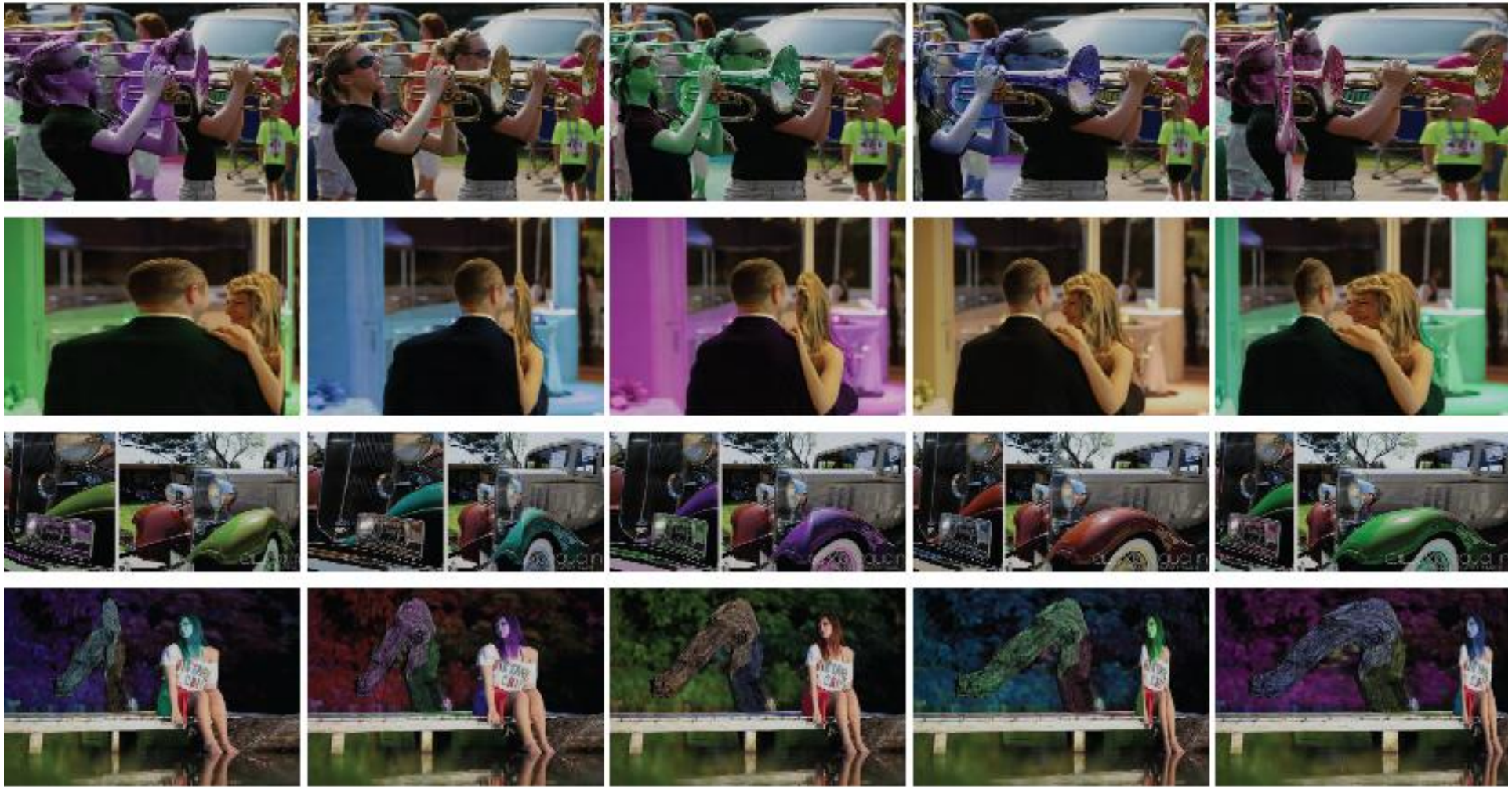
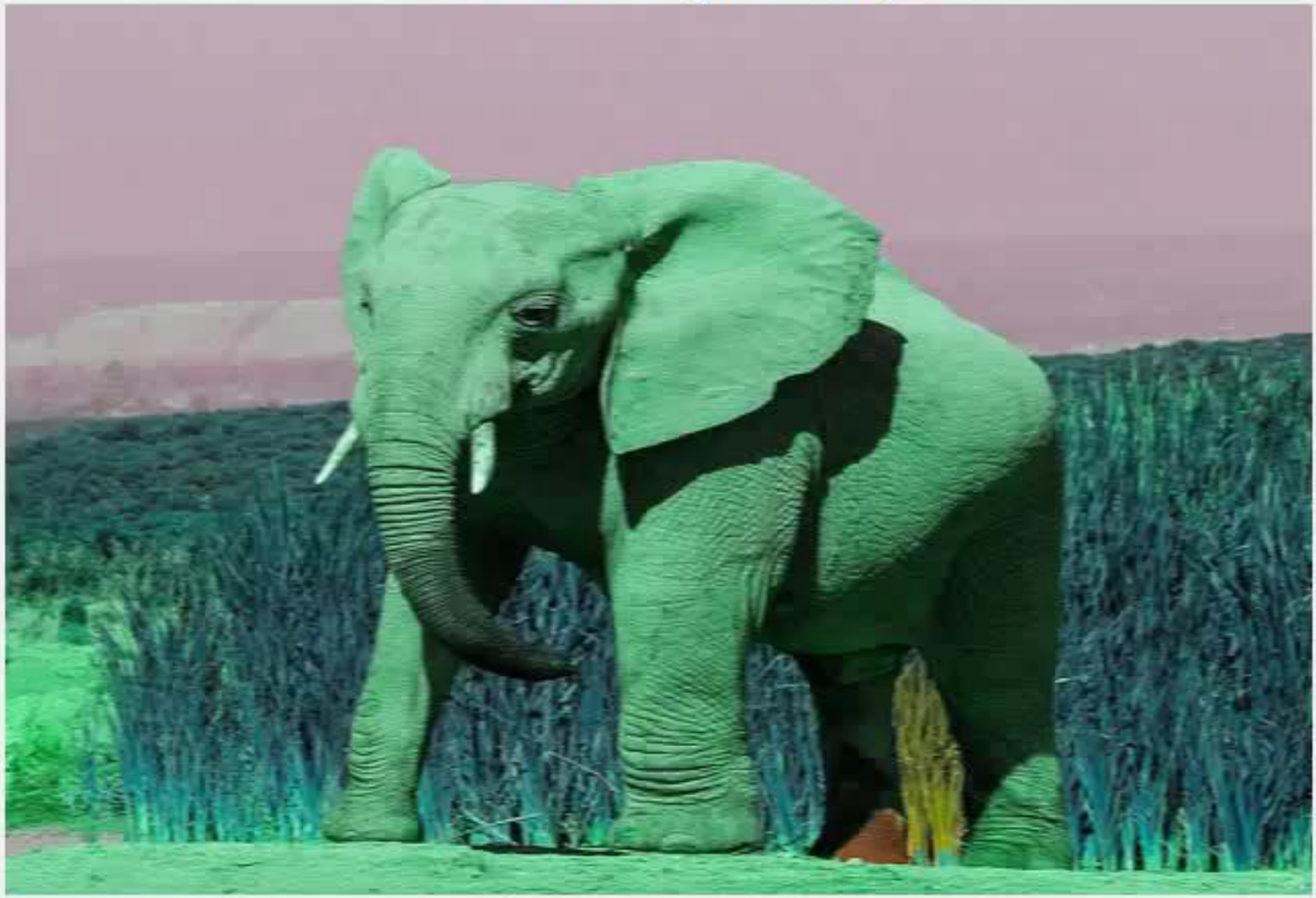
Seam Carving!



