Introduction to machine learning and deep learning

Luis Herranz, Computer Vision Center (Barcelona)

Organitzadors:

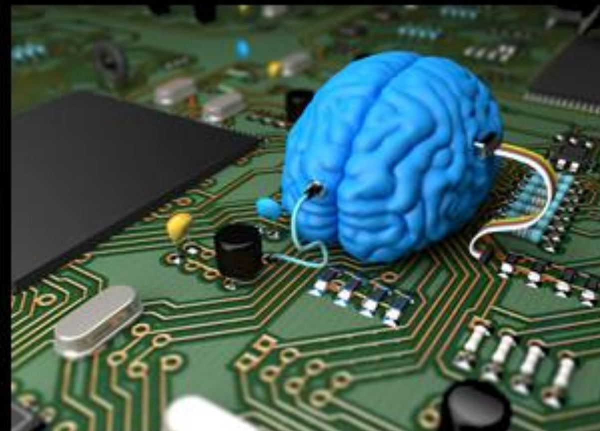
Parc de Recerca UAB · CVC · escola d'enginyeria · CORE SMART CITIES

Amb la col·laboració:

catalunya emprèn · Generalitat de Catalunya Departament d'Empresa i Ocupació · Unió Europea Fons social europeu L'FSE inverteix en el teu futur

Deep Learning

What society thinks I do

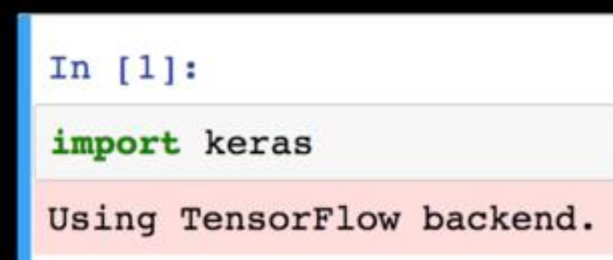


What my friends think I do

What other computer scientists think I do

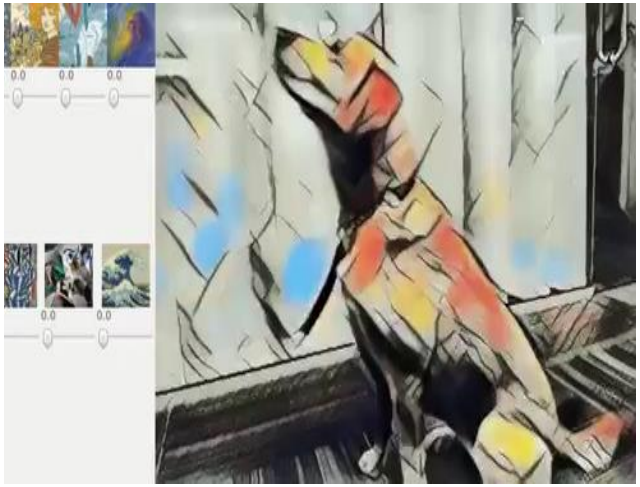
t mathematicians think I do

What I think I do



What I actually do

(Some) deep learning applications



Style transfer

Autonomous driving

Robotics

Music composition

Slide credit: David Vázquez

Today's plan

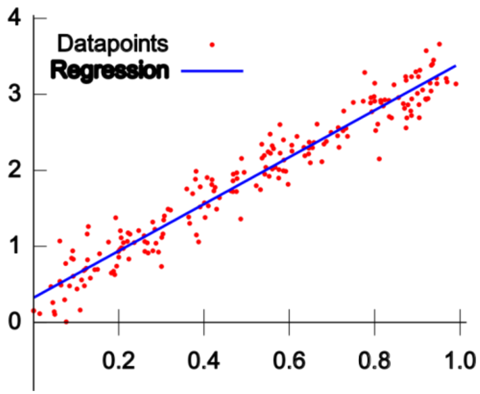
- Machine learning
- Neural networks
- Deep learning in computer vision
- Fun stuff

Today’s plan

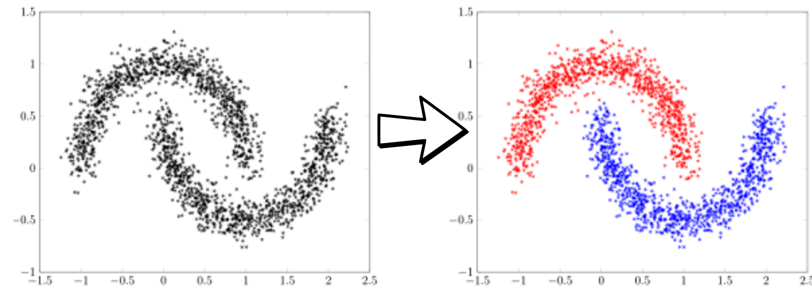
- Machine learning
  - Basic concepts
  - A toy example
- Neural networks
- Deep learning in computer vision
- Fun stuff

ML problems (according to output)

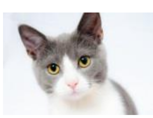
Regression
(real valued output)

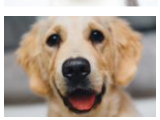


Clustering
(find groups in the data)

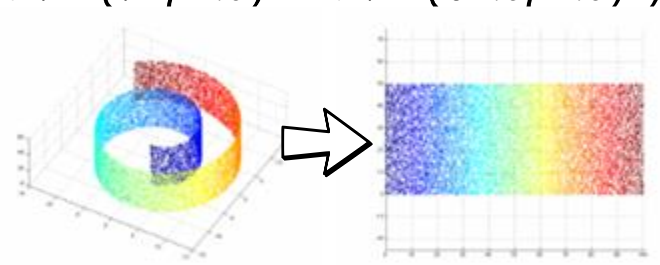


Classification
(discrete output)

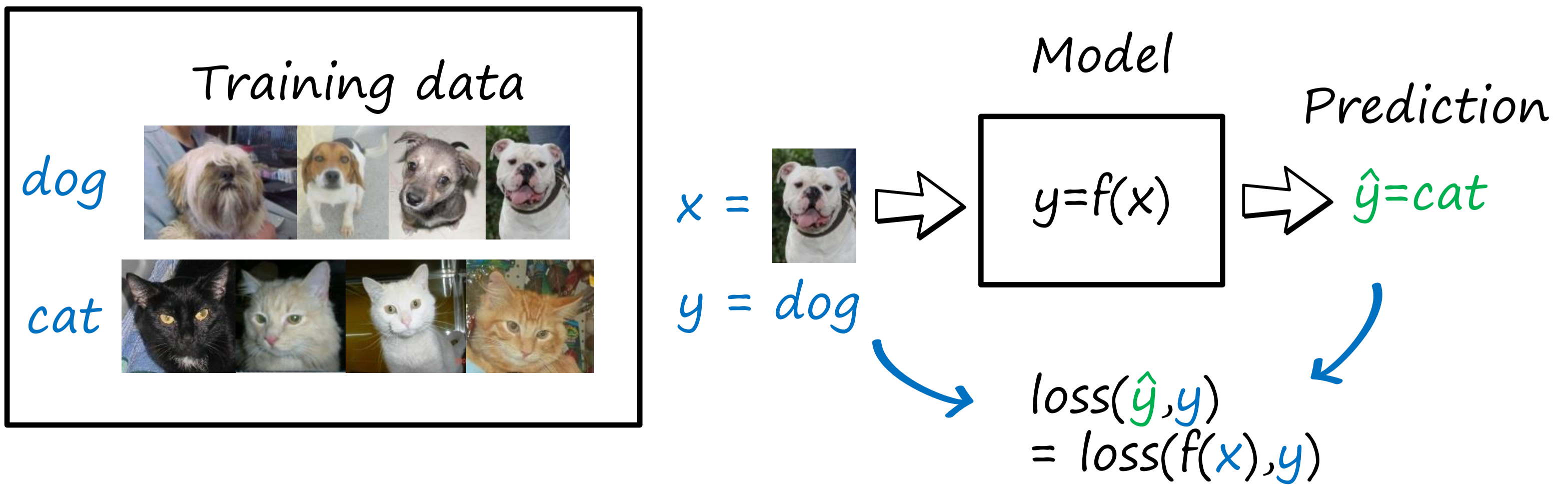

cat


dog

Dimensionality reduction
(dim(input) > dim(output))



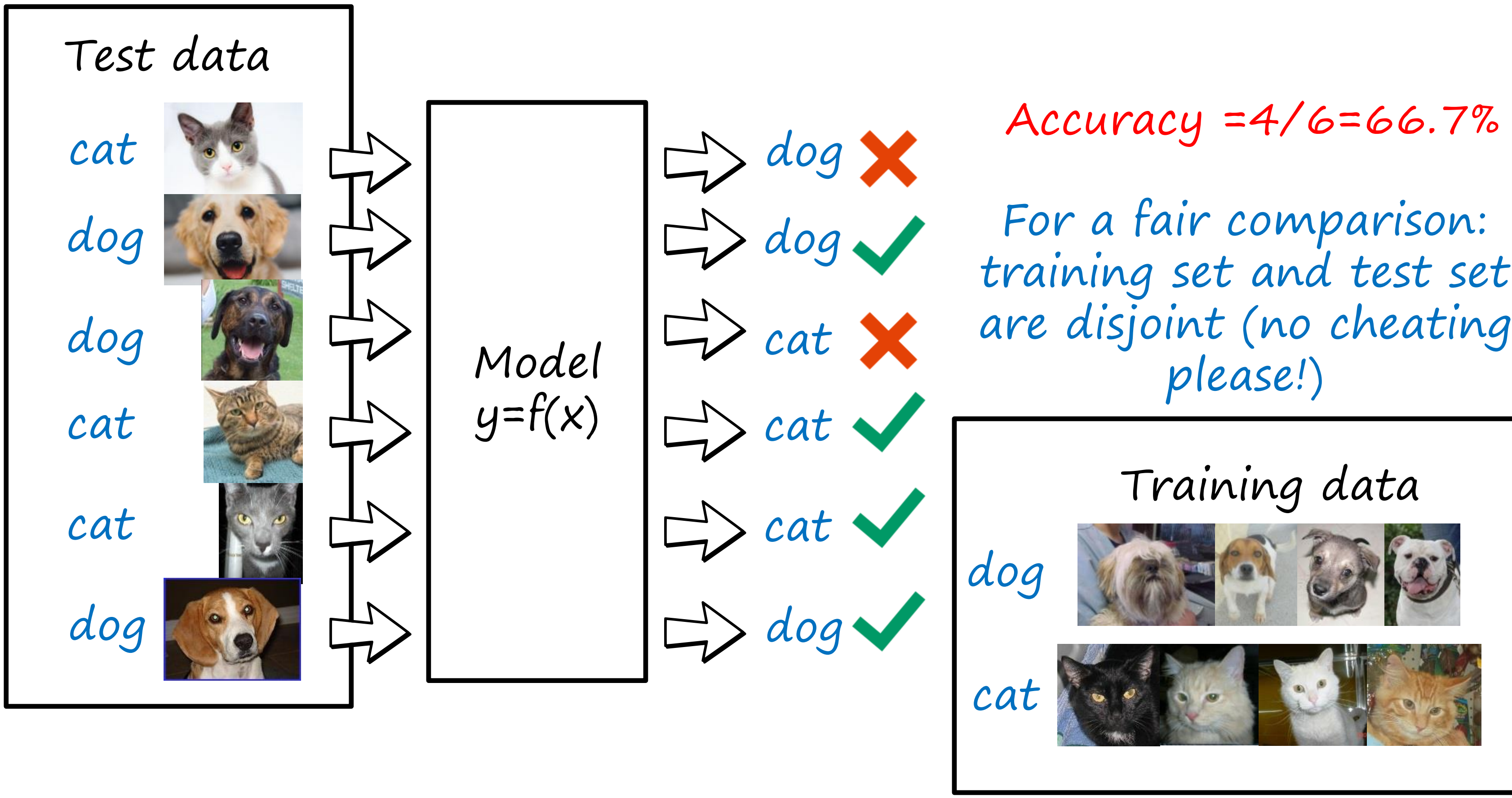
Machine learning in a nutshell



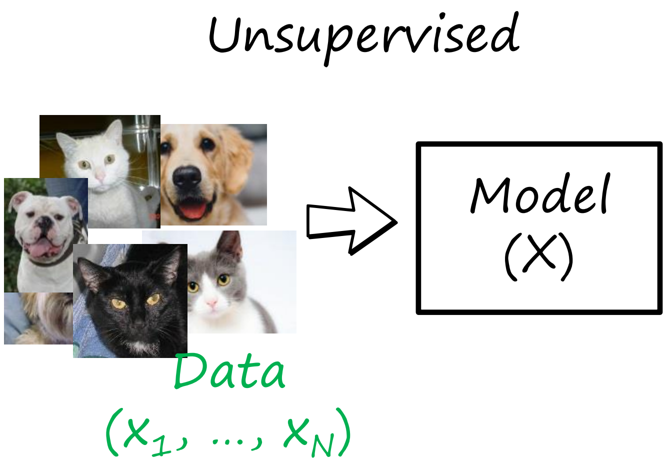
Objective: loss function
(how good/bad our prediction is)

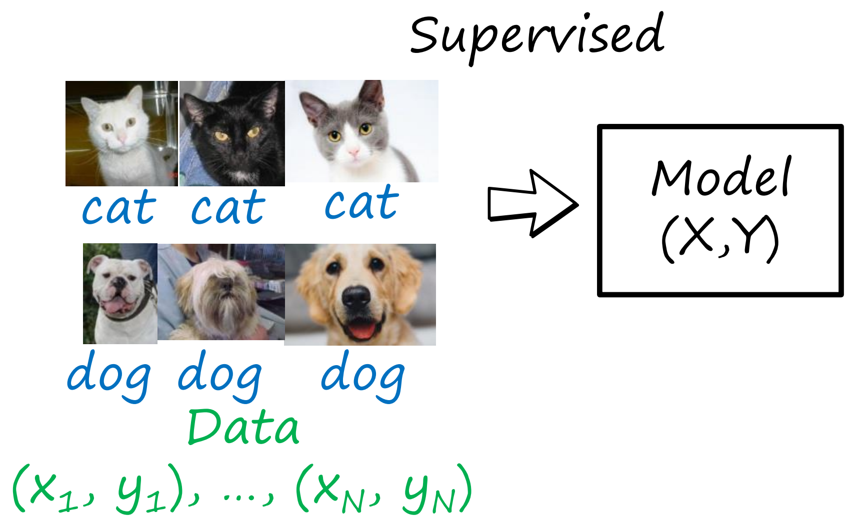
Learning/training
We want to optimize the objective: minimize the average loss in the dataset

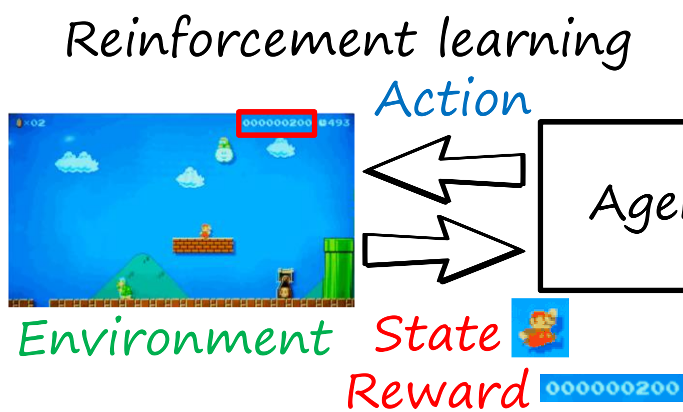
Evaluation


Test data
cat
dog
dog
cat
cat
dog
Model
y=f(x)
dog
dog
cat
cat
cat
dog
Accuracy =4/6=66.7%
For a fair comparison: training set and test set are disjoint (no cheating please!)
Training data
dog
cat

Learning paradigms







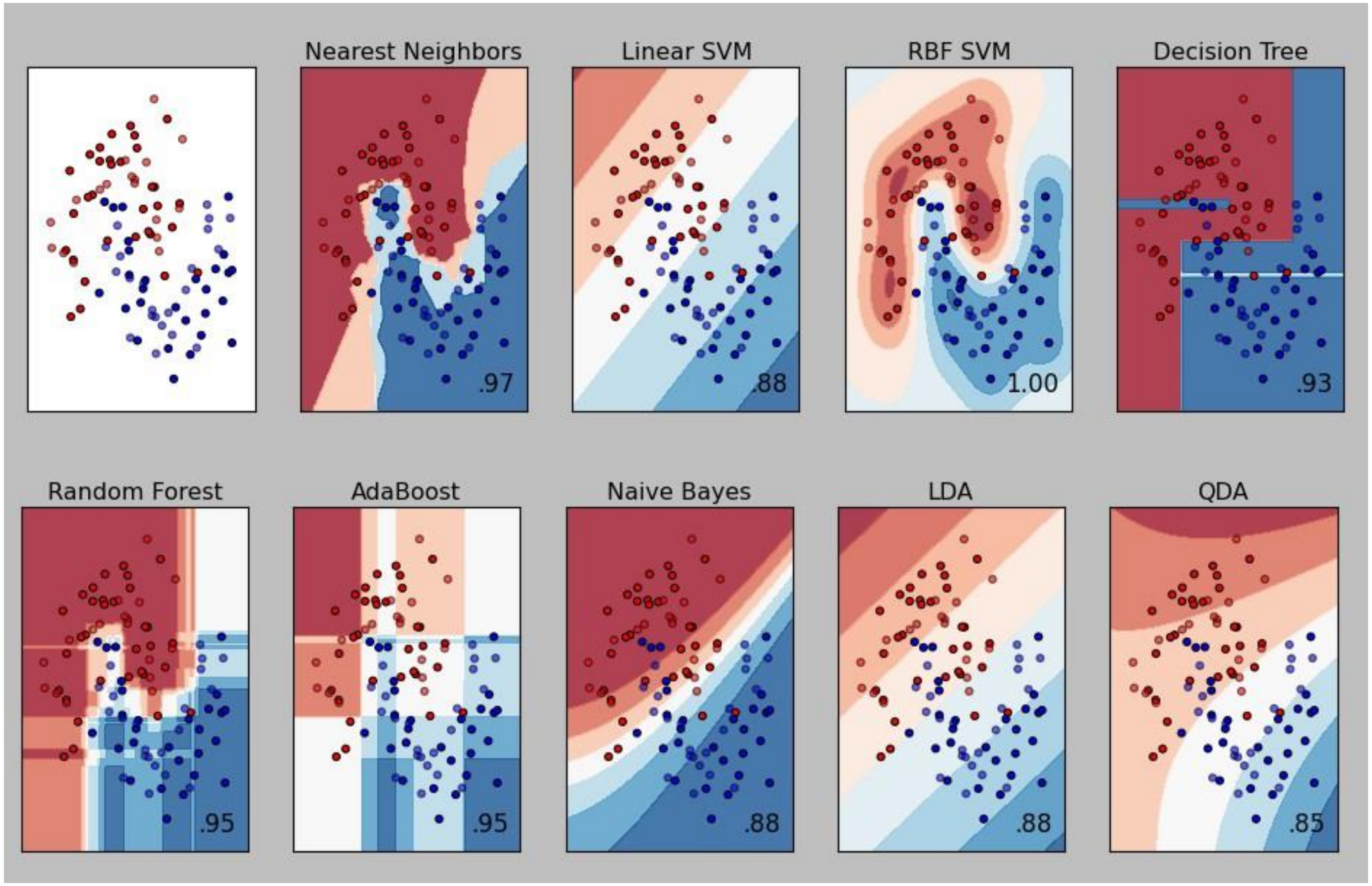
Machine learning applications

- Computer vision
- Autonomous vehicles
- Natural lenguage processing (NLP)
- Speech recognition
- Translation
- Unmanned aerial vehicles (UAV)
- Planning
- Decision taking
- Data mining
- Product recommendation
- …

Machine learning zoo

- Clustering
- Rule-based learning
- Decision trees and random forests
- Bayesian learning
- Boosting
- Ensemble learning
- Matrix factorization
- Genetic and evolutionary algorithms
- Support vector machines
- Kernel methods
- Artificial neural networks
  - Deep learning
- …

