CAP 4453
Robot Vision

Dr. Gonzalo Vaca-Castaño

gonzalo.vacacastano@ucf.edu

Credits

- Some slides comes directly from:
  - Yosesh Rawat
  - Justin Johnson
  - Andrew Ng

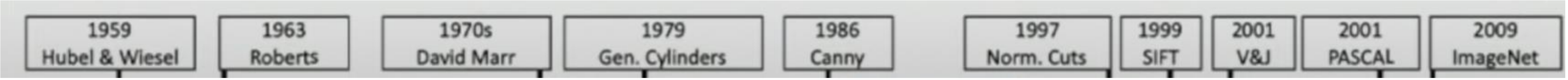
Robot Vision

14. Introduction to Deep Learning I

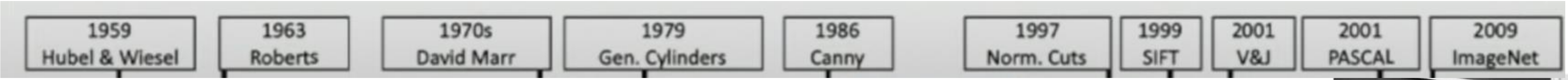
Outline

- What is Machine Learning ?
  - Main basic problems: regression, classification
  - Supervised vs unsupervised
  - generalization, overfitting
- What is Deep learning?
  - What is Neural network
  - Activation functions
  - Define error
  - What are you optimizing?
  - Chain rule
  - Back-propagation
  - Why deep? How deep?
    - Hyper-parameters
  - Problems with NN. What happened in the 80's?
    - Vanishing gradient problem
    - Number of parameters
- What kind of problems DN can solve?
  - Regression, classification
  - Computer vision: object detection, semantic segmentation, super-resolution,
  - Time series: NLP, visual questioning/answering
  - Generative models: impersonators ()

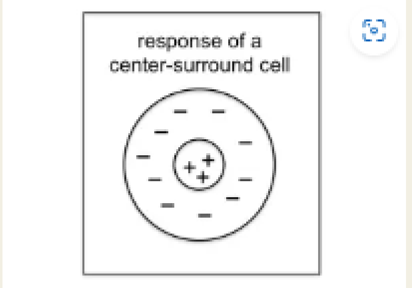
Introduction


1959
Hubel & Wiesel
1963
Roberts
1970s
David Marr
1979
Gen. Cylinders
1986
Canny
1997
Norm. Cuts
1999
SIFT
2001
V&J
2001
PASCAL
2009
ImageNet

Hubel and Wiesel



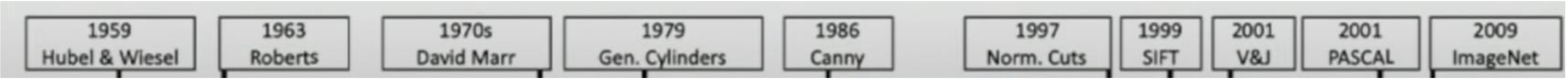
- In **1959**, **David H. Hubel** and **Torsten N. Wiesel** conducted groundbreaking experiments that significantly expanded our understanding of sensory processing (visual system)
- Experiments on cats and kittens as models for humans, and in the 1970s they repeated the experiments on primates.
- In 1981 they won Nobel Prize
- Each cell fires when you shine light in a specific small circula area of the visual field, with different cells responding to light in different places.
- Record the activity of neurons in cat brains while presenting various visual stimuli



Depiction of center-surround cell responses. This cell is excited by light presented in a small, circular central area (plus signs) and inhibited by light in the surrounding area (minus signs).

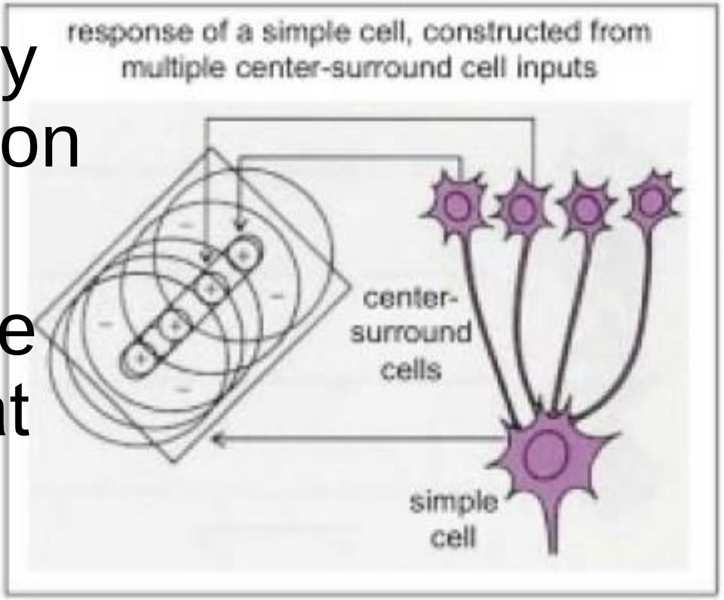
[How Hubel and Wiesel Revolutionized Neuroscience and Made Me a Neuroscientist | Brains Explained (brains-explained.com)]

Hubel and Wiesel



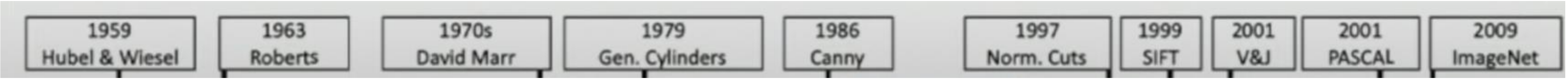
- Trying respond to traditional visual stimuli such as dots: No response.
- Showed pictures of beautiful women from a magazine: No response.

- they were recording from a silent neuron, it suddenly started firing like crazy as they changed the projection slide that they were using to present the stimuli.
- Turns out, the cell was responding to the edge of the slide. That's when Hubel and Wiesel discovered that there are cells specialized for detecting lines



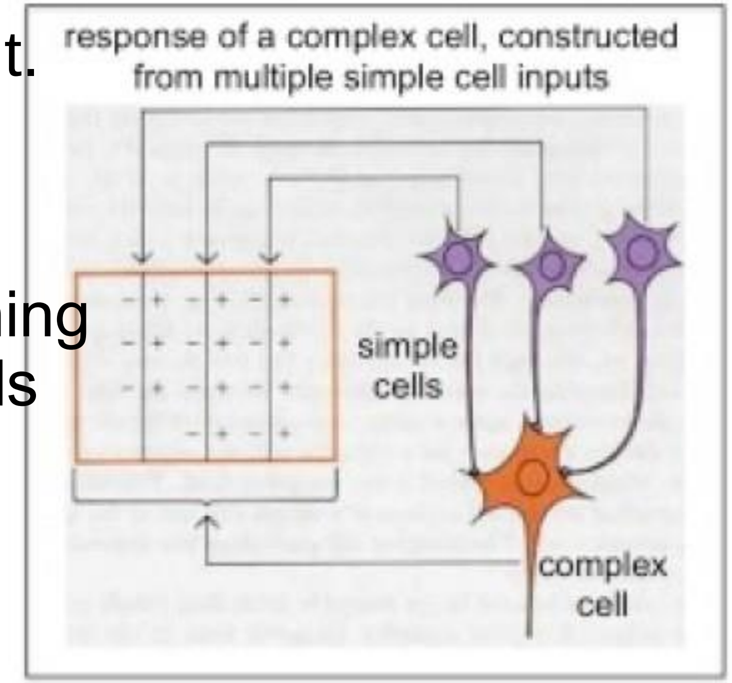
[How Hubel and Wiesel Revolutionized Neuroscience and Made Me a Neuroscientist | Brains Explained (brains-explained.com)](brains-explained.com)

Hubel and Wiesel



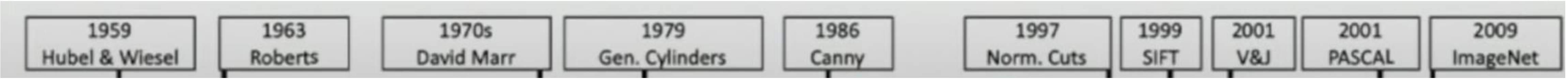
- They discovered another type of cell in the visual cortex, which they called a complex cell. complex cells were activated by lines of a specific orientation, but many of them responded best to a line that was moving steadily through space.

- So now the brain can detect a totally new feature: movement.
- the visual system goes from detecting individual photons to detecting circles, lines, and movement!
- The brain isn't just a collection of neurons doing their own thing with their precious dendrites and axons; it's a network of cells that talk to each other and trade information.

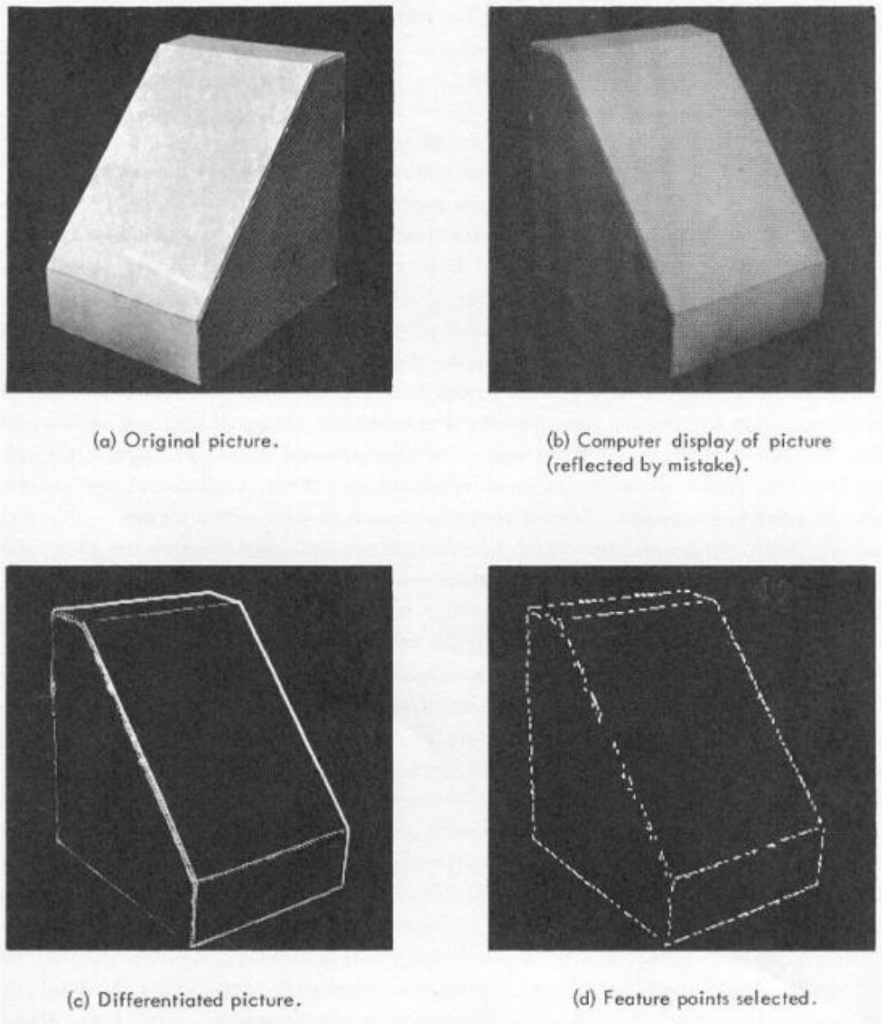


[How Hubel and Wiesel Revolutionized Neuroscience and Made Me a Neuroscientist | Brains Explained (brains-explained.com)](brains-explained.com)

Larry Roberts(1963)



- computer recognition of three-dimensional objects from image capture
- edge finding (the first edge operator), line fitting, and model-based object recognition.
- First computer vision thesis
- Roberts is considered one of the fathers of the modern internet
  - Help invent Packet Switching
  - Distributed control (multiple computers)
  - Idea of satellite connections



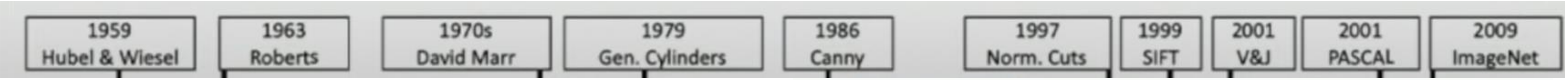
(a) Original picture.

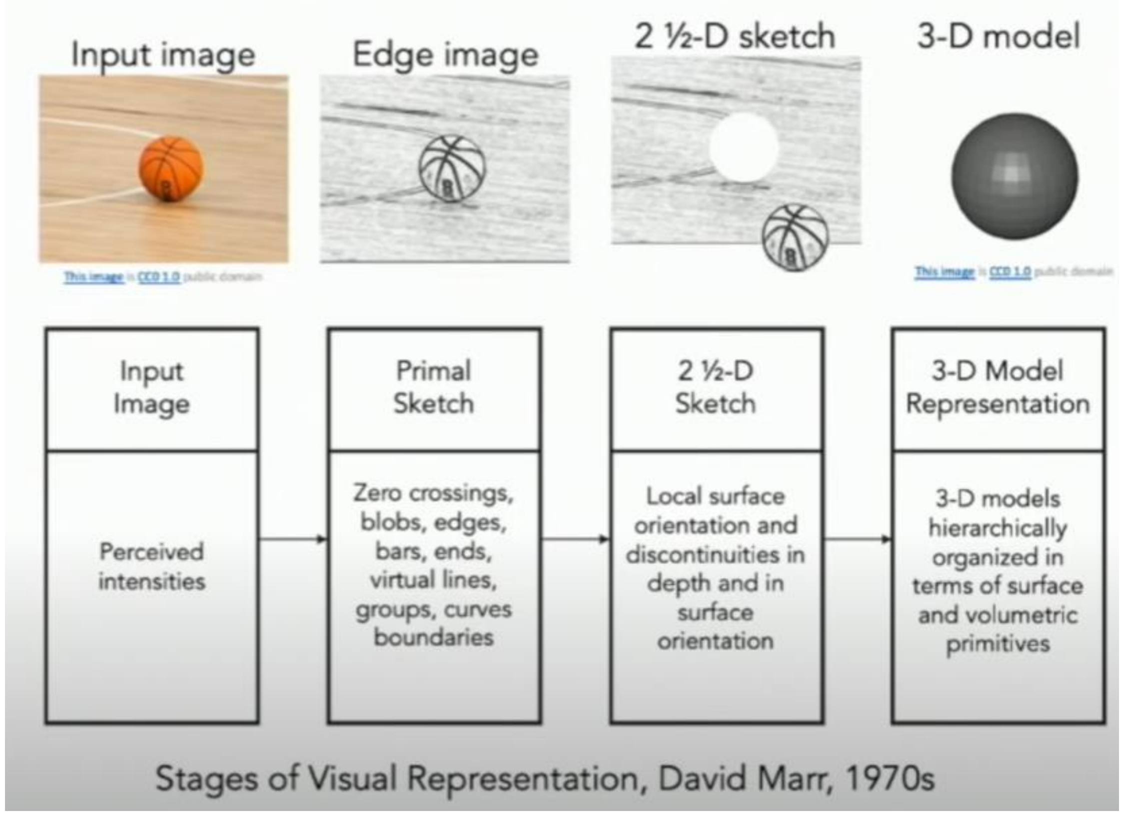
(b) Computer display of picture (reflected by mistake).

(c) Differentiated picture.

(d) Feature points selected.

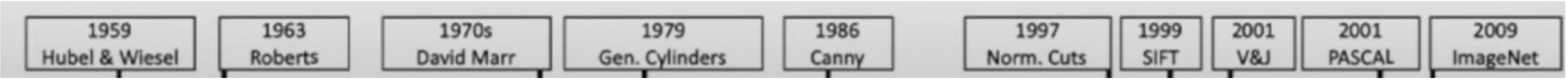
David Marr 1970s





- The ICCV best-paper award is the Marr Prize, named after British neuroscientist David Marr

Generalized cylinders



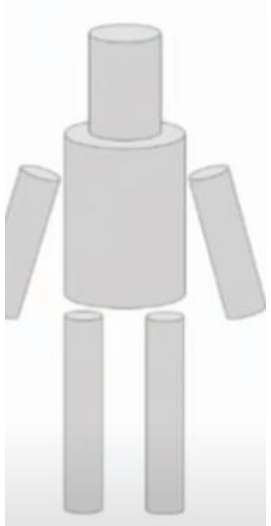
- Generalized Cylinder

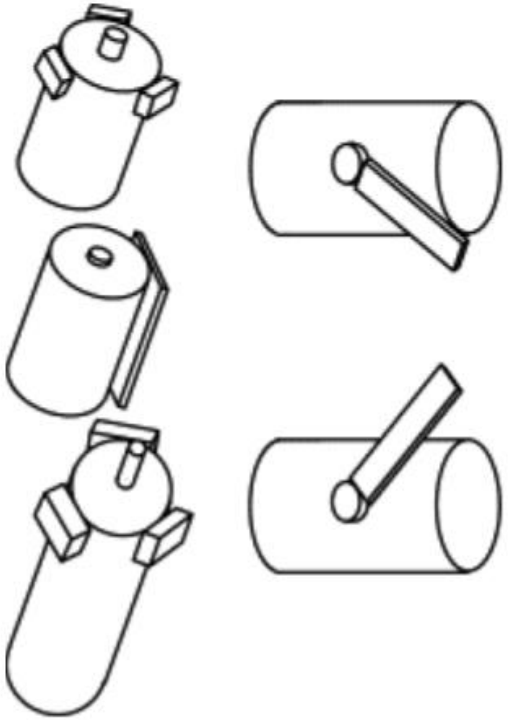
  Brooks & Binford, 1979

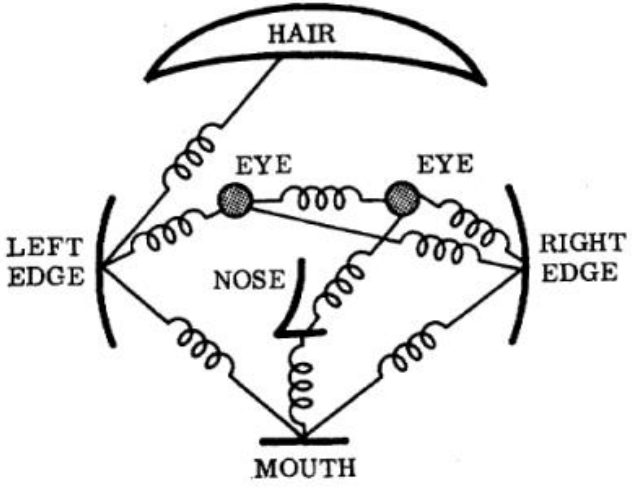
- Pictorial Structure

  Fischler and Elschlager, 1973

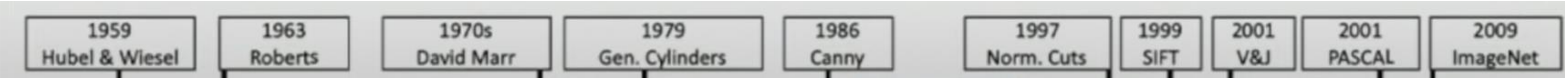
- Model complex objects from simpler parts



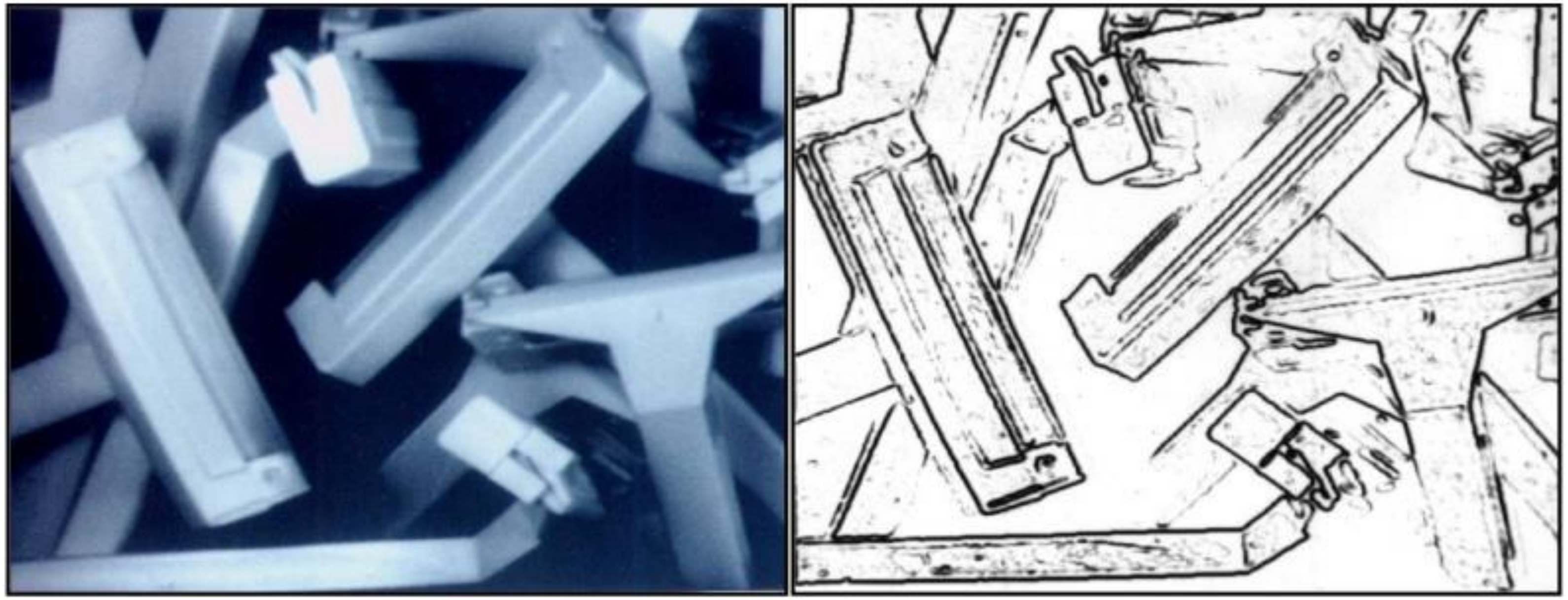




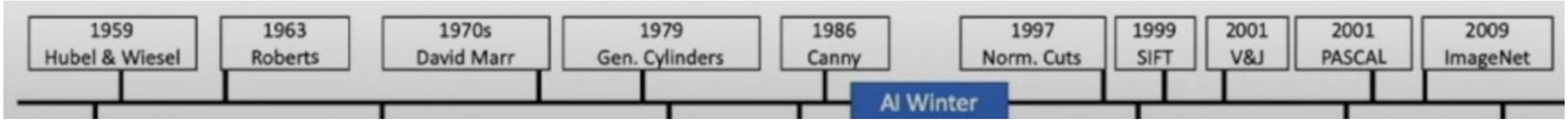
Edges (canny)

