

Introduction to Deep Learning for Geoscience

Ronan Fablet (ronan.fablet@ilt-atlantique.fr)

Course on “Data Science for Geosciences”, DSG’2020, Toulouse, January 2020

This course focuses on "**an introduction to deep learning**", not just "**how to use**" deep learning in practice.

Resources

- **Book: Deep Learning**

Goodfellow, Bengio, Courville, MIT Press

Online version <http://www.deeplearningbook.org/>

- **Online course by Andrew Ng (Stanford/Baidu)**

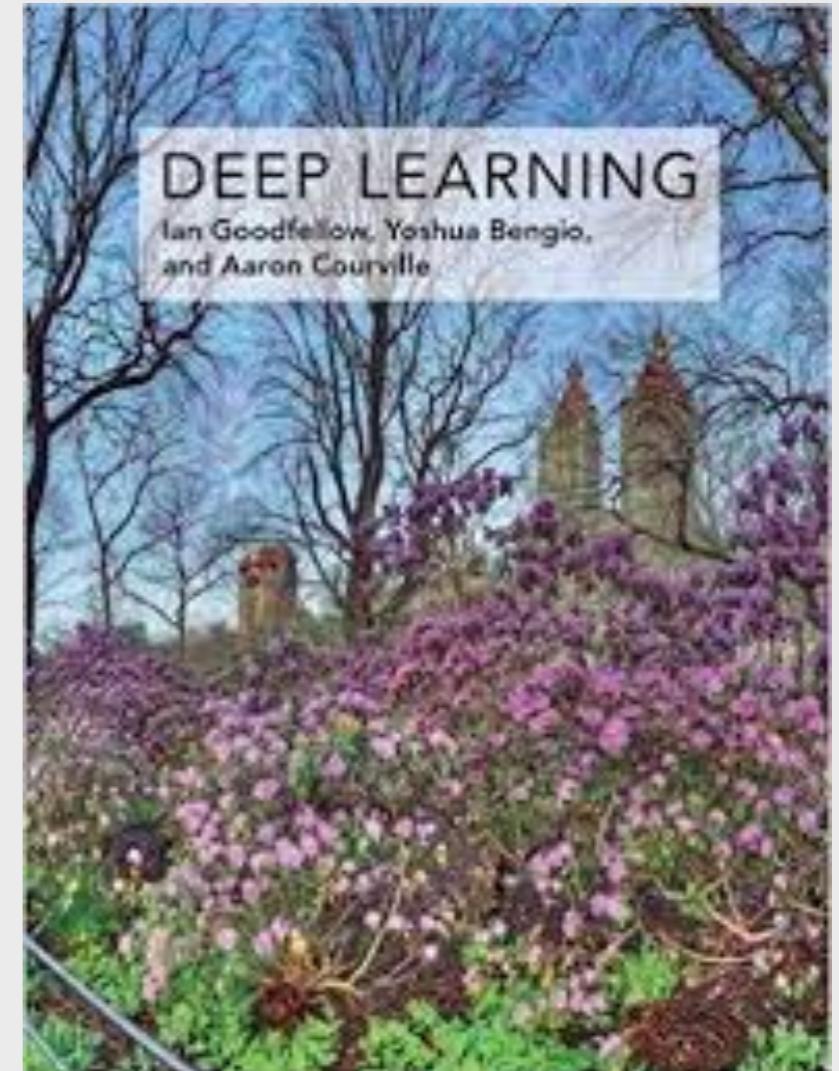
Youtube: [link](#)

Online course on Coursera: [link](#)

- **Review paper: Deep learning in neural networks by**

J. Schmidhuber

pdf: [link](#)



Neural Networks 61 (2015) 85–117
Contents lists available at ScienceDirect
Neural Networks
journal homepage: www.elsevier.com/locate/neunet

Review
Deep learning in neural networks: An overview
Jürgen Schmidhuber
The Swiss AI Lab IDSIA, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, University of Lugano & SUPSI, Galleria 2, 6928 Manno-Lugano, Switzerland

ARTICLE INFO
Article history:
Received 2 May 2014
Received in revised form 12 September 2014
Accepted 14 September 2014
Available online 13 October 2014

ABSTRACT
In recent years, deep artificial neural networks (including recurrent ones) have won numerous contests in pattern recognition and machine learning. This historical survey compactly summarizes relevant work, much of it from the previous millennium. Shallow and Deep Learners are distinguished by the depth of their *credit assignment paths*, which are chains of possibly learnable, causal links between actions and effects. I review deep supervised learning (also recapitulating the history of backpropagation), unsupervised learning, reinforcement learning & evolutionary computation, and indirect search for short programs encoding deep and large networks.

© 2014 Published by Elsevier Ltd.

Roadmap

- What is deep learning ?
- What does (machine) learning mean ?
- What are Neural Networks ?
- What are the key components of DL models ?
- A (very) short tour of the NN zoo

What is Deep Learning?

What is Deep Learning?

Artificial Intelligence

Machine Learning

Random
Forest

SVM

Nearest-
Neighbour

Deep
Learning

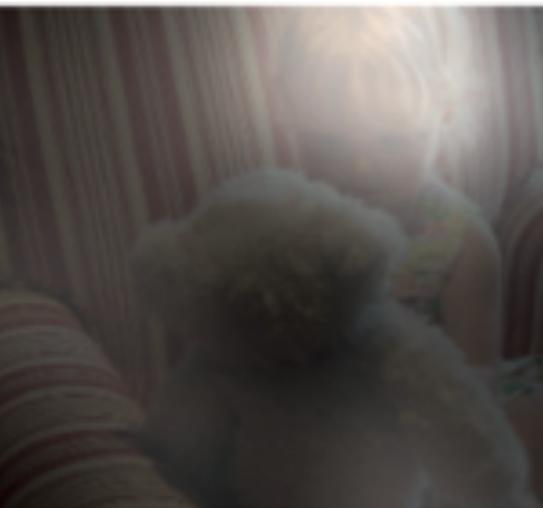
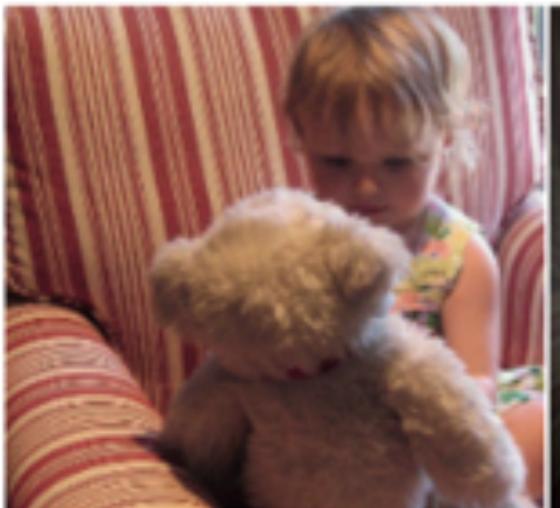
**Deep learning has rapidly become the state-of-art-framework
for a wide range of applications in computer vision, natural
language processing, signal processing...**

From Image to Text (image captioning)



A woman is throwing a **frisbee** in a park.

A **dog** is standing on a hardwood floor.



A little **girl** sitting on a bed with a **teddy bear**.



A group of **people** sitting on a boat in the water.

Automatic Translation

The image shows two side-by-side screenshots of automatic translation tools. On the left is Google Translate, displaying a French sentence "Ce cours sur l'apprentissage profond est génial." and its English translation "This deep learning course is great." On the right is DeepL, showing the same sentence and translation. A large red stamp with the text "Early 2017" is overlaid diagonally across both screens.

Google

Traduction

Désactiver la traduction instantanée

Anglais Français Arabe Déterminer la langue

Traduire

Ce cours sur l'apprentissage profond est génial.

This deep learning course is great.

48/5000

Suggérer une modification

DeepL

Translate from FRENCH (detected) ▾

Translate into ENGLISH ▾

Ce cours sur l'apprentissage profond est génial.

This course on deep learning is great.

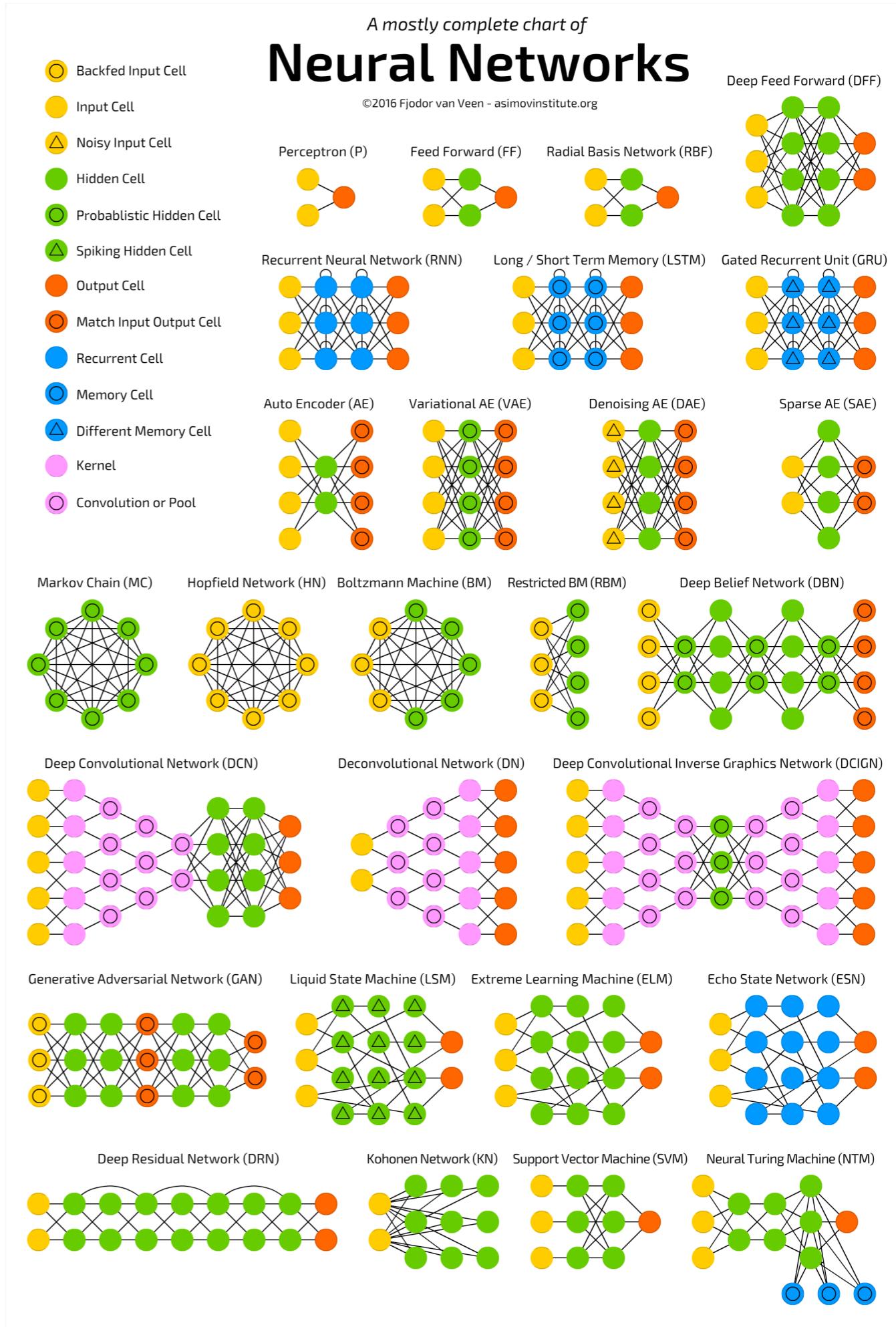
To look up words in the dictionary, just click on them.

