Sigmoid function

Neural network: terminology of NN

The intuition of neural networks

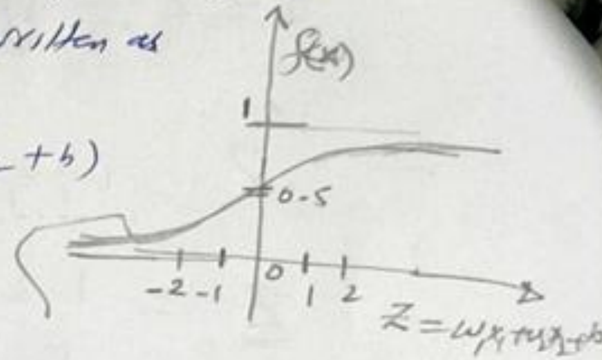
xxxx

* let we have 2 feature, then the logistic function for binary classification can be written as

$$f(x) = \frac{1}{1+e^{-(\omega_1 x_1 + \omega_2 x_2 + b)}}$$

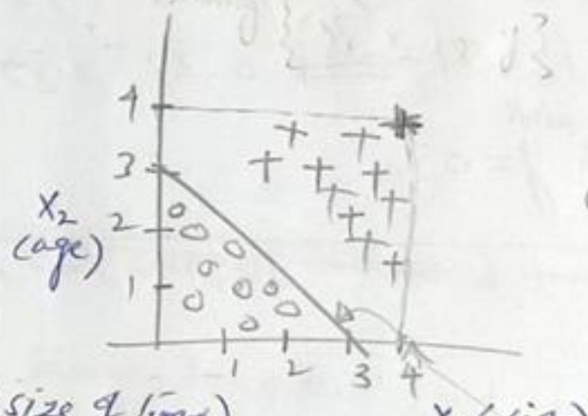
$$f(x) = \frac{1}{1+e^{-z}}$$

$$f(x) = g(z)$$



(1)

* let we have problem of binary classification for tumor is malignant / benign (1/0) & the data is given below.



(2)

$\{(x_1, x_2), y\}$ x_1 (size of tumor)

Training

⇒ So we trained the logistic function & let it learn

$\omega_1 = 1$ & $\omega_2 = 1$ $\boxed{z = \omega_1 x_1 + \omega_2 x_2 + b}$

& $b = -3$ $\Longrightarrow$ $z = x_1 + x_2 - 3 \Rightarrow z >= 0$

(3)

$x_2 = -x_1 + 3$

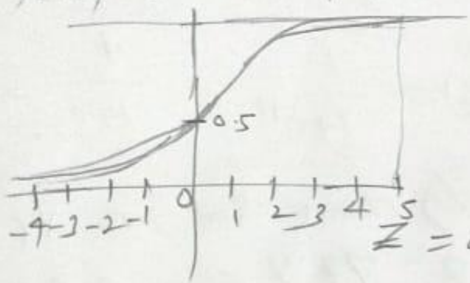
$y = mx + c \Rightarrow \boxed{y = x_2, \; m = -1, \; c = 3}$

<u>Testing</u> use the parameters that learns during training ($w_1=1$, $w_2=1$, & $b=-3$)

lets $\begin{cases} x_1 = 4 \\ x_2 = 4 \end{cases}$

$\therefore \quad z = w_1x_1 + w_2x_2 + b$

$z = 1.4 + 1.4 - 3 = 5 >= 0$,



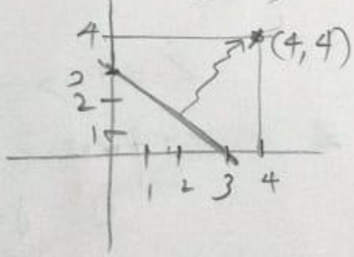
$z >= 0$ $\hat{y} = 1$

① $\boxed{5 \geq 0}$ it mean given input $x_1 = 4$, $x_2 = 4$ is belong to $\hat{y} = 1$ (Class = 1)

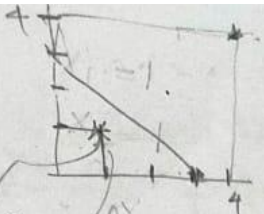
② $f(x) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-5}} = \frac{e^5}{1+e^5}$

