Imperial College
London

11 – Classification

Prof Peter YK Cheung

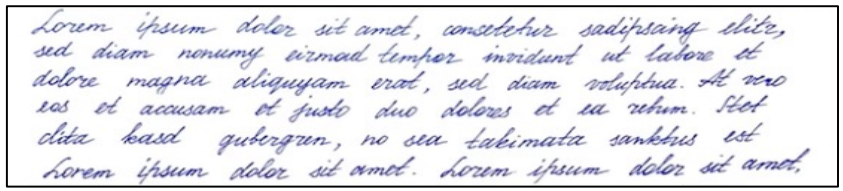
Dyson School of Design Engineering

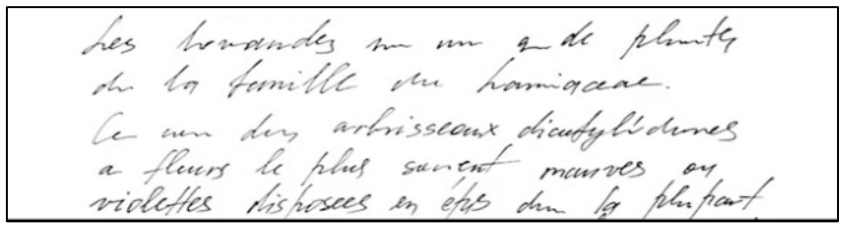
URL: www.ee.ic.ac.uk/pcheung/teaching/DE4_DVS/
E-mail: p.cheung@imperial.ac.uk