Early 2017

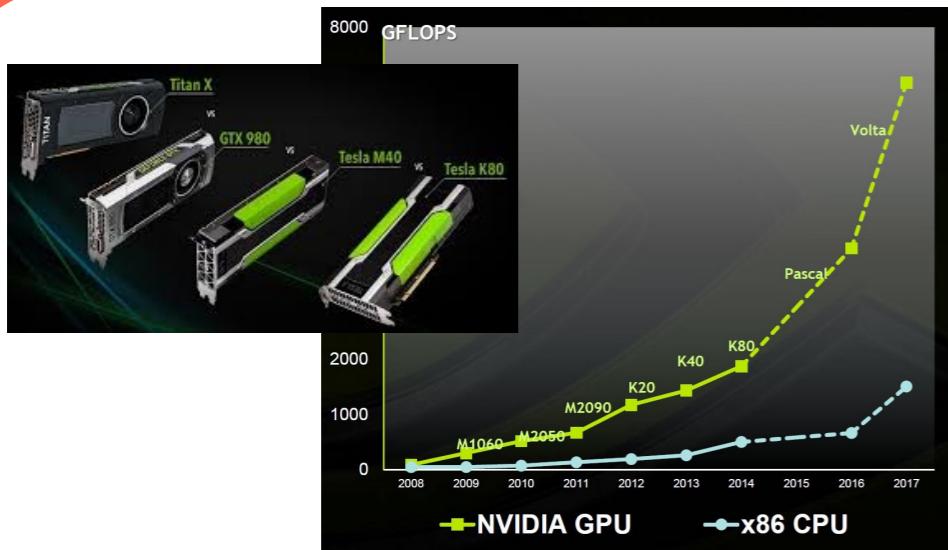
and many more.....

Deep learning relies on neural networks, which are not new...

Artificial neural networks date back to the 60's and were popular in the 80's.



Key reasons for DL emergence



High-performance computing (GPU)



Large annotated dataset (> 1M)

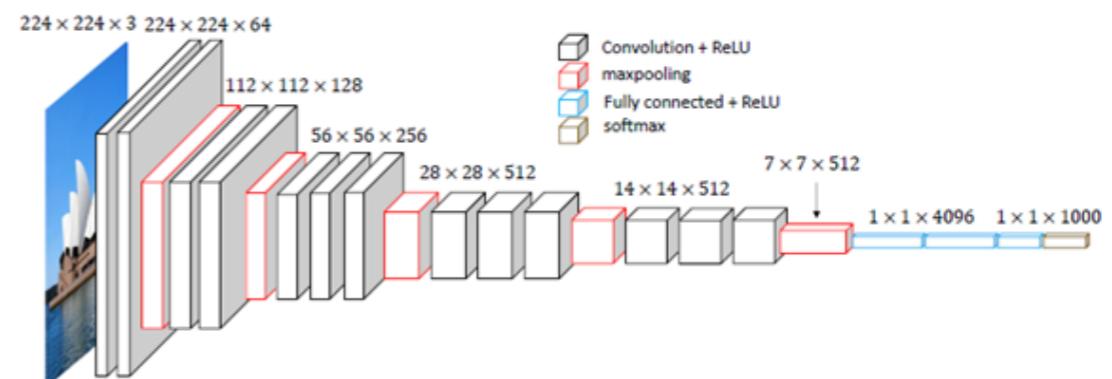


K Keras

Caffe

PYTORCH

Efficient & easy-to-use libraries



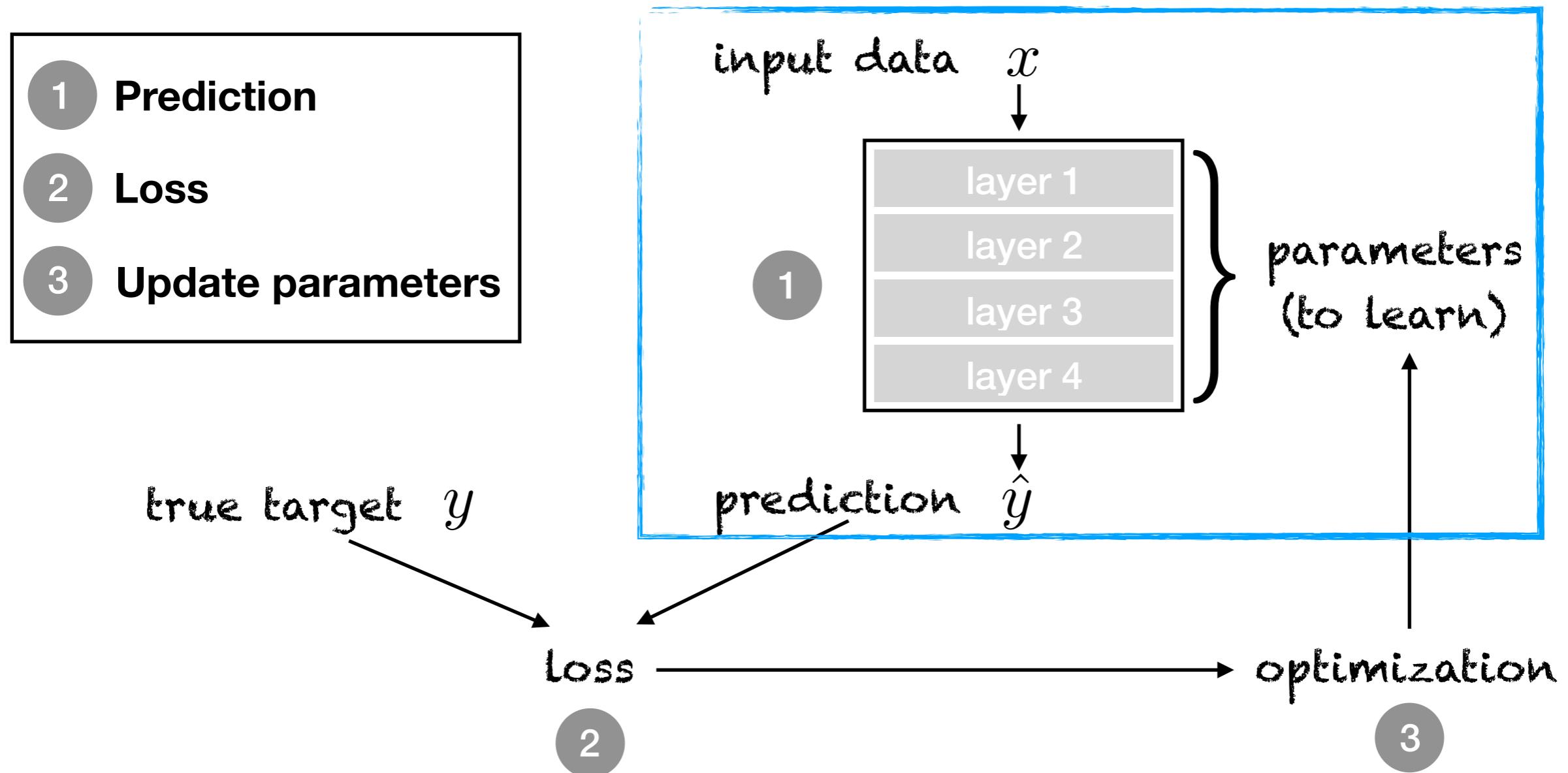
End-to-end learning

**WHAT DOES (MACHINE)
LEARNING MEAN ?**

Machine/Deep Learning

Definition (wikipedia): Machine learning algorithms build a **mathematical model** based on sample data, known as "**training data**", in order to make predictions or decisions without being explicitly programmed to perform the task.

Machine learning



Machine/Deep Learning

Mathematical formulation: Learning comes to minimising some loss function given w.r.t. model parameters and training data

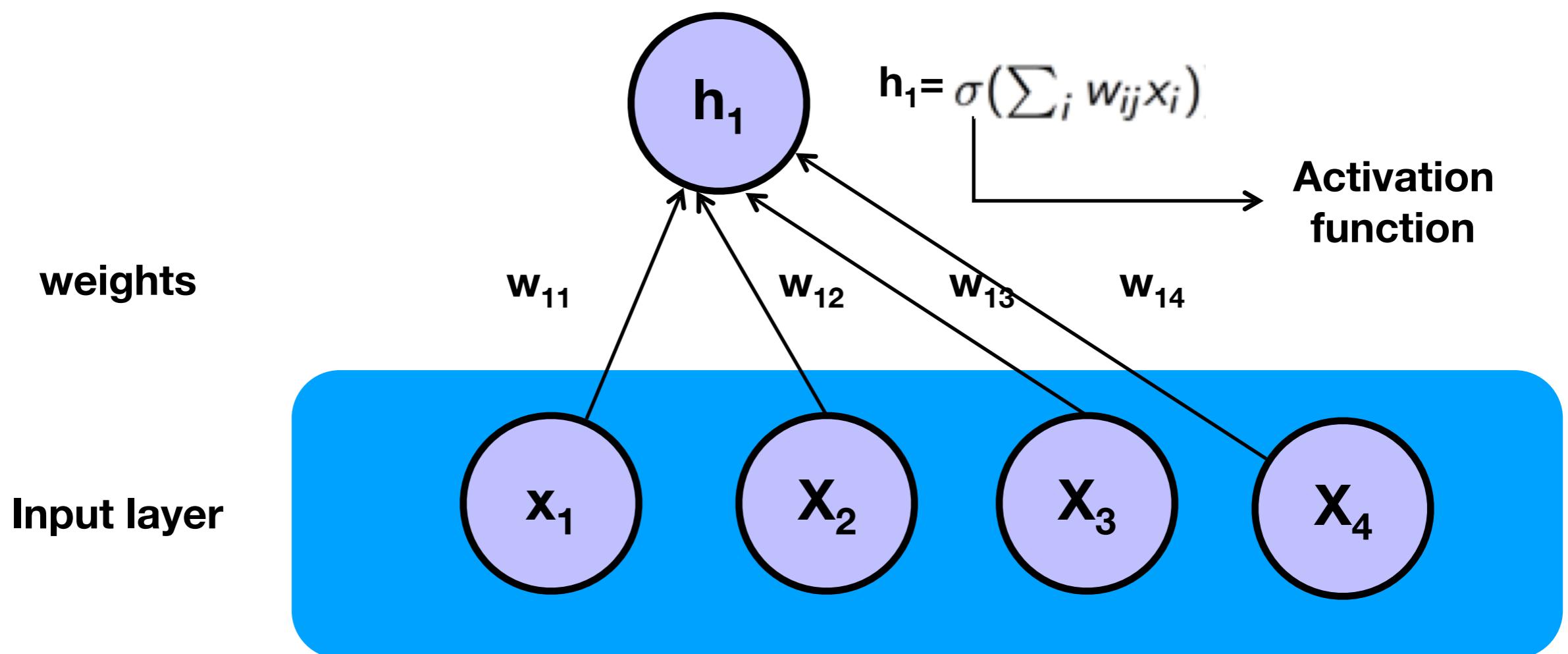
$$\hat{\theta} = \arg \min_{\theta} \mathcal{L} \left(\{x_i, y_i\}_{i \in \{1, \dots, N\}}; f_{\theta} \right)$$

Key questions:

- Which parameterisation for model f ?
- Which loss function ?

