Master Executive di II Livello

BIG DATA ANALYSIS AND BUSINESS INTELLIGENCE

Deep Learning, Consciousness and AI

Nome Docente

Francesco Pugliese, PhD

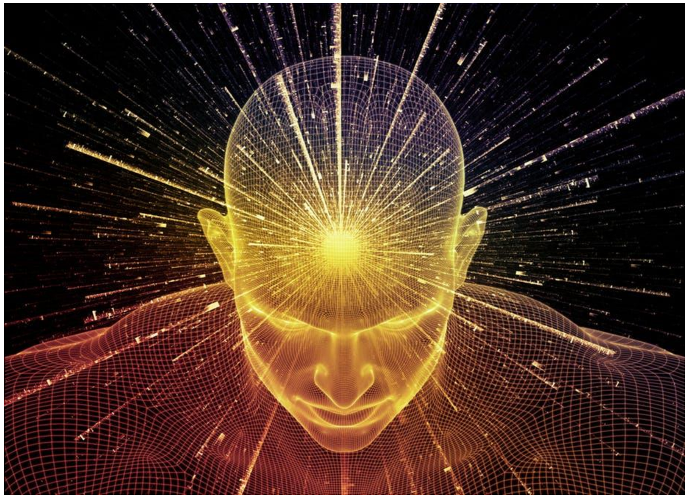
Italian National Institute of Statistics, Division "Information and Application Architecture", Directorate for methodology and statistical design

Email **Francesco Pugliese** : francesco.pugliese@istat.it

WHAT IS CONSCIOUSNESS?

WHAT IS THE SUBJECTIVE EXPERIENCE? NOBODY KNOWS

Cogito Ergo Sum (René Descartes)



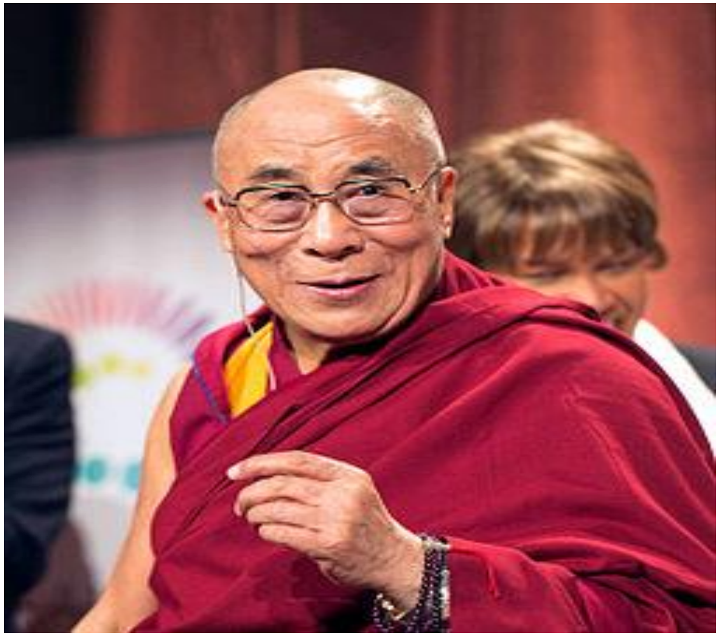
DISSOCIATION BETWEEN CORTICAL ACTIVITY AND COUSCIOUSNESS

- During episodes of unconsciousness, the child's brain is by no means inactive; it is actually hyperactive.
- The observation of similar firing rates between waking and slow-wave sleep represents both the most striking dissociation between cortical activity and cousciousssness.
- It indicates that the presence of "normal" firing levels in the cerebral cortex is not a sufficient criterion for consciousness.

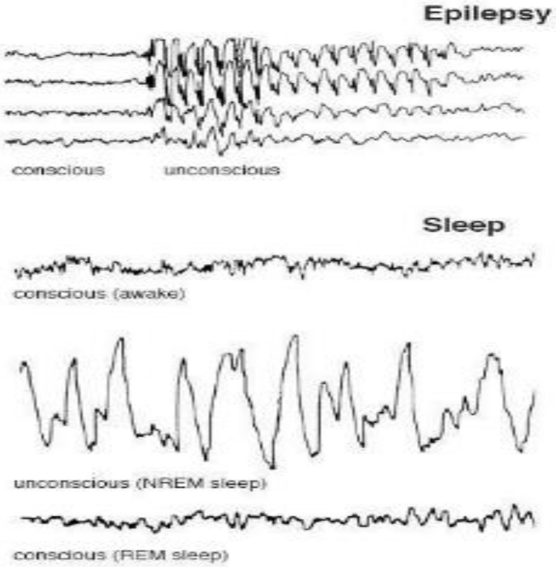
Edelman, Gerald. A Universe Of Consciousness: How Matter Becomes Imagination

WHO COMES FIRST? BRAIN OR THOUGHTS

How Thoughts can change the Brain



Conscious Experience requires PATTERNS OF NEURAL ACTIVITY THAT ARE HIGHLY DIFFERENTIATED (Gerald Edelman



Tenzin Gyatso: “What if the brain comes from consciousness instead of consciousness coming from the brain?”

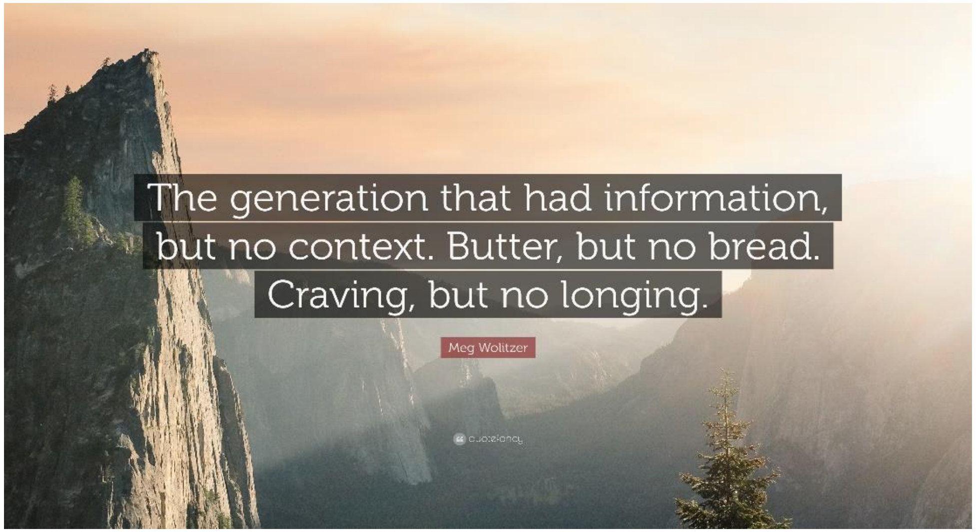
HUMAN BRAIN VS CONSCIOUSNESS

An High Dimensional and Complex Neural Network

Scientists find evidence of a Multidimensional Universe within the Brain

Reimann, M. W., Nolte, M., Scolamiero, M., Turner, K., Perin, R., Chindemi, G., ... & Markram, H. (2017). Cliques of neurons bound into cavities provide a missing link between structure and function. Frontiers in computational neuroscience, 11, 48.

DEEP LEARNING AND A.I. LIMITS

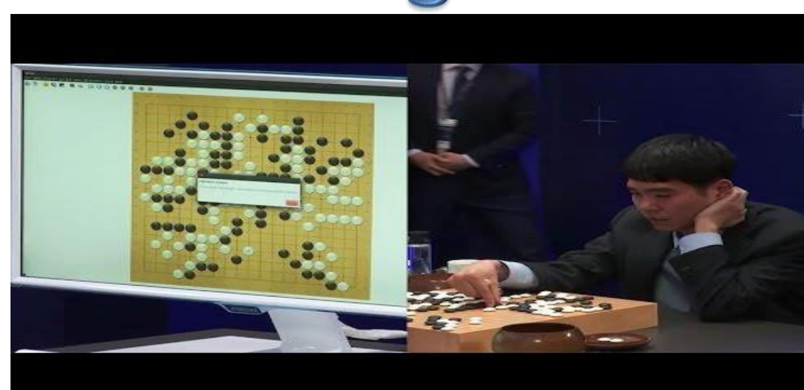


Is this a Glass or a Pot?

Where is the Human-Like Thoughts Flow?

Why and How Lea Seadol beats Alpha Go Zero once?

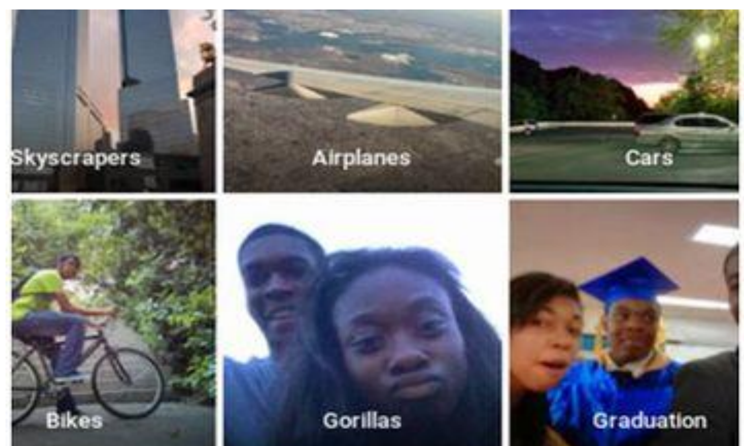
Even though all metrics say it is impossible.



DEEP LEARNING AND A.I. LIMITS

Will ever Self Driving Car cope with Mumbai Traffic?

