

Session 1 INTRO To NEURAL NETWORK

Feb 05, 2024

John Connor silently watching you become friends with artificial neural networks:



AGENDA

- ① Evaluate Algos we've read so far.
- ② NN in daily life
- ③ Inspiration from Biological model.

What we've done So Far??

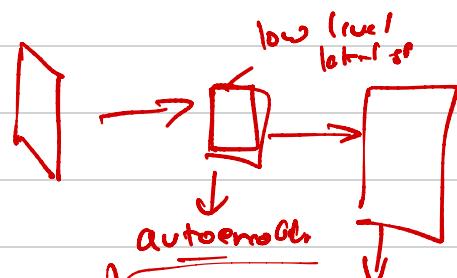
→ Machine Learning
→ Deep Learning

→ Basic

→ CNN → Image

→ NLP → text

audio → signal



autoencoder

CNN

Fears of cap

Job-switches → 2 years

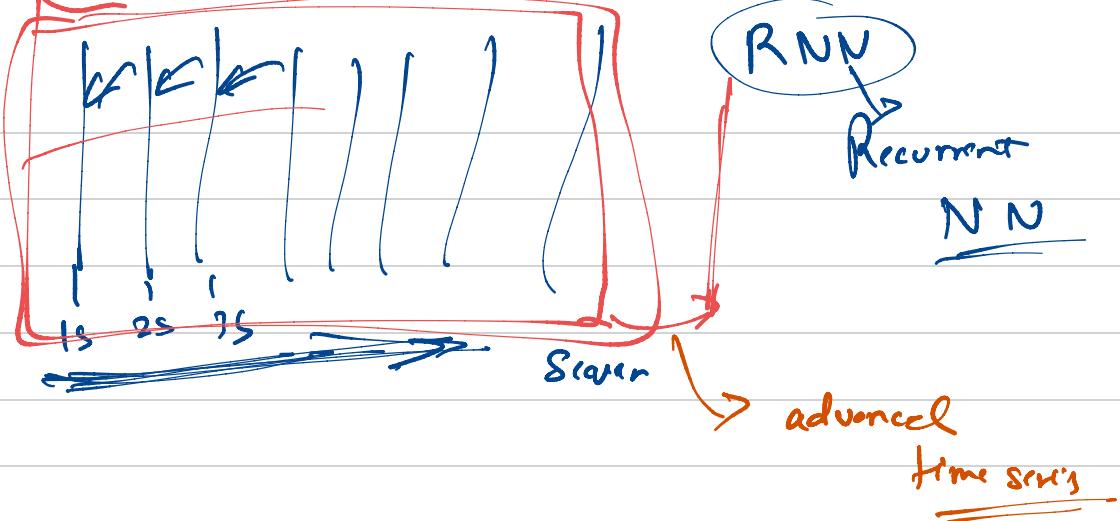
Gender

Personal
Details
You-
CR

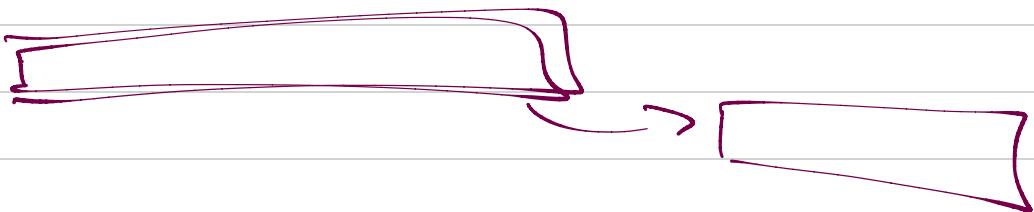


2, 3, 4, 5

NN → decent accuracy



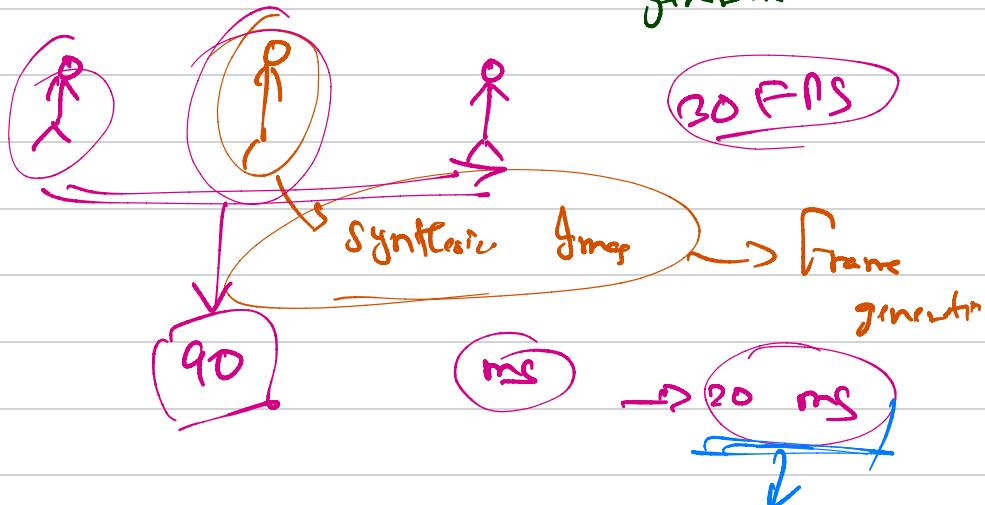
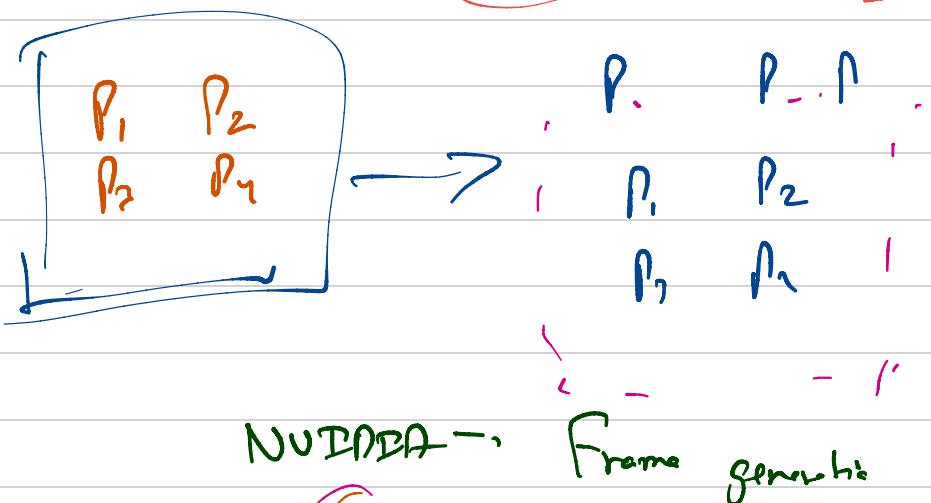
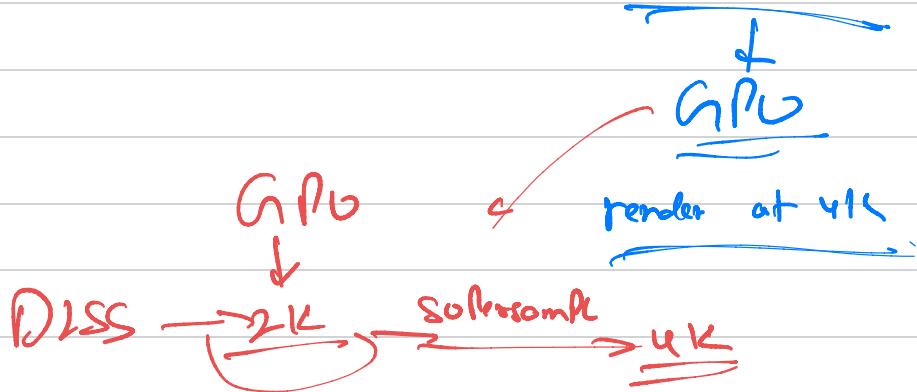
LM \Rightarrow Research Pop + factor