$f(x) = \frac{148}{149} = \boxed{0.99}$ (99%) chance the given x is belongs to class $y=1$

or 1% chance / confidence / prob that given $x = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$ is belong to class $y=1$



(i)



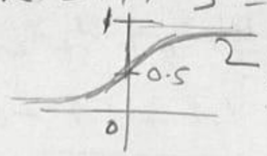
(1,1) $f(x) =$

$f(z) = w_1x_1 + w_2x_2 + b$

$\boxed{x_1 = 1,\ x_2 = 1}$, & $\boxed{w_1 = 1,\ w_2 = 1,\ b = -3}$

testing data — training parameters

$\Rightarrow z = 1 + 1 - 3 = -1 < 0$, so given input is belongs to $y = 0$

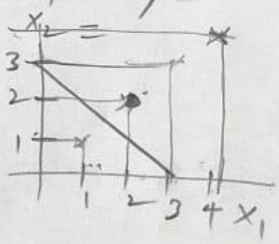


$\frac{1}{1+e^{-z}}$

$f(x) = \frac{1}{1+e^{+1}} = \frac{1}{1+e} = \frac{2.7}{3.7} = 0.27$

1. $\Rightarrow$ It means 0.27 (27%) chance that given $x(1,1)$ is belong to class $y = 1$ & 73% chance that given x is belongs to class $y = 0$

(ii) $x_1 = 2,$



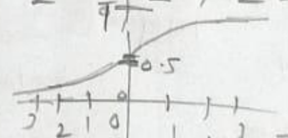
Training: $f(x) = w_1x_1 + w_2x_2 + b$

$w_1 = 1;\ w_2 = 1,\ b = -3$

Testing: $x_1 = 3\ \ x_2 = 2$

$z = w_1x_1 + w_2x_2 + b$

$z = 2 + 2 - 3 = 1$



$z = 1 >= 0$

so given x is belong to $y = 1$

Testing data		
X		
4	1	2
4	1	2

$\rightarrow f(x) = \frac{1}{1+e^{-1}} = \frac{e}{1+e} = \frac{2.7}{3.7} = 0.72$ chance that given x is belong to class $y = 1$ & 0.28% chance that given x is belong $y = 0$

Neural network terminology:

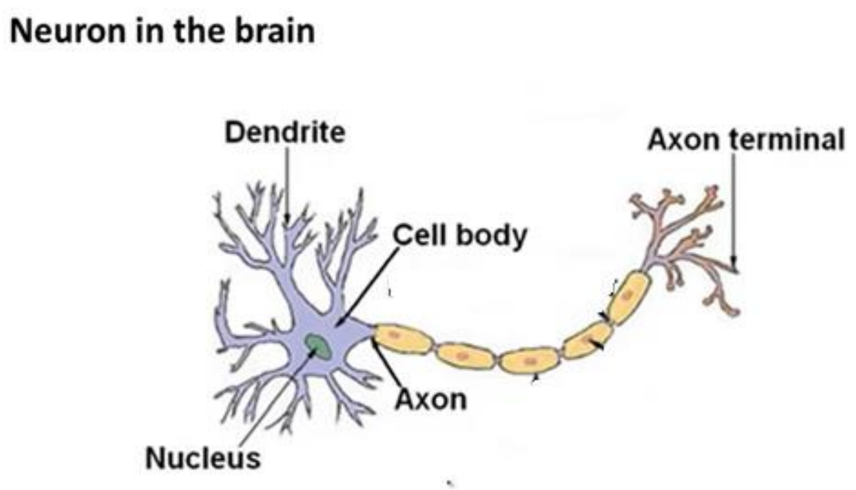
1. **Neural Networks learn a complex nonlinear decision boundary**

    a. Deep NN needs few hidden units (features/ parameters) to learn nonlinear decision boundaries compare to logistic regression/shallower network **[circuit theory]**

<u>Model representation 1</u>

- How do we represent neural networks (NNs)?
    - Neural networks were developed as a way to simulate networks of neurons