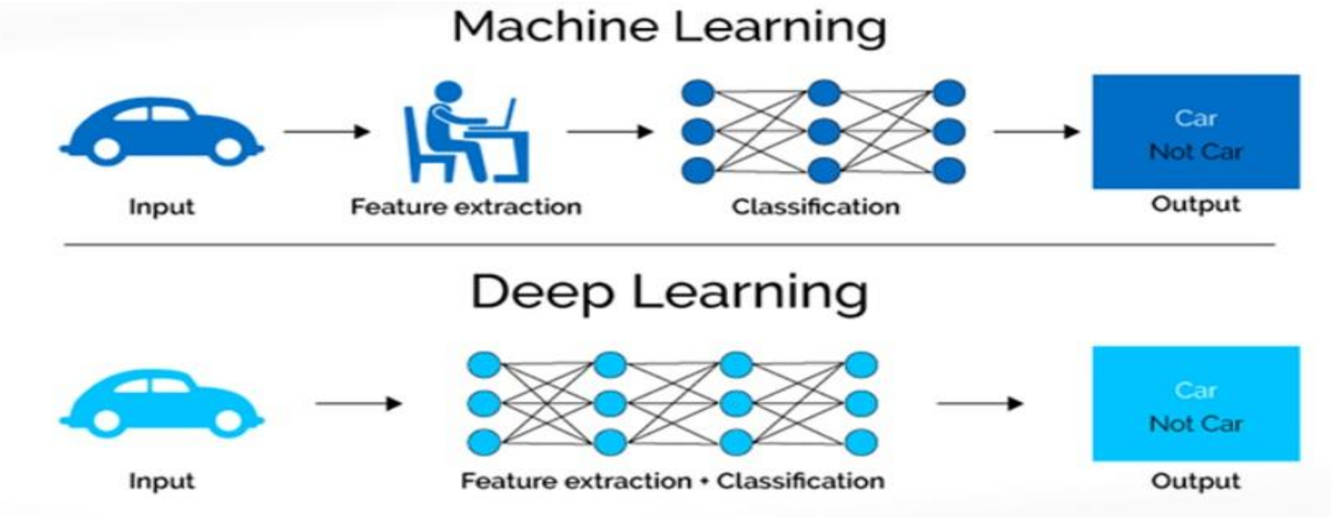
Will ever AI have context information ?



What is Deep Learning ?

Deep Learning refers to algorithms that automatically 'model' high-level abstractions in data

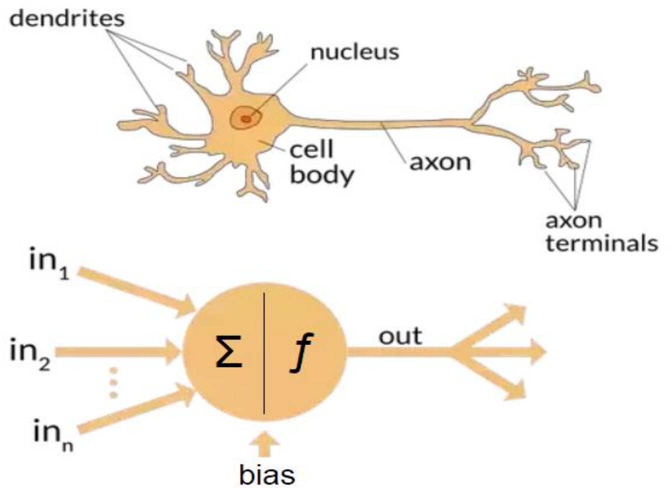
i. here 'model' means: define, find, recognize and exploit
ii. here 'automatically' means: directly from data, without hinging upon handcrafted, task-specific features.



ARTIFICIAL NEURAL NETWORKS

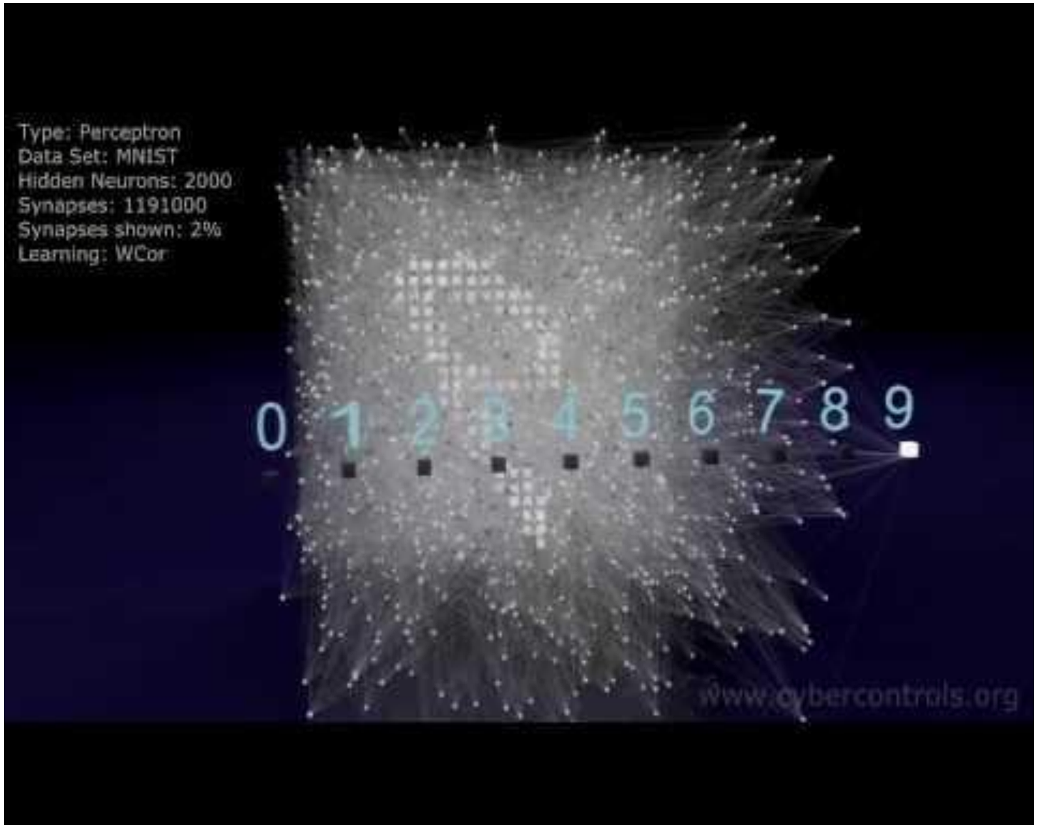
- **Artificial neural networks (ANN) or Connectionist Systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains (van Gerven & Bohte, 2018)**