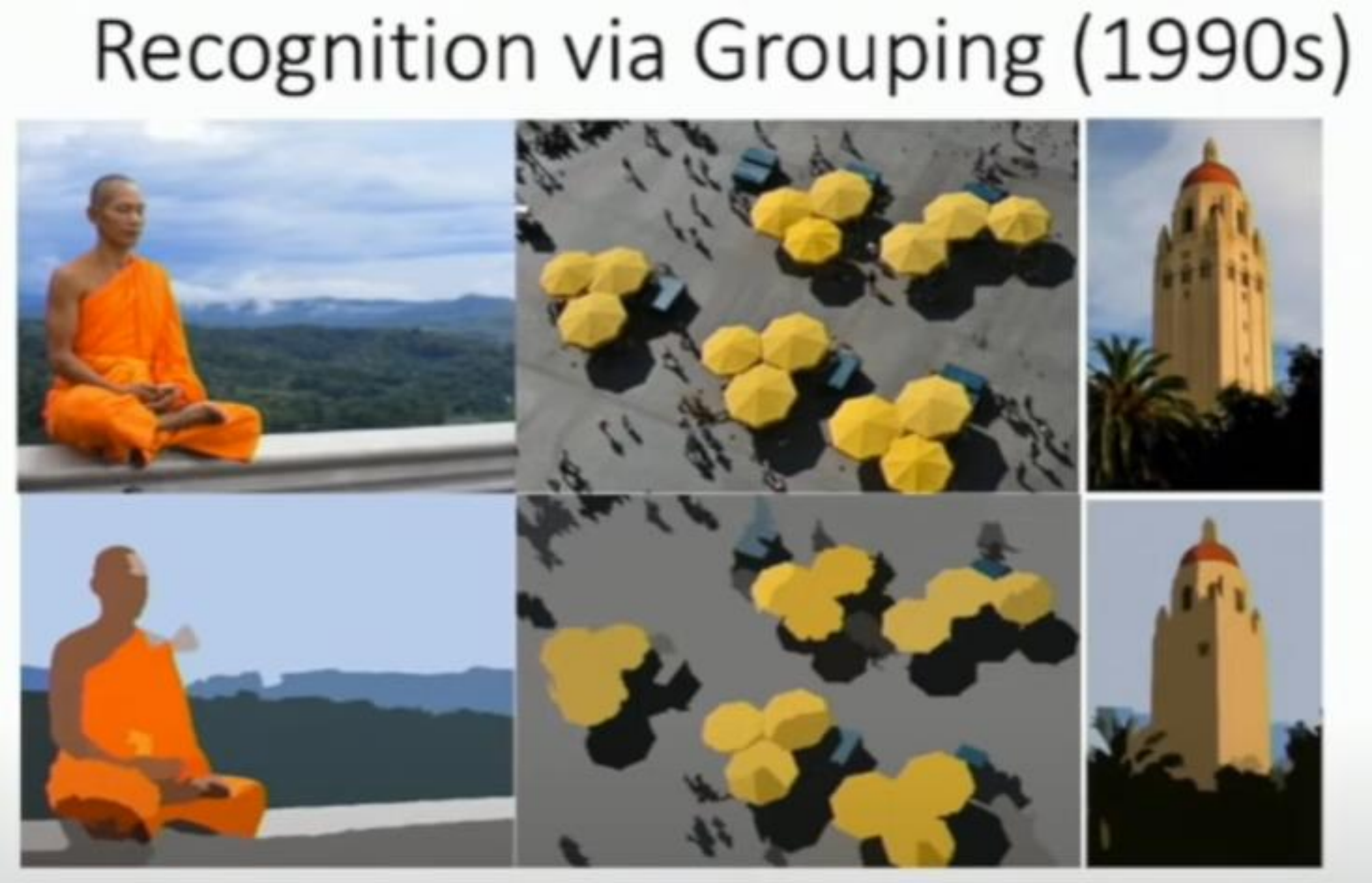


- Recognize objects from edges

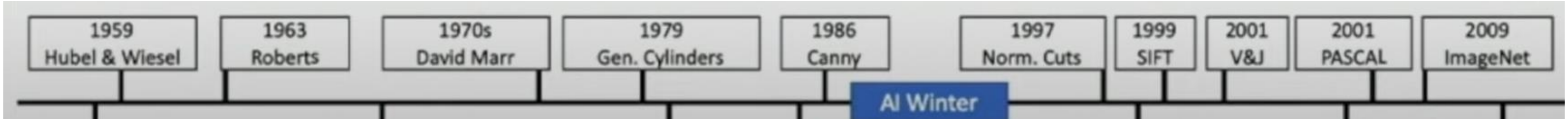


Normalized cuts (1997)



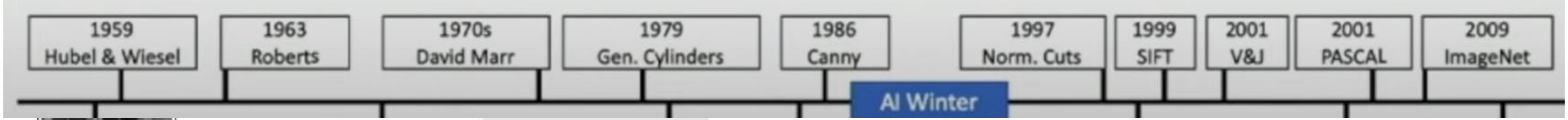


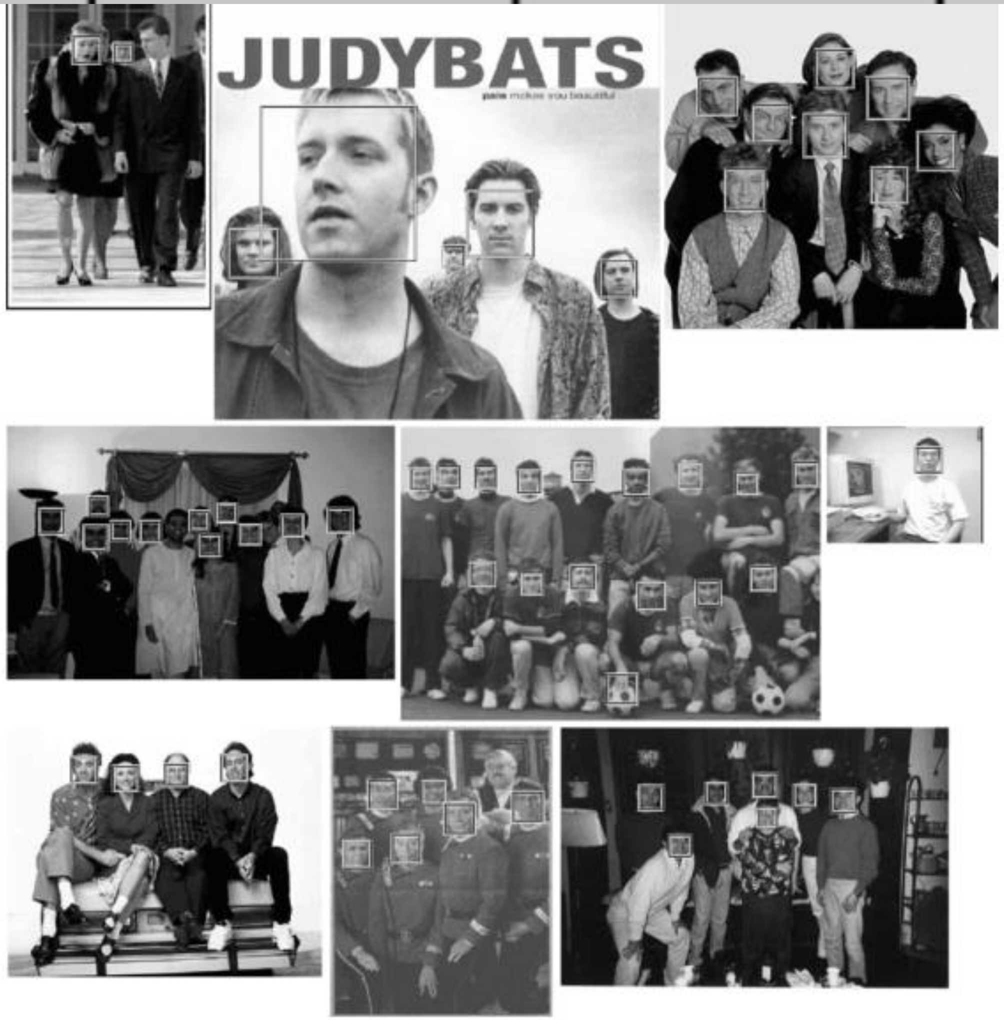
SIFT (1999)


1959
Hubel & Wiesel
1963
Roberts
1970s
David Marr
1979
Gen. Cylinders
1986
Canny
AI Winter
1997
Norm. Cuts
1999
SIFT
2001
V&J
2001
PASCAL
2009
ImageNet


Recognition via Matching (2000s)
STOP
STOP

Voila and Jones (2001): Face detection





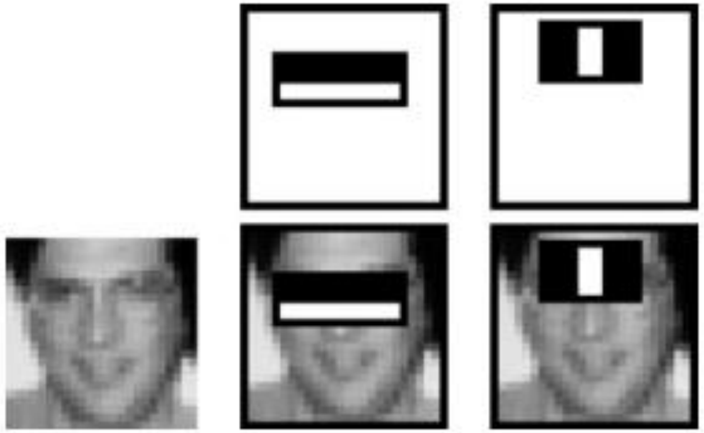
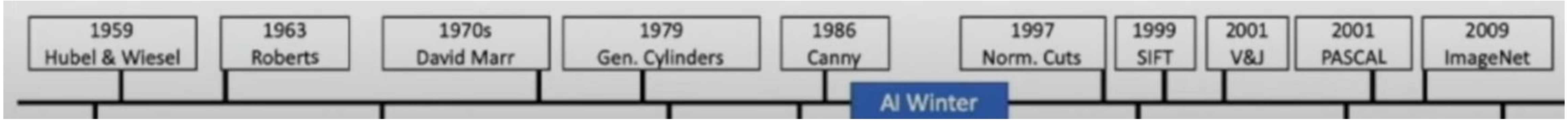


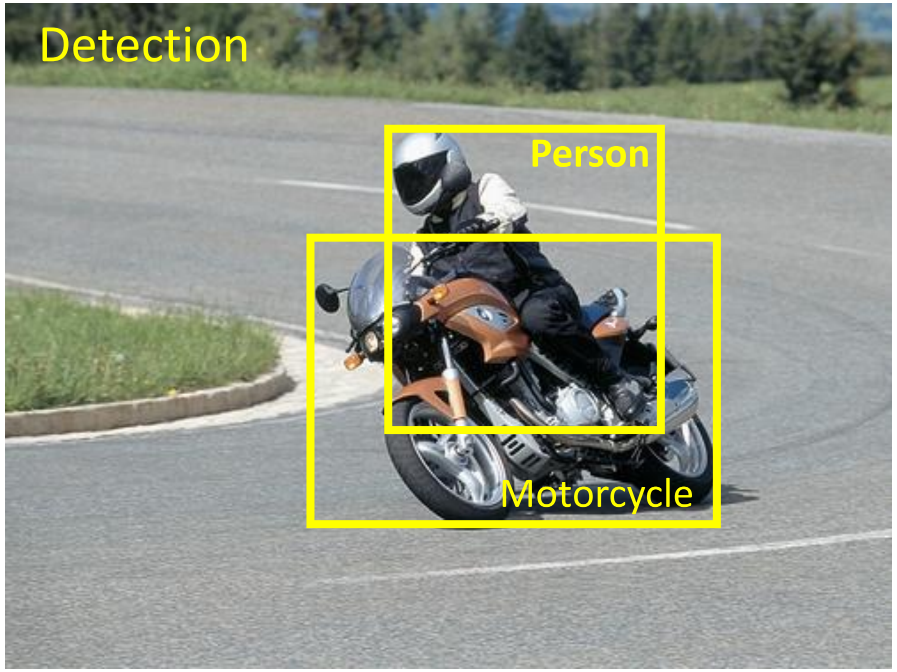
Figure 5. The first and second features selected by AdaBoost. The two features are shown in the top row and then overlayed on a typical training face in the bottom row. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. The feature capitalizes on the observation that the eye region is often darker than the cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose.

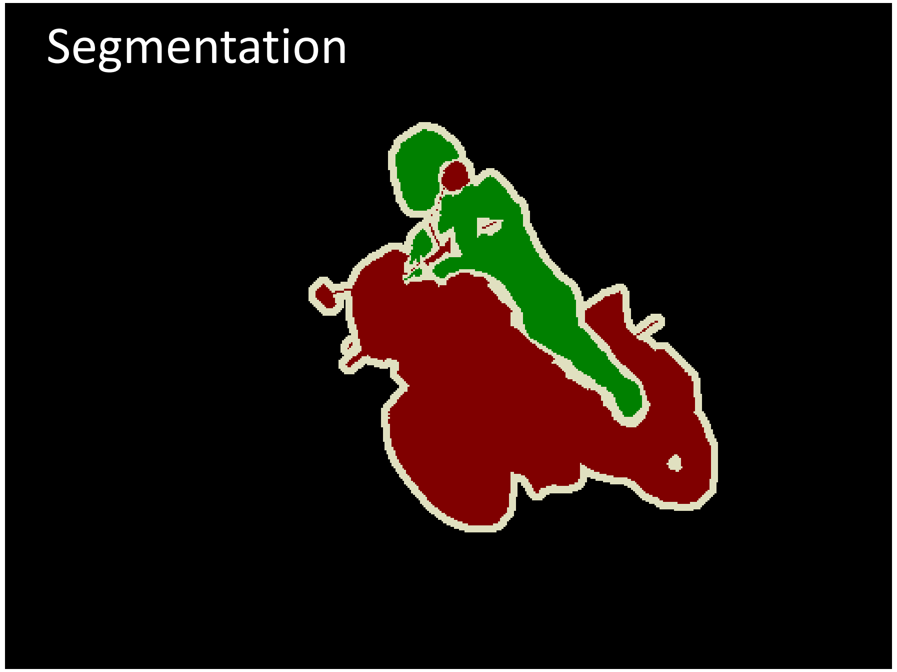
Pascal Challenge: Object detection (2005-2012)



20 object classes **22,591 images**

Classification: person, motorcycle

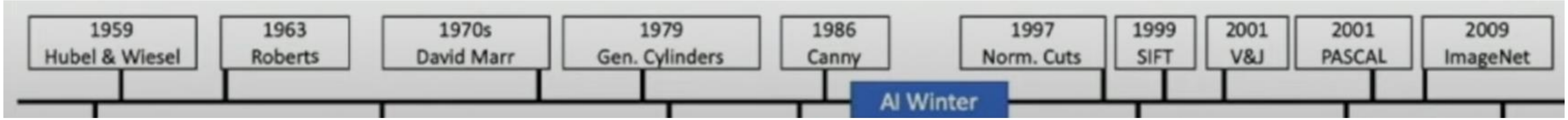




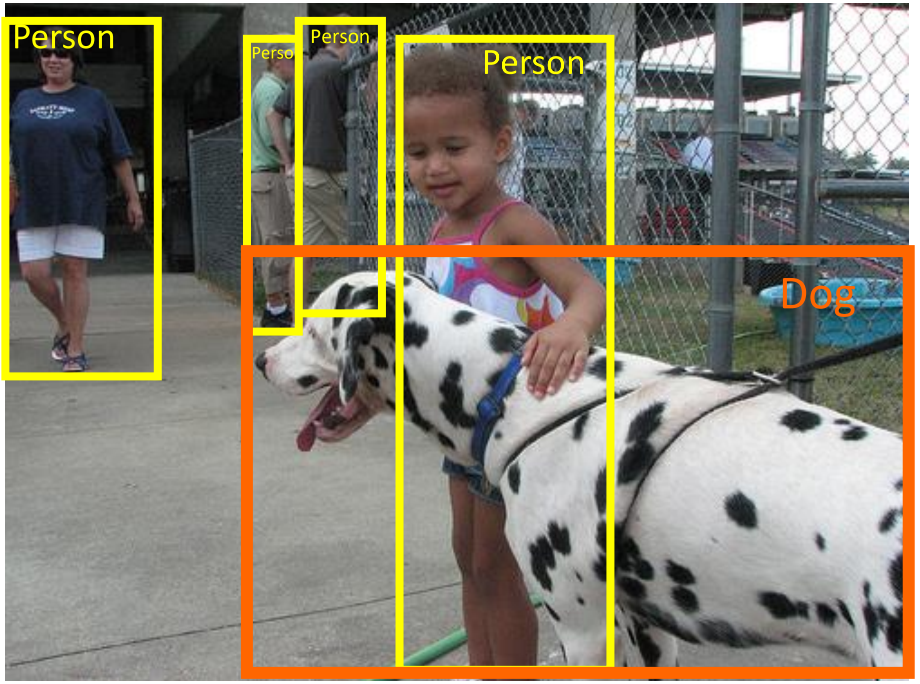
Action: riding bicycle

Everingham, Van Gool, Williams, Winn and Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. IJCV 2010.

IMAGENET Large Scale Visual Recognition Challenge (ILSVRC) 2010-2014

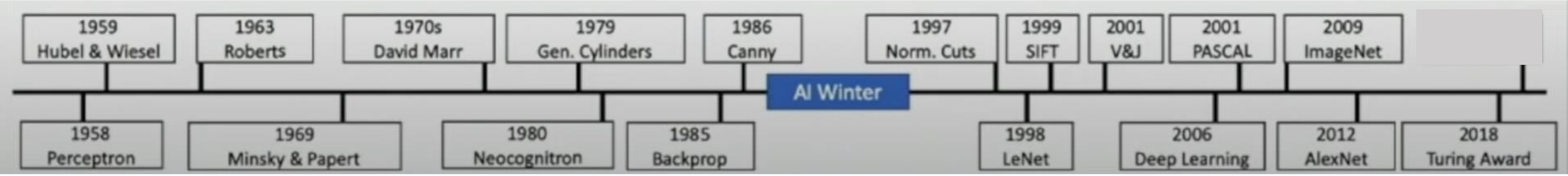


200 object classes	**517,840 images**	**DET**
1000 object classes	**1,431,167 images**	**CLS-LOC**

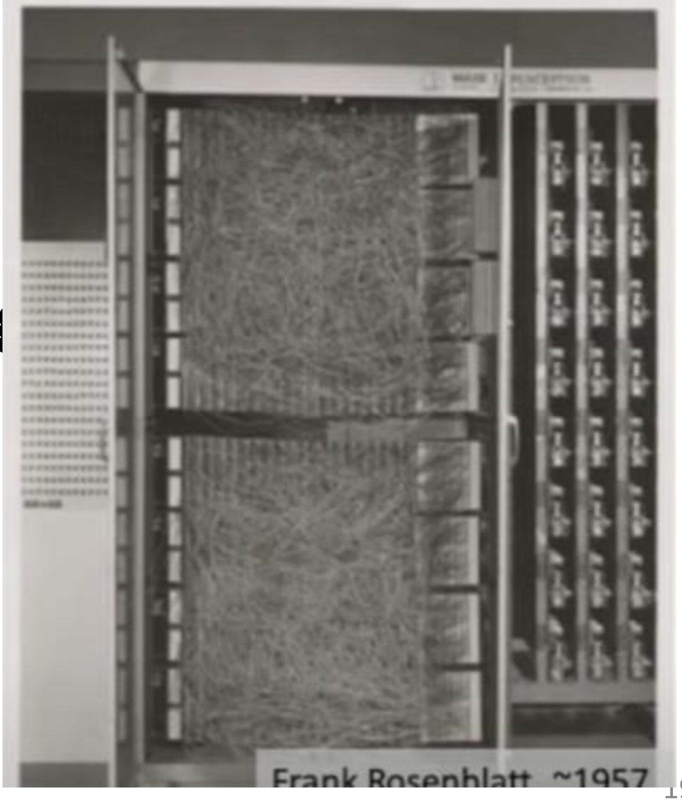


http://image-net.org/challenges/LSVRC/

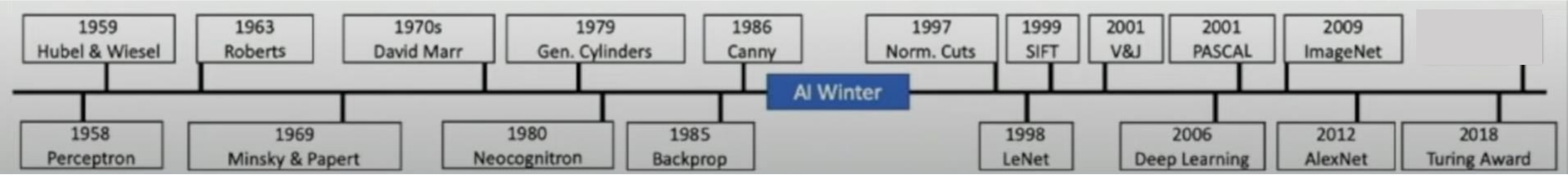
Perceptron 1958



- One of the earliest algorithm could learn from data
  - Linear classifier
- Implemented in hardware. Weights stored in potentiometers updated with electric motors during learning
- Connected to a 20x20 camera
- Could learn to recognize letters

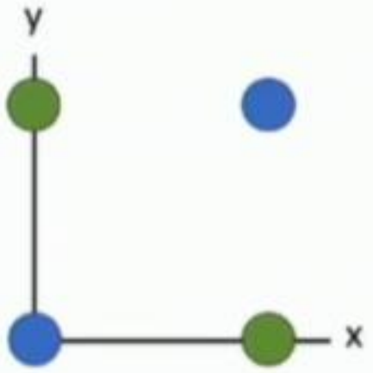


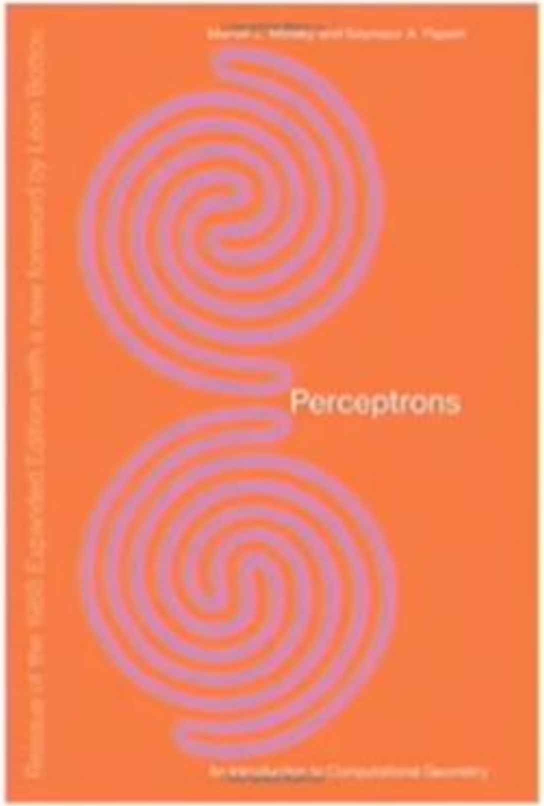
Mutilayer Perceptron idea



Minsky and Papert, 1969

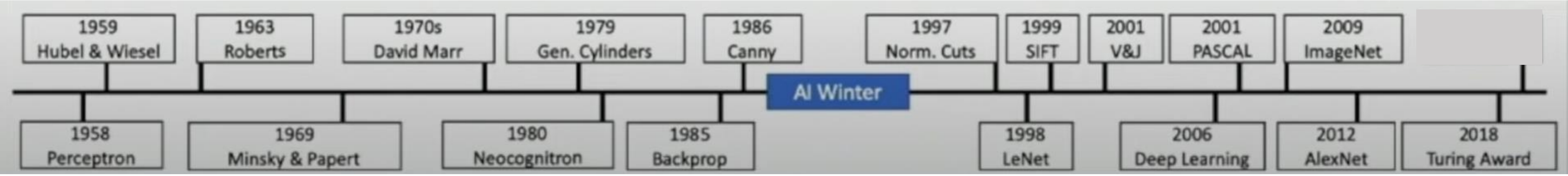
X	Y	F(x,y)
0	0	0
0	1	1
1	0	1
1	1	0





Showed that Perceptrons could not learn the XOR function
Caused a lot of disillusionment in the field