Recognition and Classification Problem

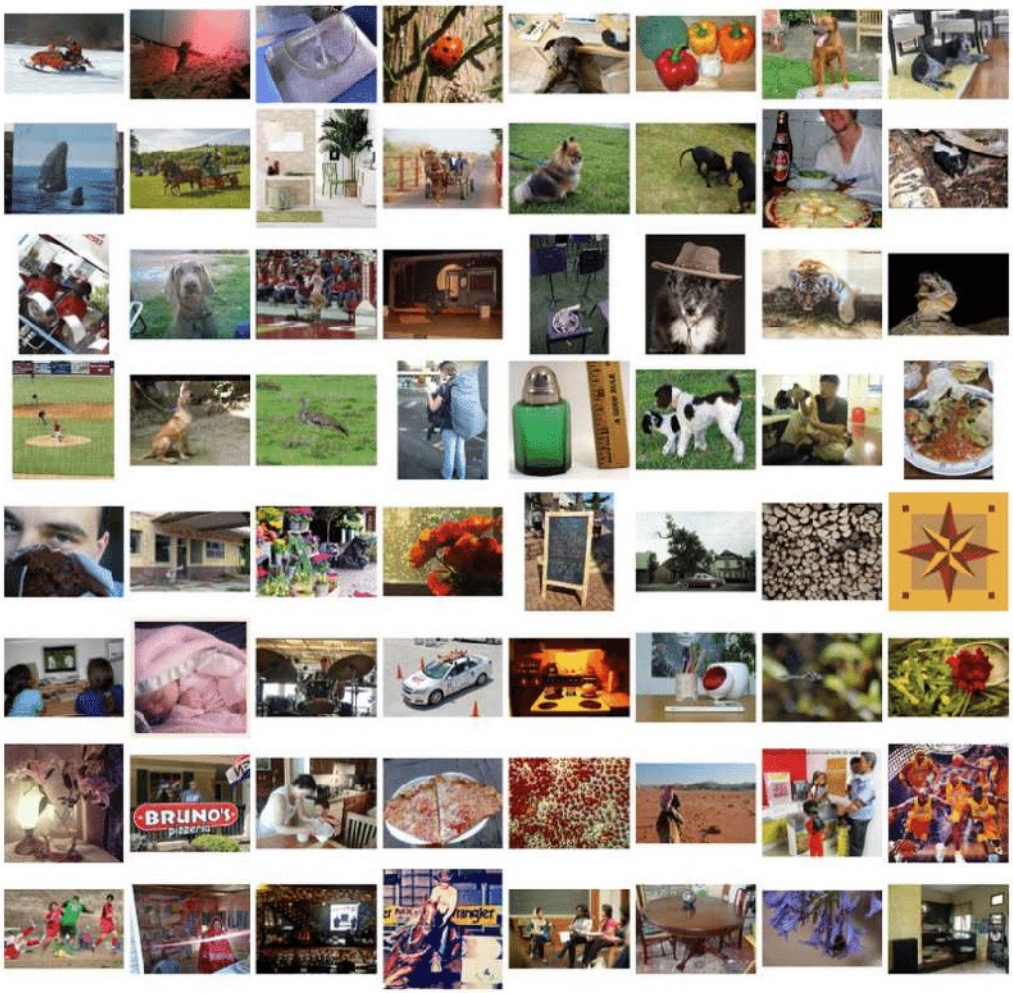
Face recognition

Handwriting recognition





ImageNet (14 million images) – object recognition

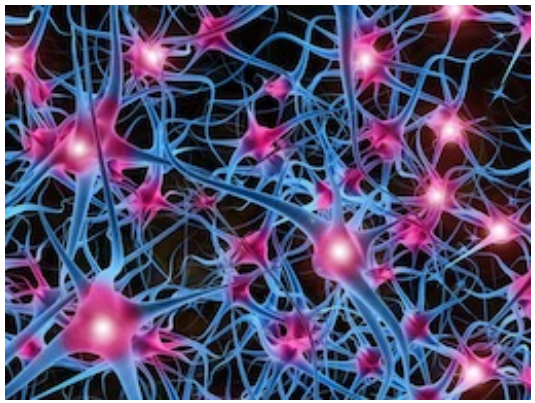


Brain is good in Recognition & Classification

Human brain

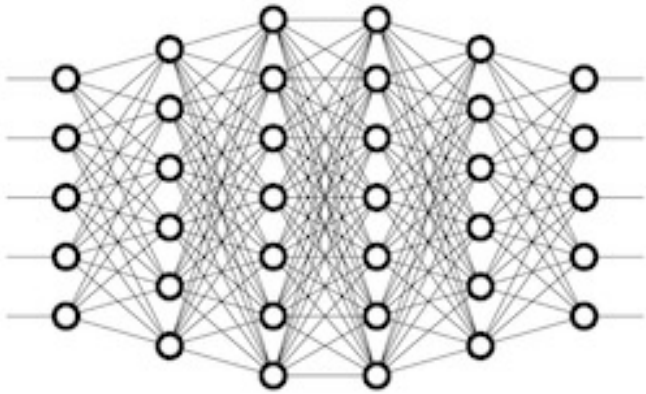
- Weight: 1.5 kg or 2% of total body
- Volume: ≈ ½ sphere with 6.5cm radius
- Consume 20% of our energy

Network of neurons



- 100 billion neurons
- 100 trillion synapses

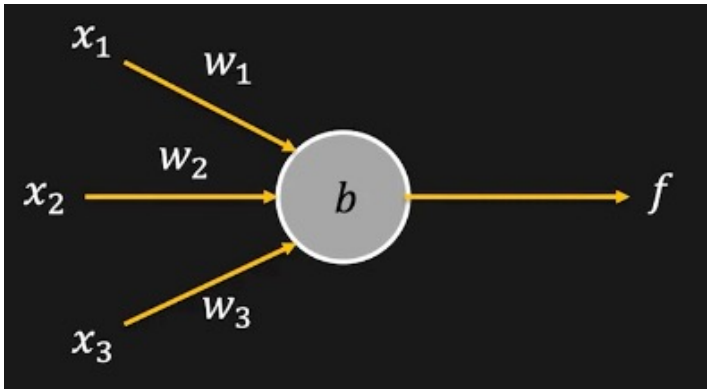
Artificial neural network



- Largest today has 160 billion parameters
- Around 0.16% of human brain

Perceptron

Inputs



Output

$$f = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq -b \\ 1 & \text{if } \sum_j w_j x_j > -b \end{cases}$$