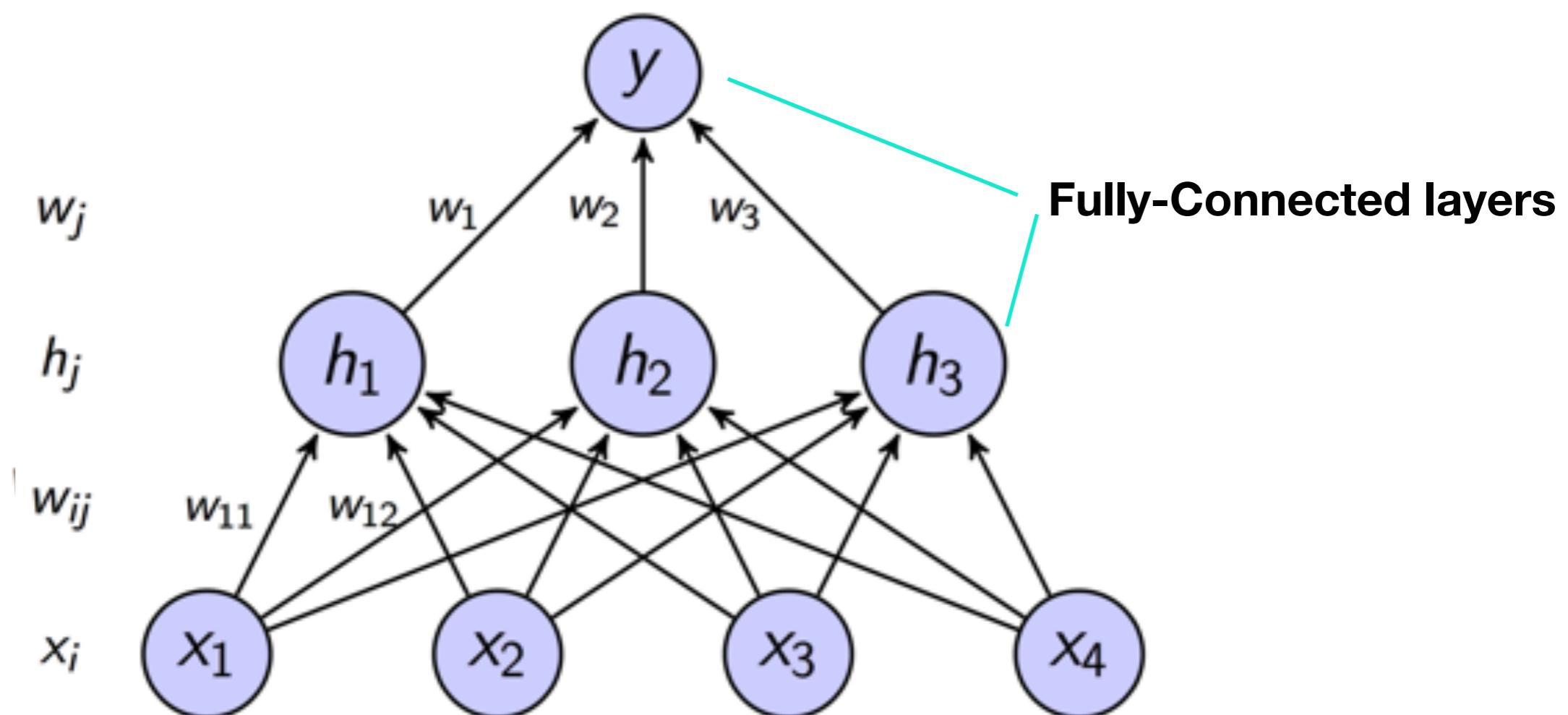
NEURAL NETWORKS: KEY PRINCIPLES & BUILDING BLOCKS

The artificial neuron



$$f(x) = \sigma(\sum_j w_j \cdot h_j) = \sigma(\sum_j w_j \cdot \sigma(\sum_i w_{ij}x_i))$$

Multilayer Perceptron (MLP)



$$f(x) = \sigma\left(\sum_j w_j \cdot h_j\right) = \sigma\left(\sum_j w_j \cdot \sigma\left(\sum_i w_{ij} x_i\right)\right)$$

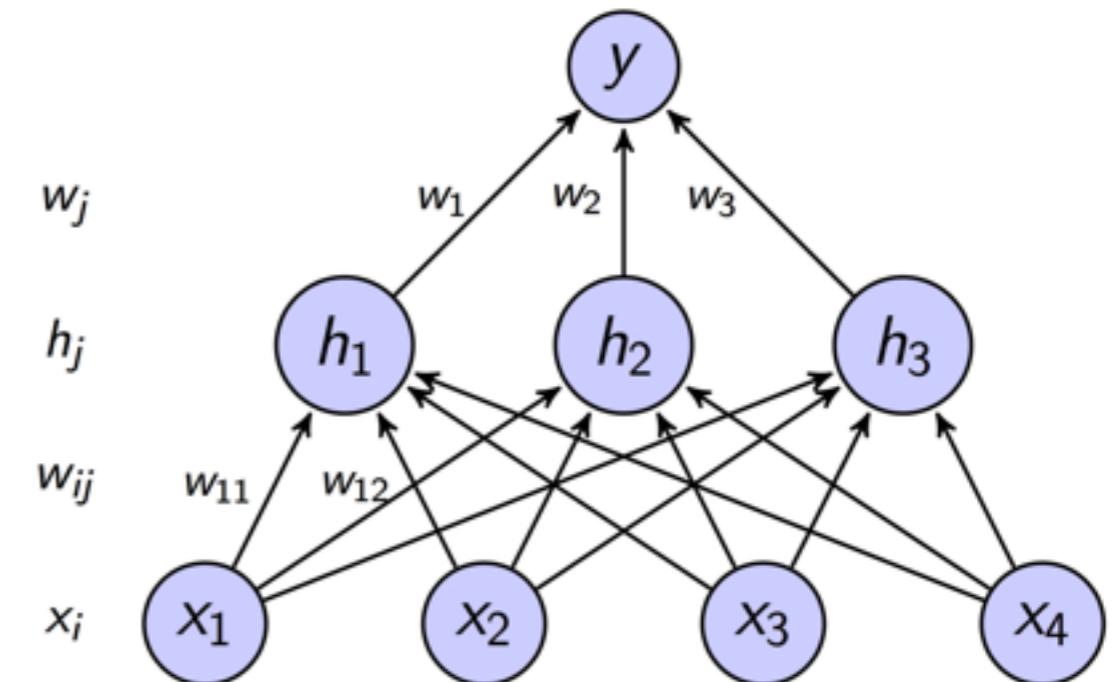
Multilayer Perceptron (MLP)

- **Neuron:** basic unit, inspired from neuroscience

- **Feedforward:** no feedback connection

- **Network:** composition of functions

- **Parametric model:** $y=f(x_1, x_2, x_3, x_4, \Theta)$



- **Training:** estimation of model parameter to minimise some loss function

**Let's play with MLPs using
tensorflow playground**

<http://playground.tensorflow.org/>

Neural networks: composition idea

- Approximation through the composition of (simple) elementary functions:

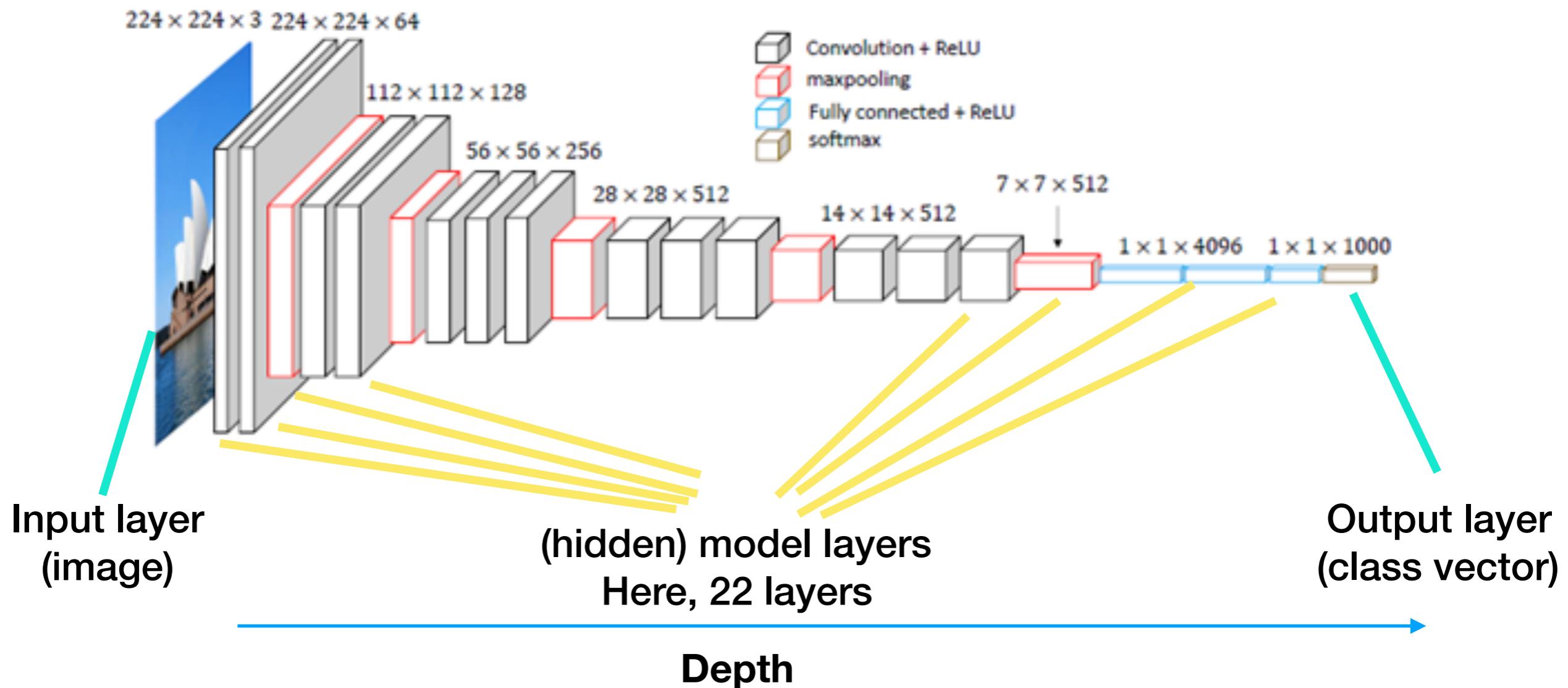
$$f_{\theta}(x) = f_{\theta_N} \circ \dots \circ f_{\theta_2} \circ f_{\theta_1}(x)$$

- Key features:
 - Any continuous function can be approximated as the composition of elementary functions
 - Analytical/exact computation of the derivative of f with respect to parameters and input variables
 - Direct exploitation of gradient-based optimisation schemes