Convolutional Networks

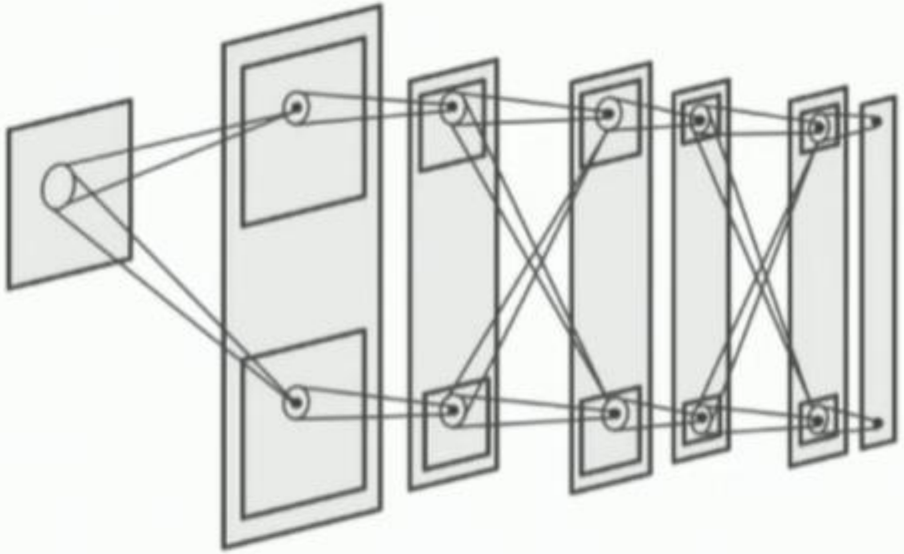


Neocognitron: Fukushima, 1980

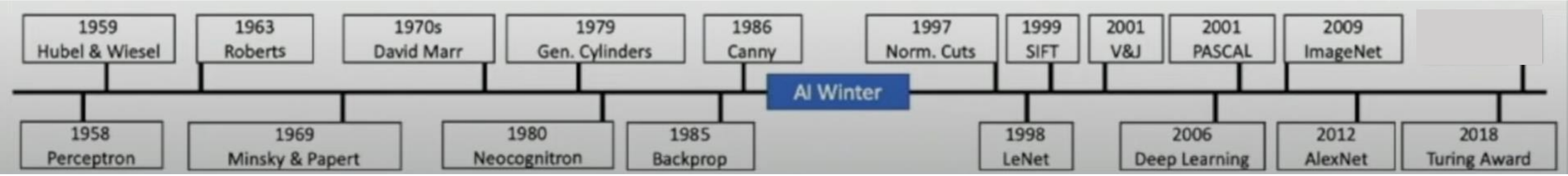
Computational model the visual system, directly inspired by Hubel and Wiesel's hierarchy of complex and simple cells

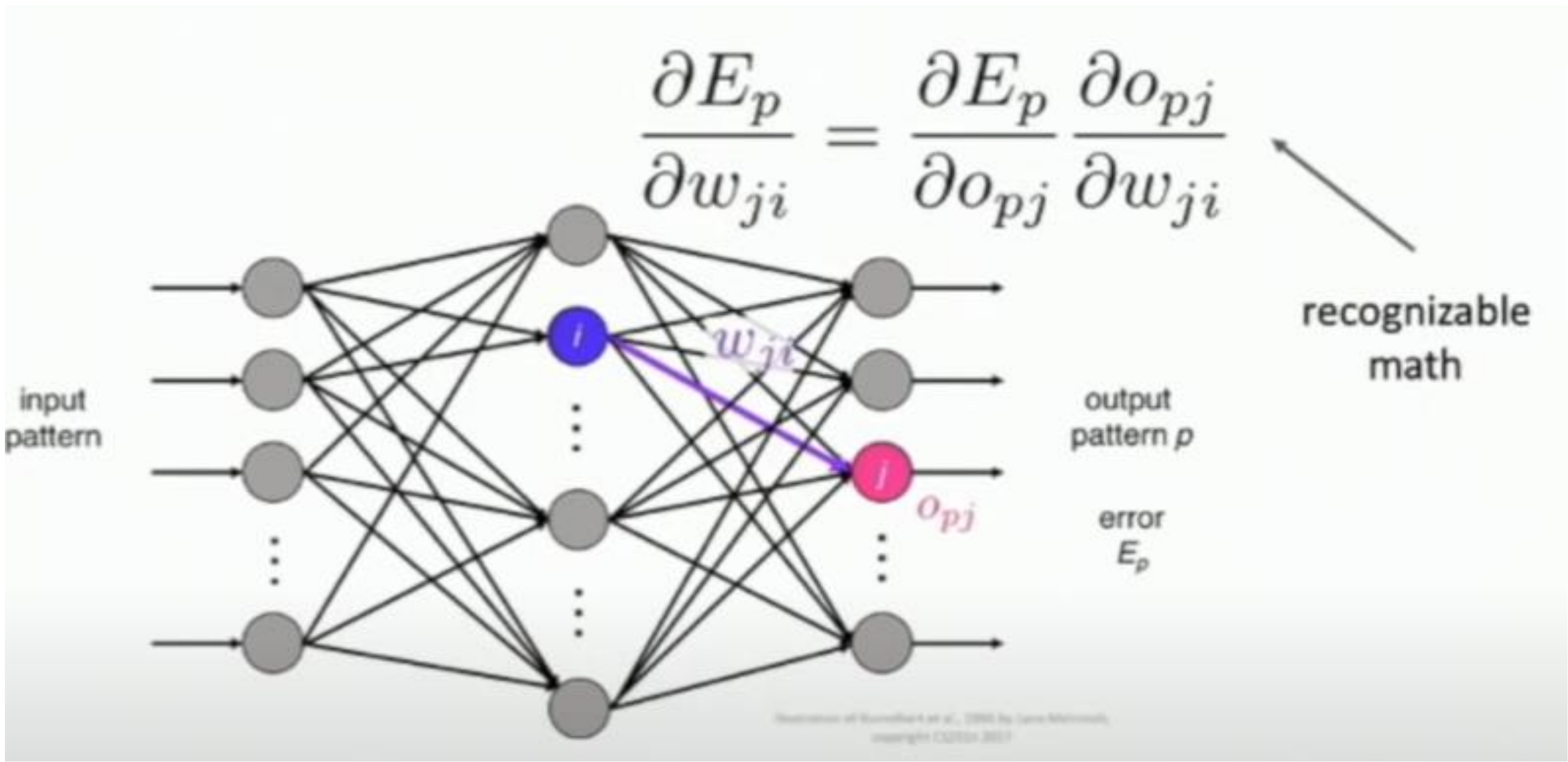
Interleaved simple cells (convolution) and complex cells (pooling)

No practical training algorithm



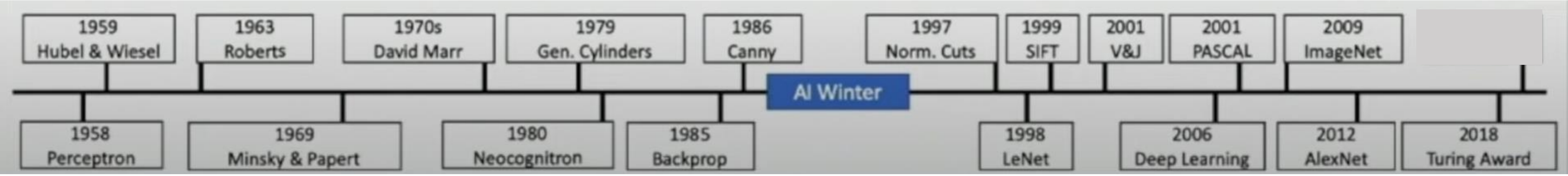
Back propagation





Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J. (1986-10-09). "Learning representations by back-propagating errors". *Nature*. **323** (6088): 533–536

Lenet: First trainable convolutional neural network



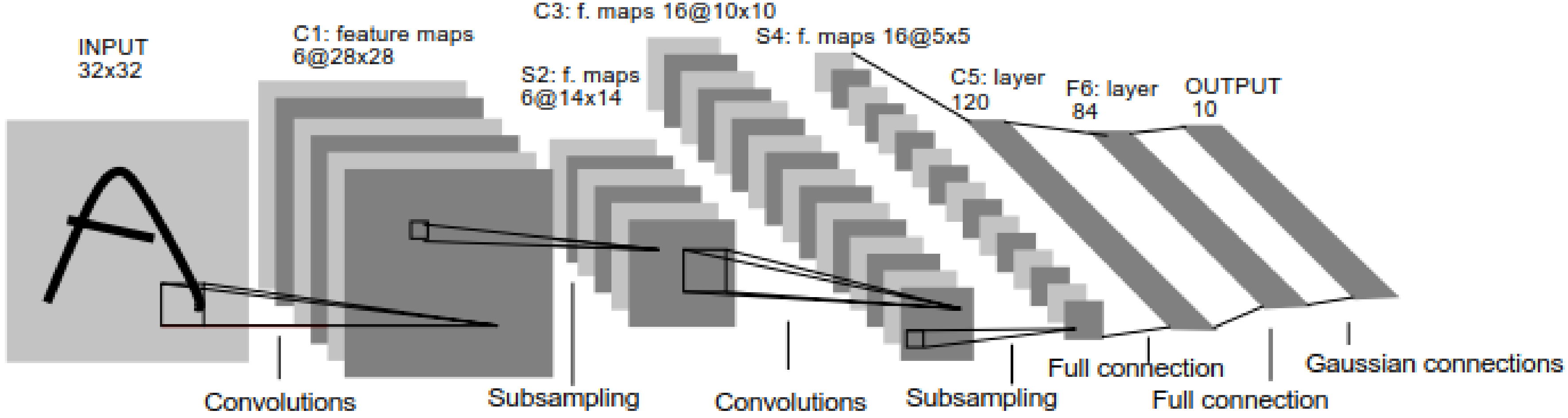
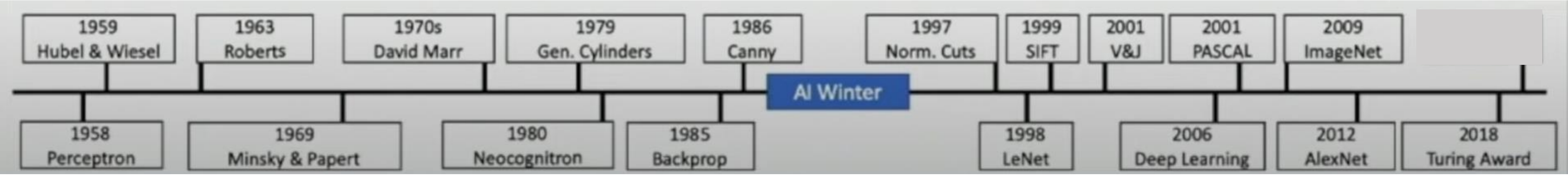


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998

Deep learning

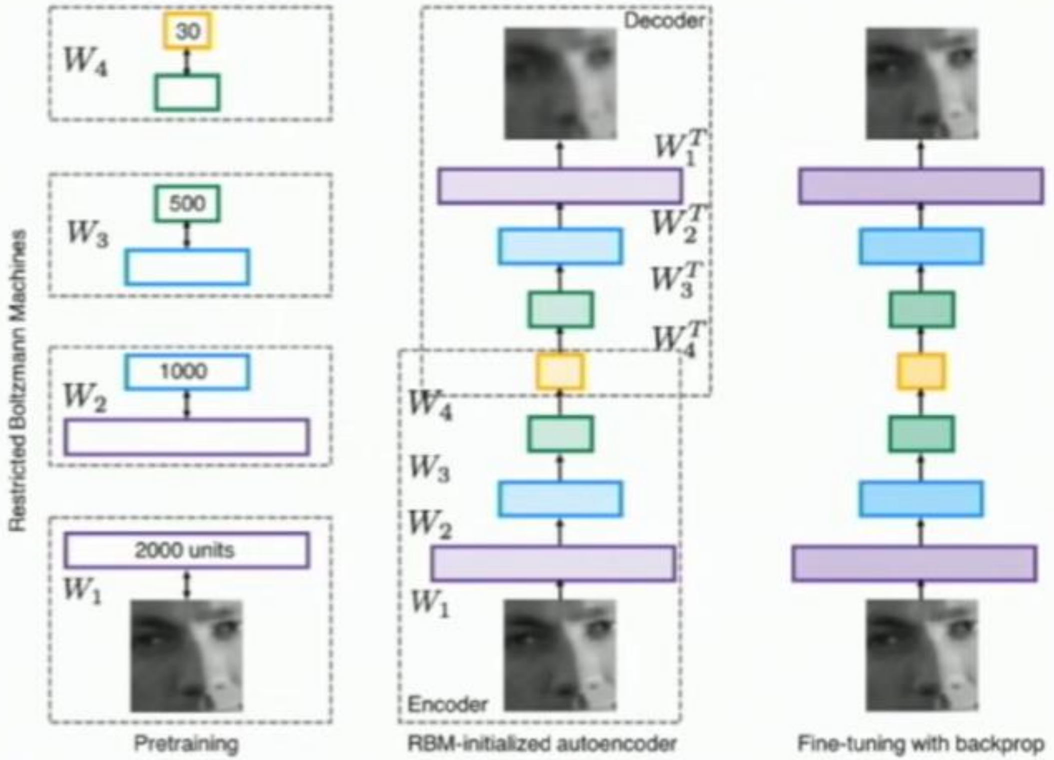


2000s: "Deep Learning"

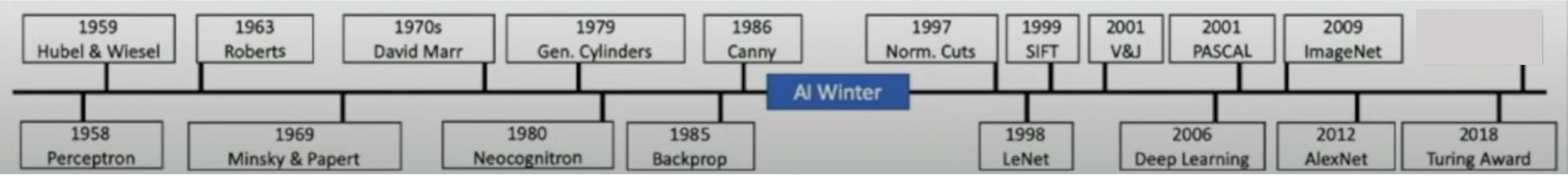
People tried to train neural networks that were deeper and deeper

Not a mainstream research topic at this time

Hinton and Salakhutdinov, 2006
Bengio et al, 2007
Lee et al, 2009
Glorot and Bengio, 2010

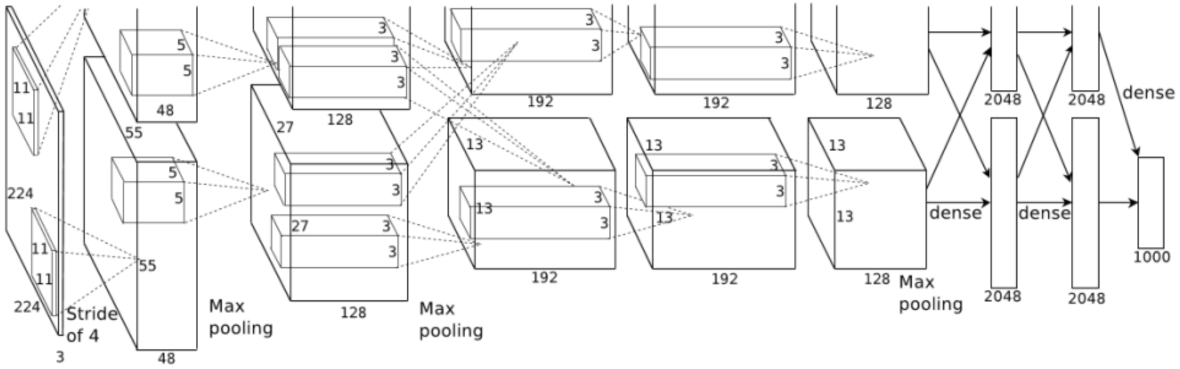


AlexNet



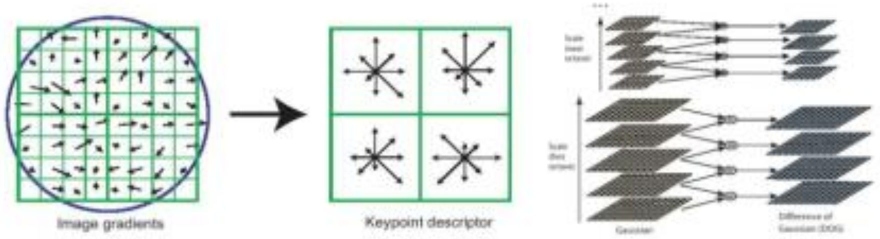
AlexNet

- 1.2 million high-resolution images from ImageNet LSVRC-2010 contest
- 1000 different classes (softmax layer)
- NN configuration
  - NN contains 60 million parameters and 650,000 neurons,
  - 5 convolutional layers, some of which are followed by max-pooling layers
  - 3 fully-connected layers

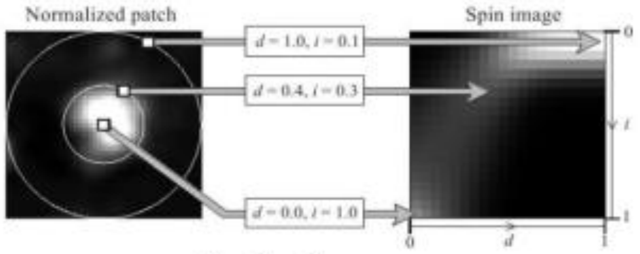


Krizhevsky, A., Sutskever, I. and Hinton, G. E. “ImageNet Classification with Deep Convolutional Neural Networks” NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada

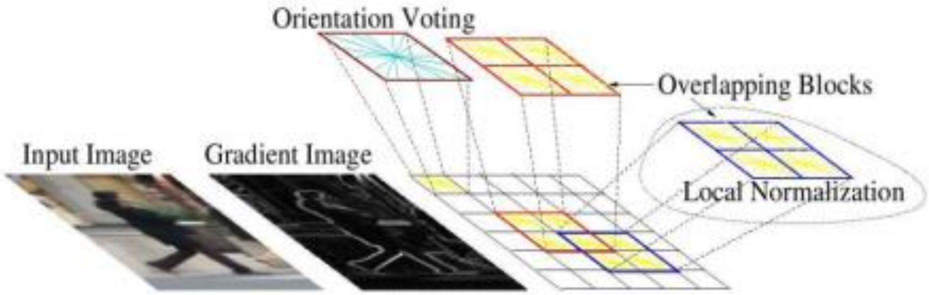
Some feature representations



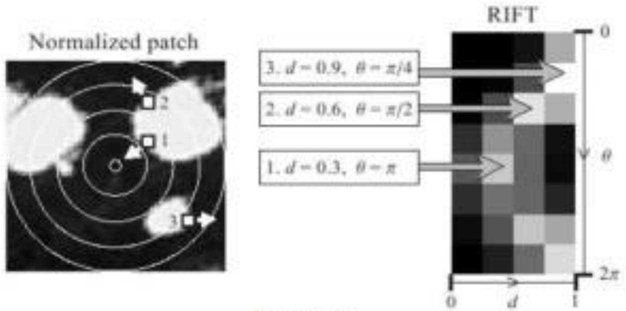
SIFT



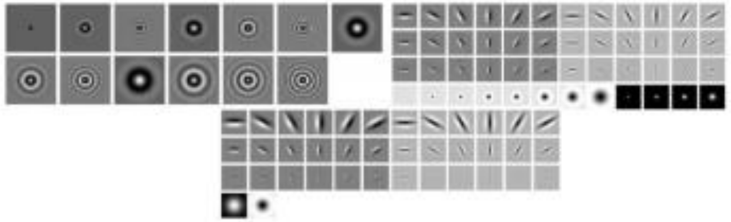
Spin image



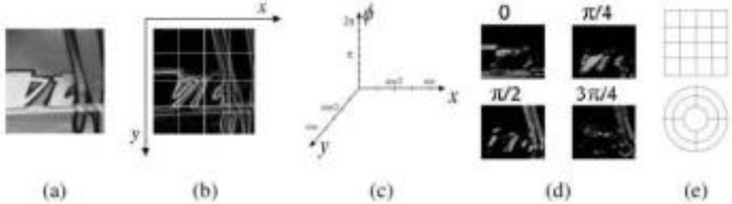
HoG



RIFT

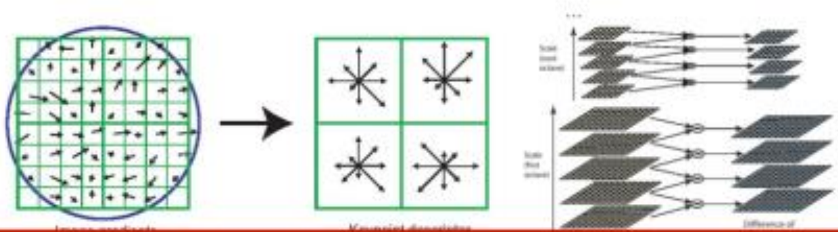


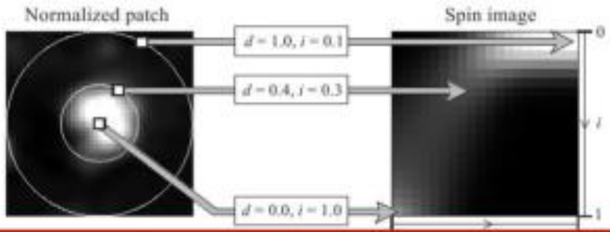
Textons



GLOH

Some feature representations

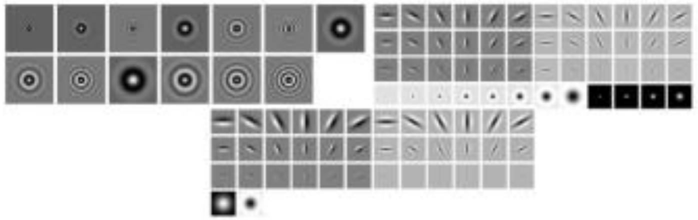




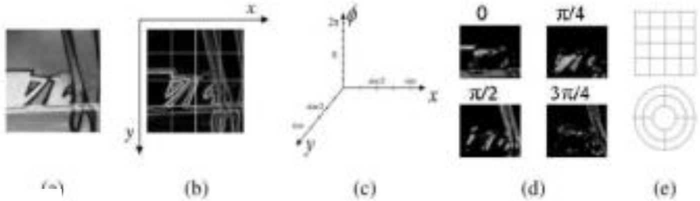
Coming up with features is often difficult, time-consuming, and requires expert knowledge.