$$f = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases}$$

$w_j = weights,$
$b = threshold\ (or\ bias)$

Source: Rosenblatt

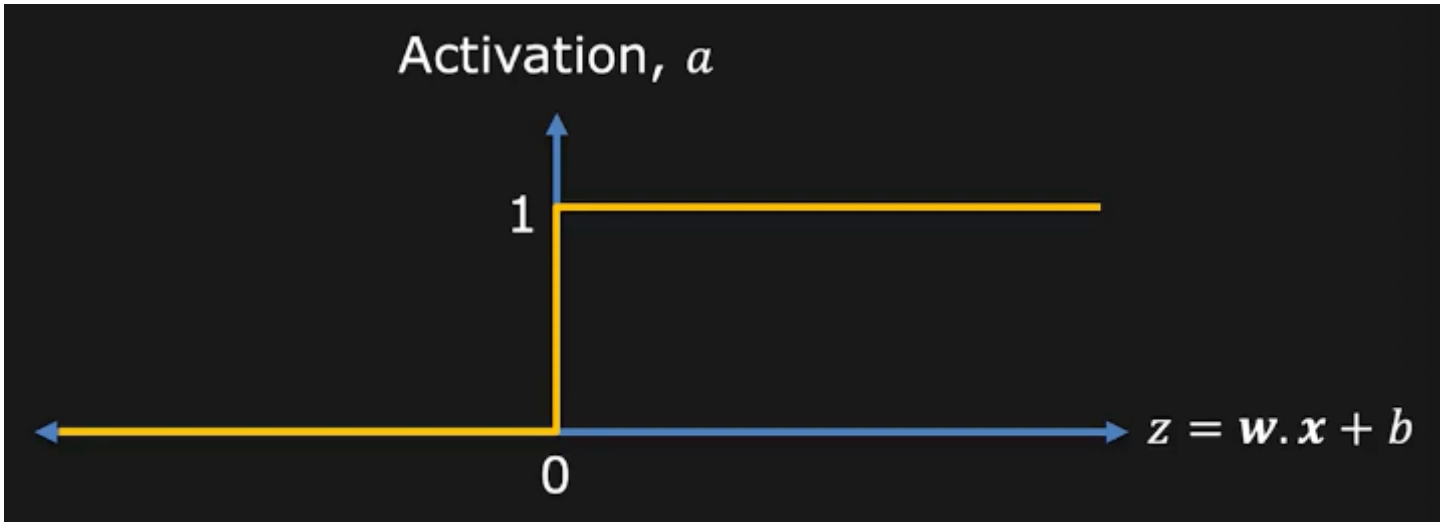
Activation Function of a Perceptron

$$f = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases}$$

Let $z = w \cdot x + b$

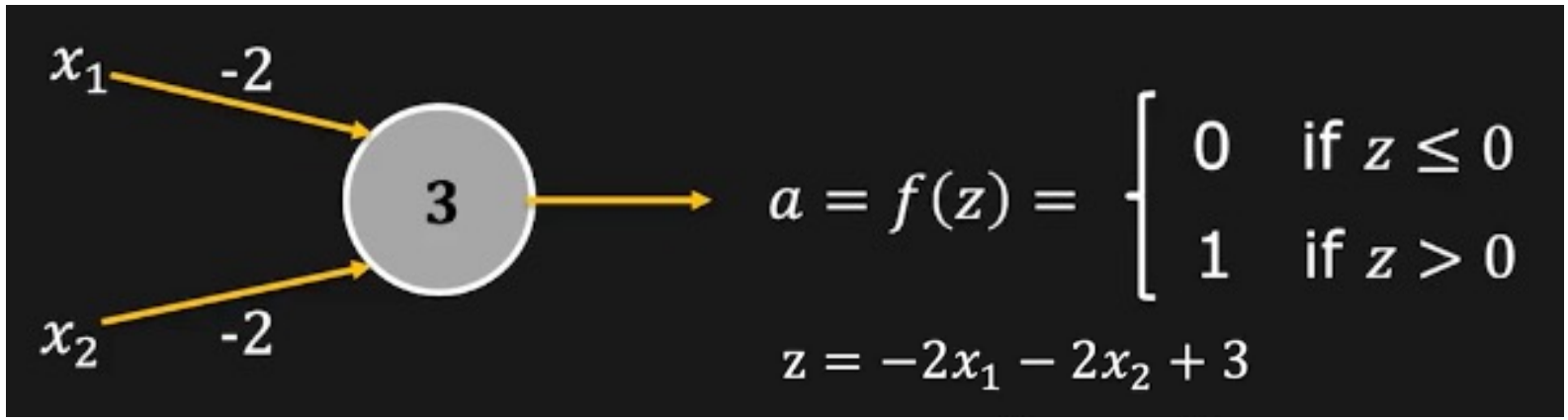
Activation Function is a unity step function $u(t)$

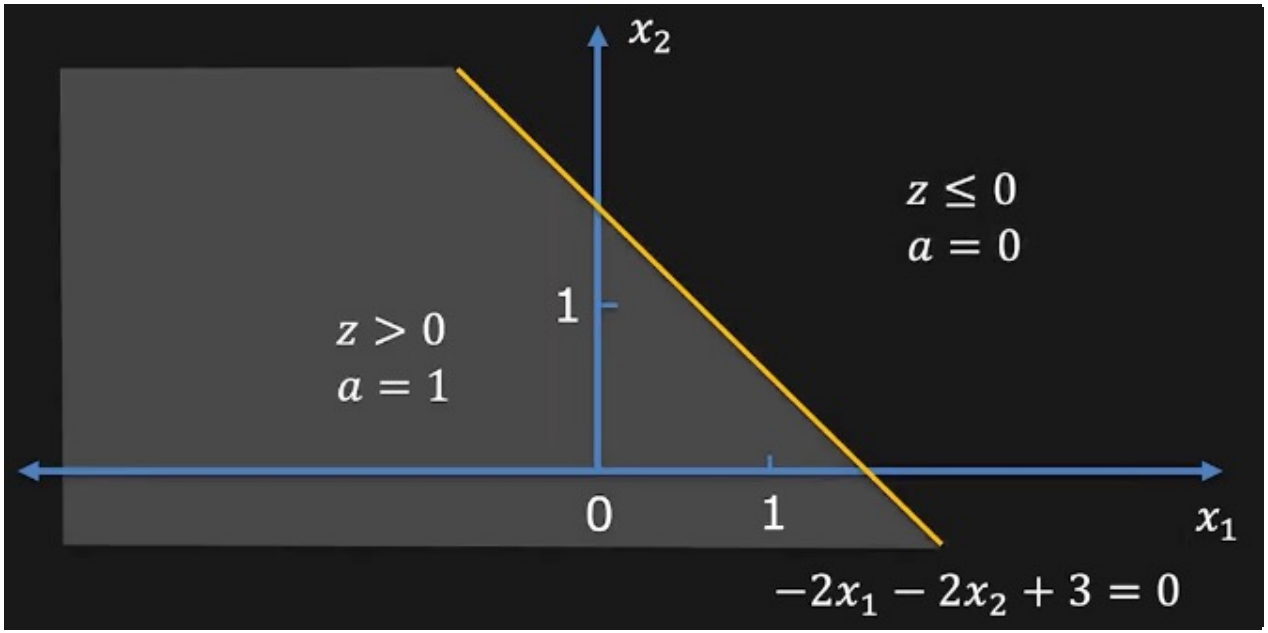
$$a = f(z) = \begin{cases} 0 & \text{if } z \leq 0 \\ 1 & \text{if } z > 0 \end{cases}$$