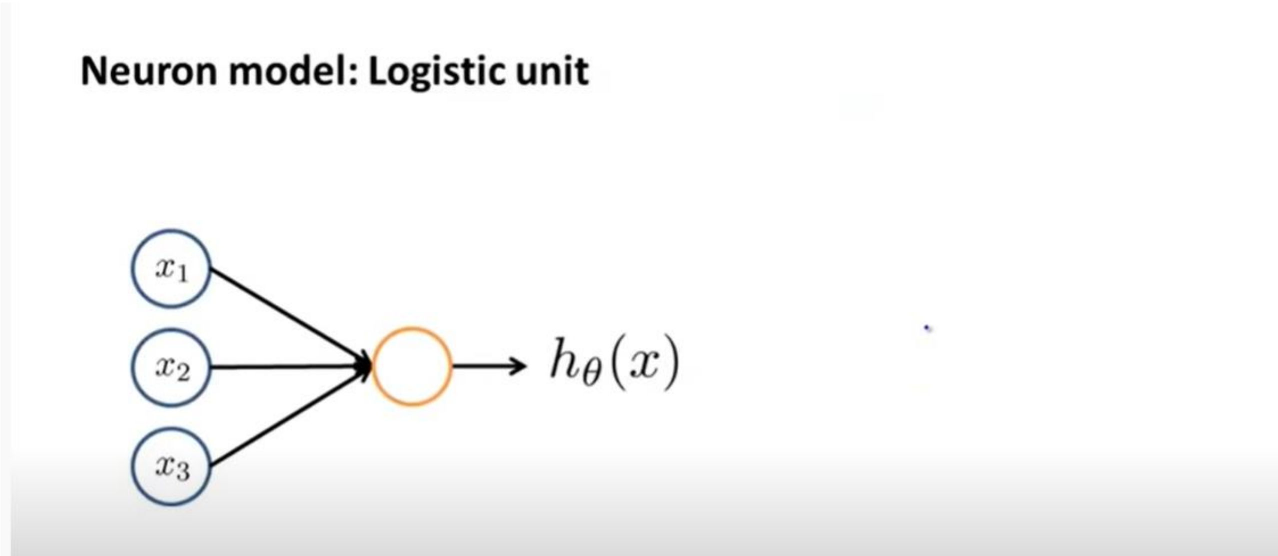
Single Neuron:



- Three things to notice
    - Cell body
    - Number of input wires (dendrites)
    - Output wire (axon)
- Simple level
    - Neuron gets one or more inputs through dendrites
    - Does processing/ computation through the cell body
    - Sends output down the axon
- Neurons communicate through electric spikes
    - Pulse of electricity via axon to another neuron

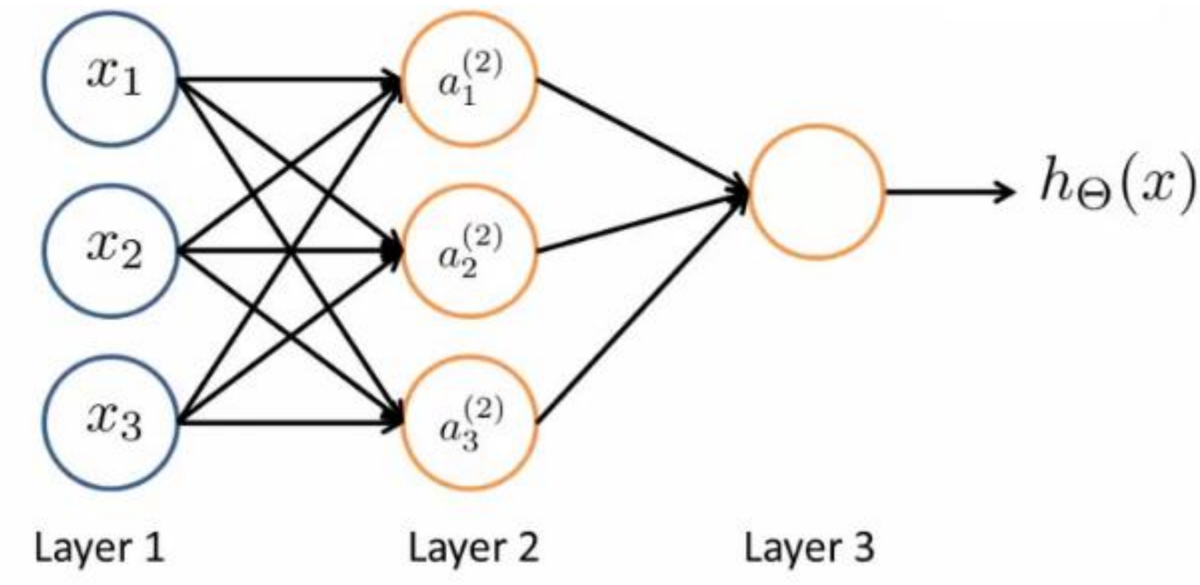
Artificial neural network - representation of a neuron