HoG

RIFT



Textons

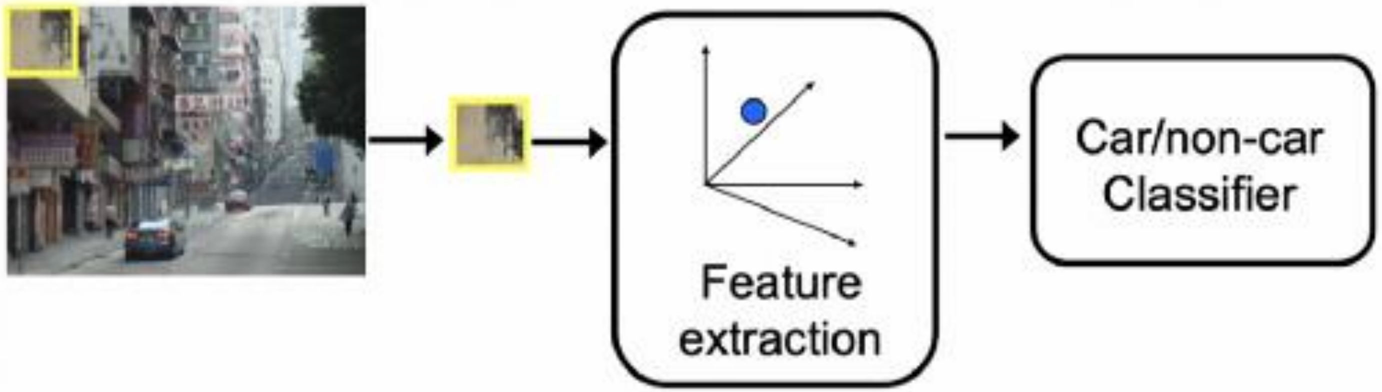


GLOH

What is Machine Learning ?

- *machine learning* is using data to detect patterns. It is the same thing as *AI*. *
- What is new?
  - faster
  - cheaper
  - Bigger
  - Feature engineering is generally replaced by Feature learning
- What is the goal of the algorithms?
  - make predictions about future observations of data in the same format (generalization)
  - *input data + weights → f (weights)*

* https://towardsdatascience.com/machine-learning-for-people-who-dont-care-about-machine-learning-4cf0495dee2c

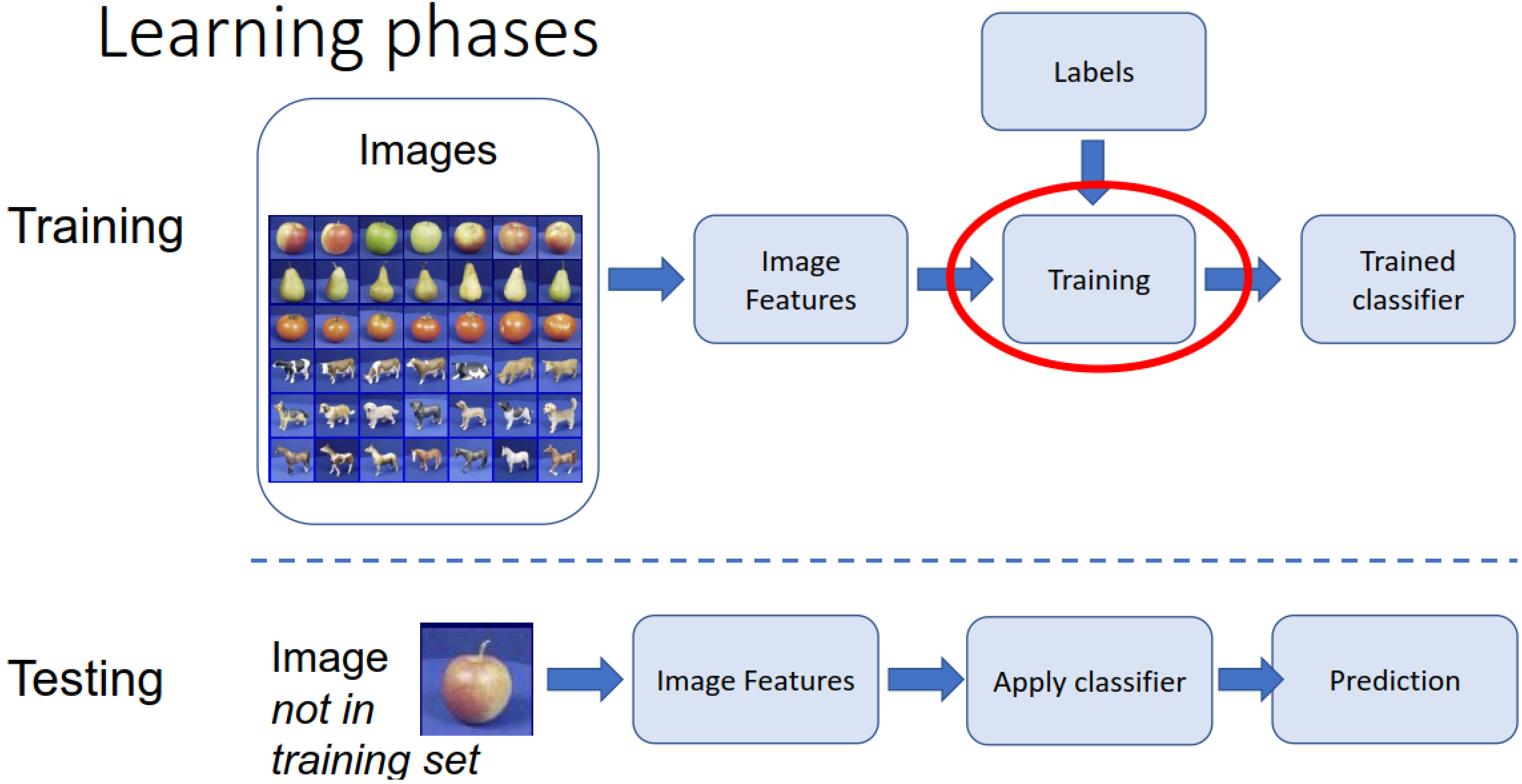


Today

Feature engineering -> Feature learning

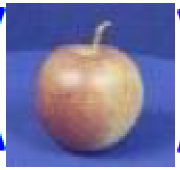
Expert knowledge Data

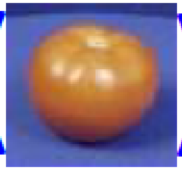
Learning phases

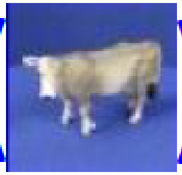


The machine learning framework

- Apply a prediction function to a feature representation of the image to get the desired output:

f() = “apple”

f() = “tomato”

f() = “cow”

The machine learning framework

$$\mathbf{f(x) = y}$$

Prediction function or *classifier* — Image feature — Output (label)

Training: Given a *training set* of labeled examples:

$$\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$$

Estimate the prediction function f by minimizing the prediction error on the training set.

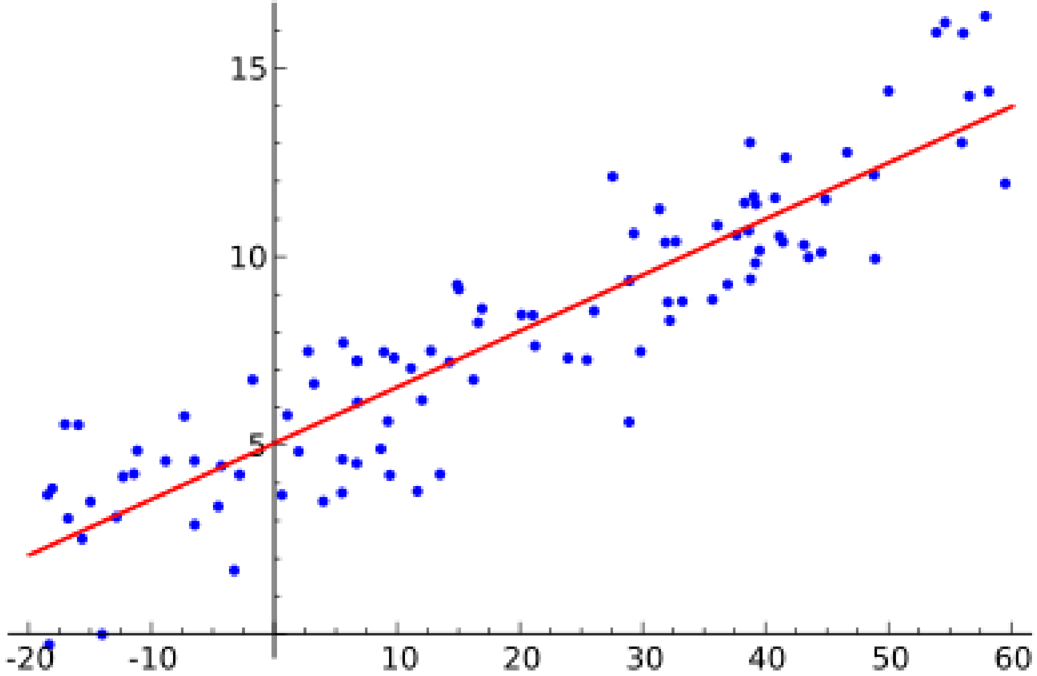
Testing: Apply f to an unseen *test example* $\mathbf{x}_u$ and output the predicted value $y_u = f(\mathbf{x}_u)$ to *classify* $\mathbf{x}_u$.

What is machine learning?

- If let's say f is a linear function in N dimensions ,$X = [x_1, x_2, \dots, x_N]$, what do you learn?
  - $f(w_1, w_2, \dots, w_{N+1}) = w_1 x_1 + w_2 x_2 + \cdots w_N x_N + w_{N+1}$
  - You learn the weights w that match better that function

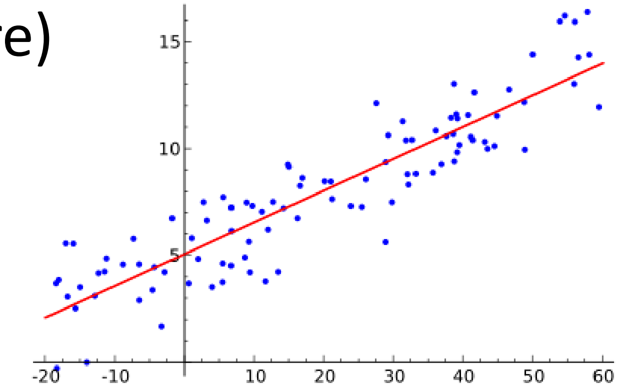
- Simplest case N=1,
  - *Input Data is number (X axis)*
  - *output value is the Y axis*
  - $f(w_1, w_2\,) = w_1 x_1 + w_2$

Finding these values is called **training**

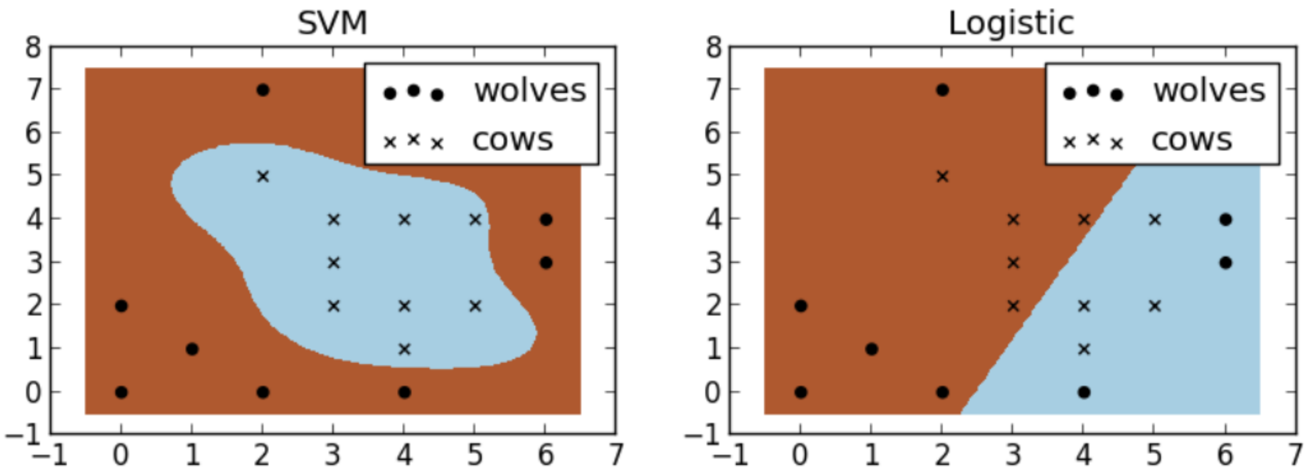


Basic problems in machine learning

- **You can break most of the machine learning problems in 2 categories**:
  - Regression: predicting a value (such as price or time to failure)



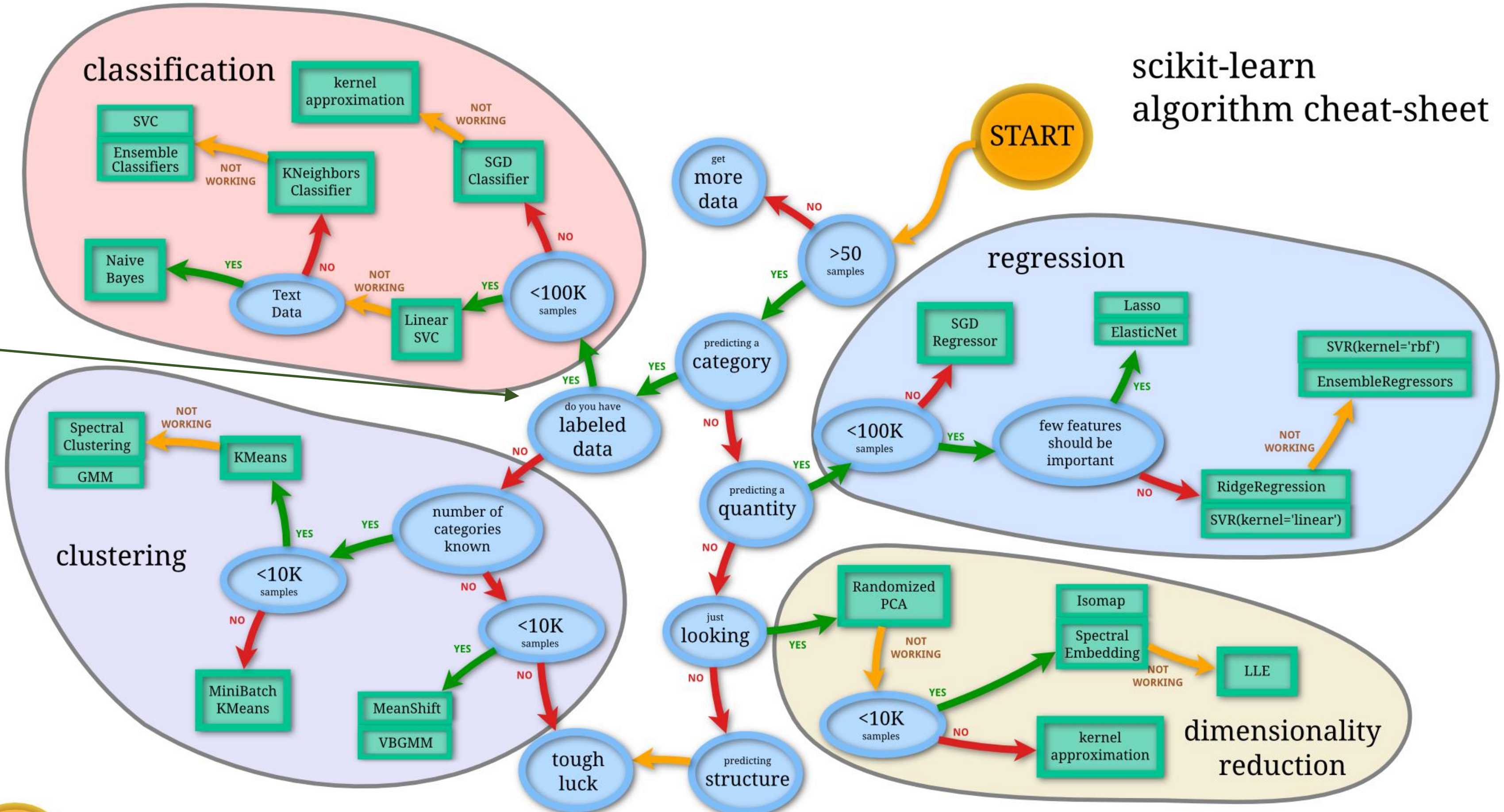
  - classification — predicting the category of something (dog/cat, good/bad, wolf/cow)



