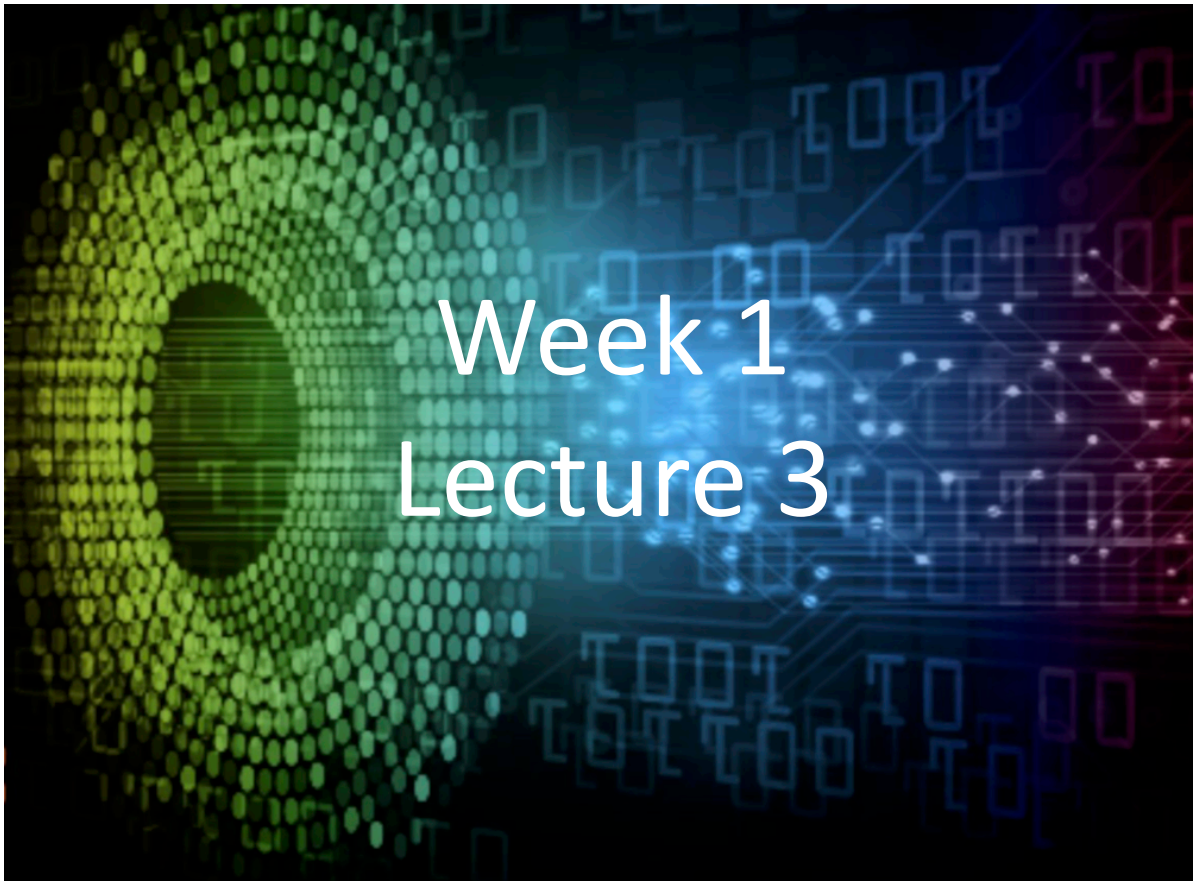
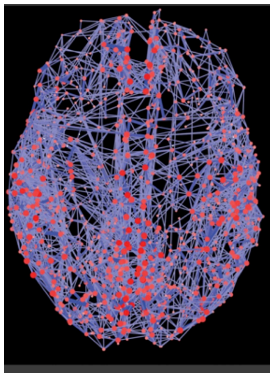
Introduction to Deep Learning
Applications and Theory



ECE 596 / AMATH 563

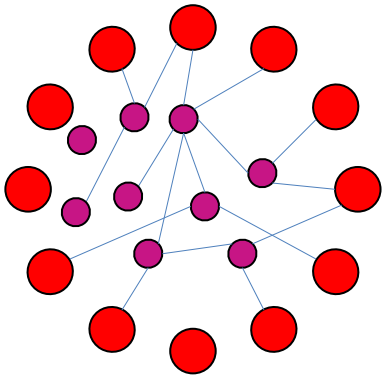
Previous Lecture: Learning from the Brain



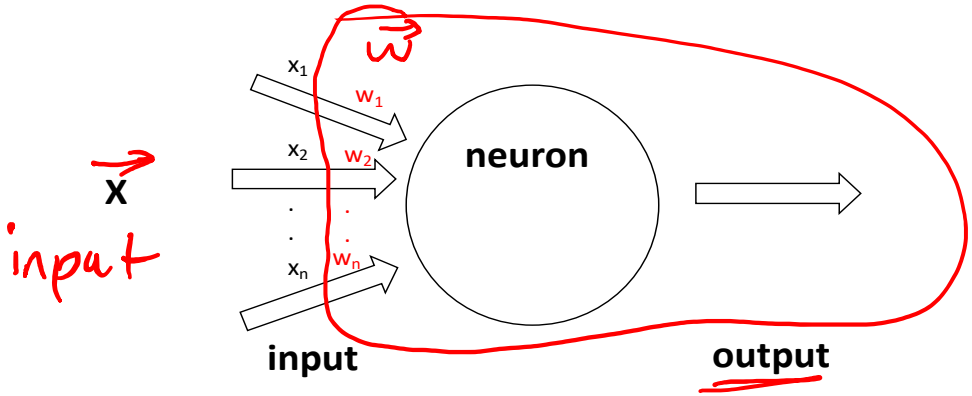
1. Neurons: simple computational units
2. Well connected
3. Parallelism and Recurrence
4. Hierarchically structured
5. Learning
    1. Features
    2. Changing connections