Slide to the Original Image



‹ ›

Submit

Labradoodle or fried chicken

Puppy or bagel

Sheepdog or mop

Chihuahua or muffin

Barn owl or apple

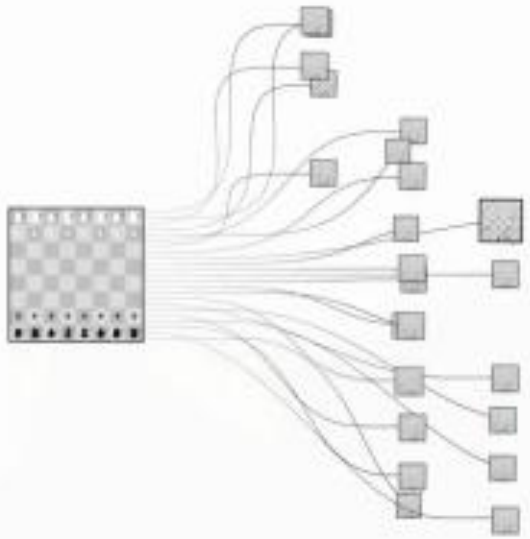
Parrot or guacamole

Raw chicken or Donald Trump

But, we human actually lose!

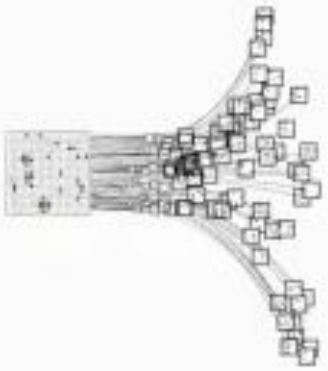
- A demo that shows **we, human, lose**, on the classification task, we are proud of, we have been **trained** for millions of years!
- If we want to make it hard for bots, it has to be hard for human as well.

How would you crack it?



Chess: 10^{47}

Deep Blue, Feb 10, 1996



Go: 10^{170}

AlphaGo, March, 2016

We (will) lose on many specific tasks!

- Speech recognition
- Translation
- Self-driving
- …

- BUT, they are not AI yet…
- Don’t worry until it dates with your girl/boy friend…

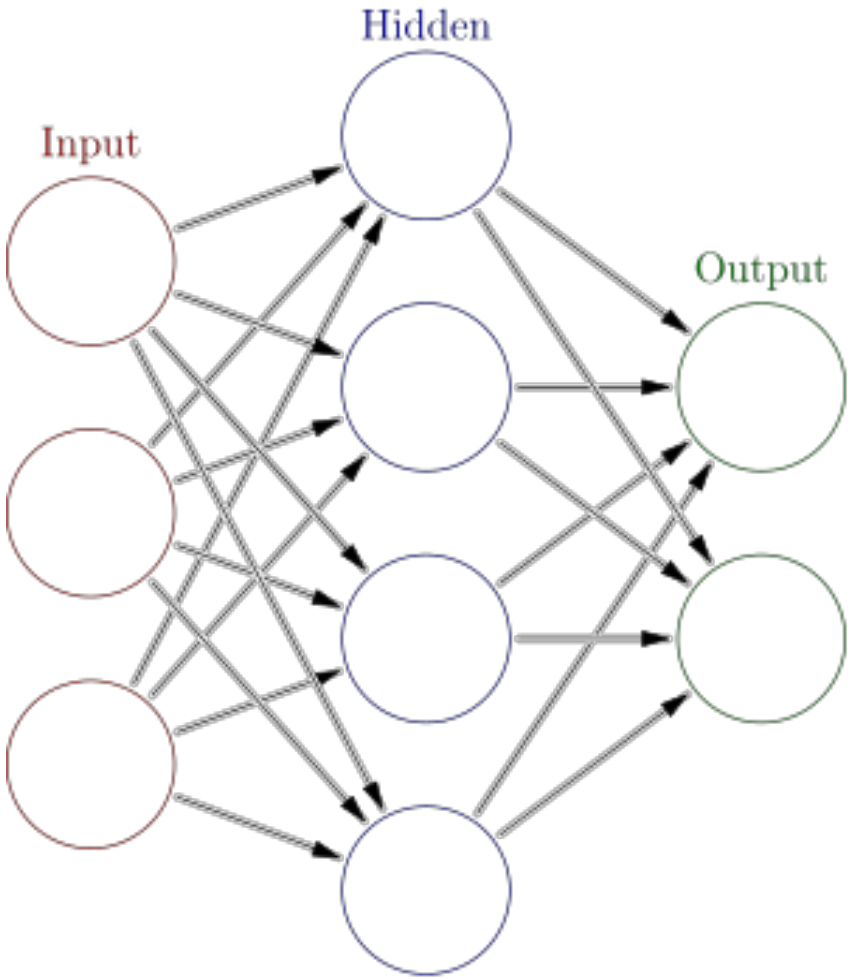
Deep learning is so cool for so many problems…


OLD CROW
MODERN CROW
Update Yourself - It saves a lot of extra effort

A Brief Introduction to Deep Learning

- Artificial Neural Network
- Back-propagation
- Fully Connected Layer
- Convolutional Layer
- Overfitting

Artificial Neural Network



1. Activation function
2. Weights
3. Cost function
4. Learning algorithm

Live Demo

Neurons are functions

- Let's start with a complex one!

$$f(x, y) = x + y$$

- Given $x = a, y = b$, how to update x *and* y to make $f(x, y)$ larger?
- Follow gradient directions!

$$f(x, y) = x + y \qquad \rightarrow \qquad \frac{\partial f}{\partial x} = 1 \qquad \frac{\partial f}{\partial y} = 1$$

$$x = a + 0.01 * 1,$$
$$y = b + 0.01 * 1$$
$$f(x, y): a + b \rightarrow \mathrm{a} + \mathrm{b} + 0.02$$

Neurons are functions

- A more complex one!

$$f(x, y) = x * y$$

- Given $x = a, y = b$, how to update x *and* y to make $f(x, y)$ larger?
- Follow gradient directions!

$$f(x, y) = xy \qquad \rightarrow \qquad \frac{\partial f}{\partial x} = y \qquad \frac{\partial f}{\partial y} = x$$