Machine learning zoo



http://www.androidauthority.com/what-is-machine-learning-621659

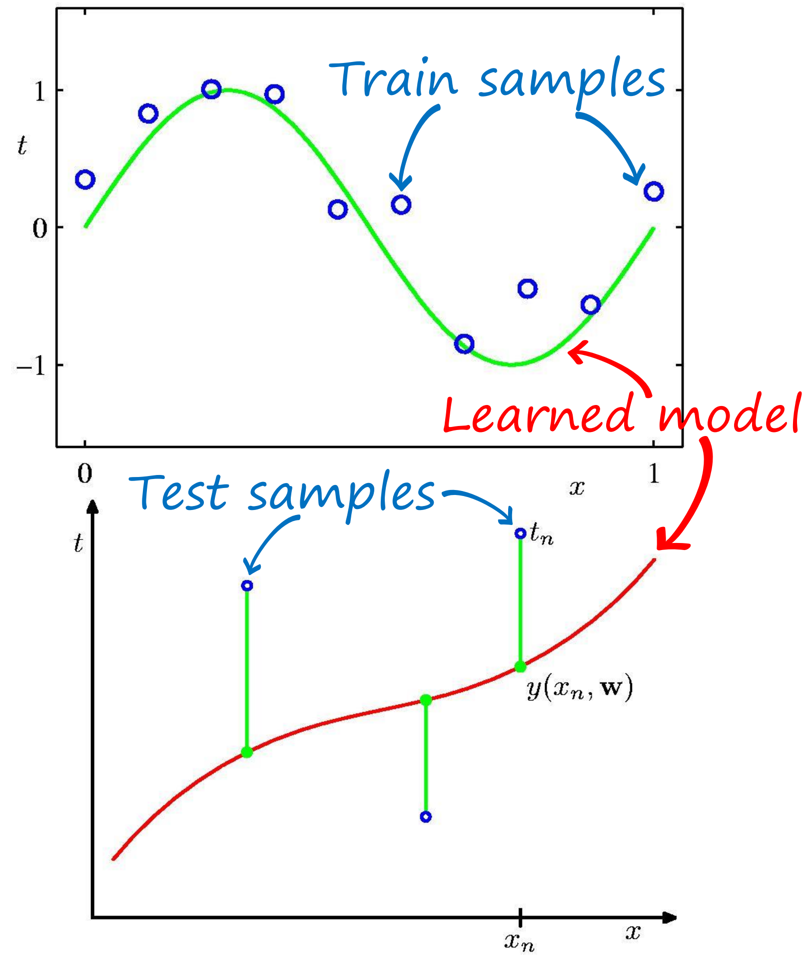
Toy example: fitting a curve

- Data: 2-D points
- Regression problem
- Model: polynomial curve

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

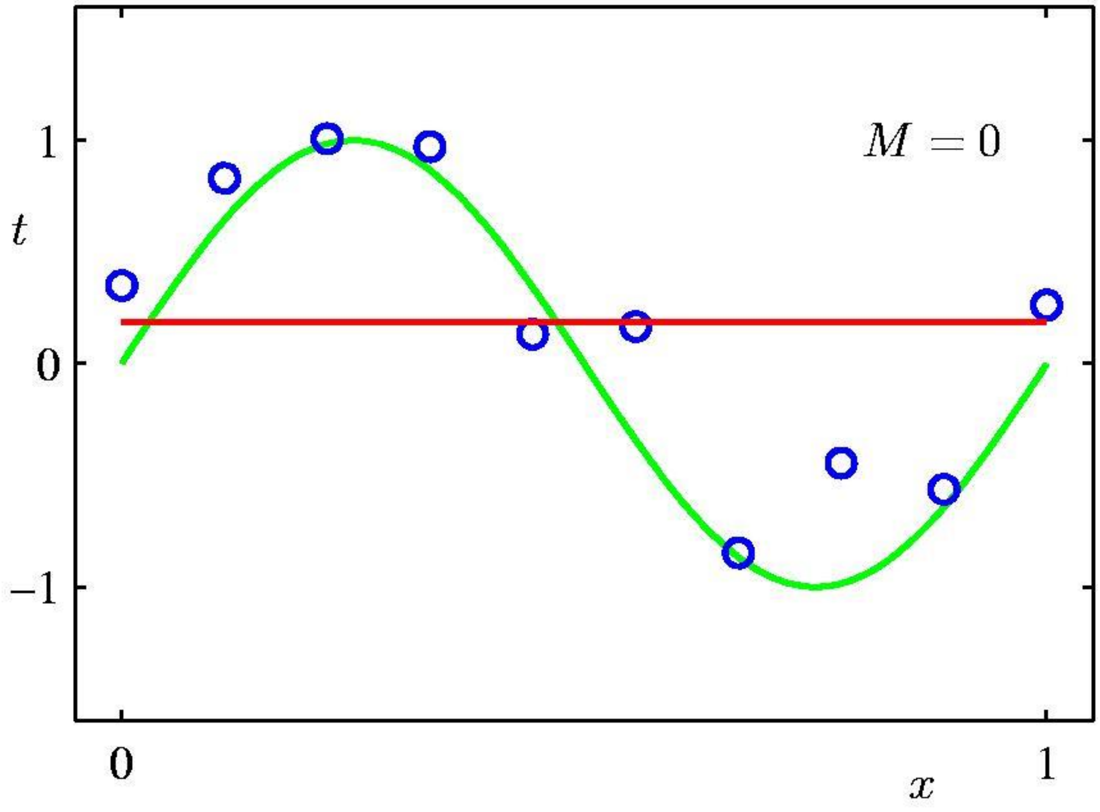
- Error function:
  - Sum-of-Squares

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$



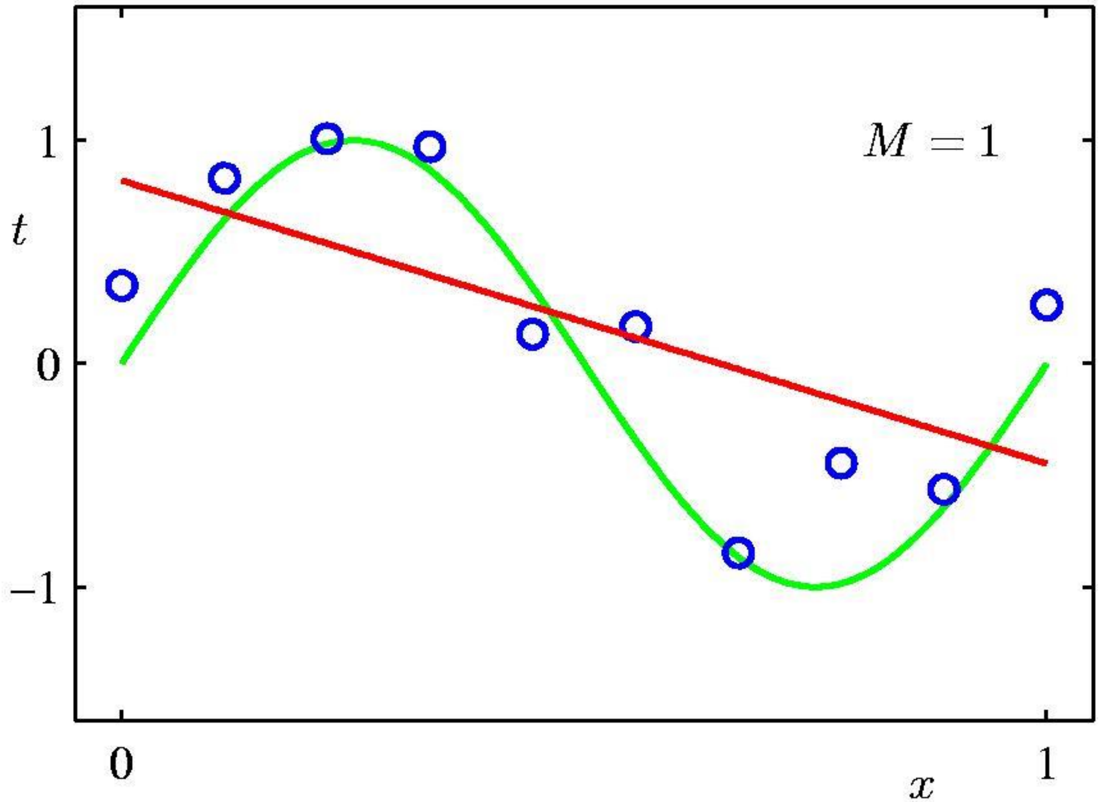
From C. Bishop's Pattern Recognition and Machine Learning https://www.microsoft.com/en-us/research/people/cmbishop

Toy example: 0^{th} Order Polynomial



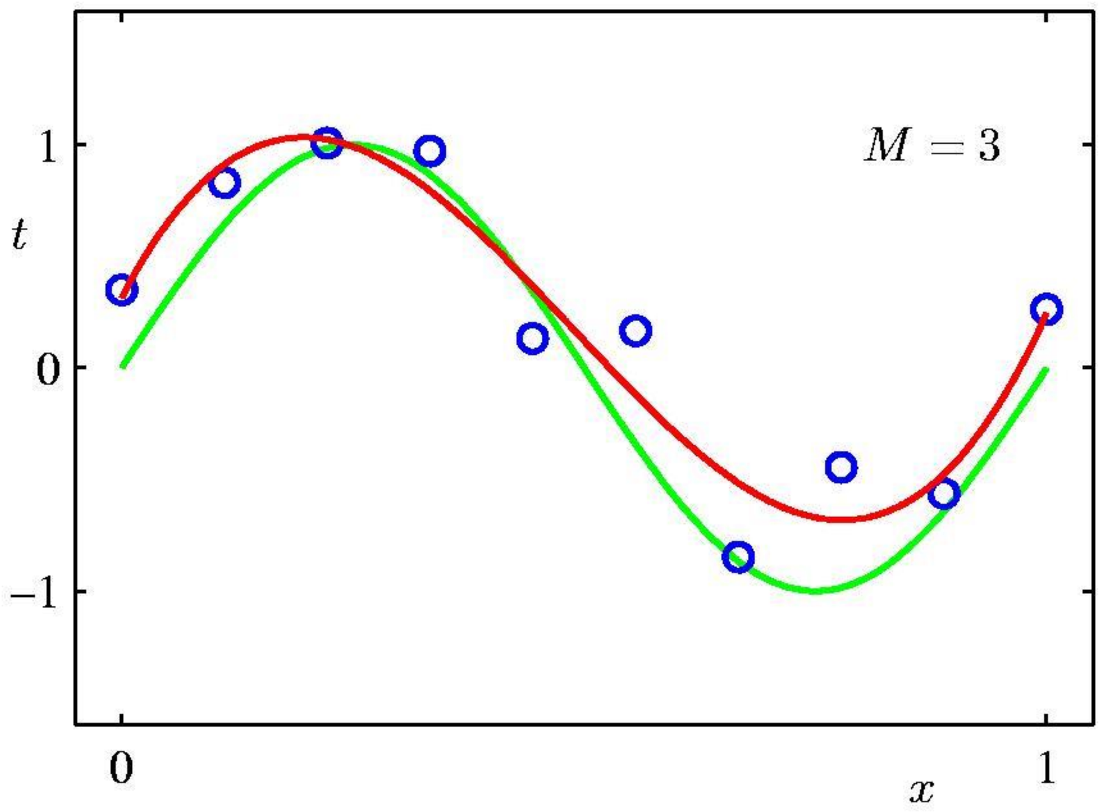
From C. Bishop’s Pattern Recognition and Machine Learning https://www.microsoft.com/en-us/research/people/cmbishop

Toy example: 1st Order Polynomial



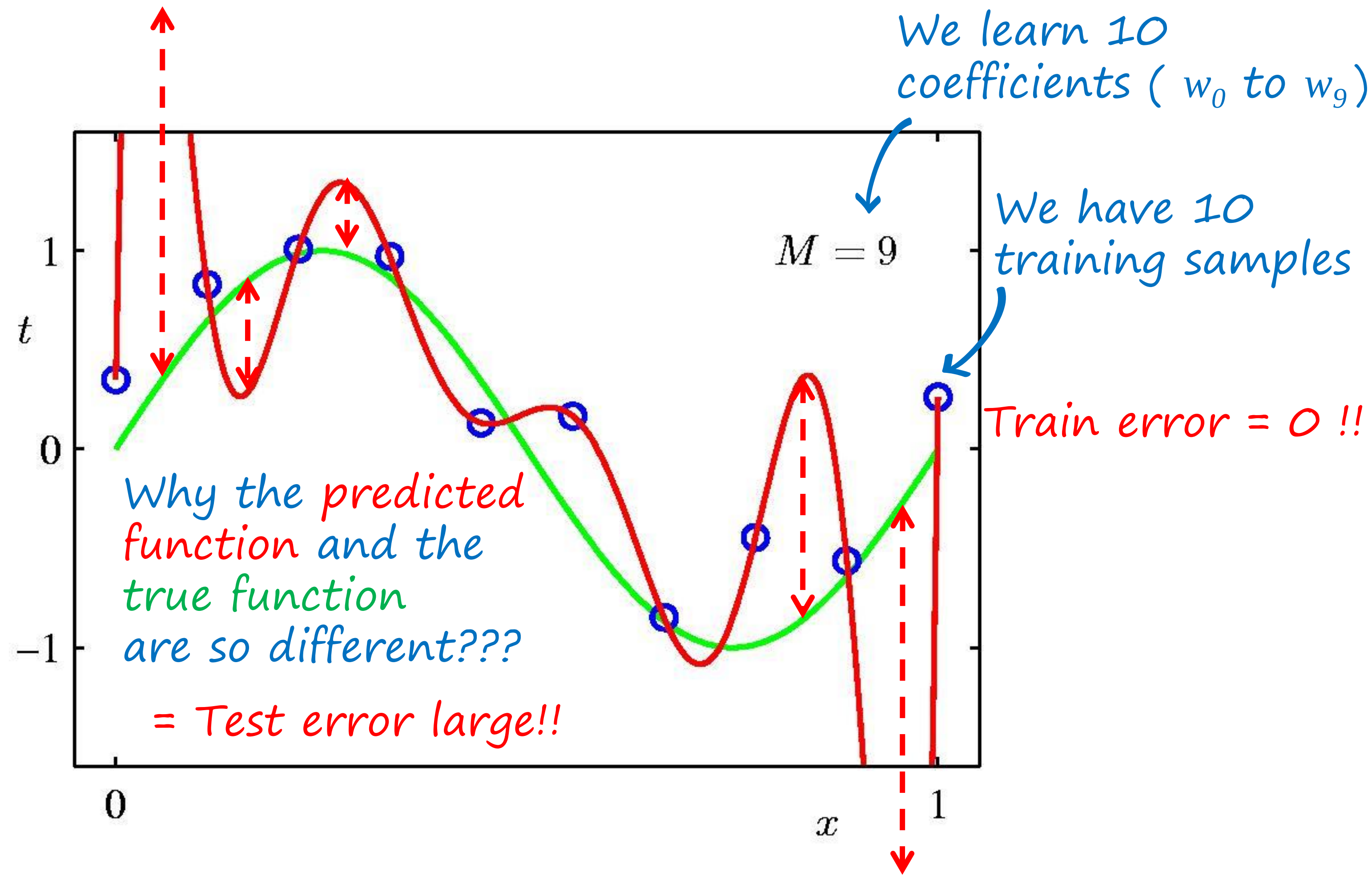
From C. Bishop's Pattern Recognition and Machine Learning https://www.microsoft.com/en-us/research/people/cmbishop

Toy example: 3rd Order Polynomial



From C. Bishop's Pattern Recognition and Machine Learning https://www.microsoft.com/en-us/research/people/cmbishop

Toy example: 9th Order Polynomial



From C. Bishop's Pattern Recognition and Machine Learning https://www.microsoft.com/en-us/research/people/cmbishop

Toy example: Polynomial Coefficients

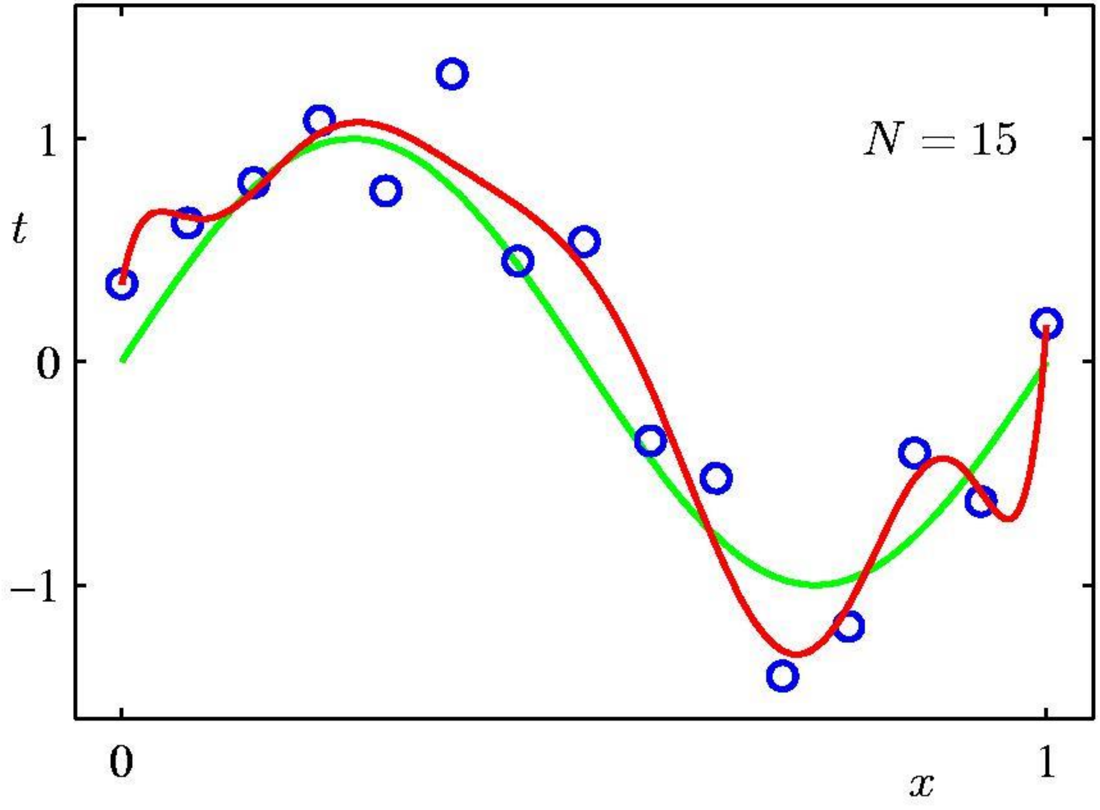
	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Too high!!

Toy example: more data

9th Order Polynomial

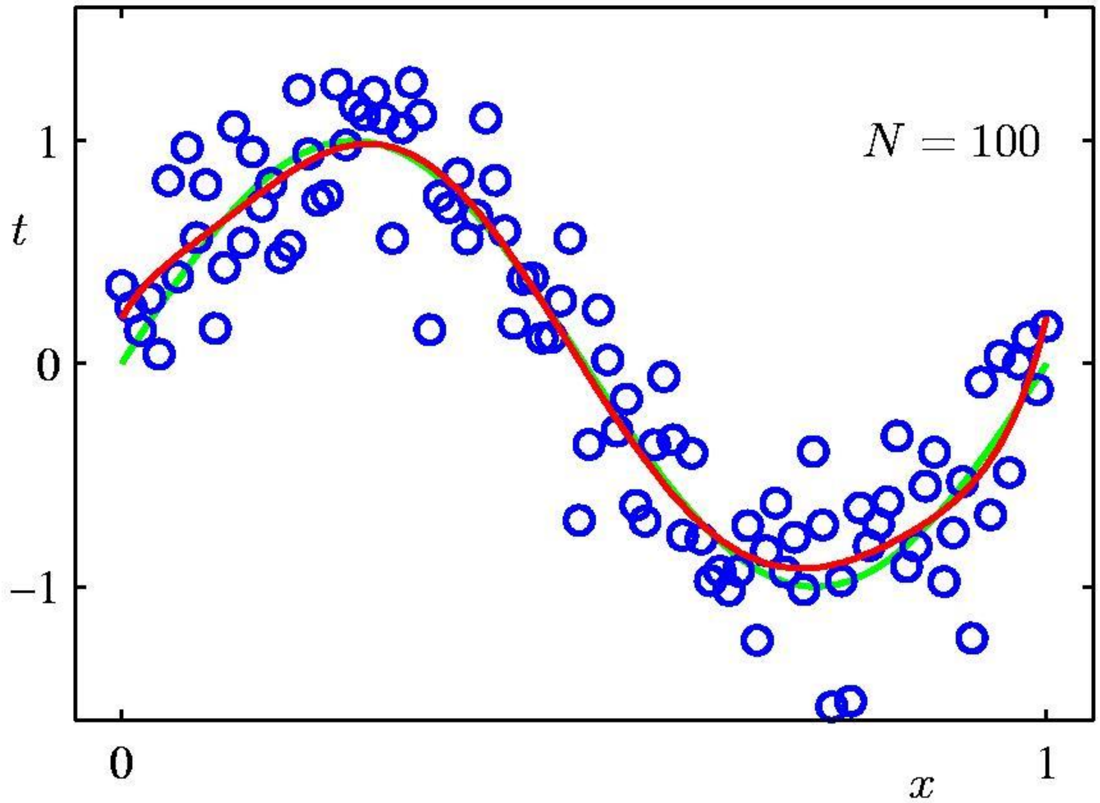
Train error > 0
Test error lower



Toy example: even more data

9th Order Polynomial

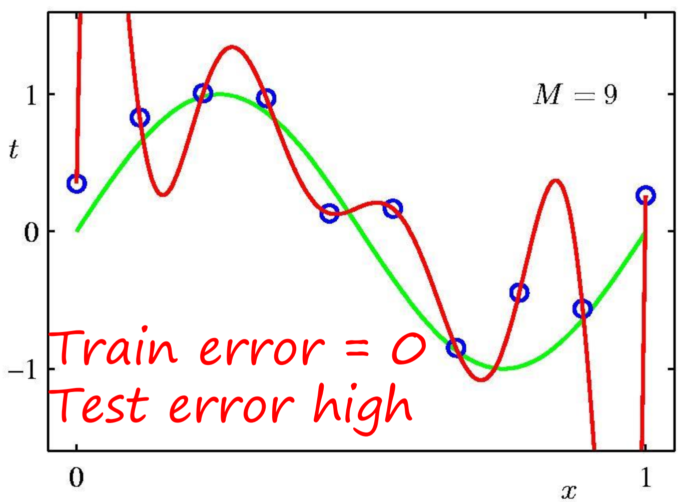
Train error ≈ test error

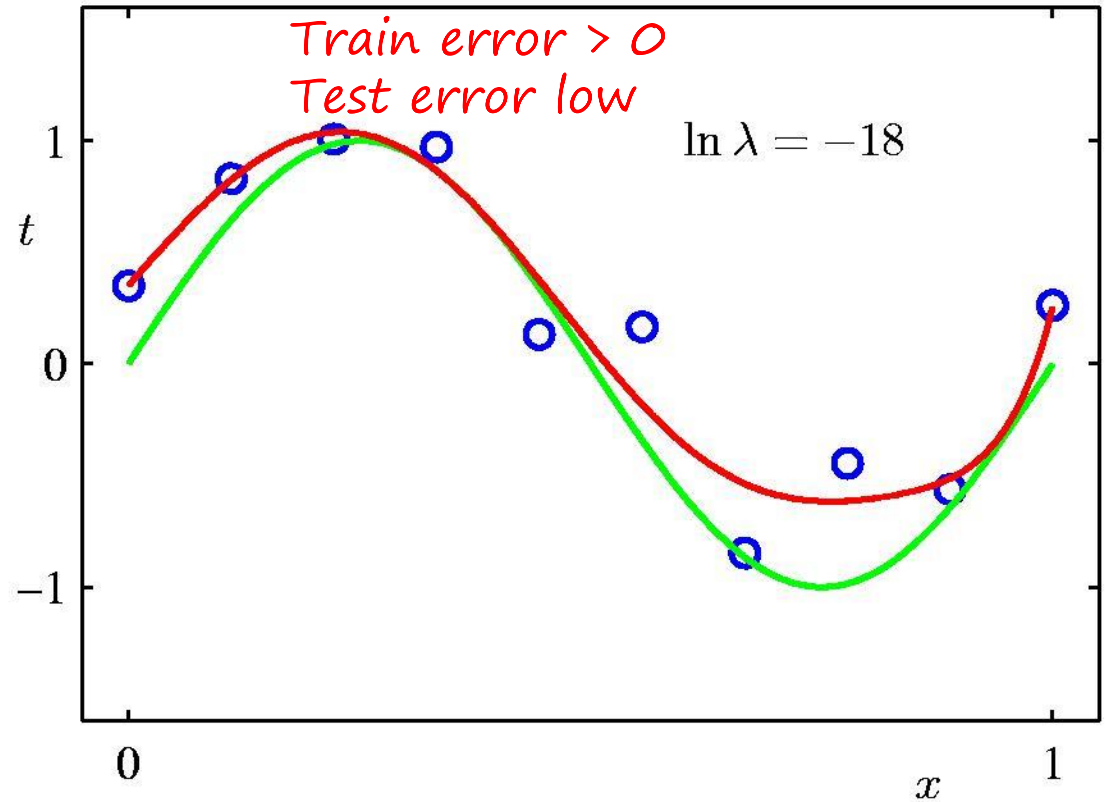


Toy example: regularization

Penalize large coefficient values $\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$

No penalty (λ=0)



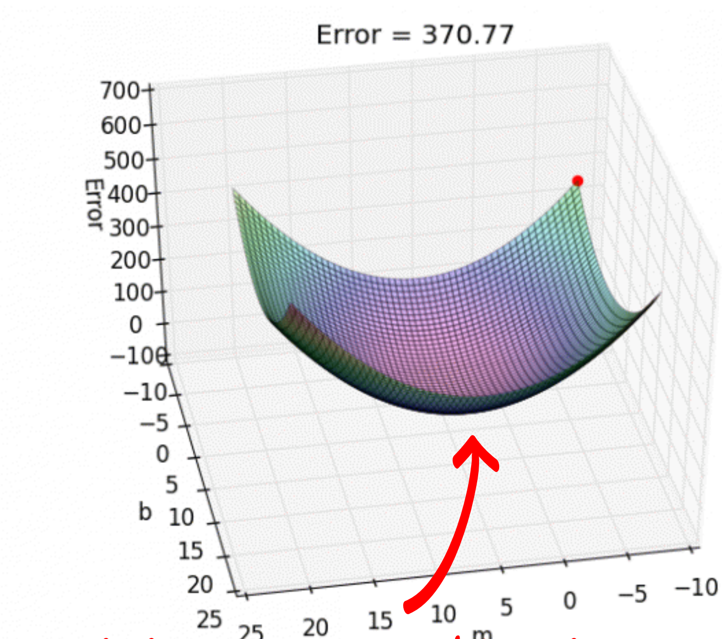


Training: gradient descent

What if there is no analytical solution for the minimization?

