Deep learning

An introduction to artificial neural networks

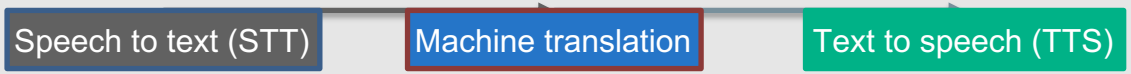
Agenda

- Deep Learning in Action & History
- Neural Network
- Deep Learning: CNN and RNN

Deep Learning in Action

- The event happened on Nov 8, 2012

Speech Recognition Breakthrough



Deep Learning is the hero behind the curtain

Fast forward 5 years

- AlphaGo and AlphaGo zero

Mastering the game of Go without human knowledge

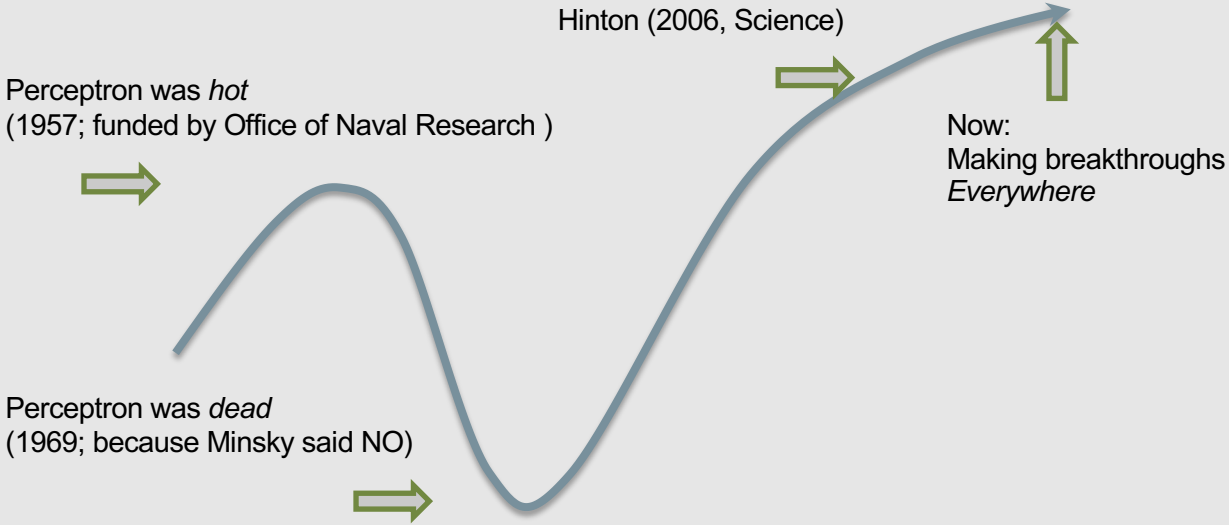
David Silver[1]*, Julian Schrittwieser[1]*, Karen Simonyan[1]*, Ioannis Antonoglou[1], Aja Huang[1], Arthur Guez[1], Thomas Hubert[1], Lucas Baker[1], Matthew Lai[1], Adrian Bolton[1], Yutian Chen[1], Timothy Lillicrap[1], Fan Hui[1], Laurent Sifre[1], George van den Driessche[1], Thore Graepel[1] & Demis Hassabis[1]

A long-standing goal of artificial intelligence is an algorithm that learns, *tabula rasa*, superhuman proficiency in challenging domains. Recently, AlphaGo became the first program to defeat a world champion in the game of Go. The tree search in AlphaGo evaluated positions and selected moves using deep neural networks. These neural networks were trained by supervised learning from human expert moves, and by reinforcement learning from self-play. Here we introduce an algorithm based solely on reinforcement learning, without human data, guidance or domain knowledge beyond game rules. AlphaGo becomes its own teacher: a neural network is trained to predict AlphaGo's own move selections and also the winner of AlphaGo's games. This neural network improves the strength of the tree search, resulting in higher quality move selection and stronger self-play in the next iteration. Starting *tabula rasa*, our new program AlphaGo Zero achieved superhuman performance, winning 100–0 against the previously published, champion-defeating AlphaGo.

Deep learning in a nutshell

- A machine-learning approach, based on *artificial neural network*
  - But goes *very*, *very* deep (and wide, too!)

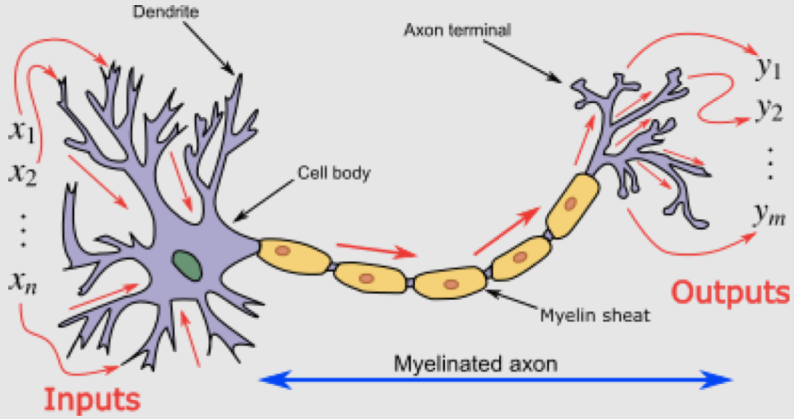
A history of perseverance

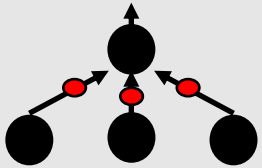


Agenda

- Deep Learning in Action & History
- Neural Network

Brain cell and a(n artificial) neuron



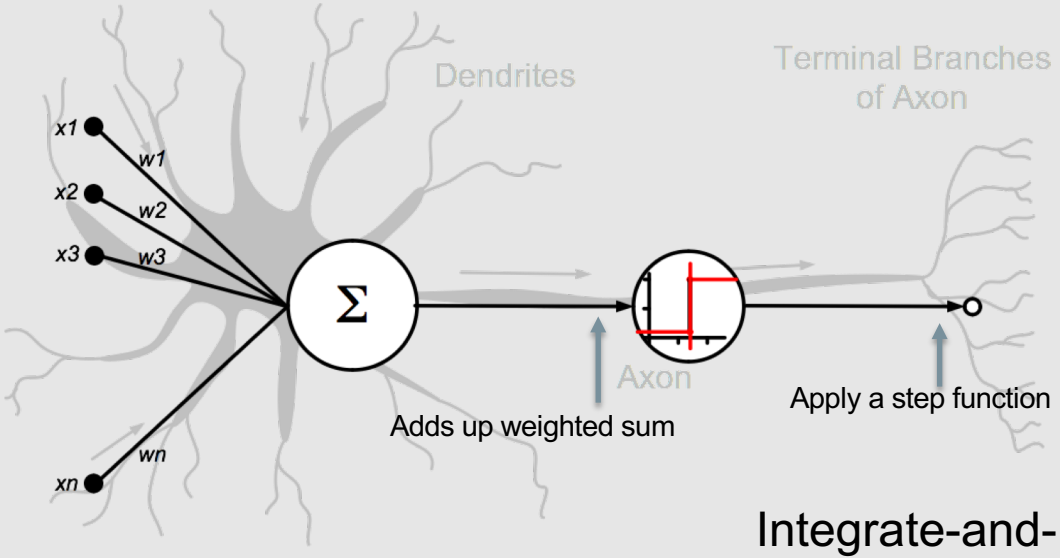


An artificial neuron:

- Receive input
- Change the internal state (*activation*) according to that input
- Produce output depending on the input and activation

Multi-inputs: the neuron

- One neuron (with threshold)



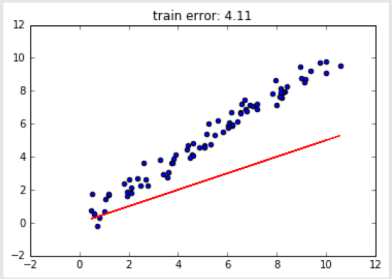
Integrate-and-Fire

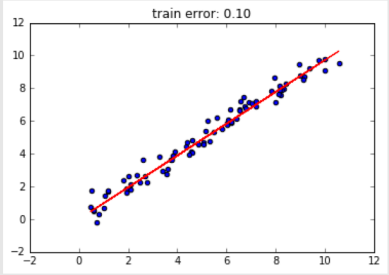
Last lecture: Linear regression

- Model: $y = wx$
- Model parameter: w
- $E(w) = \frac{1}{2n}\sum_{i=1}^{n}(wx_i - y_i)^2$
- **Goal: change w such that E decreases**

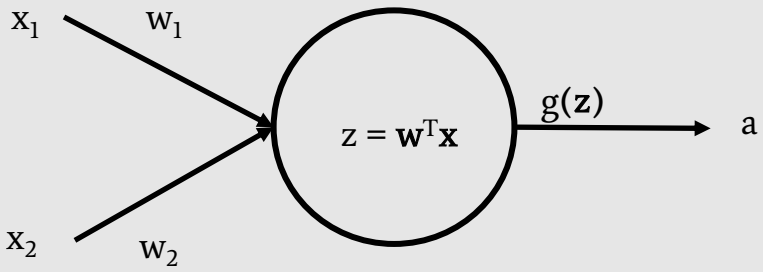
The simplest neural network!