This Lecture: **Neural Networks Fundamentals**

1. Single neural unit
2. Connecting neurons
3. Flow and Recurrences
4. Layers
5. Learning and Data



1. Single Neuron



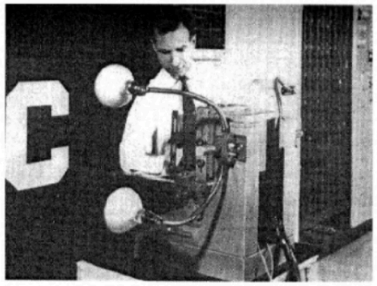
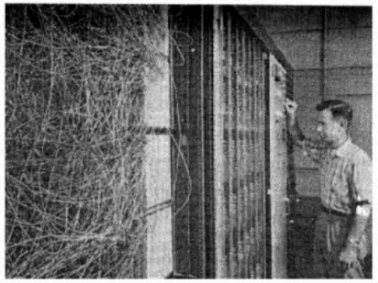
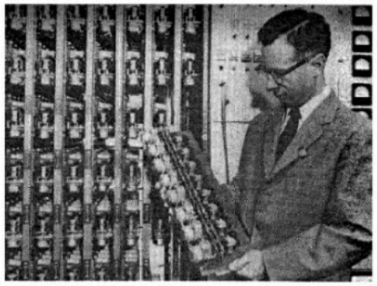
integration	activation	input to other neurons
$I = x_1 * w_1 + \ldots + x_n * w_n$	$f(I)$	$y = f(I)$

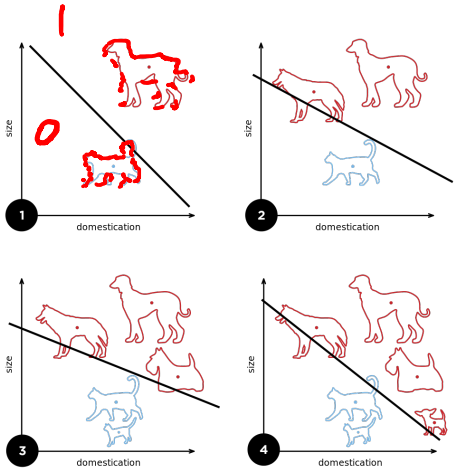
$$\sum_{i=1}^{n} x_i w_i + b \qquad f(\sum_{i=1}^{n} x_i w_i + b) \qquad y = f(\sum_{i=1}^{n} x_i w_i + b)$$

Q1. Why do we need activation?

Q2. To what ML problem we reduce operation of single neuron if we do not include it?

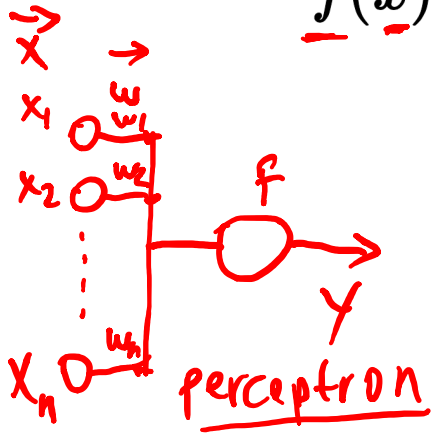
Perceptron

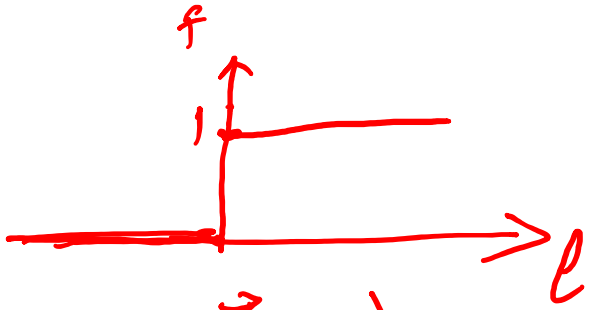


The perceptron, 1957

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



$$y = f(\vec{w} \cdot \vec{x} + b) = \begin{cases} 1 & \ell > 0 \\ 0 & o.w \end{cases}$$



Learning (online):