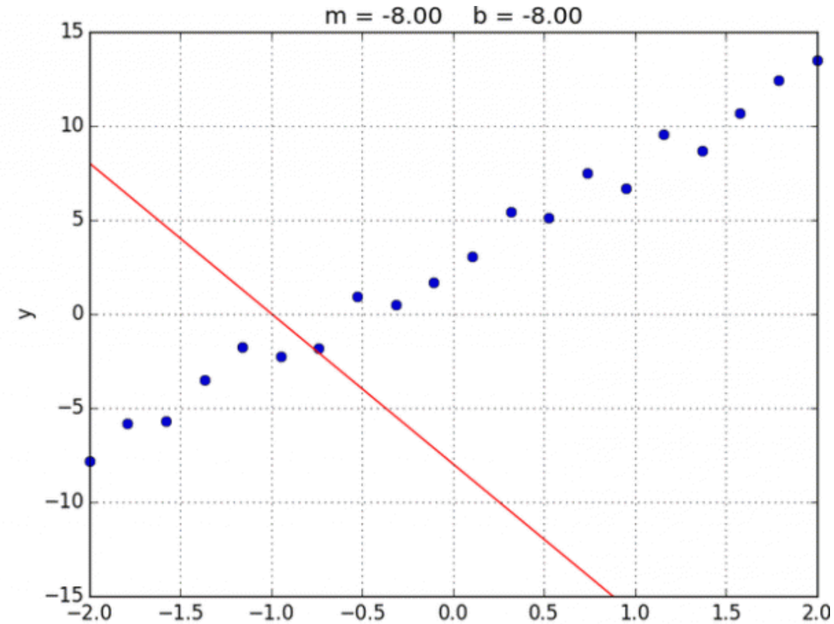
Go step by step in the descending direction

Parameter space



Minimum (our objective)

Model $f(x) = mx + b$



Update rule
(γ=learning rate)

$$m_{j+1} = m_j - \gamma \frac{\partial E}{m_j}$$

$$b_{j+1} = b_j - \gamma \frac{\partial E}{b_j}$$

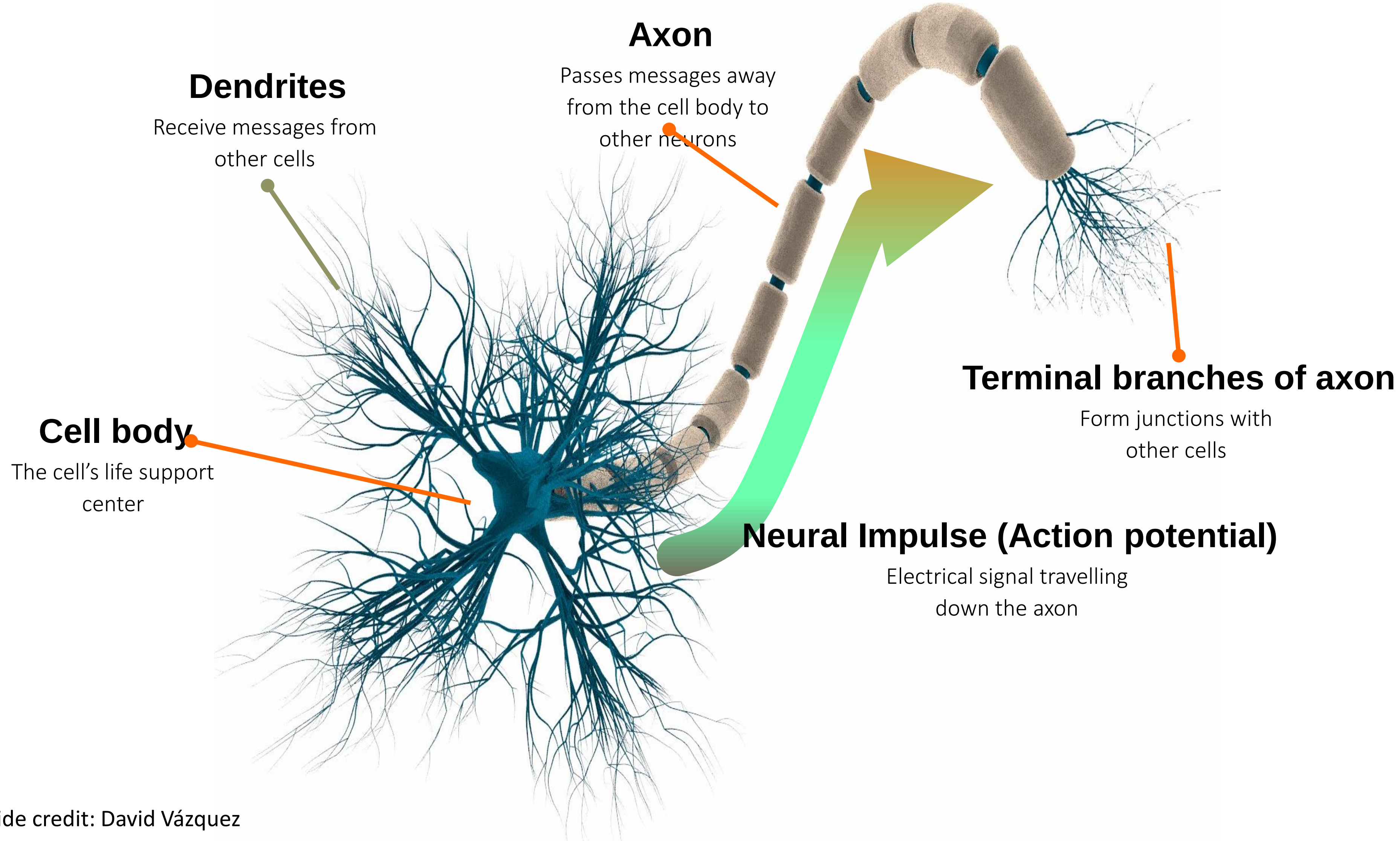
https://alykhantejani.github.io/a-brief-introduction-to-gradient-descent/

Today's plan

- Machine learning
- Neural networks
  - Artificial neural networks
  - Convolutional neural networks
  - Recurrent neural networks
  - Deep learning
- Deep learning in computer vision
- Fun stuff

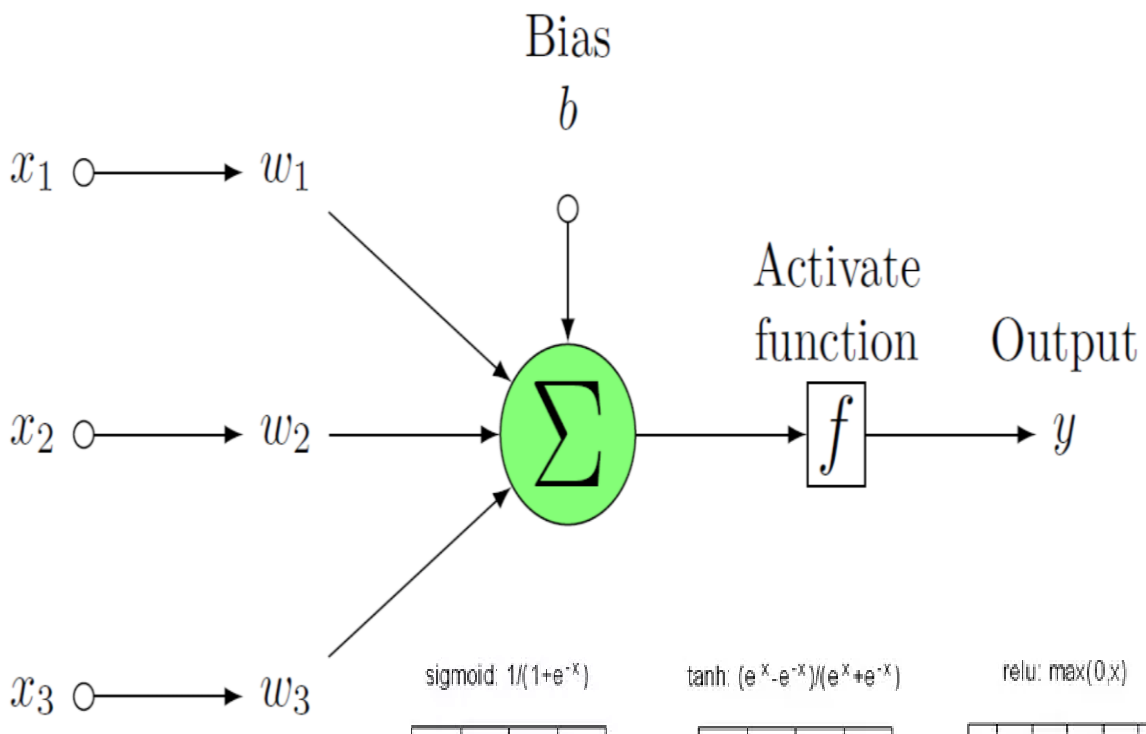
Biological neuron

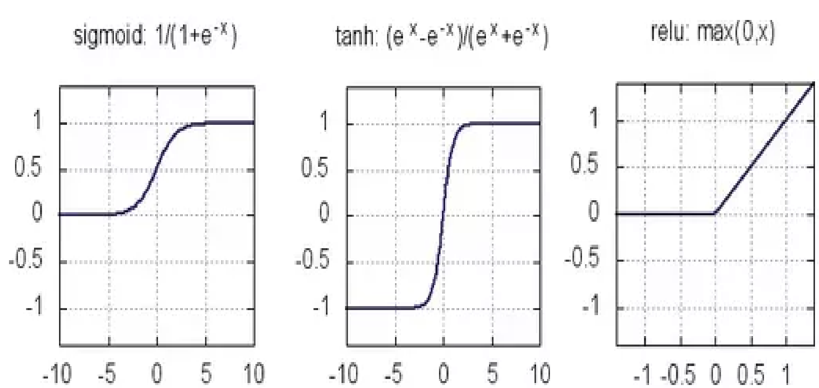
A cell that processes and transmits information through electrical and chemical signals (i.e. synapses)



Slide credit: David Vázquez

Artificial neuron





Neuron pre-activation (or input activation): $a(x) = \sum_{i=0}^{n}(w_i x_i) + b = W^T x + b$

Neuron output

$$y = f(a(x)) = f(W^T x + b)$$

The activation function (f) is a non-linear function. For instance Sigmoid, Tanh or ReLU

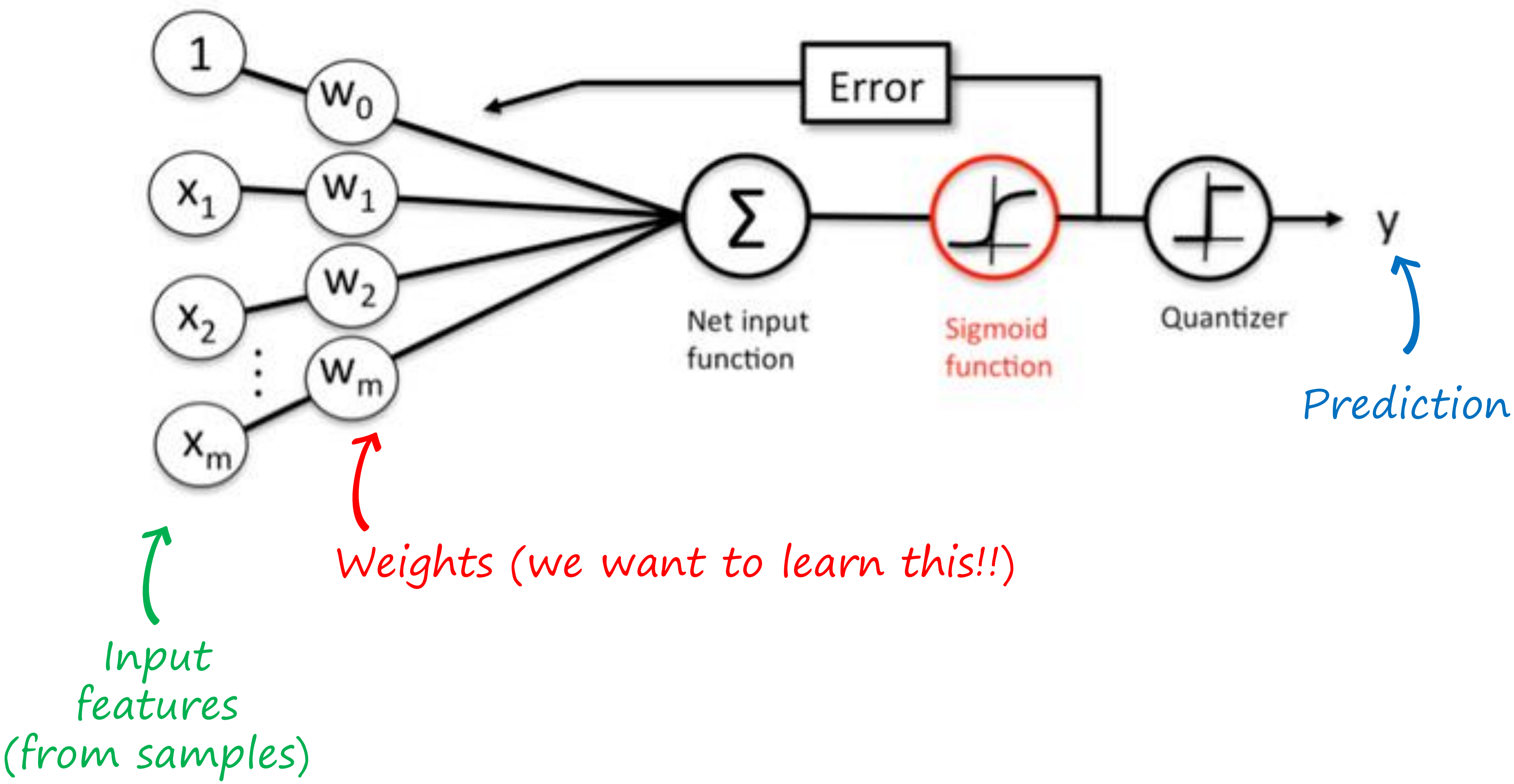
Human brain vs artificial neural network

Human brain: ~100-1000 trillion synapses

Artificial neural network: ~1-10 billion synapses

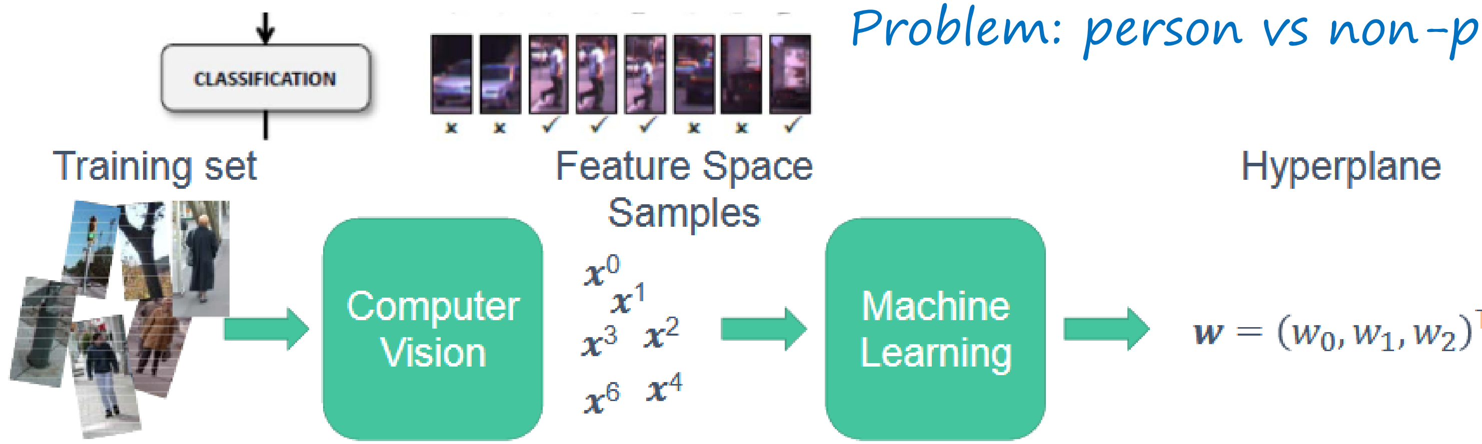
Slide credit: David Vázquez and https://www.otexts.org/fpp/9/3

One layer: logistic regression



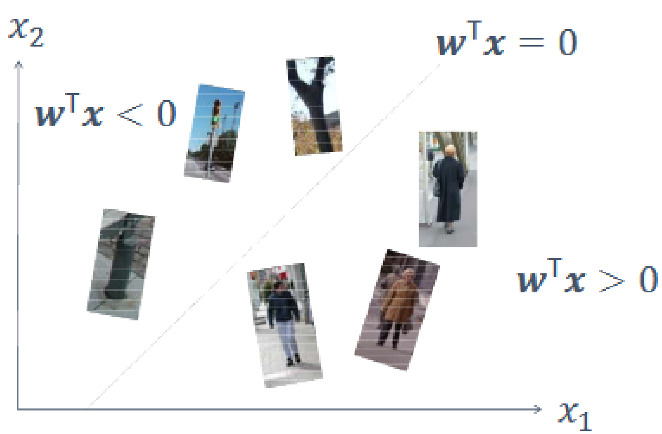
One layer: linear classifier (binary)

Problem: person vs non-person



Only can learn linear separation

- Linear binary classifier in a feature space of $n=2$ dimensions.



$$w = (w_0, w_1, w_2)^\mathrm{T}$$

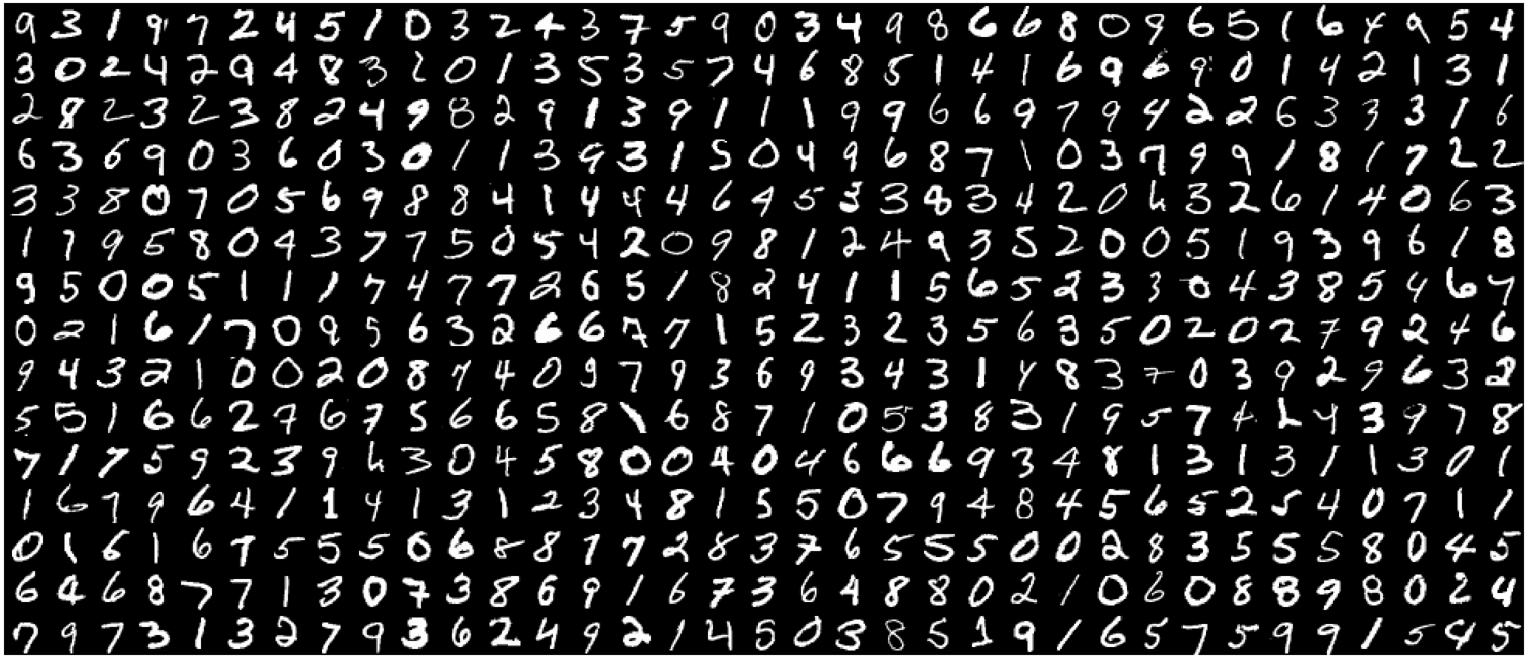
$$x = (1, x_1, x_2)^\mathrm{T}$$

$$\mathit{Classify}(x; w, \mathrm{T}) = \mathit{Threshold}(w^\mathrm{T}x, \mathrm{T})$$

$$\mathit{Threshold}(y, \mathrm{T}) = \begin{cases} 0 \text{ si } y < \mathrm{T} \\ 1 \text{ si } y > \mathrm{T} \\ ? \text{ si } y = \mathrm{T} \quad \text{(design decision)} \end{cases}$$

Slide credit: Antonio Lopez

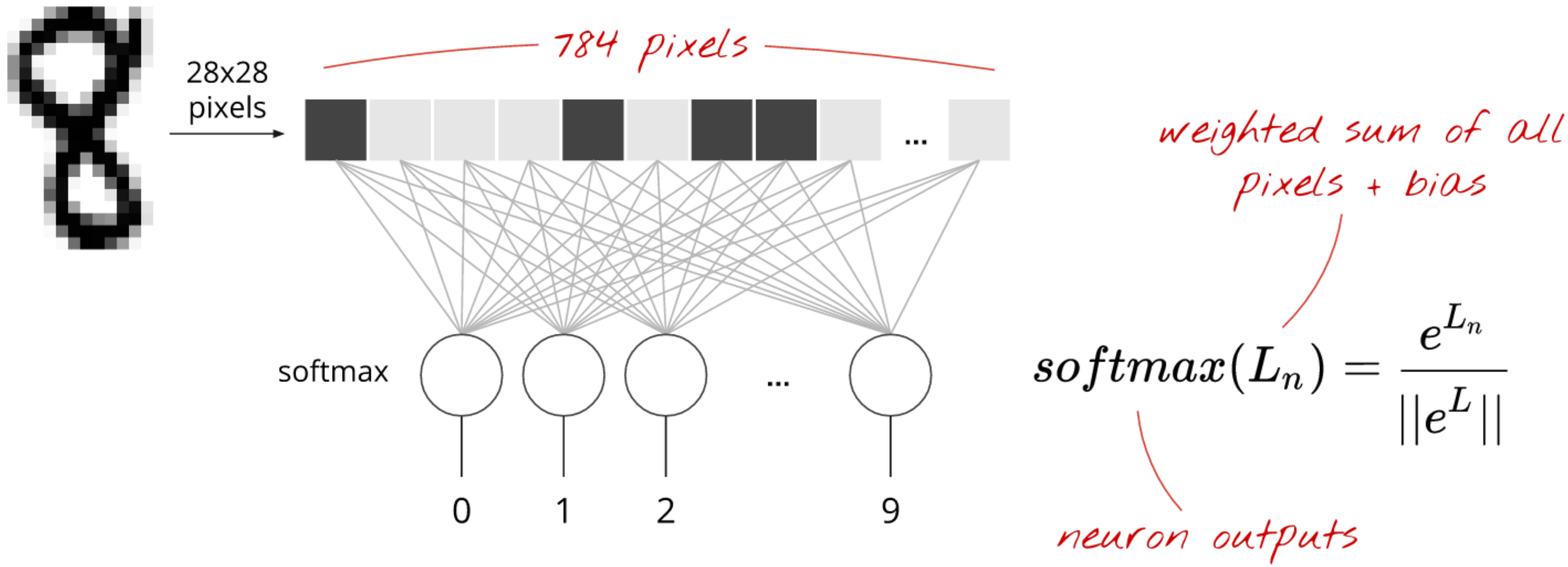
Demo: MNIST dataset



10 classes (0,1,2,3,4,5,6,7,8,9)
60000 images for training, 10000 for test
28x28 grayscale images (=784 values)

https://github.com/martin-gorner/tensorflow-mnist-tutorial/blob/master/mnist_1.0_softmax.py