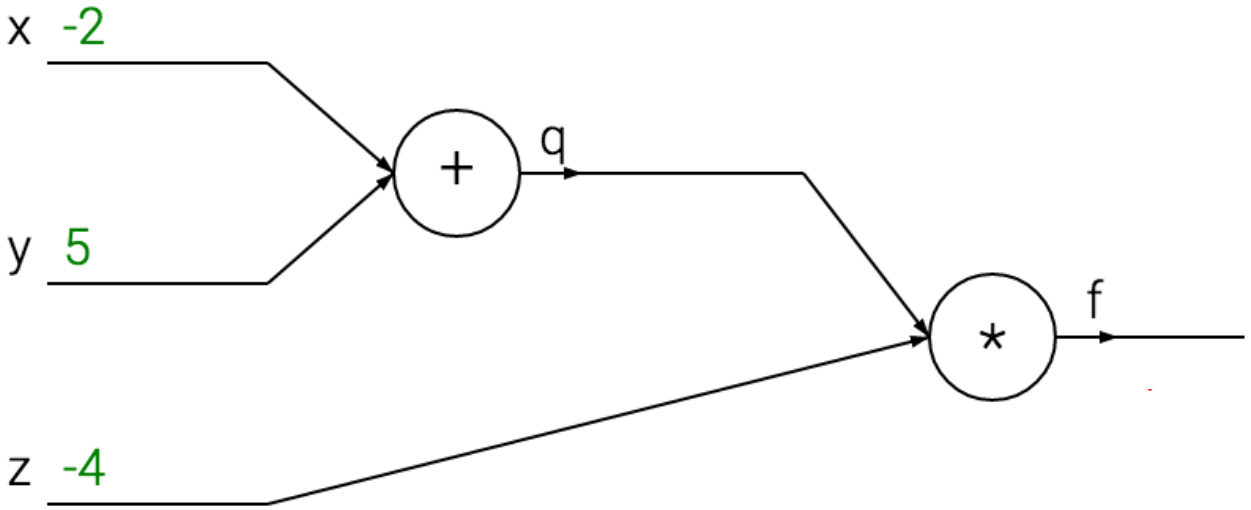
$$x = a + 0.01 * b,$$
$$y = b + 0.01 * a$$
$$f(x, y): a * b \rightarrow (\mathrm{a} + 0.01 * \mathrm{b})(\mathrm{b} + 0.01 * \mathrm{a})$$
$$f(x, y): 4 * (-3) \rightarrow 3.97 * (-2.96)$$

Back-propagation

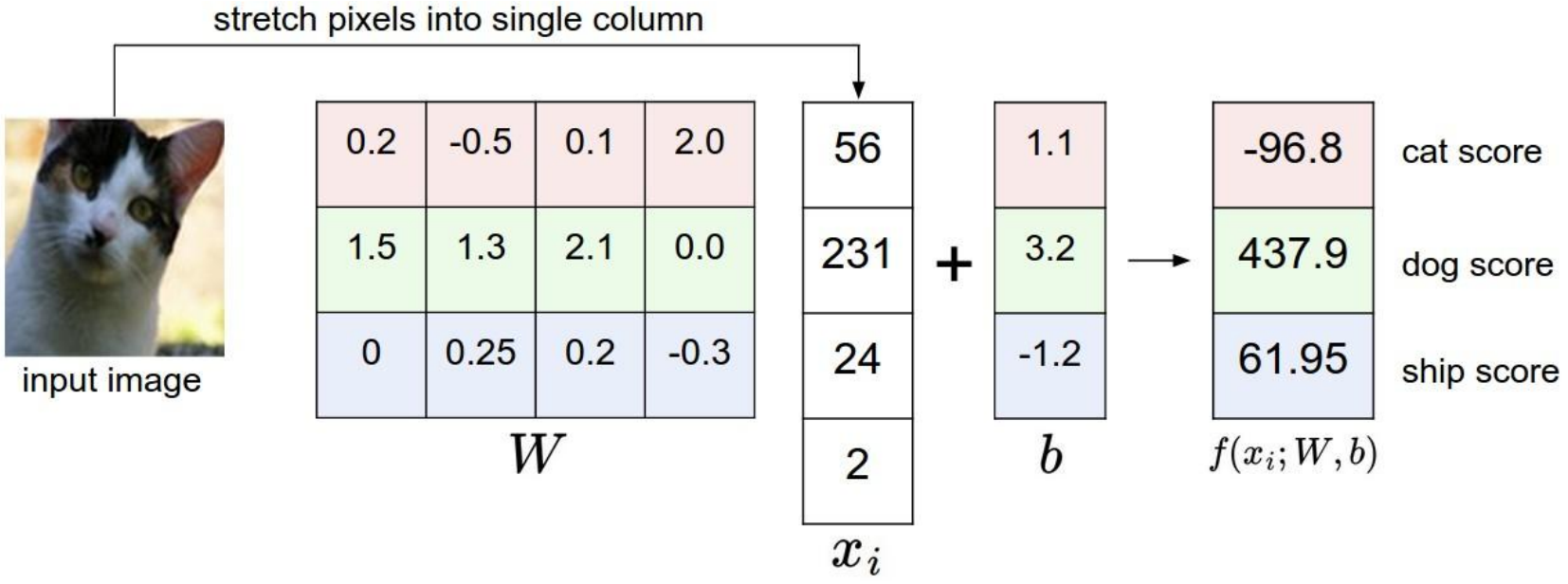
- An extremely complex one!

$$f(x, y, z) = (x + y) * z$$

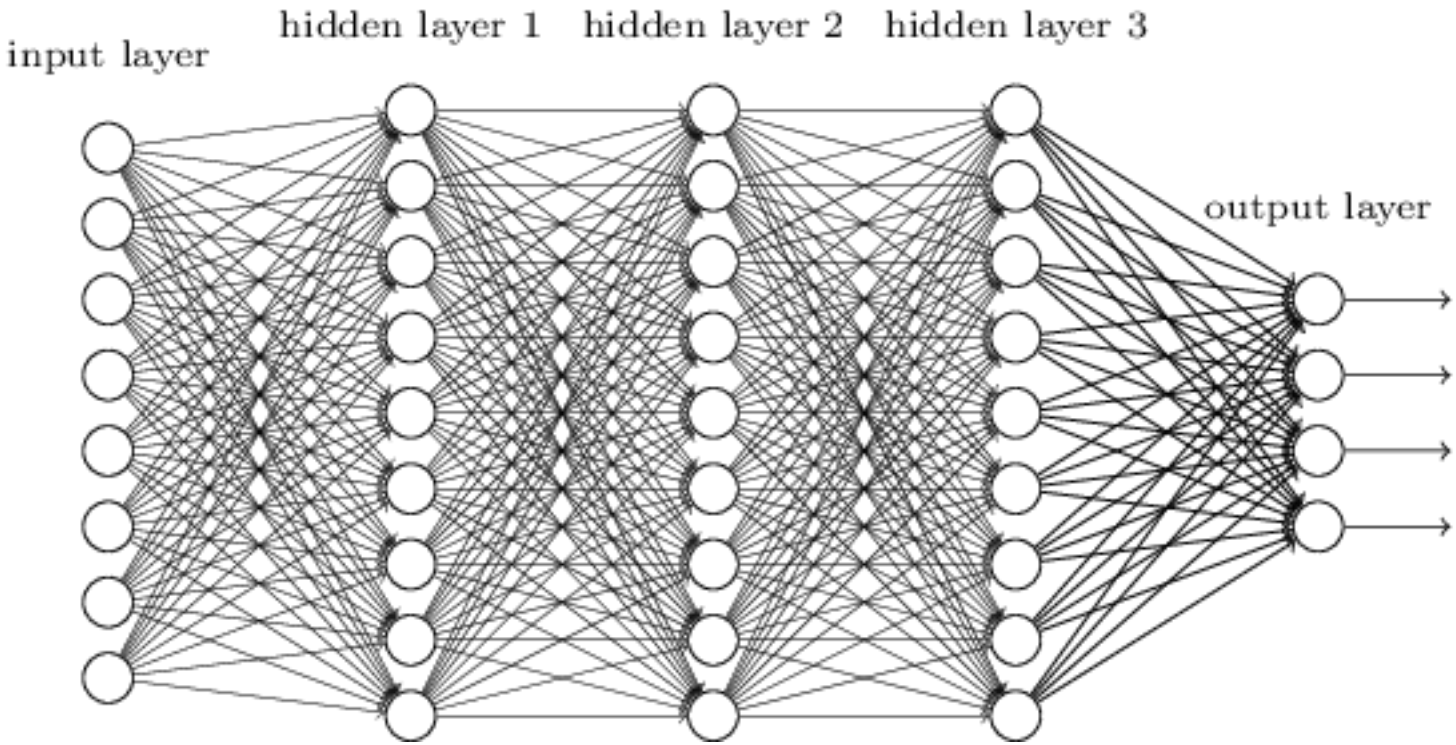
- Let $q(x, y) = (x + y)$, then $f(x, y, z) = q(x, y) * z$
- Chain rule: $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$