1 input : x
1 output: y





Terminology



A perceptron

$z = \mathbf{w}^T\mathbf{x} = w_1x_1 + w_2x_2+b$

$a = g(z)$

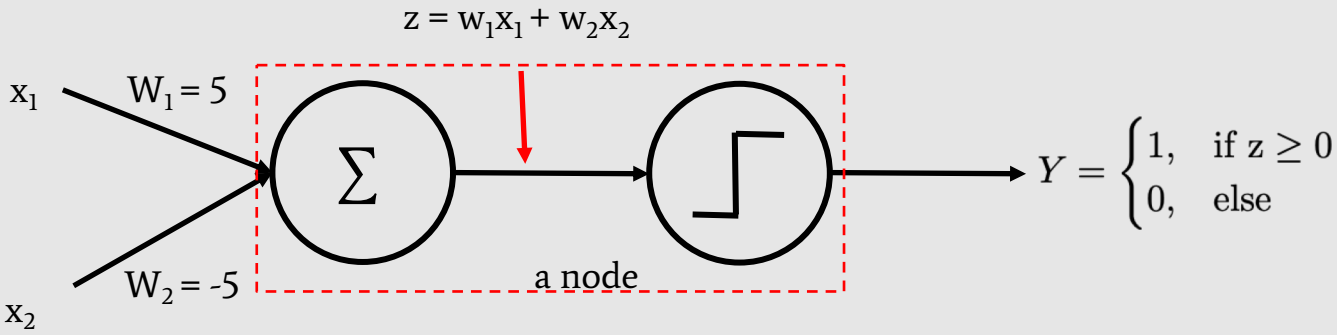
a: output of neuron

g: activation function

w: input weights

b: bias

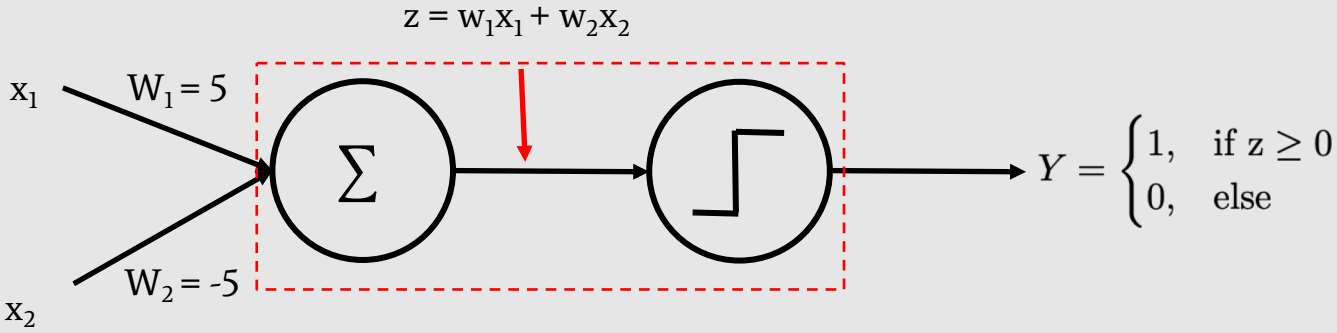
What does this neuron do?



x_1	x_2	z	Y
1	-1	?	?
-1	1	?	?

It likes certain pattern, and hates others.

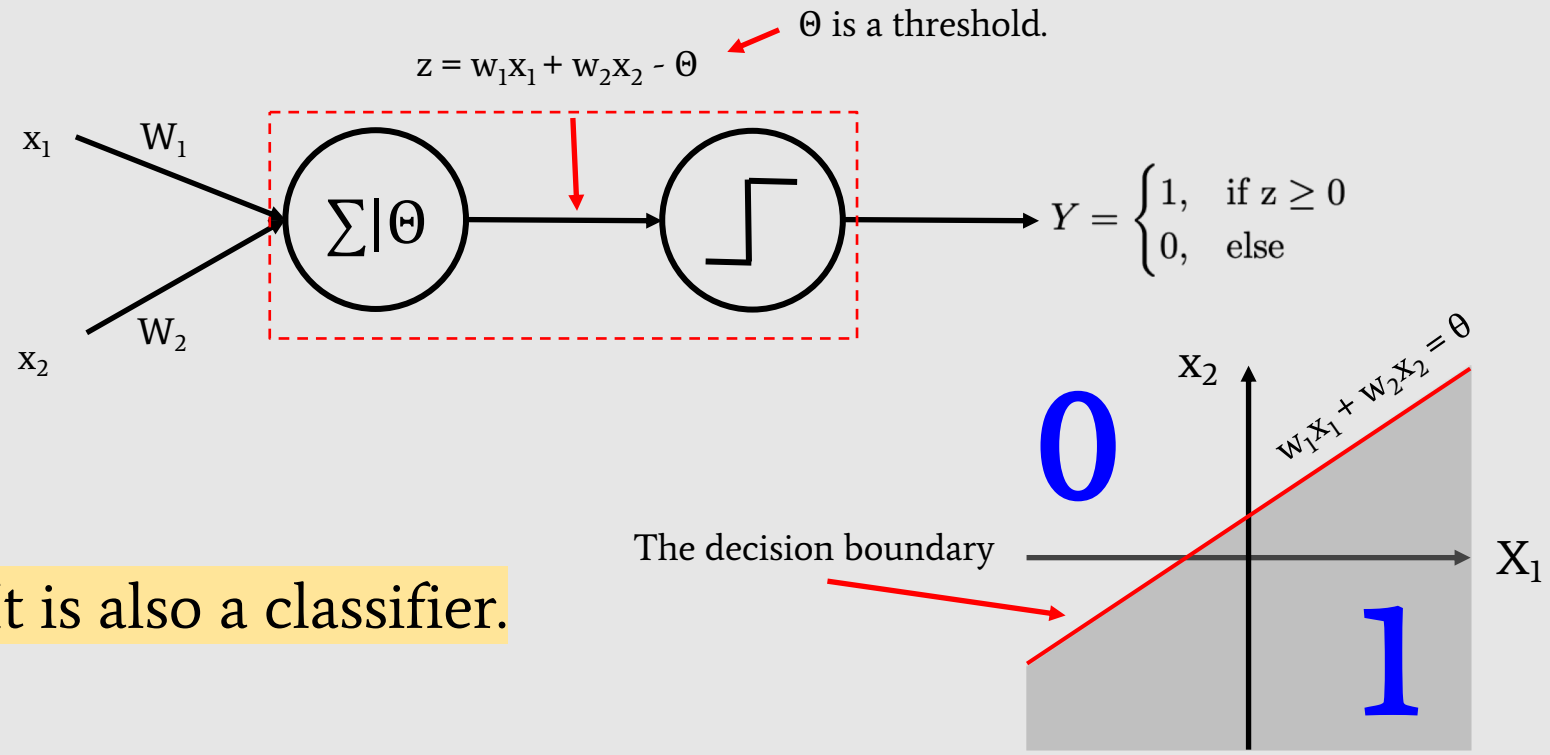
What does this neuron do?



x_1	x_2	z	Y
1	-1	10	1
-1	1	-10	0

It likes certain pattern, and hates others. → It is a detector.

What does this neuron do?

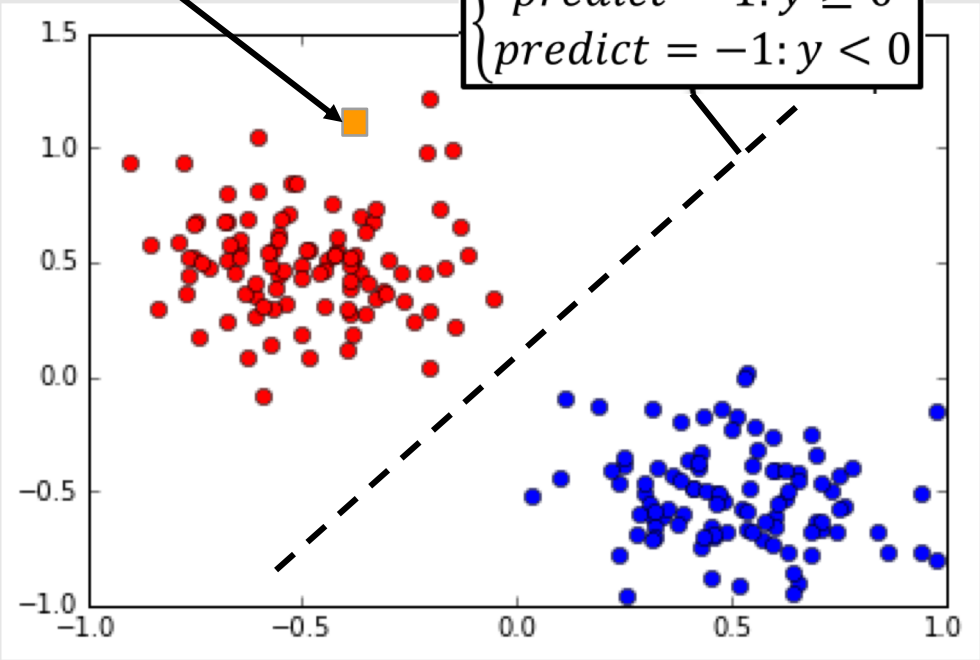


It is also a classifier.

Recall the last lecture: linear classifier

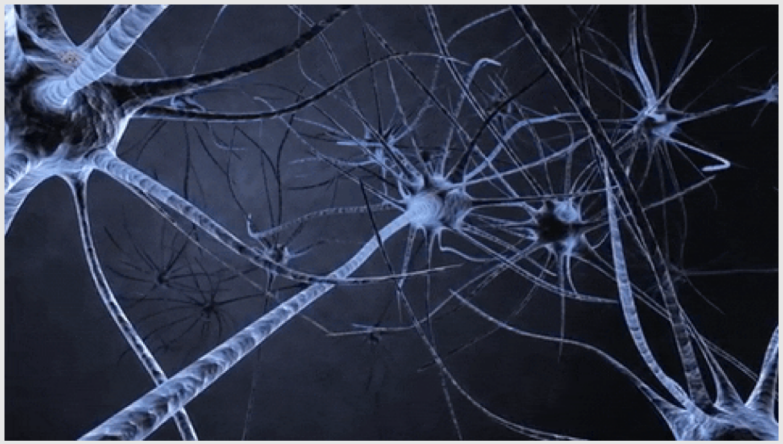
What label does this data has?

$$y = w_1 \times x_1 + w_2 \times x_2$$
$$\begin{cases} predict = 1 : y \geq 0 \\ predict = -1 : y < 0 \end{cases}$$