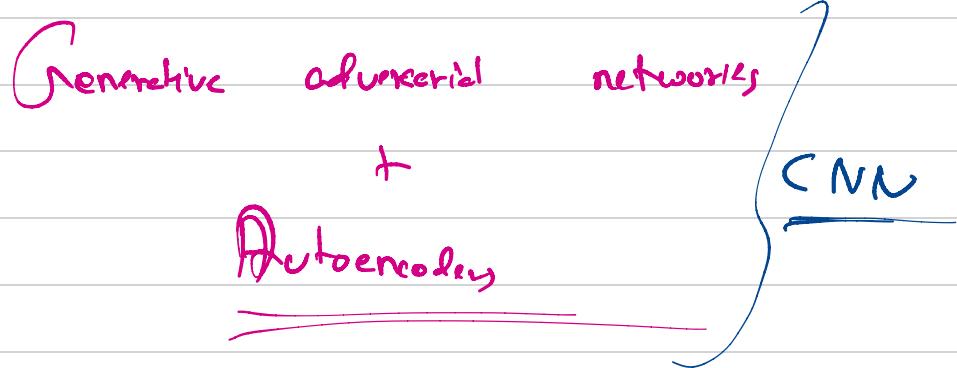
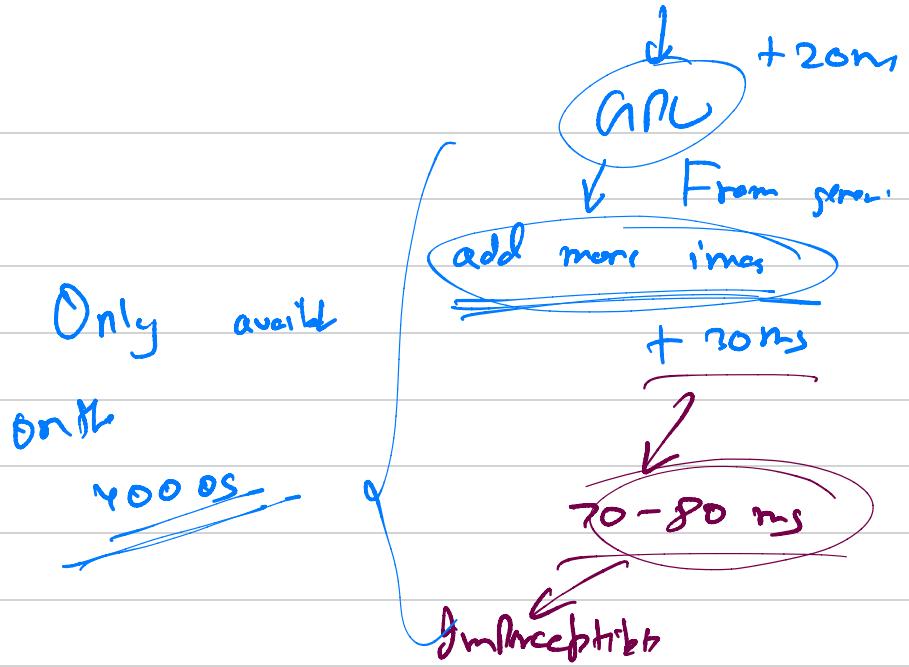


DHSS \rightarrow Nvidia

30 FPS \rightarrow Frames Per Second
 \downarrow more smooth $\frac{\text{more}}{\text{FPS}}$

4K resolution → computationally expensive





In 1957 \rightarrow Perceptron \rightarrow $w^T x + b \geq 0$
 \Leftrightarrow v/c class

\downarrow
Data v.v-less \rightarrow Signal($w^T x + b$) \rightarrow V-loo (conf. bound)