Now, serious stuff, a bit…



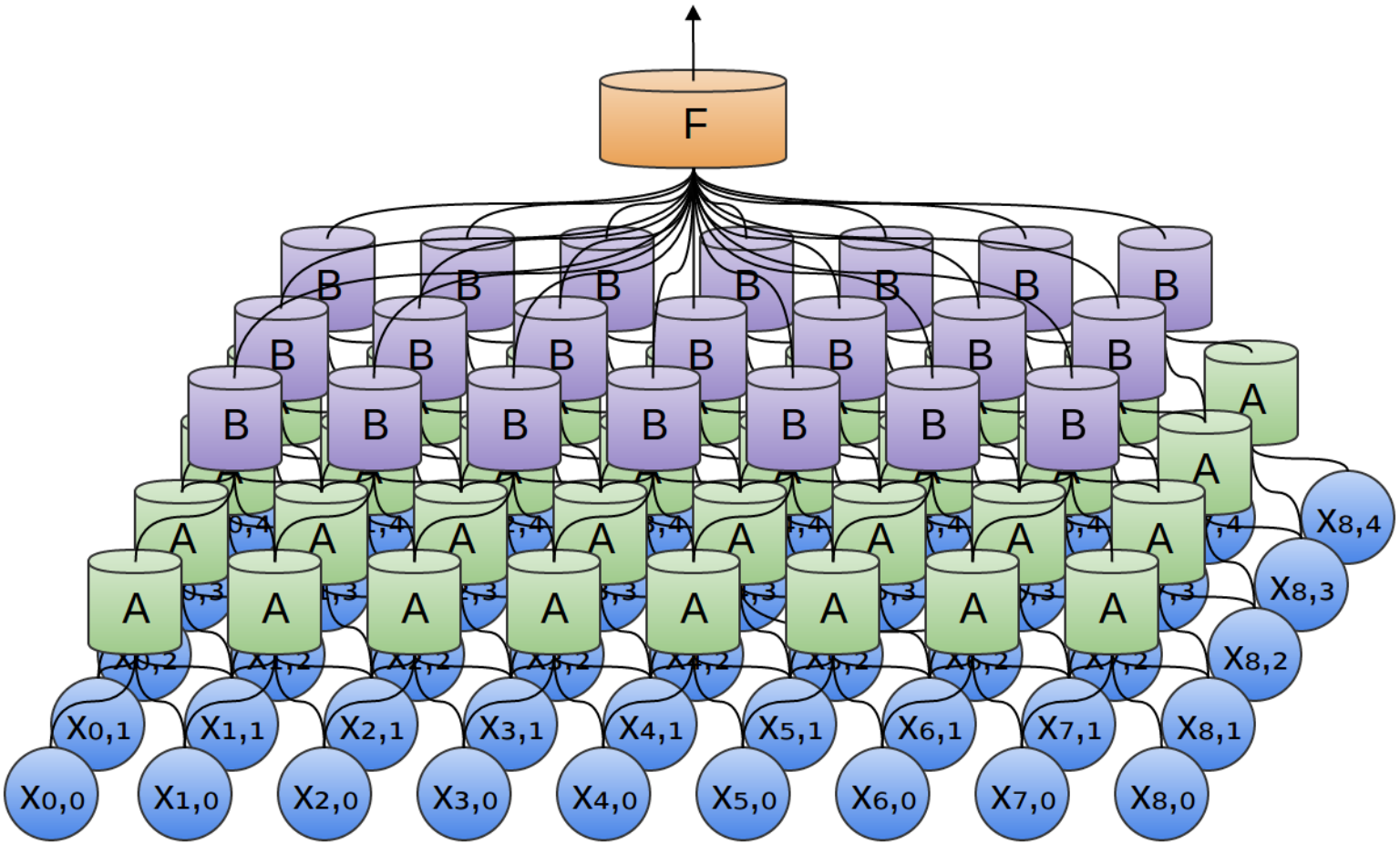
Fully Connected Layers



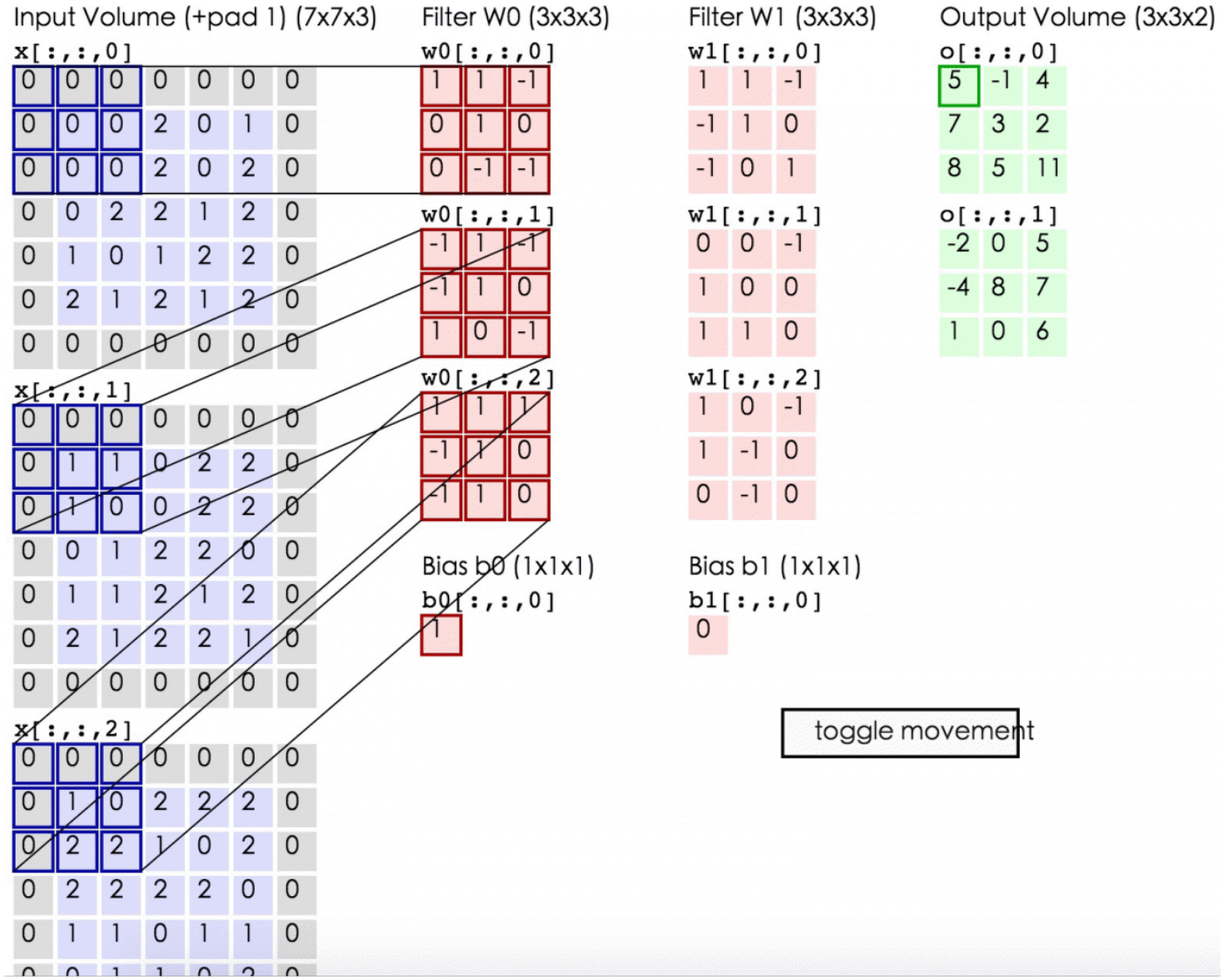
“When in doubt, use brute force.”
--Ken Thompson

