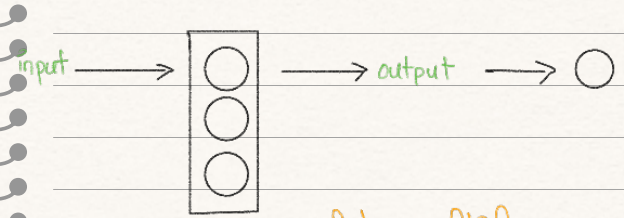
Subject: Nueral Networks / /



Demand Prediction:

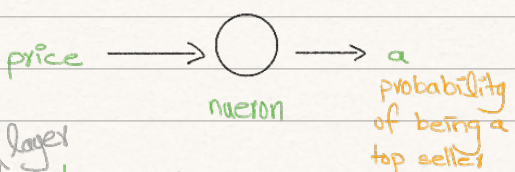
- A product will be a top seller?

x = price

$a = f(x) = \frac{1}{1+e^{(wx+b)}}$

↓ activation

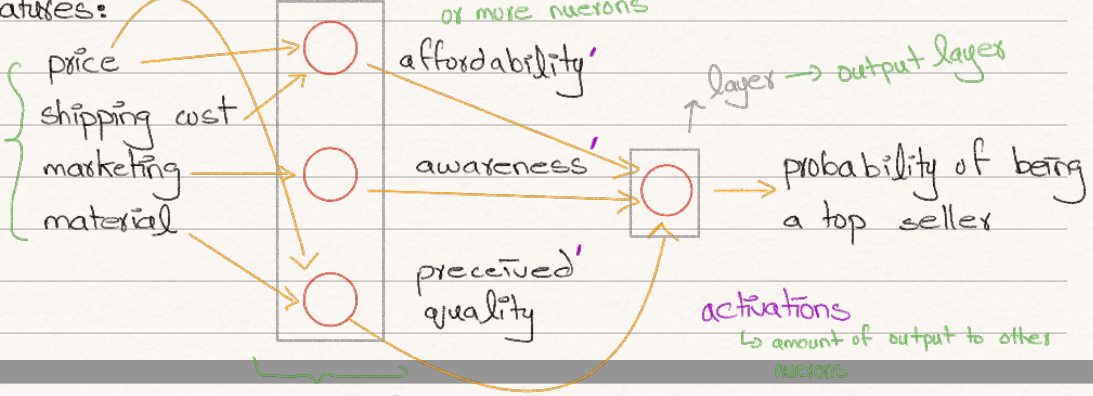
↳ How much a nueron is sending a high output to other nuerons



layer

↳ can have one or more nuerons

Features:

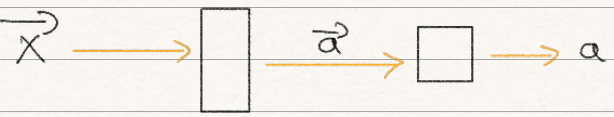


hidden layer

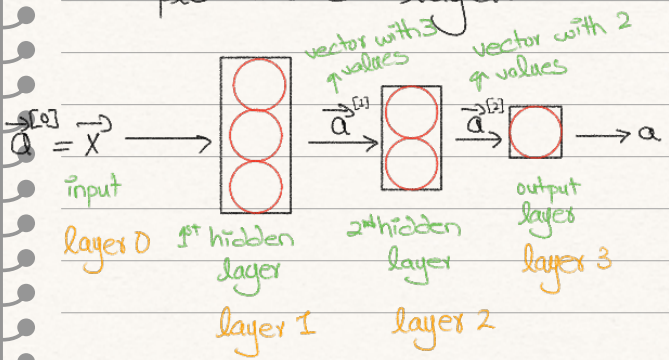
Subject: / /

4 numbers 3 numbers 1 number

As it would to difficult to manually choose which nueron will have access to which input feature. Therefore, each nueron has access to every input of the previous layer. In above example, the nueron will pick apporiate features.