1980 \rightarrow Geoffrey Hinton \rightarrow Backpropagation

Geoff Hinton after writing the paper on backprop in 1986



\rightarrow 2006

\downarrow lack of data

+

Storage problem

+

Computation Power,

2010 \rightarrow

\rightarrow 2012

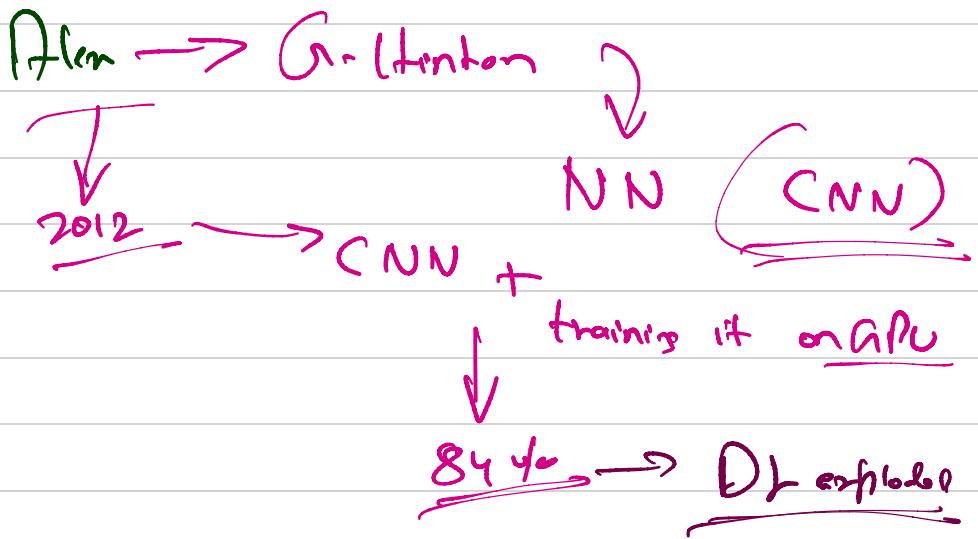
ImageNet \rightarrow

1000 categories

Freq

Jul 11 2012 → SUM (Pretty Fast)

66.00
↓
66.02



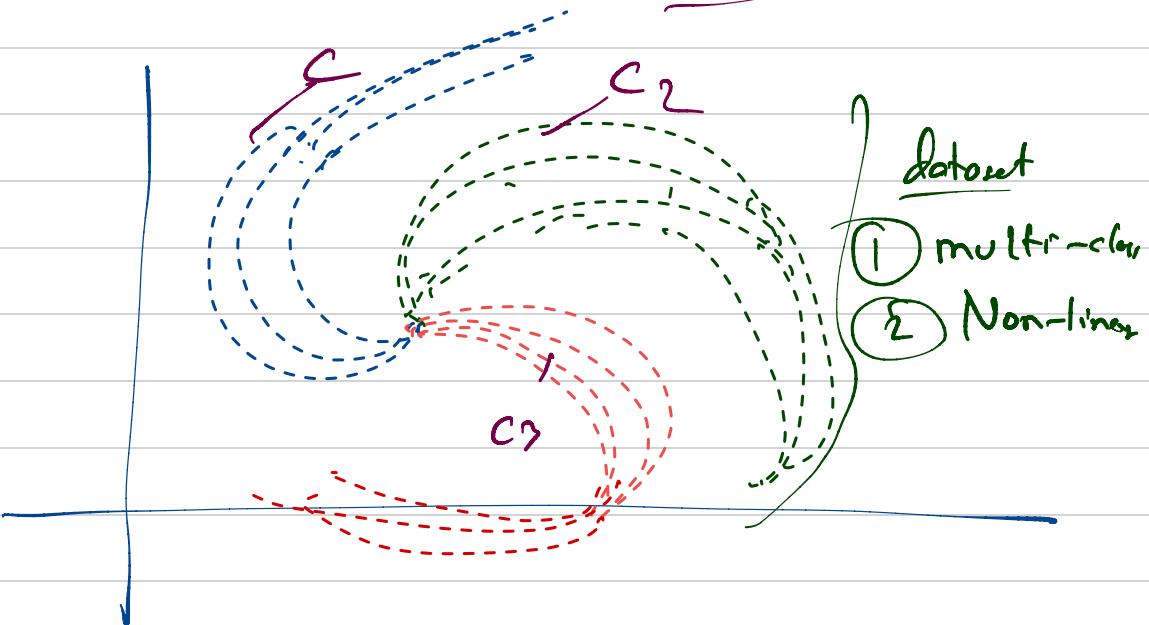
2014 → Google Net → 94%

→ Foo → Create → LIN

Google
↓
~~TensorFlow~~

Meta / Facebook
↓
PyTorch

= PyTorch



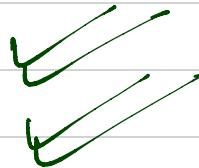
① Log- Regression → Linear
for multiclass →