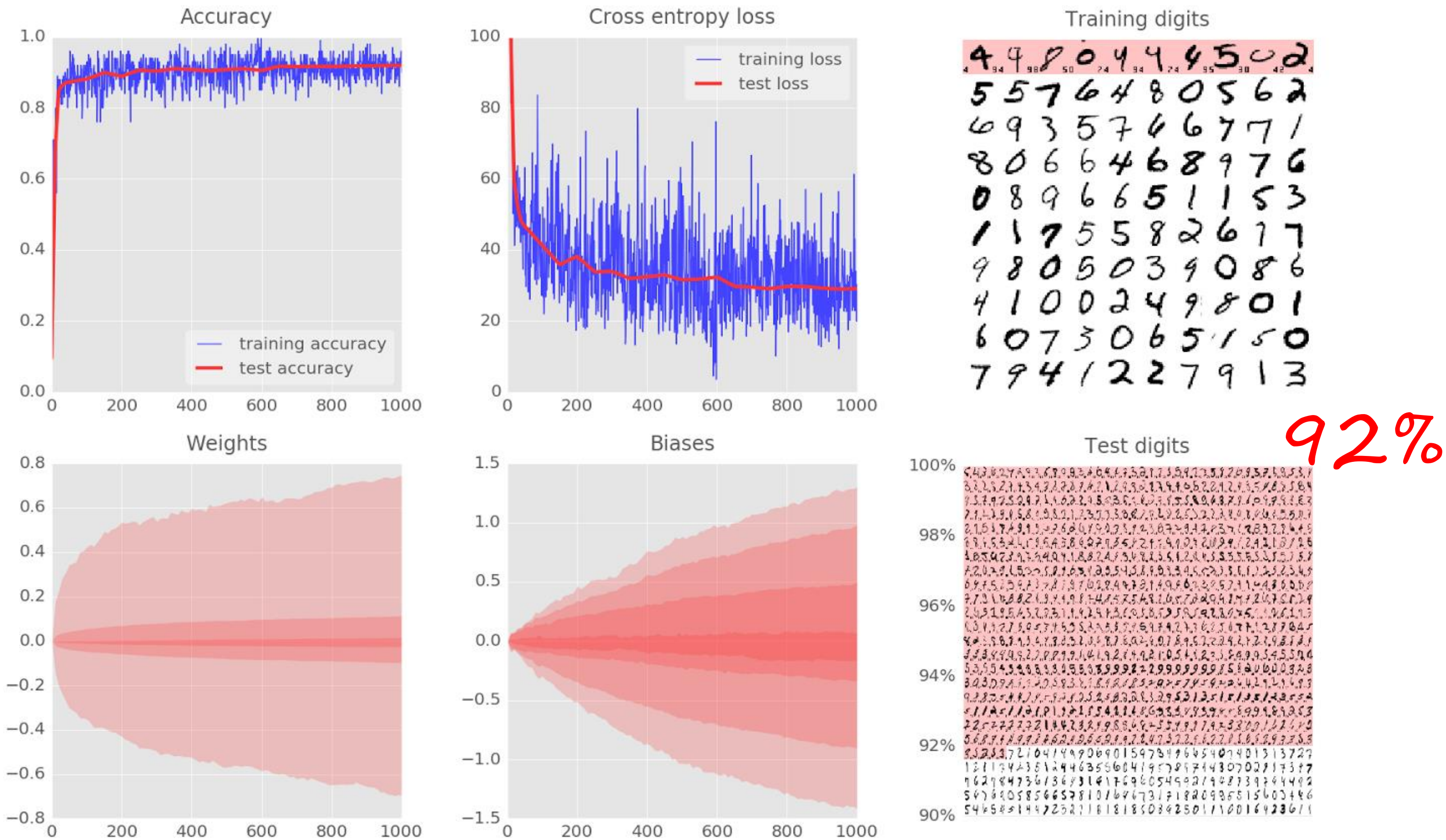
Demo: logistic regression (1 layer NN)



$$softmax(L_n) = \frac{e^{L_n}}{||e^L||}$$

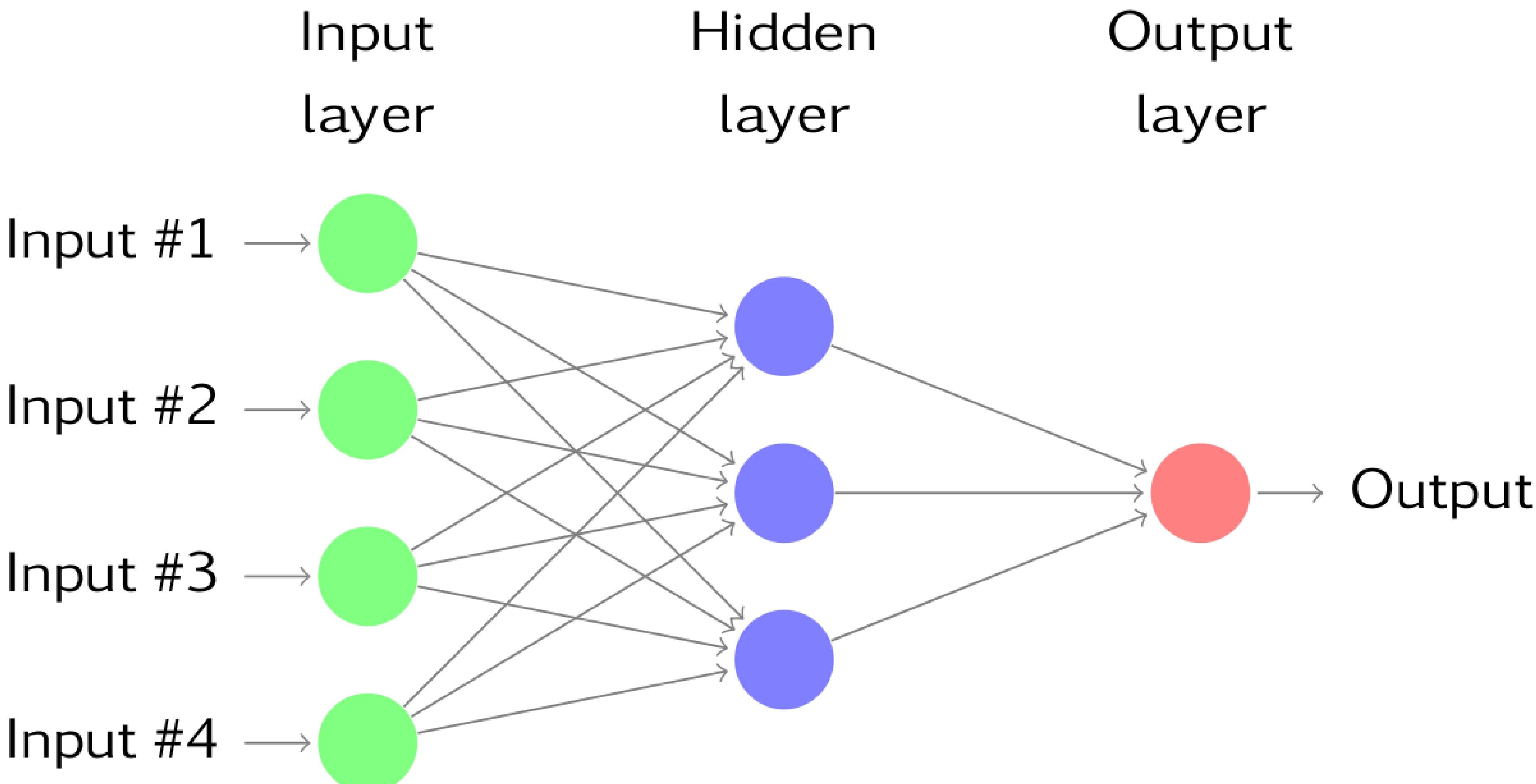
https://github.com/martin-gorner/tensorflow-mnist-tutorial/blob/master/mnist_1.0_softmax.py

Demo: logistic regression (1 layer NN)

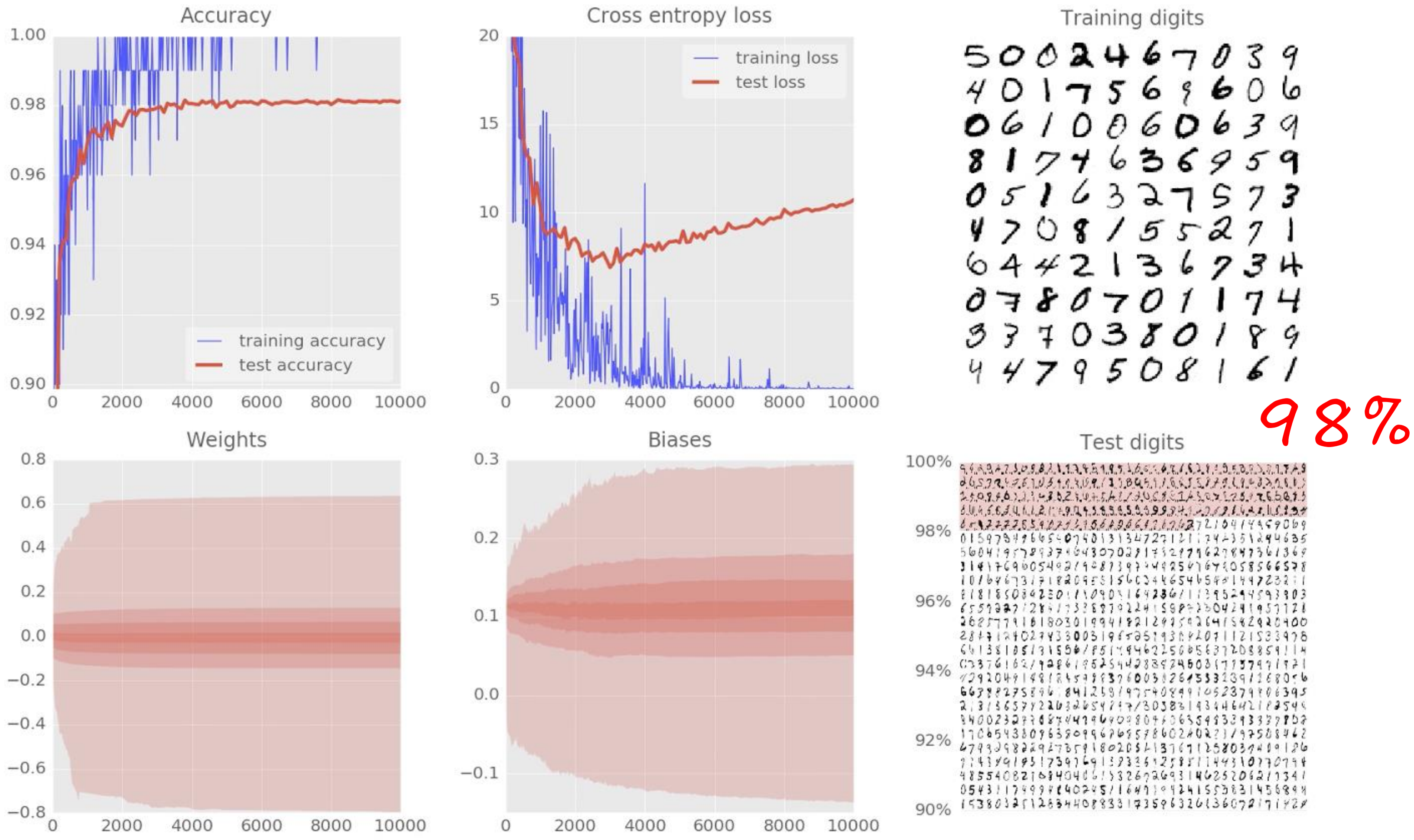


https://github.com/martin-gorner/tensorflow-mnist-tutorial/blob/master/mnist_1.0_softmax.py

Two layers


Input
layer
Hidden
layer
Output
layer
Input #1
Input #2
Input #3
Input #4
Output

Demo: multilayer network

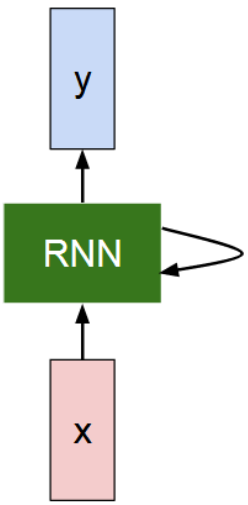


https://github.com/martin-gorner/tensorflow-mnist-tutorial/blob/master/mnist_2.2_five_layers_relu_lrdecay_dropout.py

Recurrent neural networks (RNNs)

(Vanilla) Recurrent Neural Network

The state consists of a single *"hidden"* vector **h**:



$$h_t = f_W(h_{t-1}, x_t)$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

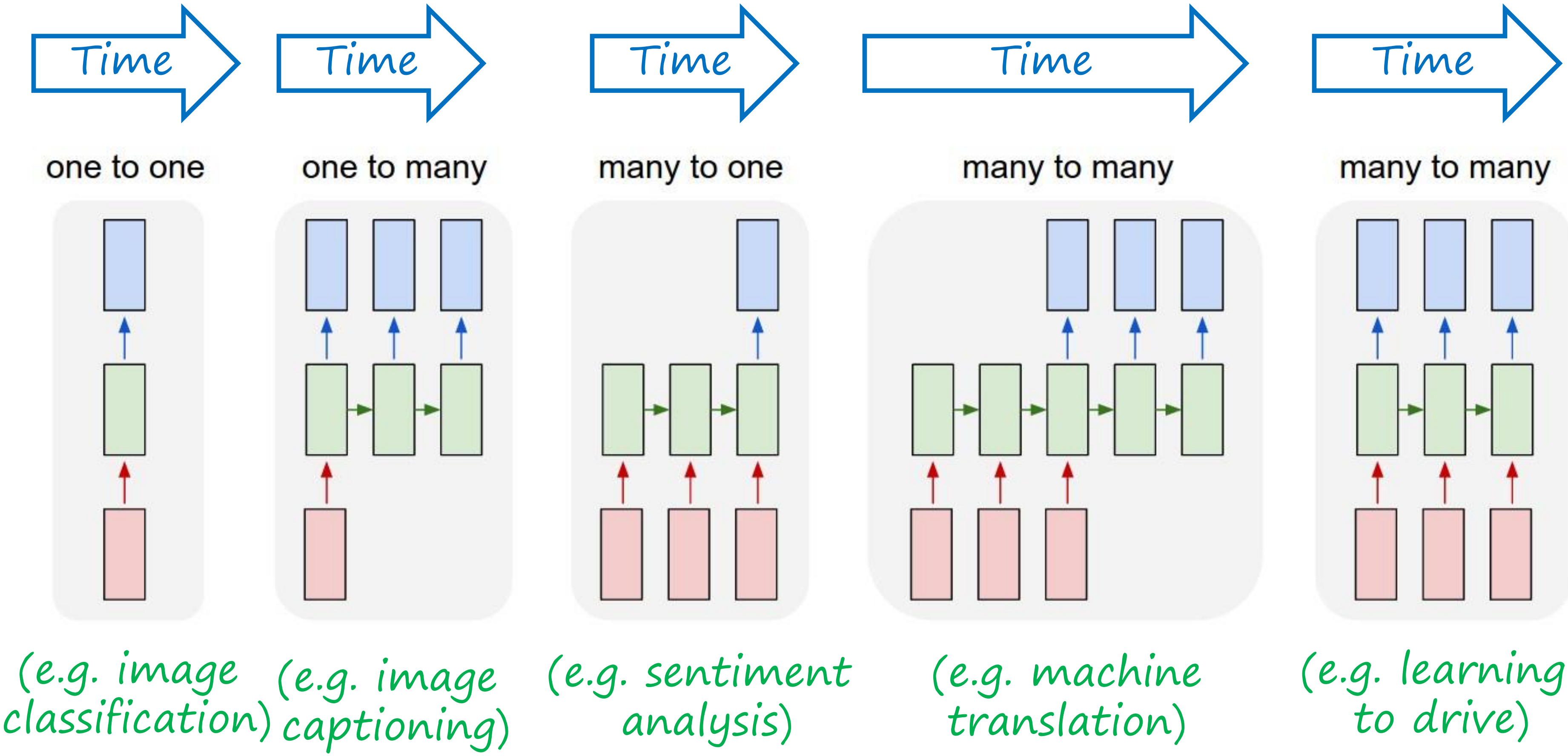
$$y_t = W_{hy}h_t$$

Use Long Short-term Memory (LSTM) units (nobody uses this simple vanilla RNNs in practice)

Adapted from: Convolutional Neural Networks for Visual Recognition
http://cs231n.stanford.edu/

Adapted from: Convolutional Neural Networks for Visual Recognition
http://cs231n.stanford.edu/

Unrolling RNNs



Adapted from: A. Karpathy
http://karpathy.github.io/2015/05/21/rnn-effectiveness

Demo: recurrent neural networks

Guess the letter!

See how well you can figure out what comes next. It's always a letter of the English alphabet (case insensitive) or a space, a comma, an apostrophe, or a period. That's 30 possibilities for every character. Just start typing! The correct text appears here:

Th

What you enter will appear here:

Th

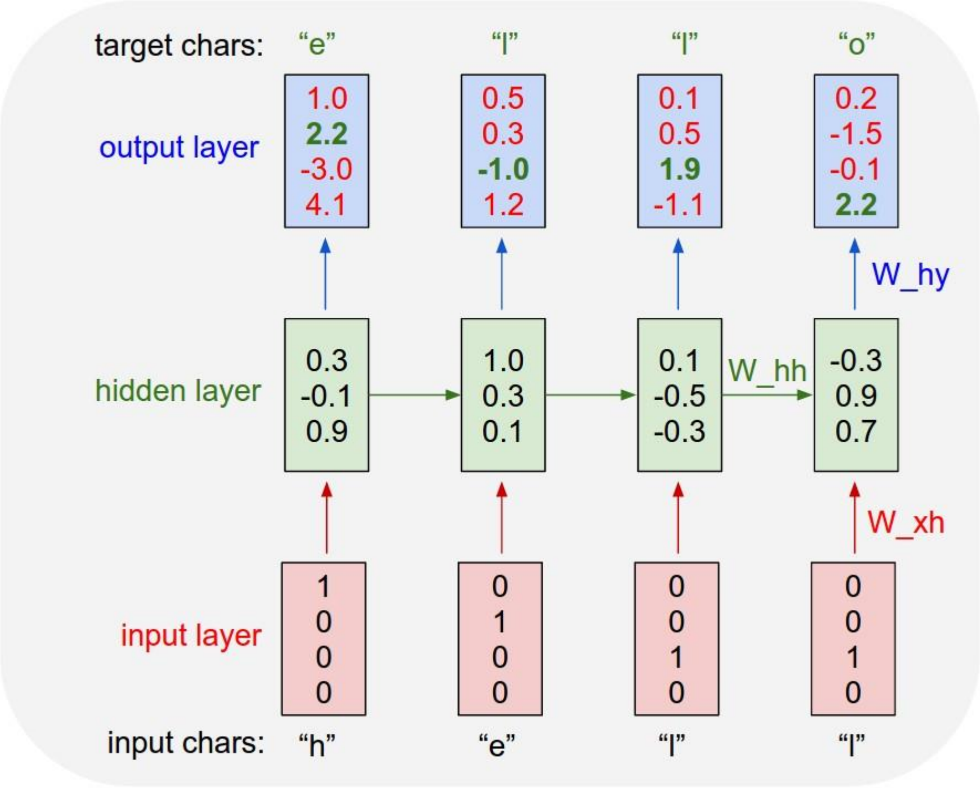
[incomplete] *accuracy*: [unknown]; *perplexity*: [unknown]

(0 of 29 characters; 0 correct) next passage (This is passage 1 of 20.)

No keyboard? Select here: (Then zoom out.)

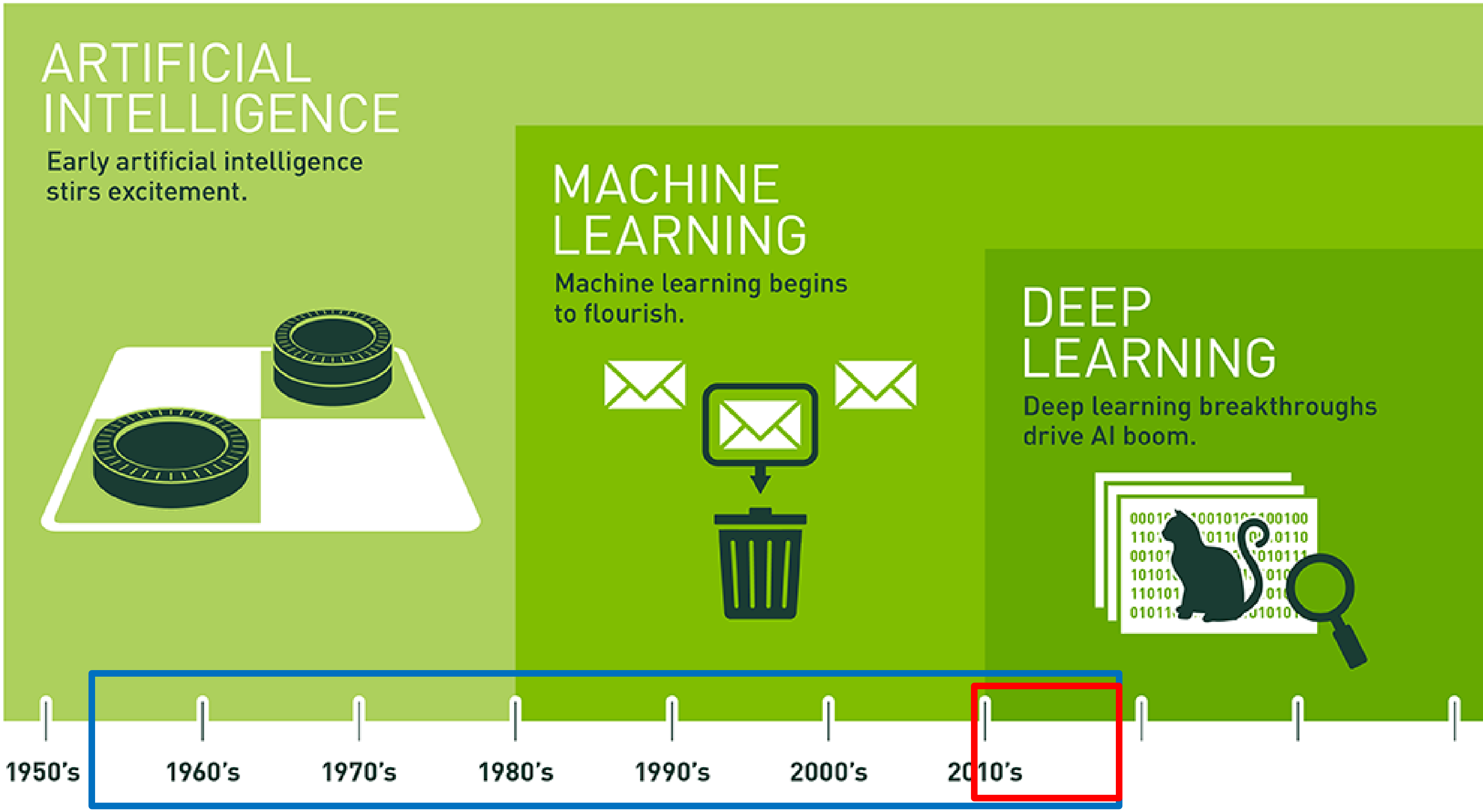
http://planspace.org/perplexity

Demo: recurrent neural networks



Demo: https://github.com/martin-gorner/tensorflow-rnn-shakespeare

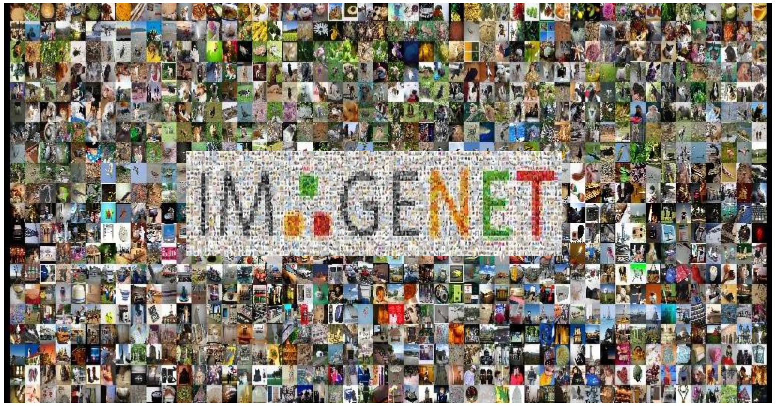
What is deep learning?



Artificial neural networks = Deep learning

https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai

Why does it work now?