**What are deep learning
models ?**

Basics of DL models

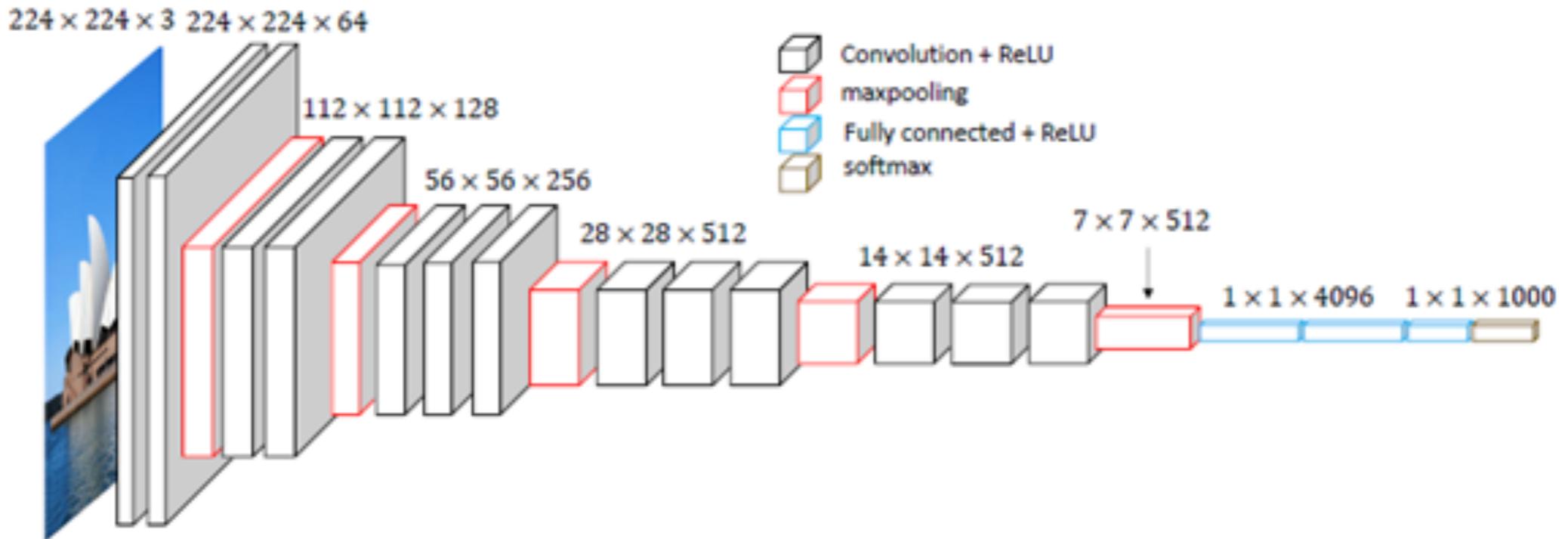
DL models are (in general) feedforward models. VGG16 as an illustration



The more layers, the deeper..... Some models may have up to several hundreds to thousands of layers.

Basics of DL models

DL models are (in general) feedforward models. VGG16 as an illustration



Elementary components

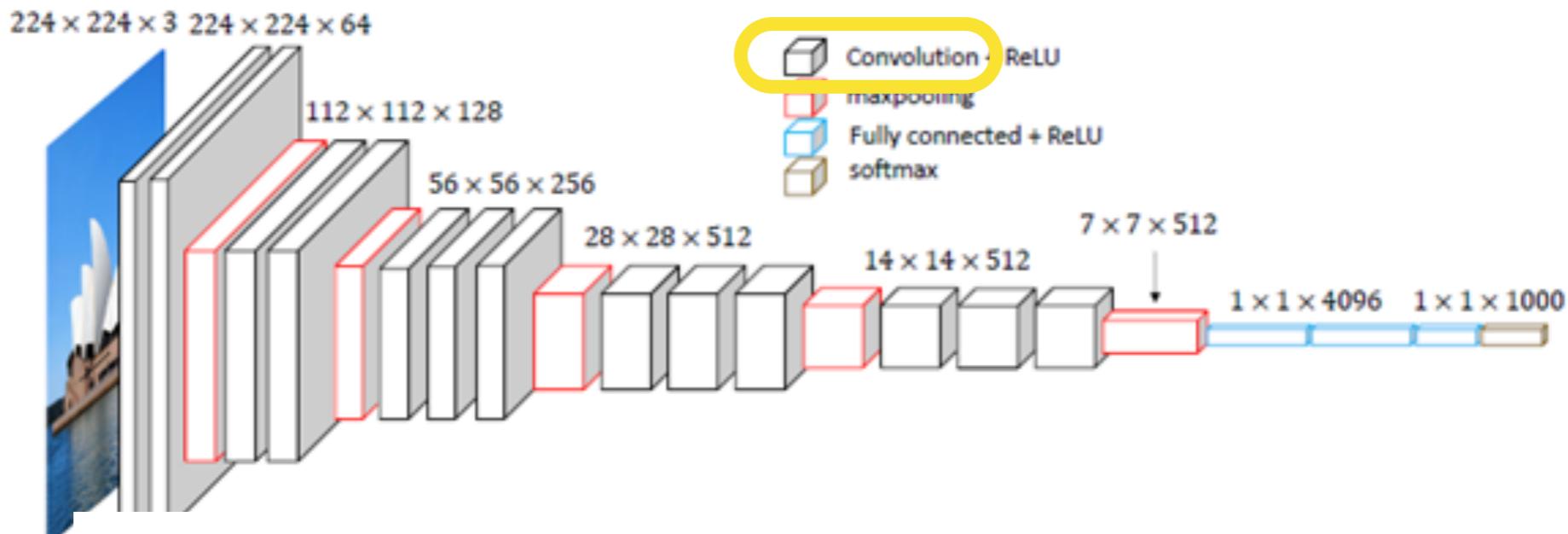
Convolution layers

Activation layers

Pooling layers

Dense layers

Basics of DL models



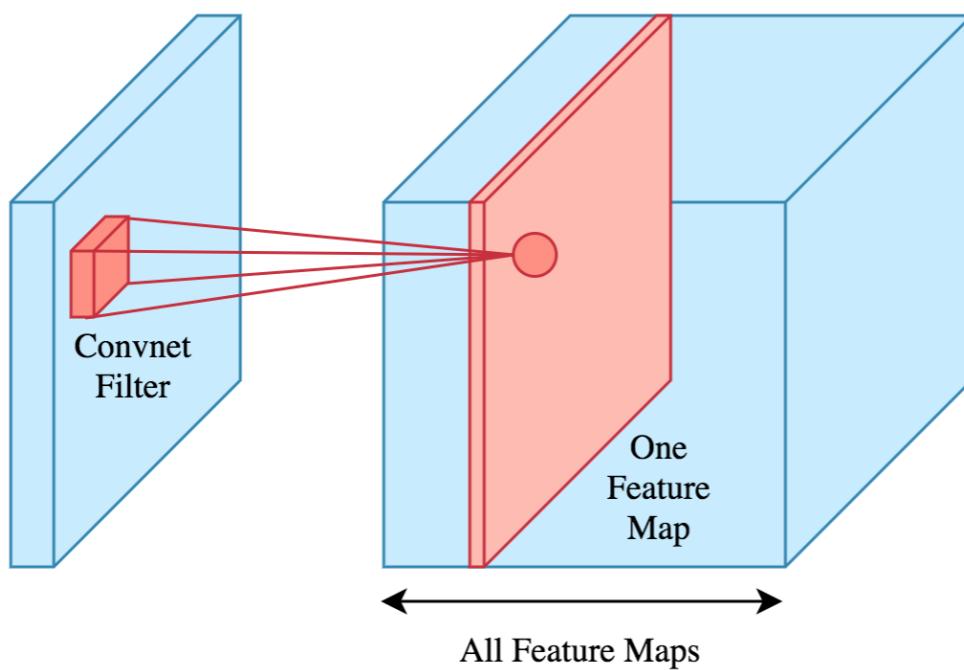
Elementary components

Convolution layers

Activation layers

Pooling layers

Dense layers

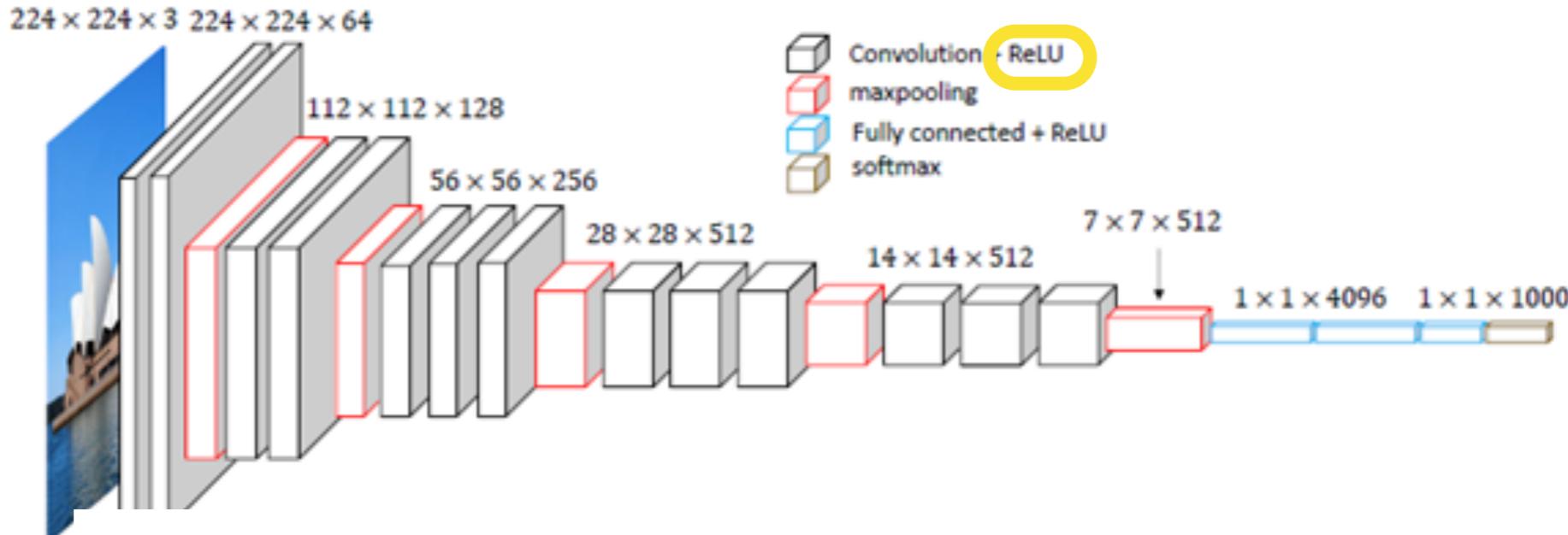


Number of parameters:
Filter size x Number of filters

e.g. $3 \times 3 \times K \times N_{\text{filt}}$

Independent on the sizes of the input
and output layer

Basics of DL models



Elementary components

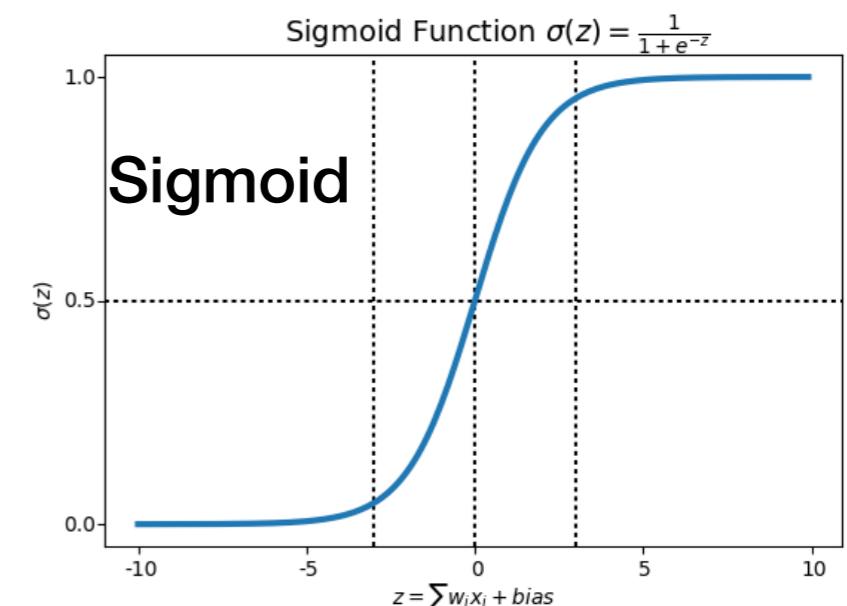
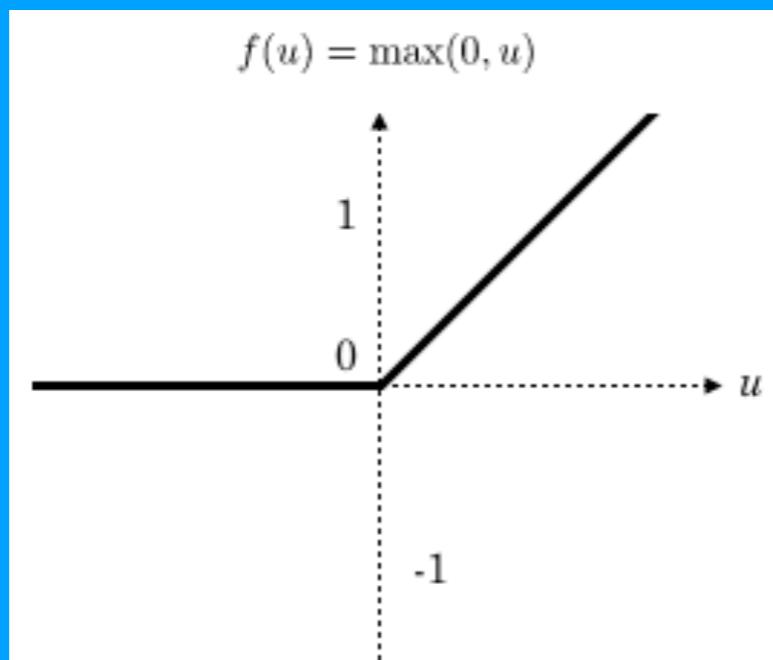
Convolution layers