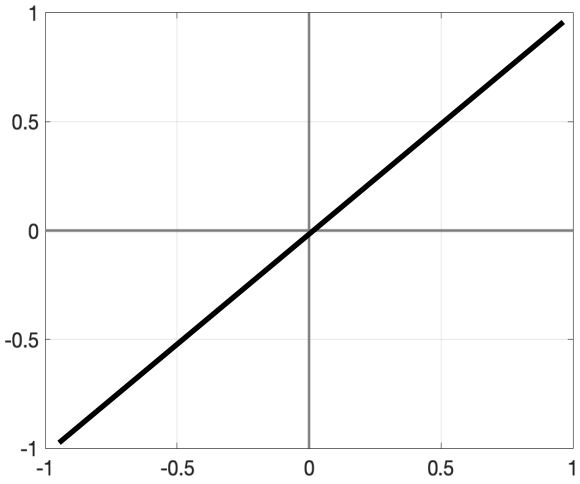
$$k: \quad y_k = f(\vec{w}_k \cdot \vec{x}_k + b_k)$$

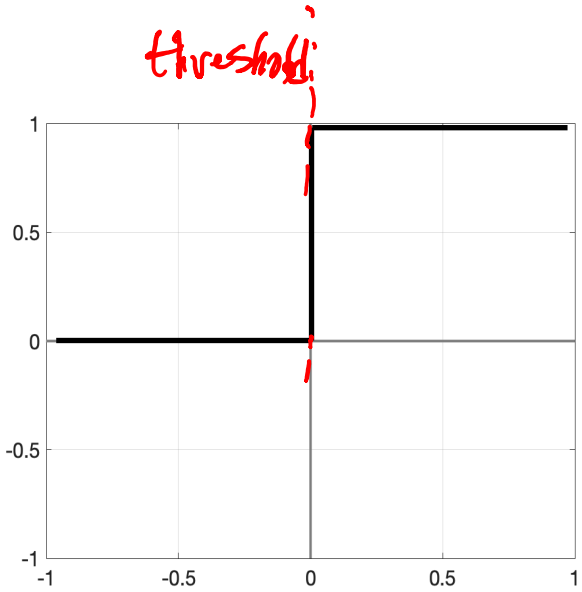
$$D = \{(\vec{x}_1, d_1), (\vec{x}_2, d_2), \ldots (\vec{x}_k, d_k) \ldots\}$$

$$\vec{w}_{k+1} = \vec{w}_k + r(d_k - y_k)\vec{x}_k$$

$$b_{k+1} =$$

Activation





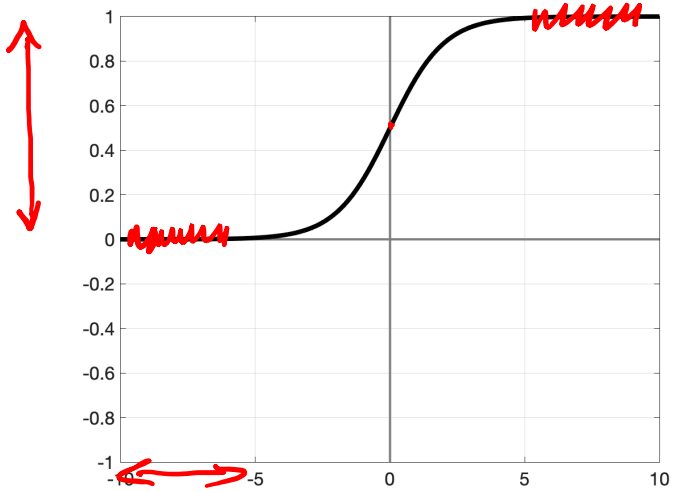
Linear

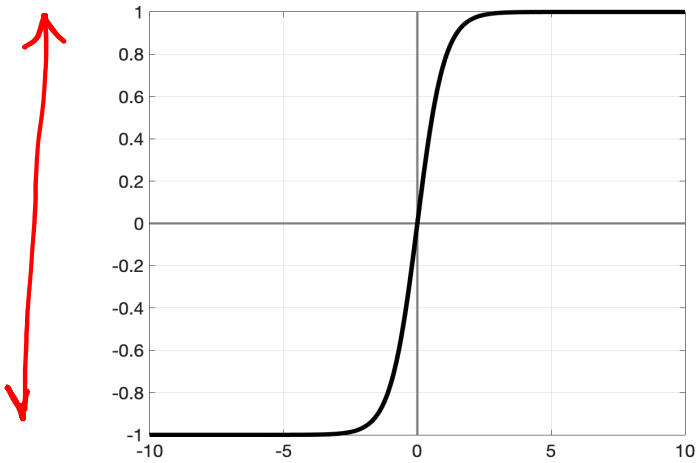
$$f = ax + b$$

Step

$$f(n) = \begin{cases} 1 & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Activation