Basic problems in machine learning

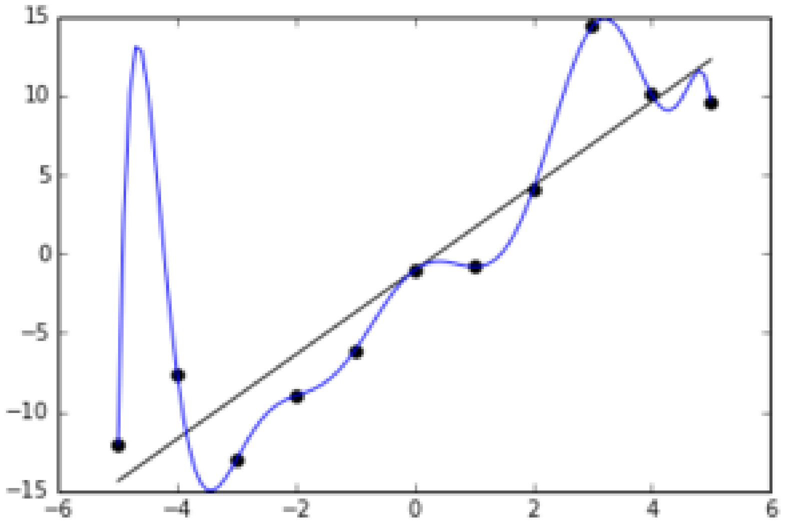
FROM SCIKIT-LEARN LIBRARY

- Supervised
- Unsupervised
- Semi-supervised

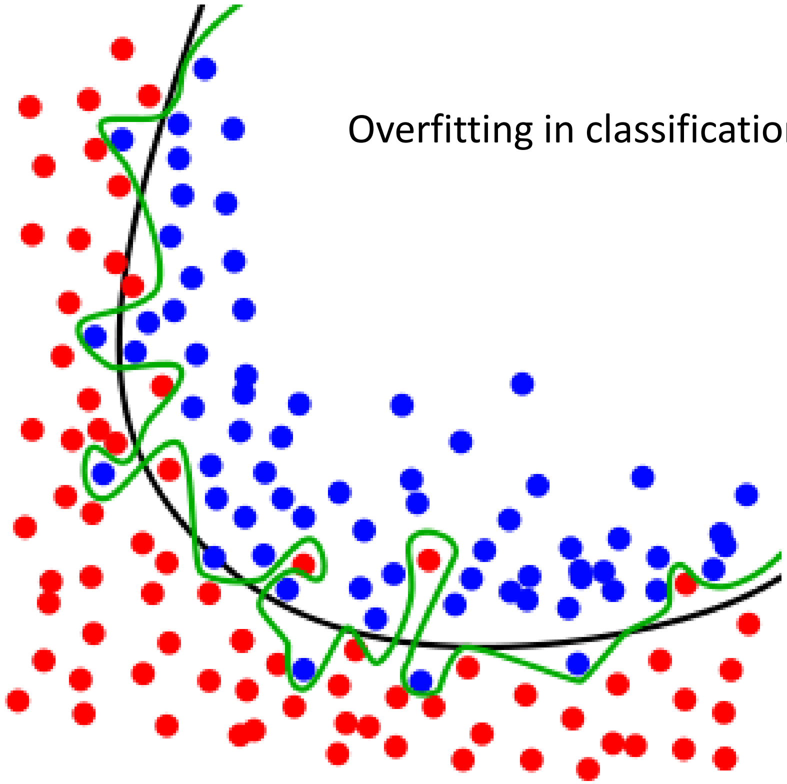


Generalization AND overfitting
WITH TRAINING DATA

Overfitting in regression

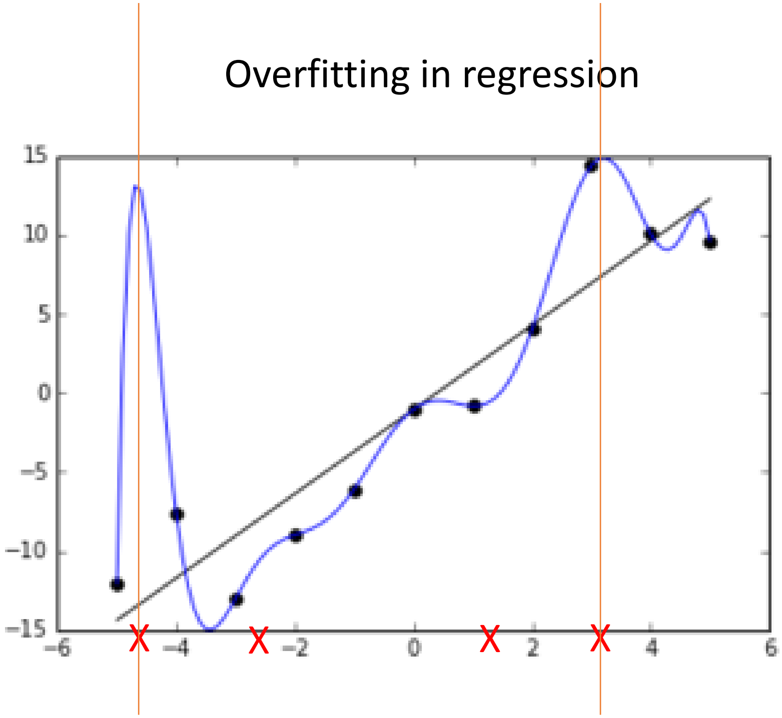


Overfitting in classification

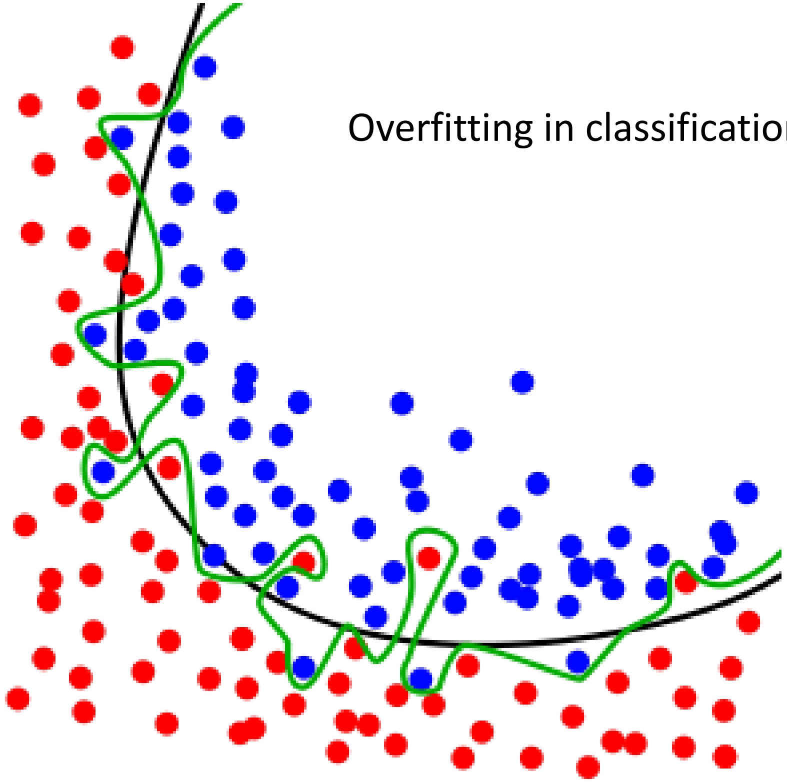


Generalization AND overfitting

WITH NEW TESTING DATA



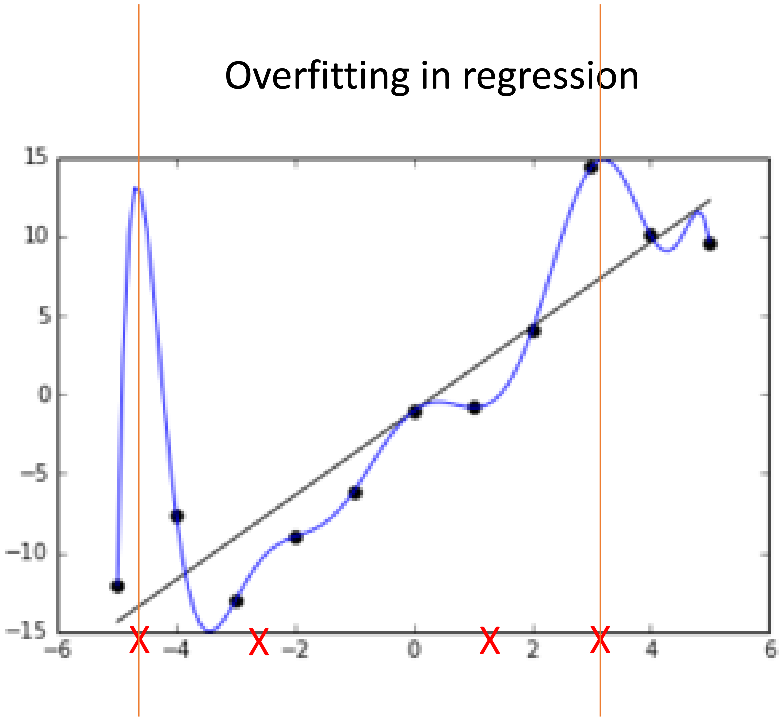
Overfitting in regression



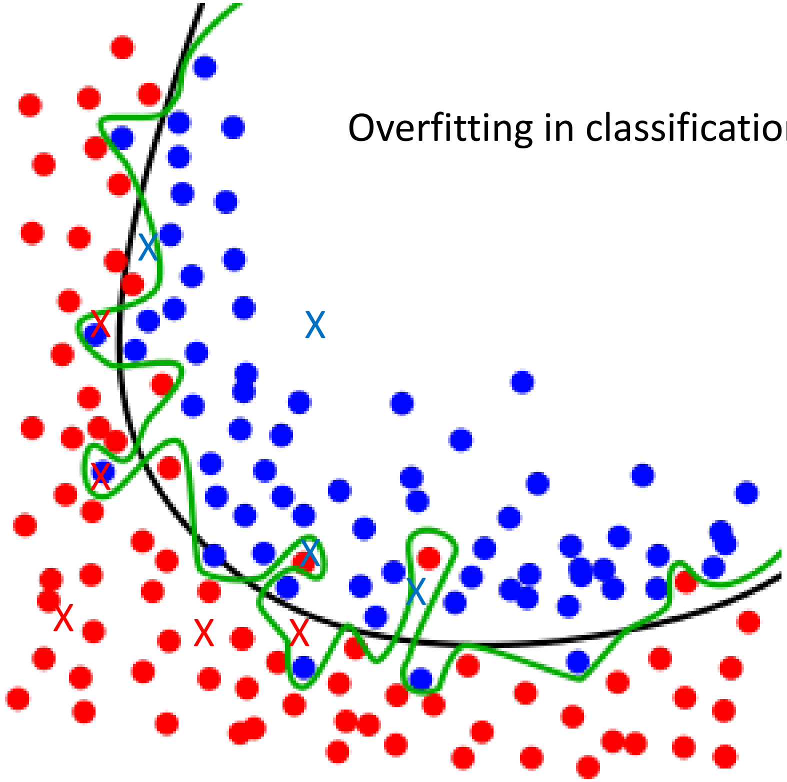
Overfitting in classification

Generalization AND overfitting
WITH NEW TESTING DATA

Overfitting in regression

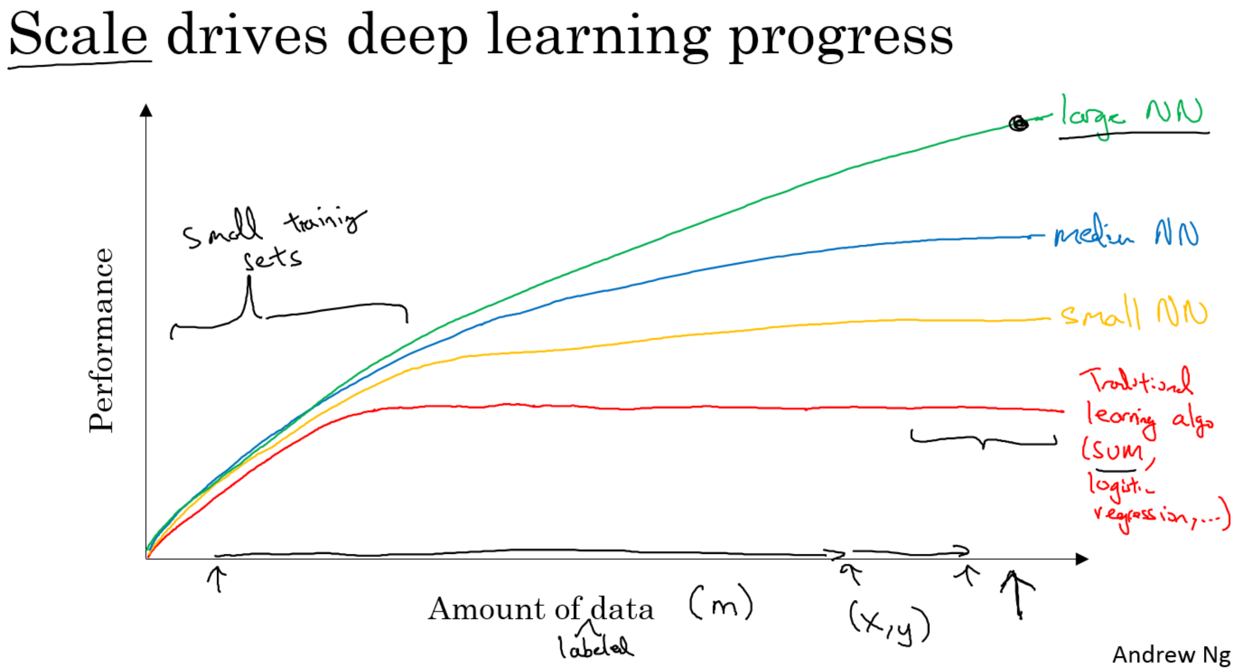


Overfitting in classification



So far …

- Machine learning = AI
- Goal: general function for input data
- Training process: Find parameters for the model
- Supervised: you have labeled data
- Unsupervised: you do not have labeled data
- Semi-supervised: some of your data is labeled
- Overfitting: training adjust very well to your training data, but do not generalize

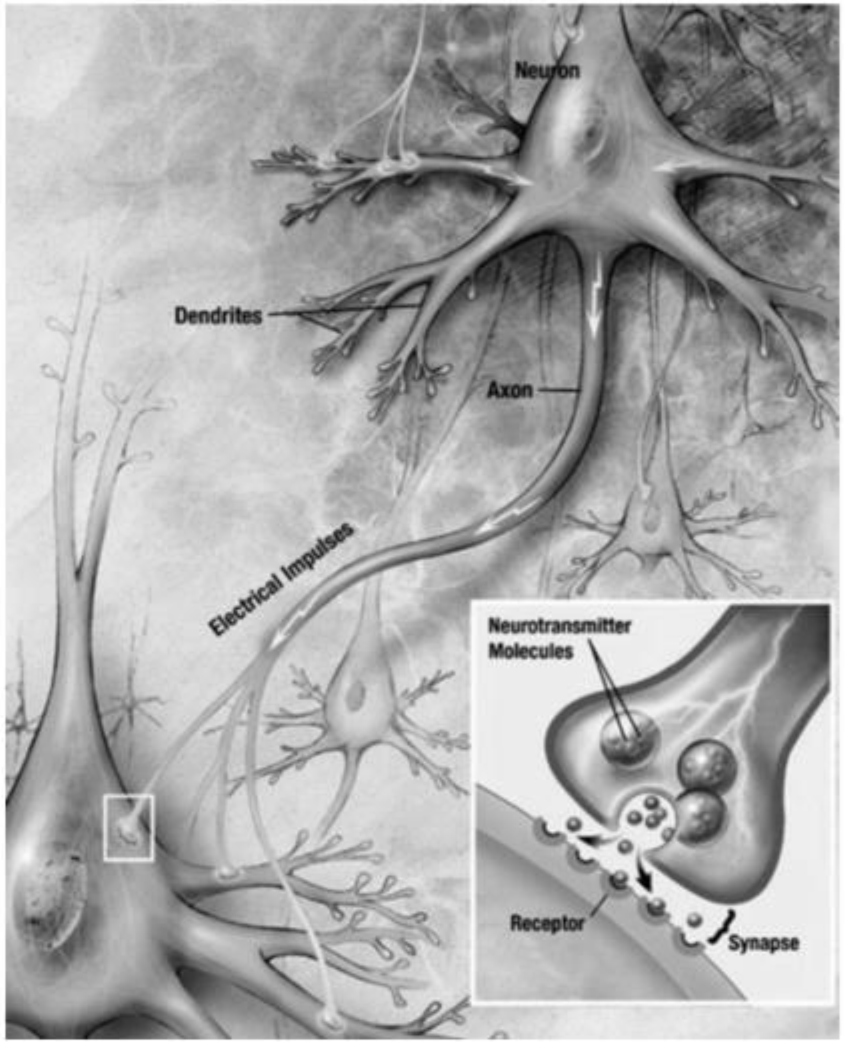


What is deep learning?

What is deep learning?

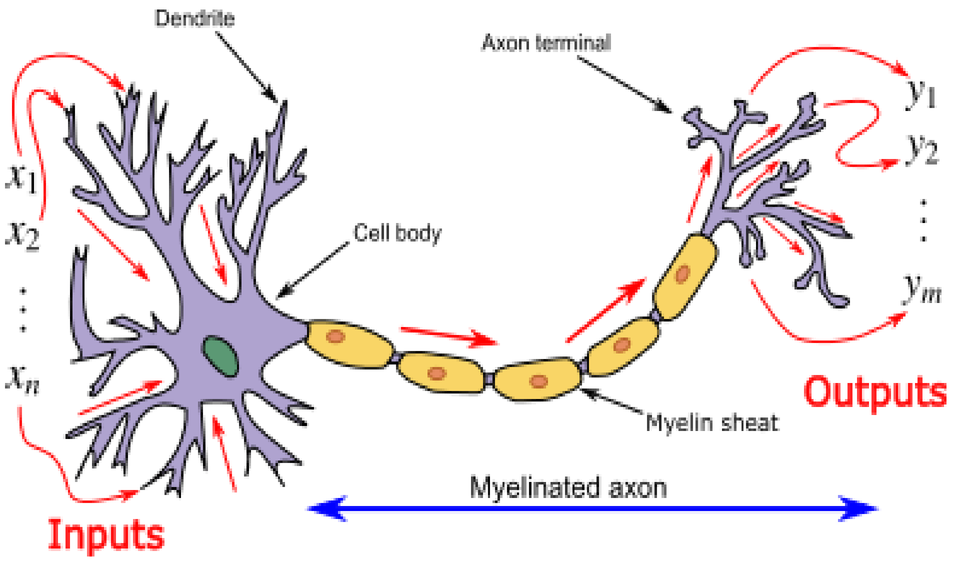
- A machine learning technique that solves problems with enormous amount of data.
  - Huge number of tunable parameters
  - Highly non-linear
  - Based on neural networks
    - A stack of neural networks layers
  - It is data driven (not hand-crafted features)

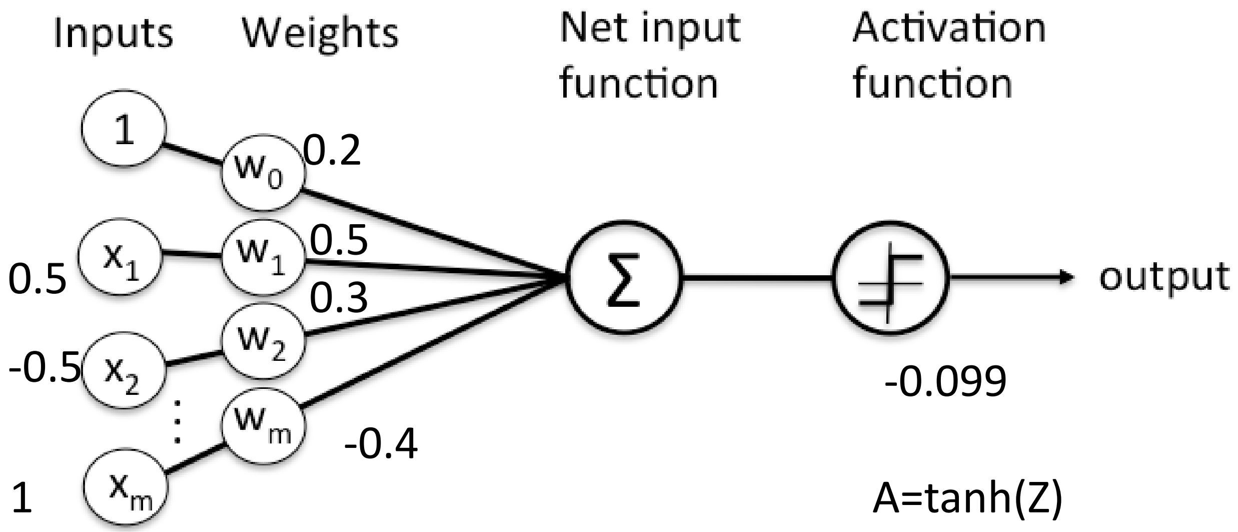
Neurons in the Brain



- Brain is composed of **neurons**
- A neuron receives input from other neurons (generally thousands) from its dendrites
- Inputs are approximately **summed**
- When the input exceeds a threshold, the neuron sends an electrical spike that travels from the body, down the axon, to the next neuron(s)

What is a neuron?



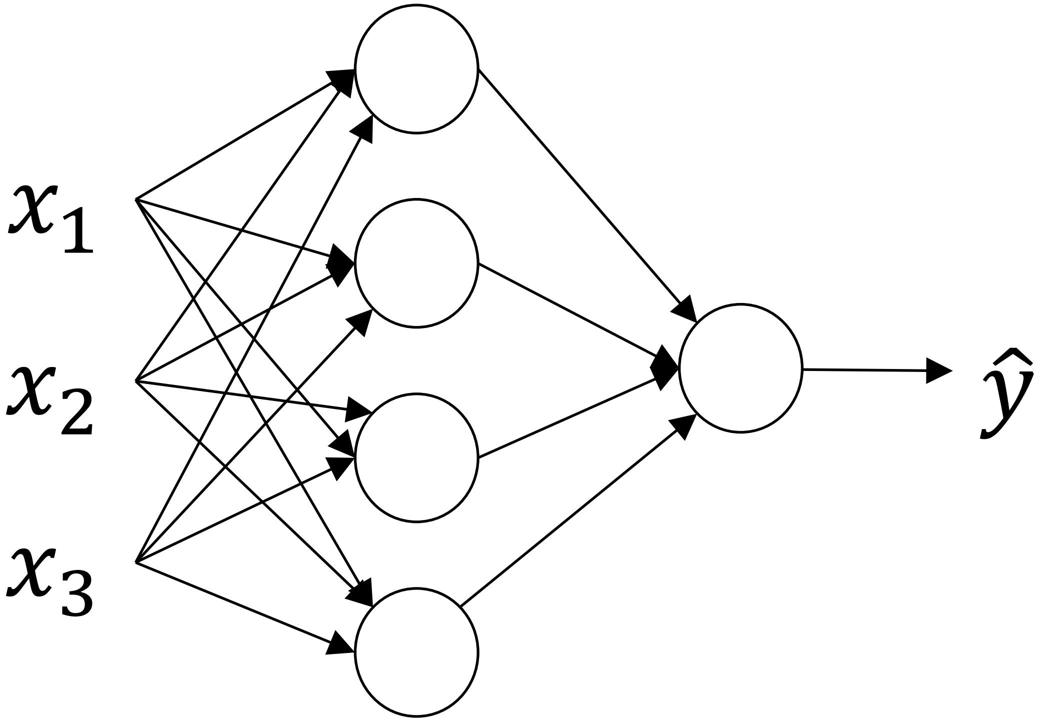


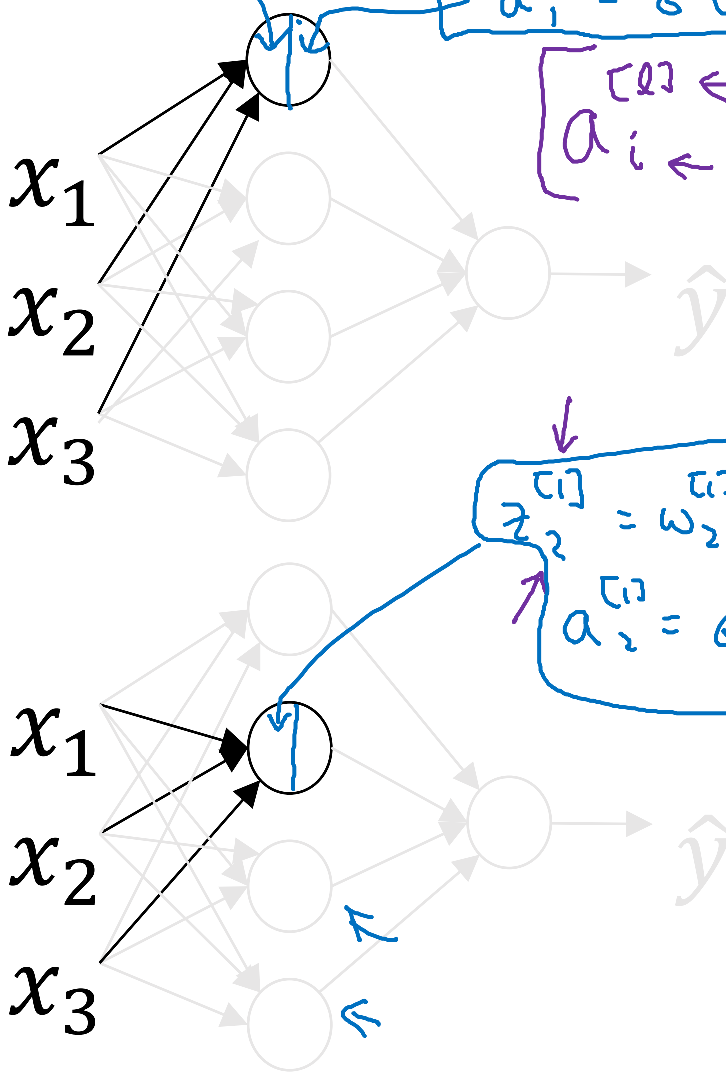
$$z = w^T x$$

$$a = \sigma(z)$$

$$a = \hat{y}$$

What is a neural network?





$z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]}$

$a_1^{[1]} = \sigma(z_1^{[1]})$

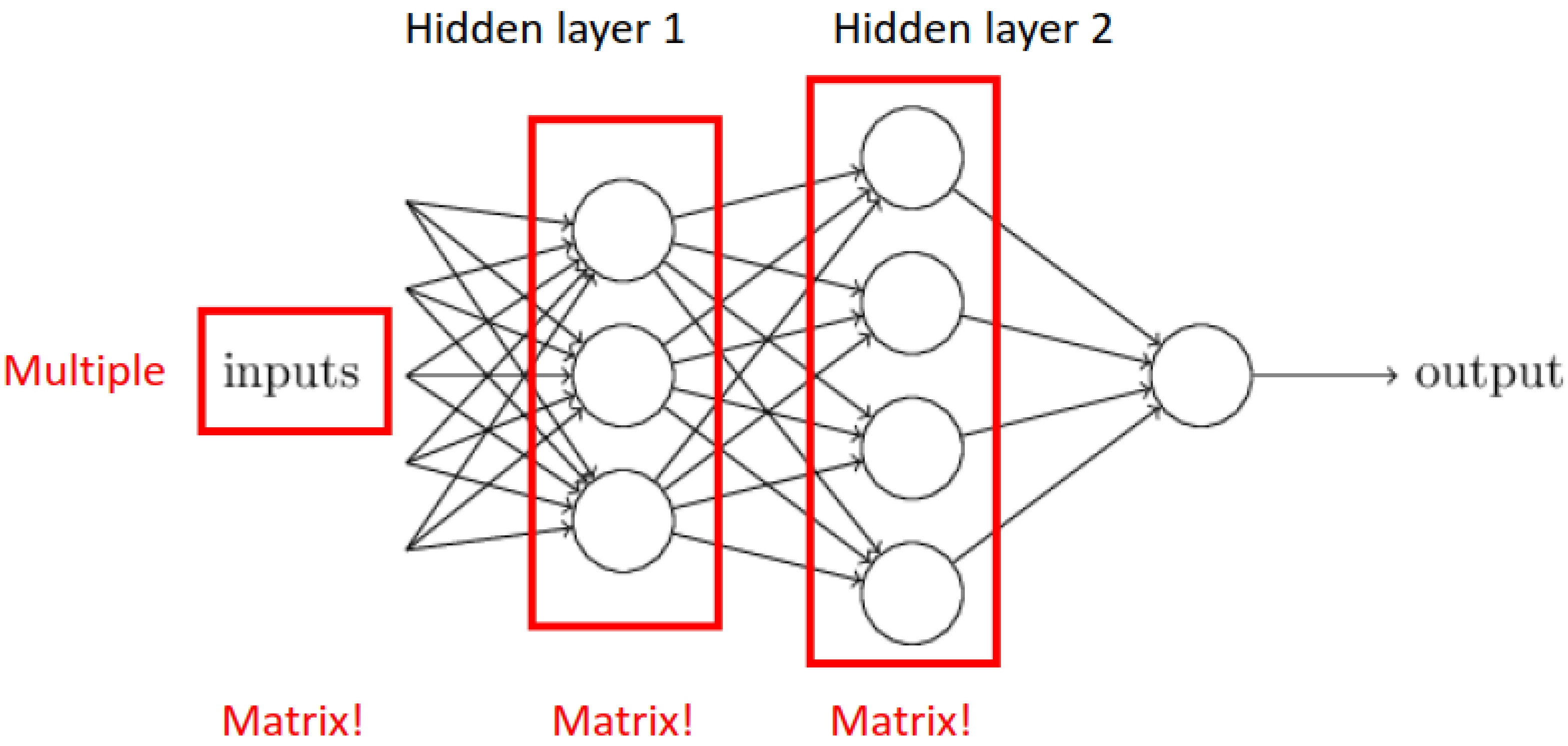
$a^{[l]}$ ← layer

a_i ← node in layer.