Biological Neuron Vs Artificial Neuron



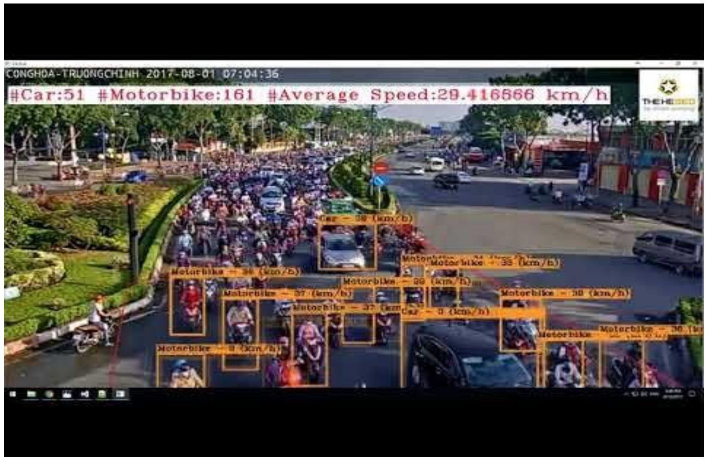
Neuron Activation

$$y_j = \Phi\left(\sum_{i=1}^{N} w_{ij} X_i - \theta_i\right)$$

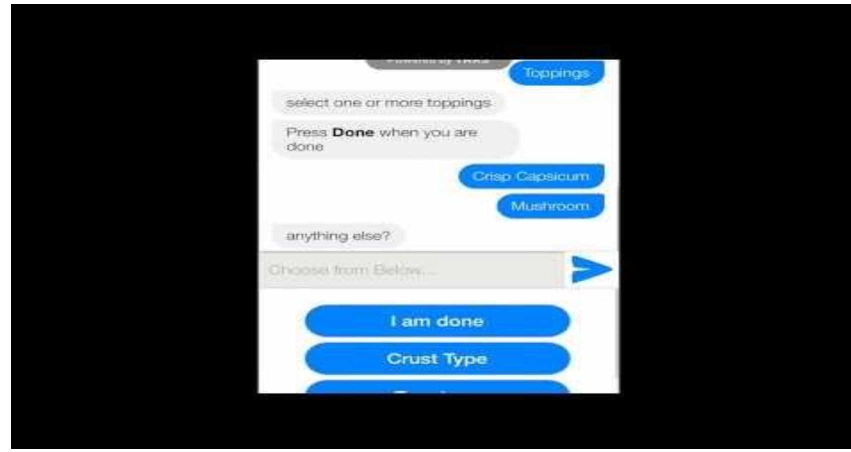


DEEP LEARNING BREAKTROUGHTS

Computer Vision



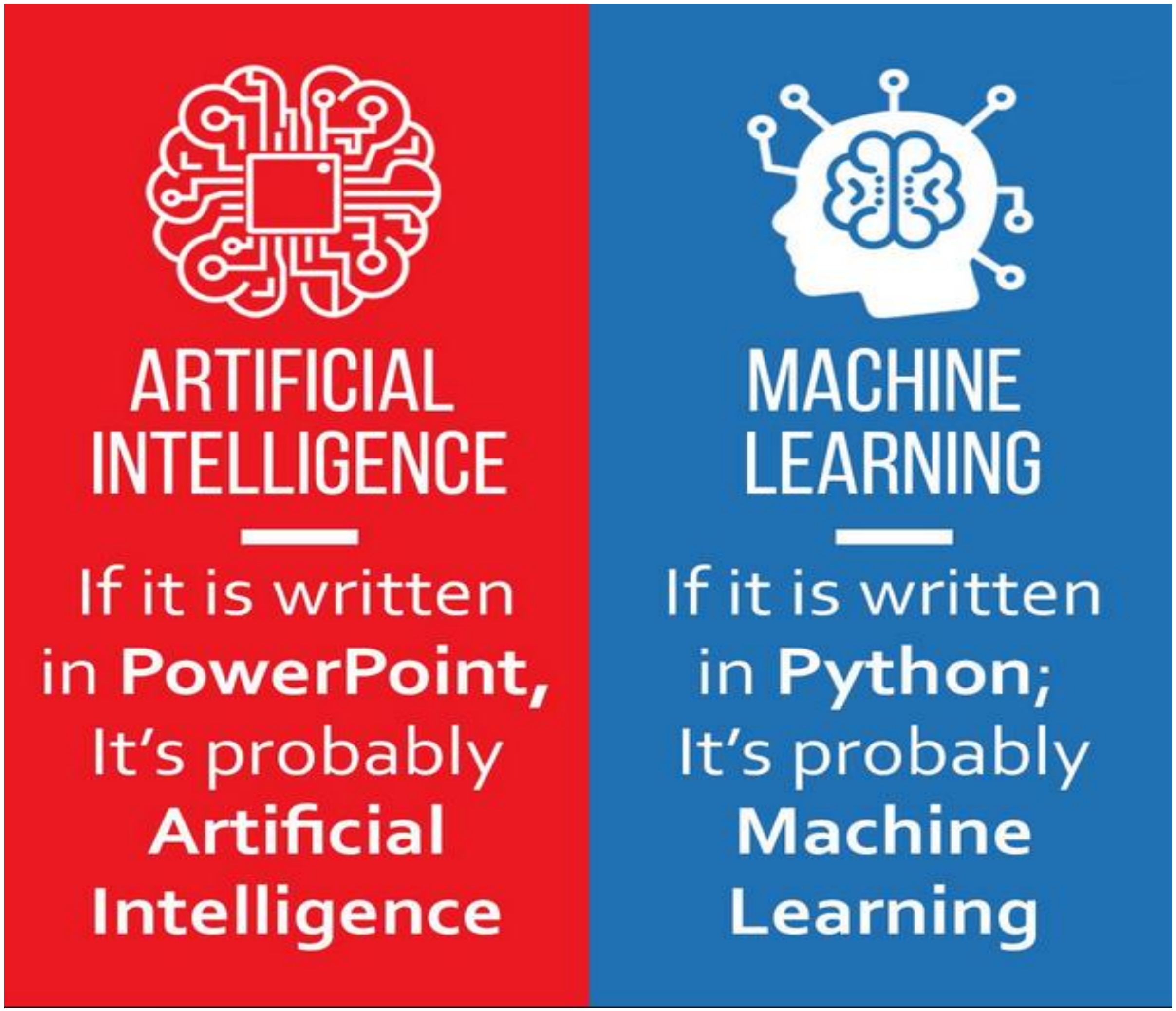
Natural Language Processing



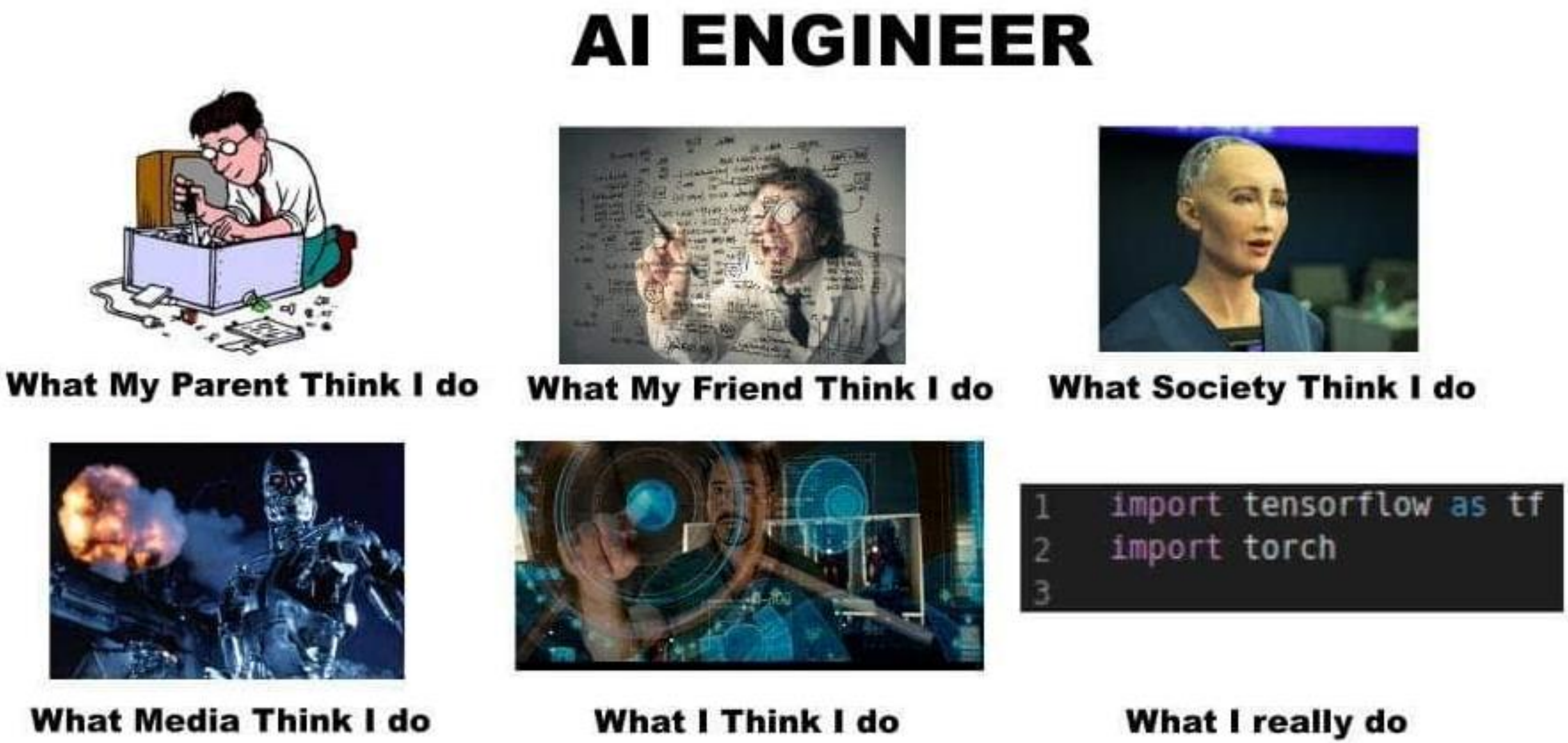
A truly general Artificial Intelligence

A Generative Artificial Intelligence

Difference between AI and ML



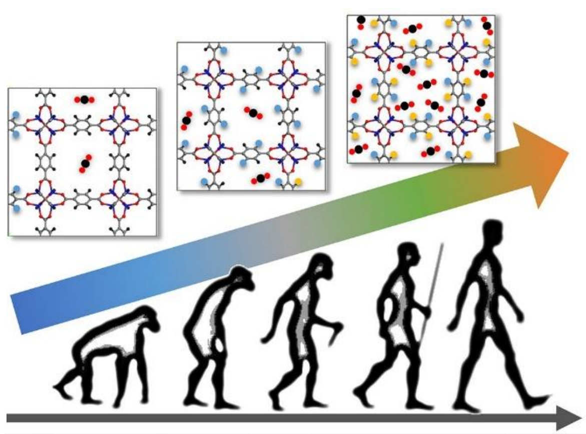
Artificial Intelligence Engineer



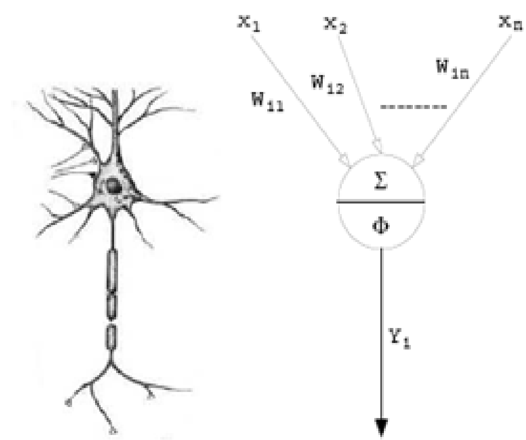
One traditional application ANN: Evolutionary robotics

In order to overcome the problems associated with the robotic system decomposition of traditional approaches (i.e. behavior-based robotics), evolutionary robotics can be used, where the robotic system is able to self-organize

[*Nolfi, S., Floreano, D ., 2000*].



