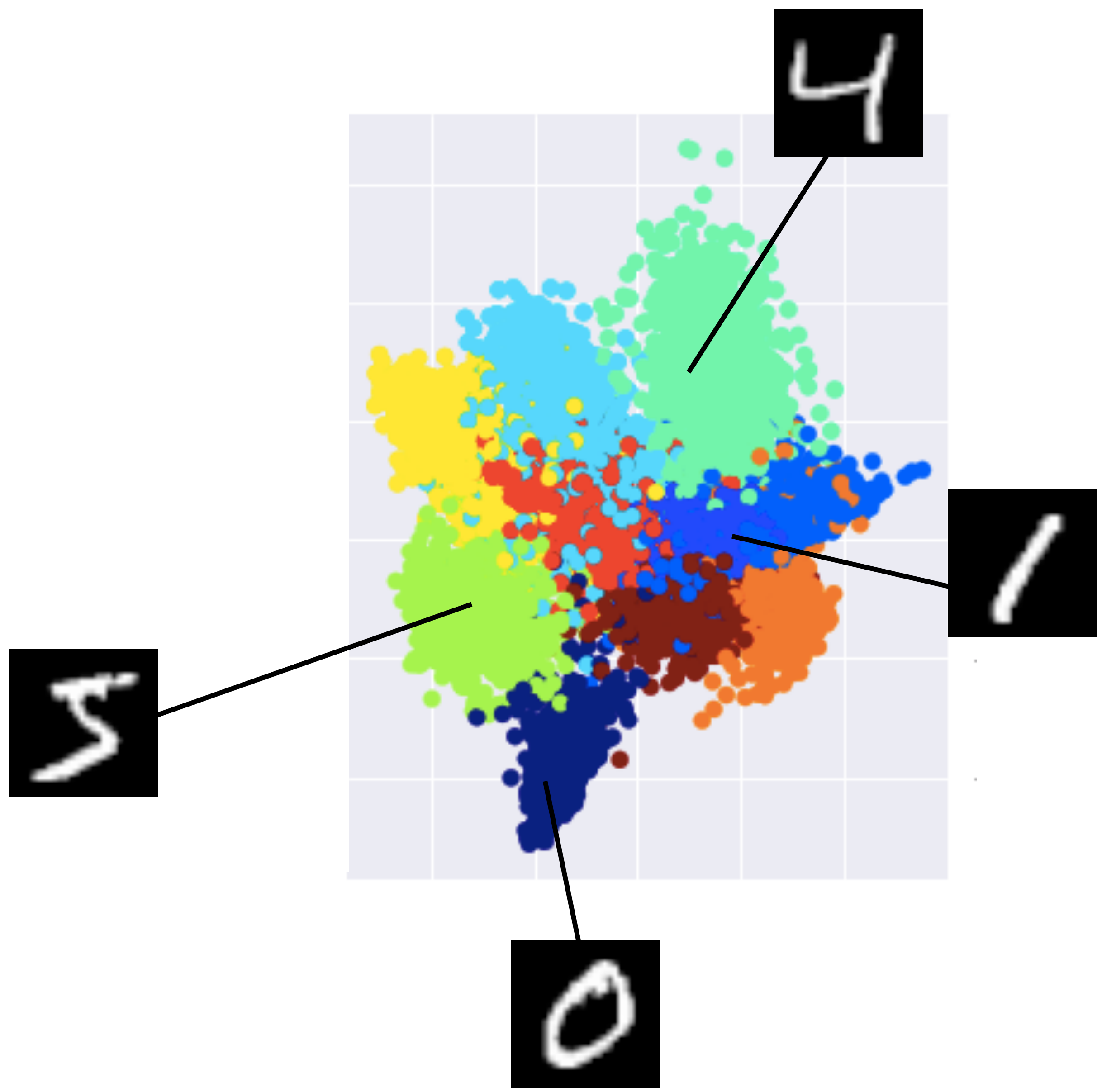
Deep Neural Networks

Alexandre Dauphin



Outline

1. Introduction

2. Fully Connected neural networks

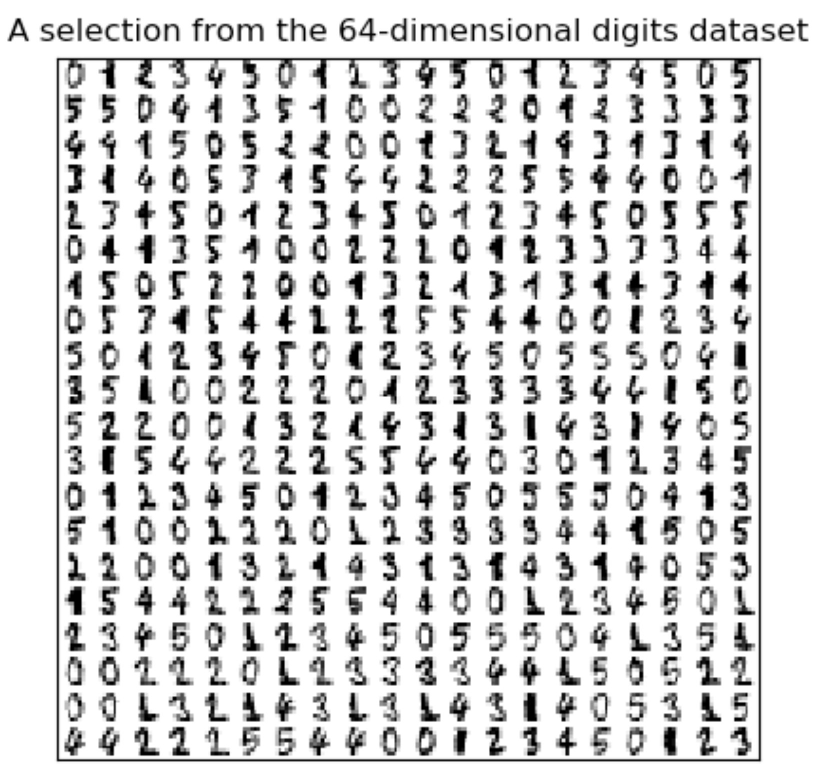
3. Unsupervised methods

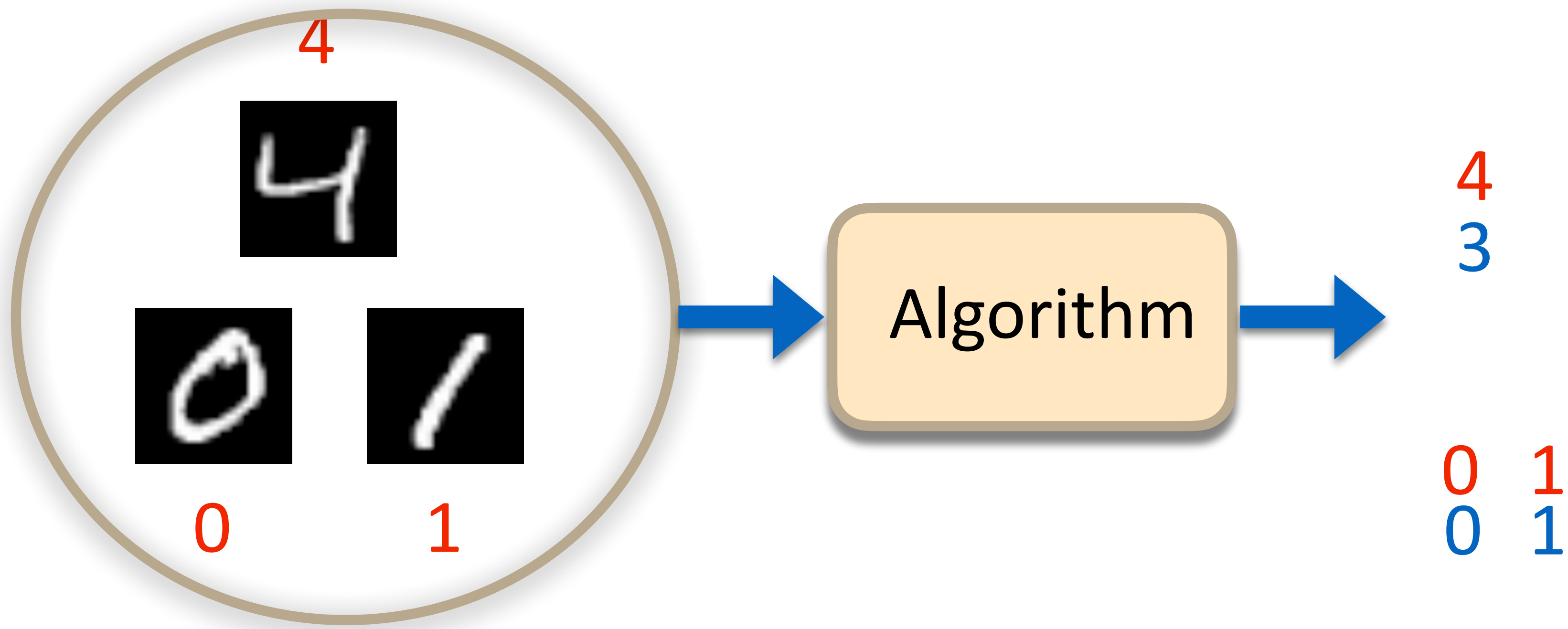
4. Convolutional neural networks

1. Introduction

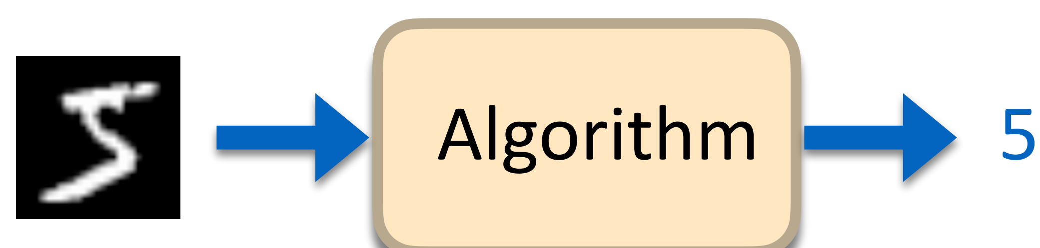
Supervised Learning

- Source data are labelled
- Predict unlabelled data





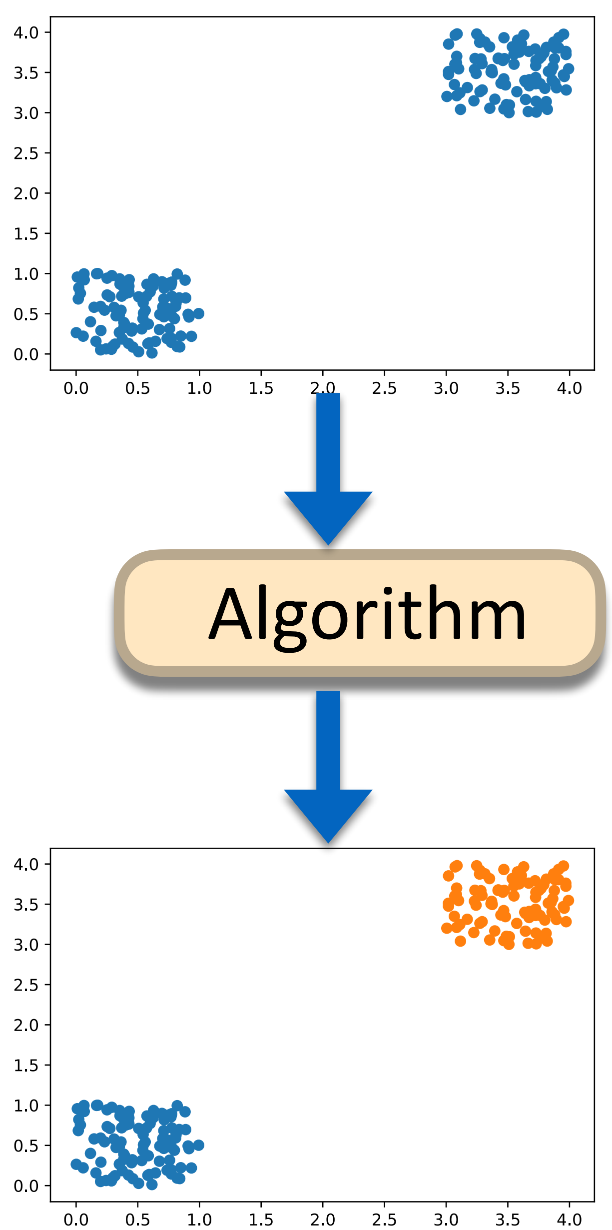
After training

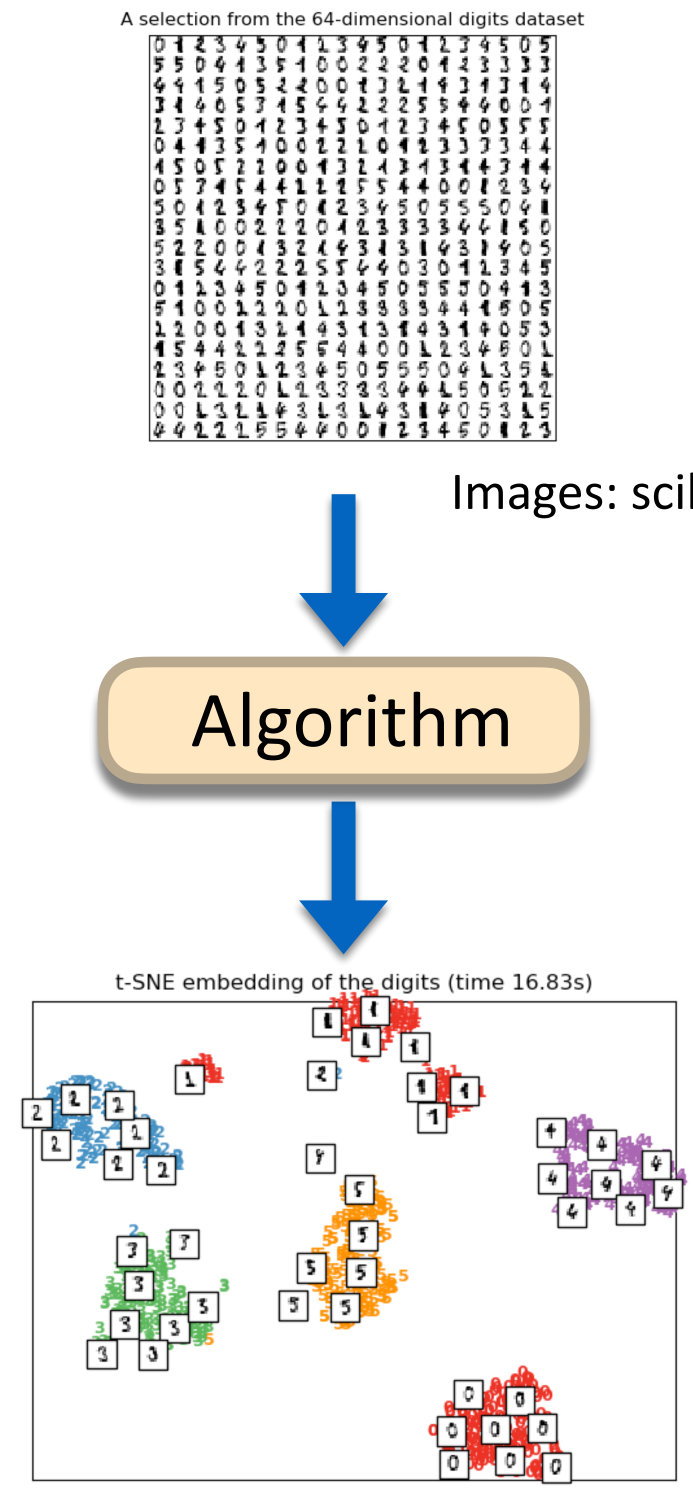


Images: scikit learn

Unsupervised Learning

- Unlabeled data
- try to find some structure in the data





Images: scikit learn

Supervised vs Unsupervised in this course

- Deep neural networks and deep convolutional neural networks (Supervised)
- Principal component analysis (Unsupervised)
- K-means (Unsupervised)

If we have time:

- Auto encoders (Unsupervised)

Goals of the course

- Introduction to deep learning
- Introduction to Fastai (deep learning library)
- Introduction to convolutional neural networks
- Analysis of the feature space
- How to prevent overfitting?
- Transfer learning