Large amount of (annotated) data

Large amount of computing resources (GPUs)

Better understanding of training algorithms

torch

Caffe

theano

TensorFlow

Large community of researchers and open source implementations

Slide credit: David Vázquez

Today’s plan

- Machine learning
- Neural networks
- Deep learning in computer vision
- Fun stuff

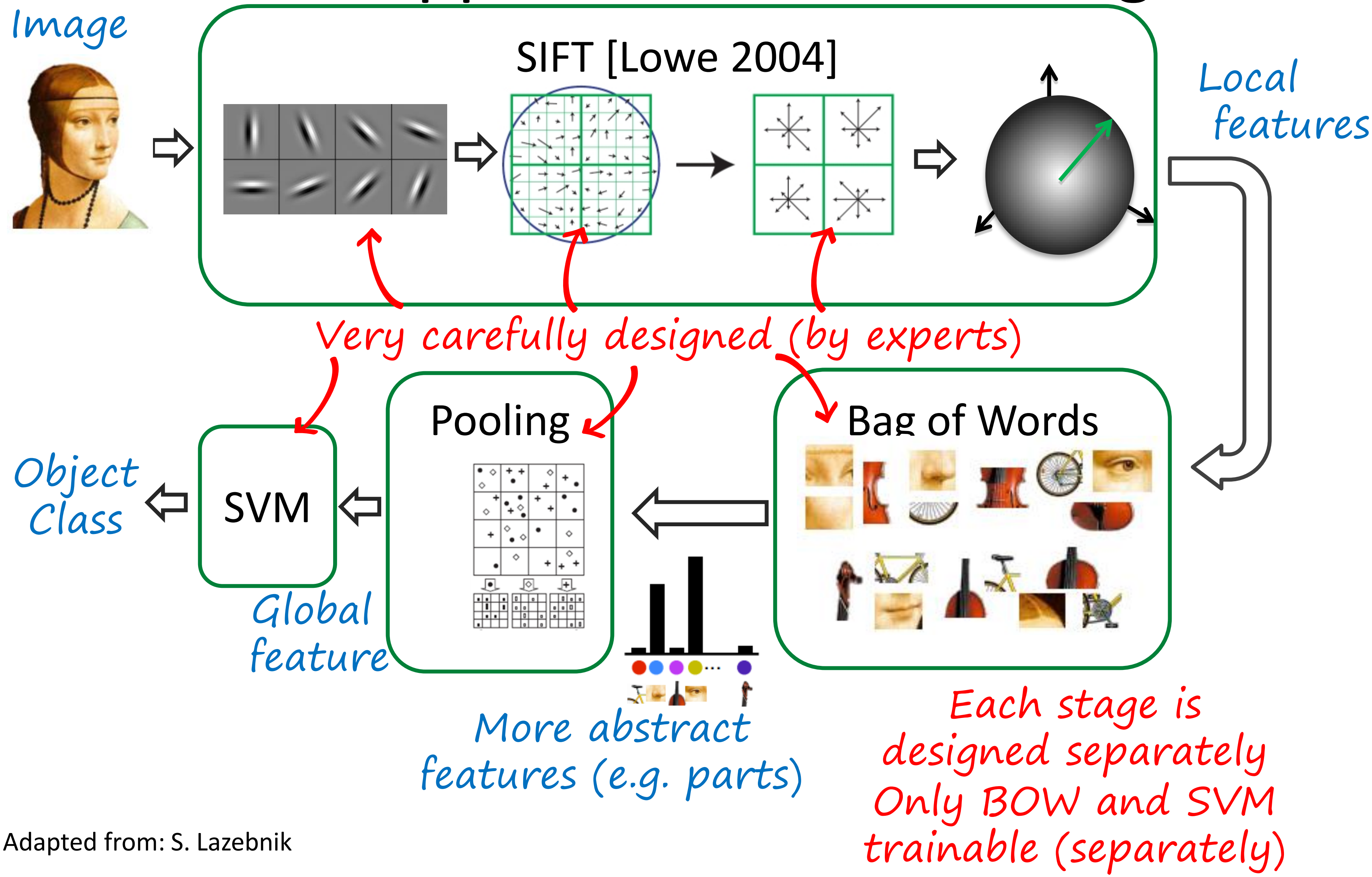
Traditional Recognition Approach

Hand-designed feature extraction → Trainable classifier → Object Class

- Features are not learned
- Trainable classifier is often generic (e.g. SVM)

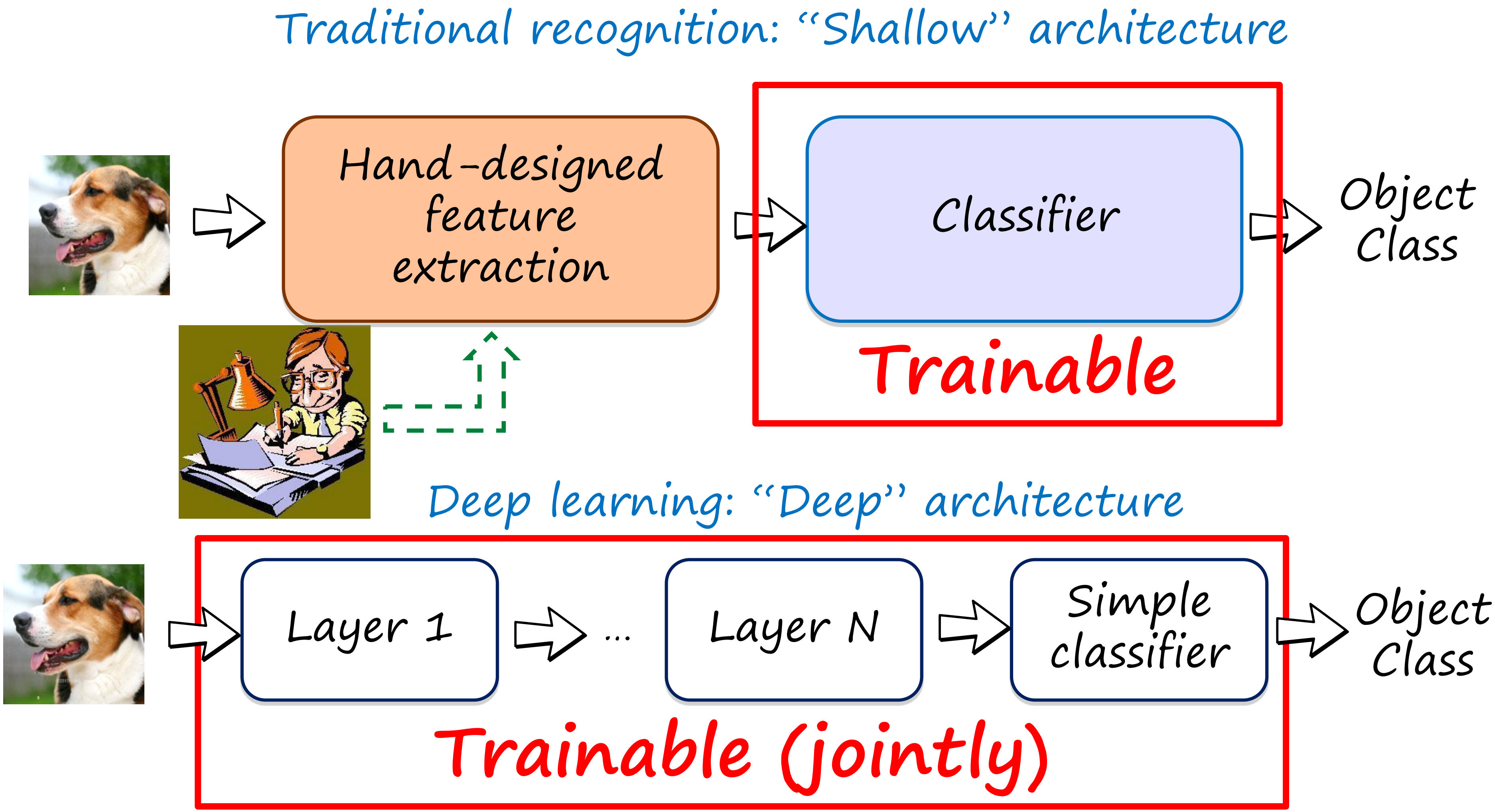
Slide credit: S. Lazebnik

Traditional approach to visual recognition



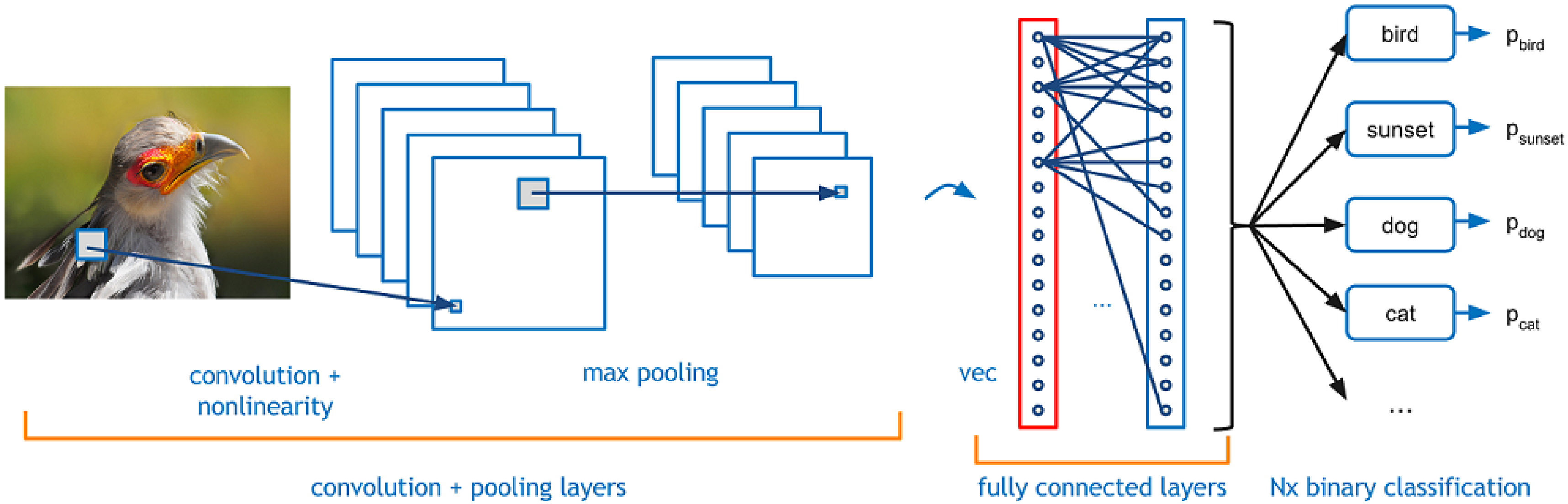
Adapted from: S. Lazebnik

"Shallow" vs. "deep" architectures



Adapted from: S. Lazebnik

Convolutional neural networks (CNNs)

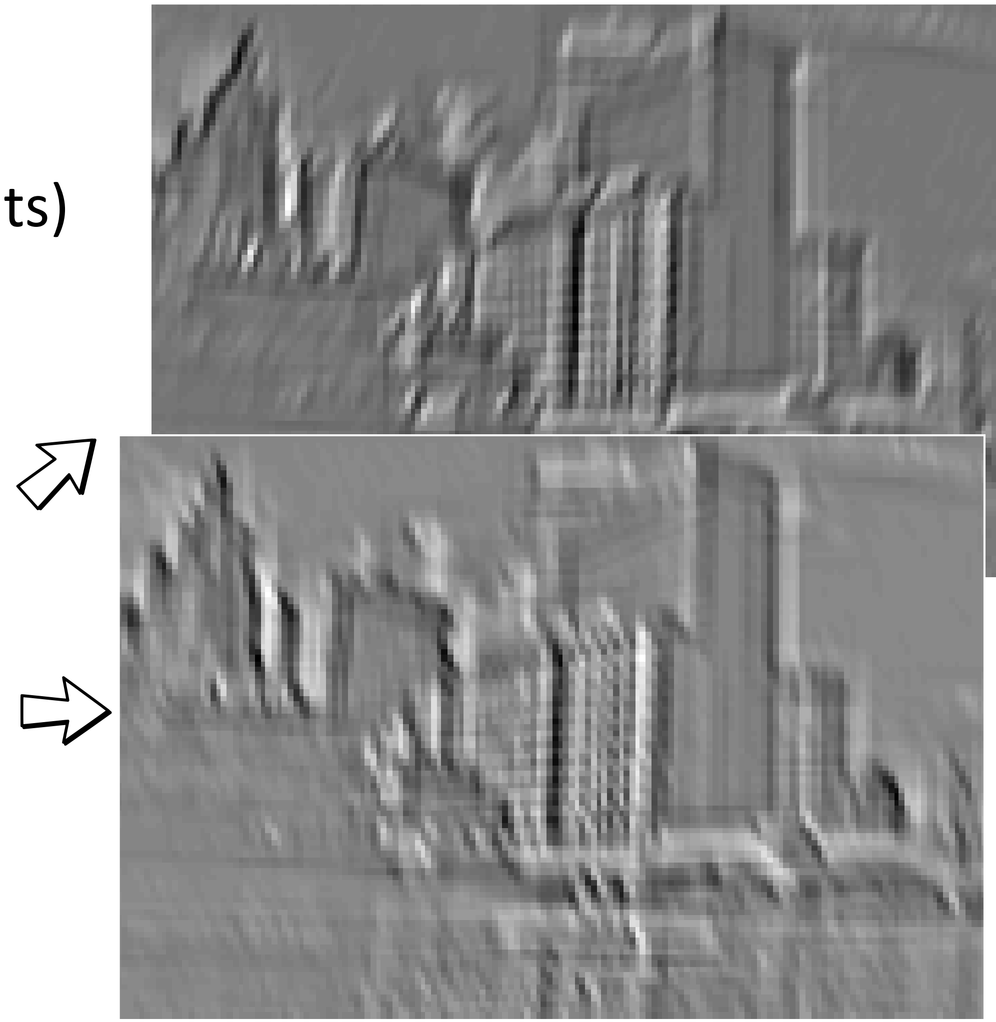


Inspired by the visual system
Old idea (Fukushima 1980, LeCun 1986)
Worked well in character recognition
Larger networks didn't work

Filters as linear convolutions

- Dependencies are local
- Translation invariance
- Few parameters (filter weights)
- Stride can be greater than 1 (faster, less memory)

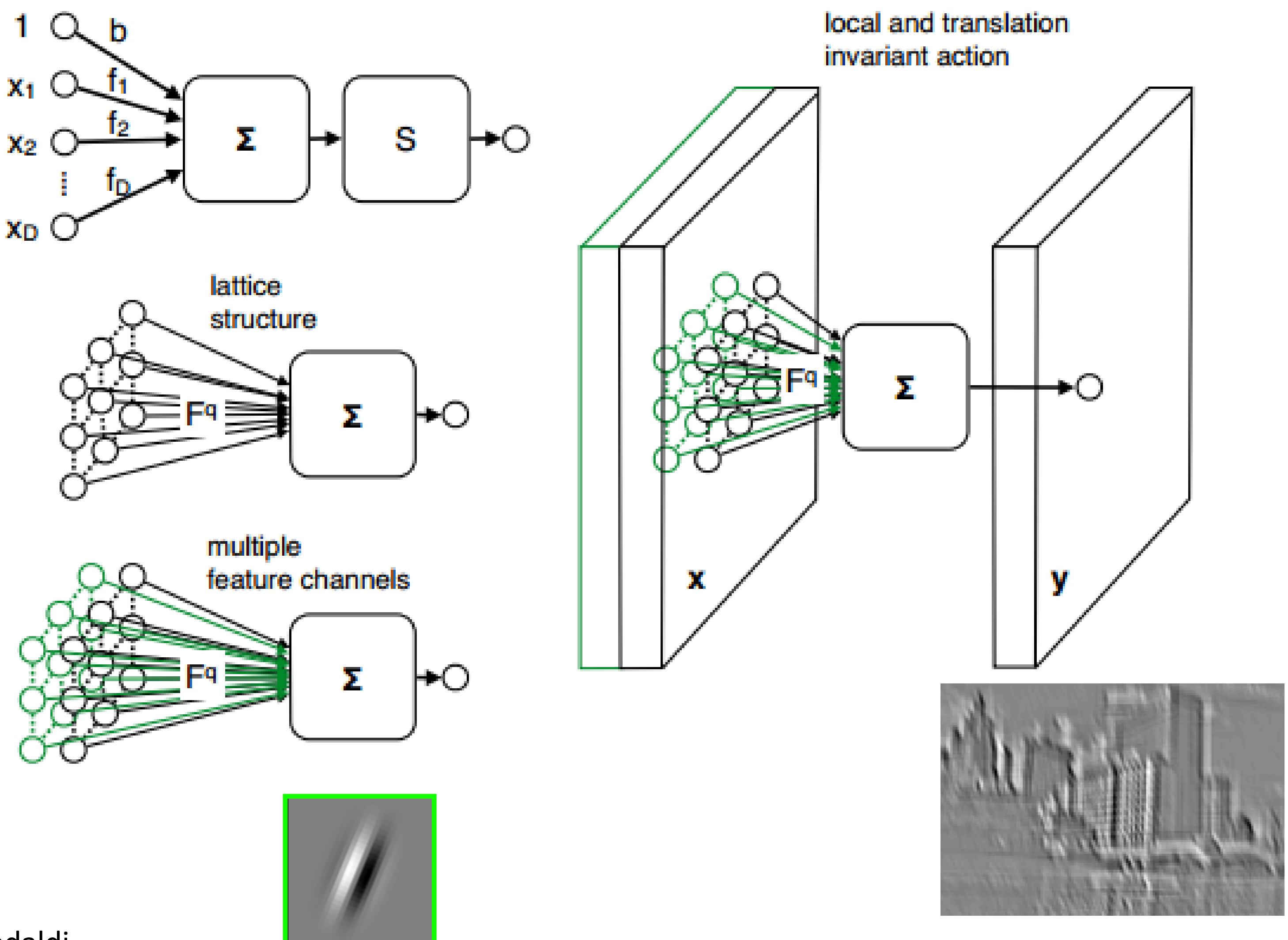
Input



Feature Map

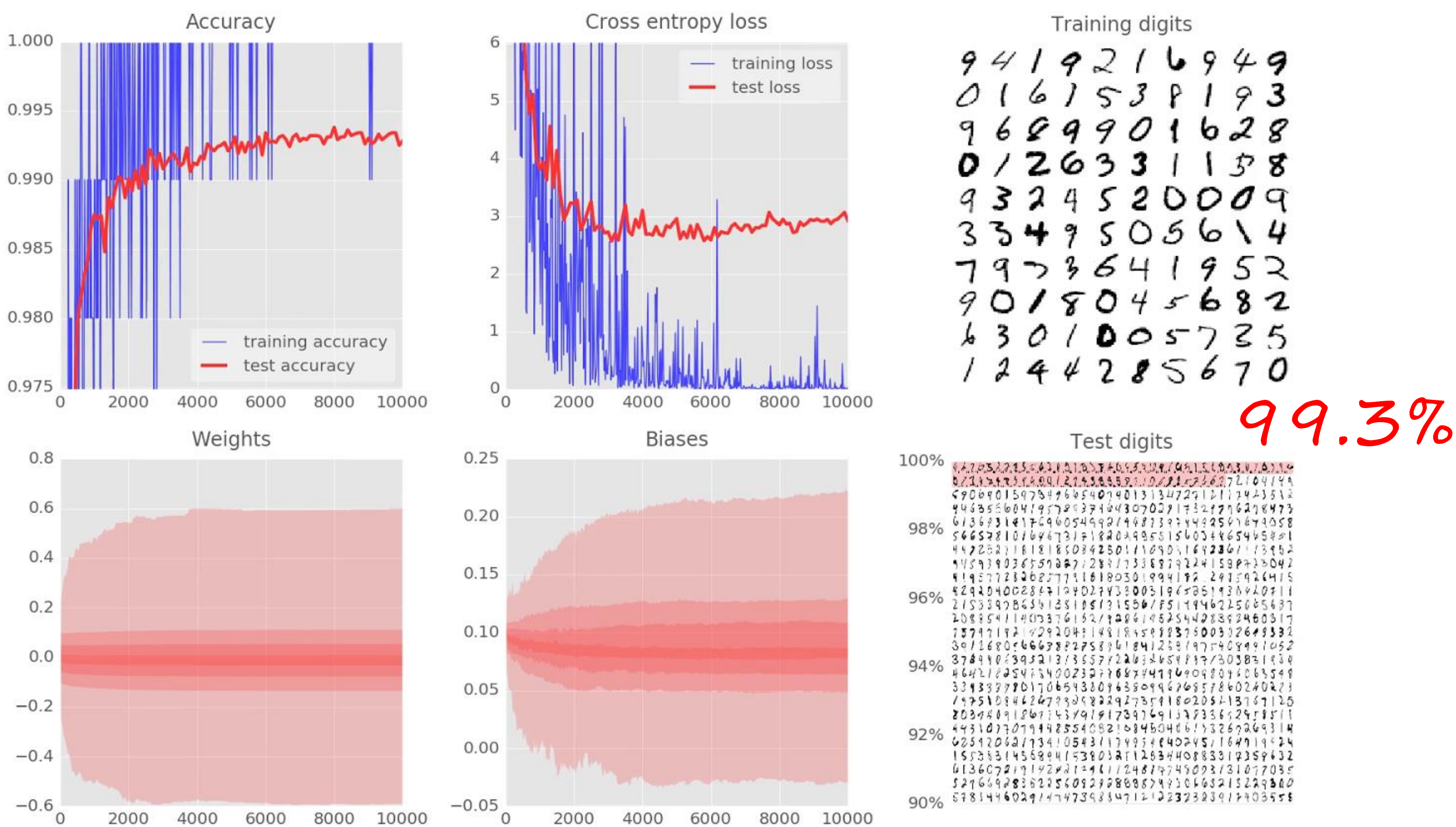
Slide credit: S. Lazebnik

Linear convolution as a neural network



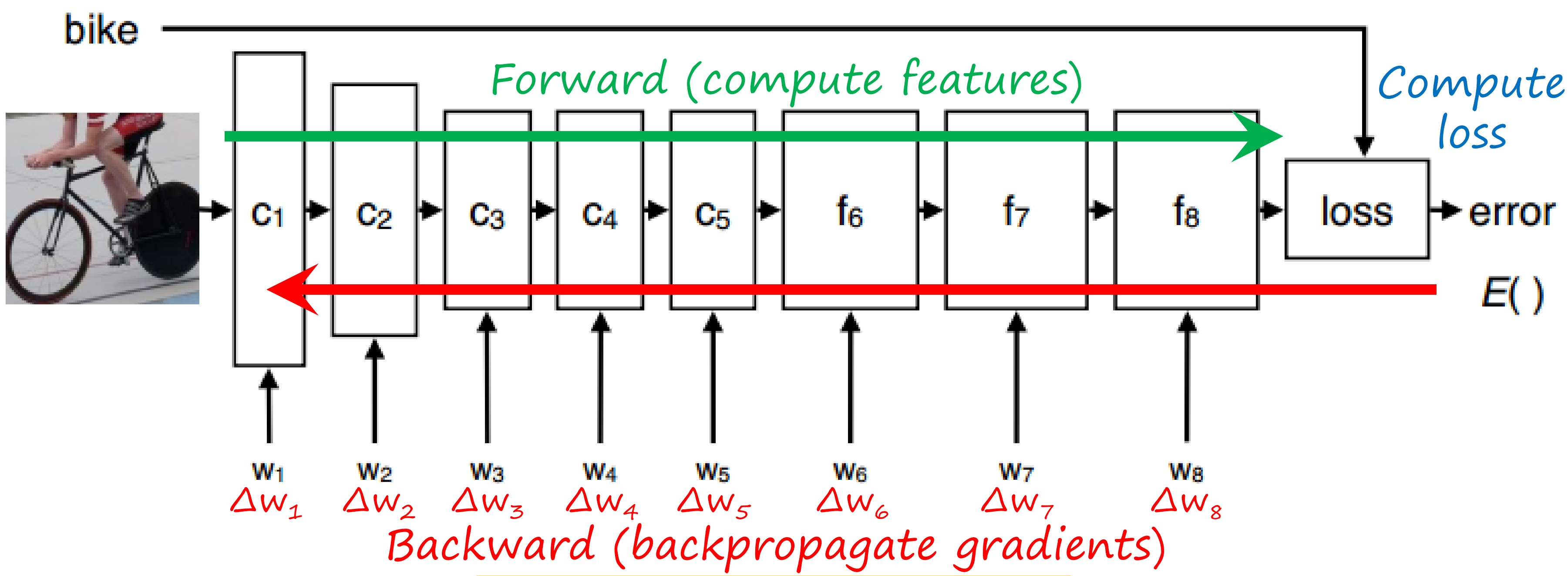
Slide credit: A. Vedaldi

Demo: convolutional network



https://github.com/martin-gorner/tensorflow-mnist-tutorial/blob/master/mnist_4.2_batchnorm_convolutional.py
https://www.youtube.com/watch?v=CzR7iqEo6ps