Source: Rosenblatt

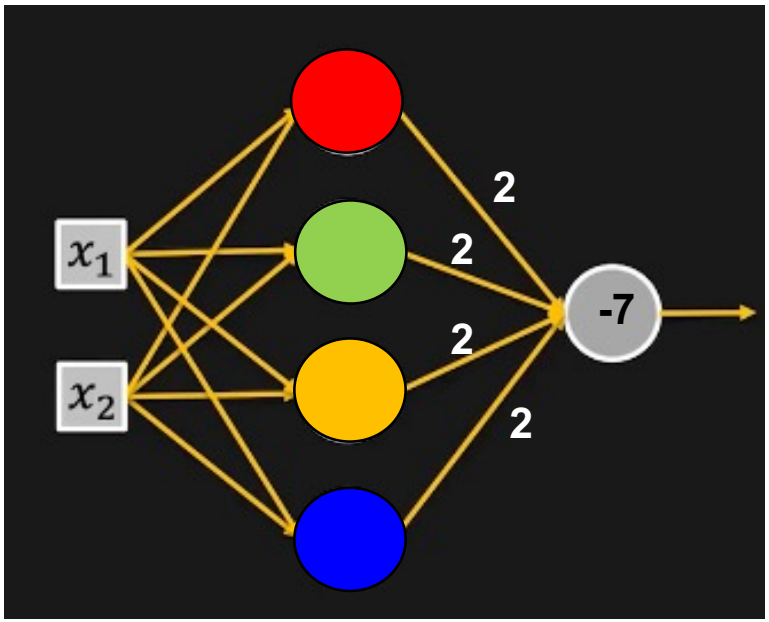
Perceptron is a Linear Classifier

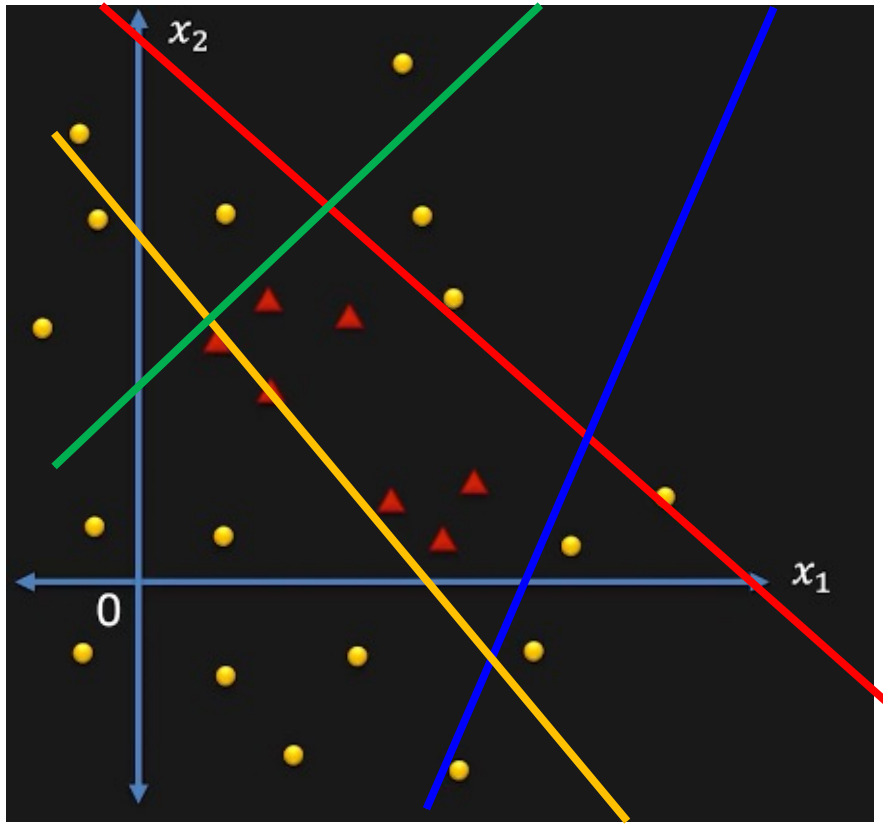




Source: Rosenblatt

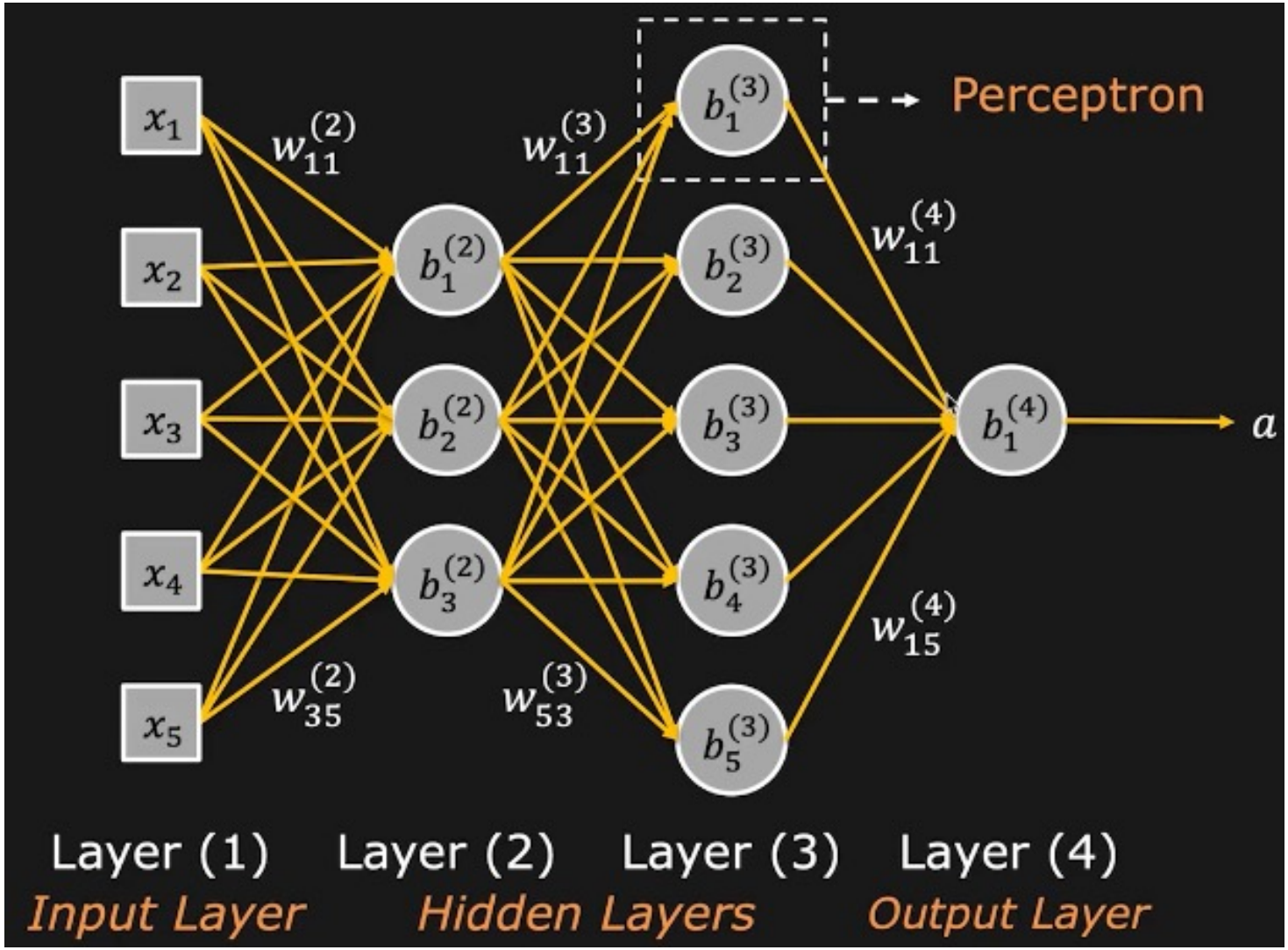
A Network of Perceptrons



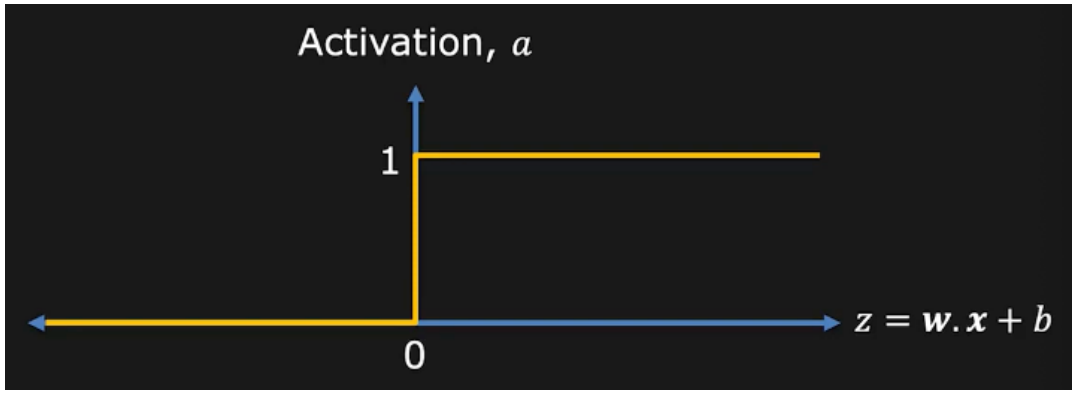