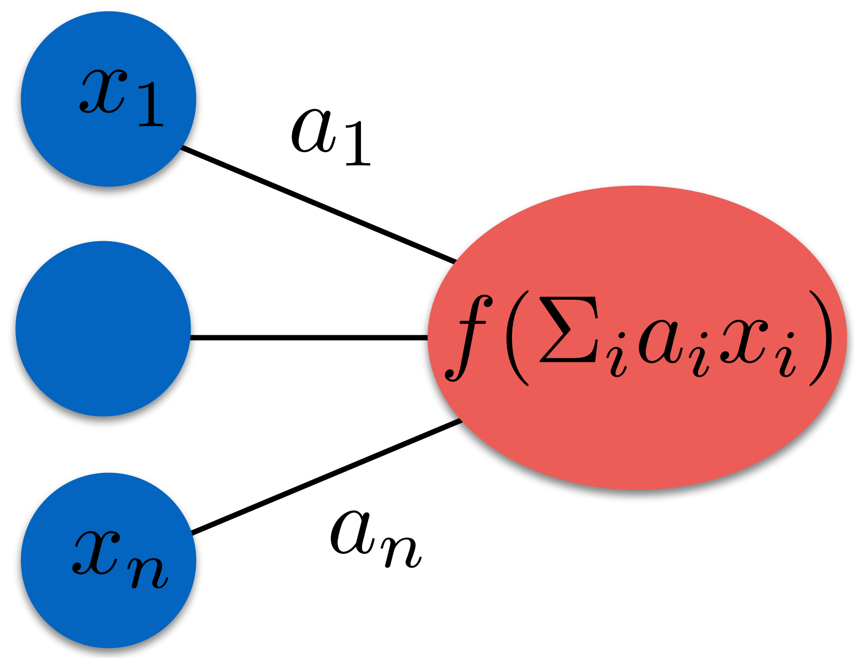
2. Fully connected neural networks

Neural Networks

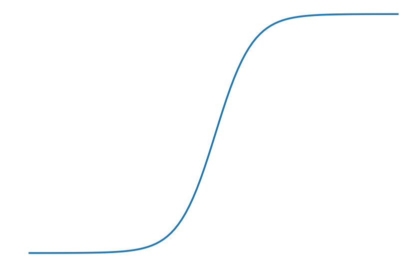
Data Set

$$\{(X, y)\}$$

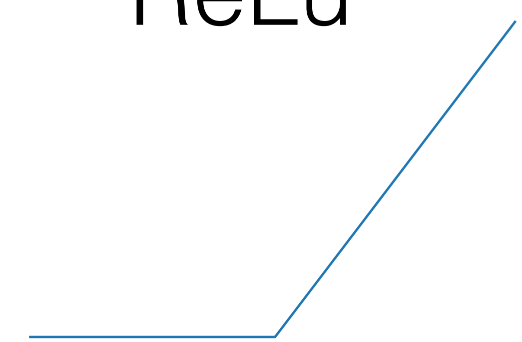
Input element   Label



sigmoid

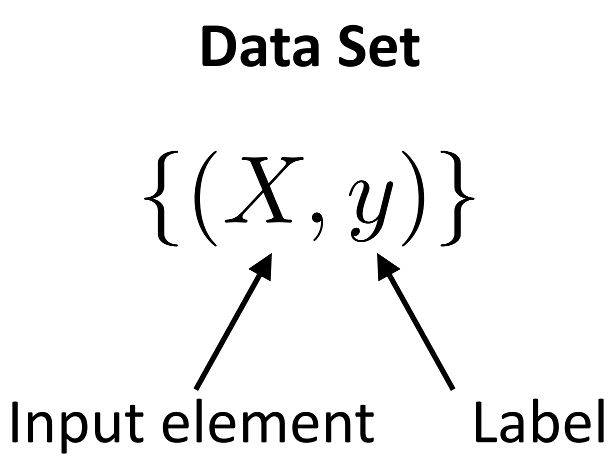


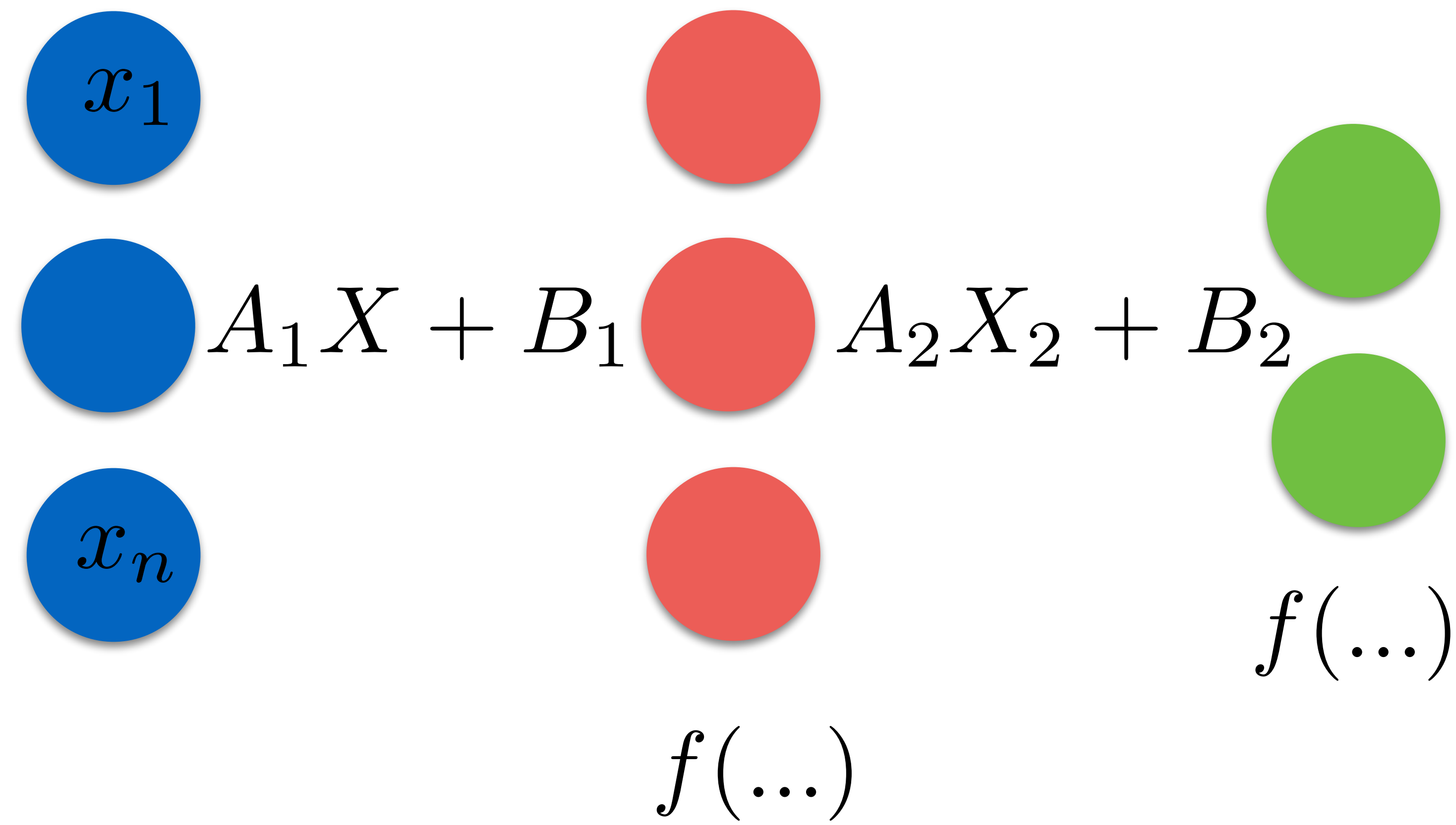
ReLu



- f is an activation function.
- if f(x)=Θ(x) (Heaviside), then one recovers the perceptron.
- The choice of the activation function depends on the problem.
- The activation function allows onto have nonlinearities.
- In modern neural networks, the most used activations functions are the sigmoid and ReLu.

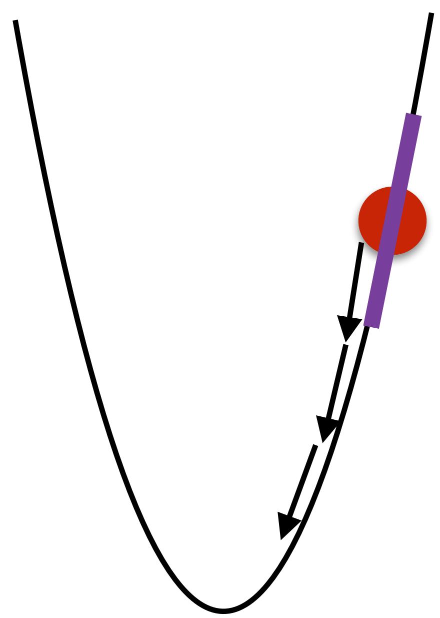
Neural Networks





- Cost function: $C = \frac{1}{N}\Sigma_j (y_{\mathrm{pred},j} - y_j)^2$
- Goal: minimize the cost function given the training set.
- Adjust the weights A_i and B_i to minimize the cost function.