Training a CNN



$$\operatorname{argmin} E(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_8)$$

Update weights (SGD)

$w_i = w_i - \gamma \Delta w_i$

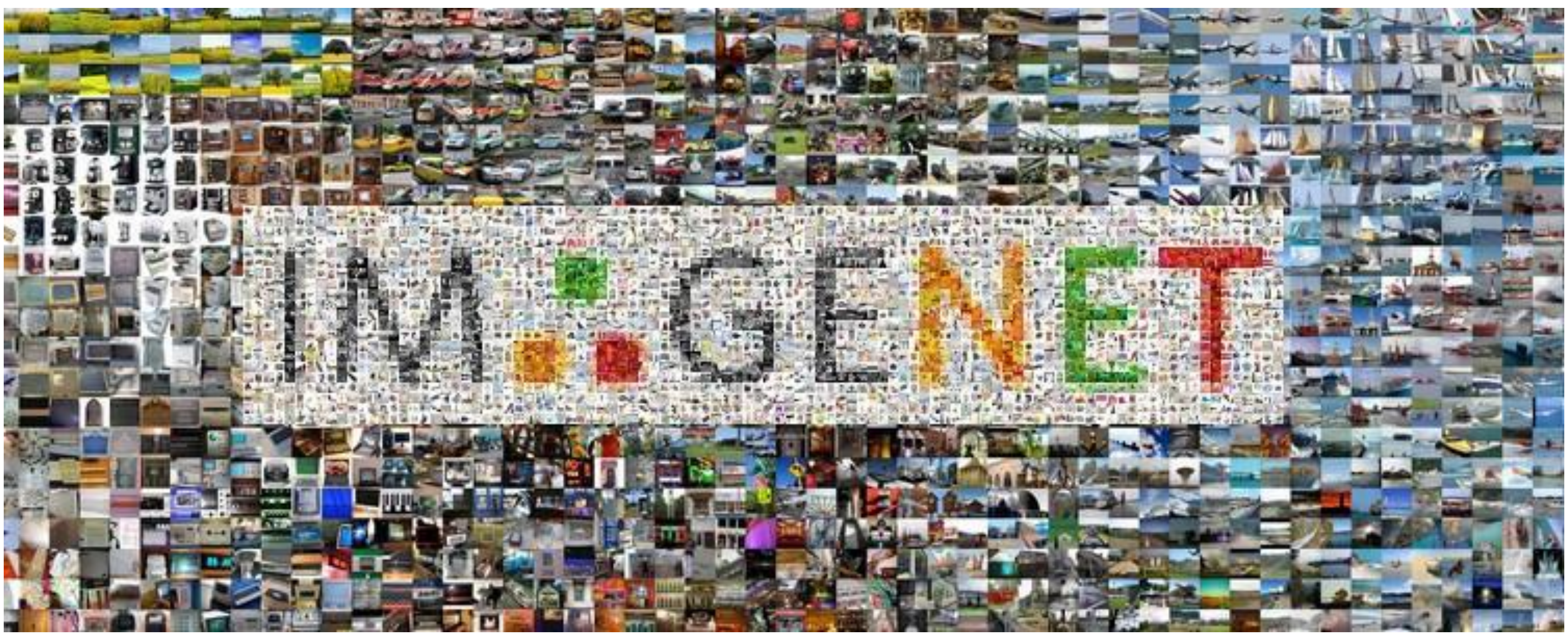
Stochastic gradient descent
(with momentum, dropout, ...)

Adapted from A. Vedaldi

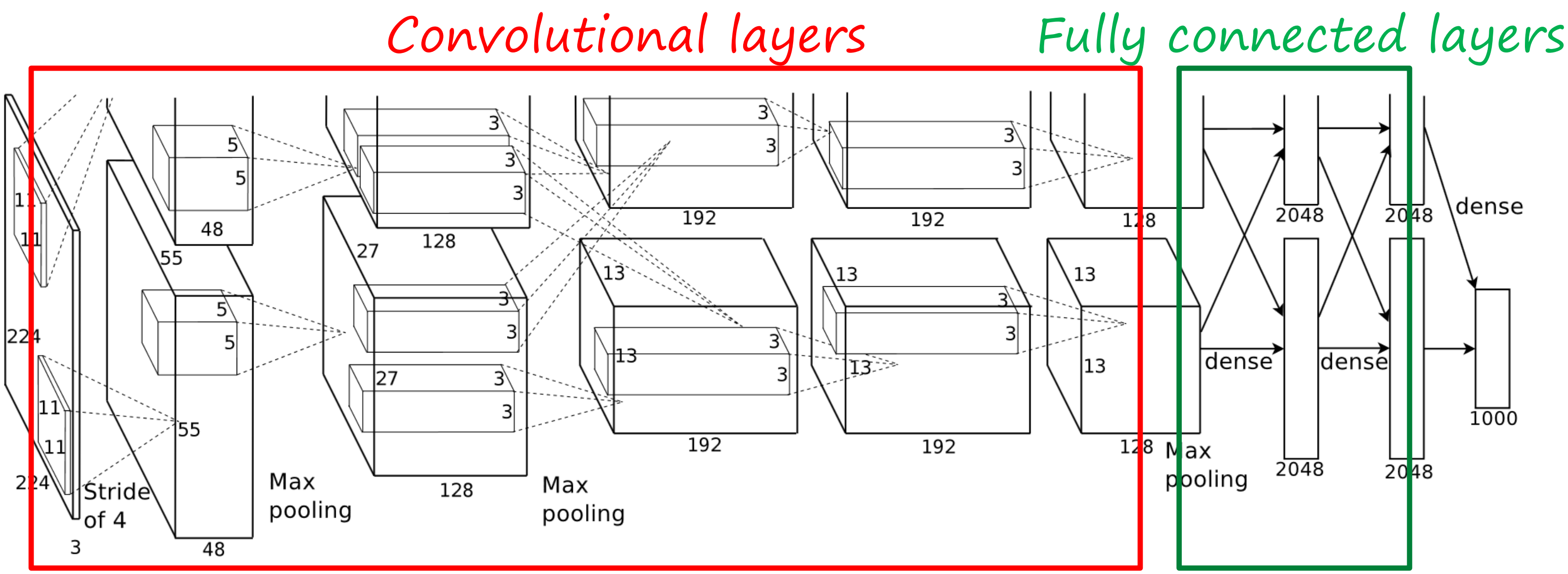
The revolution: ImageNet

Large visual database
- Over 10 million images
- Manually annotated
- Follows WordNet concept structure

ILSVRC2012
- Subset with 1000 classes
- 1.2 million images
- Used in competitions
- Used to train most of the networks



The revolution: AlexNet



- Khrizevsky et al winning ImageNet 2012. AlexNet:
  - 7-layers NN (5 conv layers+2 fully connected)
  - 650k neurons
  - 60 million parameters (630 million connections)
  - Trained on two GPUs for about a week

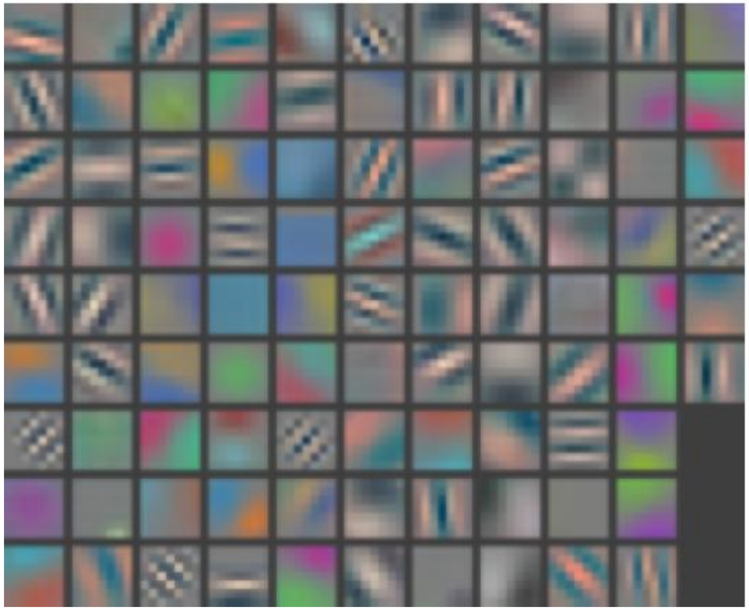
AlexNet in ImageNet ILSVRC 2012

Method	Classification (top 5 error, 1000 categories)	Classification and localization
Supervision (AlexNet CNN)	**16.4%**	**34.1%**
University of Tokyo	26.1%	53.6%
Oxford University Computer Vision Group	26.9%	50.0%
INRIA	27.0%	-
University of Amsterdam	29.5%	-

Shallow models

Understanding AlexNet: layer 1

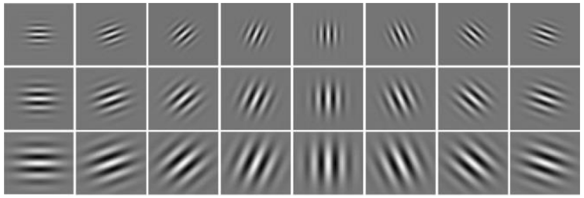
Learned from data!



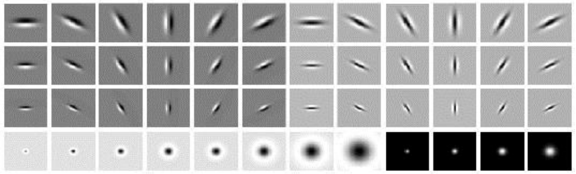
Layer 1 filters
(convolutional)

This looks familiar

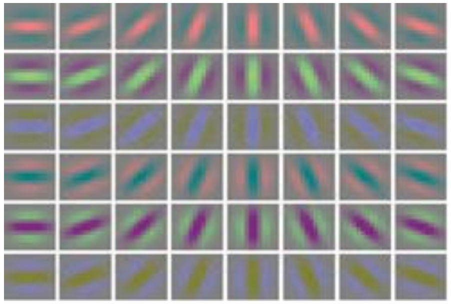
Handcrafted



Gabor filters



LM filters

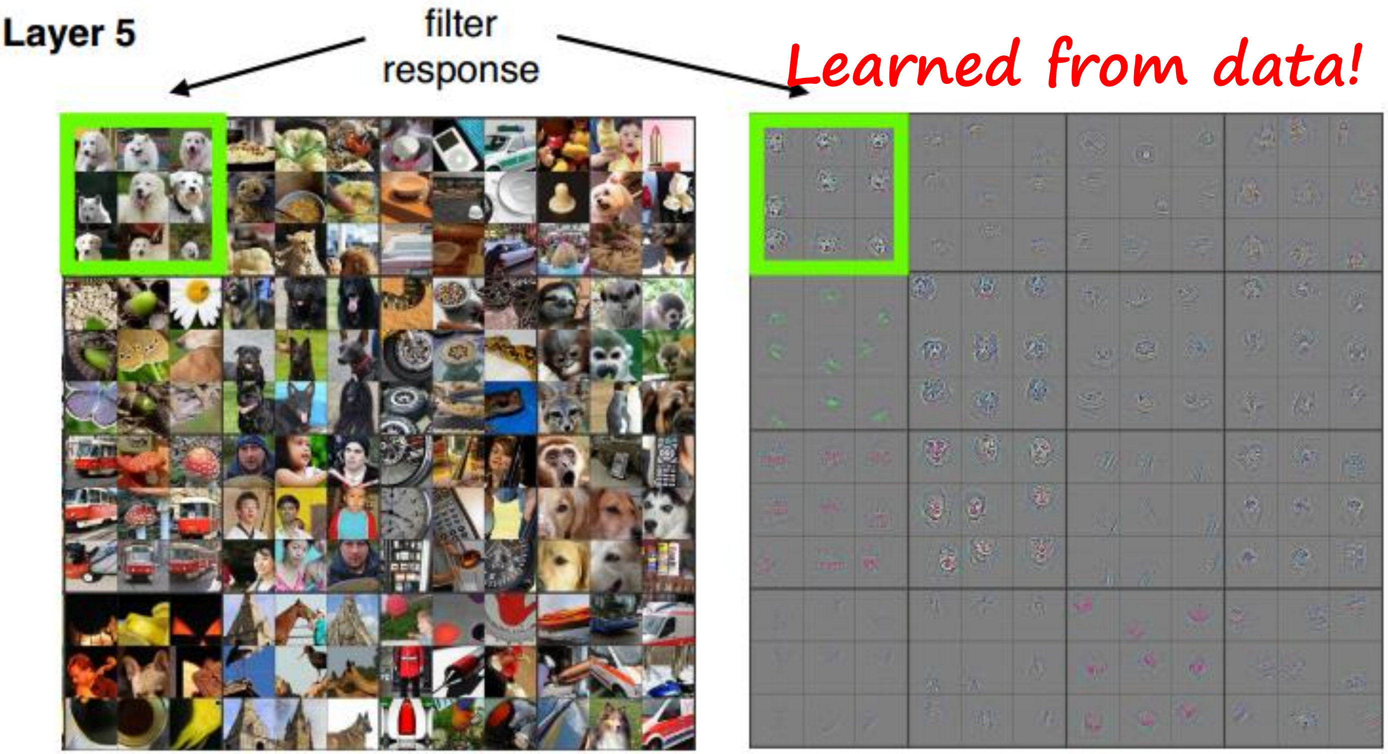


Color versions

Understanding AlexNet: layer 5

Visualize sample images that excite a given neuron the most

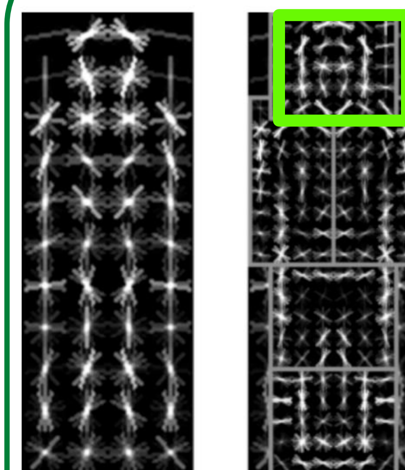
Learned from data!



top 9 exciting patches
for each neuron

their deconvnet
reprojection

Handcrafted
(partly learned)

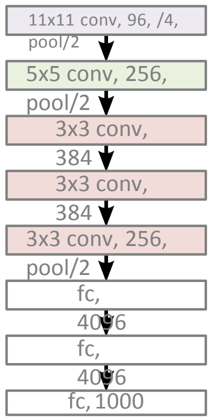


Deformable
Part Model

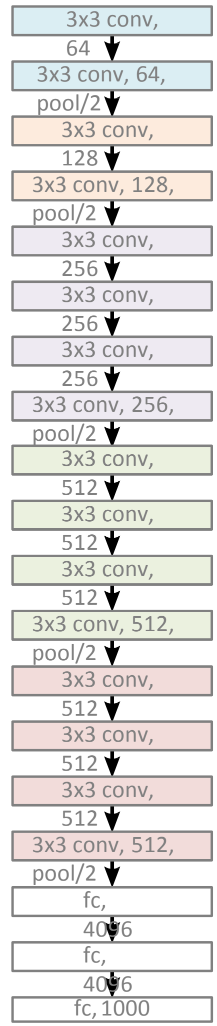
Adapted from: A. Vedaldi

Deeper architectures

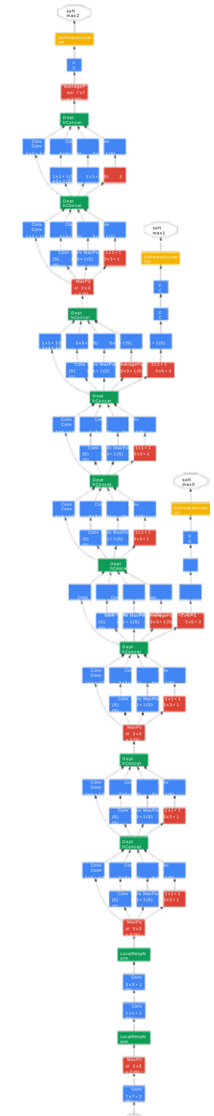
AlexNet, 8 layers (ILSVRC 2012)



VGG, 19 layers (ILSVRC 2014)



GoogleNet, 22 layers (ILSVRC 2014)

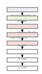


More layers, but smaller kernels

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.

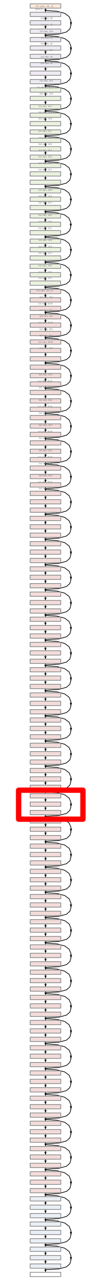
Ultra deep networks

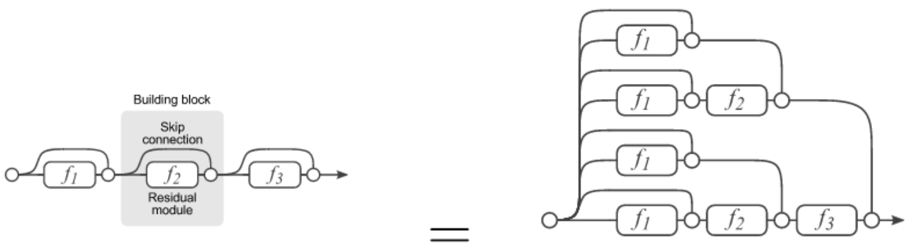
AlexNet
8 layers
(ILSVRC 2012)



VGG
19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)

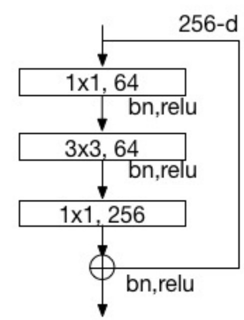




(a) Conventional 3-block residual network

(b) Unraveled view of (a)

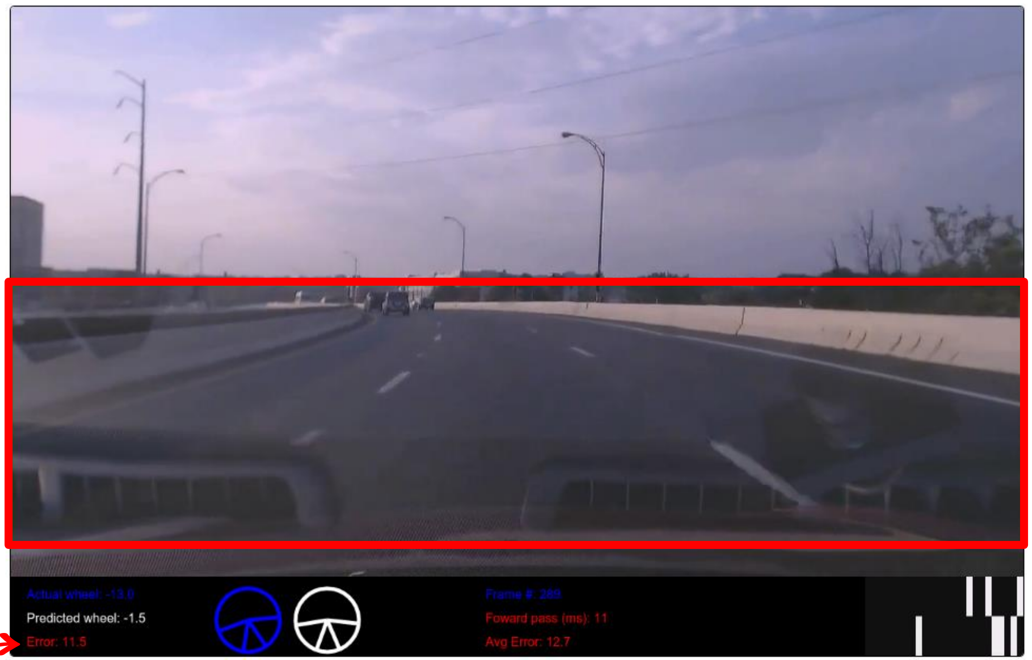
Interpretation as ensembles
of not-so-deep networks
(effective depth ≈ 20)



Residual block

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.