Activation layers

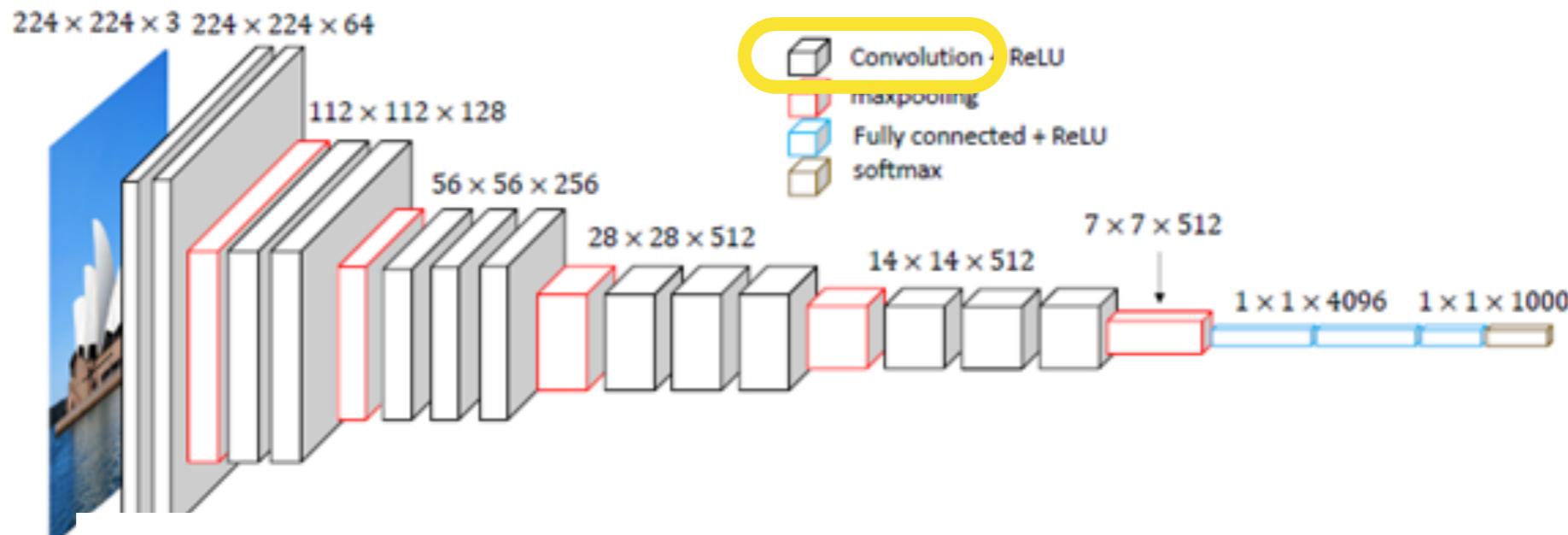
Pooling layers

Dense layers

ReLU
(Rectified
Linear Unit)



Basics of DL models



Elementary components

Convolution layers

Activation layers

Pooling layers

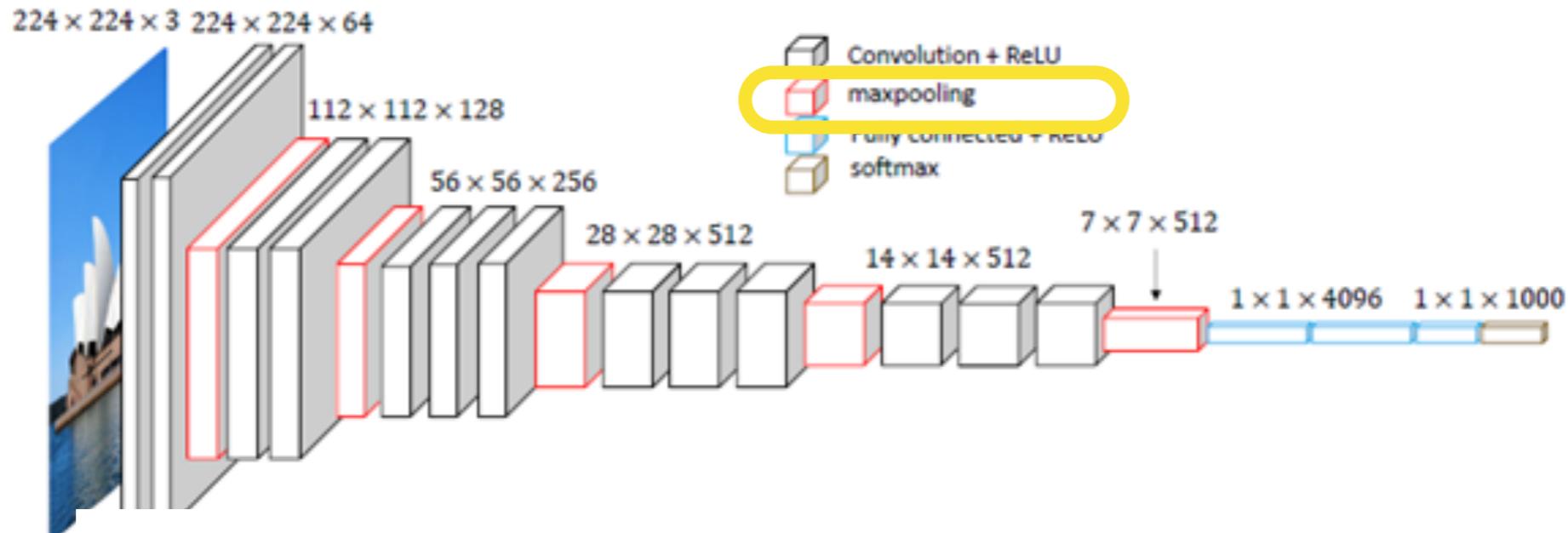
Dense layers

Number of parameters:
Keras function
`model.add(Conv2D(32, kernel_size=(3, 3),
activation='relu',
input_shape=input_shape))`