Model parameter:
w_1, w_2

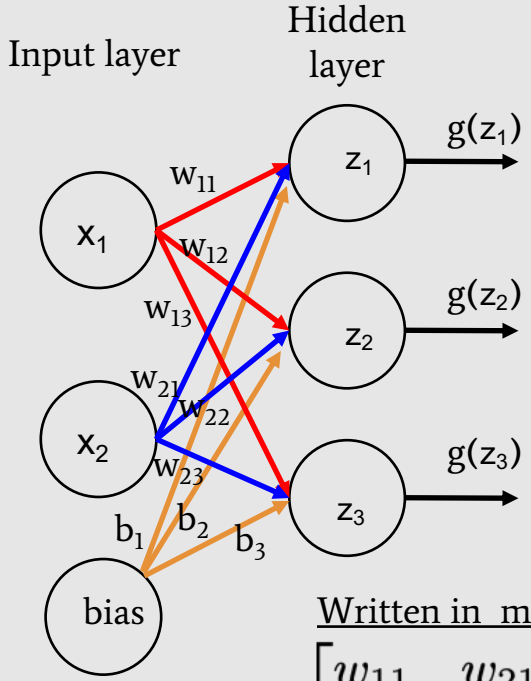
Forming extremely complex networks



A number of connected neurons make up a network. (Connectionism)

Neurons (~10^{11}) → Brain

More general form: linear transformation + a shift



The input of activation function:

$$\mathbf{z} = W^T_{(2\times 3)}\mathbf{x} + \mathbf{b}$$

From linear algebra, we know

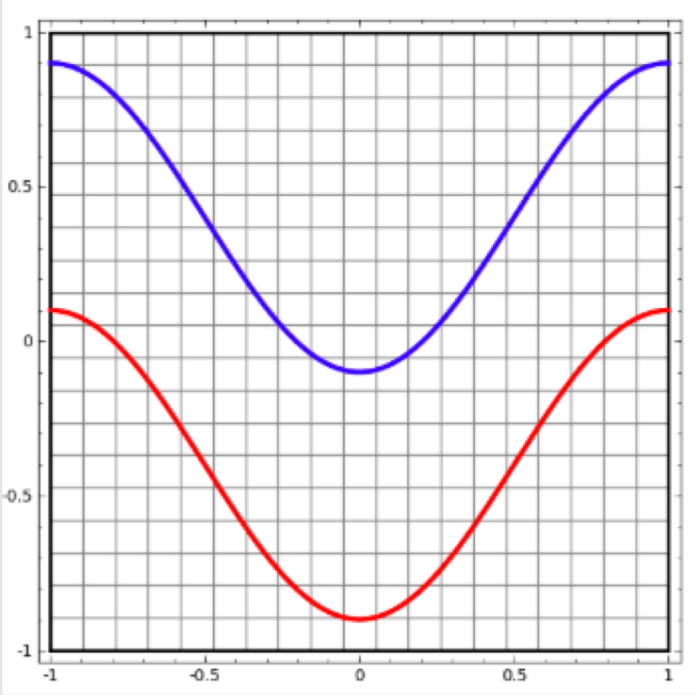
- x: a point in the original space
- y: a point in the target space
- W: stretch and rotate
- b: shift

It performs a linear transformation by w and a translation by b.

Written in matrix:

$$\begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

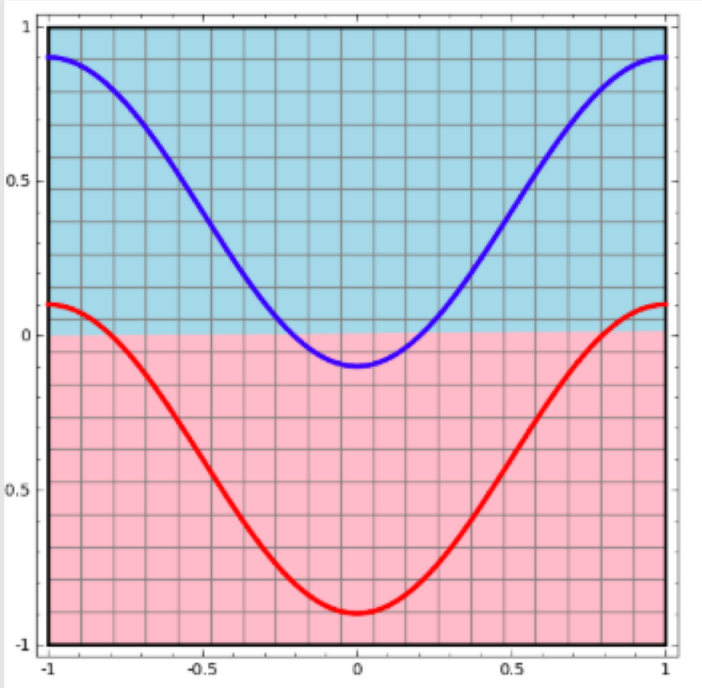
Effects of depth of layers



Let's look at an example.

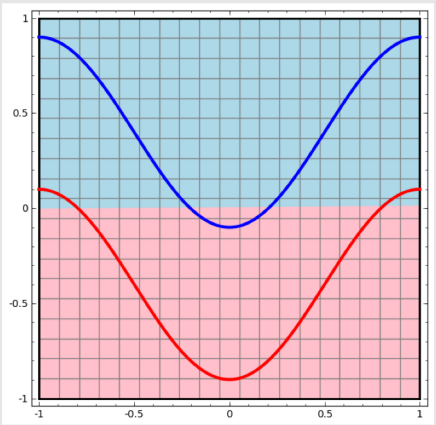
- The left figure shows an simple dataset: two curves on a plane. The neural network will learn to classify points as belonging to one or the other.

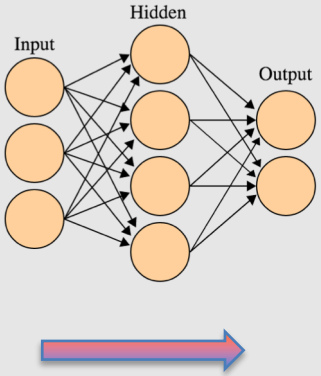
Effects of depth of layers

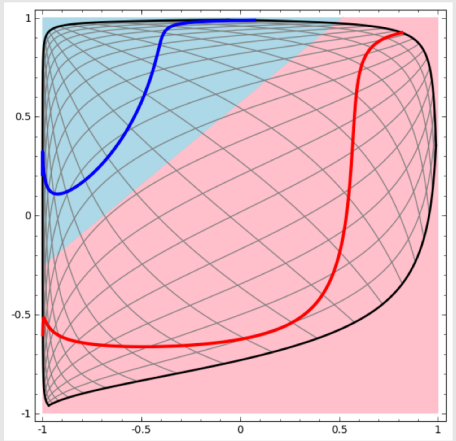


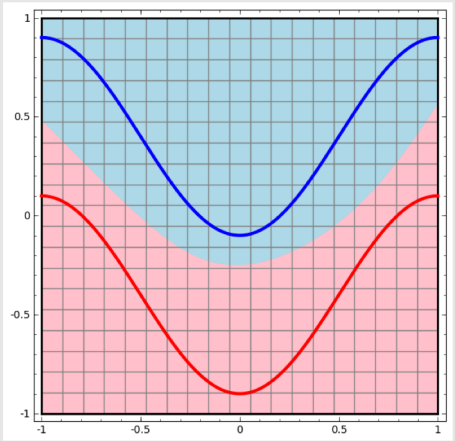
- We start from a neural network with only an input layer and an output layer (i.e., a perceptron).
- It can only divide the two classes of data with a line.

Effects of depth of layers (cont.)



