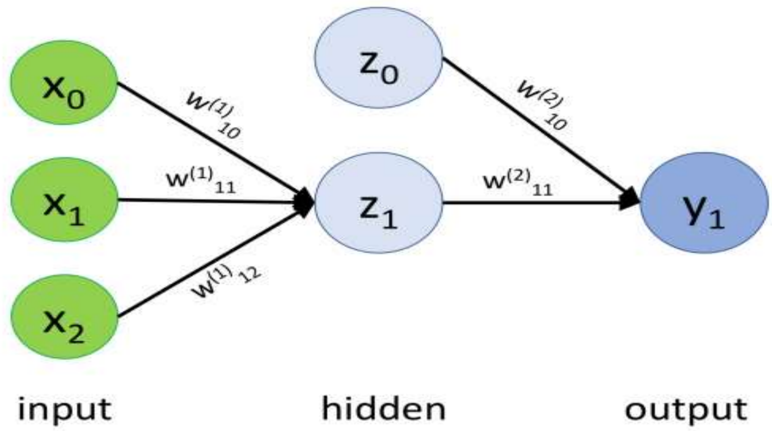
1- Computing activations: Compute the **output value of y1**



In all examples, x = [x0 x1 x2], where x0=1, z0=1

Assume sigmoid activation function in the hidden and output layer

Initialize all weights to 0.1

First example: x = [1 1 0]

Second example: x = [1 0 1]

Third example: x = [1 1 1]

2- Omar is very picky about the toppings he likes on his pizza. His favorite toppings are ham and pineapple. He only likes these toppings when they are on separate pizzas and hates it when they are mixed! He is indifferent about whether his pizza contains mushrooms. Here is a table summarizing his topping preferences:

ham?	pineapple?	mushroom?	good pizza?
no	no	no	no
no	no	yes	no
no	yes	no	yes
no	yes	yes	yes
yes	no	no	yes
yes	no	yes	yes
yes	yes	no	no
yes	yes	yes	no

Consider the following perceptron network:

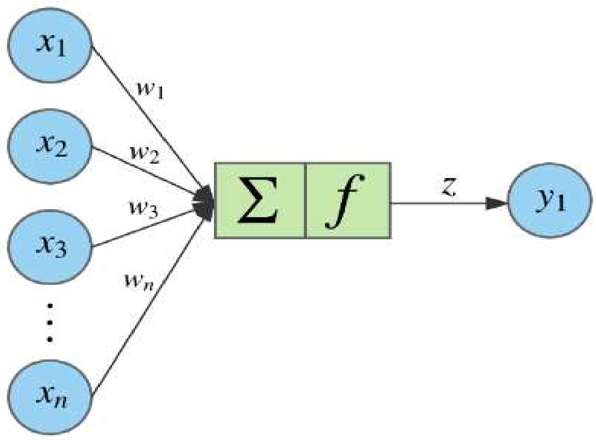
The **inputs** are set to 1 if the topping is present and 0 if not present. The output node uses a **logistic activation function** and has a **bias** term of 1.0. An output below 0.5 means Omer does not like the pizza and above 0.5 means Omer likes the pizza.

Manually modify the network **weights** to try to maximize the number of correct classifications.

Notes: .

Learning rate is 0.2

Hints:



$$z = f(b + x \cdot w) = f\left(b + \sum_{i=1}^{n} x_i w_i\right)$$

$$x \in d_{1\times n},\ w \in d_{n\times 1},\ b \in d_{1\times 1},\ z \in d_{1\times 1}$$

Perceptron learning rule (weight update)

$$w_{i,j}{}^{(\text{next step})} = w_{i,j} + \eta\left(y_j - \hat{y}_j\right)x_i$$

- $w_{i,j}$ is the connection weight between the i^{th} input and the j^{th} neuron.
- x_i is the i^{th} input value of the current training instance.
- $\hat{y}_j$ is the output of the j^{th} output neuron for the current training instance.
- y_j is the target output of the j^{th} output neuron for the current training instance.
- η is the learning rate