put

All Feature Maps

Basics of DL models



Elementary components

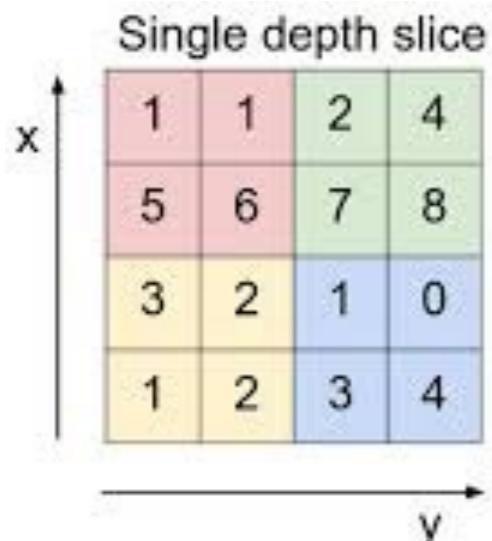
Convolution layers

Activation layers

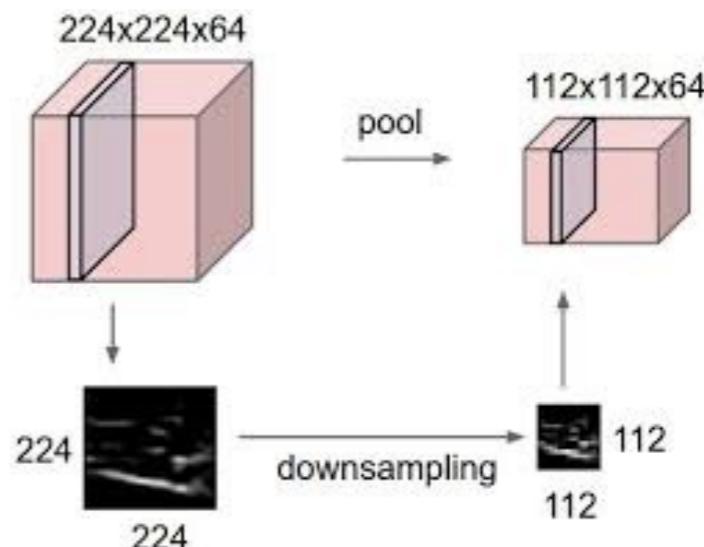
Pooling layers

Dense layers

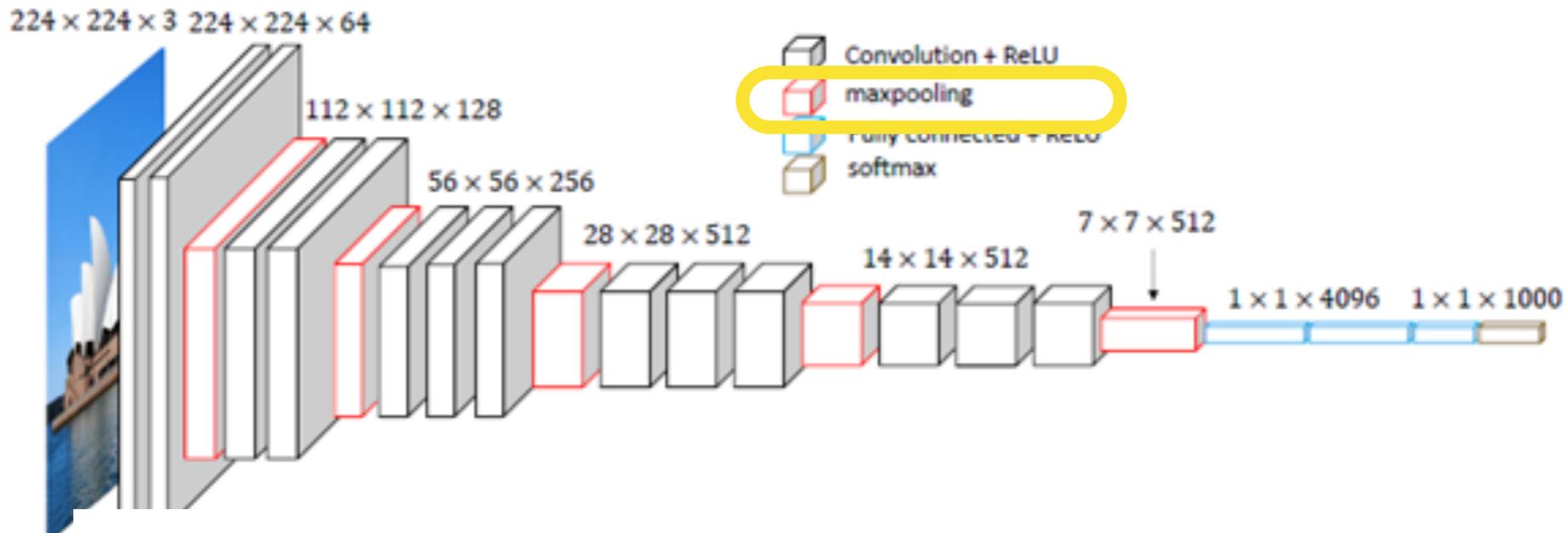
An example of max-pooling operator



Pooling downsamples the input layer



Basics of DL models



Elementary components

Convolution layers

Activation layers

Pooling layers

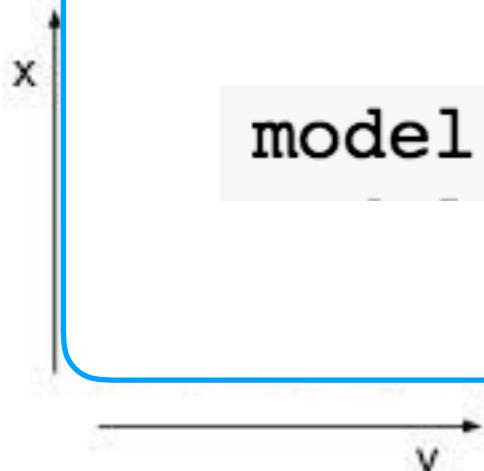
Dense layers

An example of max-pooling operator

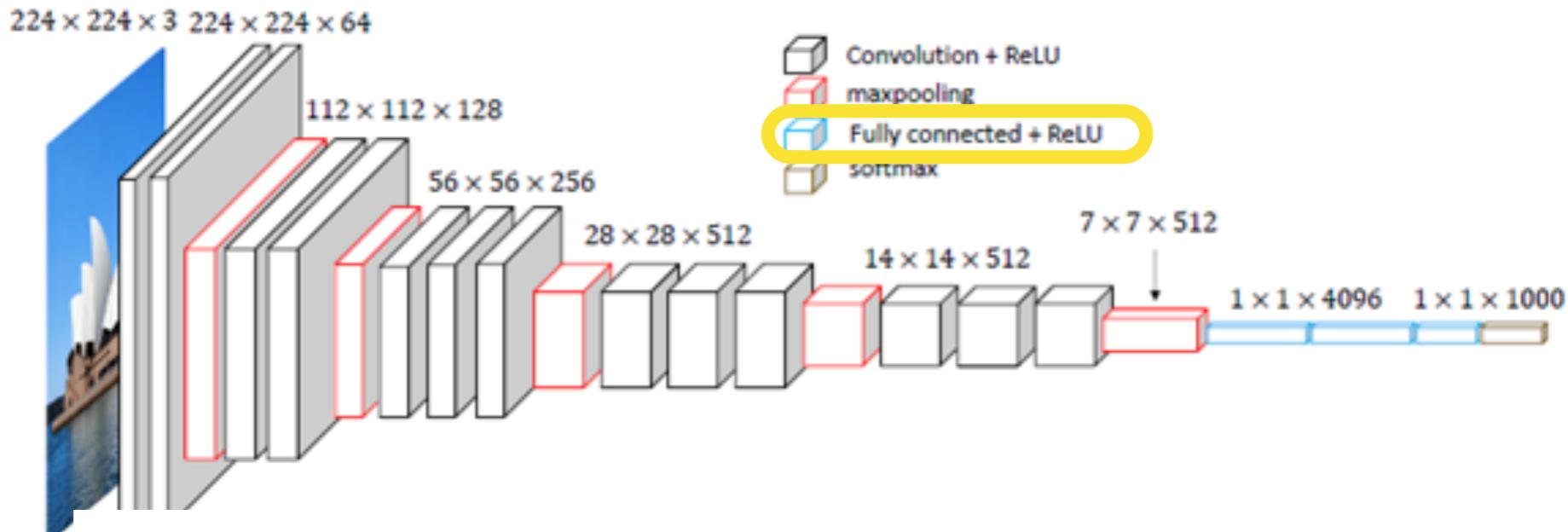
Pooling downsamples the input layer

Keras function

```
model.add(MaxPooling2D(pool_size=(2, 2)))
```



Basics of DL models



Elementary components

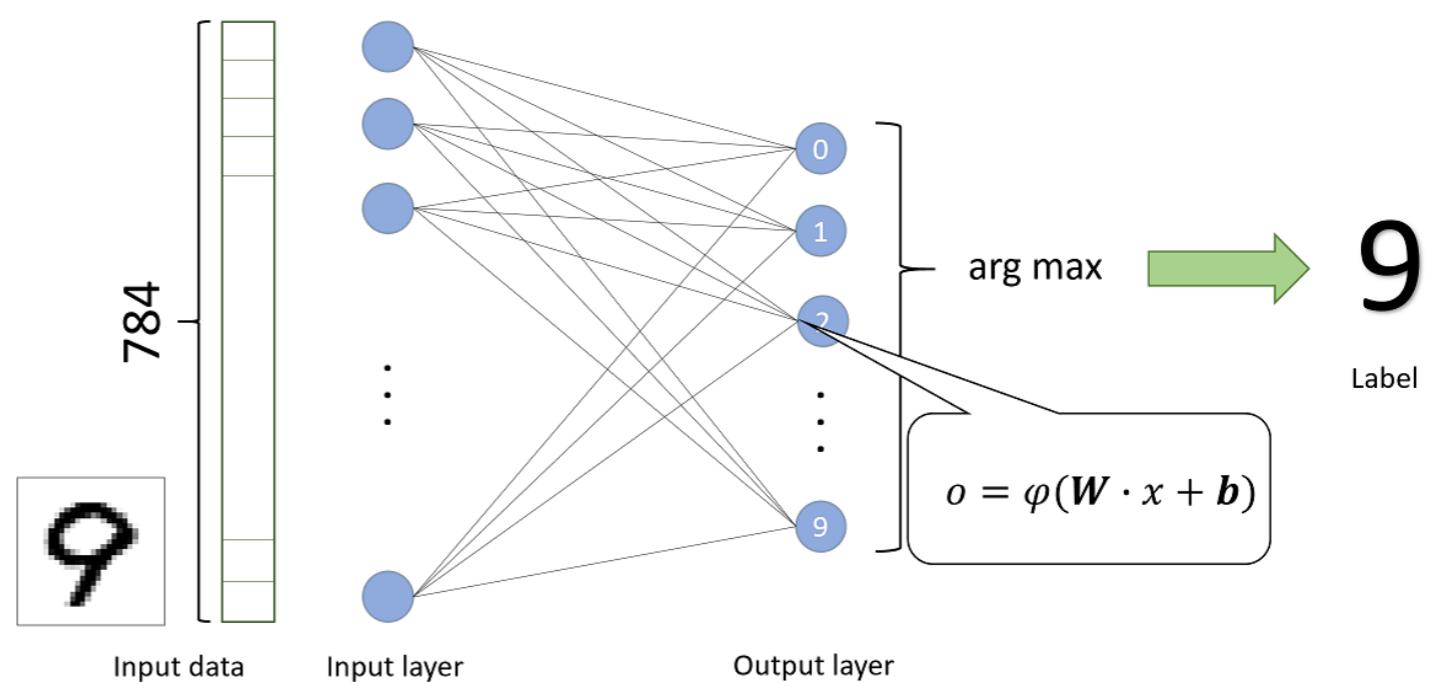
Convolution layers

Activation layers

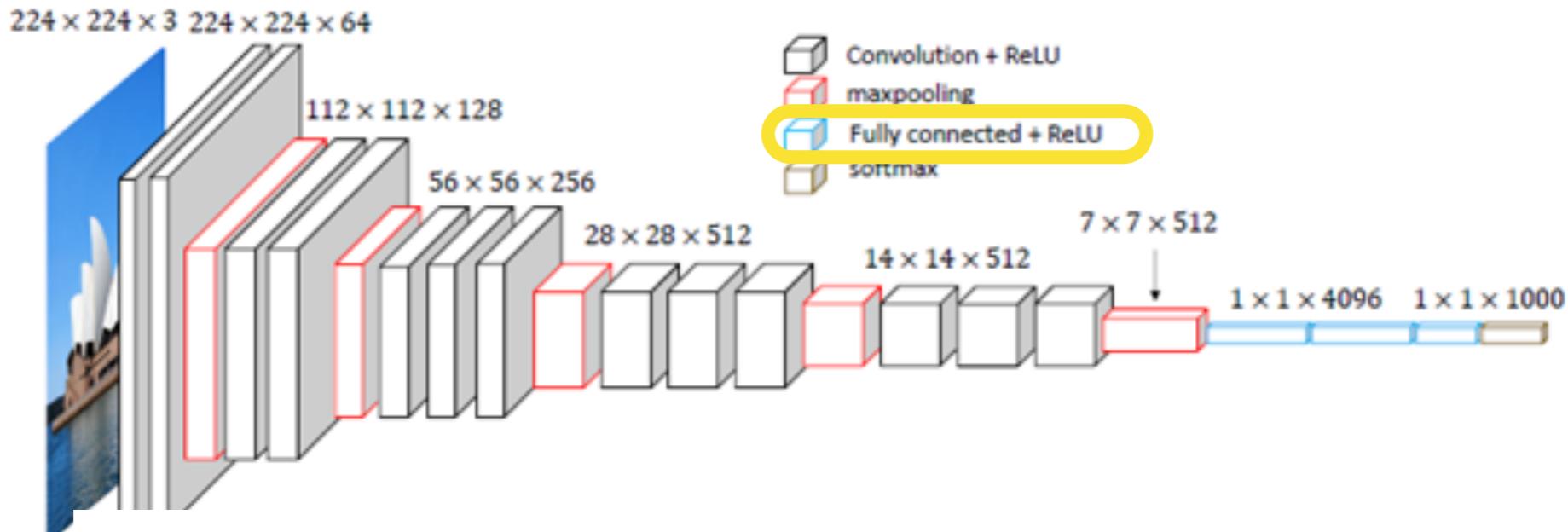
Pooling layers

Dense layers

Dense layers
or
Fully-connected (FC) layer
as in a classic MLP



Basics of DL models



Elementary components

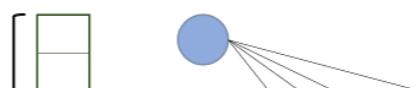
Convolution layers

Activation layers

Pooling layers

Dense layers

Dropout



Keras function

```
model.add(Dense(128, activation='relu'))  
model.add(Dense(num_classes, activation='softmax'))
```