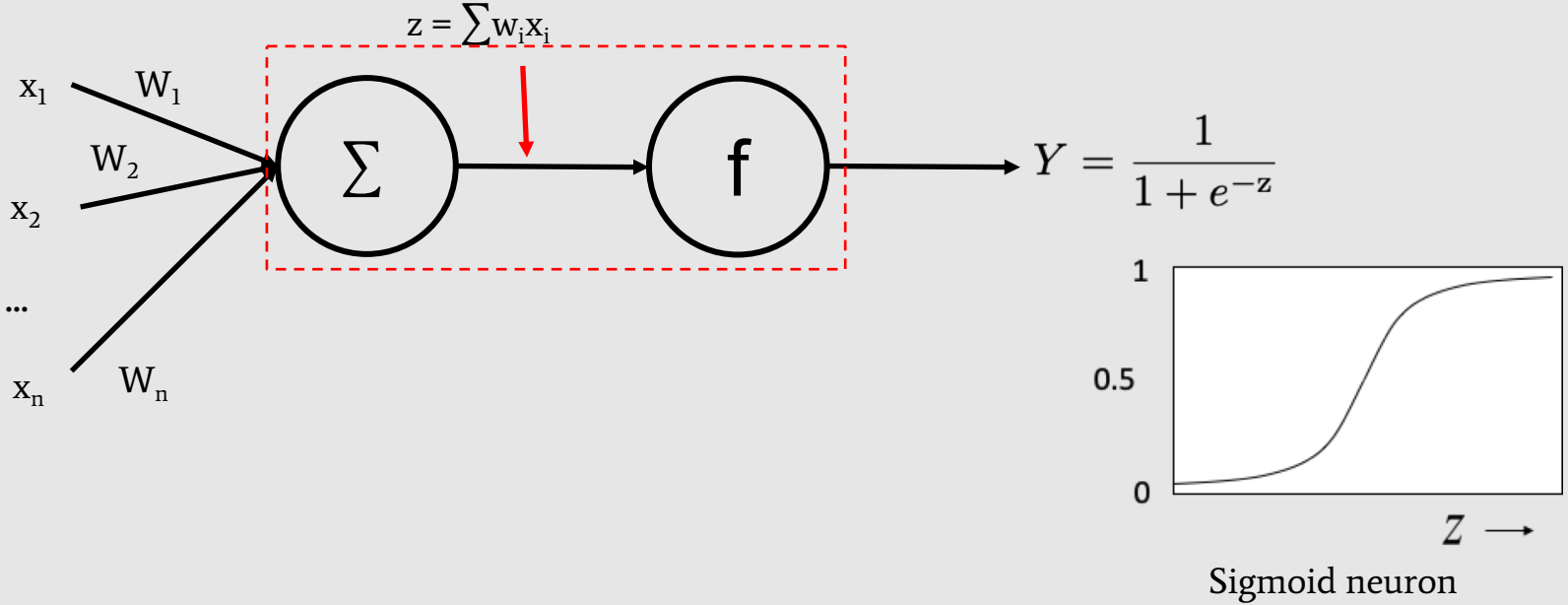






Adding non-linearity

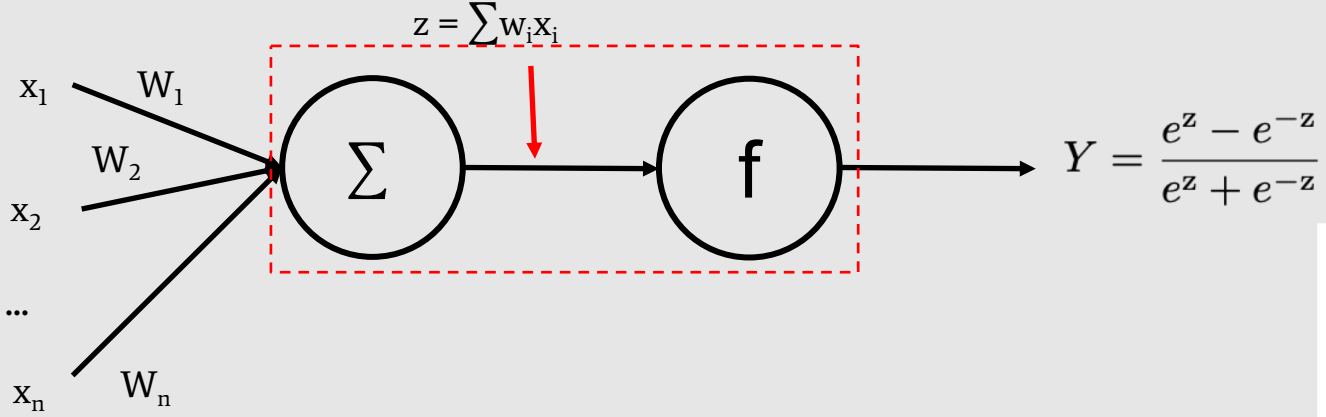
Any f is fine as long as it is differentiable (allow error to pass back to use delta-rule)

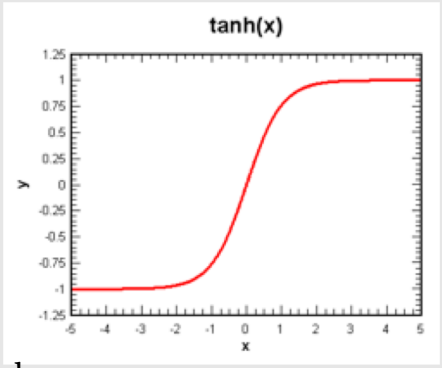


Sigmoid neuron

Adding non-linearity

Any f is fine as long as it is differentiable (allow error to pass back to use delta-rule)

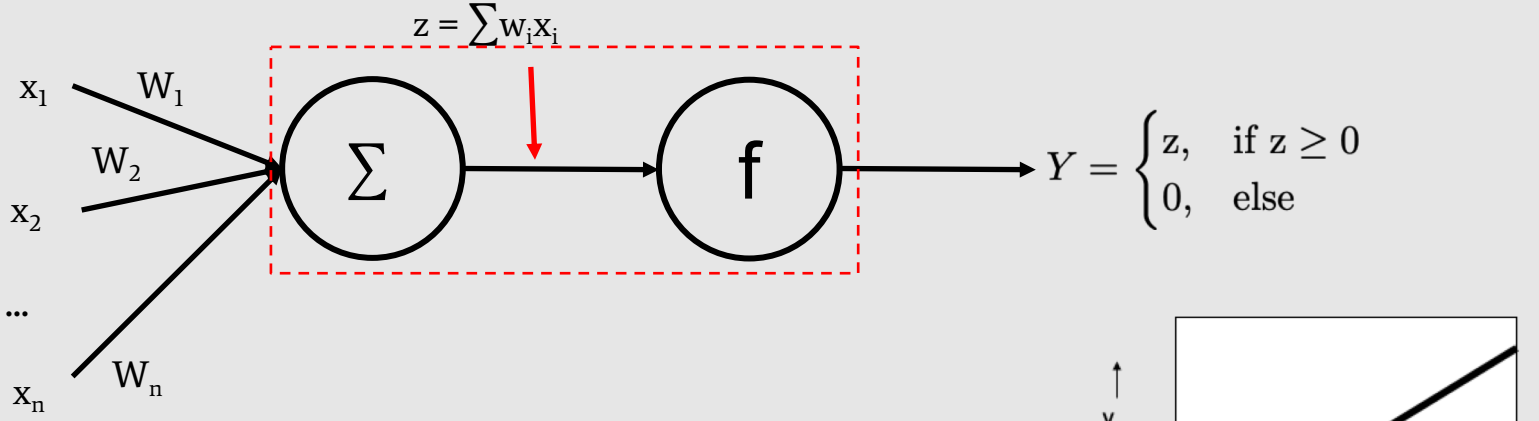




tanh neuron

Adding non-linearity

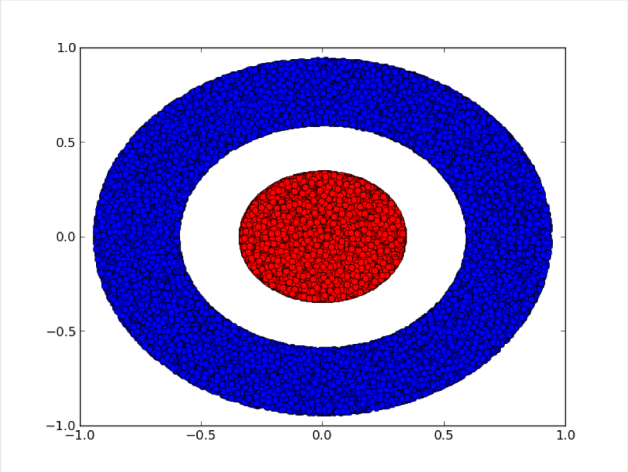
Any f is fine as long as it is differentiable (allow error to pass back to use delta-rule)

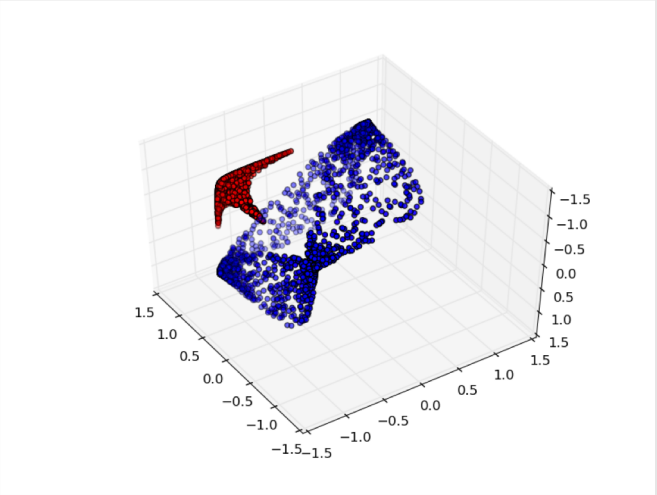


Rectified linear neuron

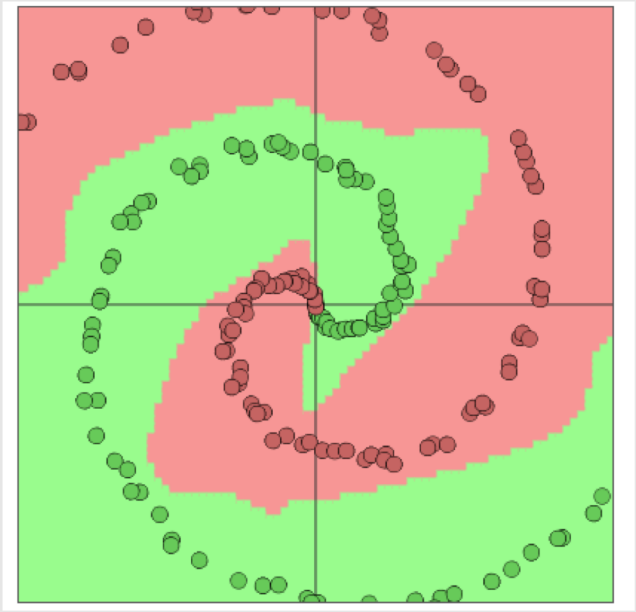
Adding non-linearity

- In addition to stretch, rotate and shift
- “cuts, breaks, folds...”





Multilayer non-linear classifier in action

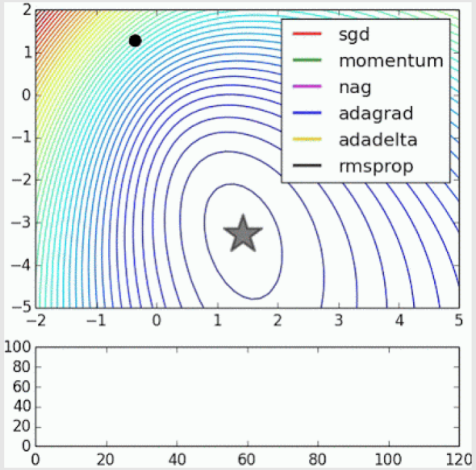


http://cs.stanford.edu/people/karpathy/convnetjs//demo/classify2d.html

Training the neuron network

The loss function: $J(\Theta) = \frac{1}{M}\sum_{i=1}^{M} \ell(y^{(i)}, f(\mathbf{x}^{(i)}))$, where $\Theta = \{\mathbf{w}, b\}$.

- To minimize J(Θ), we use gradient descent.
- We start from some randomly chosen Θ, then update it by $\Theta = \Theta - \lambda \frac{\partial J}{\partial \Theta}$
- This guarantees the loss function will reduce.