- In an artificial neural network, a **neuron is a logistic unit**
  - Feed input via input wires
  - Logistic unit does computation
  - Sends output down output wires
- That logistic computation is just like logistic regression hypothesis calculation
- TERMINOLOGY: bias, weights, layers(input, hidden, output)



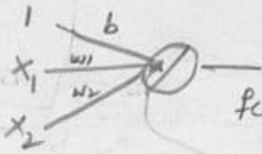
- Very simple model of a neuron's computation
  - Often good to include an x_0 input - the **bias unit**
    - This is equal to 1
- This is an artificial neuron with a sigmoid (logistic) activation function
  - ϴ vector may also be called the **weights** of a model
- The above diagram is a single neuron

Below we have a group of neurons together



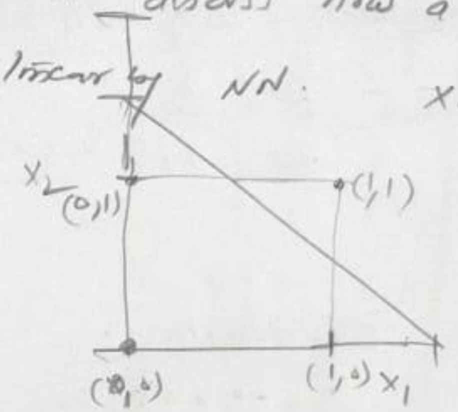
- Here, input is x_1, x_2 and x3 => belong to **input layer (layer1)**
- Three neurons in **layer 2**
    - We could also call output second layer - i.e. ($a_1^{(1)}$, $a_2^{(1)}$ and $a_3^{(1)}$)
- Final fourth neurons which produce the output is **the output layers**
        - Which again we *could* call $a_1^{(3)}$
- The First layer is the **input layer**
- Final layer is the **output layer** – which produces value computed by a hypothesis
- Middle layer(s) are called the **hidden layers**
    - You don't observe the values processed in the hidden layer

• In NN, a single neuron can be consider as sigmoid function



$$f(x) = \frac{1}{1+e^{-(w_1x_1+w_2x_2+b)}} \quad \text{(weights)}$$

• let discuss how a complex decision boundary in linear by NN. XNOR, by consider a single neuron NN.



⇒ For 2 feature x_1 & x_2

$$z = w_1x_1 + w_2x_2 + b$$

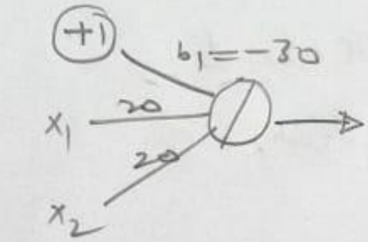
$$f(x) = g(z)$$

$$= \frac{1}{1+e^{-(w_1x_1+w_2x_2+b)}}$$

$$f(x) = \frac{1}{1+e^{-z}}$$

$$f(x) = g(z)$$

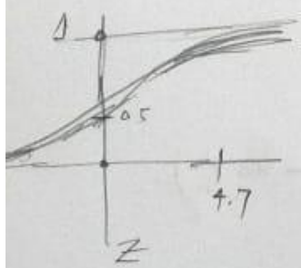
AND GATE



⇒ During training we learnt

$$\boxed{z = (20x_1 + 20x_2 - 30)}$$

$$f(x) = \frac{1}{1+e^{-z}}$$



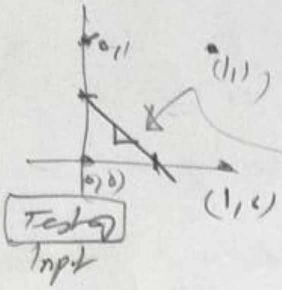
x_1	x_2	x_1 AND x_2
0	0	$g(-30) \approx 0$ ✓
0	1	$g(-10) \approx 0$ ✓
1	0	$g(-10) \approx 0$ ✓
1	1	$g(10) \approx 1$ ✓