Demo: learning to drive



Goal: steer the driving wheel following the road

Regression problem

⇨ CNN ⇨ Angle

Loss= |Angle1 - Angle2|

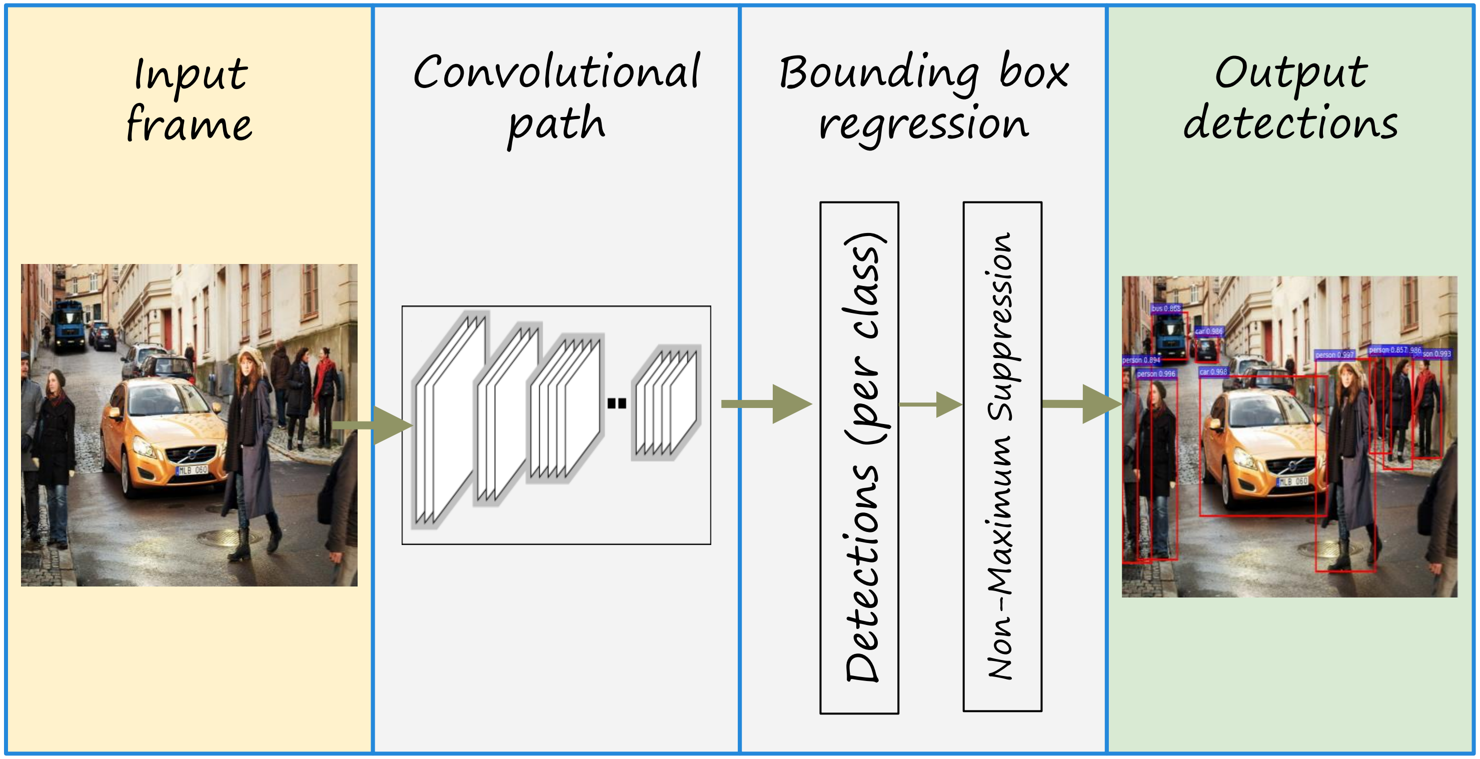
Angle2
(predicted by CNN)

Angle1
(human driver)

MIT Course on Deep Learning for Self-Driving Cars
http://selfdrivingcars.mit.edu/deepteslajs

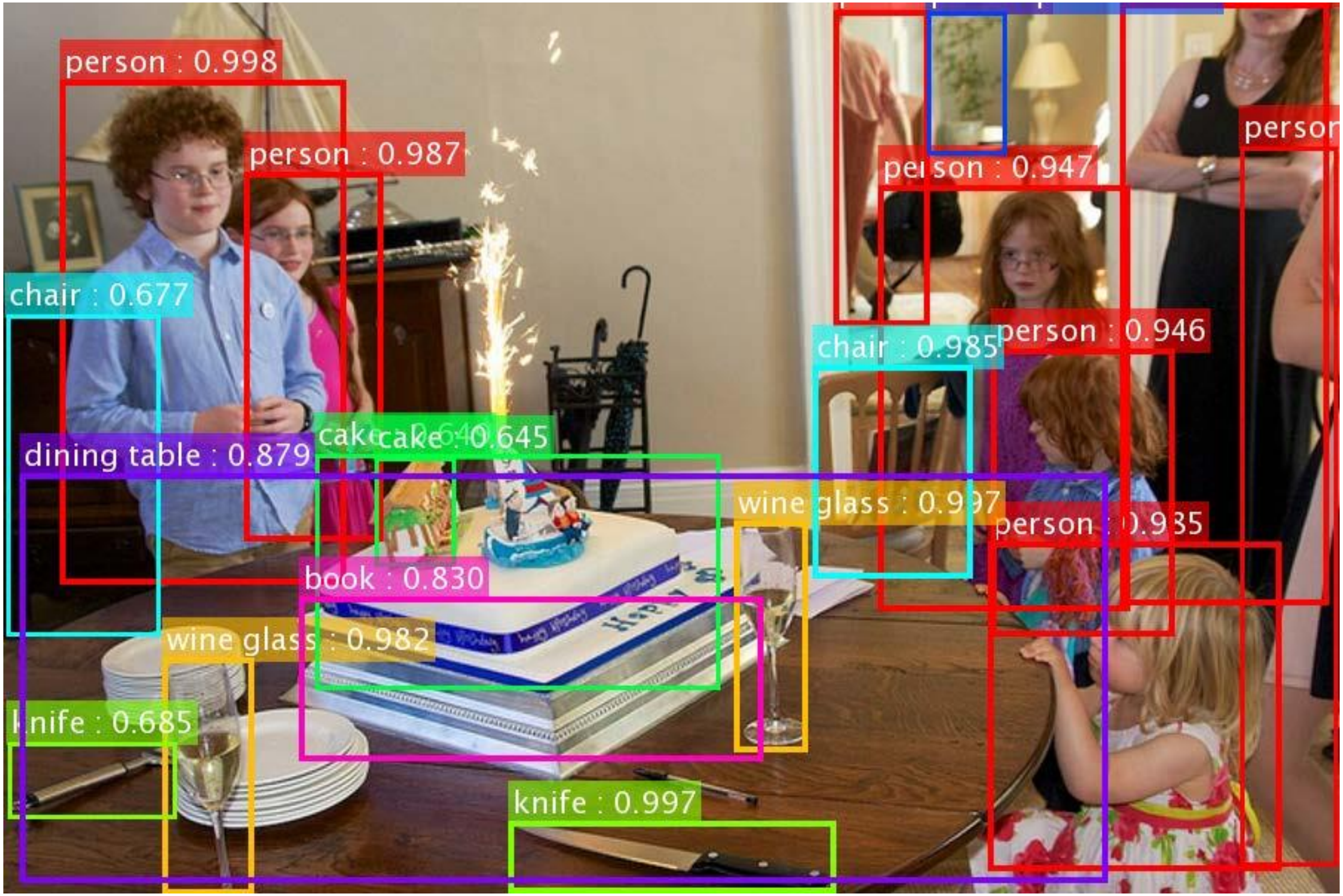
Object detection

Given an image detect the interesting objects (localization+class)



Slide credit: David Vázquez

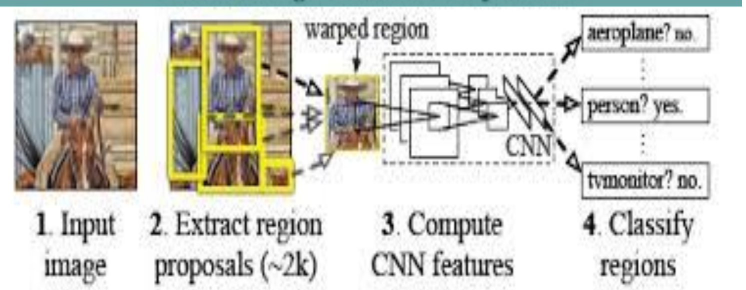
Object detection+classification



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.

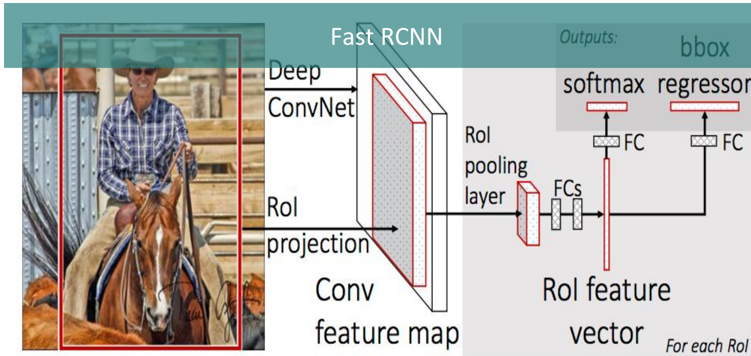
Object detection architectures

RCNN

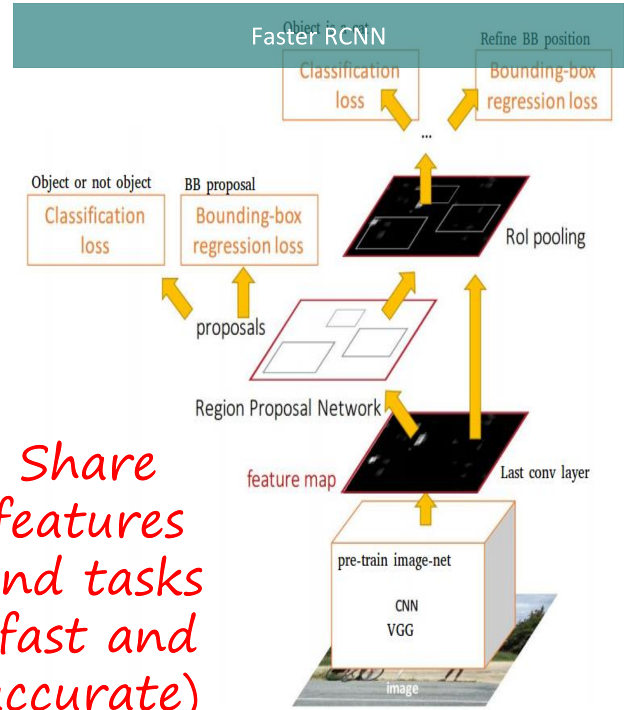


Very slow (each region separately)

Fast RCNN

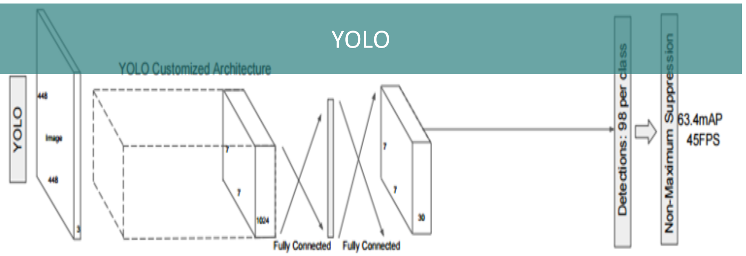


Faster RCNN

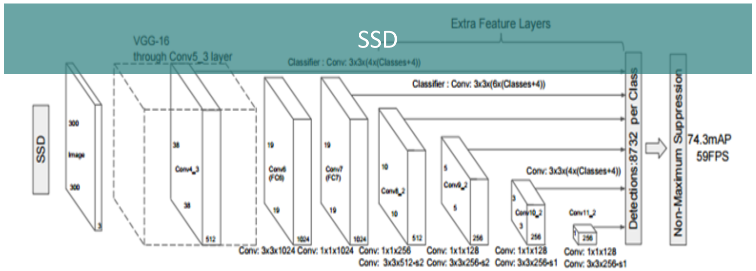


Share features and tasks (fast and accurate)

YOLO



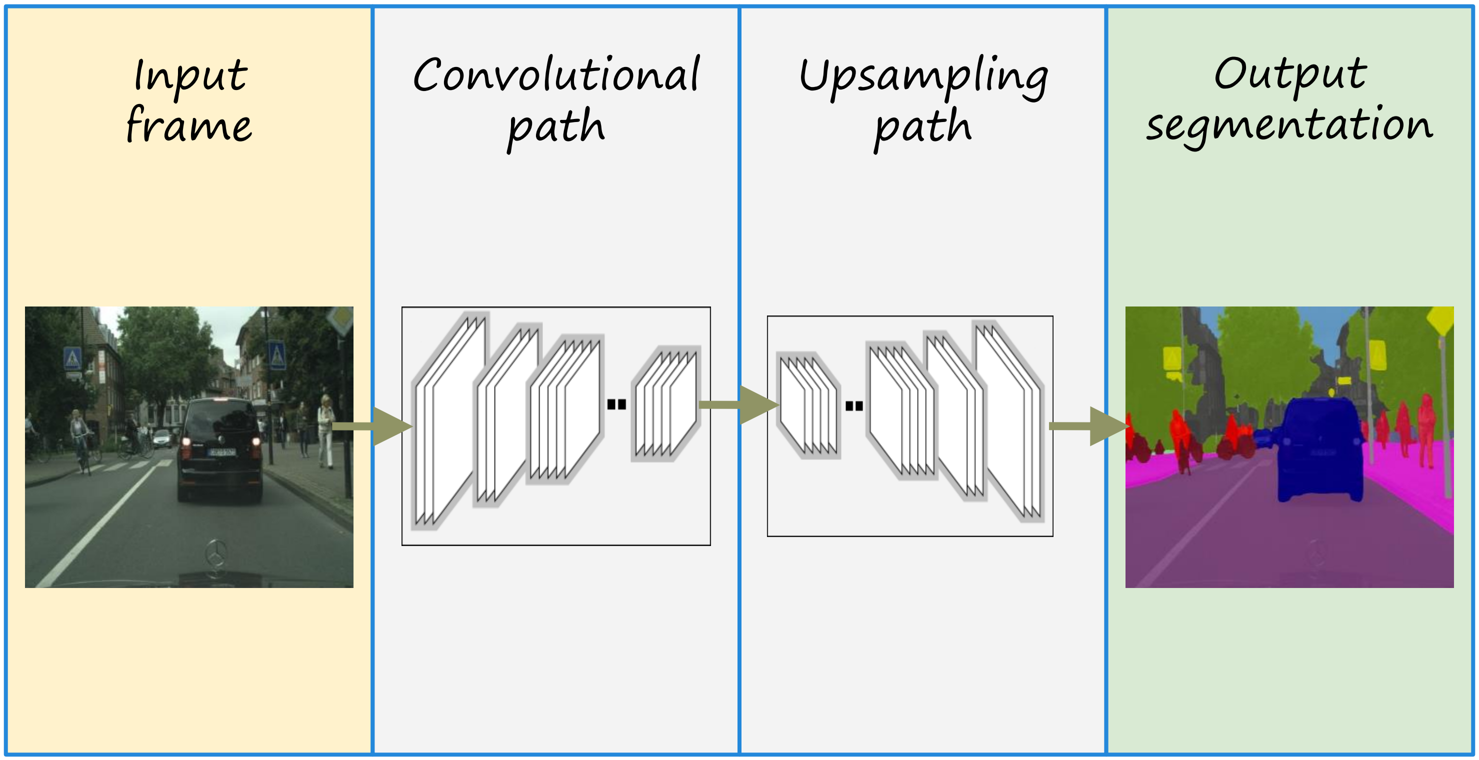
SSD



Slide credit: David Vázquez

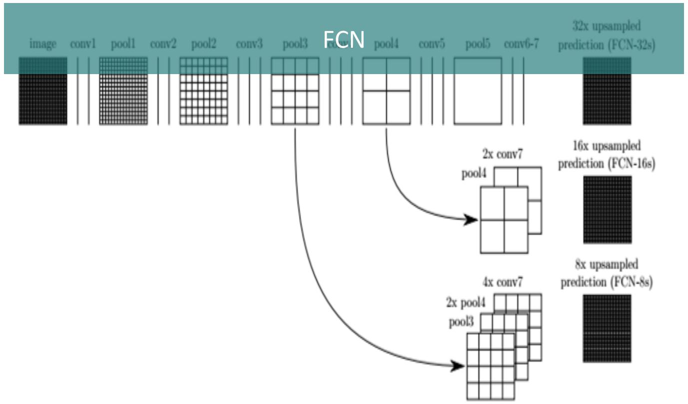
Image segmentation

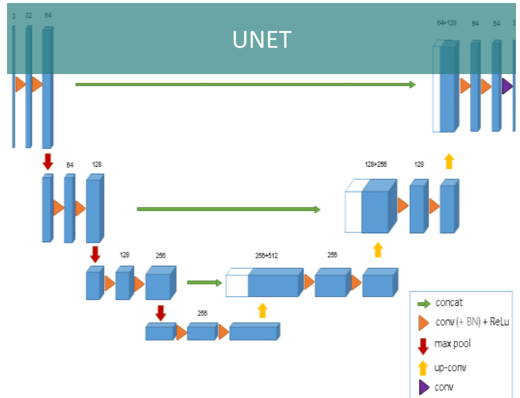
Given an image classify each pixel with its category

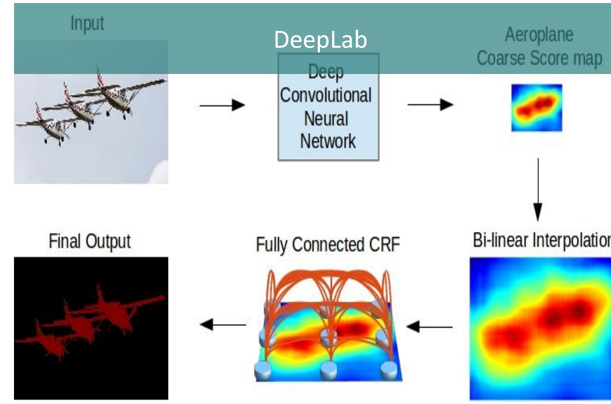


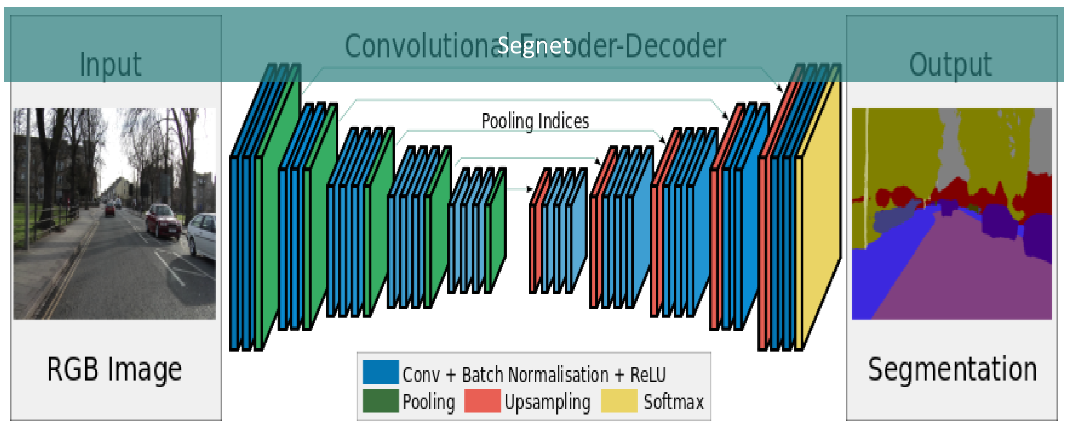
Slide credit: David Vázquez

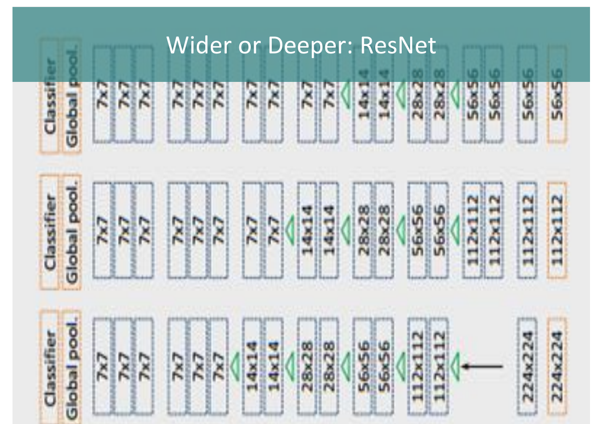
Image segmentation











Slide credit: David Vázquez

Today’s plan

- Machine learning
- Neural networks
- Deep learning in computer vision
- Fun stuff

Deep learning frameworks

Developed by: Berkeley
Type: Imperative
Base language: C++
Interfaces: C/C++, Python, MATLAB
Multi-GPU: Yes

Developed by: NYU
Type: Imperative
Base language: LUA
Interfaces: C/C++, Lua, Python
Multi-GPU: Yes

Developed by: UdM (MILA)
Type: Symbolic
Base language: Python
Interfaces: Python
Multi-GPU: No

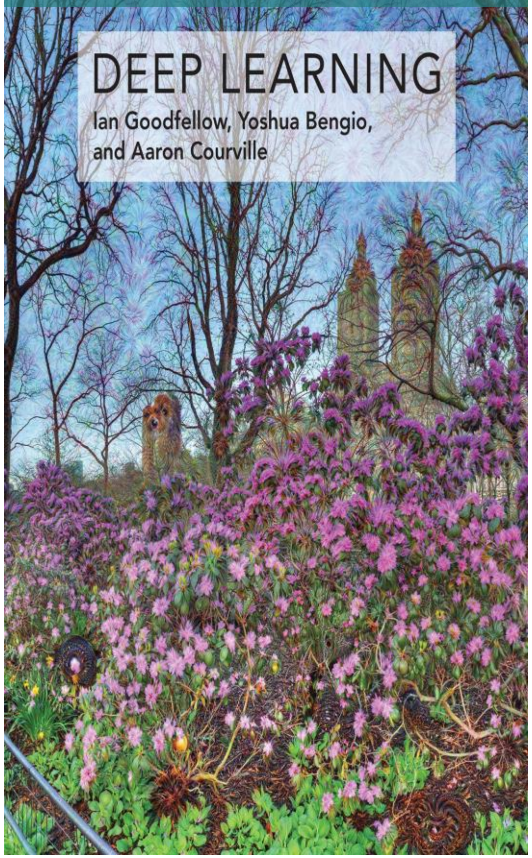
Developed by: Google
Type: Symbolic
Base language: C++
Interfaces: C++, Python
Multi-GPU: Yes

Developed by: UdM (MILA), Google
Type: Symbolic
Base language: Theano & TensorFlow
Interfaces: Python
Multi-GPU: Yes