Input data

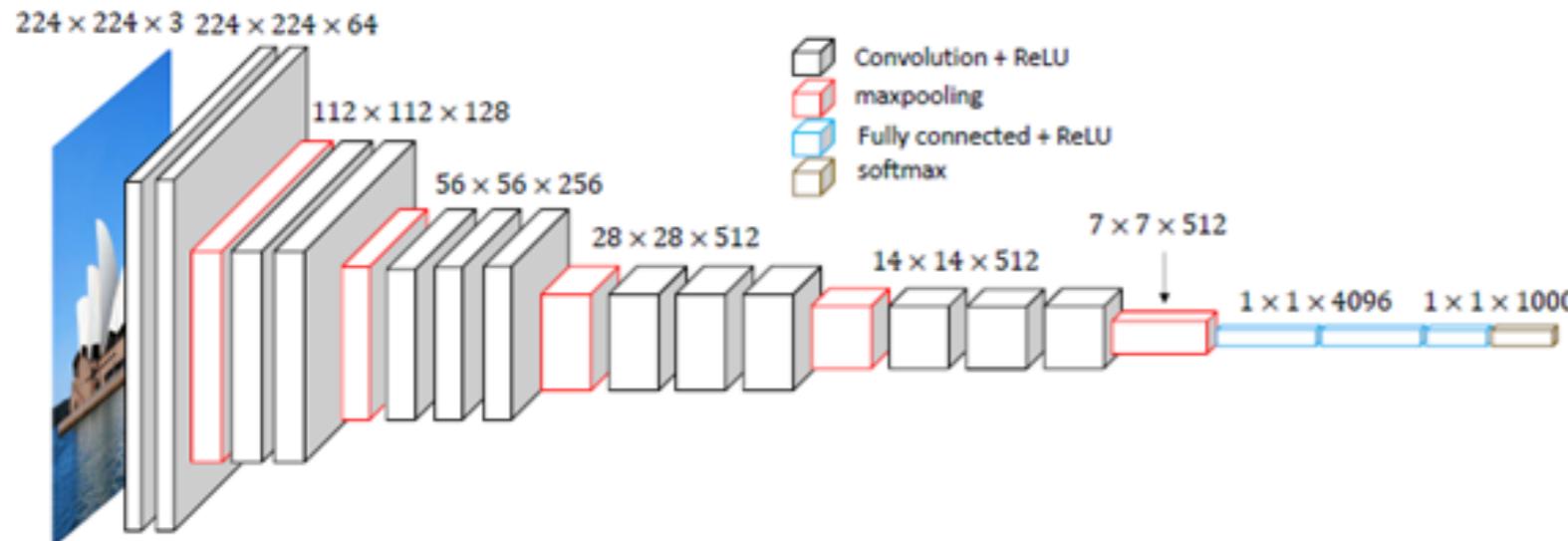


Input layer



Output layer

Examples of DL models for object recognition



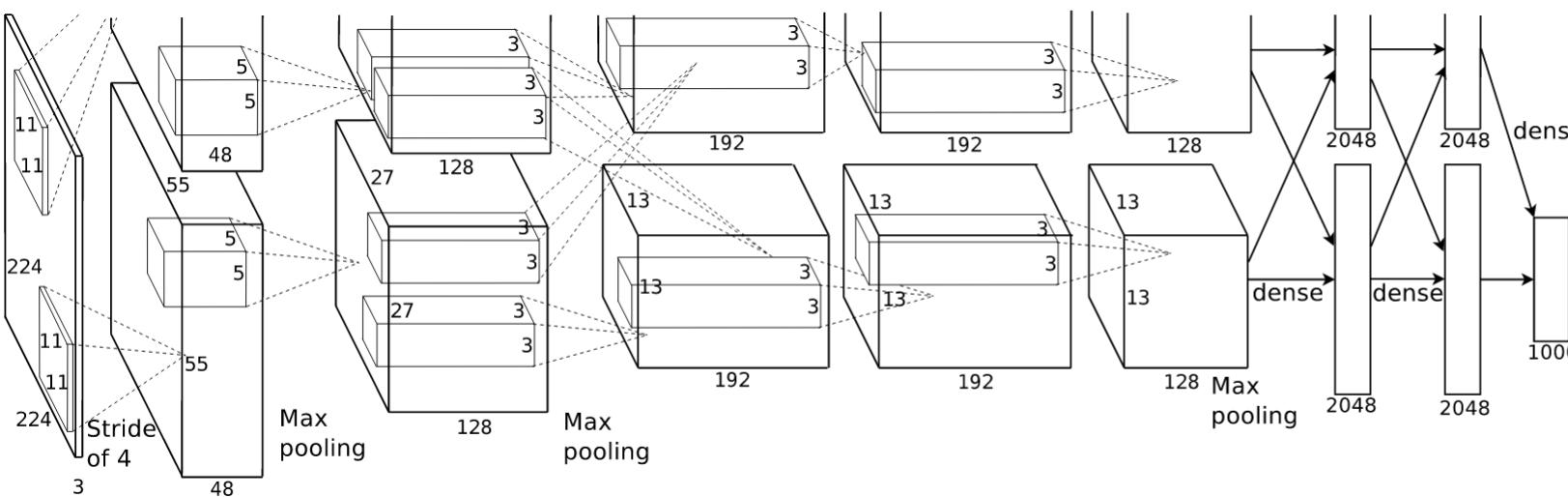
VGG16
($<100M$ of parameters)

Google inception

($5M$ of parameters)



Convolution
Pooling
Softmax
Other



AlexNet
($60M$ of parameters)

A short tour of the NNs zoo

2 (at least) key aspects

NN architecture (parameterization of model f)
Training loss

Autoencoder and dimensionaly reduction

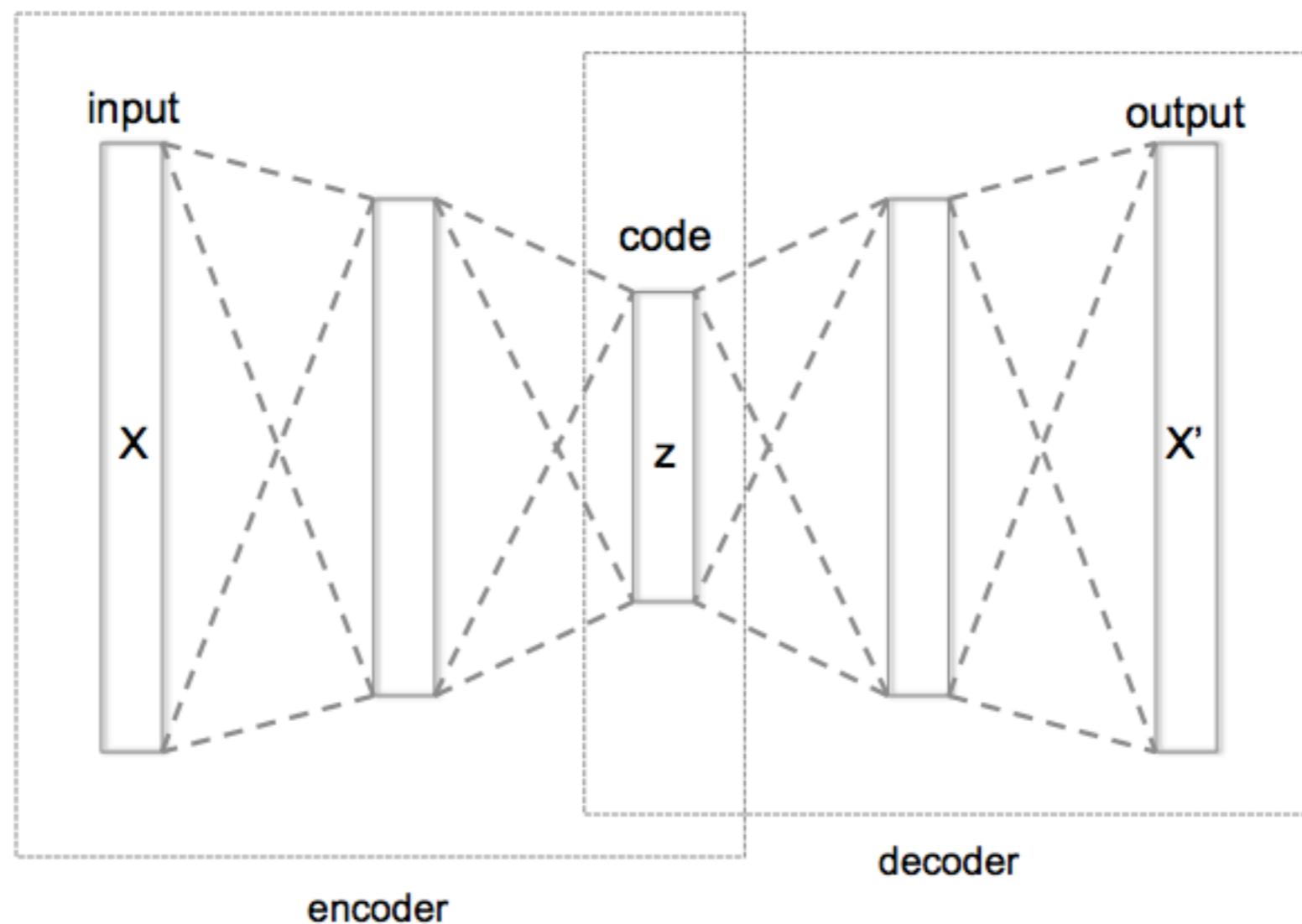
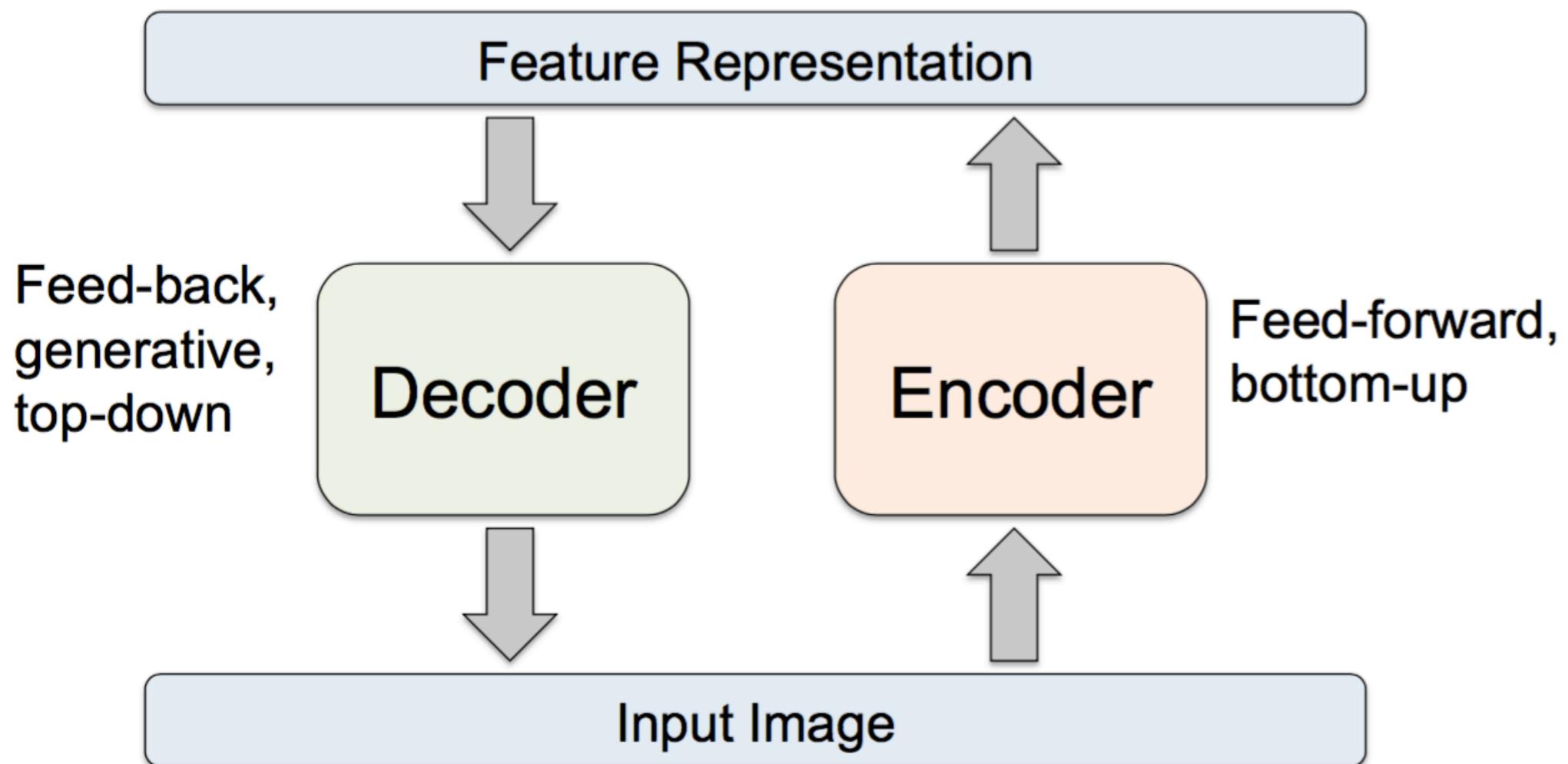