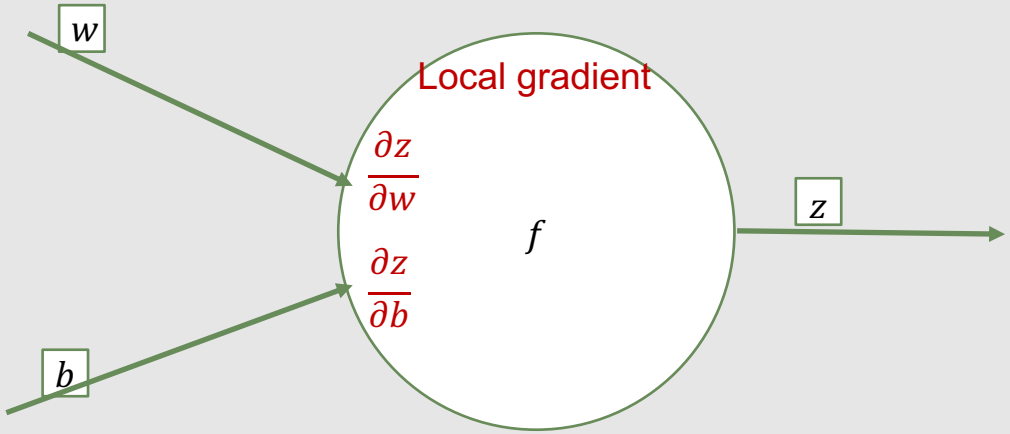
http://www.denizyuret.com/2015/03/alec-radfords-animations-for.html

Parameter Optimization –back propagation

Parameter optimization: $\Theta -= \lambda \frac{\partial L}{\partial \Theta}$

Loss function: L

? $\frac{\partial L}{\partial w}$ $\frac{\partial L}{\partial b}$

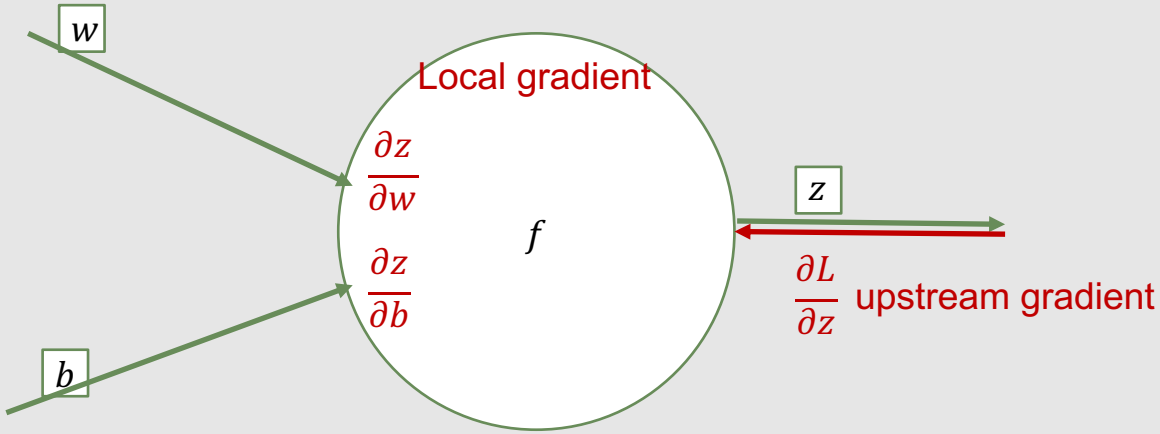


Parameter Optimization –back propagation

Parameter optimization: $\Theta -= \lambda \frac{\partial L}{\partial \Theta}$

Loss function: L

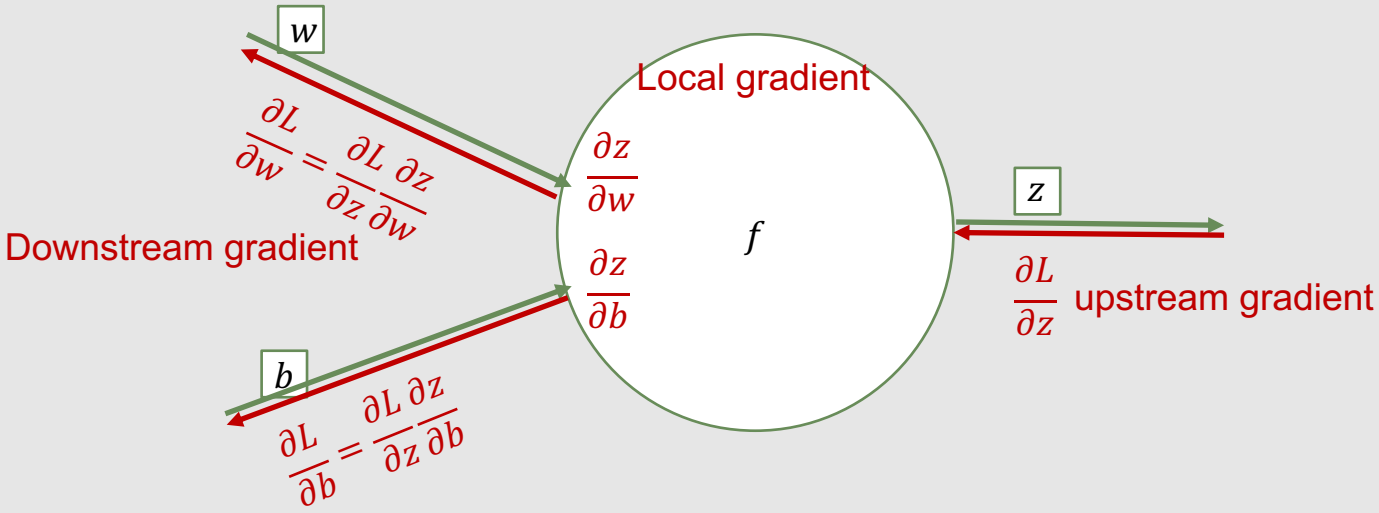
? $\frac{\partial L}{\partial w}$ $\frac{\partial L}{\partial b}$



Parameter Optimization –back propagation

Back-propagation with chain rule

Update with: $\Theta \mathrel{-}= \lambda \frac{\partial L}{\partial \Theta}$



Agenda

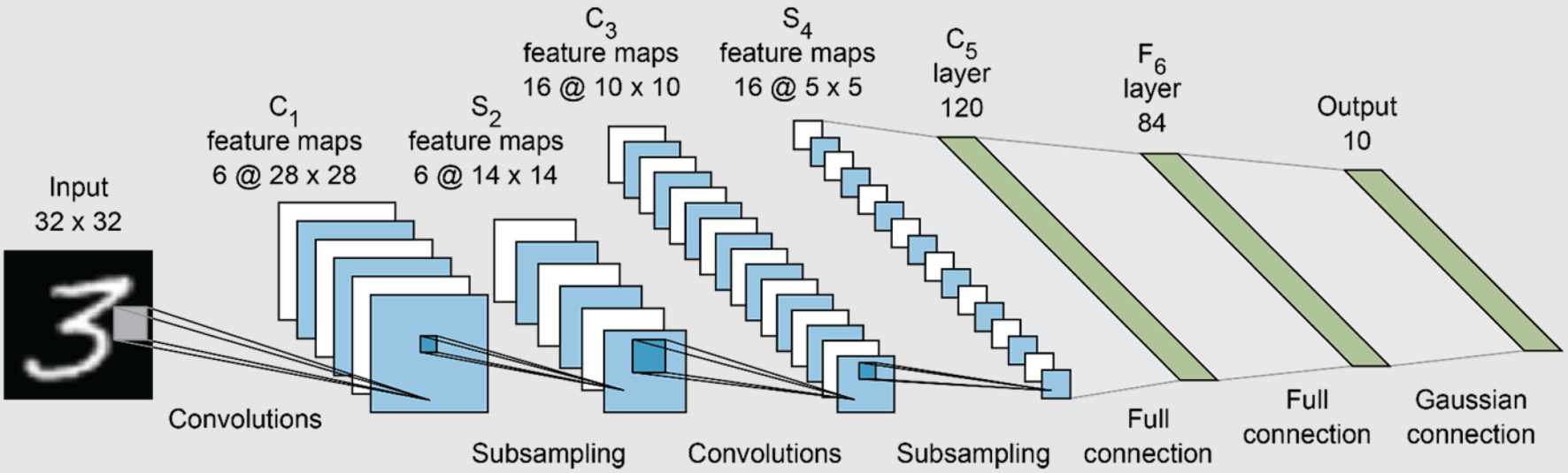
- Deep Learning in Action & History
- Neural Network
- Deep Learning: CNN and RNN

Convolutional neural network (CNN)

CNNs are a specialized kind of neural network for processing data that has a known grid-like topology.

- e.g., Images (i.e., 2-D grid) ; data that measured at regular time intervals
- Convolution: a mathematical operation employed by the network

Read of the US checks!



- LeNet5
- Invented by Prof Yann LeCun from NYU-Courant
- [Demo]

A simple convolution

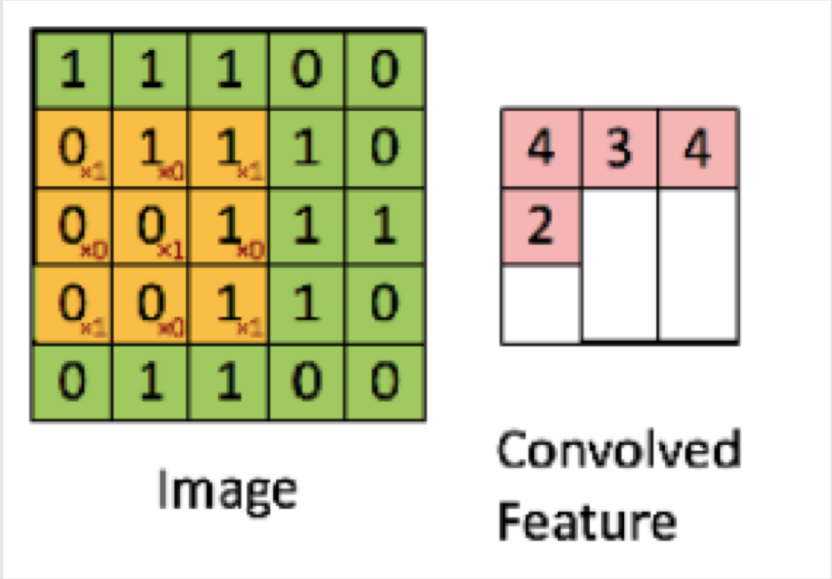
Original Image

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

filter/kernel /feature detector

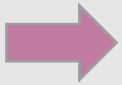
Feature map



CNN is a sparsely connected ANN

- A 2-D array can be reshaped into an 1-D vector.