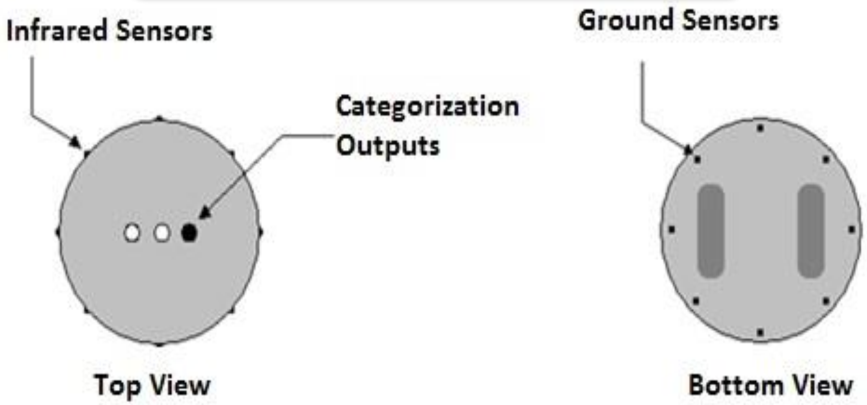
NeuroGenetic Algorithm



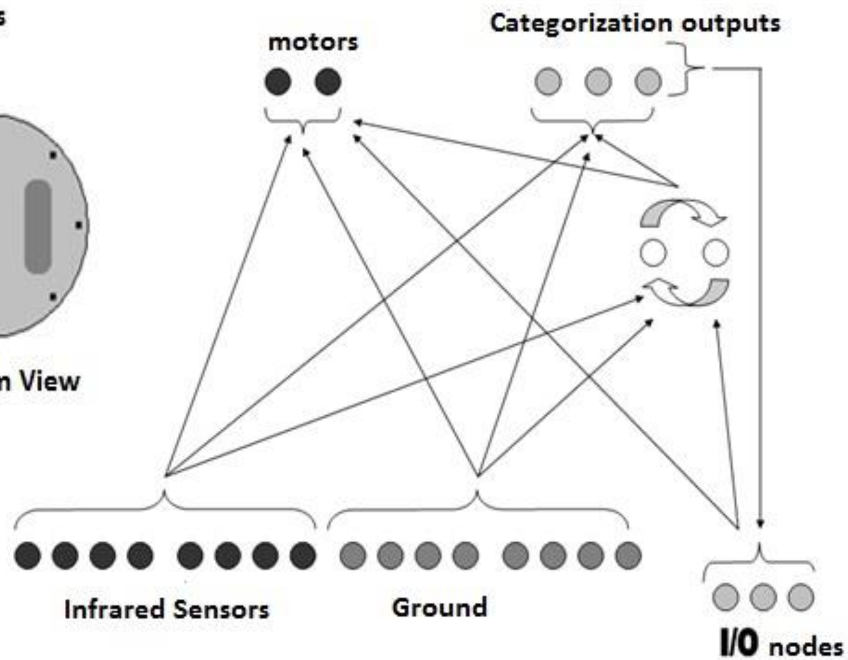
Control Neural Network

EXPERIMENTAL SETUP

ROBOT'S CHASSIS



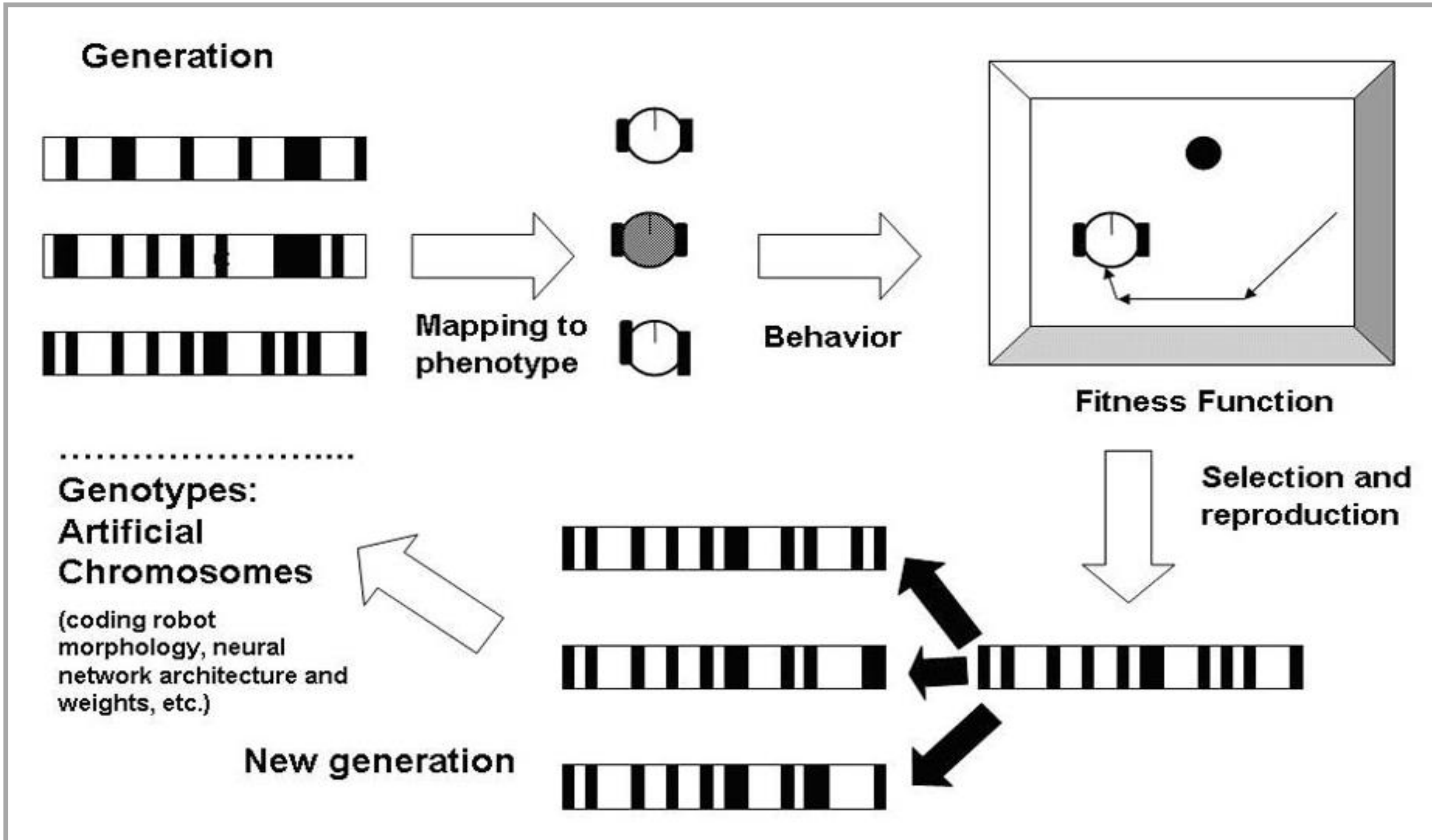
NEURAL NETWORK



LEAKY Activation

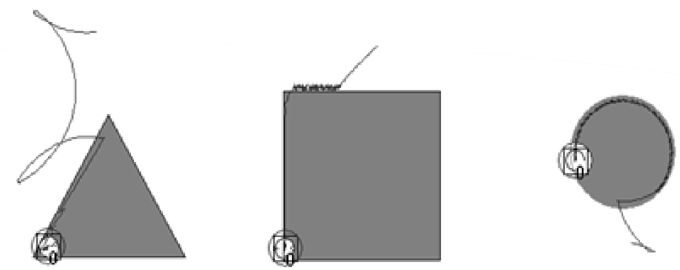
$$A_j = t_j + \sum \mathrm{w_{ij}O_i}\ ,\ O_j = \delta_j O^{t-1} + (1 - \delta_j)\left(1 + \frac{1}{e^{A_j}}\right), 0 \le \delta_j \le 1$$