- We can use multiple layers of perceptrons to build a complex classifier.

A Multi-layer Perceptron

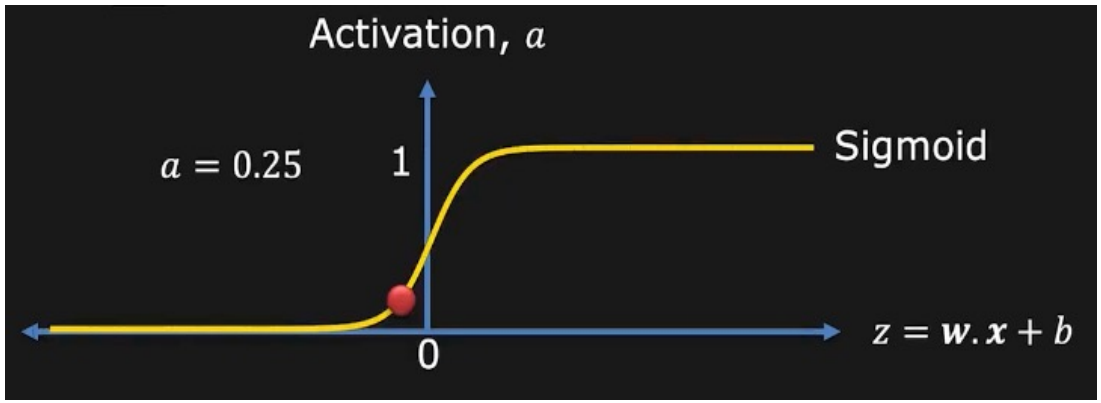


Sigmoid Activation Function



$$a = f(z) = \begin{cases} 0 & \text{if } z \leq 0 \\ 1 & \text{if } z > 0 \end{cases}$$