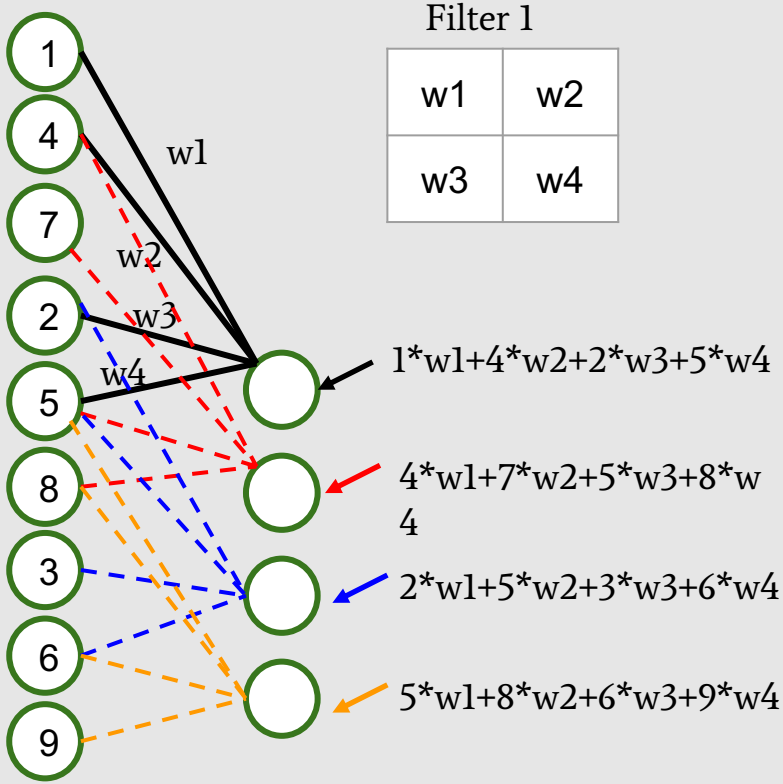
1	4	7
2	5	8
3	6	9



1
4
7
2
5
8
3
6
9

So, the convolution is equal to a sparsely connected network

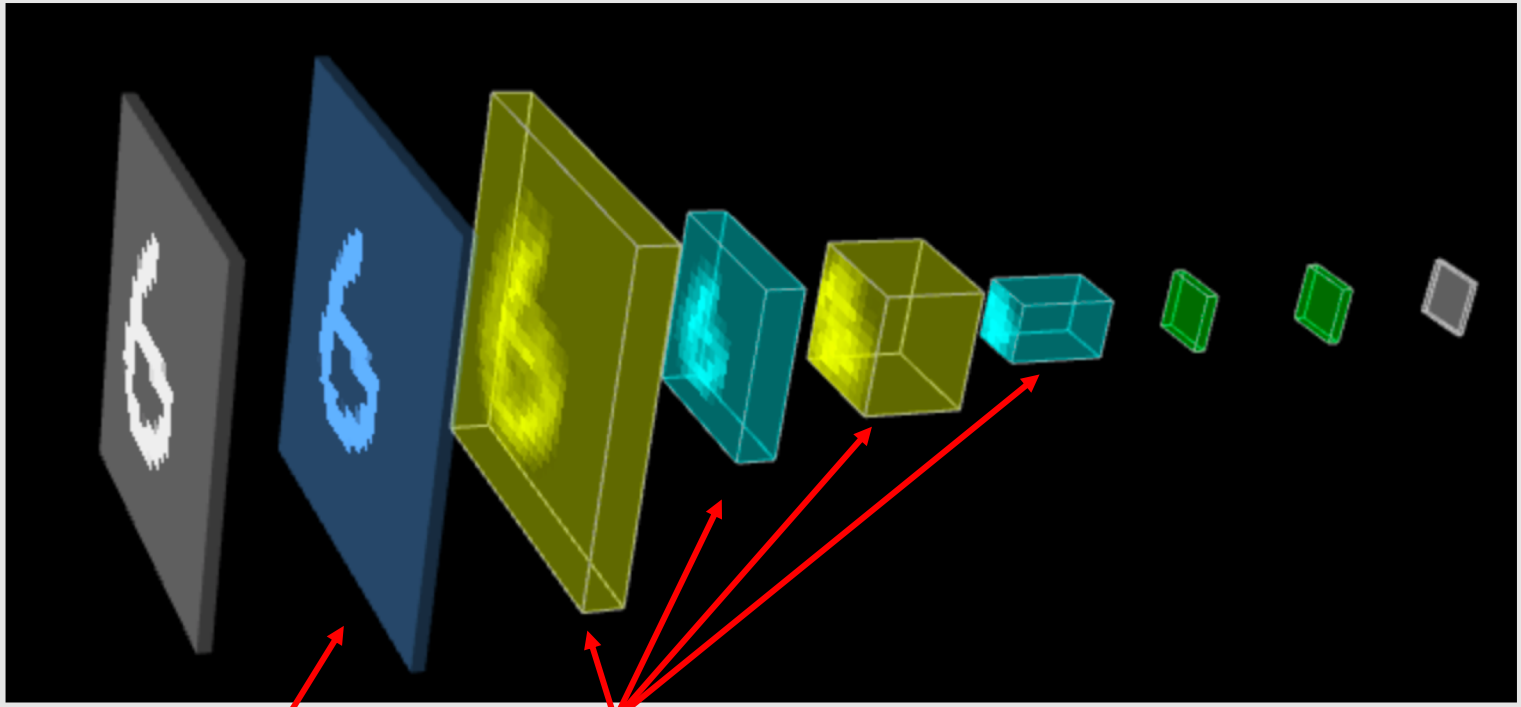
If we convolve the array with filter 1, the corresponding network is shown in the right.



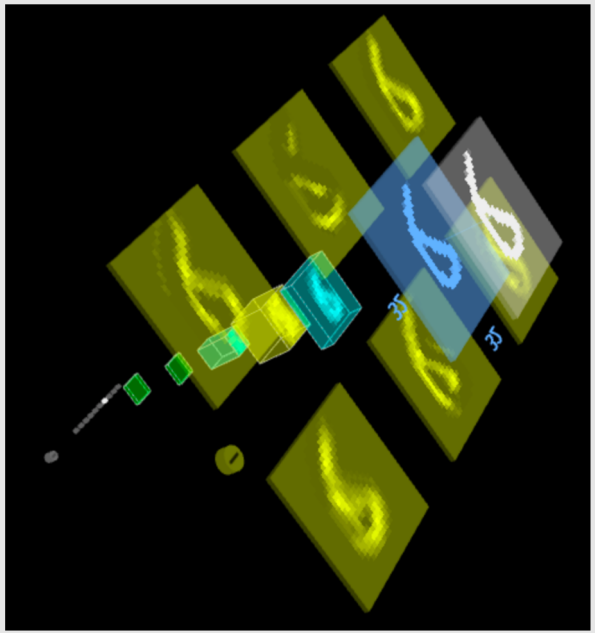
Filter 1

w1	w2
w3	w4

An Online LeNet



An Online LeNet

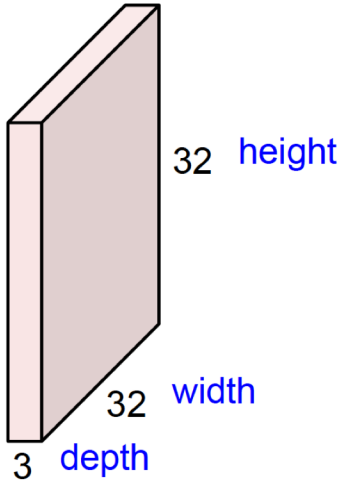


We can visualize the feature maps layer by layer.

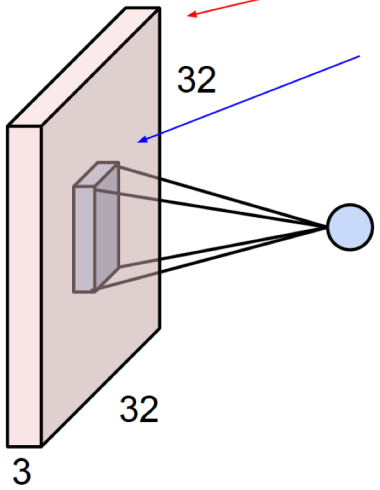
- The 1st layer feature maps retain most of the information present in the image.
  - In CNN, the first layers usually act as edge detectors.
- You may notice that, as we go deeper, the feature maps look less like the original image. This is because that deeper feature maps encode high level features, like “6 is a circle connecting with a curve”

