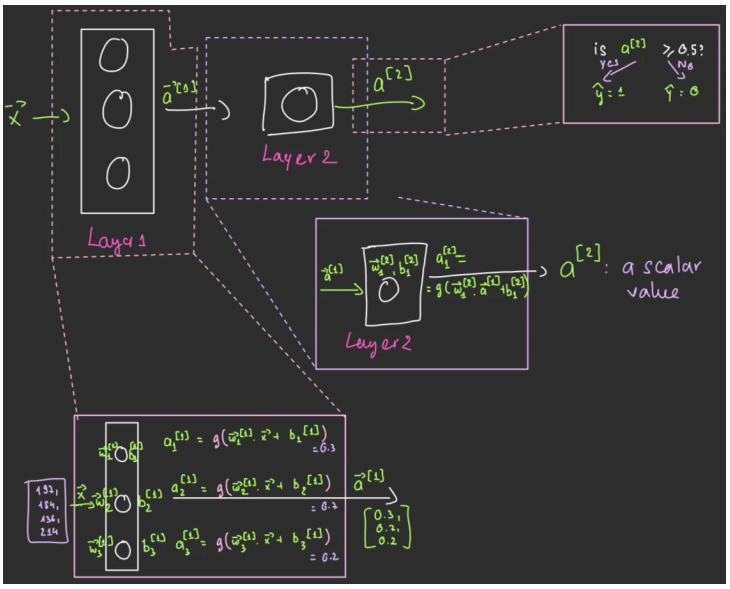
Neural Network model

Neural network layer



source: 'my note'

- **Neural Network Layers**: each neuron processes input features using logistic regression.
- **Hidden Layer Computation**: computes activation values based on input features, producing a vector of outputs.
- **Output Layer**: takes the activation vector from the hidden layer to generate the final prediction.
- **Binary Prediction**: A thresholding step can convert the output into a binary prediction (e.g., yes/no).

For l layers in the model, consider the prediction's simple model:

$$\underset{\text{input layer}}{\vec{x}^{[0]}} \rightarrow \underset{\substack{\text{hidden layer}\\ \text{activation value}}}{\vec{a}^{[1]}} \rightarrow \cdots \rightarrow \underset{\substack{\text{hidden layer}\\ \text{activation value}}}{\vec{a}^{[l-1]}} \rightarrow \underset{\substack{\text{output layer}\\ \text{activation value}}}{a^{[l]}}$$

With $\vec{w}_j^{[l]}, b_j^{[l]}$ we can calculate the **activation value of layer** l, unit (**neutron**) j:

$$\vec{w}_j^{[l]}, b_j^{[l]} \rightarrow a_j^{[l]} = g(\vec{w}_j^{[l]} \cdot \vec{a}^{[l-1]} + b_j^{[l]})$$

- $\vec{a}^{[l-1]}$: output of layer $l-1$ (previous layer)
  - If $l-1=0$: $\vec{a}^{[0]}=\vec{x}$
- $\vec{w}_j^{[l]}, b_j^{[l]}$: parameters w & b of layer l, unit j.
- g : sigmoid ⇒ **“activation function”**

The output a from each neuron in a layer is collected into an activation vector $\vec{a}$, which then serves as the input to the following layer:

$$\vec{a}^{[l]} = [a_1^{[l]} \ \dots \ a_j^{[l]}]$$

In the final output layer, producing a scalar value, then we compare it to a threshold:

$$\text{is } a^{[l]} \geq \text{threshold}$$
$$\text{Yes: } \hat{y}=1$$
$$\text{No: } \hat{y}=0$$

Inference: making prediction (foward propagation)

The above steps is Foward Propagation.

Forward propagation is the process of computing the output of a neural network by passing input data through its layers.