$z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]}$

$a_2^{[1]} = \sigma(z_2^{[1]})$

Composition

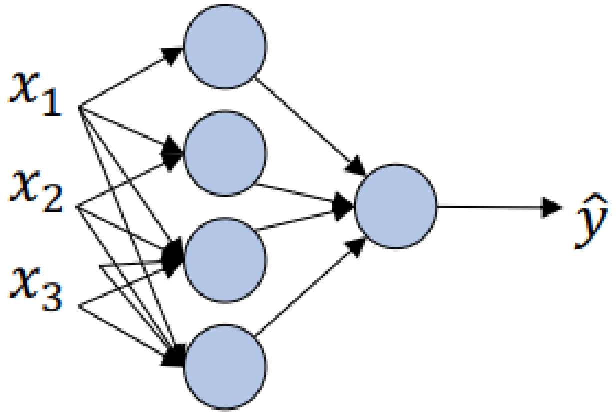


It's all just matrix multiplication!

GPUs -> special hardware for fast/large matrix multiplication.

Composition: activation function

- Activation function must be a non-linear function.
  - Other case the output will be a linear function
    - Image you have 2 layers



$$\mathbf{z}^{[1]} = \mathbf{W}^{[1]}\,\mathbf{x} + \mathbf{b}^{[1]}$$

$$\mathbf{z}^{[2]} = \mathbf{W}^{[2]}\,\mathbf{z}^{[1]} + \mathbf{b}^{[2]}$$

$$\begin{aligned}\mathbf{z}^{[2]} &= \mathbf{W}^{[2]}\,\mathbf{z}^{[1]} + \mathbf{b}^{[2]} \\ &= \mathbf{W}^{[2]}\,[\mathbf{W}^{[1]}\,\mathbf{x} + \mathbf{b}^{[1]}] + \mathbf{b}^{[2]} \\ &= \mathbf{W}^{[2]}\,\mathbf{W}^{[1]}\,\mathbf{x} + \mathbf{W}^{[2]}\,\mathbf{b}^{[1]} + \mathbf{b}^{[2]} \\ &= \mathbf{W}\,\mathbf{x} + \mathbf{b}\end{aligned}$$

$$\hat{y} = \mathbf{z}^{[2]} = \mathbf{W}\,\mathbf{x} + \mathbf{b}$$

The output is always a linear function of the input!

Problem 1 with all linear functions

- We have formed chains of linear functions.
- We know that linear functions can be reduced
  - g = f(h(x))

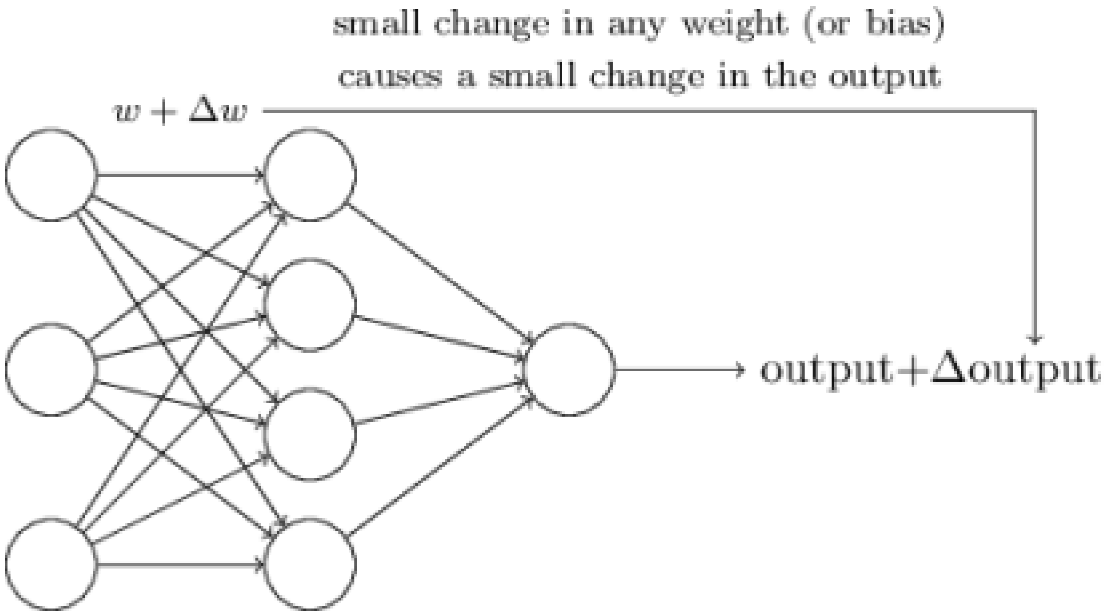
Our composition of functions is really
just a single function : (

Problem 2 with all linear functions

Linear classifiers:

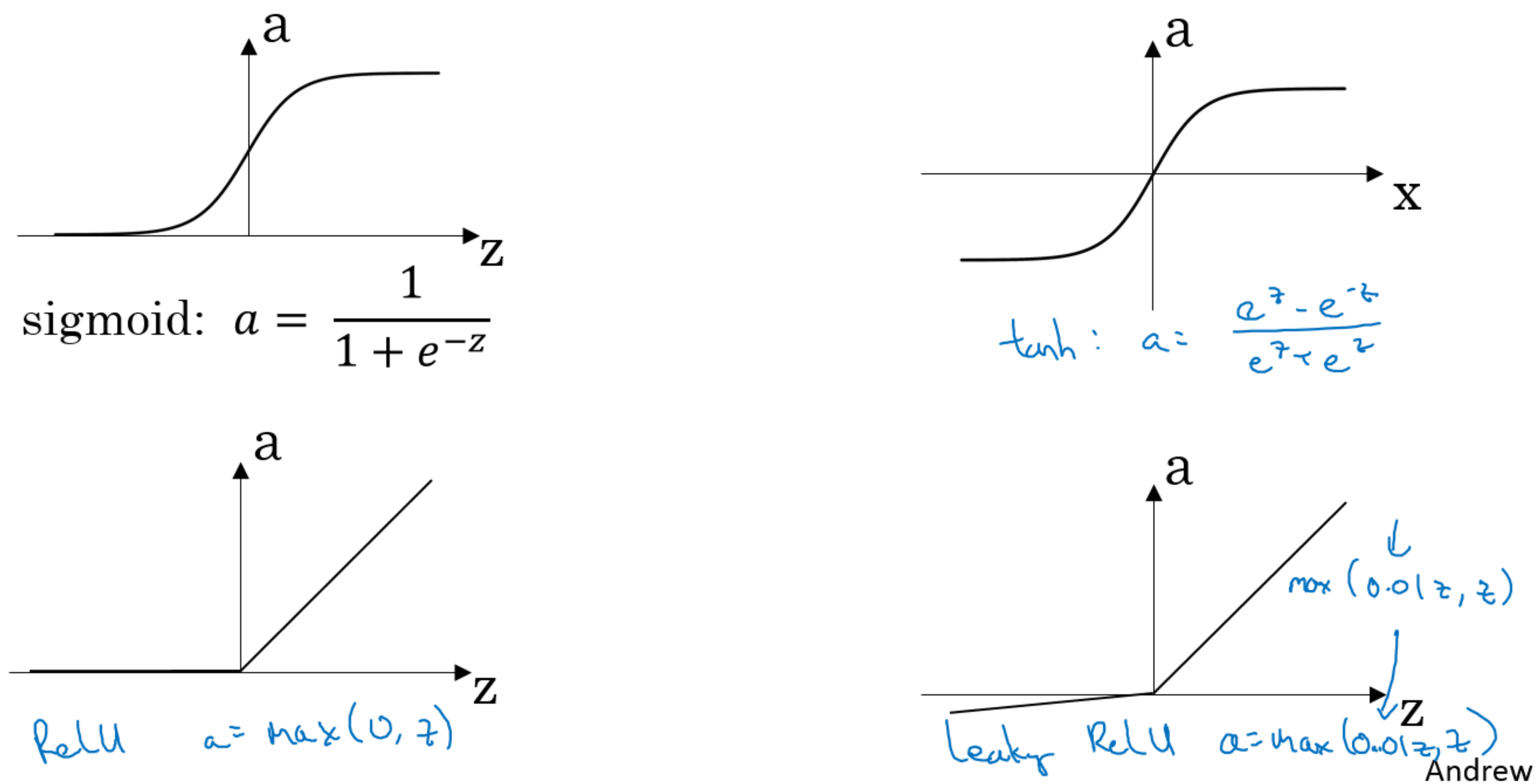
small change in input can cause large change in binary output.

We want:

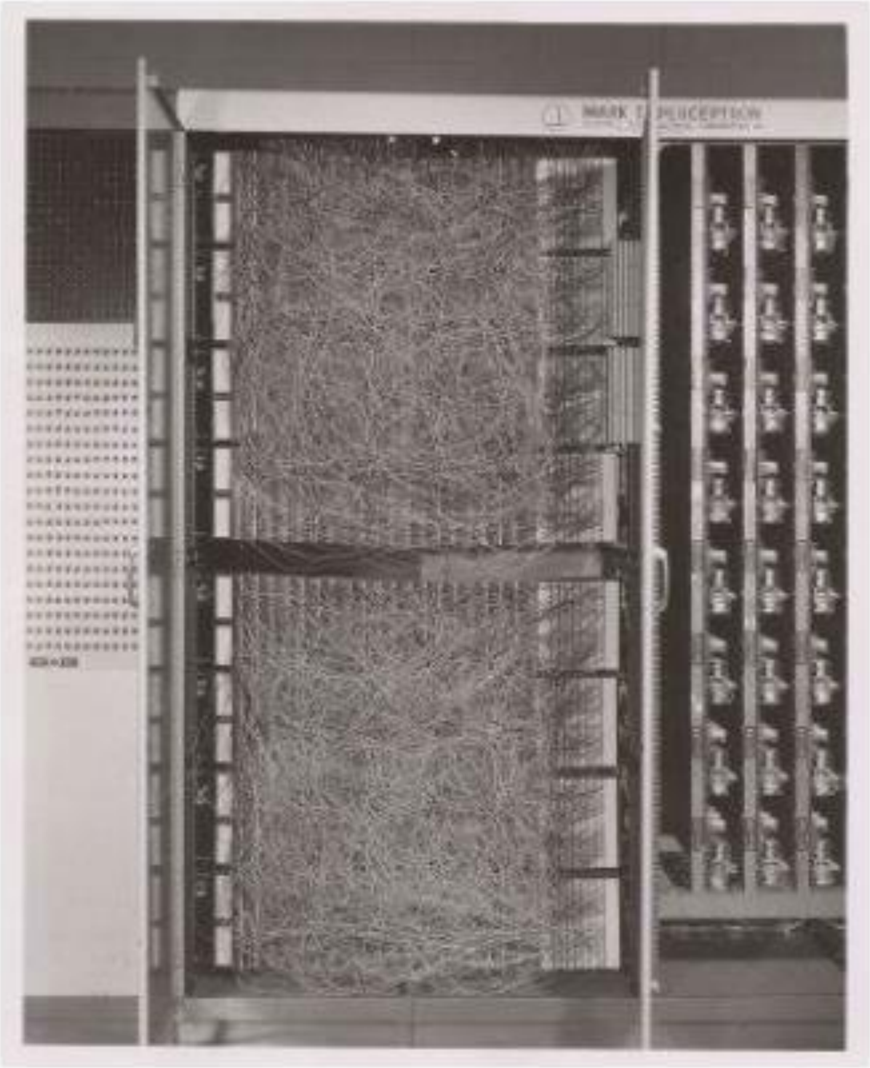


Activation function

Pros and cons of activation functions



Andrew Ng



Mark 1 Perceptron
c.1960

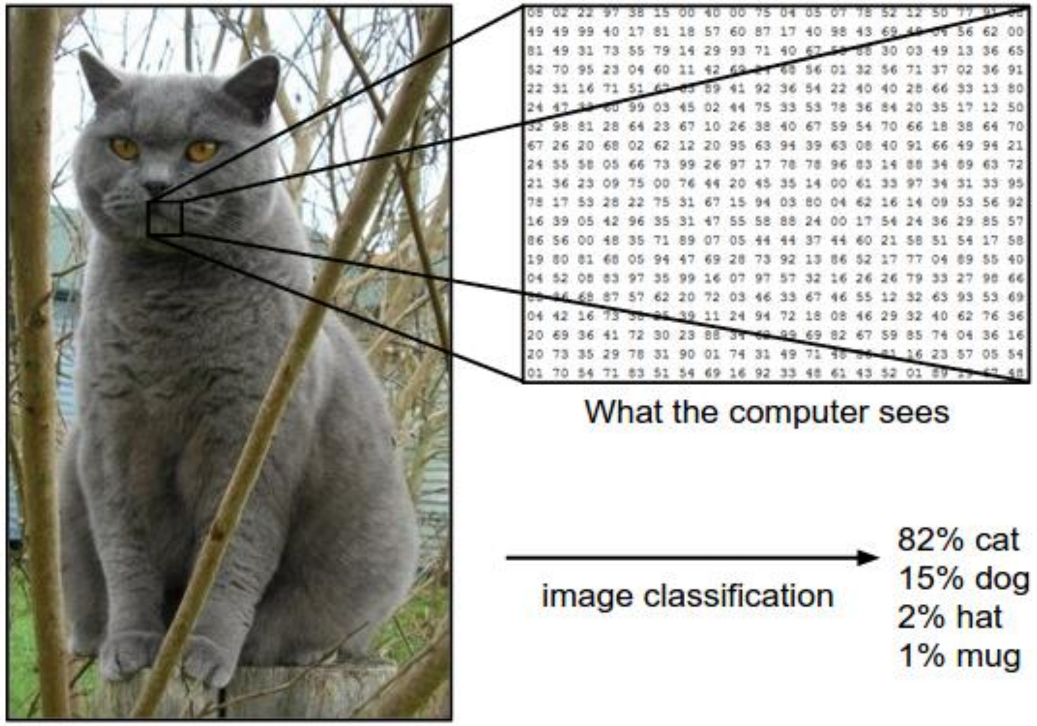
20x20 pixel
camera feed

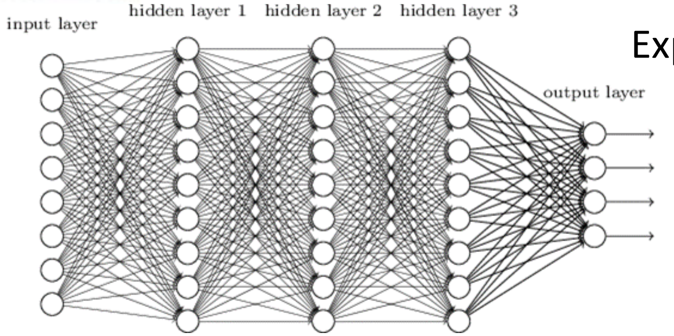
Loss function

- Error: Difference between expected value and obtained value
- Example: Image classification
- Loss: sum errors in the training dataset

$$J_1 = \frac{1}{m} \sum_{train} | \hat{y_i} - y_i |$$

$$J_2 = \frac{1}{m} \sum_{train} (\hat{y_i} - y_i)^2$$



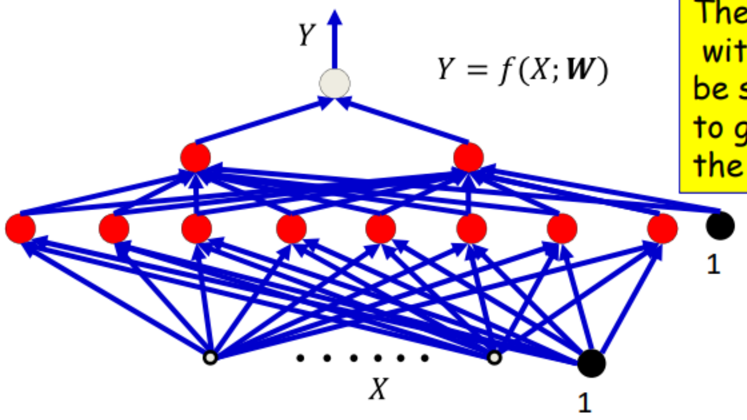


Expected Output

1
0
0
0

What are you optimizing?

What we learn: The parameters of the network



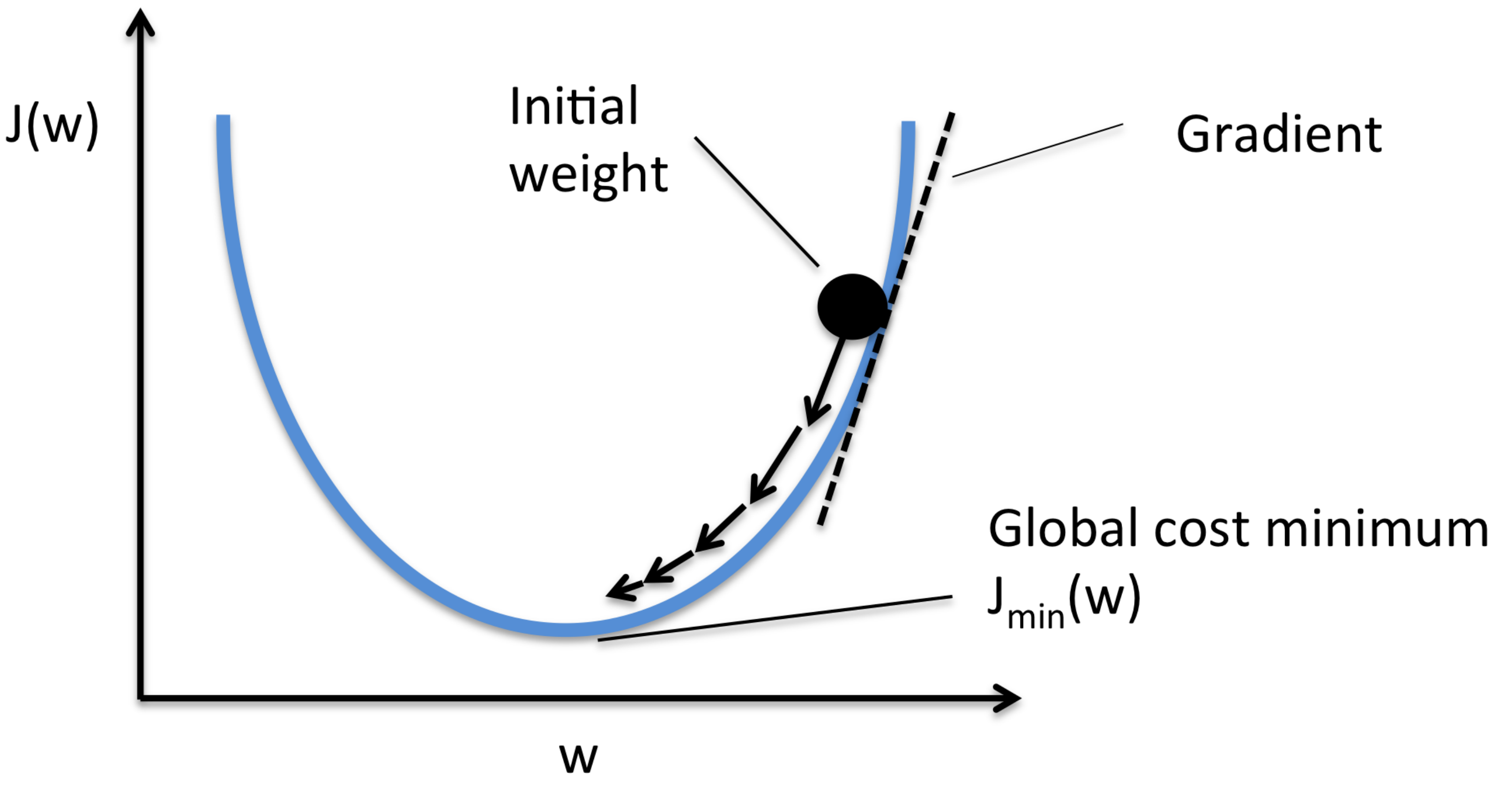
The network is a function f() with parameters W which must be set to the appropriate values to get the desired behavior from the net

- **Given:** the architecture of the network
- The parameters of the network: The weights and biases
  - The weights associated with the blue arrows in the picture
- *Learning the network :* Determining the values of these parameters such that the network computes the desired function

13

- Goal: Minimize the loss function !!

How to minimize a function ?