Gradient Descent



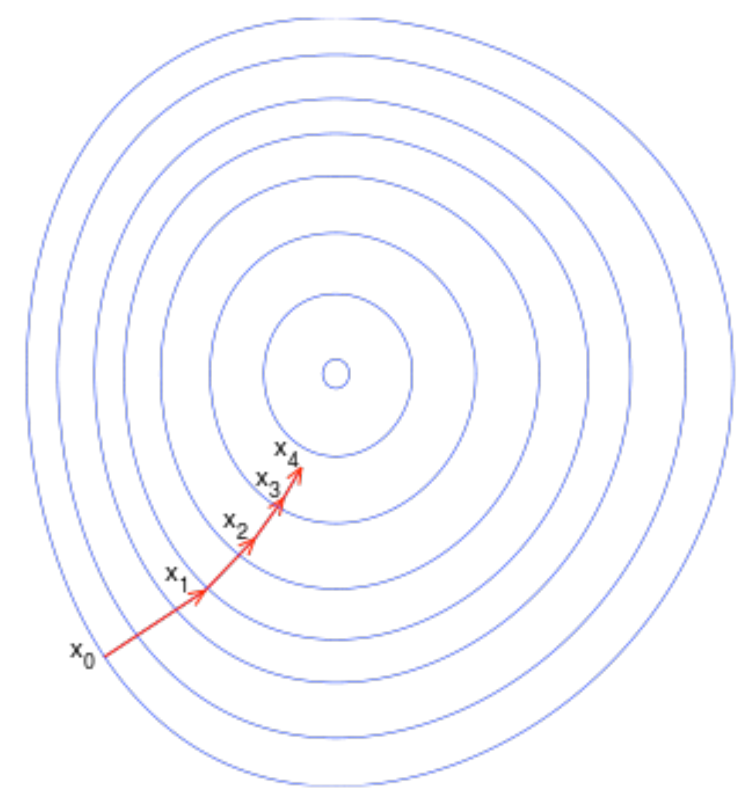


Image: wikipedia

$$A_i^j = A_i^{j-1} - \theta \frac{\partial C}{\partial A_i}$$

Learning rate

Stochastic Gradient Descent

- The gradient descent should be done on the whole training set.
- Gradient descent → very costful
- Alternative: Compute the gradient of the cost function on a random subset of the Training set.
- Much less costful.
- It induces a stochasticity.
- Actually good to avoid overfitting and local minima.

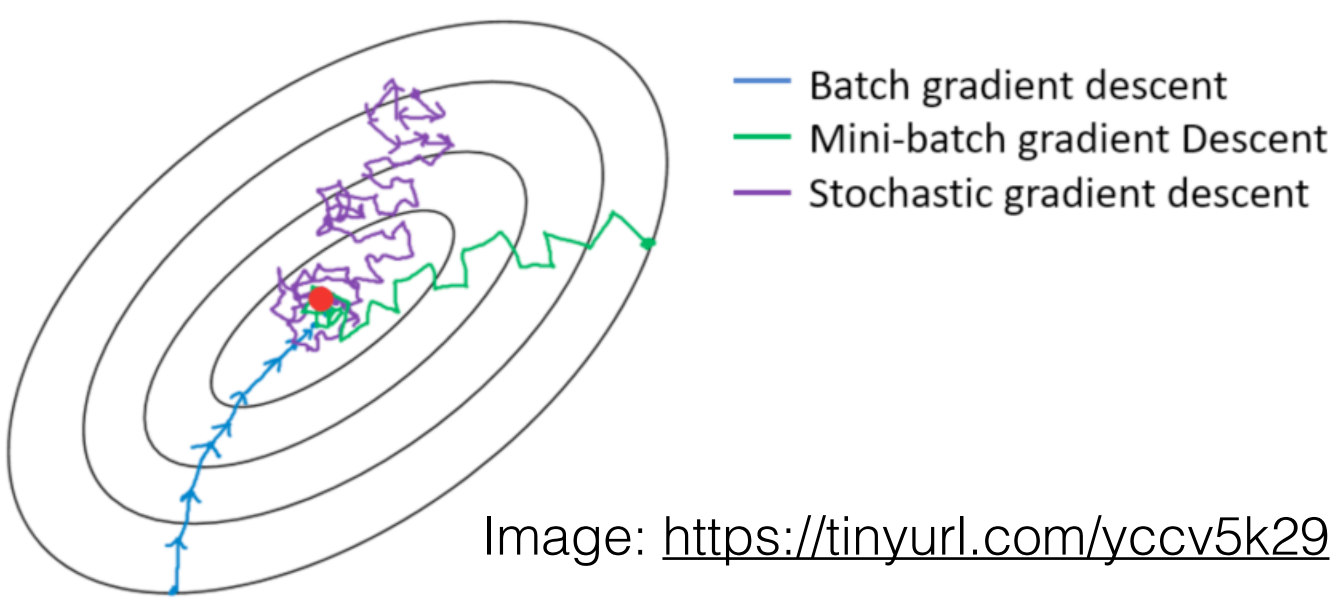
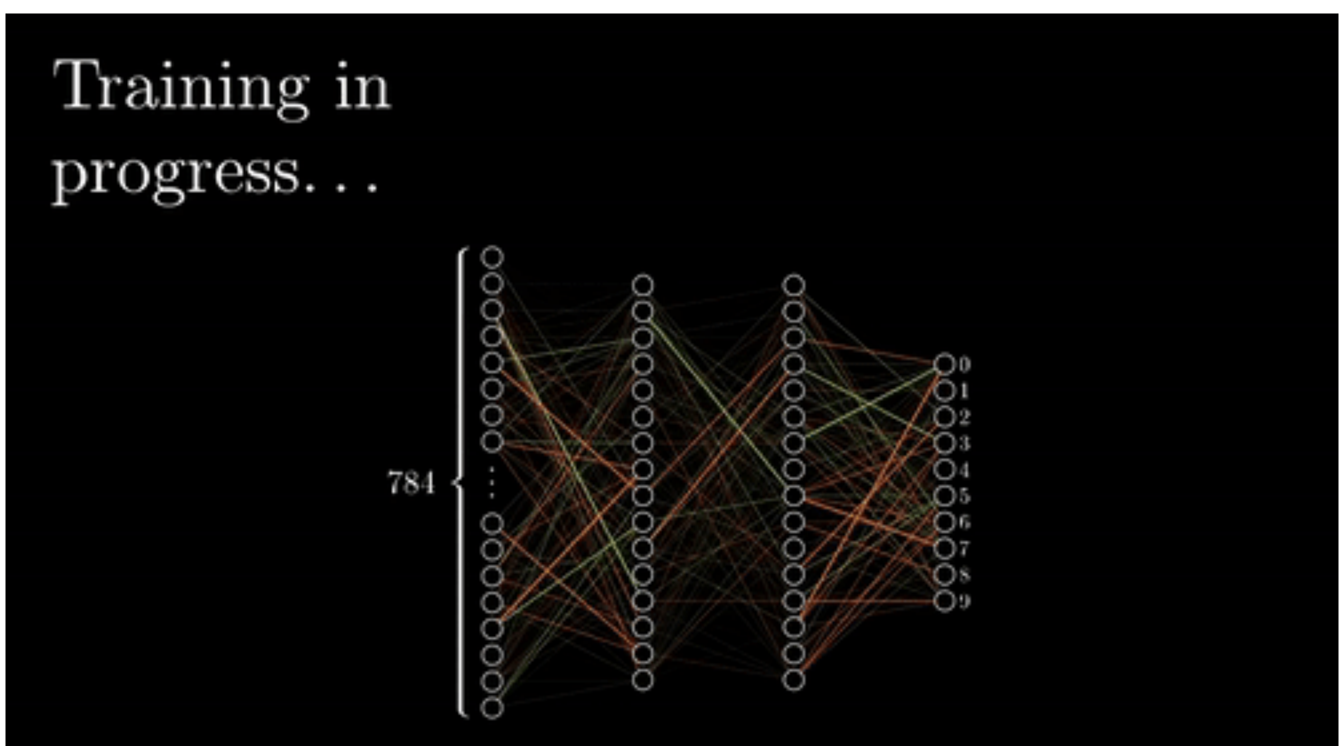