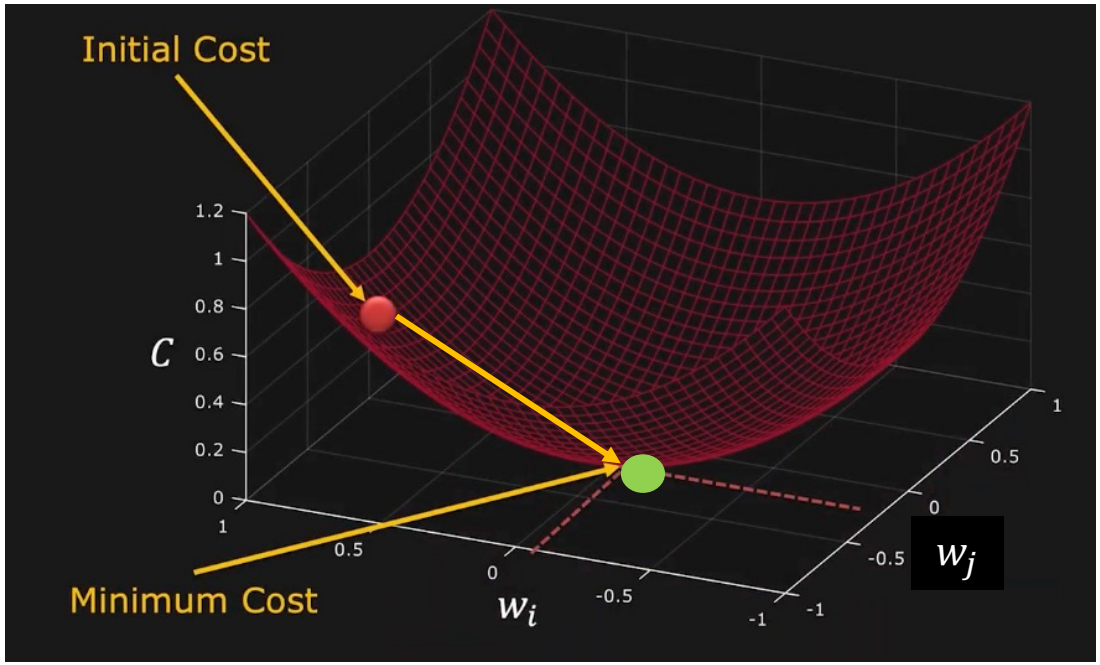
- Step function activation causes output to change abruptly.
- 1st derivatives → ∞.
- Potential for unstable network.



$$a = \sigma(z) = \frac{1}{1 + e^{-z}}$$

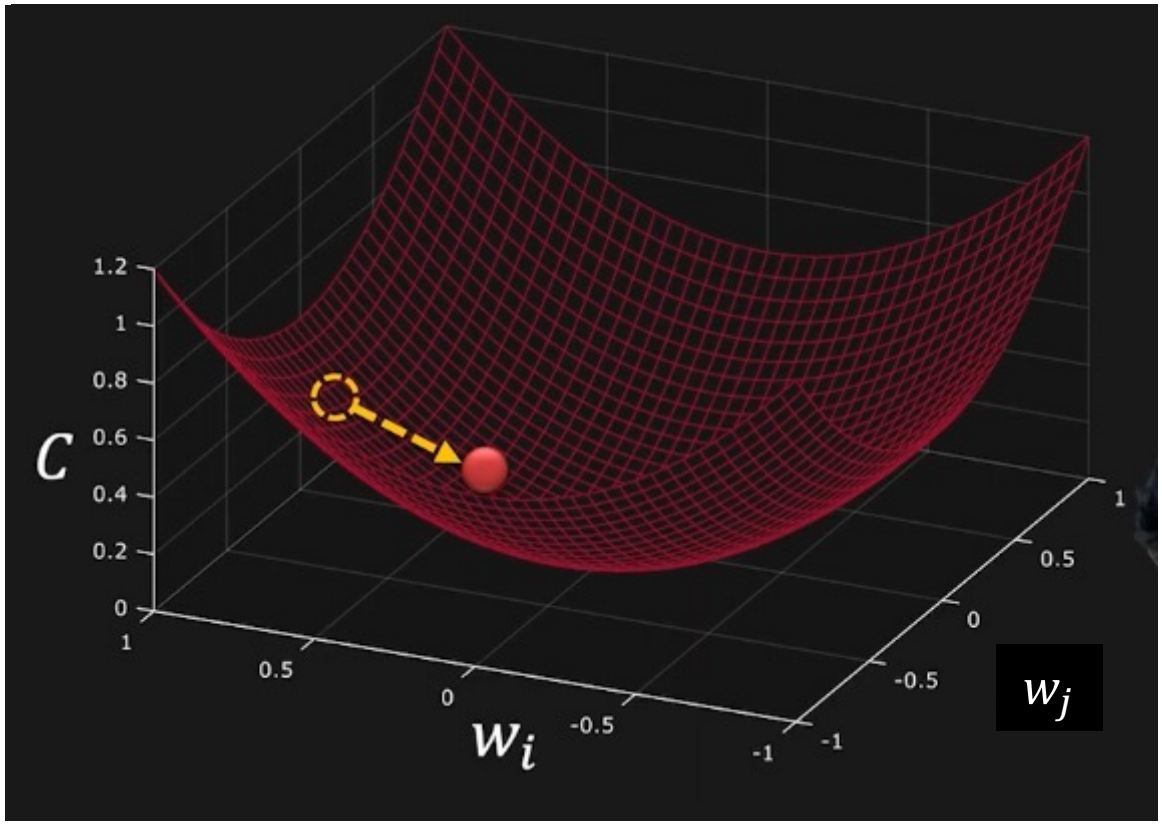
- Sigmoid function activation has a gently transition.
- Function good for differentiation.
- Output changes smoothly with changing weights and threshold.
- Allow backpropagation training.