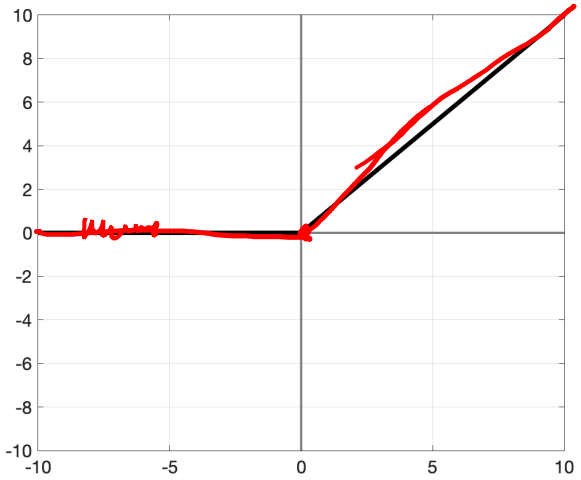
Sigmoid

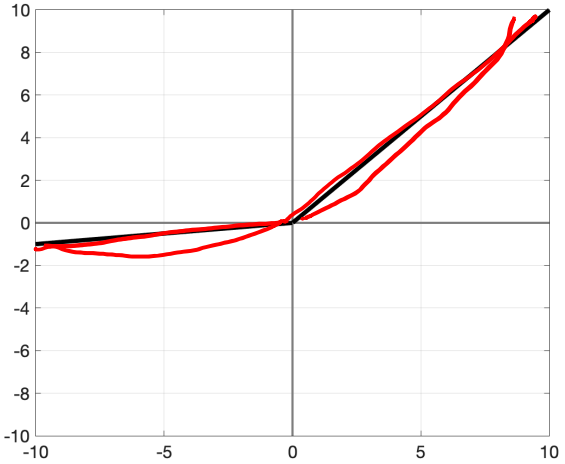
$$f = \frac{1}{1 + e^{-x}}$$

Tanh

$$f = \tanh(x)$$

Activation





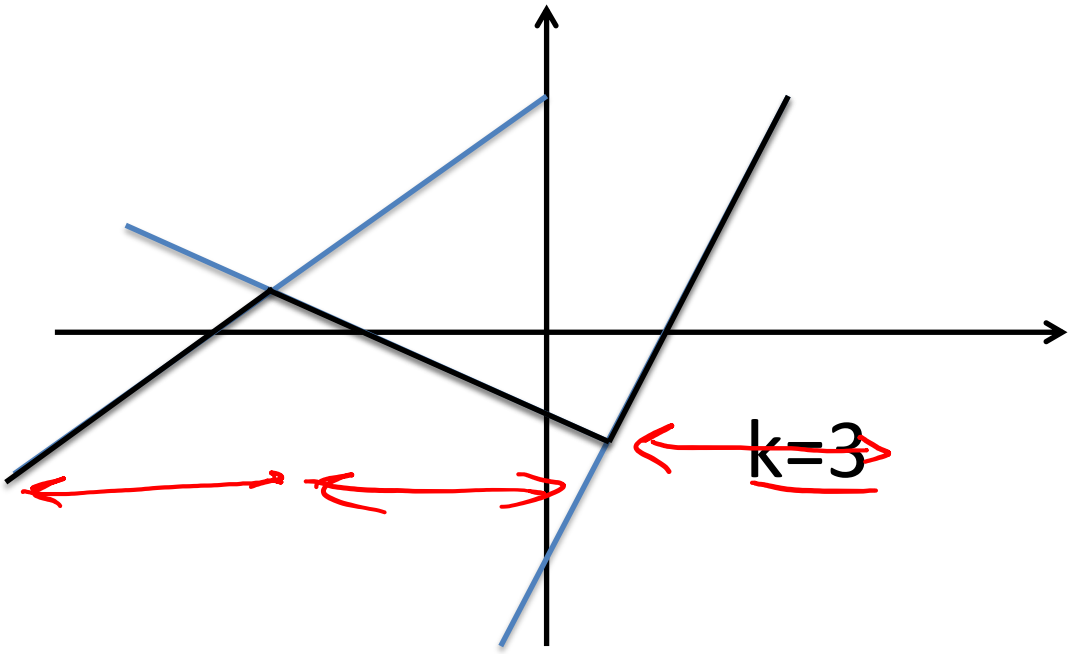
ReLU: Rectified Linear Unit

$$f = \max(x, 0)$$

Leaky (parametric) ReLU

$$f = \max(x, ax)$$

Activation



Maxout

$$f = \max(w_1^T x + b_1, w_2^T x + b_2)$$ k=2

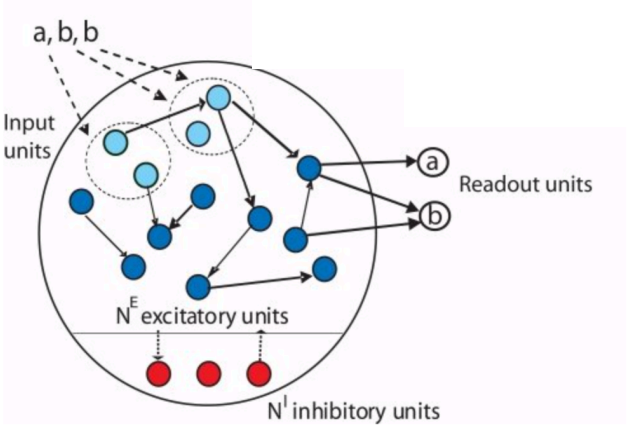
Questions

Q3. Which activation (if any) the two class perceptron presented here is implementing?

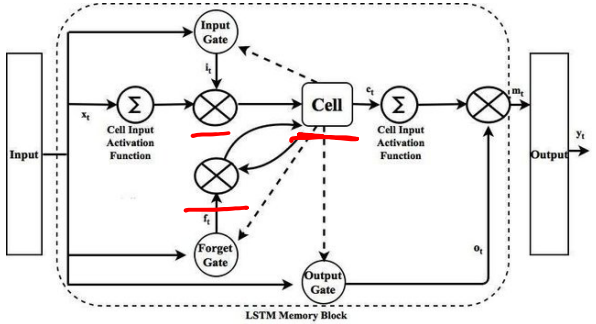
Q4. What are the aspects that need to be considered when choosing activation functions?