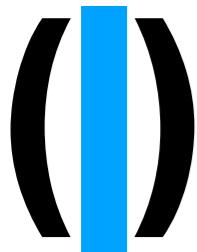
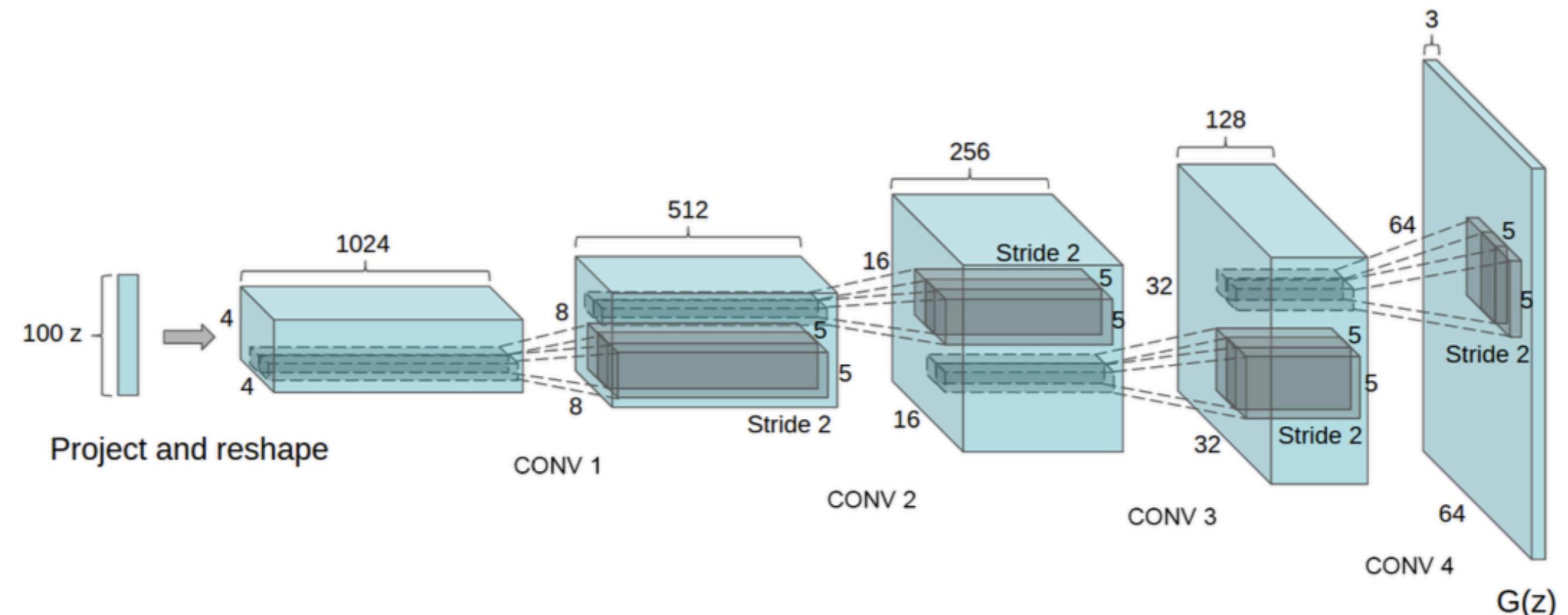
Figure from wikipedia

Autoencoders



- Details of what goes inside the encoder and decoder matter!
- Need constraints to avoid learning an identity.

Generative Models (e.g., GAN)



Generative Modeling



man
with glasses



man
without glasses



woman
without glasses



woman with glasses



Radford et al. 2015

NNs for ODE

$$\frac{dx(t)}{dt} = \sigma (y(t) - x(t))$$

$$\frac{dy(t)}{dt} = x(t) (\rho - z(t)) - y(t)$$

$$\frac{dz(t)}{dt} = x(t) y(t) - \beta z(t)$$

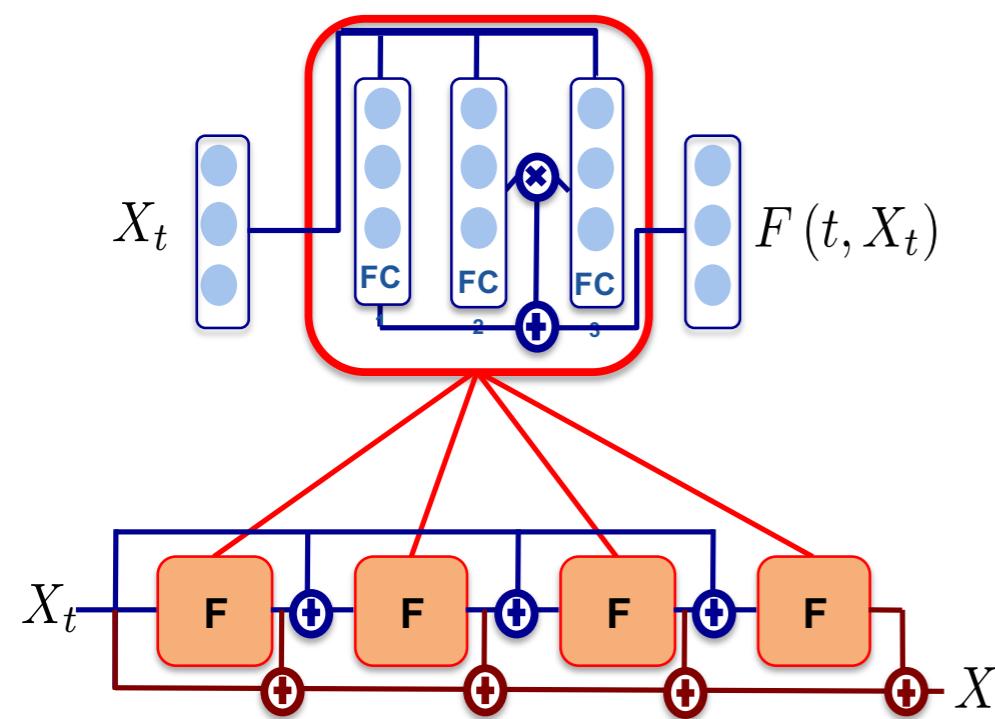
Lorenz-63 equations

Two key steps:

- NN parameterisation for operator F

$$d_t X_t = F_\theta(X_t)$$

- NN parameterization for the associated integration scheme



Adjoint operator:

- Adjoint of the forward integration scheme
- Backward NN integration of Adjoint of the forward integration scheme

Losses & training strategies

Synoptic (point-wise) losses

$$\arg \min_{\theta} \sum_i \|y_i - f_{\theta}(x_i)\|_p^p$$

Likelihood losses

$$\arg \min_{\theta} \sum_i (y_i - f_{\theta}(x_i))^t \Sigma_{\theta}^{-1} (y_i - f_{\theta}(x_i)) - 1/n \log |\Sigma_{\theta}|$$

Distribution-based cost (Optimal transport)

Generative Adversarial Networks

How to deal with situations where one cannot define an explicit criterion to train a generative model but numerous examples are available ?

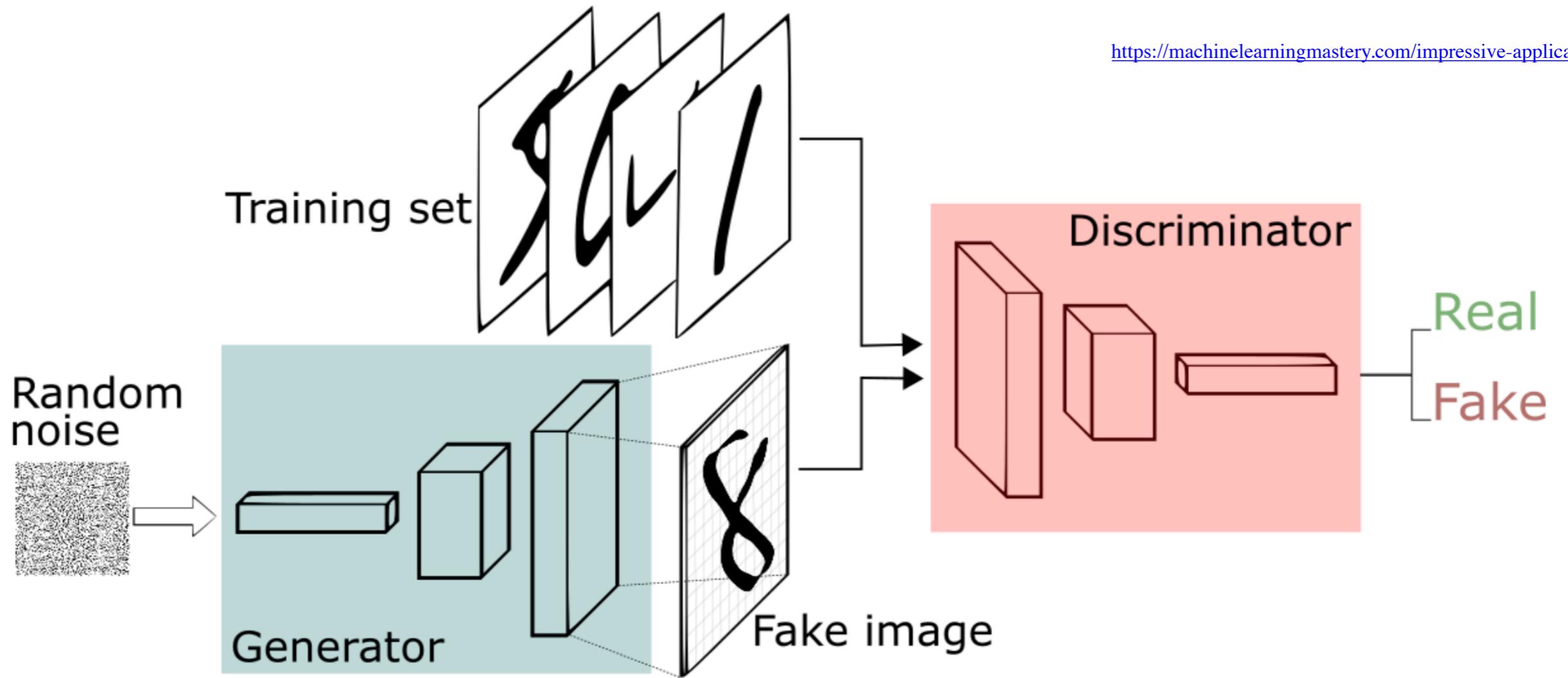


Image credit: Thalles Silva

Generative Adversarial Networks

Image processing examples

<https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/>