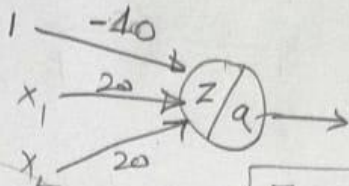
$20x_2 = -20x_1 + 30$

$x_2 = -x_1 + 1.5$

$y = mx + c$

OR GATE   During learning we learnt





$$\boxed{Z = 20x_1 + 20x_1 - 10}$$

$$f(x) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(20x_1+20x_2-1}}$$

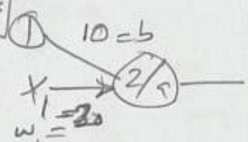
x_1	x_2	x_1 OR x_2 = f(x)
0	0	$g(-10) \approx 0$
0	1	$g(10) \approx 1$
1	0	$g(10) \approx 1$
1	1	$g(30) \approx 1$

$20x_2 = -20x_1 + 10$

$\boxed{x_2 = -x_1 + 0.5}$

$\boxed{y = -mx + 0.5c}$

NOT GATE



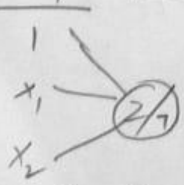
$$f(x) = g(w_1x_1 + b) = g(-20x_1 + 10)$$

x_1	NOT x_1
0	$g(10) \approx 1$
1	$g(-10) \approx 0$

$f(0) = g(10) = 1$

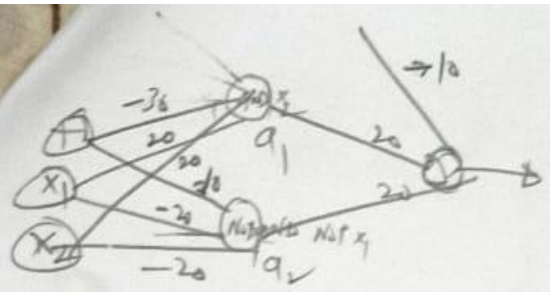
$f(1) = g(-10) = 0$

NOT x_1 AND NOT x_2



$$f(x) = g(10 - 20x_1 - 20x_2)$$

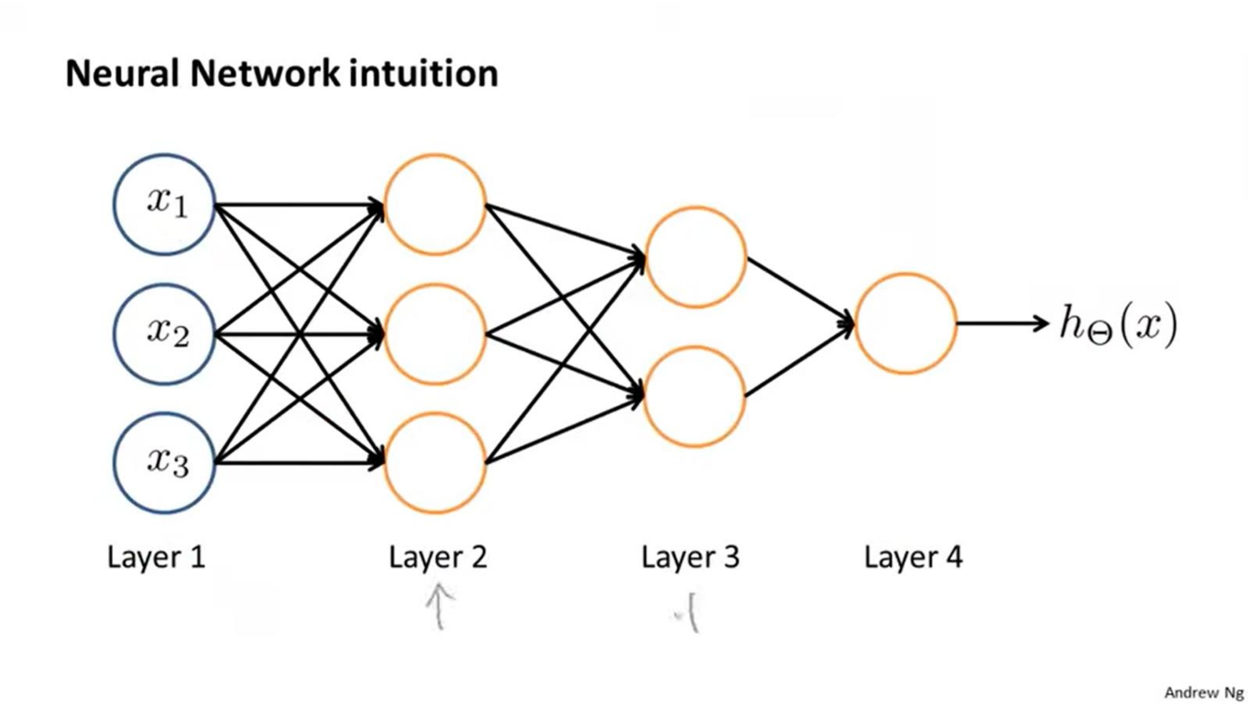
x_1	x_2	NOT x_1 AND NOT x_1
0	0	$g(10) \approx 1$
0	1	$g(-10) \approx 0$
1	1	$g(-10) \approx 0$ $g(-30) \approx 0$