Q5. What could be a limiting factor in using maxout with large k?

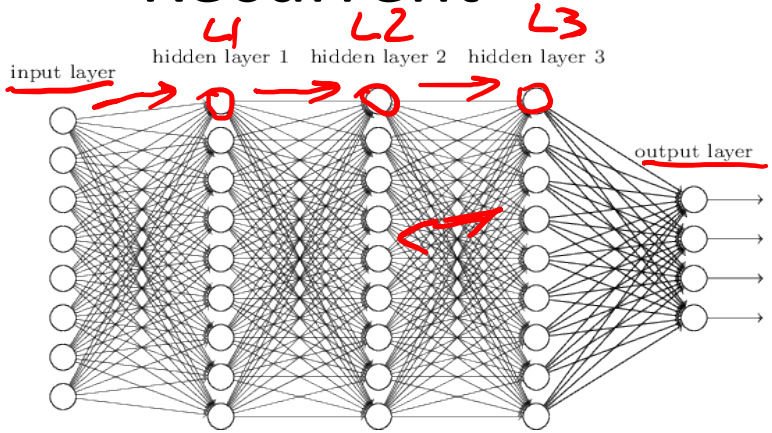
2. Connecting Neurons



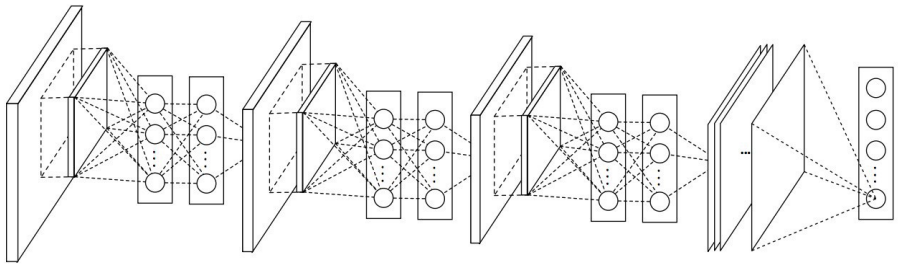
Recurrent



Gates



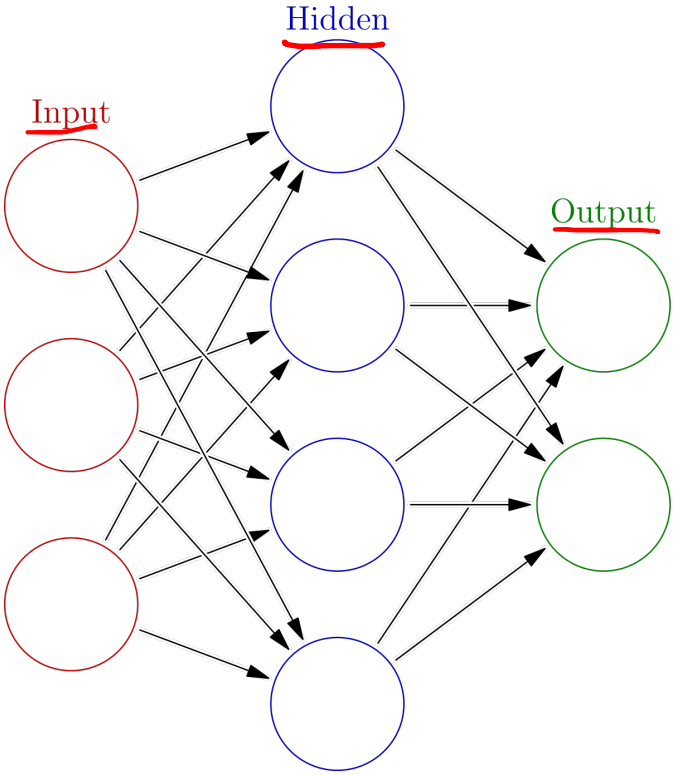
Layered



Local Layered

Q6. List as many aspects as possible that layered architecture could be useful?