“If brute force is possible...”
--Yangyan Li

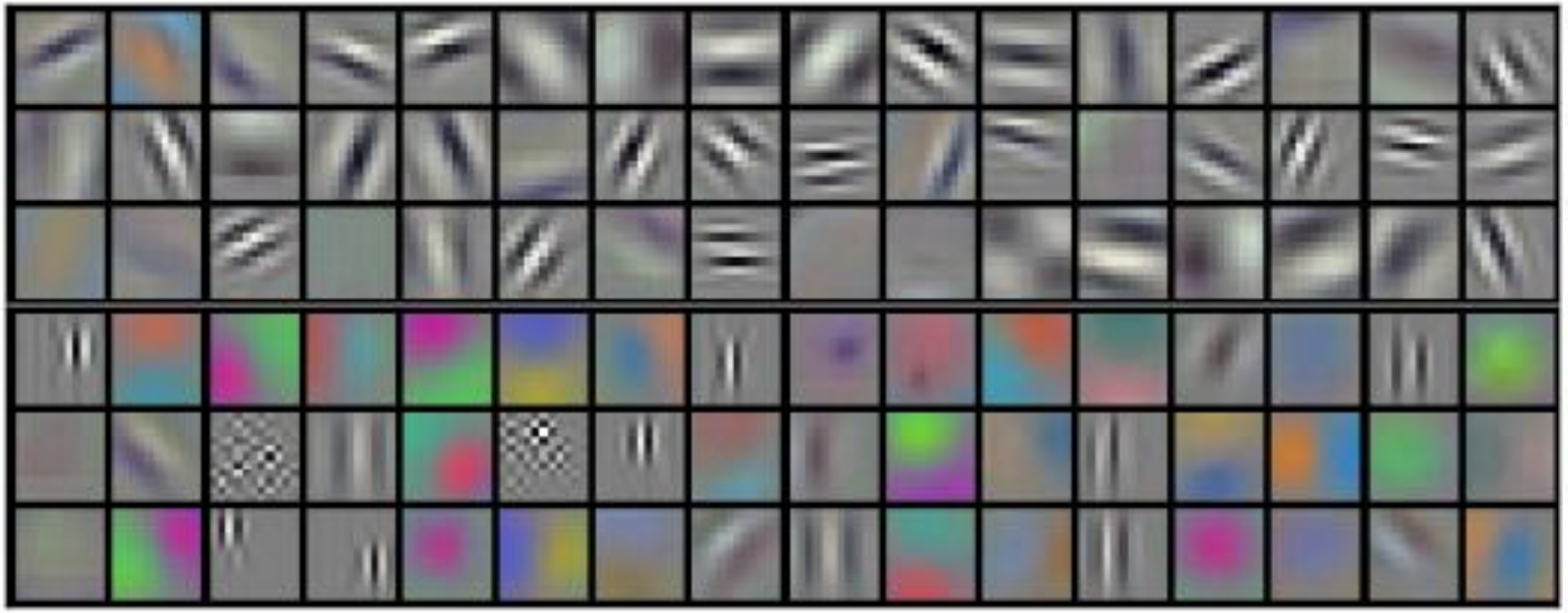
Convolutional Layers



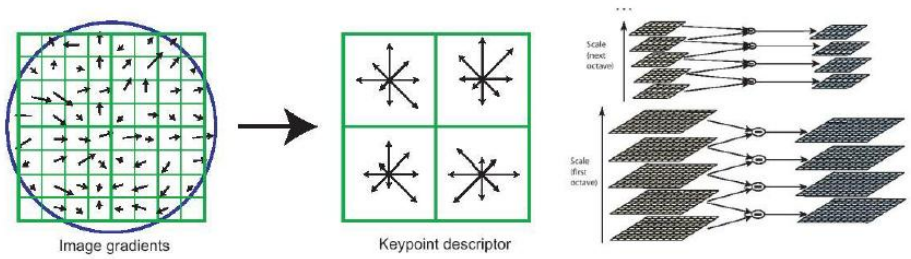
Convolutional Layers



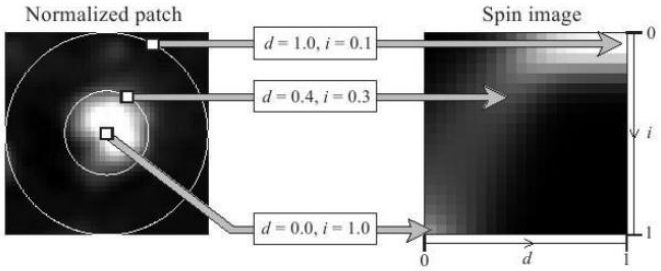
Convolution Filters



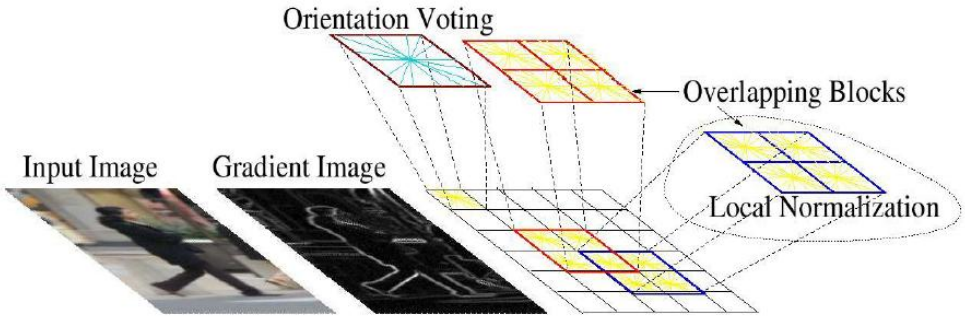
Computer vision features



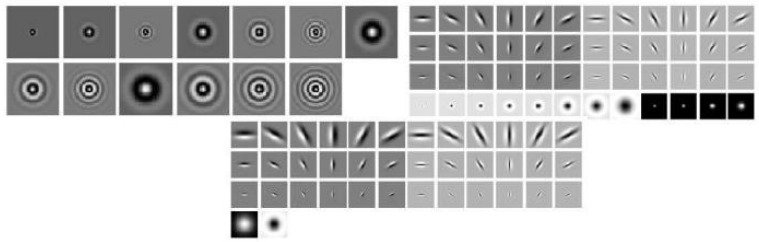
SIFT



Spin image



HoG