Optimize Weights in Neural Network - Training



- Gradient of C with respect to $[w_i, w_j]$ is ∇C, where $\nabla C = \begin{bmatrix} \frac{\partial C}{\partial w_i} & \frac{\partial C}{\partial w_j} \end{bmatrix}$
- If there is a change in weight of: $\Delta w = \begin{bmatrix} \Delta w_i \\ \Delta w_j \end{bmatrix}$
- The change in C will be: $\Delta C = \nabla C \cdot \Delta w = \begin{bmatrix} \frac{\partial C}{\partial w_i} & \frac{\partial C}{\partial w_j} \end{bmatrix} \begin{bmatrix} \Delta w_i \\ \Delta w_j \end{bmatrix}$

Training Network Weights by Gradient Descend



Let $\Delta w = -\eta \nabla C$

where η is the learning rate.

For each step:

$$w_i \rightarrow w_i' = w_i - \eta \frac{\partial C}{\partial w_i}$$

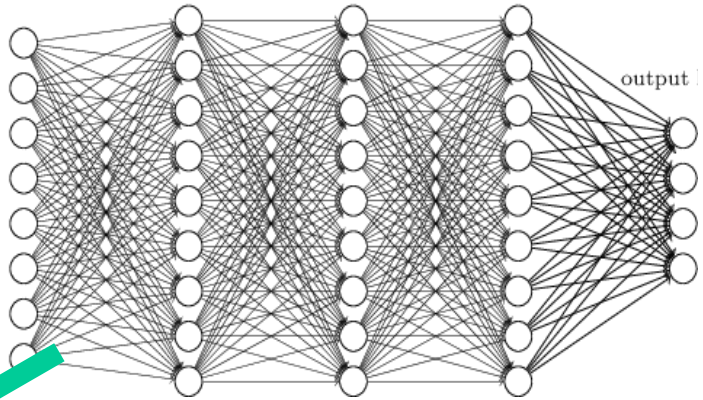
$$w_j \rightarrow w_j' = w_j - \eta \frac{\partial C}{\partial w_j}$$

- $\frac{\partial C}{\partial w}$ can be efficiently computer computed using the backpropagation algorithm.

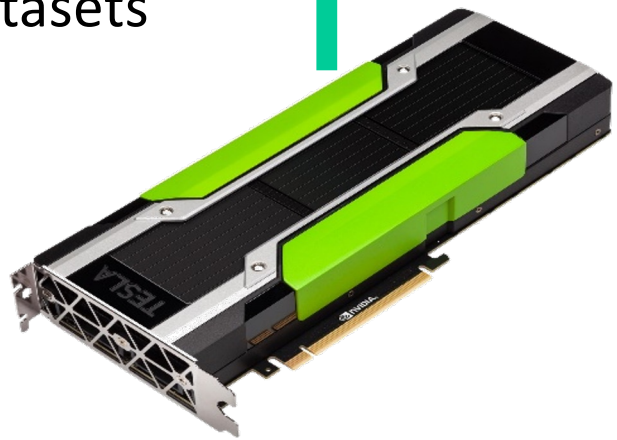
Modern Neural Network System for Visual Data

ImageNet

Big labeled datasets



Deep learning



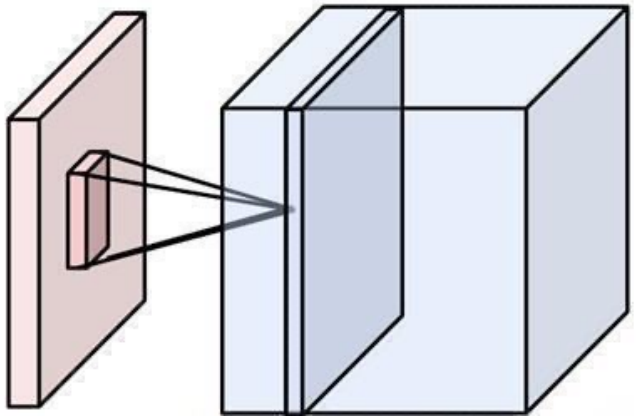
GPU technology

Source: Jayaraman

ImageNet

- Dataset containing ~14 million images in 20,000 classes.
- Manually labelled with name of main object.
- Images gathered from internet.
- Used for training neural networks.