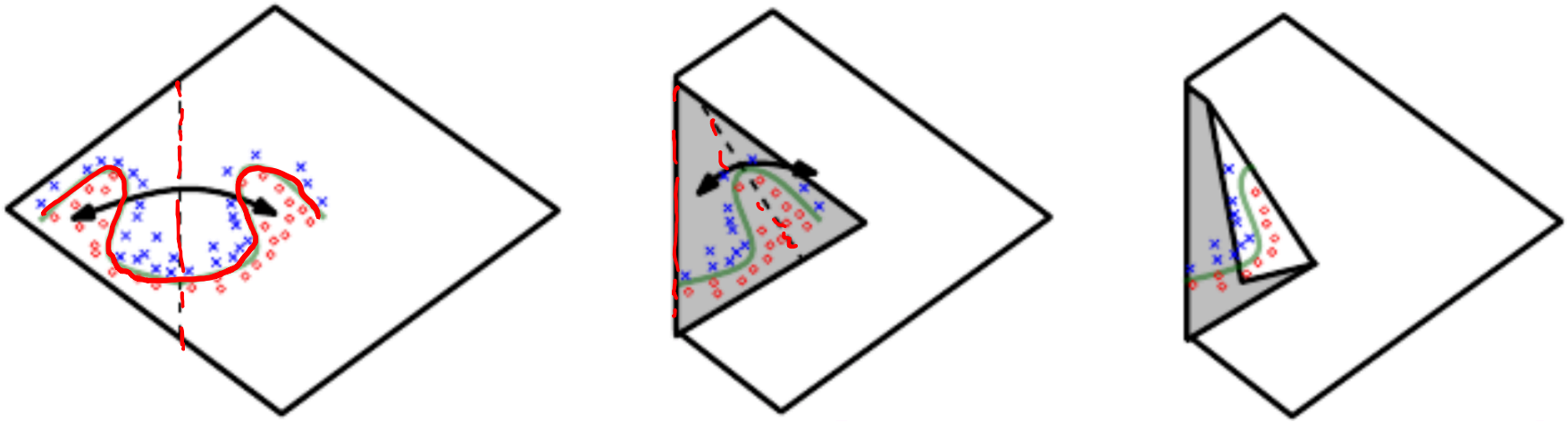
Universal Approximation



Hornik *et al.* 1989; *Cybenko* 1989 – Univ. App. Thm

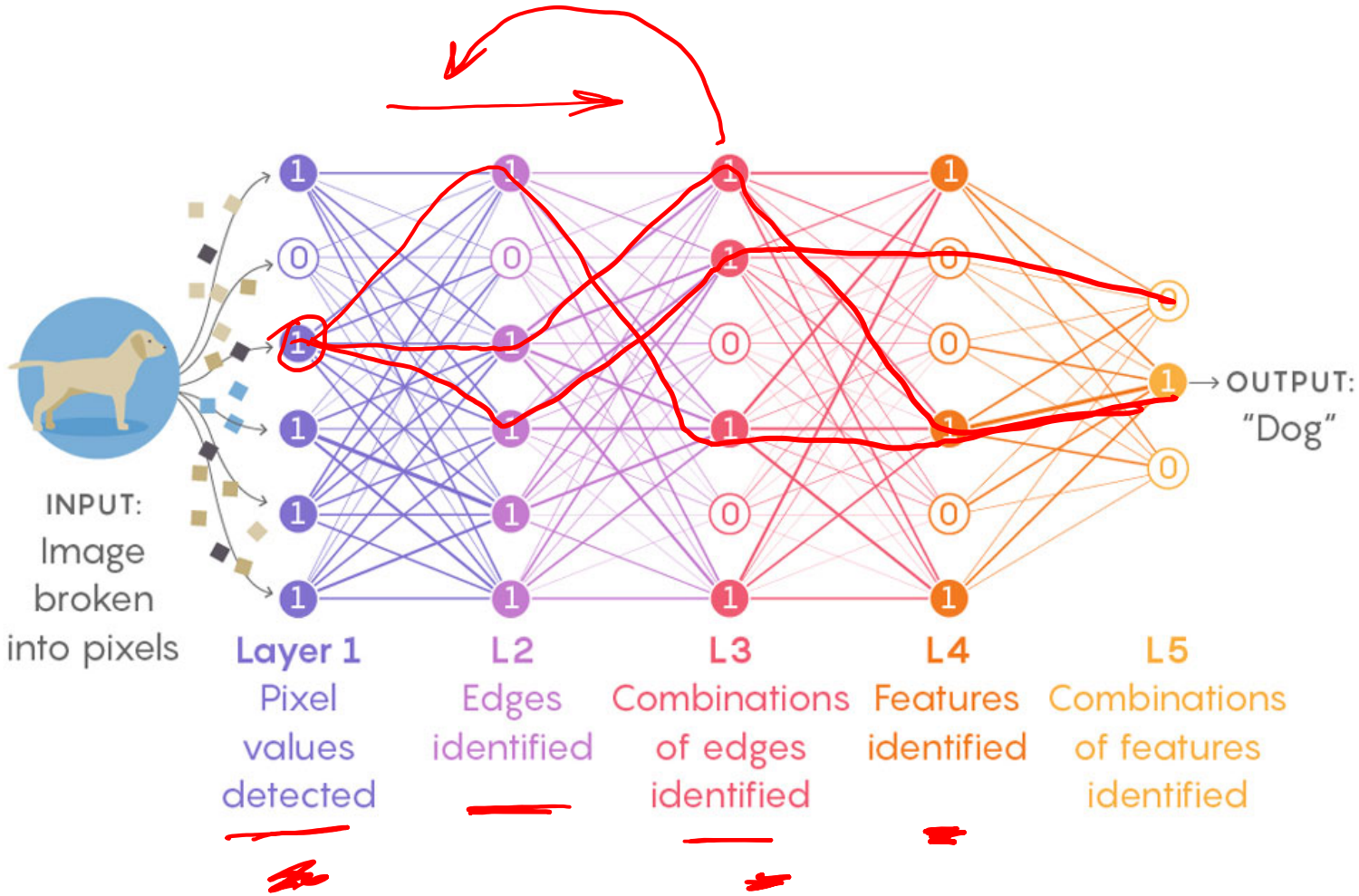
Barron 1993 – Exponential Upper Bound

Architecture that directs the flow

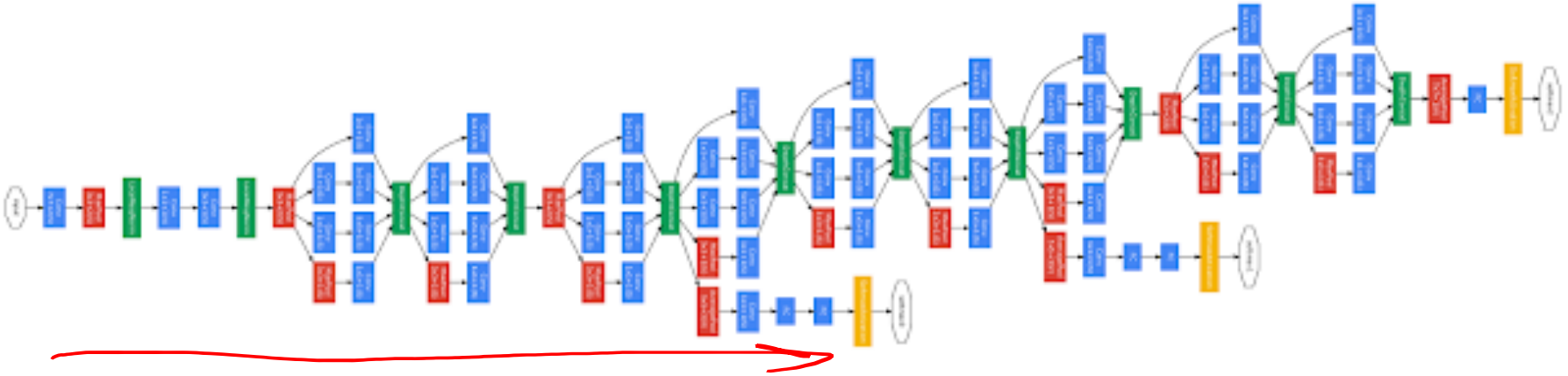


Q7*. Is there a limit to the number of layers that one can choose? Why yes or why not?