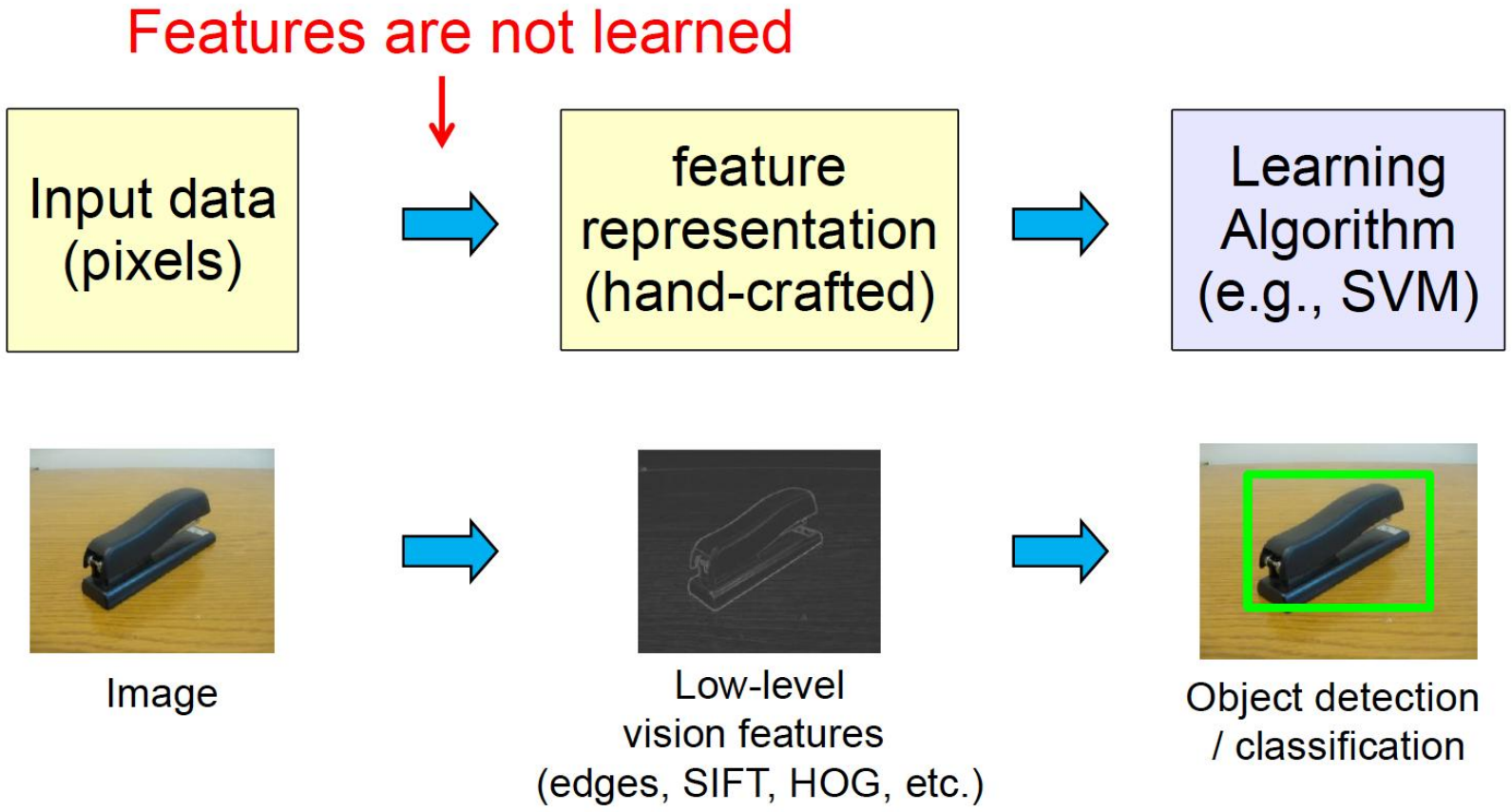


Textons

and many others:

SURF, MSER, LBP, Color-SIFT, Color histogram, GLOH, …..

Traditional Recognition Approach

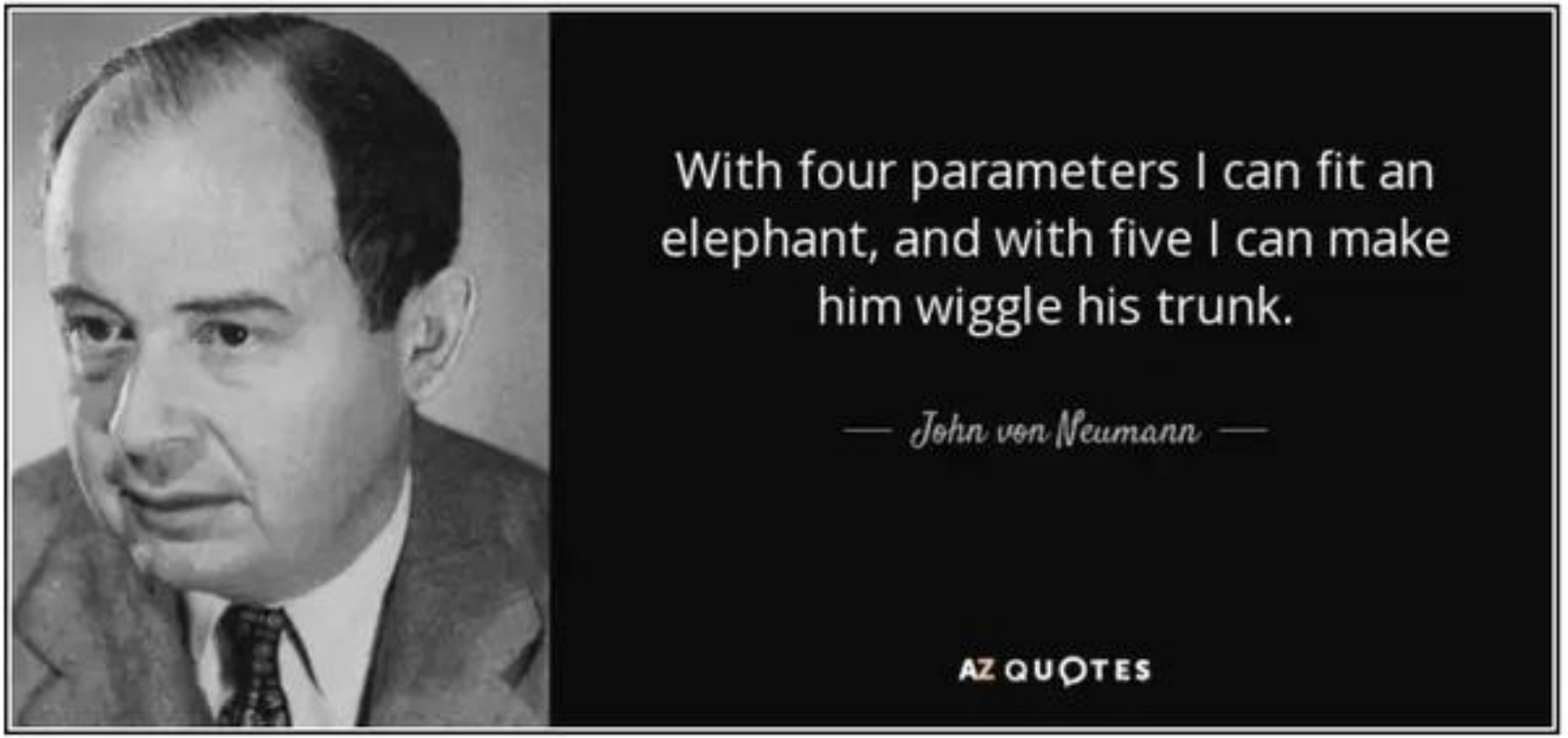


Feature Engineering vs. Learning

- Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work.
- “When working on a machine learning problem, feature engineering is manually designing what the input x's should be.”