Evolutionary Robotics

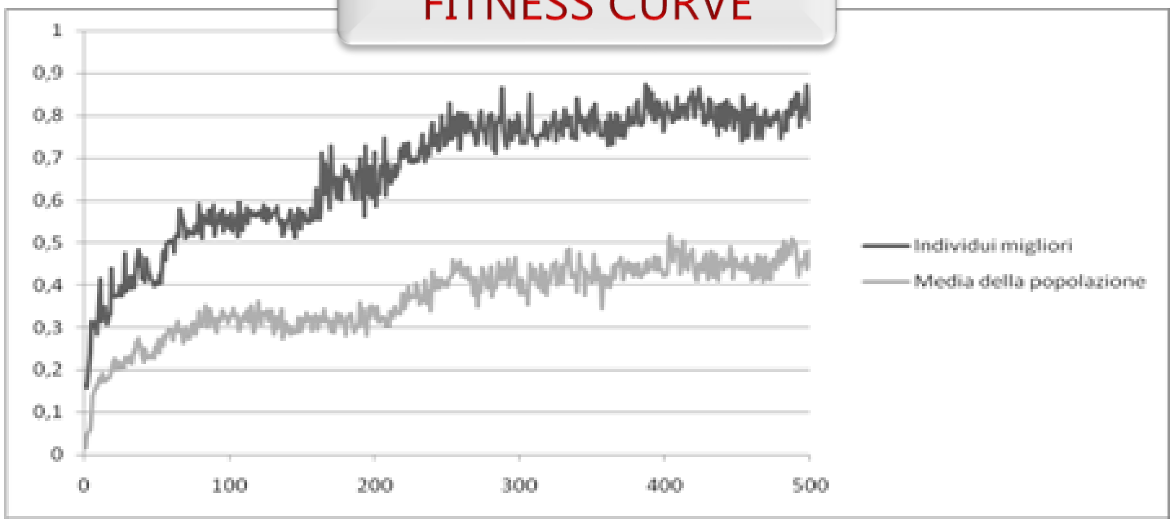


EXPERIMENTAL SETUP N.1

RESULTS



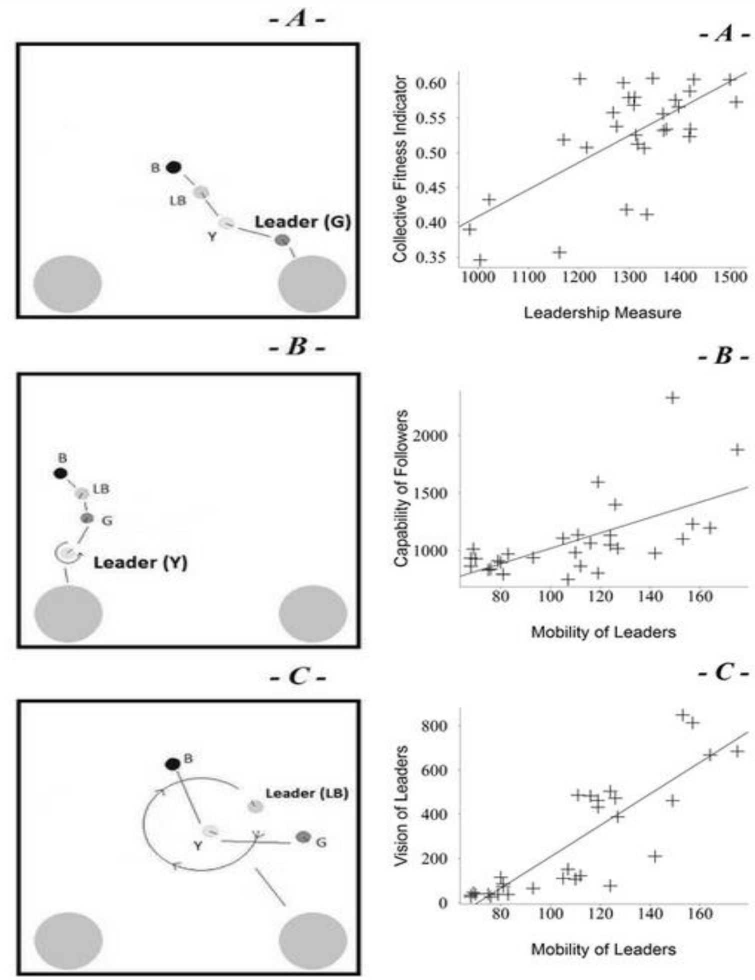
FITNESS CURVE



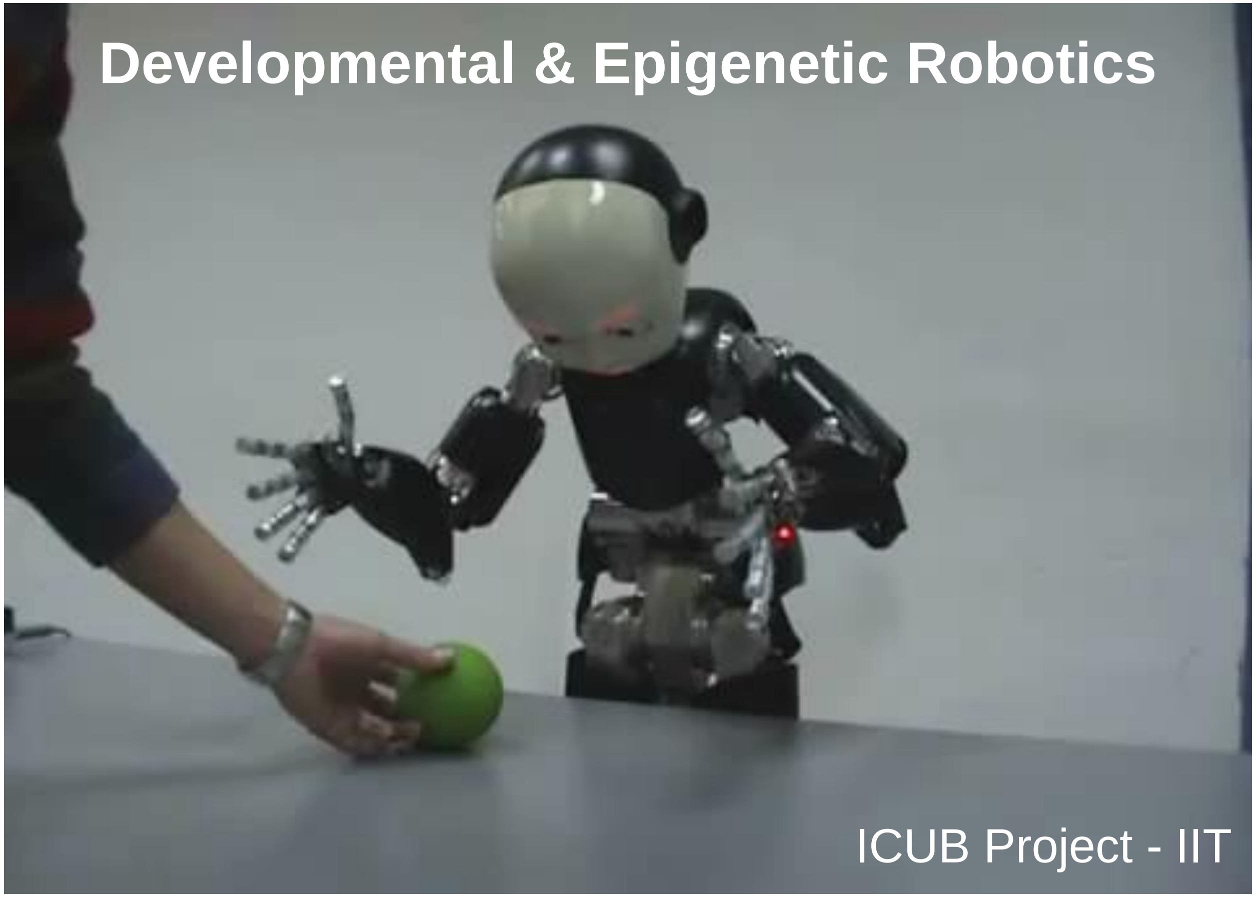
Emergence of Leadership in Robots

- Behavioural and quantitative analysis indicate that a form of leadership emerges
- Groups with a leader are more effective than groups without.
- The most skilled individuals in a group tend to be the leaders.
- Further analysis reveals the emergence of different “styles” of leadership (active and passive).

- A - Passive Leadership. - B - Weak Active Leadership.
- C - Strong Active Leadership.

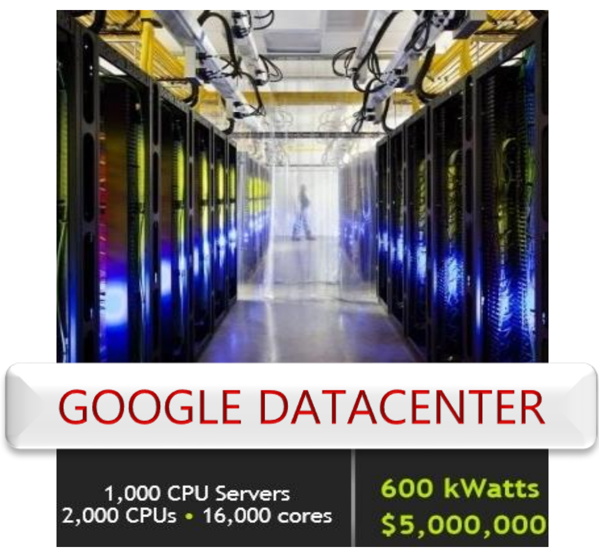


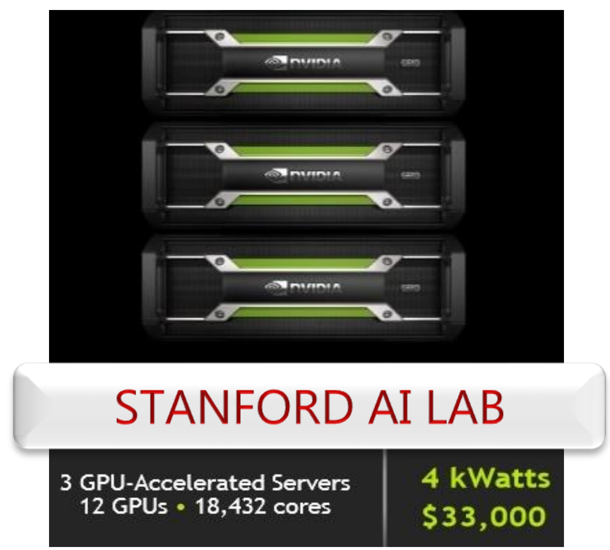
MAY Robotics help to understand social and psychological problems?



DEEP LEARNING: Neural Networks become more effective

In recent years **Deep Neural Networks** have achieved noticeably breakthroughs in research (*Bengio, 2009*). This new methodology dealing with deep neural networks and their training algorithms is called *"Deep Learning"*. **<u>So far, in all the experiments, the resulting performances were many magnitudes better than other machine learning techniques available.</u>**





DEEP LEARNING: a cutting-edge approach to Computer Vision and NLP

GPU

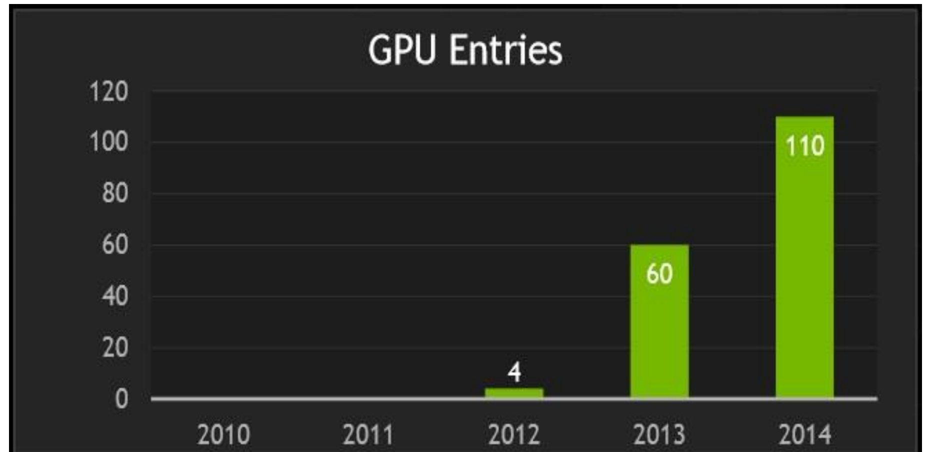
- **The advent of GPUs makes possible the training of very large neural networks with even more than 150 millions of parameters.**