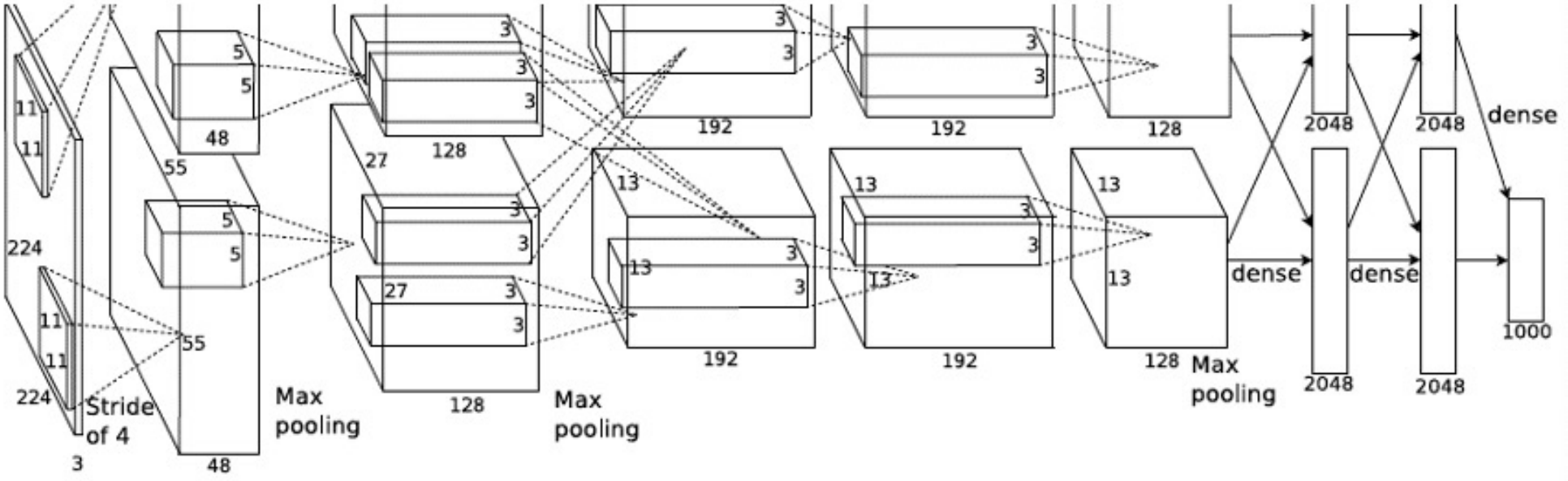
Convolutional Neural Network (CNN)



- Convolutional Neural Network (CNN):
  - Multi-layer network.
  - Use local connectivity, i.e. neurons feed from small group of neural in previous layer.
  - Weight parameters are shared across spatial positions by training shift-invariant filter kernels.
  - Popular in image recognition.

Source: Karpathy

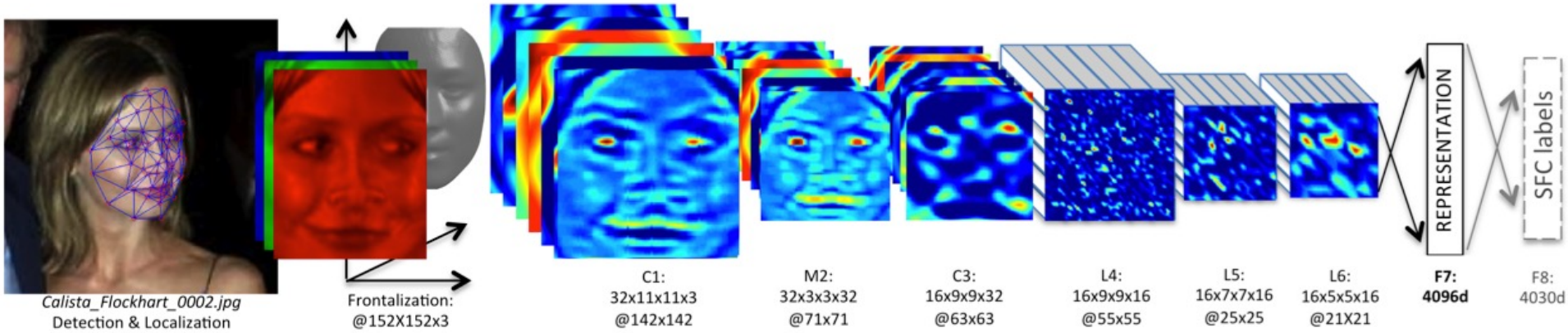
Popular Neural Network - AlexNet



- Very large CNN model with: 7 hidden layers, 650k neurons and 60 million parameters.
- Trained with 1 million images.
- Training ran for a week on two GPUs.

Source: Alex Krizhevsky

Popular Neural Network - DeepFace



- 9-layer network.
- >120 million parameters.
- Trained on 4 million facial images.
- Achieve accuracy of 97.35%.

Source: Taigman

Pre-trained Neural Network on Matlab

Neural Network	Depth	Size	Parameters (Millions)	Image Input Size
squeezenet	18	5.2 MB	1.24	227-by-227
googlenet	22	27 MB	7	224-by-224
inceptionv3	48	89 MB	23.9	299-by-299
densenet201	201	77 MB	20	224-by-224
mobilenetv2	53	13 MB	3.5	224-by-224
resnet18	18	44 MB	11.7	224-by-224
resnet50	50	96 MB	25.6	224-by-224
resnet101	101	167 MB	44.6	224-by-224
xception	71	85 MB	22.9	299-by-299
inceptionresnetv2	164	209 MB	55.9	299-by-299
shufflenet	50	5.4 MB	1.4	224-by-224
nasnetmobile	*	20 MB	5.3	224-by-224
nasnetlarge	*	332 MB	88.9	331-by-331
darknet19	19	78 MB	20.8	256-by-256
darknet53	53	155 MB	41.6	256-by-256
efficientnetb0	82	20 MB	5.3	224-by-224
alexnet	8	227 MB	61	227-by-227
vgg16	16	515 MB	138	224-by-224
vgg19	19	535 MB	144	224-by-224