x_1	x_2	$a_1^{[1]}$	$a_2^{[1]}$	$a_1^{[2]} = f(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

This NN with one input, one hidden unit, and one output layer can learn the XNOR function, which is more complex than simple AND/OR functions.

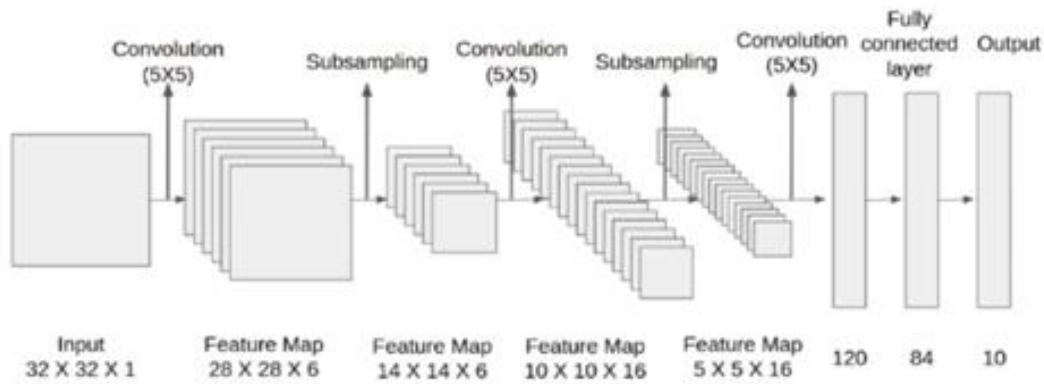
This basic institution where network learning can learn the interesting and more complex nonlinear function/decision boundaries.



LENet 5

- LeNet was introduced in the research paper "**Gradient-Based Learning Applied To Document Recognition**" in the year 1998 by Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner.
- Application: Try to read the postal code from the input image
- They used this architecture for **recognizing the handwritten** and **machine-printed characters.**
- The main reason behind the popularity of this model was its simple architecture. It is a multi-layer convolution neural network for image classification.
- Let's understand the architecture of Lenet-5. The network has 5 layers with learnable parameters and is hence named Lenet-5. It has three sets of convolution layers with a combination of average pooling. After the convolution and average pooling layers, we have two fully connected layers. At last, a Softmax classifier classifies the images into respective classes.

The Architecture of Lenet-5



- LeNet-5 CNN architecture is made up of 7 layers. The layer composition consists of 3 convolutional layers, 2 subsampling layers, and 2 fully connected layers

- Let's see the learning the interesting features of the different layers of Lenet5
- Handwritten Digit Classification - Yann Lecun
  https://www.youtube.com/watch?v=yxuRnBEczUU

- Demo of Yann LeCun work=> is pioneer of NN
- His early work for handwritten digit classification: see the
- Input area
- Column show visualization of first layer output features visualization
- Next layer
- Finally, the hidden layers learn to feed the next layers and ultimately to the final layer to recognize the handwritten digits.
- **See more detail about architecture https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/**

Neural Networks Intuition

Example:
Recognizing Images