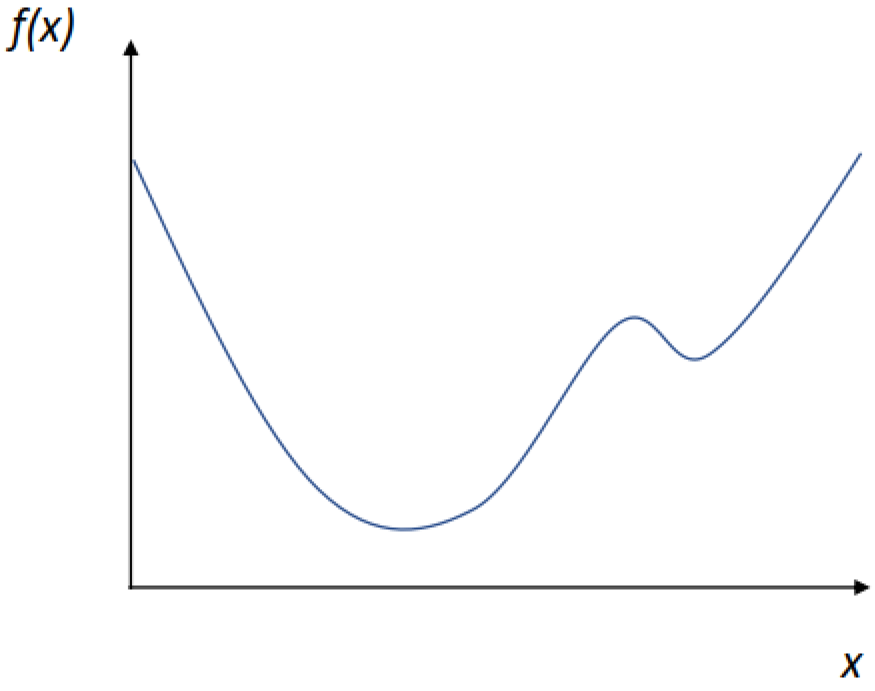


Repeat until there is almost not change

$$w_{new} = w_{prev} - \alpha \frac{dJ}{dW}$$

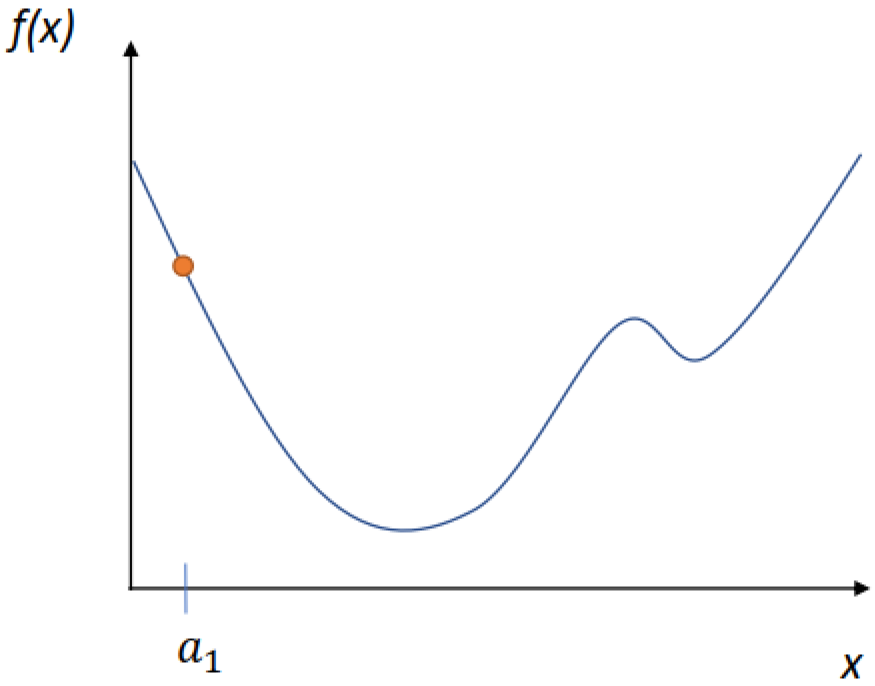
HOW TO COMPUTE THIS GRADIENT?

Gradient descent



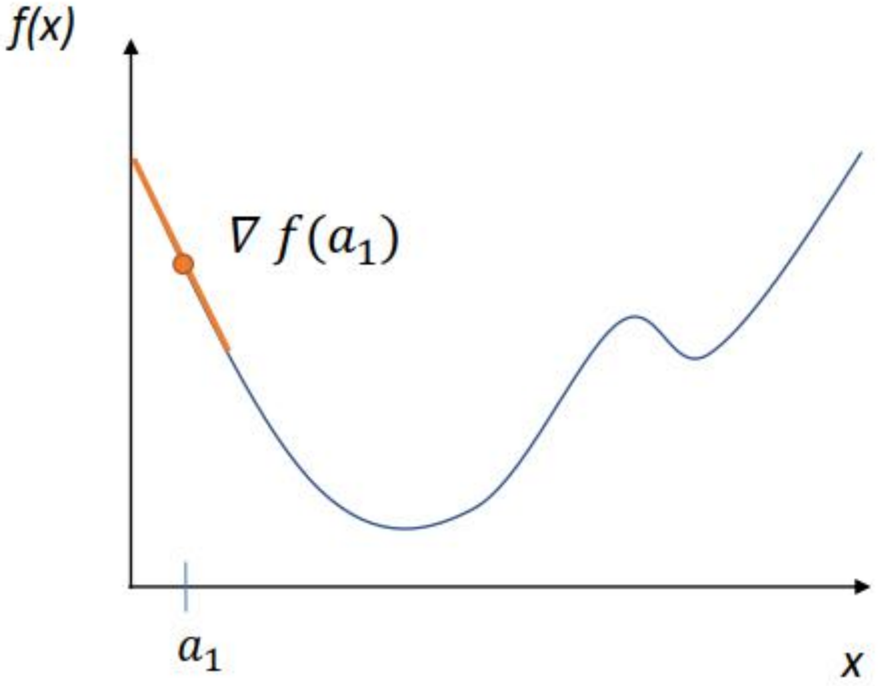
General approach

Pick random starting point.



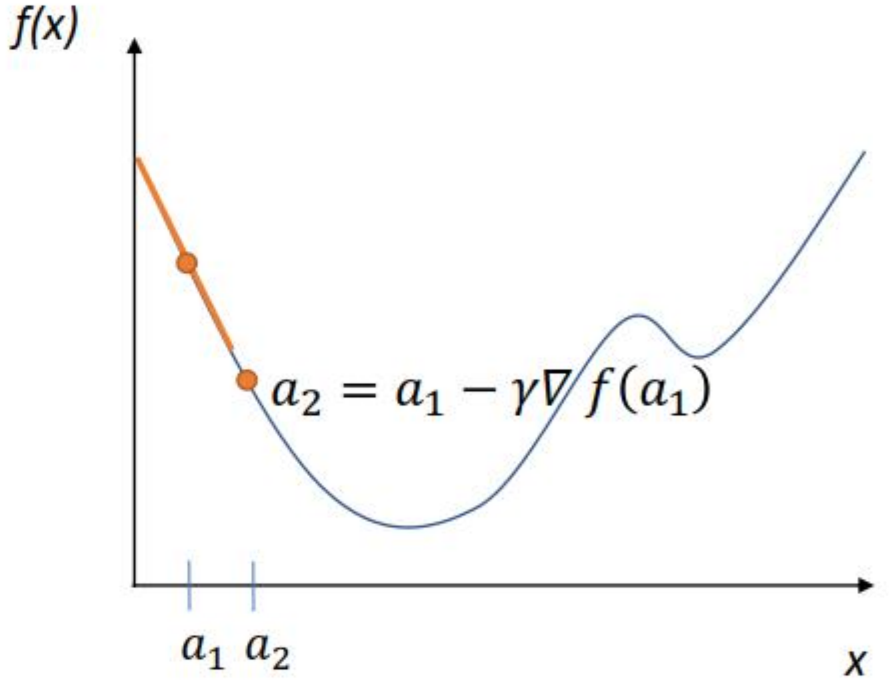
General approach

Compute gradient at point (analytically or by finite differences)



General approach

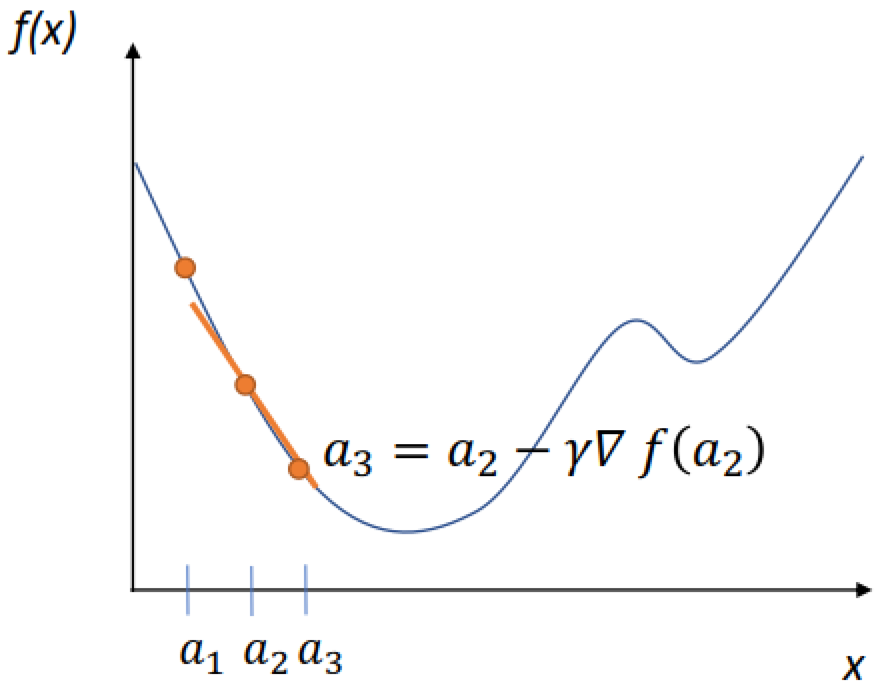
Move along parameter space in direction of negative gradient



γ = amount to move
= *learning rate*

General approach

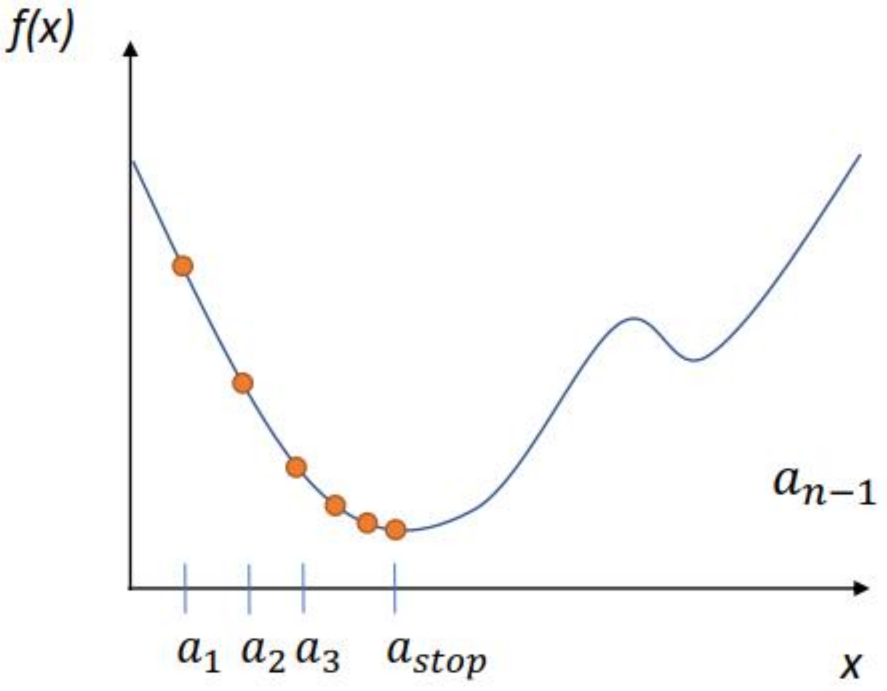
Move along parameter space in direction of negative gradient.



γ = amount to move
= *learning rate*

General approach

Stop when we don't move any more.

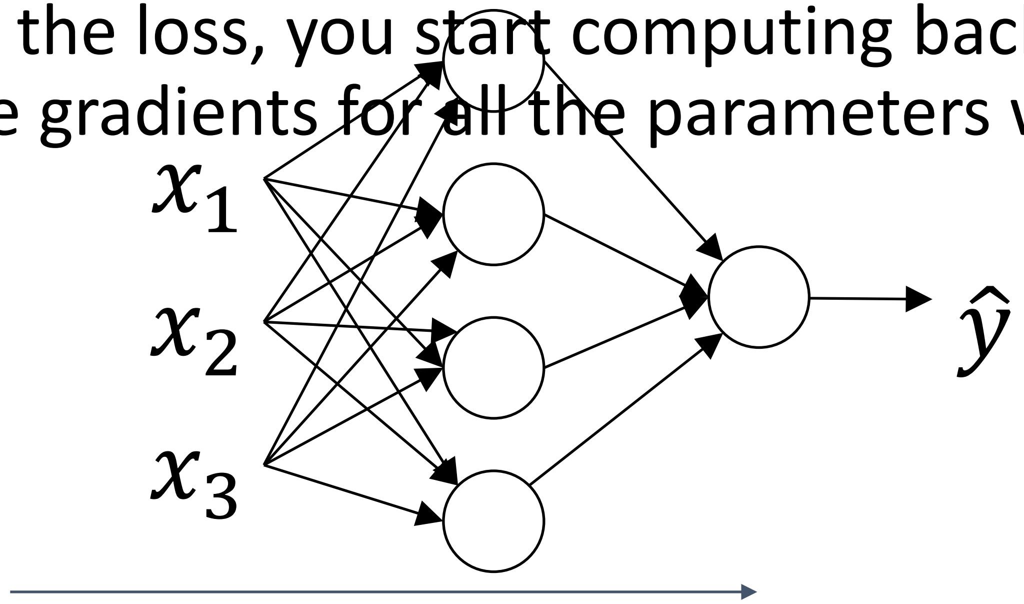


a_{stop}:

$$a_{n-1} - \gamma \nabla f(a_{n-1}) = 0$$

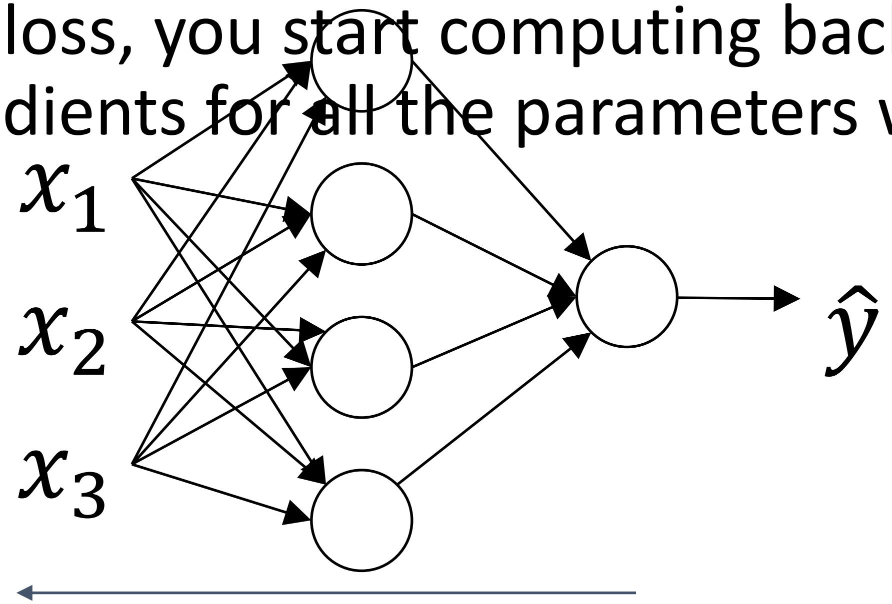
Back-propagation

- It is a technique to compute the gradient
- Gradients are necessary to get closer to the solution
- FORWARD PASS: You take the inputs, compute the outputs and loss(saving intermedia results)
- From the loss, you start computing backwards to estimate the values of the gradients for all the parameters w



Back-propagation

- It is a technique to compute the gradient
- Gradients are necessary to get closer to the solution
- FORWARD PASS: You take the inputs, compute the outputs and loss(saving intermedia results)
- From the loss, you start computing backwards to estimate the values of the gradients for all the parameters w

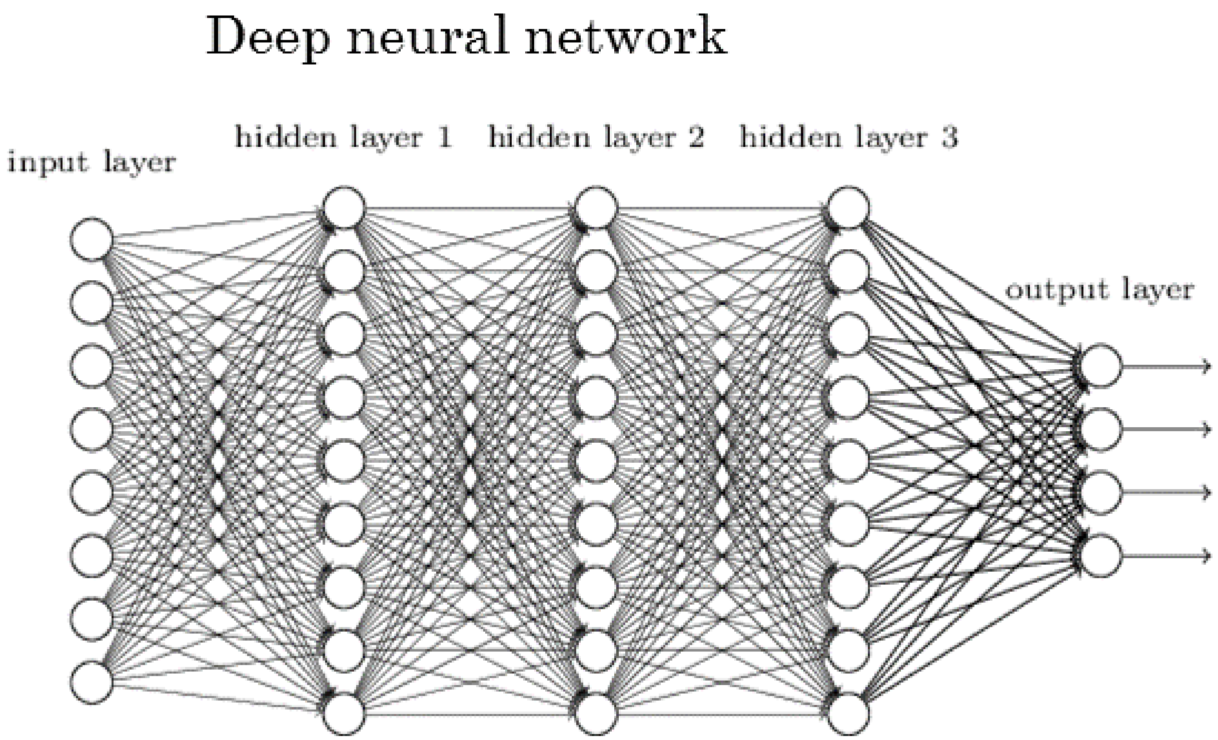


Goal: find

From Layer 2: $\frac{dJ}{dw^{[2]}_{11}}, \frac{dJ}{dw^{[2]}_{12}}, \frac{dJ}{dw^{[2]}_{13}}, \frac{dJ}{dw^{[2]}_{14}}$

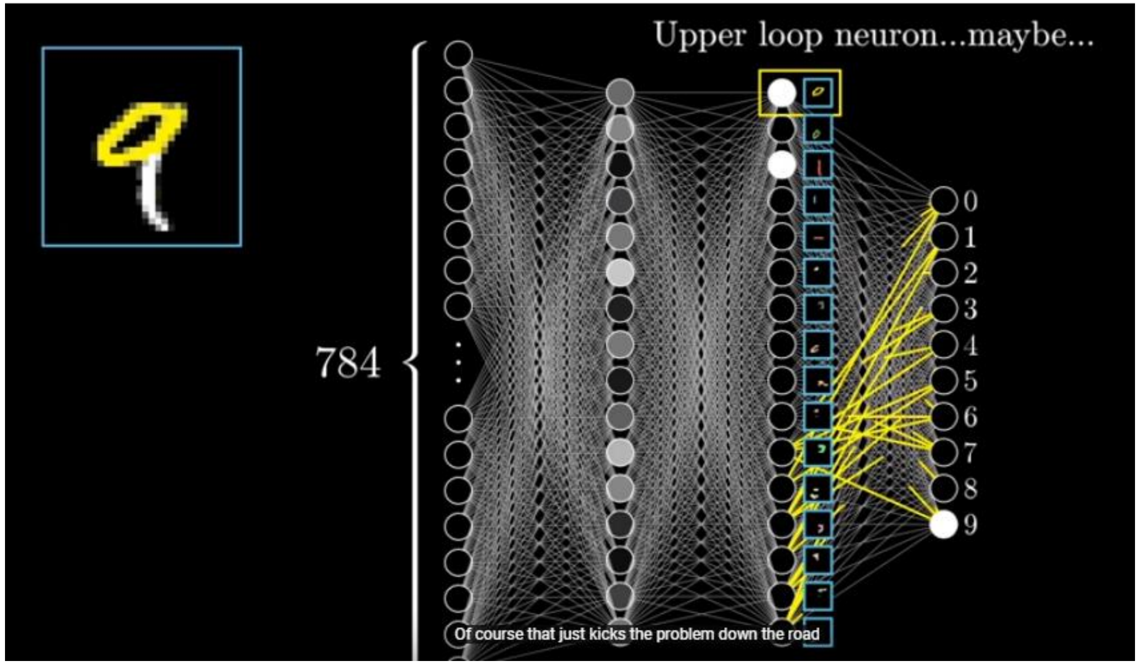
From Layer 1: $\frac{dJ}{dw^{[1]}_{11}}, \frac{dJ}{dw^{[1]}_{12}}, \ldots, \frac{dJ}{dw^{[1]}_{33}}, \frac{dJ}{dw^{[1]}_{34}}$

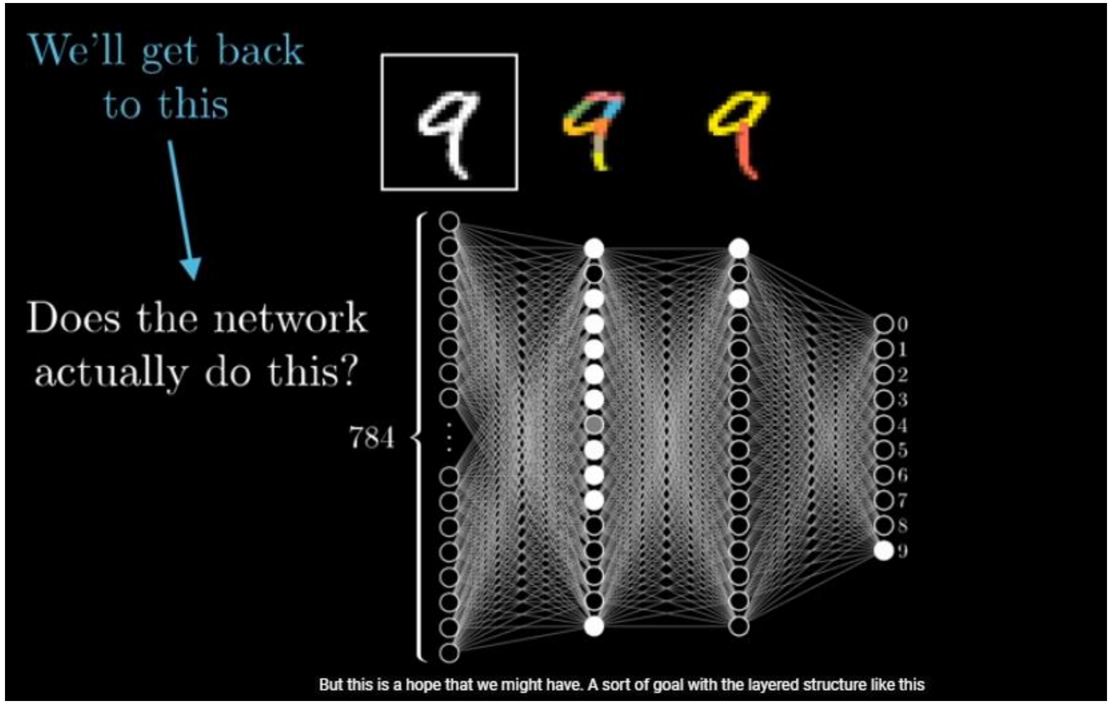
What is a deep network?