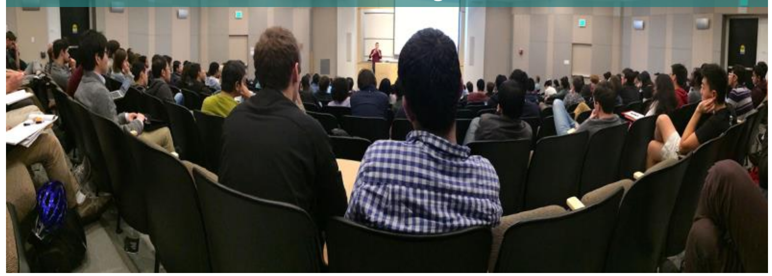
Slide credit: David Vázquez

Learning deep learning

Deep Learning Book

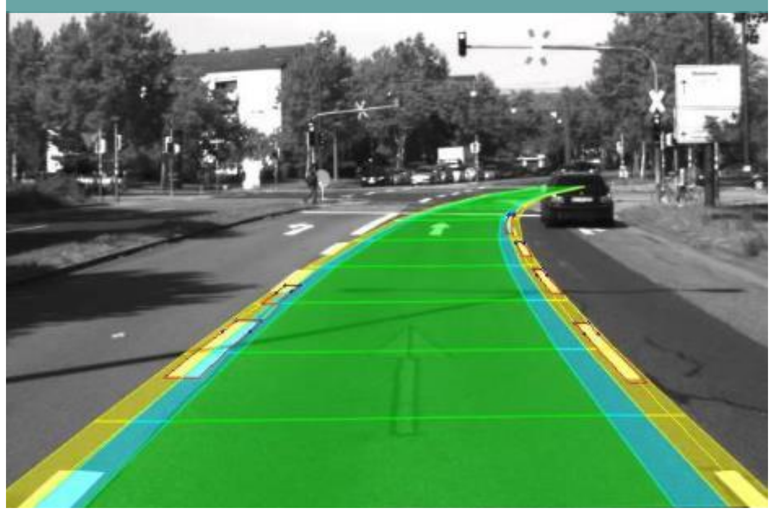


Stanford CS231n: Convolutional Neural Networks for Visual Recognition

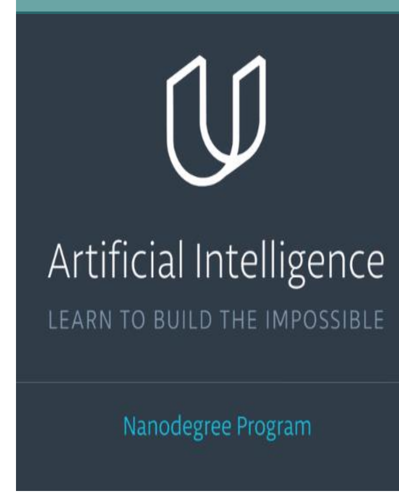


Udacity: Deep Learning

MIT 6.S094: Deep Learning for Self-Driving Cars

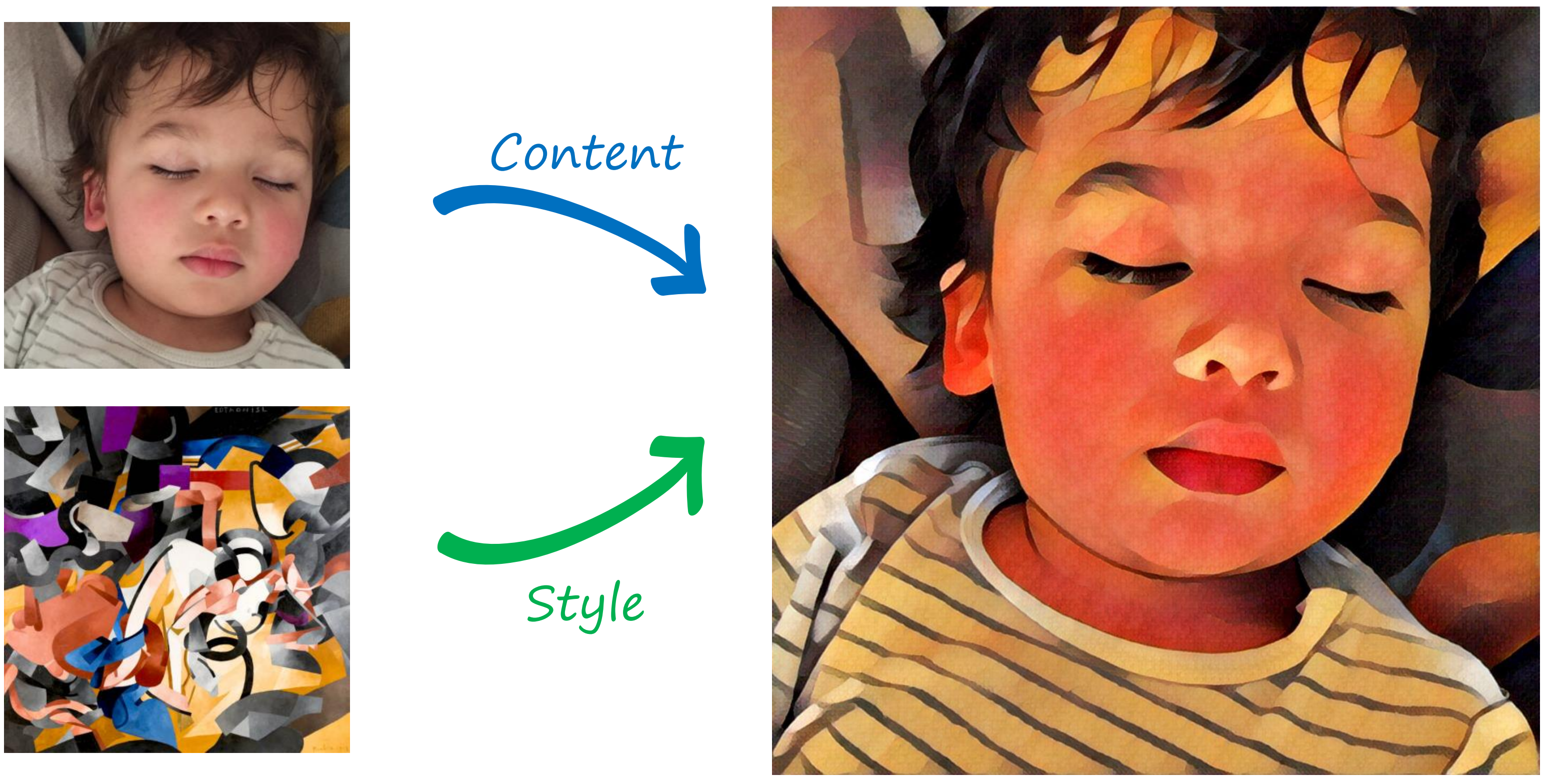


Udacity: Artificial Intelligence



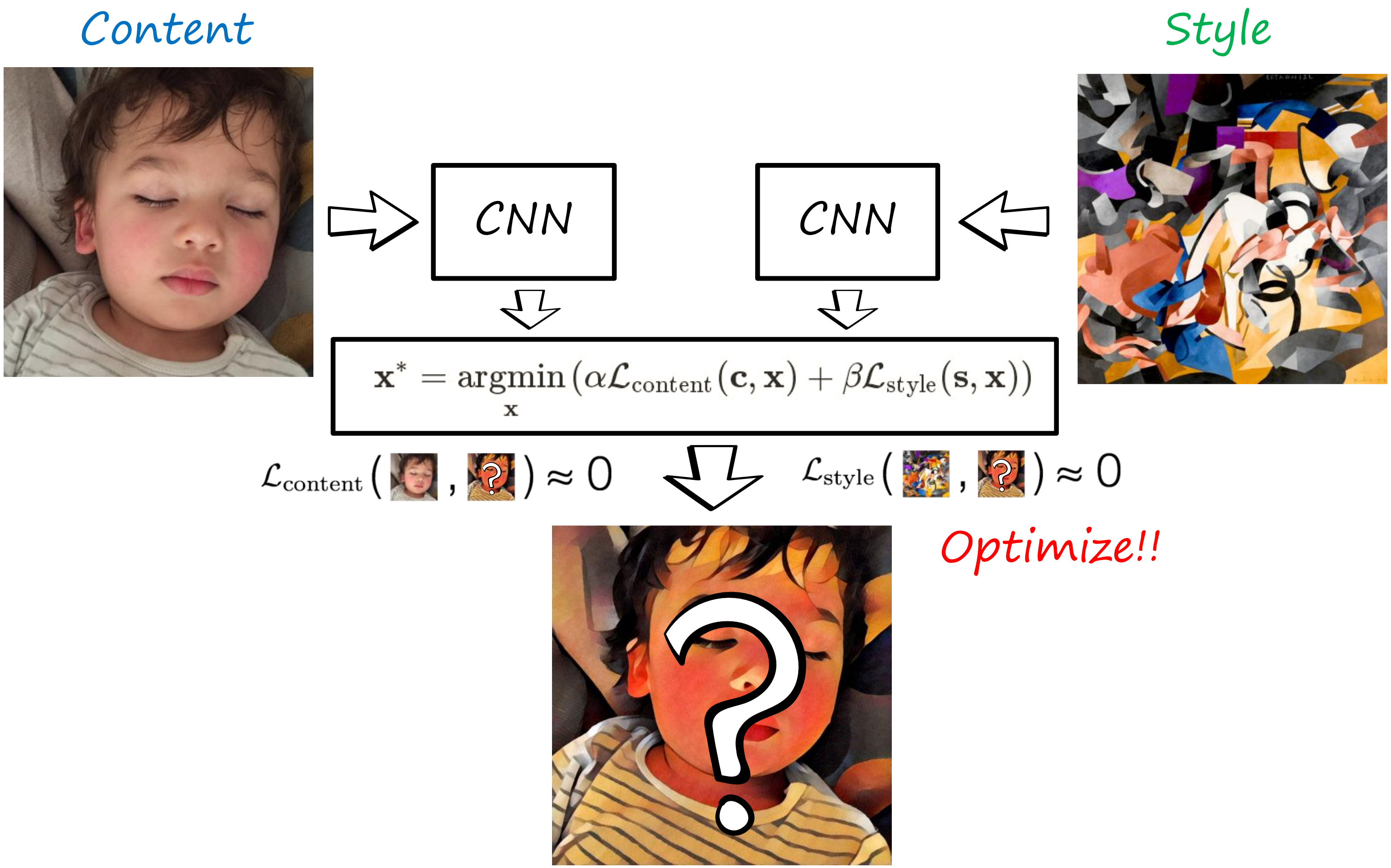
Partly adapted from: David Vázquez

Style transfer



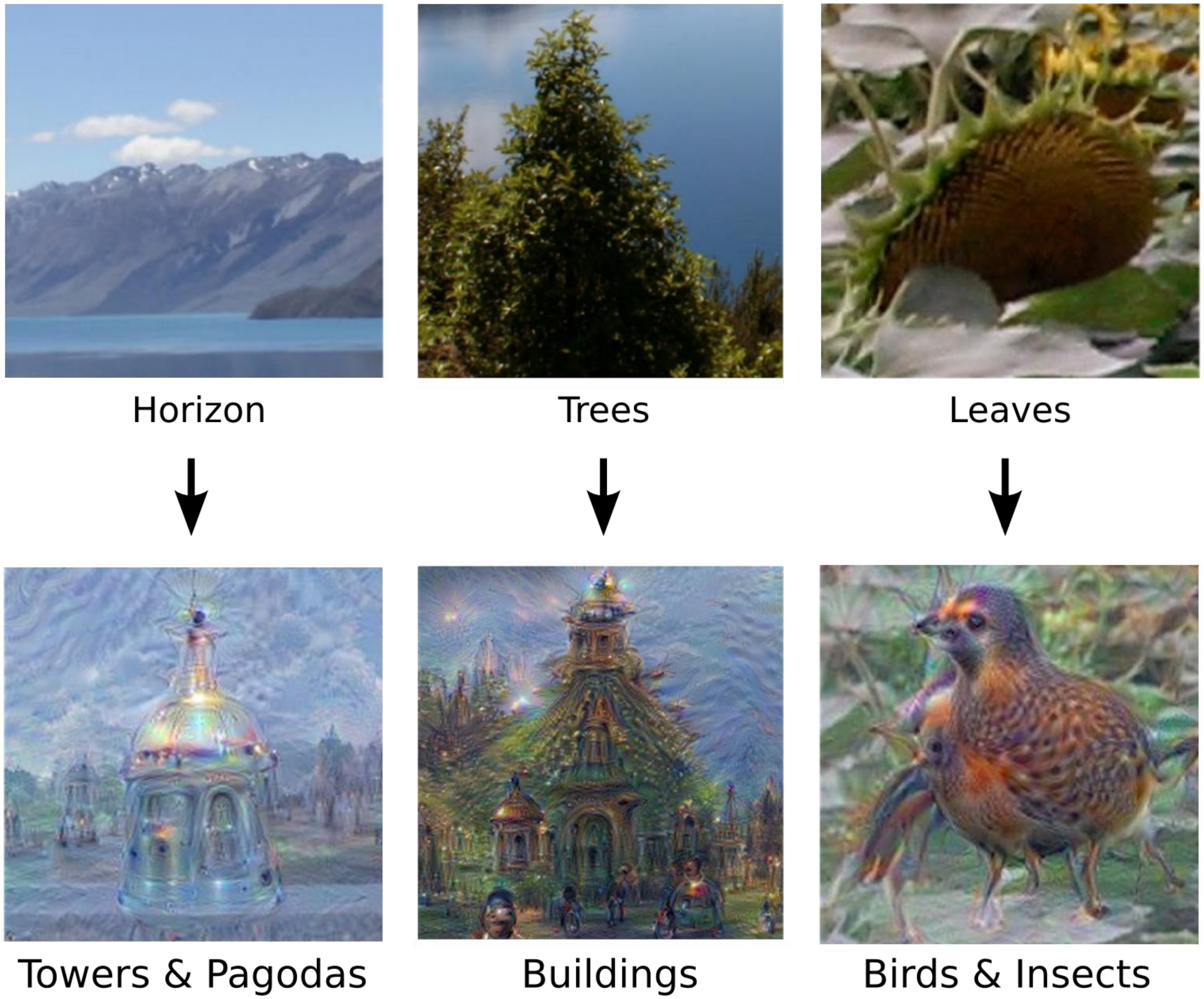
Gatys et al, A Neural Algorithm of Artistic Style, 2015
Images from https://harishnarayanan.org/writing/artistic-style-transfer

Style transfer



Gatys et al, A Neural Algorithm of Artistic Style, 2015
Images from https://harishnarayanan.org/writing/artistic-style-transfer

Deep dreams

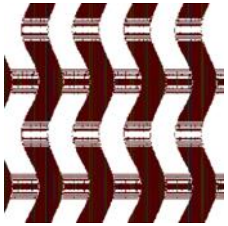
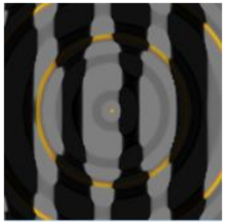


https://github.com/tensorflow/tensorflow/blob/master/tensorflow/examples/tutorials/deepdream/deepdream.ipynb

Deep dreams

Are deep networks so smart?



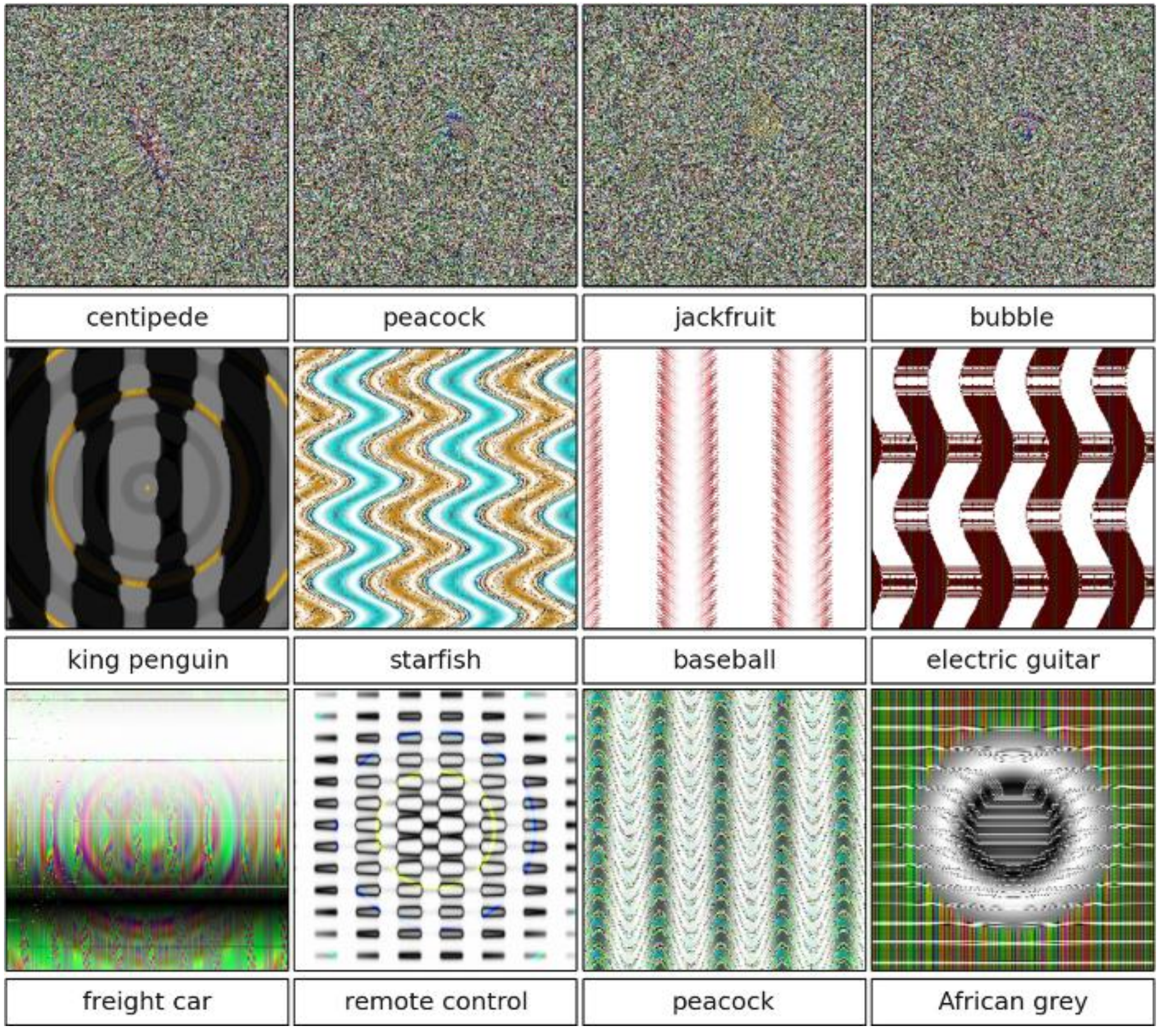
Convolutional neural network

Electric guitar (sure 99.9%)	Penguin (99.9%)	Electric guitar (99.9%)	Penguin (99.9%)

Nguyen et al, Deep Neural Networks are Easily Fooled, CVPR 2015

Are deep networks so smart?

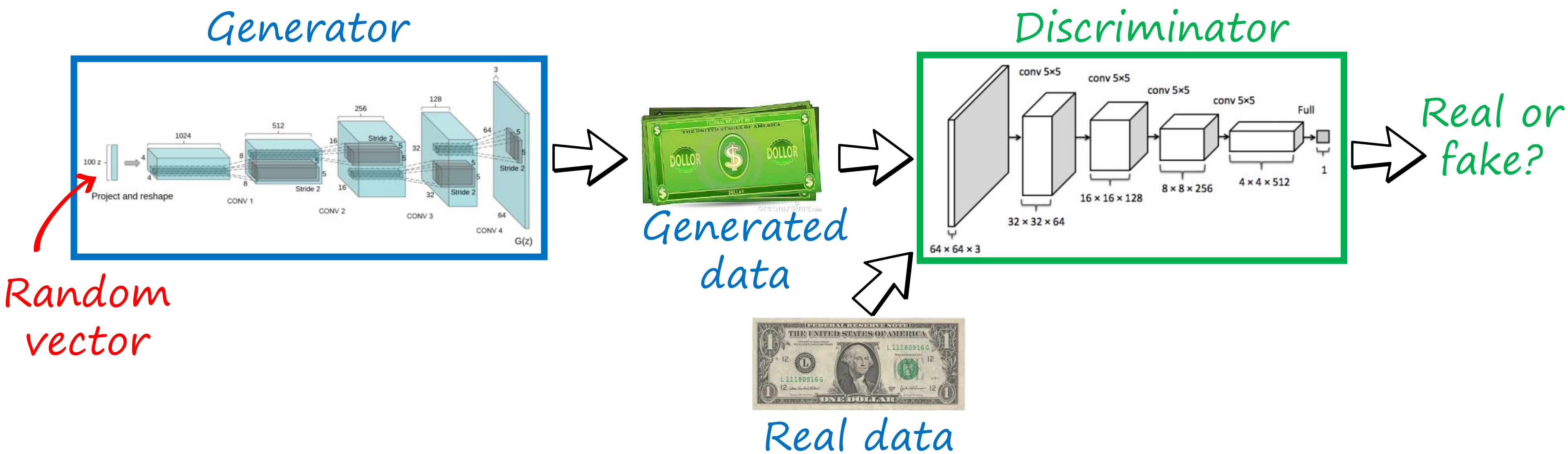
All 99.9% confidence!!



Nguyen et al, Deep Neural Networks are Easily Fooled, CVPR 2015

Networks that can imagine

Generative adversarial networks



The game (encouraged by the loss functions):

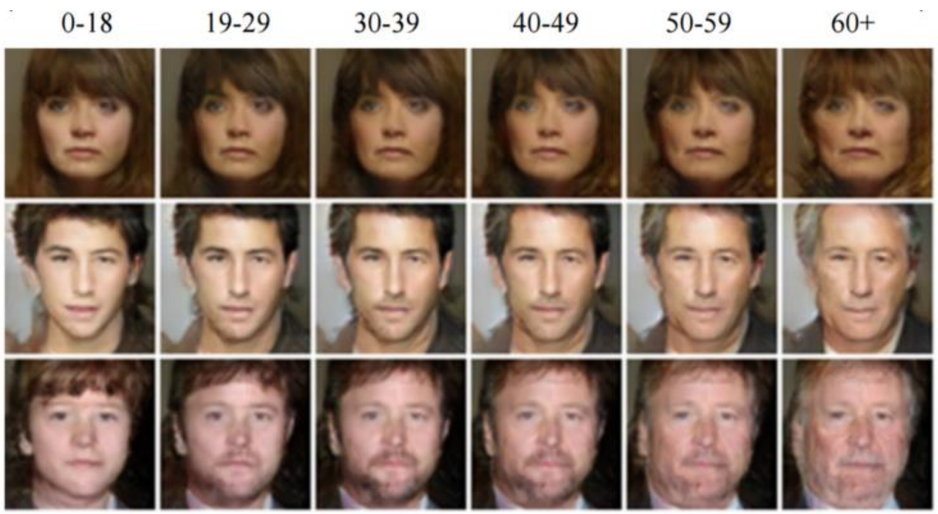
- The generator has to fool the discriminator (by generating more realistic money)
- The discriminator has to improve to detect fake money

(other approaches: autoencoders, PixelCNNs, …)

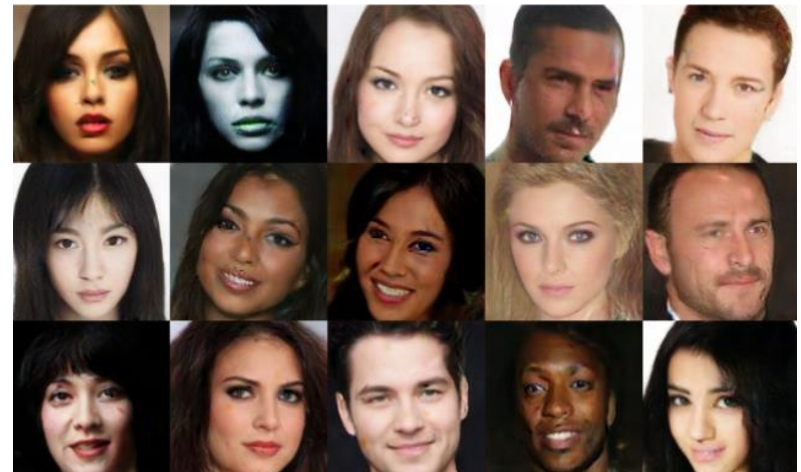
Goodfellow et al, Generative Adversarial Nets, 2014
Radford et al, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2016
A good explanation https://medium.com/@ageitgey/abusing-generative-adversarial-networks-to-make-8-bit-pixel-art-e45d9b96cee7

Generative adversarial networks

Simulate face aging



Generate faces



They are not real people
(don't exist in the real world!!!)

Interpolate faces

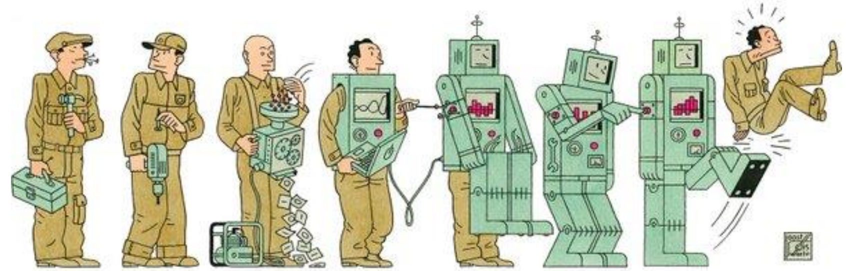
Berthelot et al, Boundary Equibilibrium Generative Adversarial Networks, 2017
Antipov et al, Face Aging with Conditional Generative Adversarial Networks, 2017

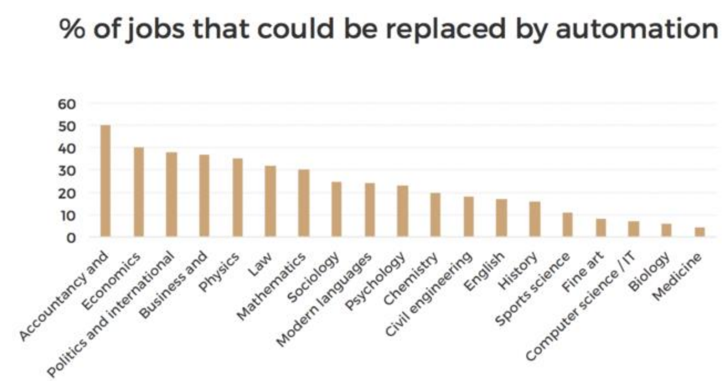
Today’s plan

- Machine learning
- Neural networks
- Deep learning in computer vision
- Fun stuff
- Some serious stuff

Responsability: DL is a powerful tool

- Social implications
  - Jobs
  - Security and privacy
  - Military use
  - Ethics and values





Frey & Osborne (2013) The future of employment: How susceptible are jobs to computerisation?

Existential risk and safe AI

- Serious concerns on AI's existencial risk
  - Other risks: climate change, nuclear holocaust, misuse of nanotech., …
- Action lines for safe AI
  - Open access to code, data and resources. Transparency in general
  - Education in machine learning
  - Differential privacy
  - Regulation
  - Human values in AI
- Some people take this very seriously
  - Future of Life Institute
    - Beneficial AI conference https://futureoflife.org/bai-2017
    - AI principles: https://futureoflife.org/ai-principles
  - Future of Humanity Institute, University of Oxford
  - Centre for the Study of Existential Risk, University of Cambridge

Learning and Machine Perception (LAMP) team

http://www.cvc.uab.es/lamp

Luis Herranz Arribas
Postdoctoral researcher
lherranz@cvc.uab.es
http://www.lherranz.org

Computer Vision Center
Edifici O, Campus UAB
08193 Bellaterra, Barcelona
http://www.cvc.uab.es