3. Flow/Pathways/Layers

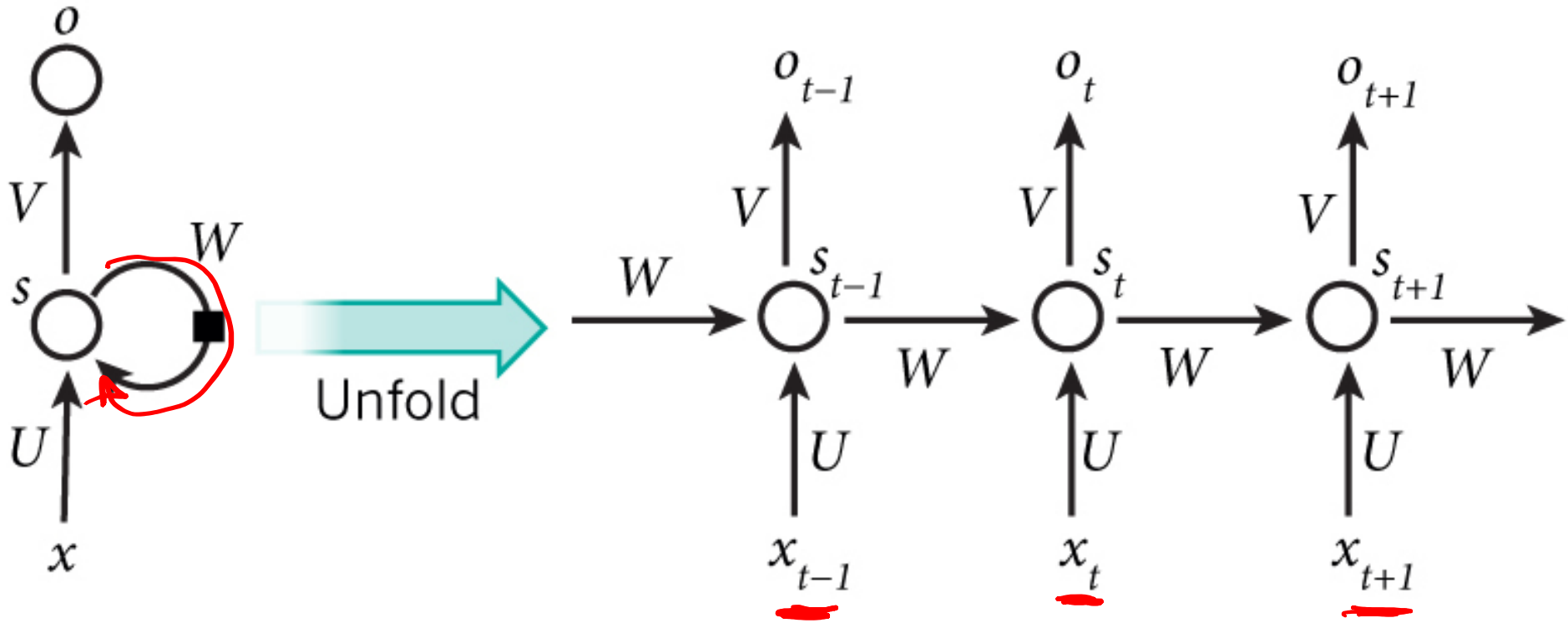


Feed Forward Deep Layered



Google Net

RNNs



Three Pillars

5. Learning

- Supervised Learning
  - Cost function
  - Labeled Data
- Unsupervised Learning
  - Generative Learning
  - Latent Representation
- Reinforcement Learning
  - Rewards
  - Optimization through Control