Polynomial regression → "manual Feature-"
eng

②

KNN

multi - class



Non - Linear

Problem ?? → Entire data in Ram
Storage

③

Decision trees:

multi - class



Non - Linear

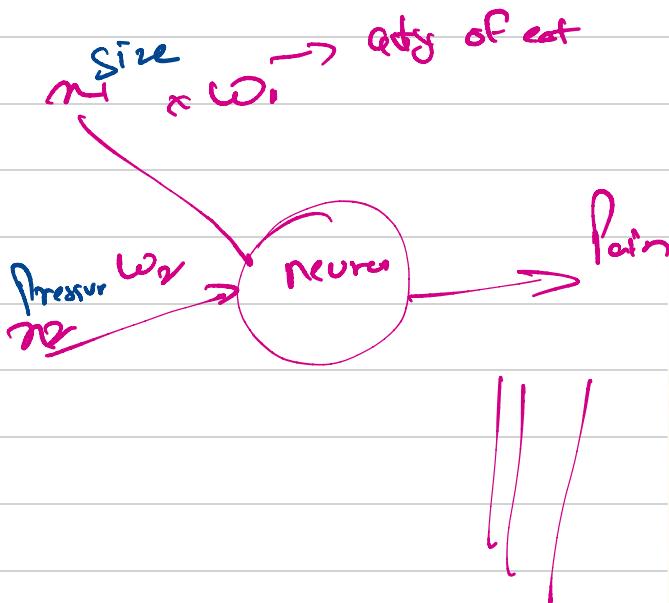


Problem: Curse of dimensionality

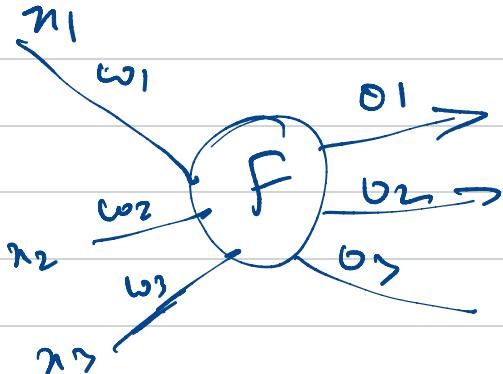
→ Neural Network

- ① Can handle big-data
- ② Requiring no manual Feature Engineering

* Universal Function approximator.



How Neural Networks work?
Neurons:



$f(\omega_1x_1 + \omega_2x_2 + \omega_3x_3 - b)$
 Activation func

$\text{ReLU} \rightarrow x, x > 0$

$0, x < 0$

$f_1 = \text{add}()$

$f_2 = \text{square}()$

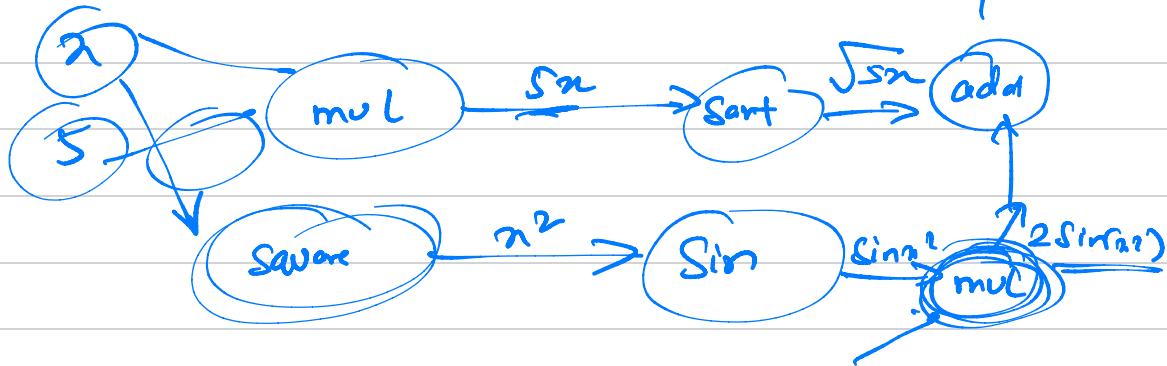
$$f(n) = 2 \sin(n^2) + \text{sort}(s_n)$$