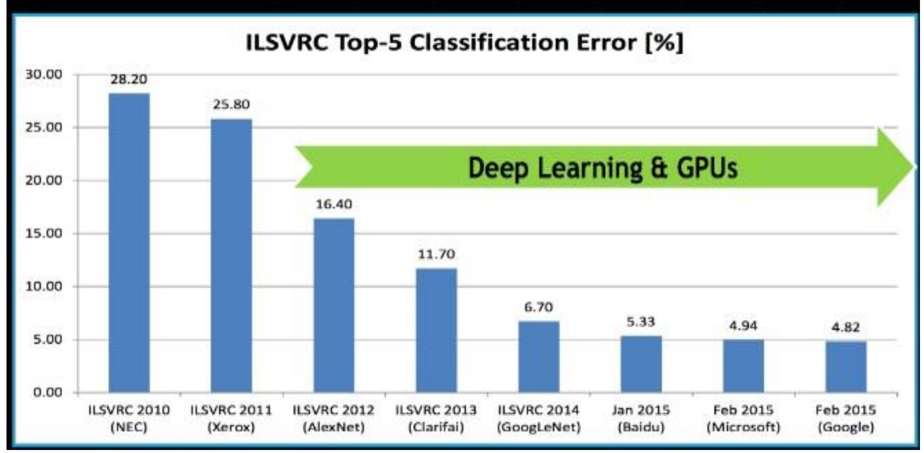
BIG DATA

- **A new generation of larger training and test sets.**

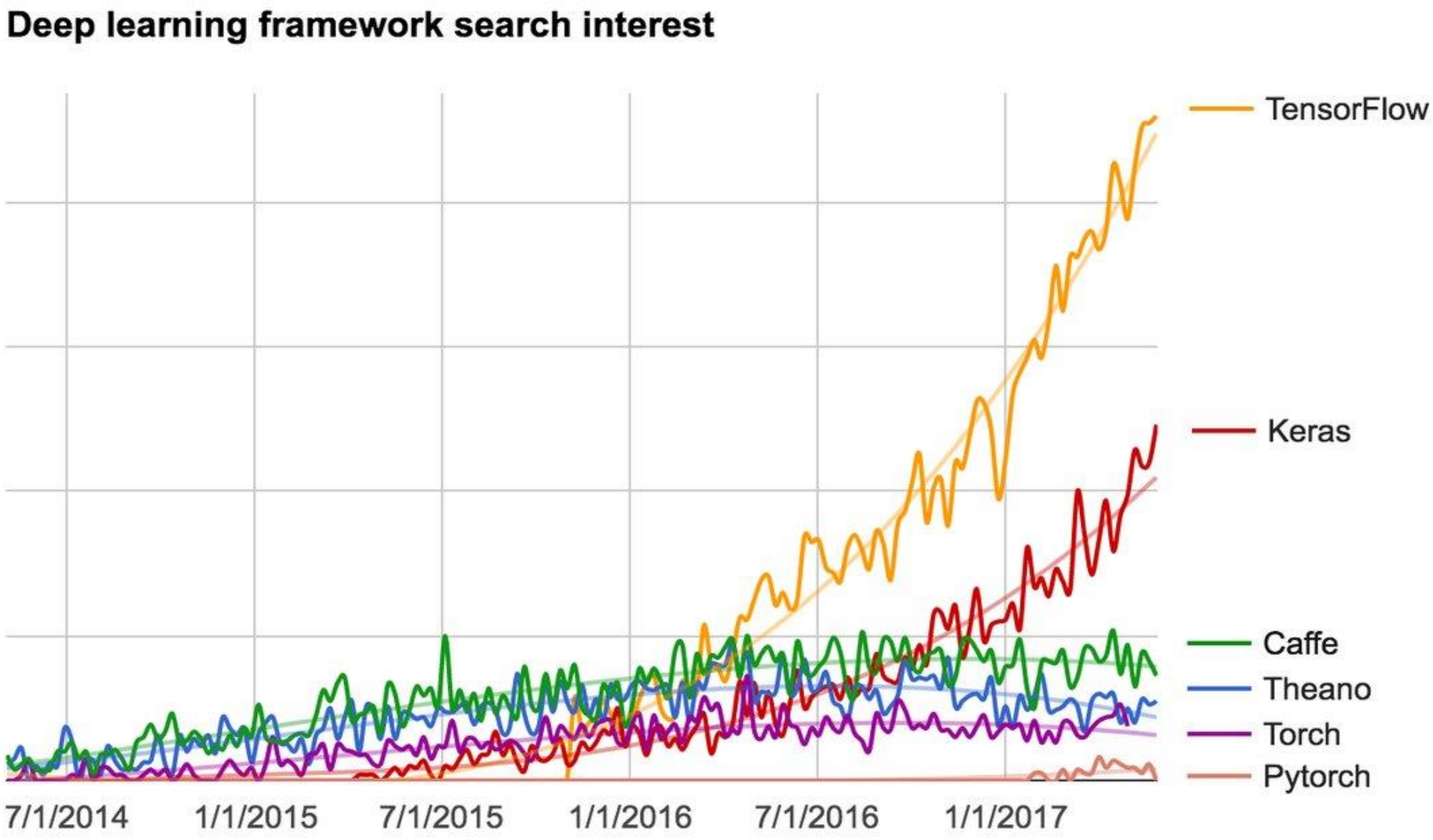
DROPOUT

- **Better model regularization techniques have been discovered such as *“Dropout”* or *“Data Augmentation”***





Evolution of Keras over years



Frameworks FOR DEEP LEARNING

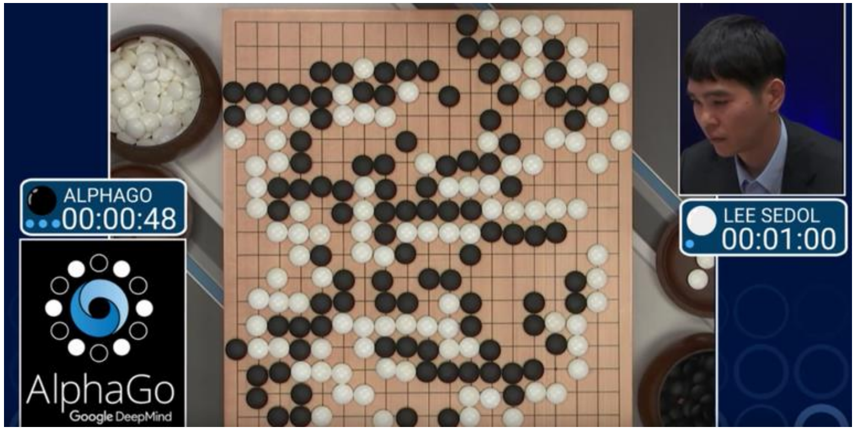
Keras is an higher-level interface for Theano (which works as backend). Keras displays a more intuitive set of abstractions that make it easy to configure neural networks regardless of the backend scientific computing library.

TensorFlow is an open-source software library for dataflow programming across a range of tasks. It is a symbolic math library, and also used for machine learning applications such as neural networks. It is used for both research and production at Google.

PyTorch is an open-source machine learning library for Python, derived from Torch, used for applications such as natural language processing. It is primarily developed by **Facebook's** artificial-intelligence research group, and **Uber's** "Pyro" software for probabilistic programming is built on it.

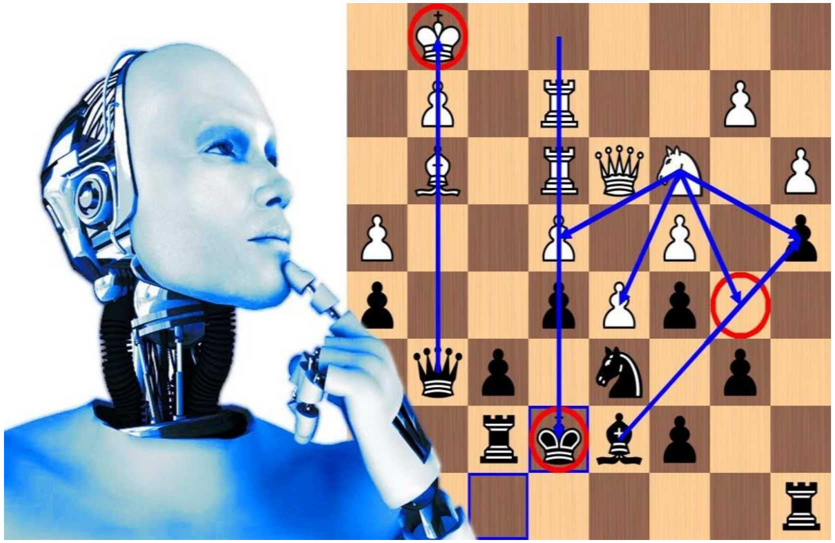
Why Deep Learning over-performed traditional statistics models?

- "Deep Learning" approaches can be **end-to-end trained** without a task-specific feature engineering.
- **These model are scalable**: adding GPUs they can be trained faster.
- **"Deep Learning is killing every problem in AI**" (*Elizabeth Gibney, 2016*)
- **Basically, statistics is not able to deal with very high dimensionalities of data as Deep Learning does**.



Alpha Zero: Mastering the games of Go and Chess without Human Knowledge

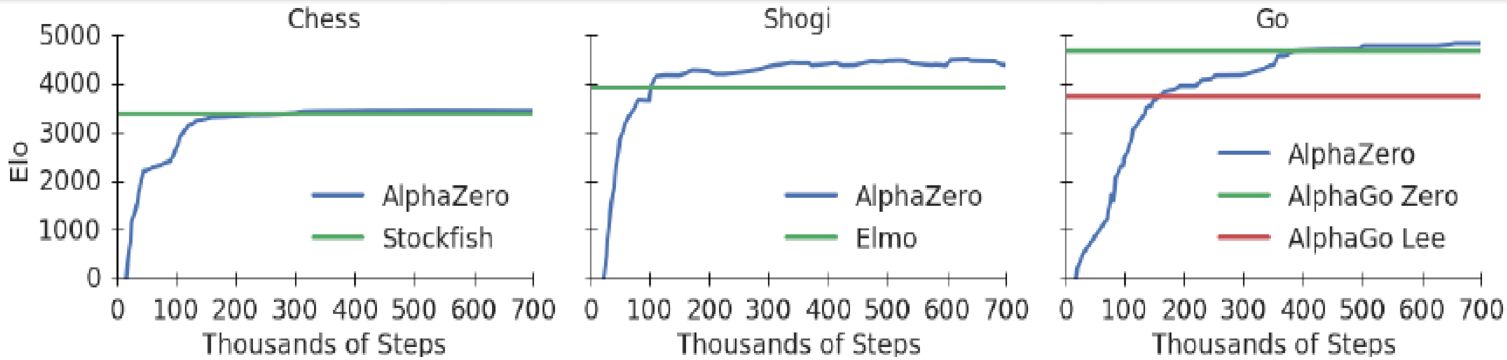
- **In Just 4 Hours, Google's AI Mastered All The Chess Knowledge in History**
- **"I always wondered how it would be if a superior species landed on Earth and showed us how they played chess. Now I know." grandmaster Peter Heine Nielsen.**

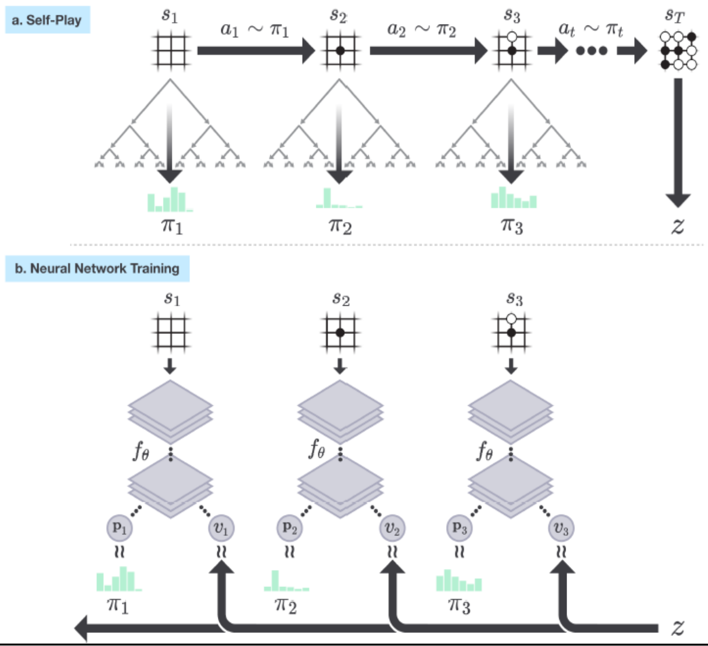


- **Google's AlphaZero Destroys Stockfish In 60 Game Matches**

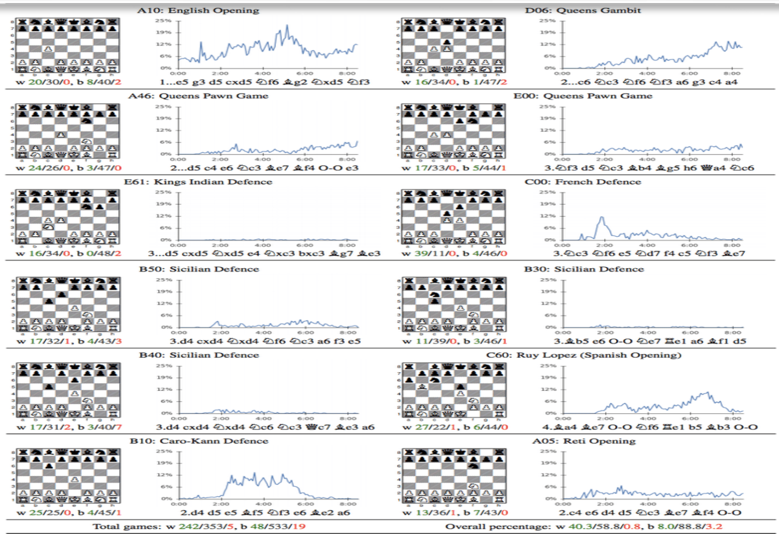
"This algorithm could run cities, continents, universes."
PETER DOCKRILL (Senior Writer)

Alpha Zero IS an Artificial Intelligence, it IS NOT just a Chess Engine..