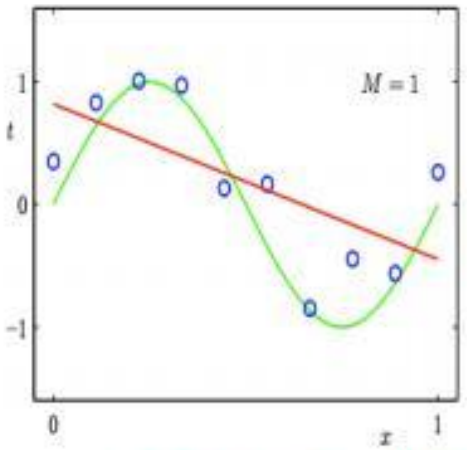
  -- Shayne Miel
- “Coming up with features is difficult, time-consuming, requires expert knowledge.”

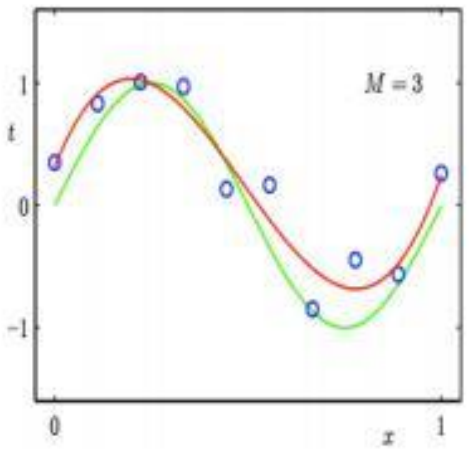
  --Andrew Ng

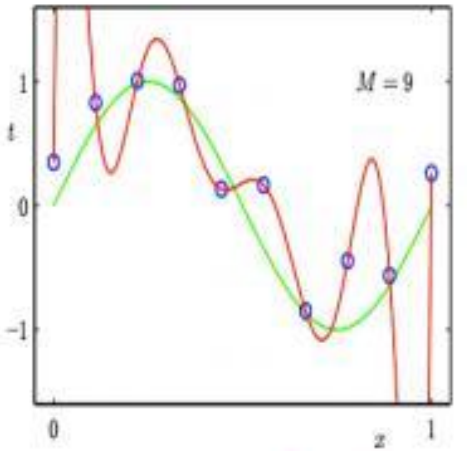

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.
— John von Neumann —
AZ QUOTES

Under- and Over-fitting examples

Regression:



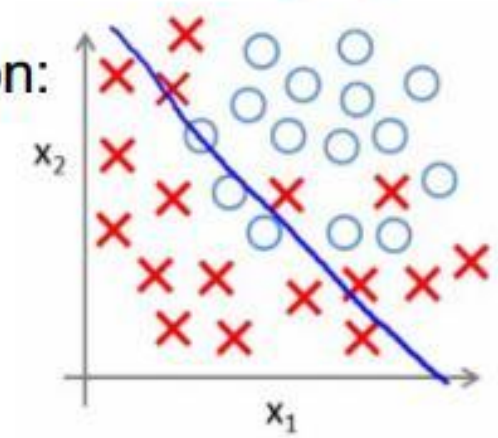


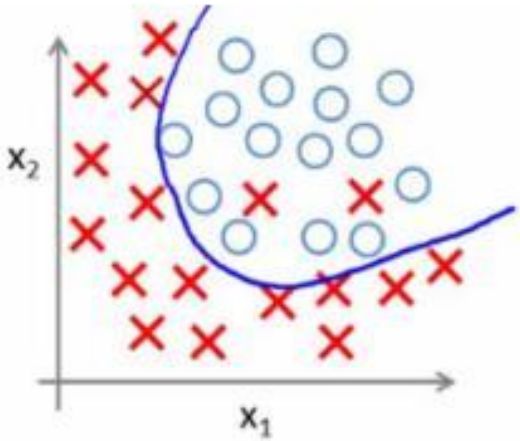


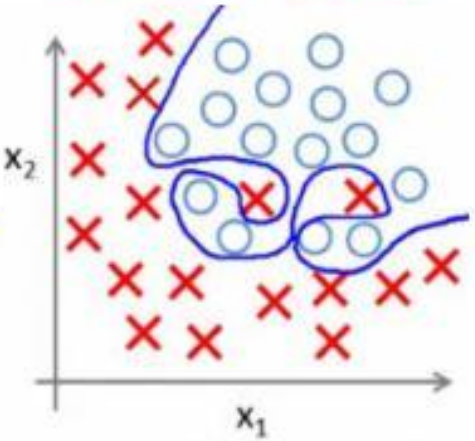
predictor too inflexible:
cannot capture pattern

predictor too flexible:
fits noise in the data

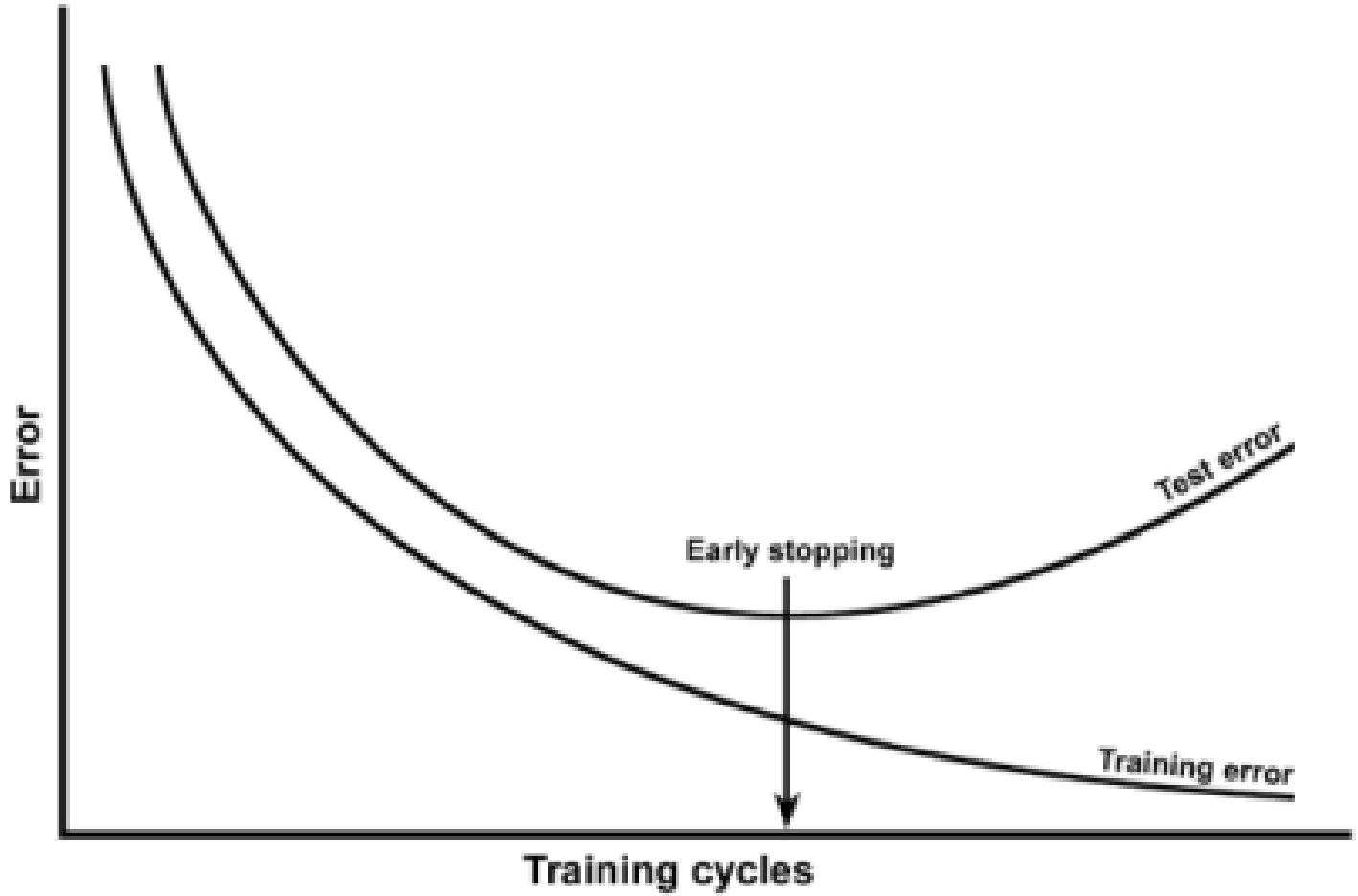
Classification:



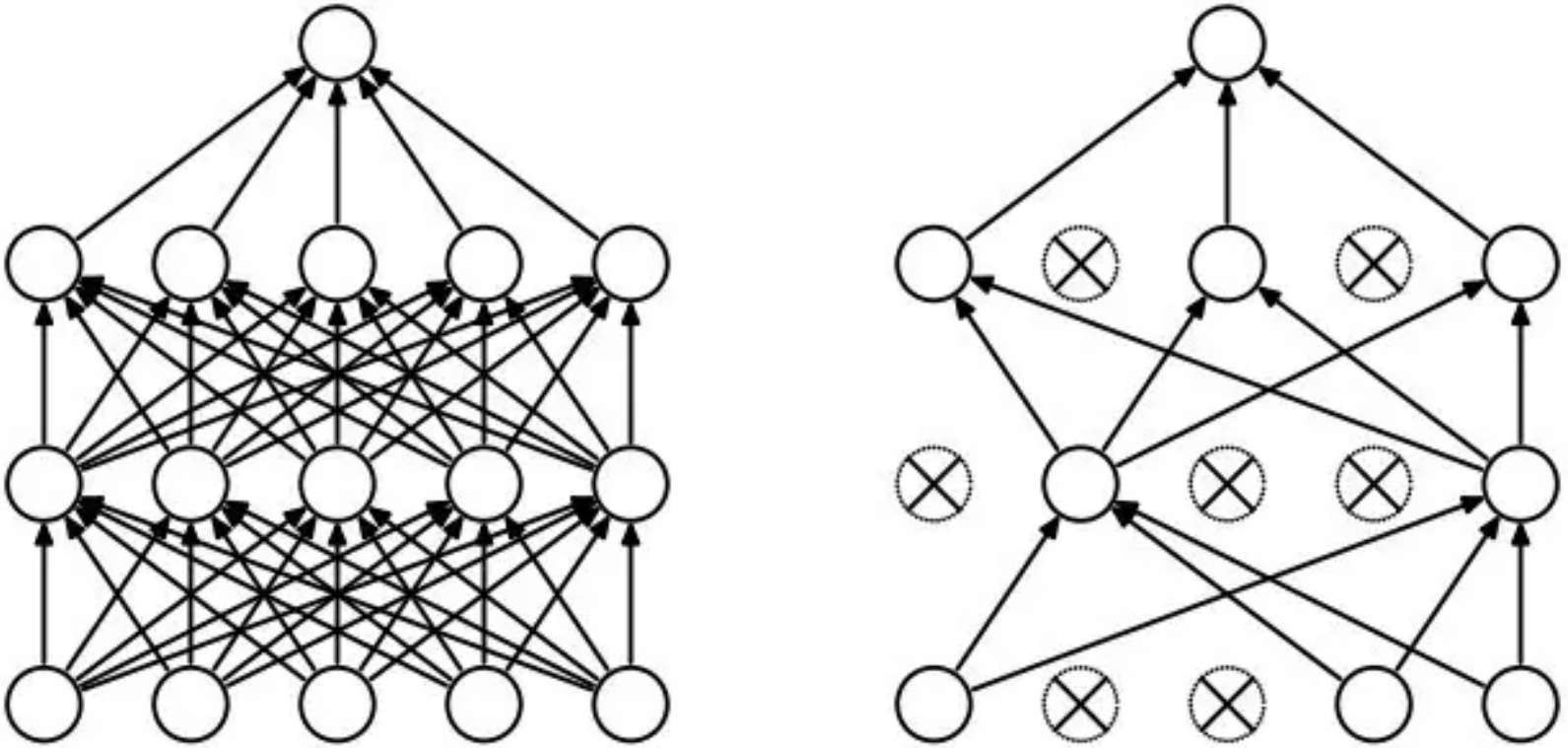




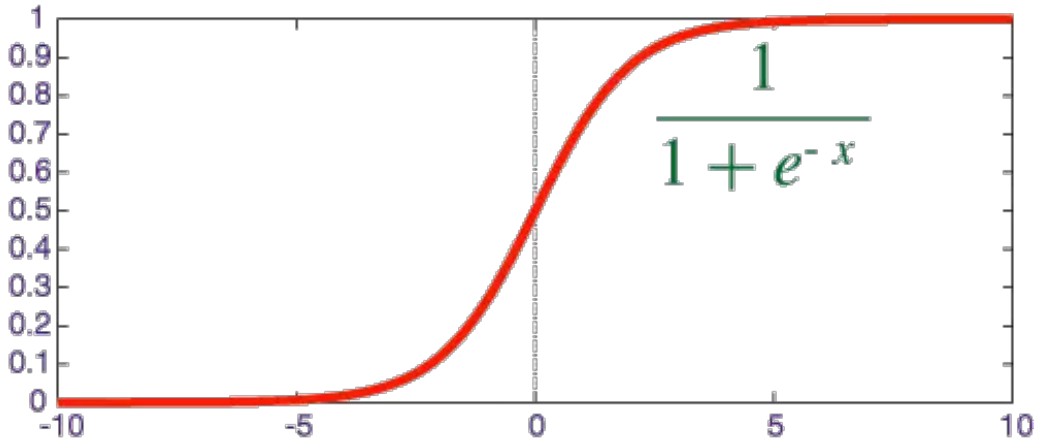
How to detect it in training process?

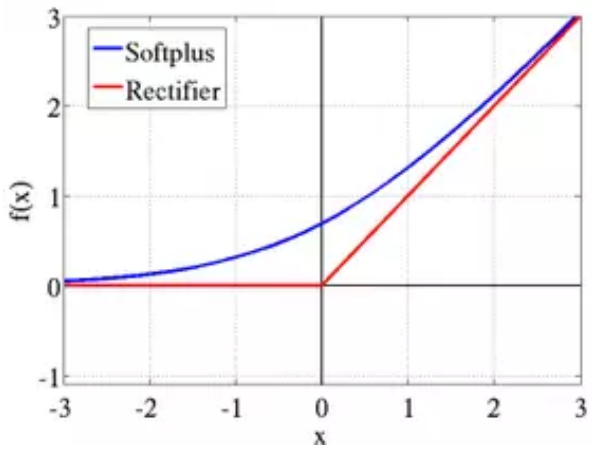


Dropout



Sigmod → ReLU


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-10
-5
0
5
10
$\frac{1}{1+e^{-x}}$


Softplus
Rectifier
f(x)
x
3
2
1
0
-1
-3
-2
-1
0
1
2
3

Sigmod → ReLU