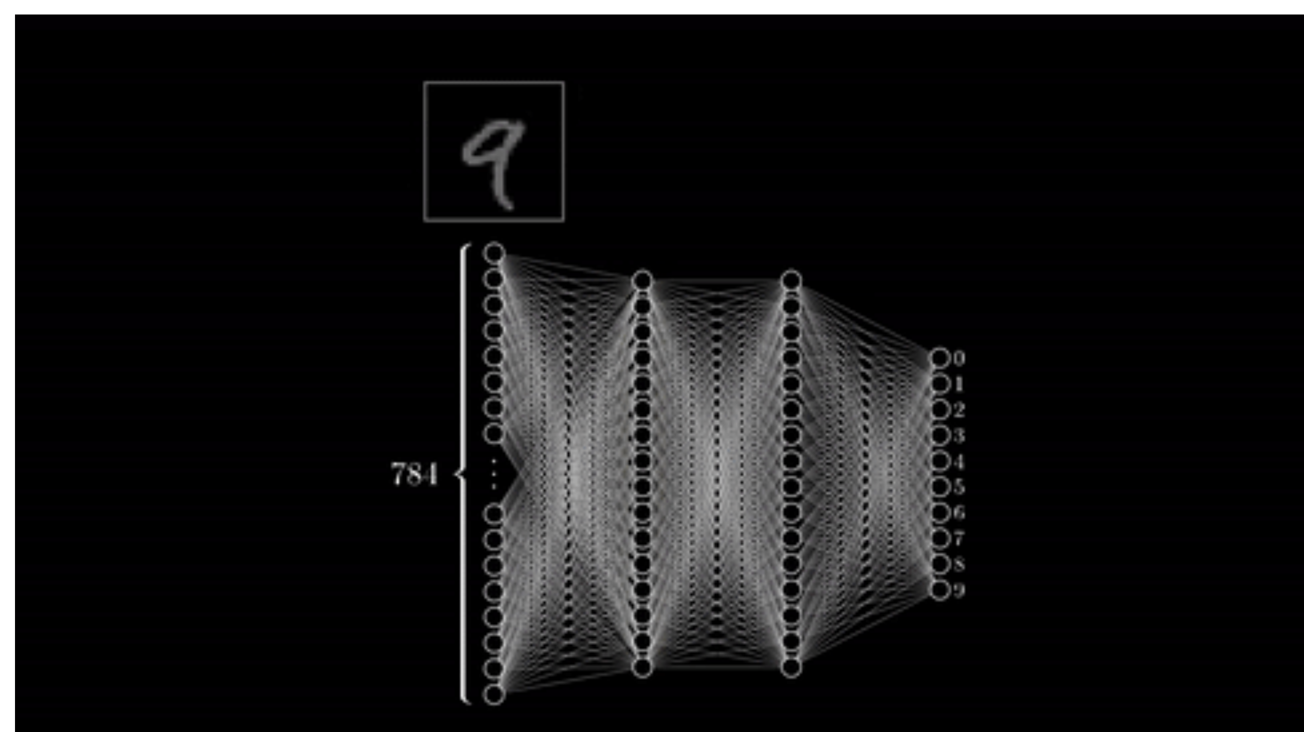
Image: https://tinyurl.com/yccv5k29

Neural Networks and GD

- Computation of the gradient: costful operation
- Fortunately, the error can be back propagated through the network with linear algebra (for the demonstration, previous lecture of Maciej) also: https://goo.gl/Zmczdy (Chapter 2)
- Very good news as this process can be massively parallelized.

https://youtu.be/aircAruvnKk



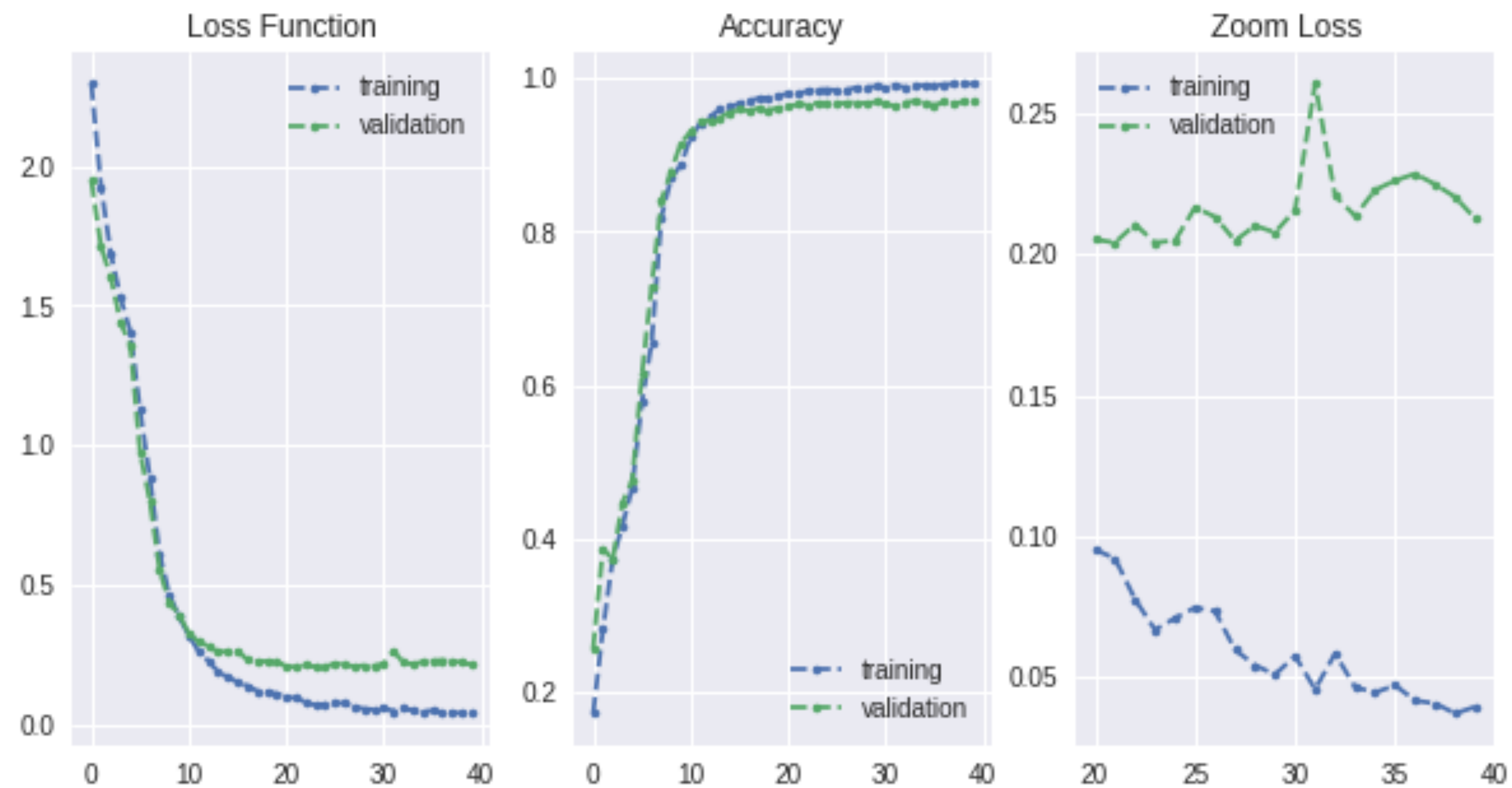


How to train the network

- Divide the set of data in 3
- Training set + validation set (60%)
- test set (40%)
- Training set (80% of the 60%), validation set (20% of the 60%)
- We train the model on the training set.
- We adjust the hyper parameters and check the overfitting on the validation set.
- After training and hyper parameters adjustment, we test the model on the test set

Overfitting

- High accuracy on the training set
- Saturated accuracy on the validation set



See the notebook for an example of overfitting

- Add regularization terms in the Cost function
- Dropout
- Data augmentation

2. Unsupervised Methods

Principal Component Analysis

- For a data set X.
- The PCA algorithms tries to fit an ellipsoid on the dataset.
- One has to compute the covariance matrix $X^T X$ and apply the singular value decomposition (SVD).
- The eigenvectors describe the different axes of the ellipsoid.
- The first eigenvector (largest eigenvalue in SVD) maximizes the variance.
- This method is often used for dimensionality reduction.