12 Chess Openings Discovered by Alphazero

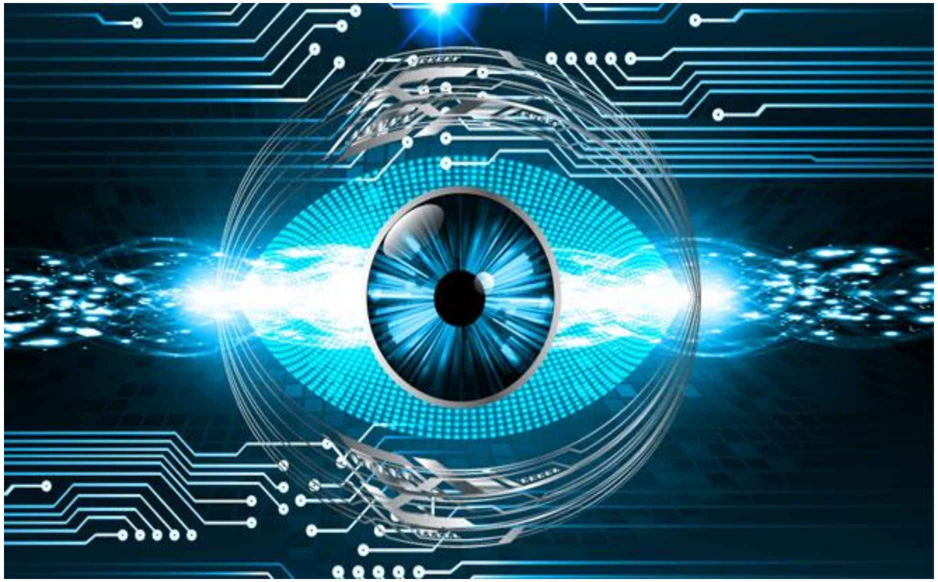


Computer Vision: Where does Traditional Statistics fail?

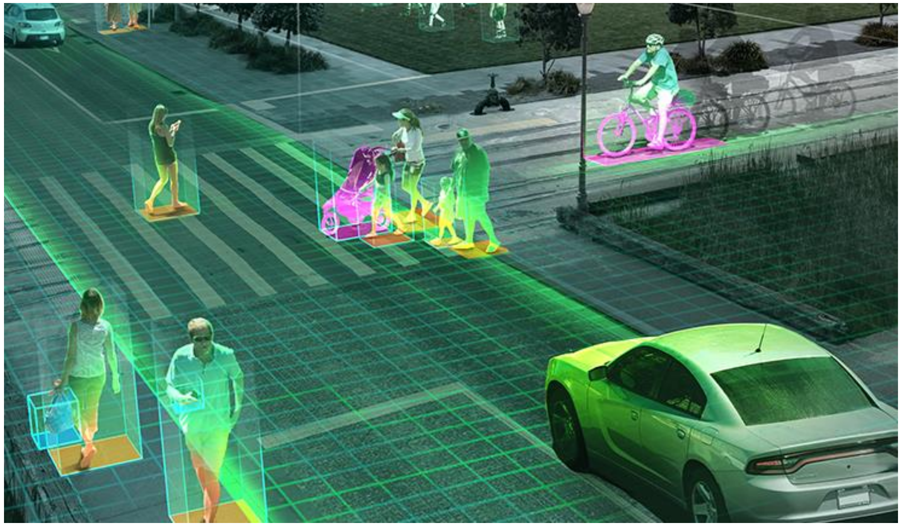
- **Computer Vision** is an interdisciplinary field that deals with the way algorithms can be made for gaining high-level understanding from digital images or videos.

- Statistical methods are not always welcome in computer vision.

- Statistical methods seem not scaling up to the challenges of computer vision problems (*Chellappa, R., 2012*).



Why Does Computer Vision matter so much?



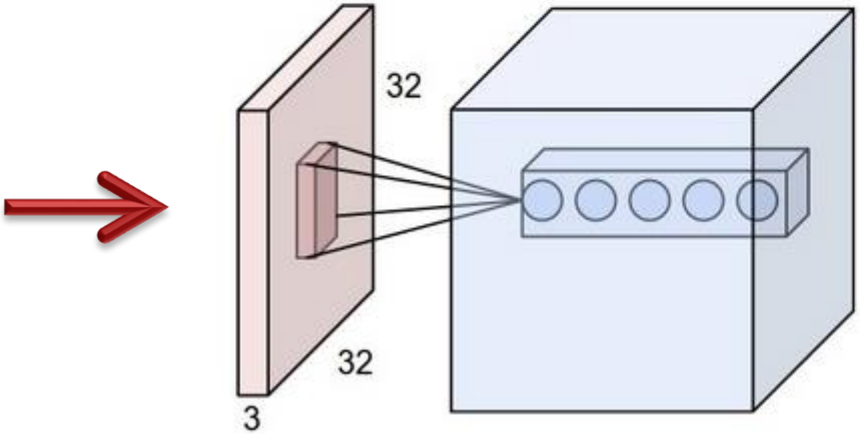
A new generation of machines might accomplish typical human tasks such as recognizing and moving objects, driving cars, cultivating fields, cleaning streets, city garbage collecting, etc.

Convolutional Neural Networks (ConvNets or CNNs)

Convolutional Neural Networks (CNN) are biologically-inspired variants of **MLPs**. We know the visual cortex contains a complex arrangement of cells (***Hubel, D. and Wiesel, T., 1968***). These cells are sensitive to small sub-regions of the visual field, called a *receptive field*. Other layers are: **RELU layer**, **Pool Layer**. Typical CNNs settings are: a) **Number of Kernels (Filters)**, b) **Receptive Field size**, b) **Padding**, c) **Stride.** These parameters are tied by the following equation:

$$(W - F + 2P)/(S + 1)$$

Each neuron in the convolutional layer is connected only to a local region in the input volume spatially. In this case there are 5 neurons along the depth all looking at the same region.