$f_3 = \text{sort}$

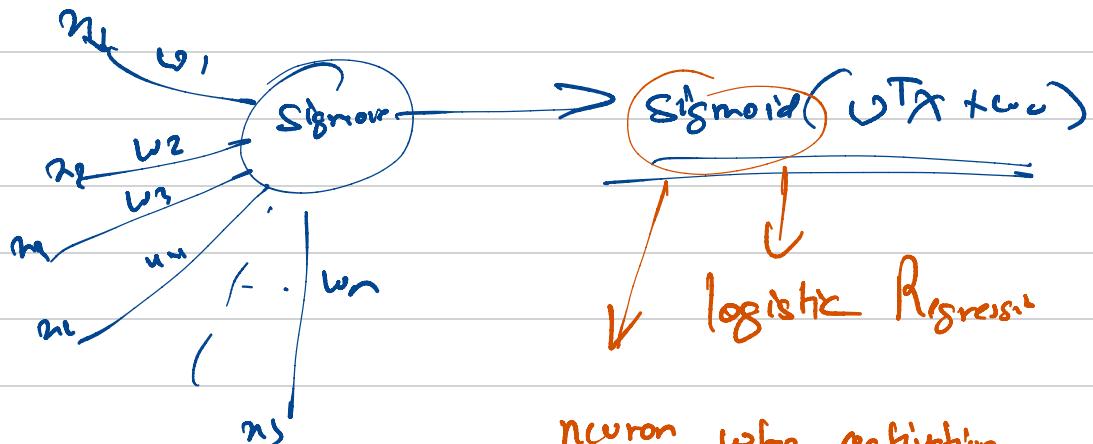
$f_4 \rightarrow \sin()$

$f_5 \rightarrow \text{mul}()$

$$2 \sin(x^2) + \text{sort}(s_x)$$



2

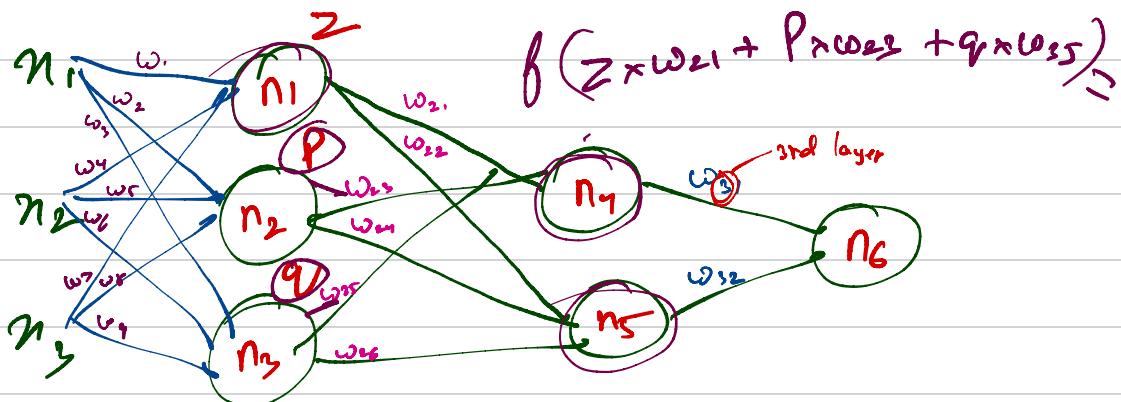


neuron after activation

func is
sigmoid

output of neuron (single neuron)

$$= f(\omega^T x + w_0)$$



$$f(x_1\omega_1 + x_2\omega_2 + x_3\omega_3) = \Sigma$$

$n_u \leftarrow \omega_2, \omega_3$