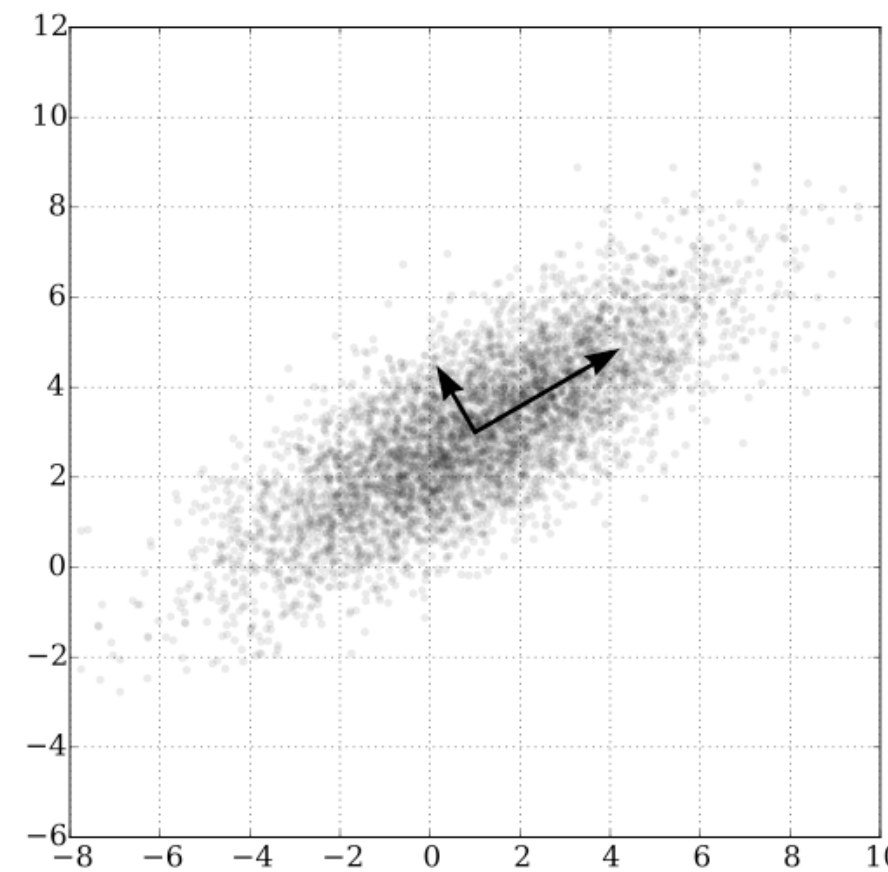
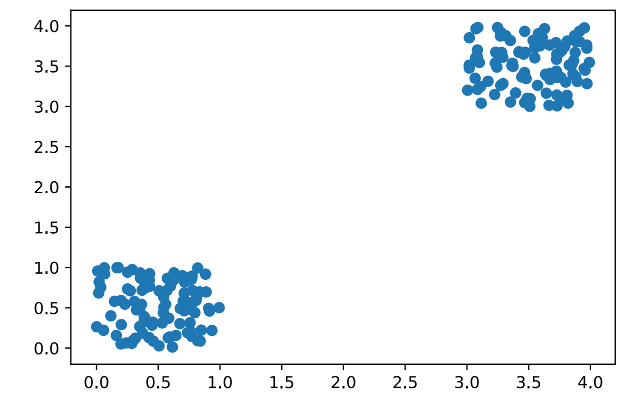
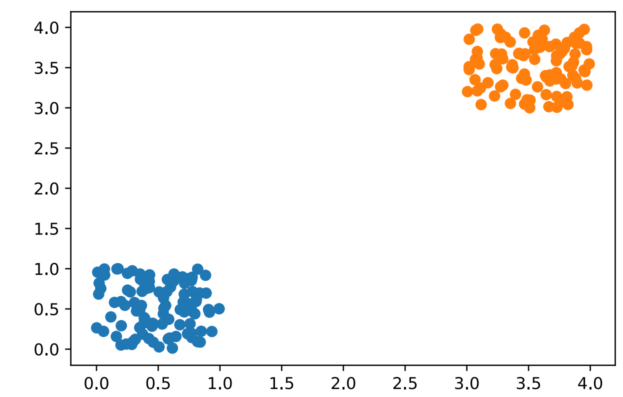


Image: wikipedia

K-means clustering



- For a data set X.
- One chooses a number of clusters
- Initialize centroids randomly
- Assign for each point of X a centroid, the closest one in distance
- Update the position of the centroid as the mean of the points included in the cluster



3. Convolutional neural networks

Convolutional Neural Networks

- Idea: use convolutional filters.
- The machine learns the parameters of the filters.

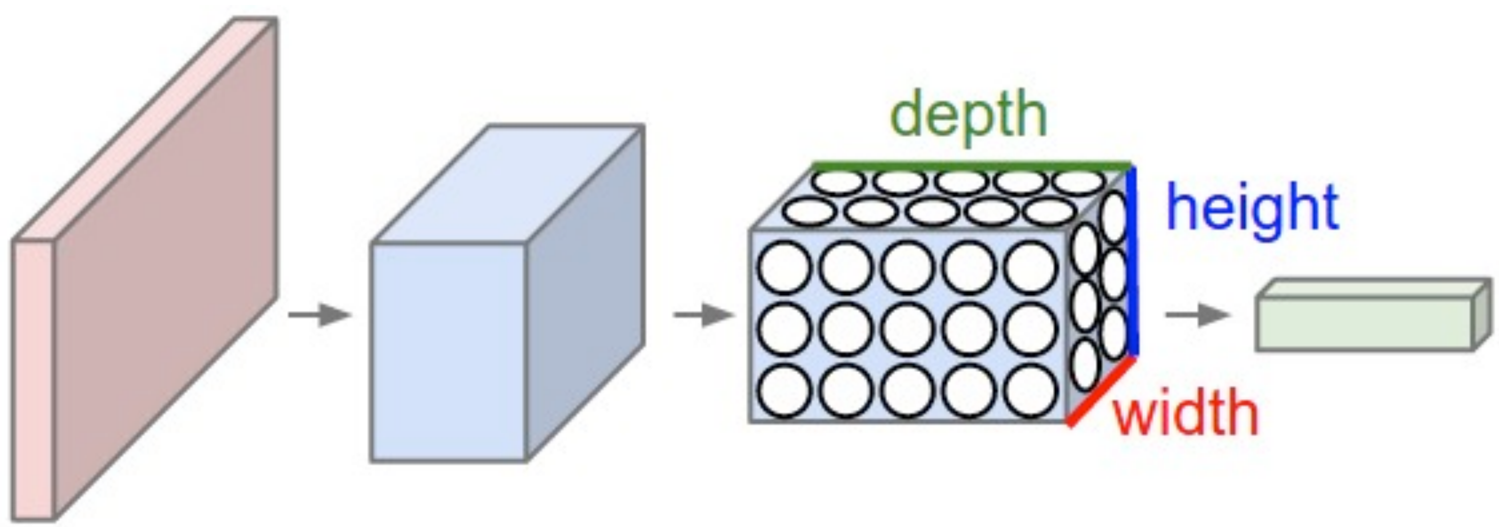
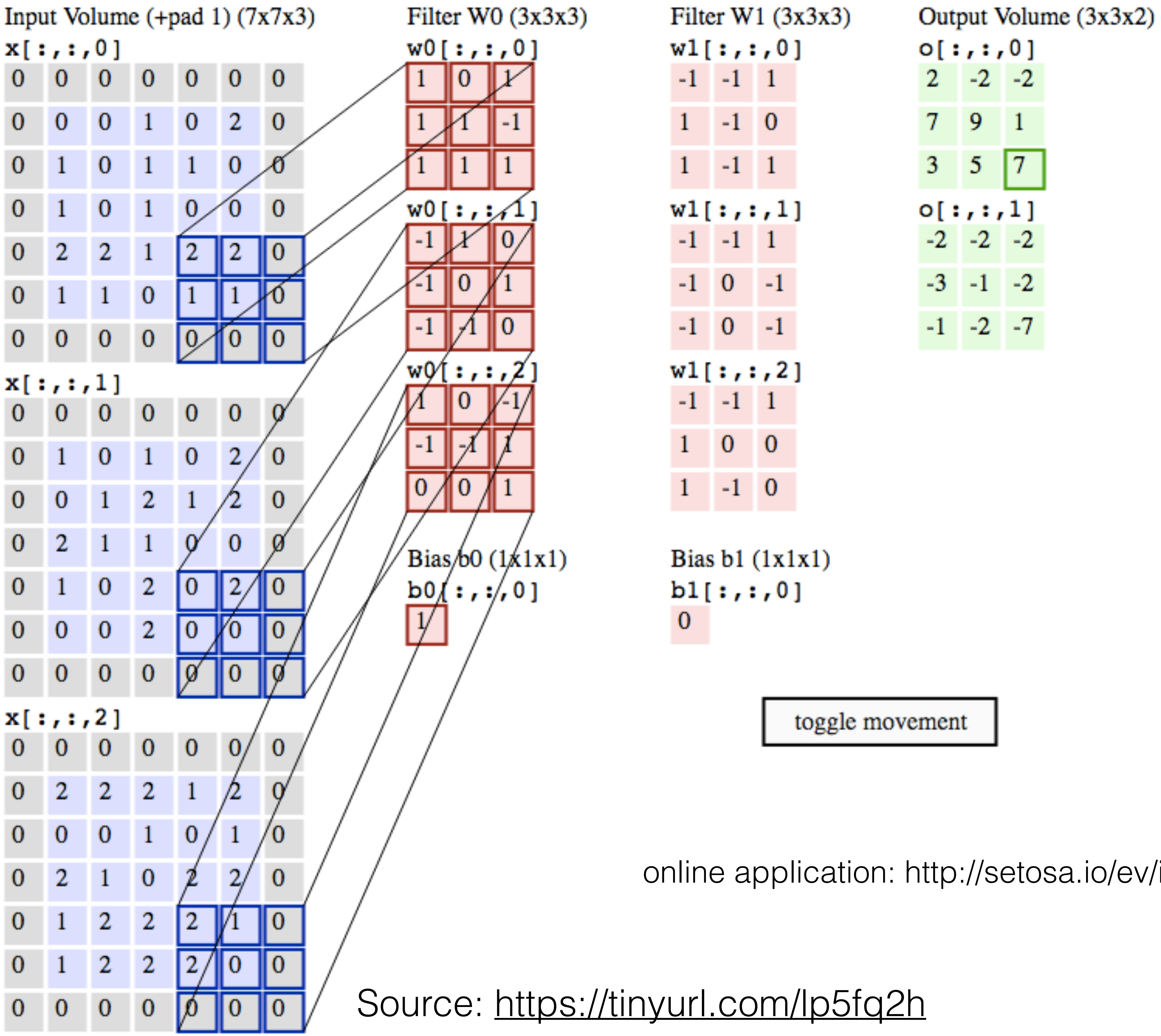