Multiple hidden layers



$a^{[l]}$ ↳ layer index

↳ output

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

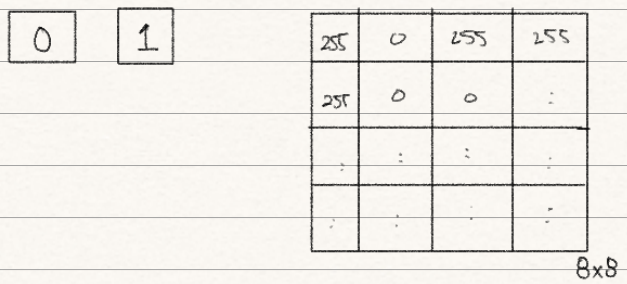
(g: activation function)

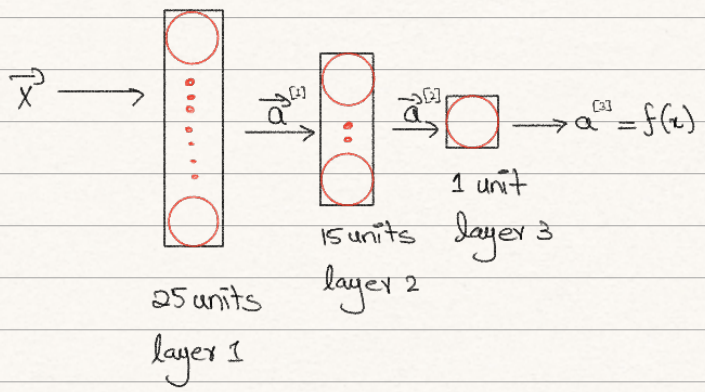
↳ activation of layer l, unit (nueron) j

Subject: / /

Inference — Foward Propagation

Digit Recognition:





As we move from left to Right, this is called forward propagation

Tensorflow:

```
x = np.array([[200.0, 17.0]])
layer_1 = Dense(units=3, activation='sigmoid')
```

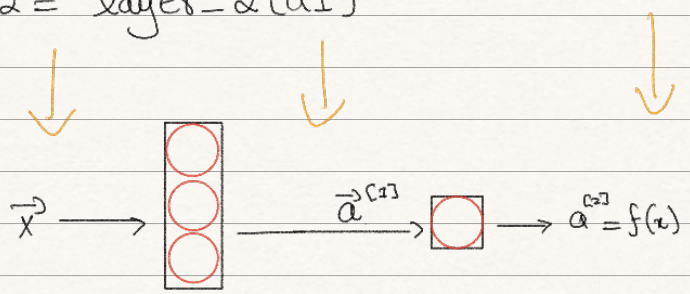
↓ no. of neurons

Subject: / /

```
a1 = layer_1(x)

layer_2 = Dense(units = 1, activation = 'sigmoid')

a2 = layer_2(a1)
```



Nueral Network in Tensorflow:

```
layer_1 = Dense(units = 3, activation = 'sigmoid')
layer_2 = Dense(units = 1, activation = 'sigmoid')

model = Sequential([layer_1, layer_2])
```

↓ Joins layers to make a nueral network

```
x = np.array([[...]])
y = np.array([...])
model.compile(....)
model.fit(x, y)
```

↳ Train on x & y

```
model.predict(x_new)
```

↳ Does inference

Subject: / /

OR

```
model = Sequential([Dense(units=3, activation='sigmoid'),
                    Dense(units=1, activation='sigmoid')])
```

The weight matrix $\mathbf{W}$ will be of shape $(5, 3)$:

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix}$$

- **Rows**: Correspond to input features x_1 to x_5.
- **Columns**: Correspond to units y_1, y_2, y_3.

Each element w_{ij} in the matrix represents the weight associated with input feature i (row) for unit j (column). These weights determine how much influence each input feature has on each unit's output.