Supervised Learning

$X \in \mathbb{R}^n$

input

$D : \{(X^{(1)}, Y^{(1)}), \ldots, (X^{(m)}, Y^{(m)})\}$

training set

typically stacked

test set, validation set

$Y \in \{0, 1\}$

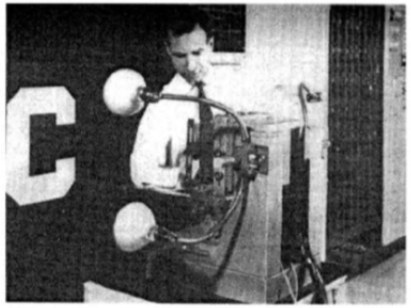
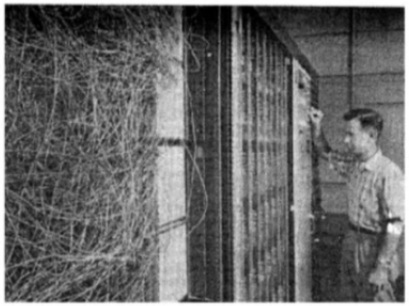
$Y \in \{0, 1\}^k$

classification

$Y \in \mathbb{R}^k$

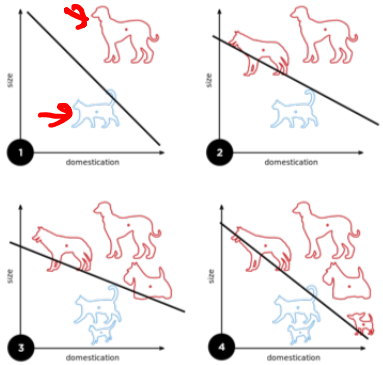
synthesis

Learning Perceptron


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

Network trains its weights based on **labeled training** data, i.e. optimizes **cost** function between expected outputs and actual outputs



The perceptron, 1957

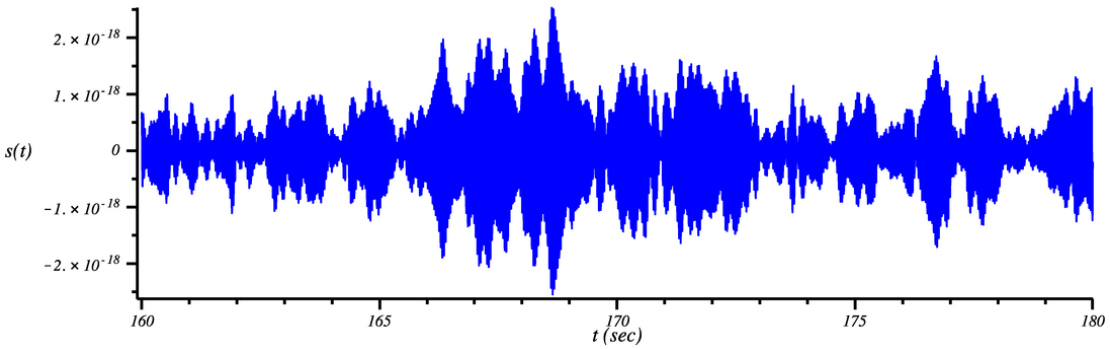
Structured Data

Choosing X and Y

Keyword	#appearances	Interest %	clicks	….
shoes	5B	80	1M	
neuralnetwork	10B	99.9	1B	…

Age	keyword1	keyword2	…
25	neuralnetwork	gameofthrones	
30	neuralnetwork	shoes	596B

Unstructured Data



audio

images

videos

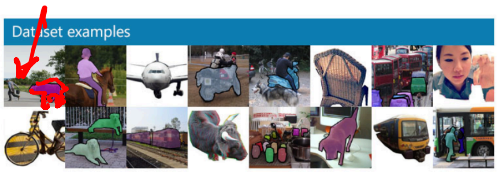
"Hello, congratulations
you have won $100000000"

natural language

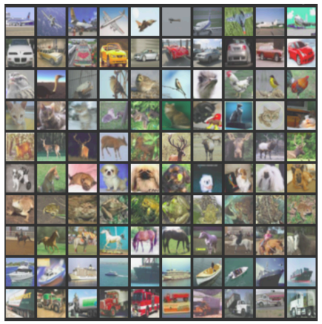
Datasets

MNIST

ImageNet



MS COCO



CIFAR10

IMDB Reviews

Free Music Archive

kaggle

Q8. Which additional benchmark datasets related to you interests can you find?

Next

- Training and Learning
  - Logistic Regression Example
  - Optimization
    - Gradient Descent
    - Stochastic Gradient Descent
    - Extensions