Image: https://tinyurl.com/lp5fq2h

Convolutional Neural Networks



online application: http://setosa.io/ev/image-kernels/

Source: https://tinyurl.com/lp5fq2h

Convolutional Neural Networks for Image Classification

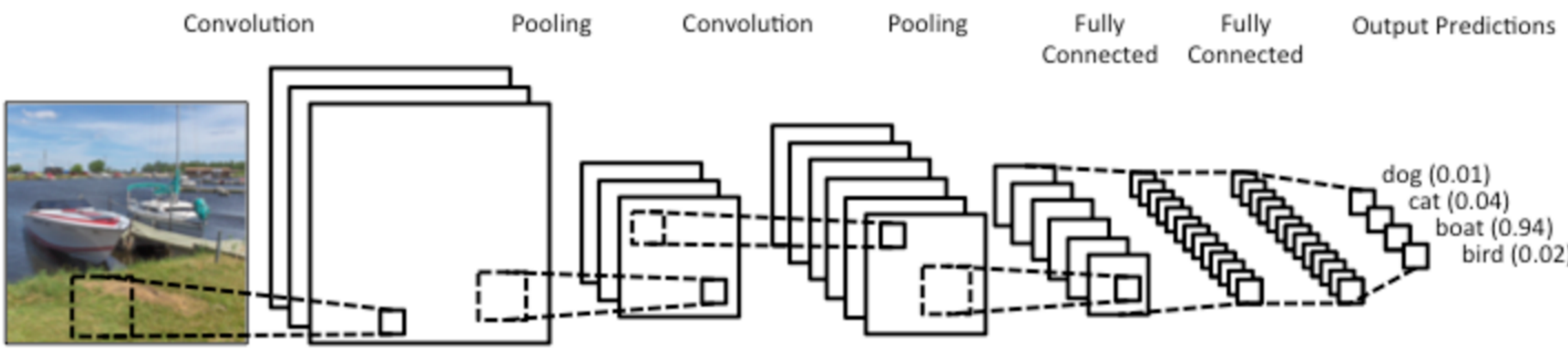
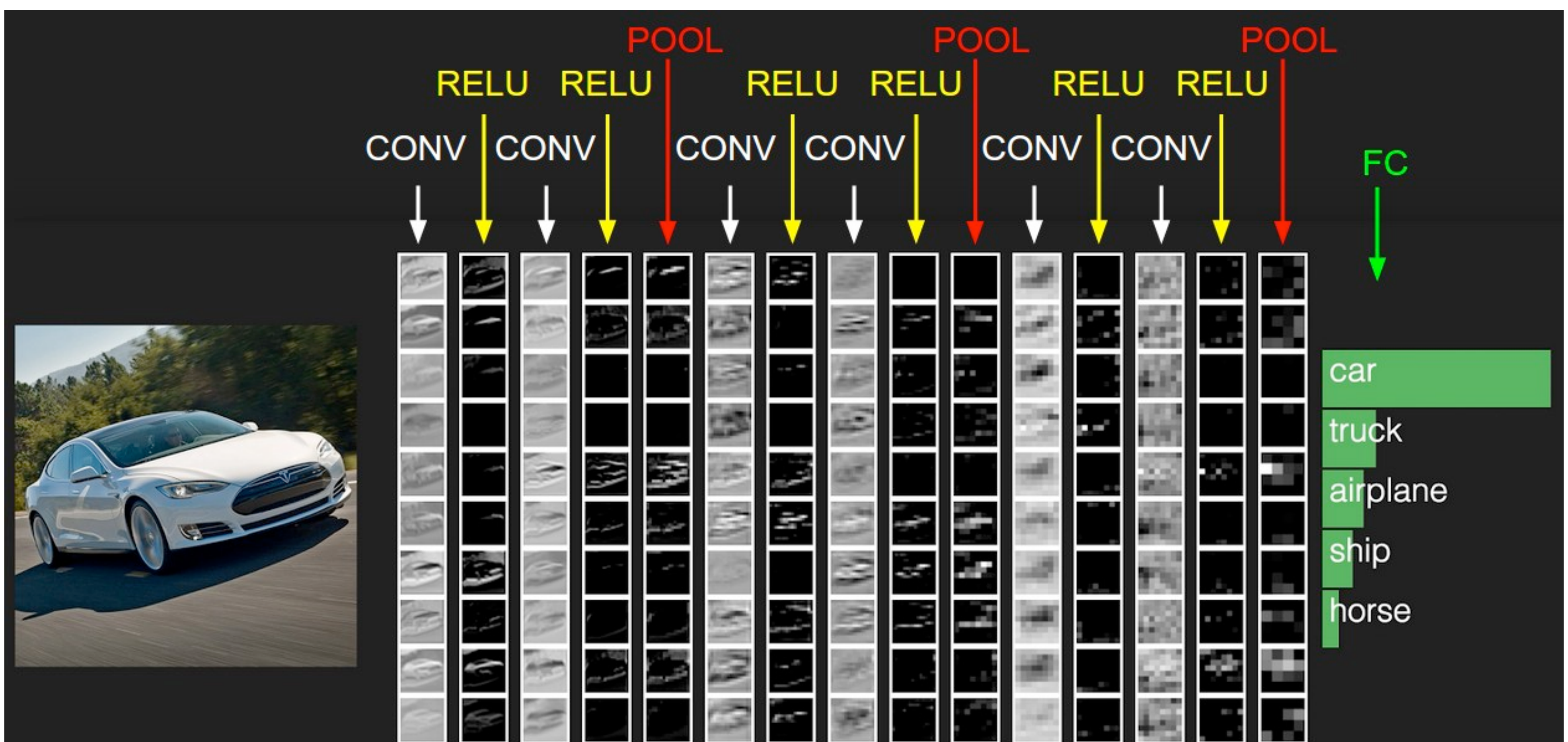


Image bottom: https://tinyurl.com/lp5fq2h

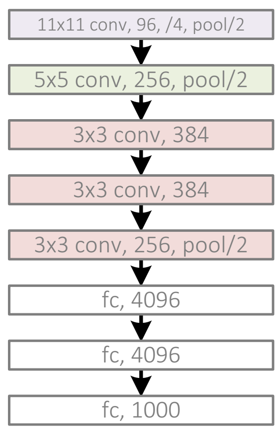


- Transfer learning

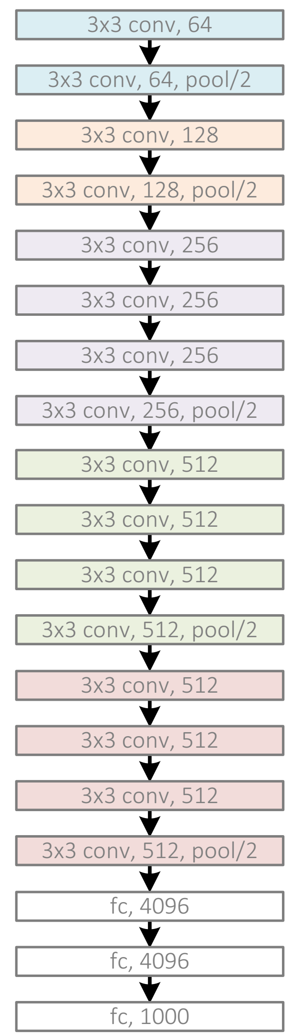
Convolutional Neural Networks for Image Classification

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



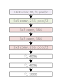
GoogleNet, 22 layers
(ILSVRC 2014)

