One Shot Learning :-

→ Mostly used for face recognitⁿ

→ Say we have 4 photos of 4 diff people (1 photo each)

→ If we try and build a Network & predict probability using Softmax then there are ↑ chances the network will not perform up to expectⁿ mainly because of 1 training eg per person

→ In a company with 1000 employees and setting up a face-recognition system with softmax will be costly and highly imperfect as training eg. are also less in number

→ Also, if a new person joins then we need to re-train the network which is a pain again

Learning a similarity function

Instead of the process described above, we try to learn a similarity funct"

$d(img1, img2)$ = degree of diff b/w images

if d is ↓ then these images are similar or might belong to a same person.

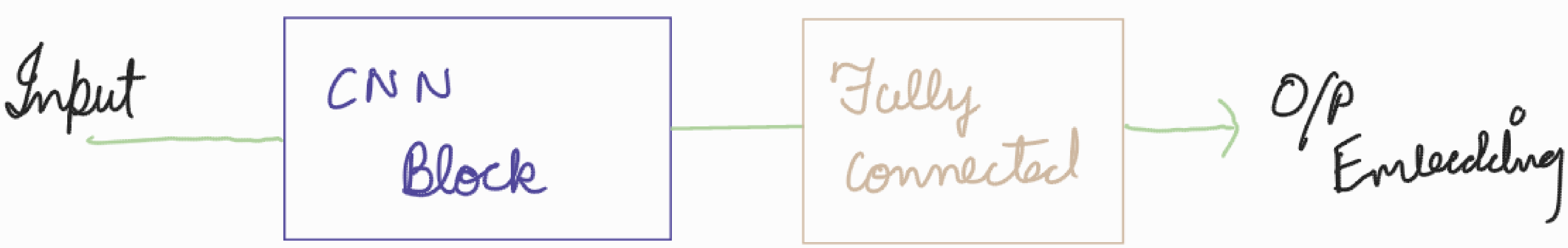
The threshold used here to determine is τ

$$d(img1, img2) \le \tau \longrightarrow \text{Same}$$
$$> \tau \longrightarrow \text{different}$$

Siamese Network

Traditional systems :-

Siamese system



For every $x^{(i)}$ we get the embedding or encoding vector. We learn the parameters such that

→ If x^i & x^j are same then :

$$\| f(x^i) - f(x^j) \|^2 \text{ is small}$$

otherwise its large.

Triplet Loss Function

Anchor Image	Image with same class as Anchor	Anchor Image	diff class from anchor
A	Positive image - P	A	Negative image - N

We want :-

$$\| f(A) - f(P) \|^2 \leq \| f(A) - f(N) \|^2$$

Or

$$\| f(A) - f(P) \|^2 - \| f(A) - f(N) \|^2 \leq 0$$

There might be corner cases like all the embedding become 0 or all embedding becomes equal, to overcome this we add a margin $\underline{\underline{\alpha}}$ to the equation

$$\boxed{\|f(A)-f(P)\|^2 - \|f(A)-f(N)\|^2 + \alpha \leq 0}$$

α also helps us separate the positive eg from -ve. If α is 9 the embedding b/w A & P should be very close & b/w A & N should be pretty far to compensate for α.

Here, this method compares 3 eg. in 1 go, that's why its called/named Triplet Loss.

Loss Function :-

Given 3 images/eg/Embeddings :- A, P, N

$$\mathcal{L}(A, P, N) = \max(\underbrace{\|f(A)-f(P)\|^2 - \|f(A)-f(N)\|^2 + \alpha}, 0)$$

ideally it should be ≤ 0 ie we should get 0 loss

We try to learn embedding such that the loss is as close to 0 as possible.

Cost Function:-

$$J = \sum_{i=1}^{m} \mathcal{L}(A^{(i)}, P^{(i)}, N^{(i)})$$

A major challange while traing this network is choosing the triplets. If we choose N at random then there are ↑ chances that we get 0 loss. We should try and pair data/image/Embeddings/classes which seems to be similar but are different. eg:-

A = Lung cancer related	A = Lung cancer
P = 〃	P = Lung cancer
N = Bacterial infection	N = Kidney cancer

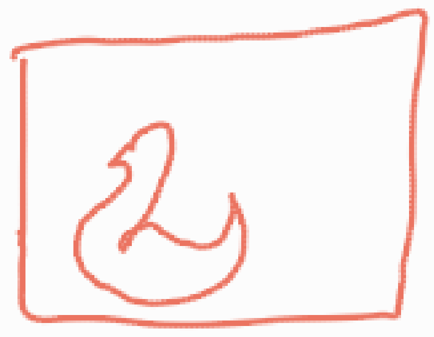
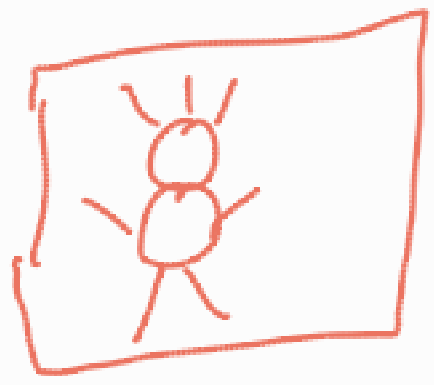
In above eg, 1^{st} one chosen at random will have ↑ chances of returning 0 loss. While for the 2^{nd} one, network will have to work harder to minimize the loss.

Contrastive Loss function

→ Like triplet loss, contrastive loss is also a distance based loss function.

→ The dataset contains an anchor observation, a positive or negative observation and a flag with value = 0 if both are same & 1 if not

→ The idea is to learn embeddings such that loss for same type of observation is close

Anchor / original Observation	Randomly / Logically Selected observation	If both are same y=1 else 0
		1
		0

The table above shows images but its actually an image / any vector.

Loss Function :-

Loss (x_o, x_R, y) is given by :-

$$\boxed{(1-y)\,\frac{1}{2}\,D(x_o, x_r) \;+\; y\,\frac{1}{2}\,\max\left(0,\, m - D(x_o, x_r)\right)}$$

where :- $D(x_o, x_r)$ is distance b/w vectors of orignal ie x_o & randomly selected observation x_r

$y = 1$ if x_o & x_r are diff & 0 if same

When $y = 0$ ie x_0 & x_r belong to same class

$$\mathcal{L}(x_0, x_r, y) = \underbrace{\frac{1}{2} D(x_0, x_r)^2}$$

We try to minimize this distance

When $y = 1$ ie x_0 & x_r belong to different class:-

$$\mathcal{L}(x_0, x_r, y) = \frac{1}{2}\left(\max\left(0, m - D(x_0, x_r)\right)\right)^2$$

Here, m is the margin usually its set to 1

Ideally, $m - D(x_0, x_r)$ should result into a -ve number

ie. if x_0 & x_r are different, distance b/w them should be higher. The algorithm penalizes if 2 different class observations are closer to each other.