Convolution Layer

32x32x3 image



Convolution Layer



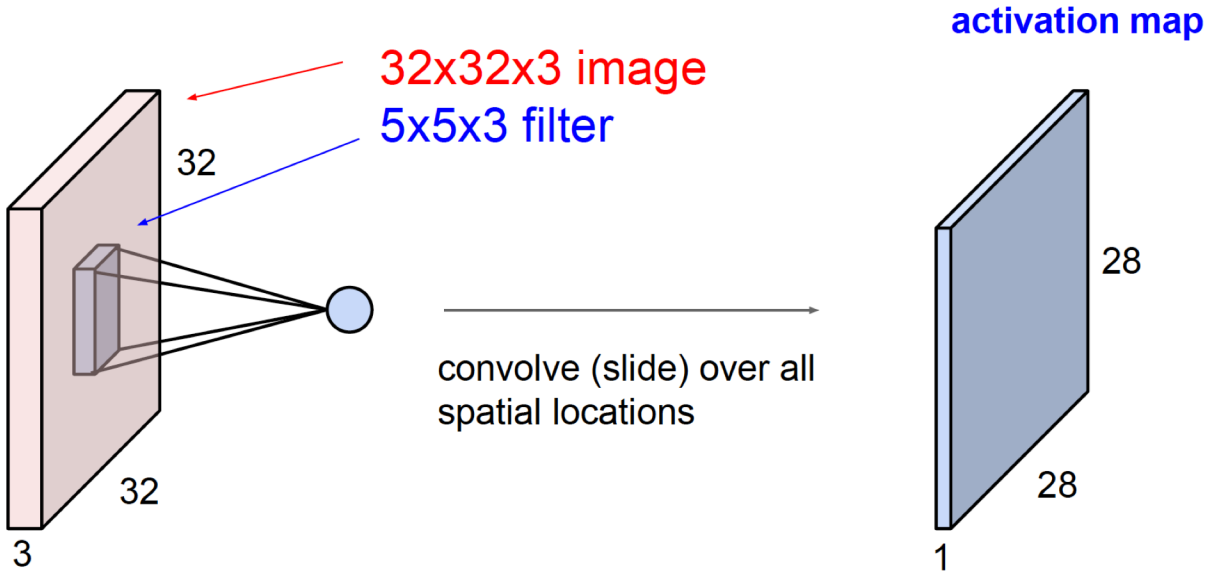
32x32x3 image

5x5x3 filter w

1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

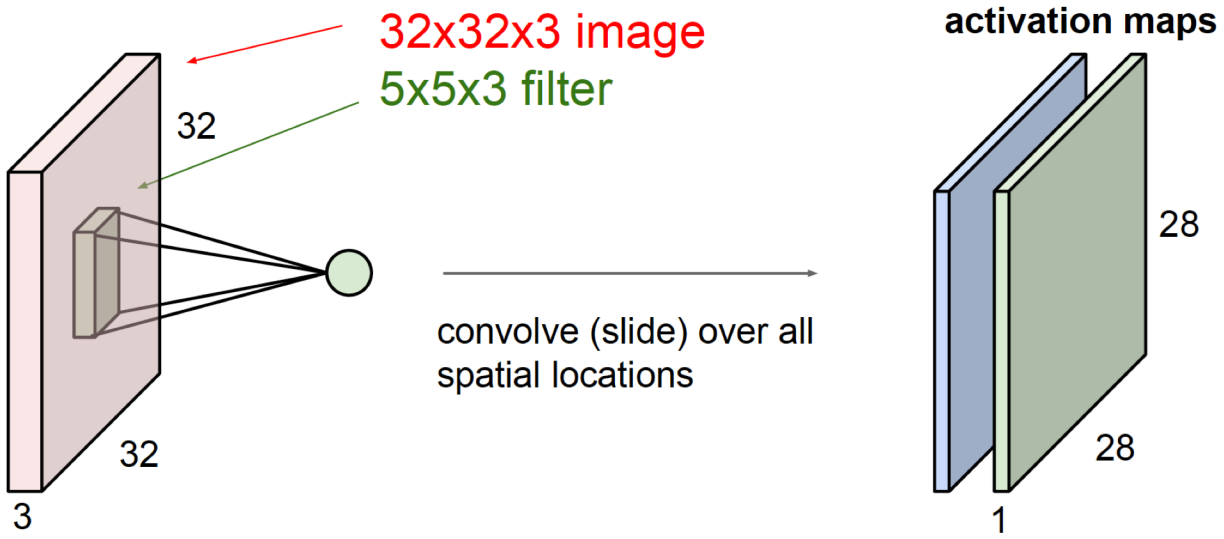
$$w^T x + b$$

Convolution Layer

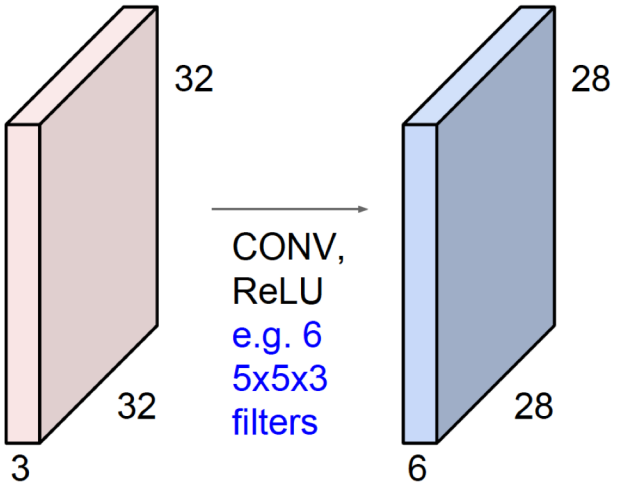


Convolution Layer

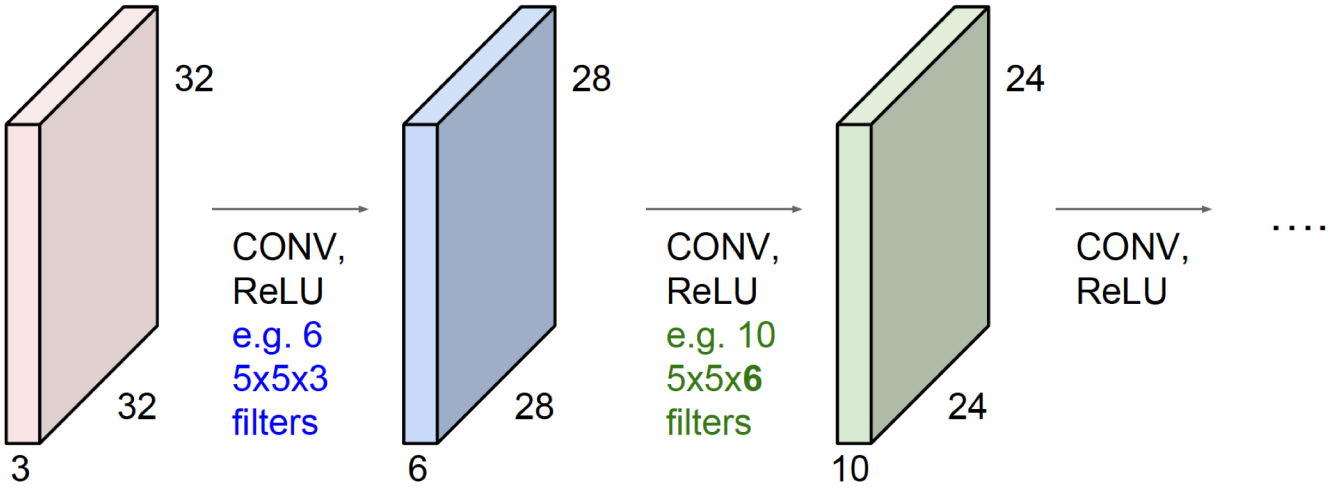
consider a second, green filter



Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions

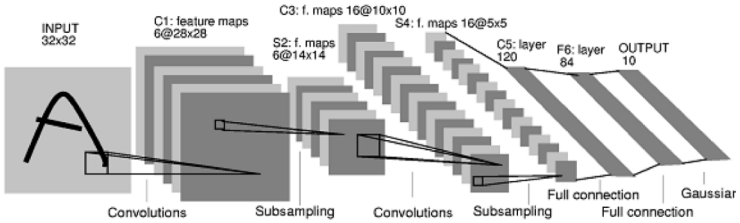


Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



1998

LeCun et al.



# of transistors

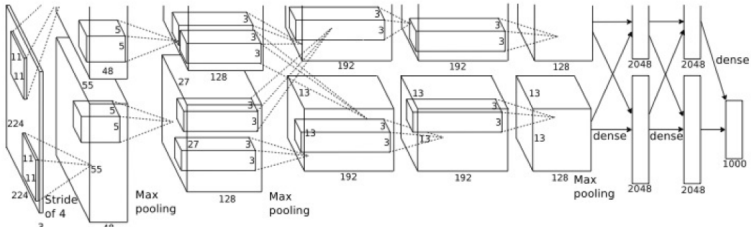
10^6

# of pixels used in training

10^7 NIST

2012

Krizhevsky et al.



# of transistors

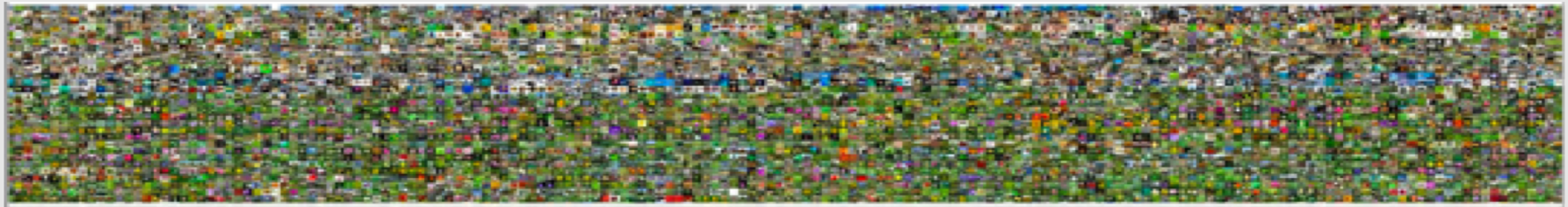
10^9

GPUs

# of pixels used in training

10^{14} IMAGENET

ImageNet project is a visual database designed for use in visual object recognition software research.