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

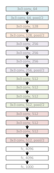
Convolutional Neural Networks for Image Classification

Revolution of Depth

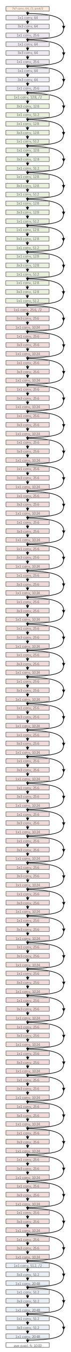
AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)

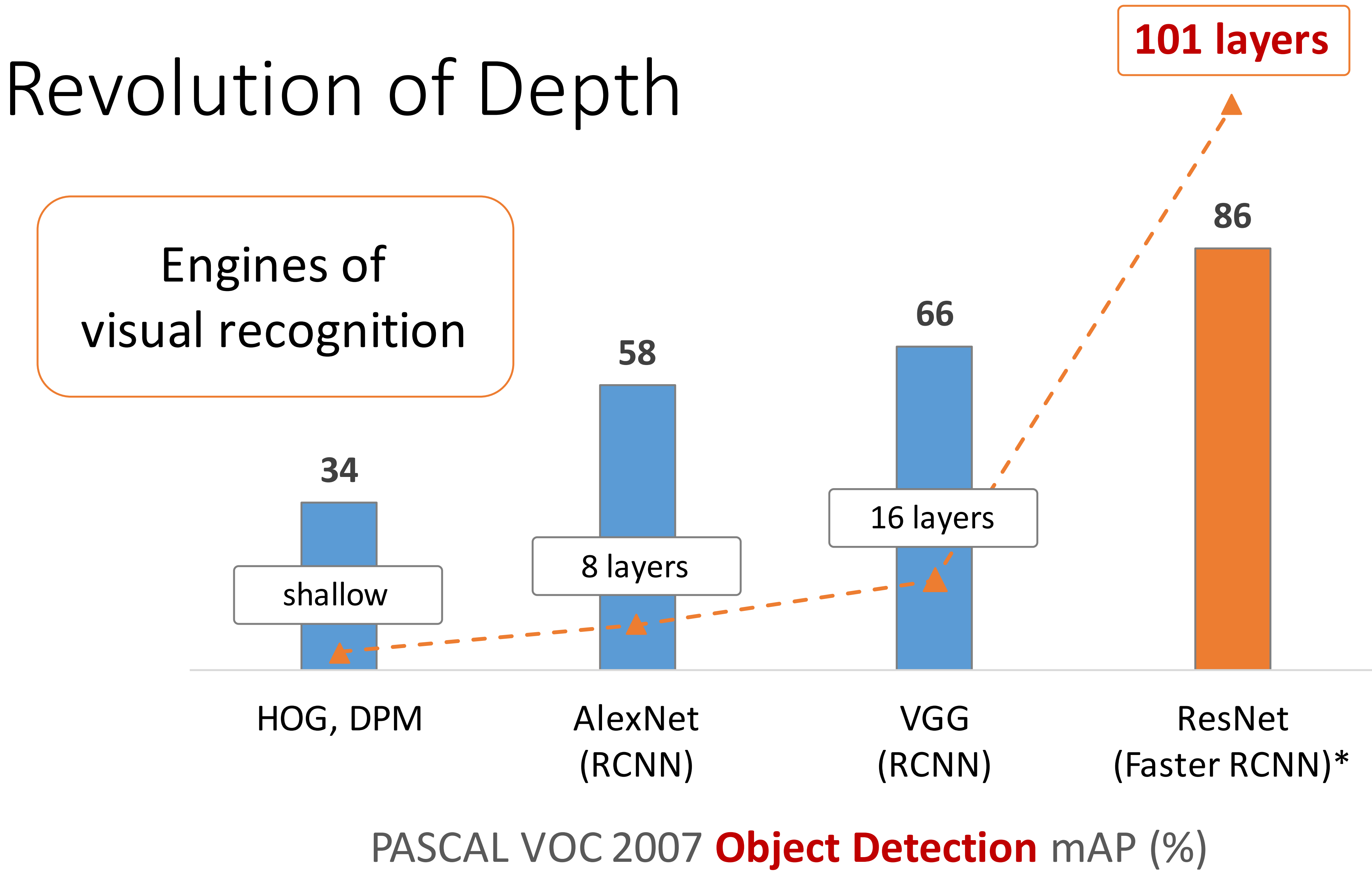


ResNet, 152 layers
(ILSVRC 2015)



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

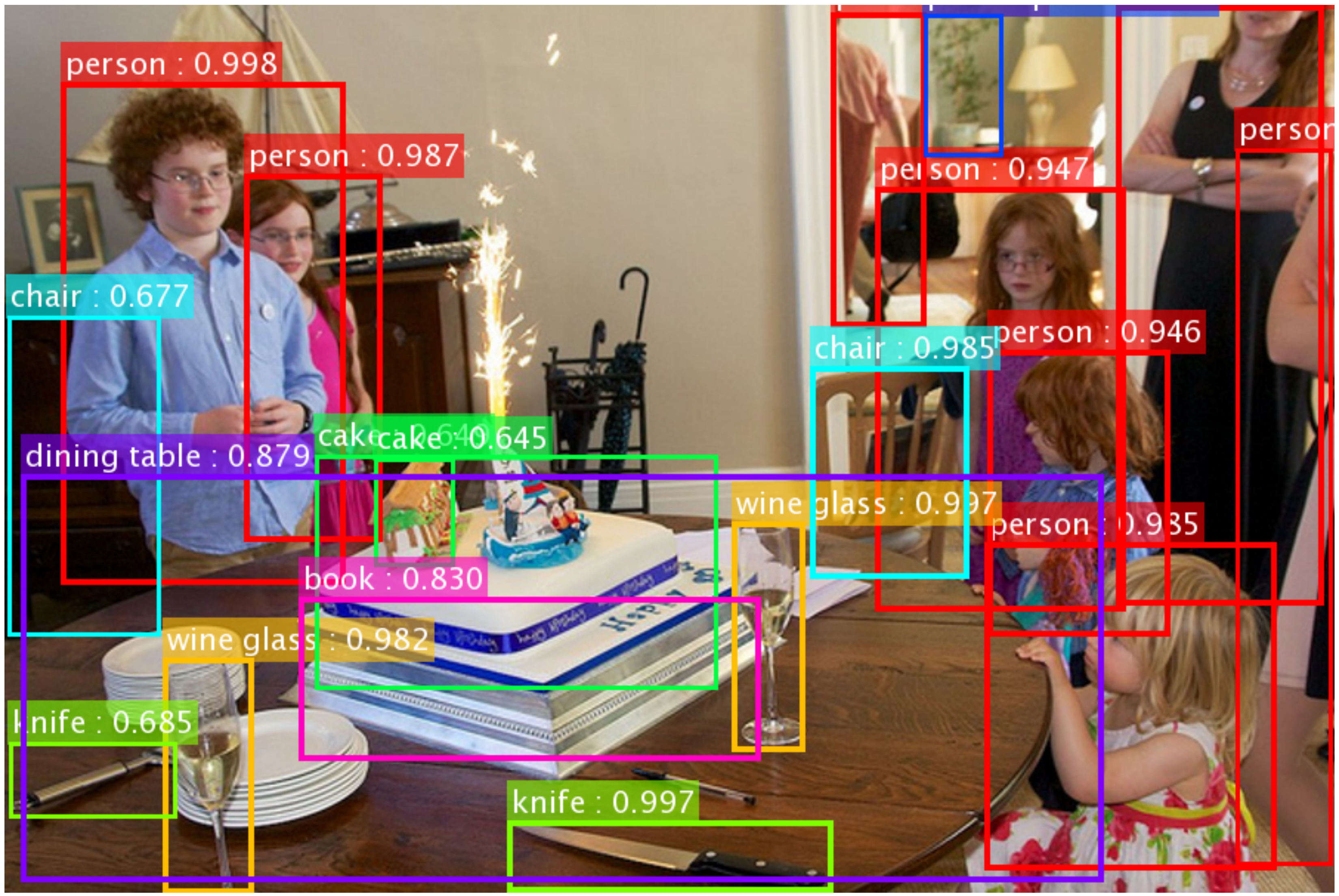
Convolutional Neural Networks for Image Classification



*w/ other improvements & more data

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

Convolutional Neural Networks for Image Classification



ResNet’s object detection result on COCO

References

- Michael's Nielsen book on neural networks: https://goo.gl/Zmczdy
- Book "The element of statistical learning" (springer)
- Stanford course on convolutional neural networks: http://cs231n.stanford.edu/
- Fastai course: https://course.fast.ai/
- Fastai book: https://github.com/fastai/fastbook