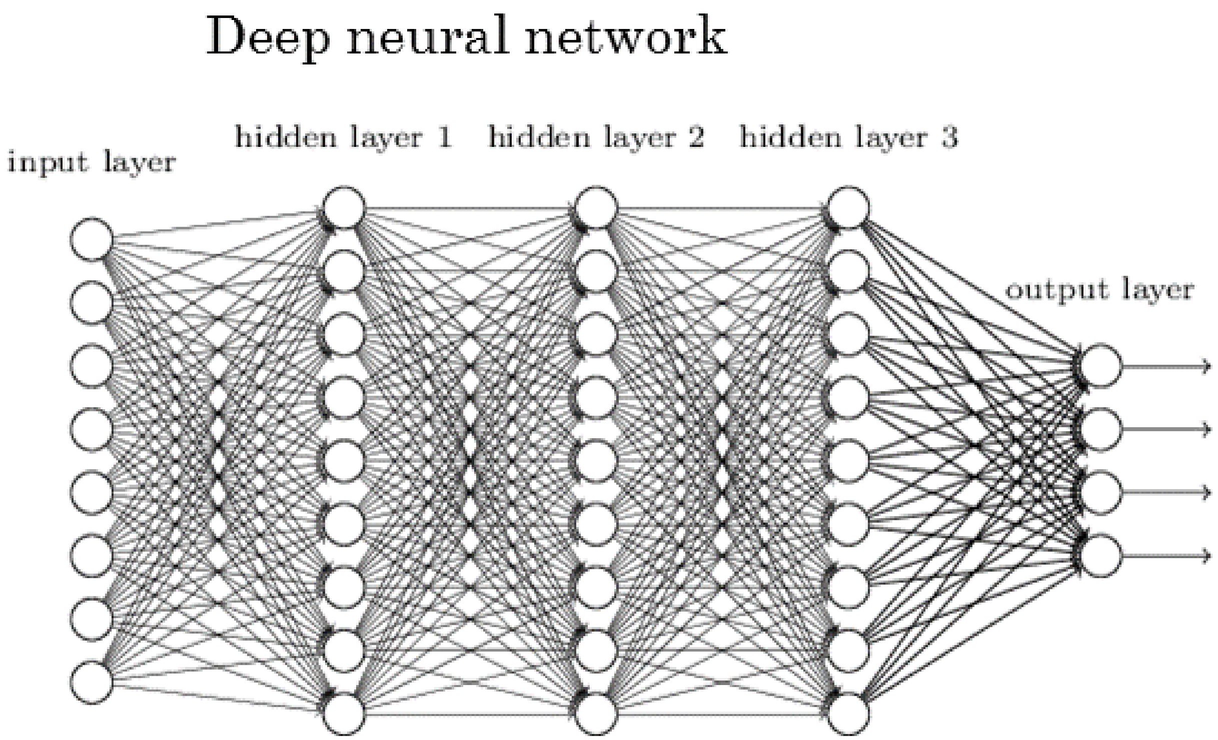
- A neural network with many layers
- Highly nonlinear

An example

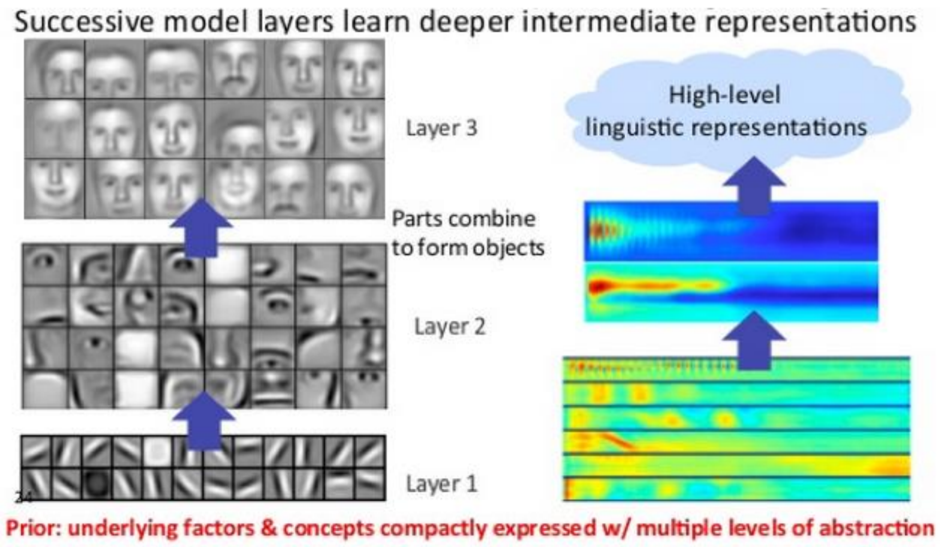

Upper loop neuron...maybe...
784
0
1
2
3
4
5
6
7
8
9
Of course that just kicks the problem down the road


We'll get back to this
Does the network actually do this?
784
0
1
2
3
4
5
6
7
8
9
But this is a hope that we might have. A sort of goal with the layered structure like this

What is a deep network?



- A neural network with many layers
- Highly non linear



So far …

- A deep network is a neural network with many layers
- A neuron in a linear function followed for an activation function
- Activation function must be non-linear
- A loss function measures how close is the created function (network) from a desired output
- The “training” is the process of find parameters (‘weights’) that reduces the loss functions
- Updating the weights as $w_{new} = w_{prev} - \alpha \frac{dJ}{dW}$ reduces the loss
- An algorithm named back-propagation allows to compute $\frac{dJ}{dW}$ for all the weights of the network in 2 steps: 1 forward, 1 backward

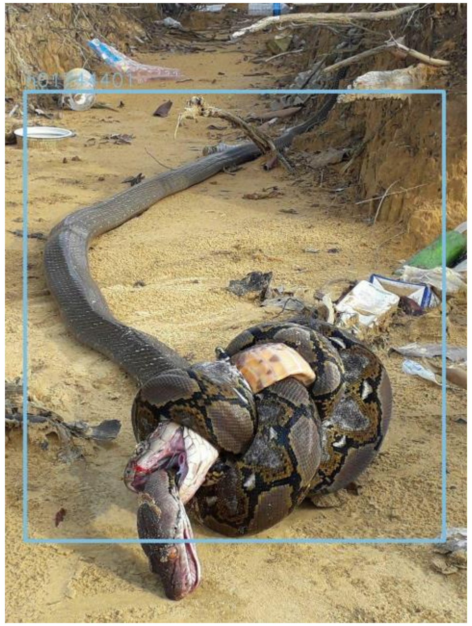
What kind of problems deep learning can solve?

What problems you can solve?

- The fundamental ones:
  - Regression: predict values
  - Classification: predict labels
- Computer vision:
  - object detection
  - semantic segmentation
  - super-resolution,
- Time series:
  - NLP
  - visual questioning/answering
- Generative models
  - impersonators ()

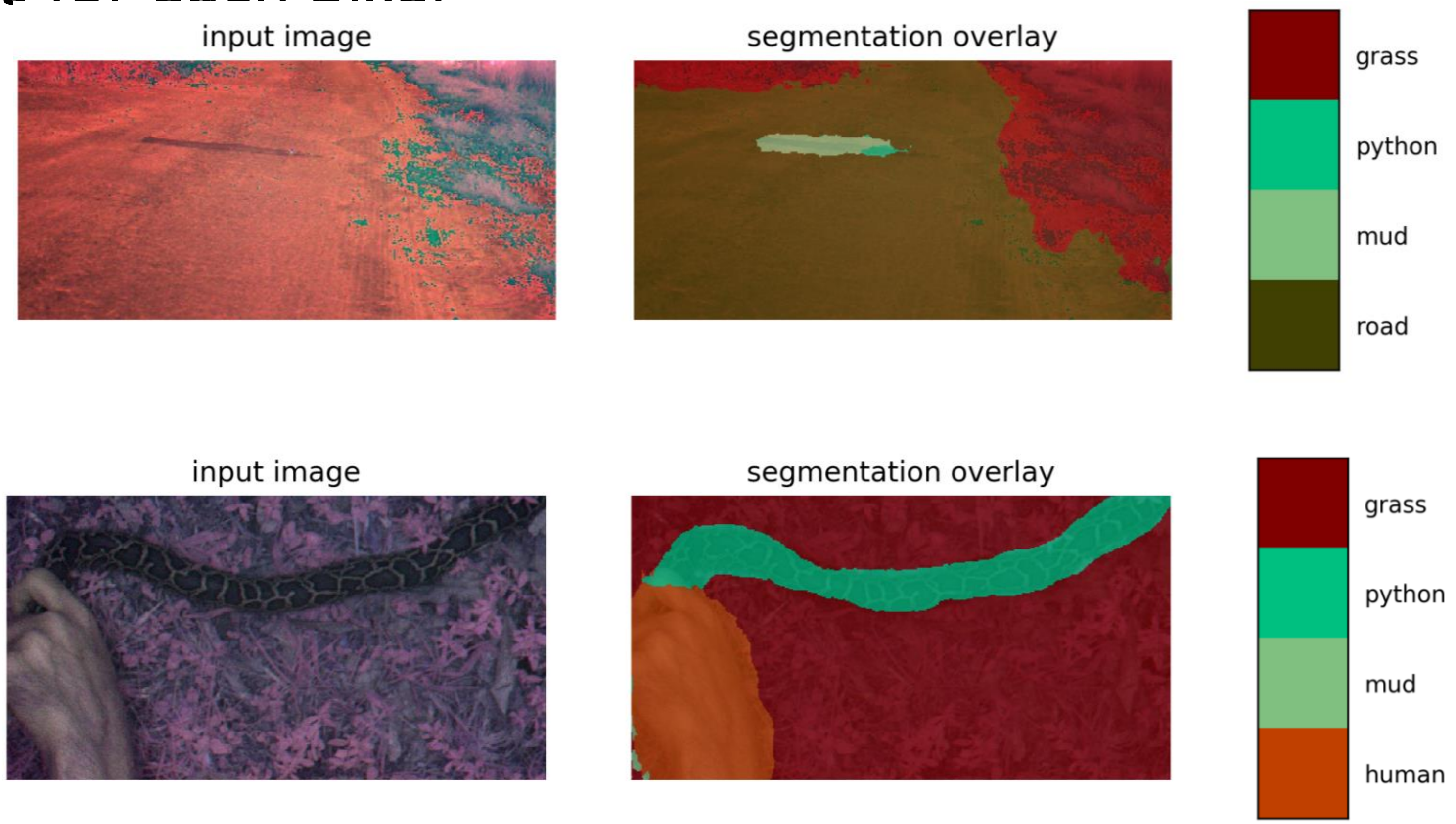
Computer vision

- Find region of interest (regression)
- Find a class label (classification)



Computer vision

- Find a class for each pixel



Computer vision
SUPER-RESOLUTION FROM A SINGLE IMAGE

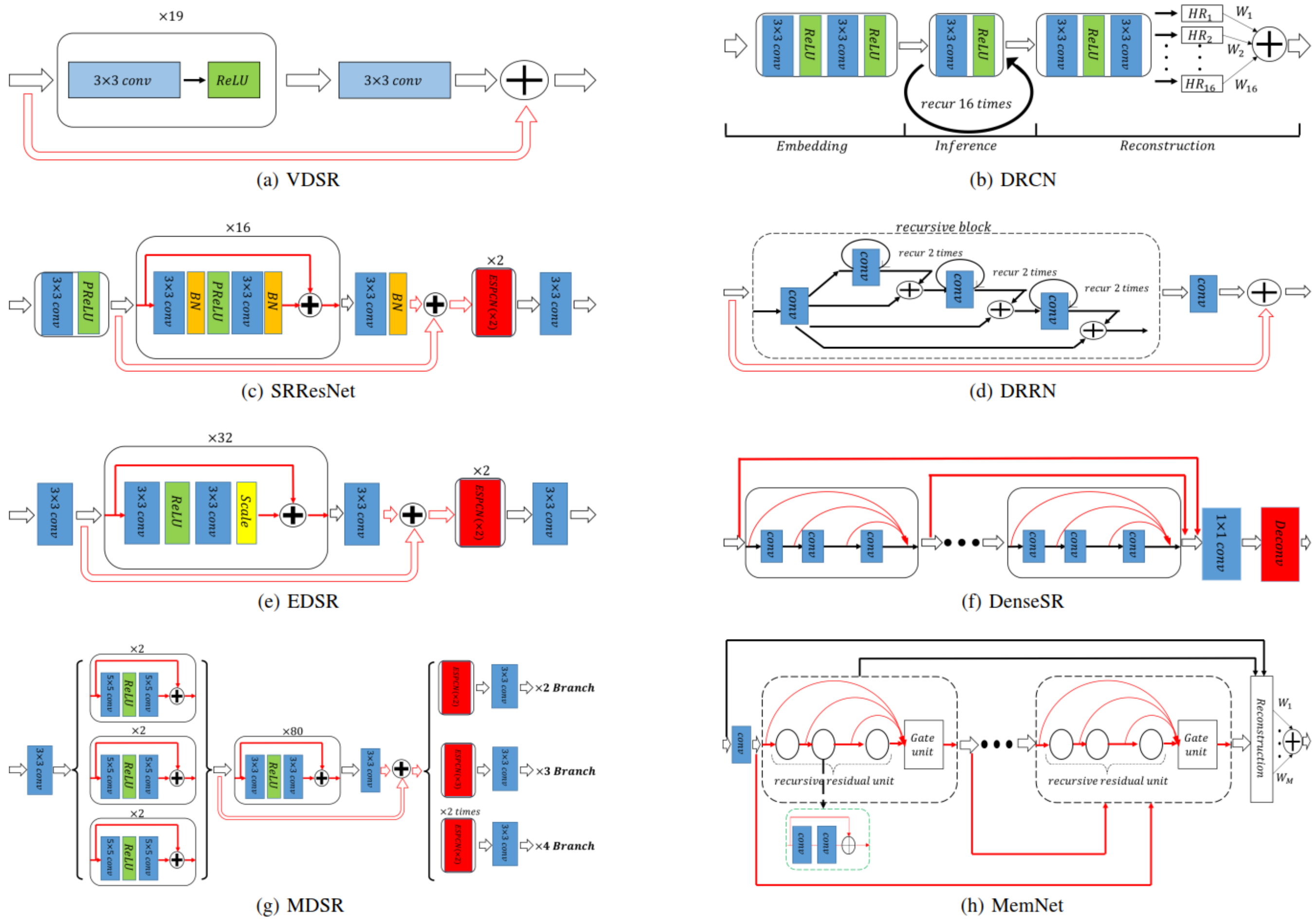


Figure 5: Sketch of several deep architectures for SISR.

(a) HR

(b) bicubic(21.59dB/0.6423)

(c) SRResNet(23.53dB/0.7832)

(d) SRGAN(21.15dB/0.6868)

(e) SRCX(20.88dB/0.6002)

Computer vision
OTHER PROBLEMS

- Super resolution from multiple images
- Denoising

Time series (RNN, LSTM, Attention models)

USE MEASUREMENT TO CHANGE STATE, USE STATE TO PREDICT FUTURE

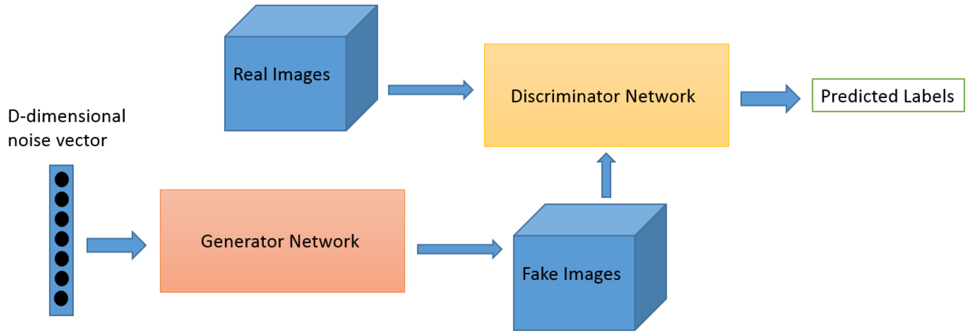
- Natural language Processing
  - Translation
  - Check Google Bert
  - Visual Questioning answer

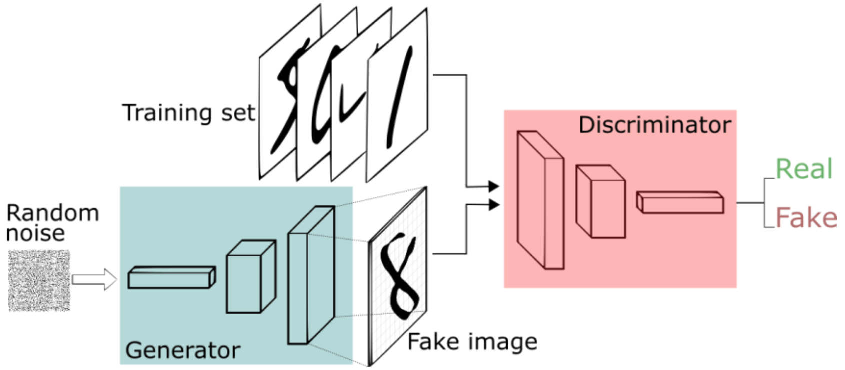
- Stocks
- Signals
  - ECG

Generative models

GAN (GENERATIVE ADVERSARIAL NETWORKS)

- Predict the data based on some loose input.
- Looks like the network is able o create somethin





Generative models

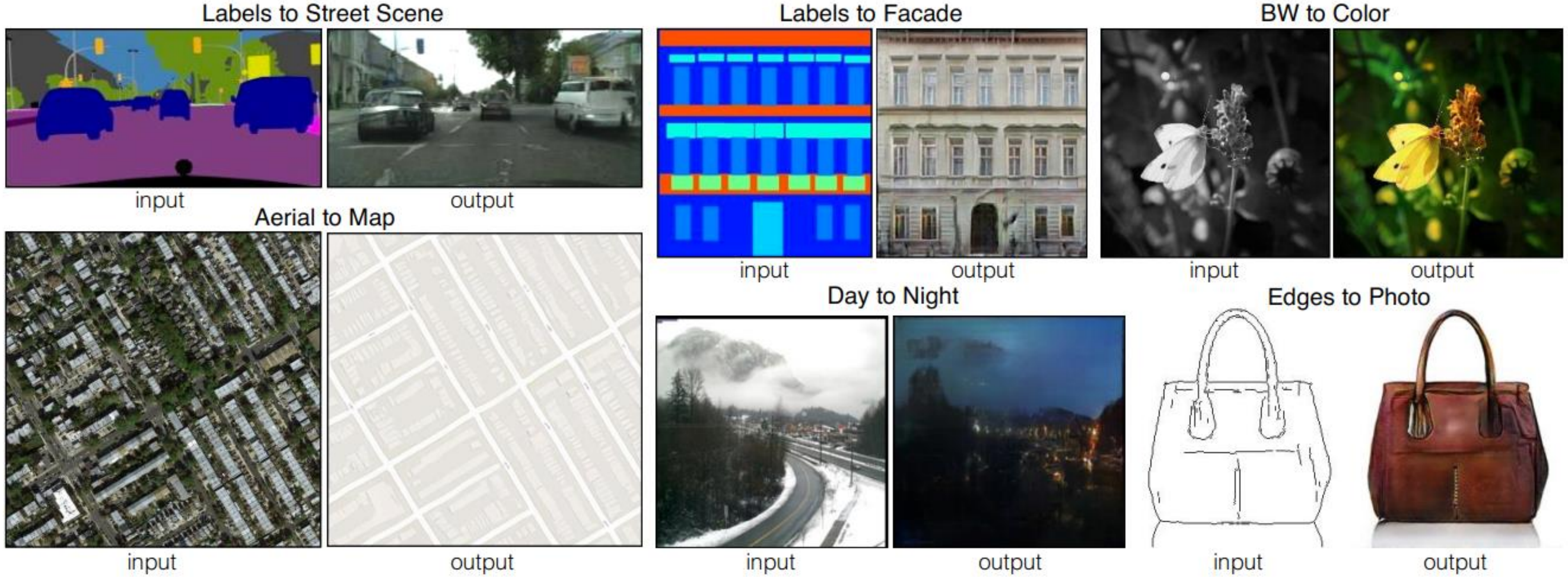


Figure 1: Many problems in image processing, graphics, and vision involve translating an input image into a corresponding output image. These problems are often treated with application-specific algorithms, even though the setting is always the same: map pixels to pixels. Conditional adversarial nets are a general-purpose solution that appears to work well on a wide variety of these problems. Here we show results of the method on several. In each case we use the same architecture and objective, and simply train on different data.

- Image-to-Image Translation with Conditional Adversarial Networks. *Phillip Isola*, *Jun-Yan Zhu*, *Tinghui Zhou*, *Alexei A. Efros*. CVPR 2017

Generative models

IMAGE CREATION FROM TEXT

- Generative Adversarial Text to Image Synthesis. *Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee.* ICML 2016

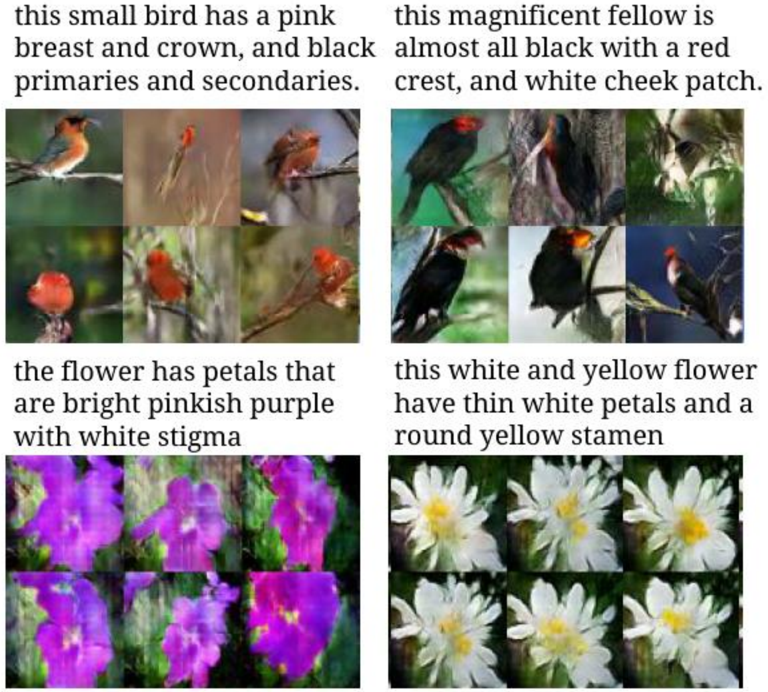


Figure 1. Examples of generated images from text descriptions. Left: captions are from zero-shot (held out) categories, unseen text. Right: captions are from the training set.

Generative models

CREATE FAKE MODELS

- https://youtu.be/p1b5aiTrGzY

NERF: NEURAL RADIANCE FIELD (3D RENDERING)

[(11) NeRF: Neural Radiance Fields - YouTube]

Questions?