IMAGENET

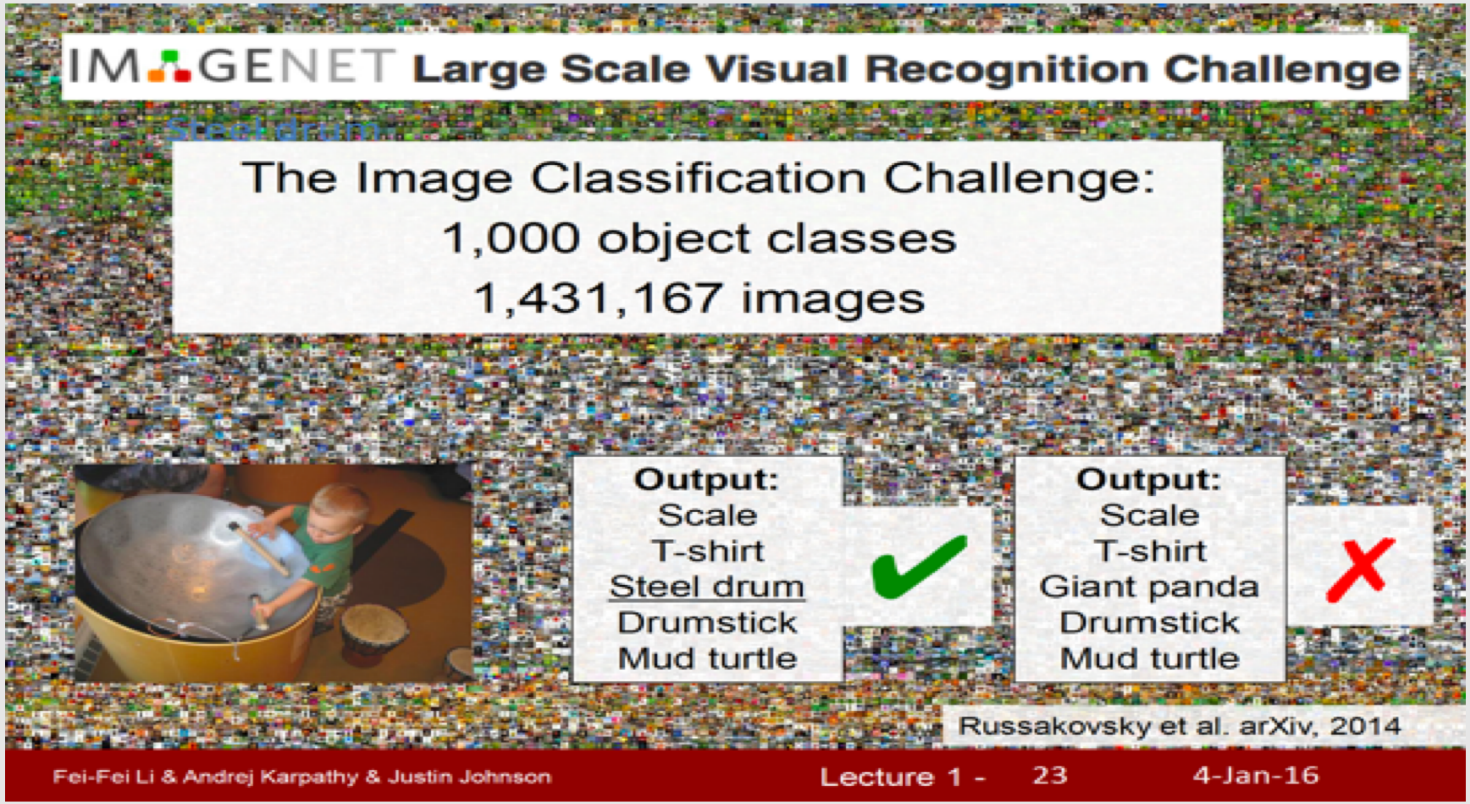
www.image-net.org

22K categories and **14M** images

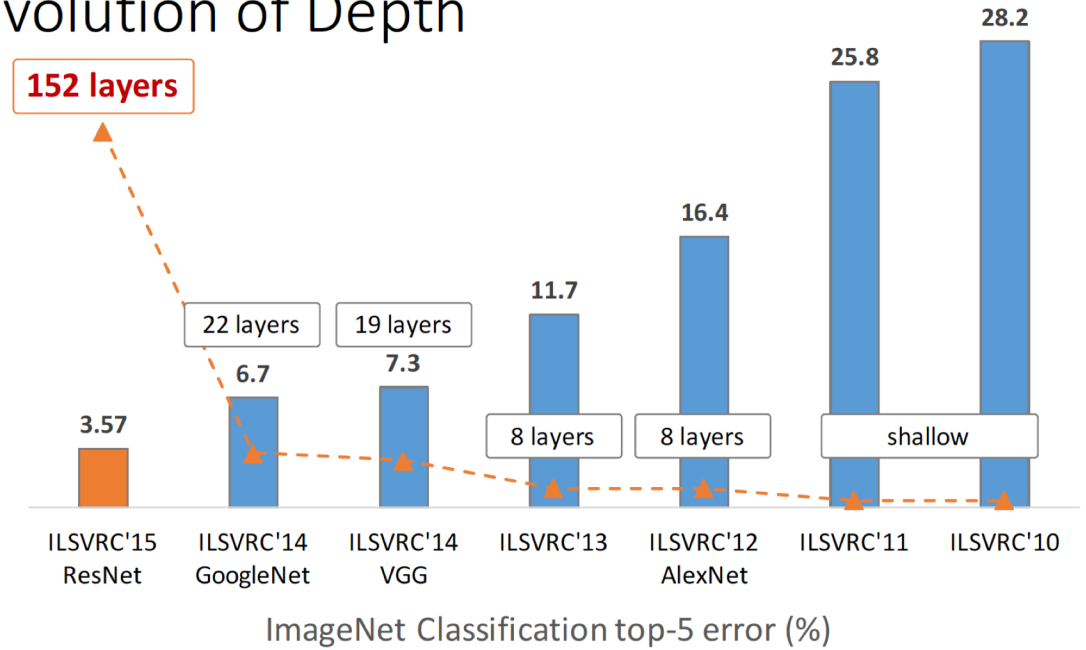
- Animals
  - Bird
  - Fish
  - Mammal
  - Invertebrate
- Plants
  - Tree
  - Flower
- Food
- Materials
- Structures
- Artifact
  - Tools
  - Appliances
  - Structures
- Person
- Scenes
  - Indoor
  - Geological Formations
- Sport Activities

Deng, Dong, Socher, Li, Li, & Fei-Fei, 2009

ImageNet: Large Scale Visual Recognition Challenge



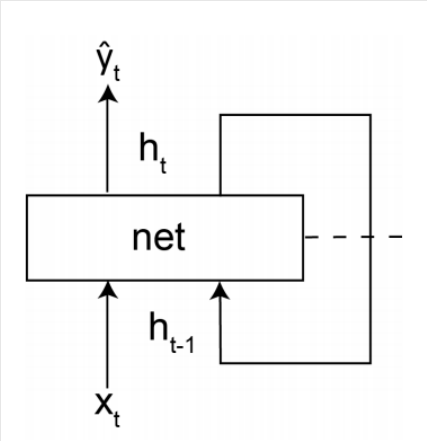
Revolution of Depth

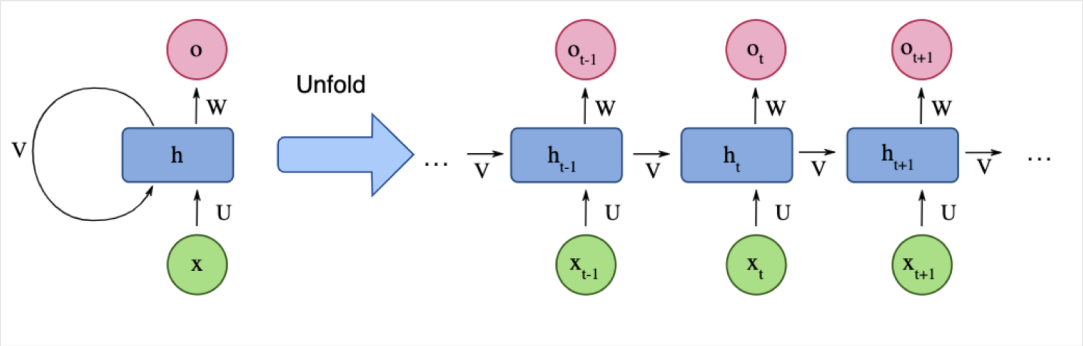


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

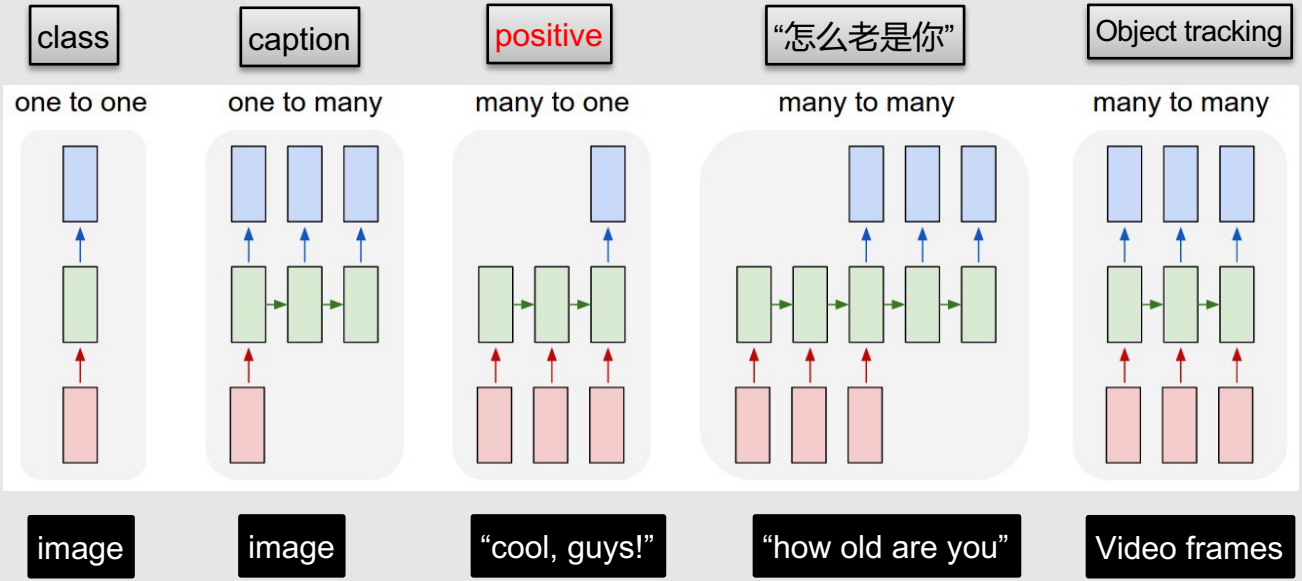
Recurrent neural networks (RNNs)

$$h_t = \tanh(Wx_t + Uh_{t-1} + b)$$
$$\hat{y}_t = g(h_t)$$





Making it recurrent



“The unreasonable effectiveness of recurrent networks”

Making it recurrent

```
VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.
```

After “reading” Shakespeare for a few hours

“The unreasonable effectiveness of recurrent networks”

Making it recurrent

```
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
```

After “reading” Linux kernels for a few hours

“The unreasonable effectiveness of recurrent networks”

Show and Tell

Figure 5. A selection of evaluation results, grouped by human rating.