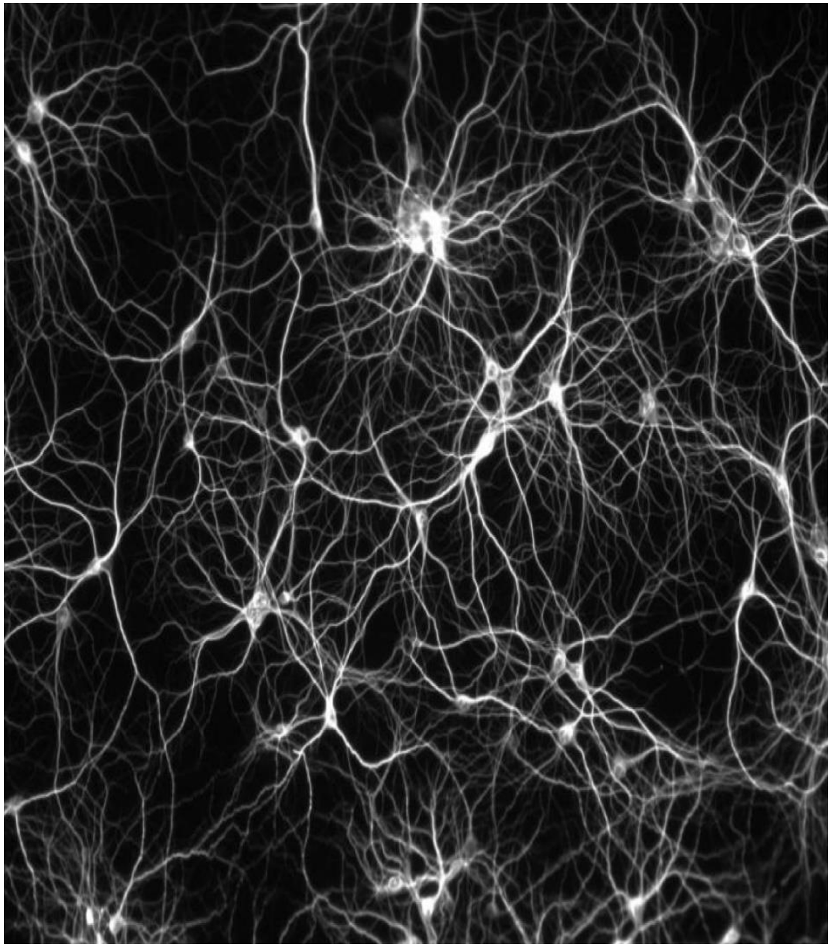


HUMAN BRAIN VS CONSCIOUSNESS

This is a Deep Neural Network with Keras

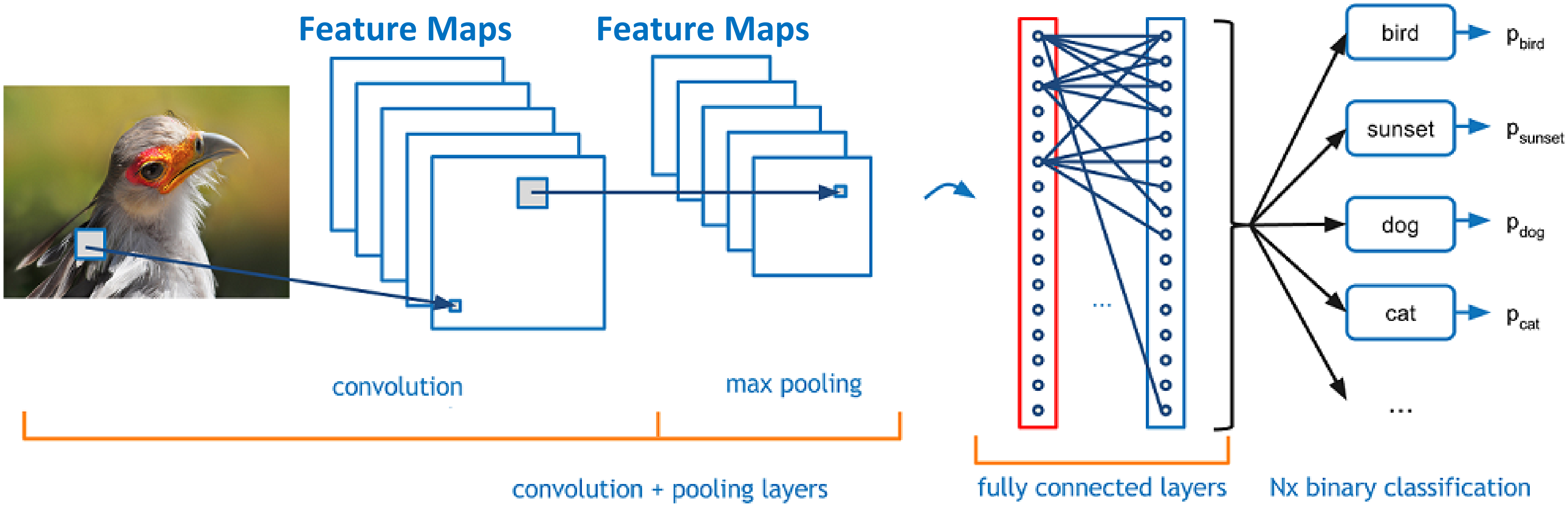
This is a Neuronal Map

```
def get_unet():
    inputs = Input((1,img_rows, img_cols))
    conv1 = Convolution2D(32, 3, 3, activation='relu', border_mode='same')(inputs)
    conv1 = Convolution2D(32, 3, 3, activation='relu', border_mode='same')(conv1)
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)

    conv2 = Convolution2D(64, 3, 3, activation='relu', border_mode='same')(pool1)
    conv2 = Convolution2D(64, 3, 3, activation='relu', border_mode='same')(conv2)
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)

    conv3 = Convolution2D(128, 3, 3, activation='relu', border_mode='same')(pool2)
    conv3 = Convolution2D(128, 3, 3, activation='relu', border_mode='same')(conv3)
    pool3 = MaxPooling2D(pool_size=(2, 2))(conv3)

    conv4 = Convolution2D(256, 3, 3, activation='relu', border_mode='same')(pool3)
    conv4 = Convolution2D(256, 3, 3, activation='relu', border_mode='same')(conv4)
    pool4 = MaxPooling2D(pool_size=(2, 2))(conv4)

    conv5 = Convolution2D(512, 3, 3, activation='relu', border_mode='same')(pool4)
    conv5 = Convolution2D(512, 3, 3, activation='relu', border_mode='same')(conv5)

    up6 = merge([UpSampling2D(size=(2, 2))(conv5), conv4], mode='concat', concat_axis=1)
    conv6 = Convolution2D(256, 3, 3, activation='relu', border_mode='same')(up6)
    conv6 = Convolution2D(256, 3, 3, activation='relu', border_mode='same')(conv6)

    up7 = merge([UpSampling2D(size=(2, 2))(conv6), conv3], mode='concat', concat_axis=1)
    conv7 = Convolution2D(128, 3, 3, activation='relu', border_mode='same')(up7)
    conv7 = Convolution2D(128, 3, 3, activation='relu', border_mode='same')(conv7)

    up8 = merge([UpSampling2D(size=(2, 2))(conv7), conv2], mode='concat', concat_axis=1)
    conv8 = Convolution2D(64, 3, 3, activation='relu', border_mode='same')(up8)
    conv8 = Convolution2D(64, 3, 3, activation='relu', border_mode='same')(conv8)

    up9 = merge([UpSampling2D(size=(2, 2))(conv8), conv1], mode='concat', concat_axis=1)
    conv9 = Convolution2D(32, 3, 3, activation='relu', border_mode='same')(up9)
    conv9 = Convolution2D(32, 3, 3, activation='relu', border_mode='same')(conv9)

    conv10 = Convolution2D(1, 1, 1, activation='sigmoid')(conv9)

    model = Model(input=inputs, output=conv10)
```

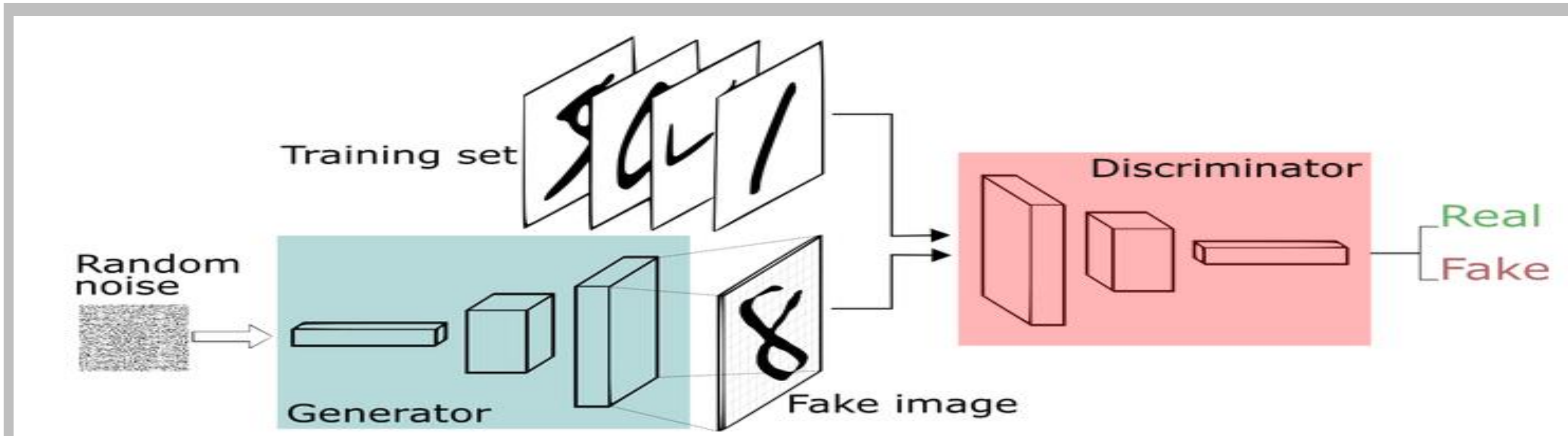


Artificial Intelligence is Just a Tool. An amazing tool to analyze data !!

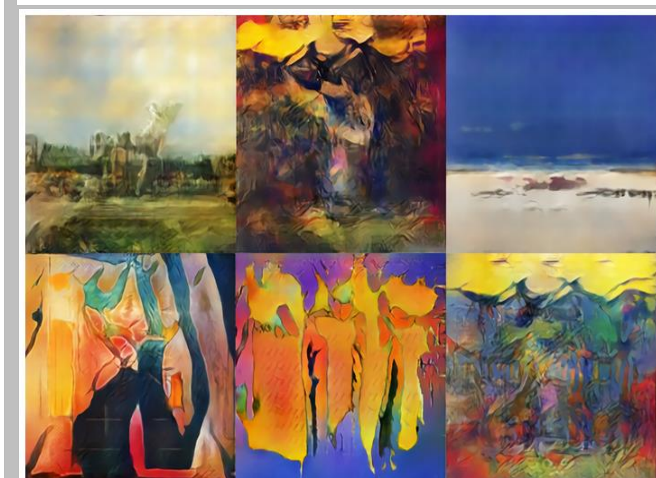
- **CNNs** were initially devised for **Image Recognition,** nowadays very often reach better-than-human accuracy
- **CNNs** need to be fed with **images,** but since for a machine images are just **numeric matrices…**
- **…**they are increasingly being used in **Natural Language Processing,** e.g. **text classification,** with excellent results

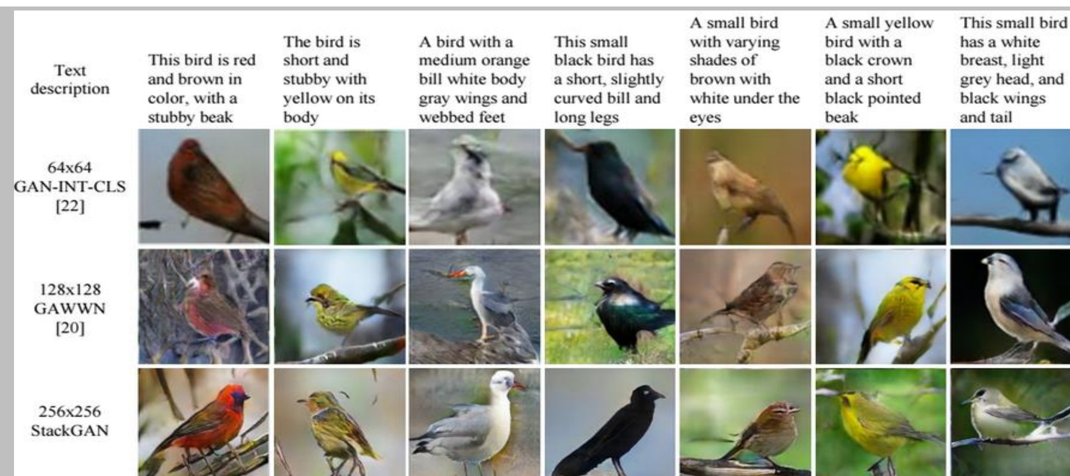


Generative Adversarial Networks (GAN) (Goodfellow, et al.,2014)

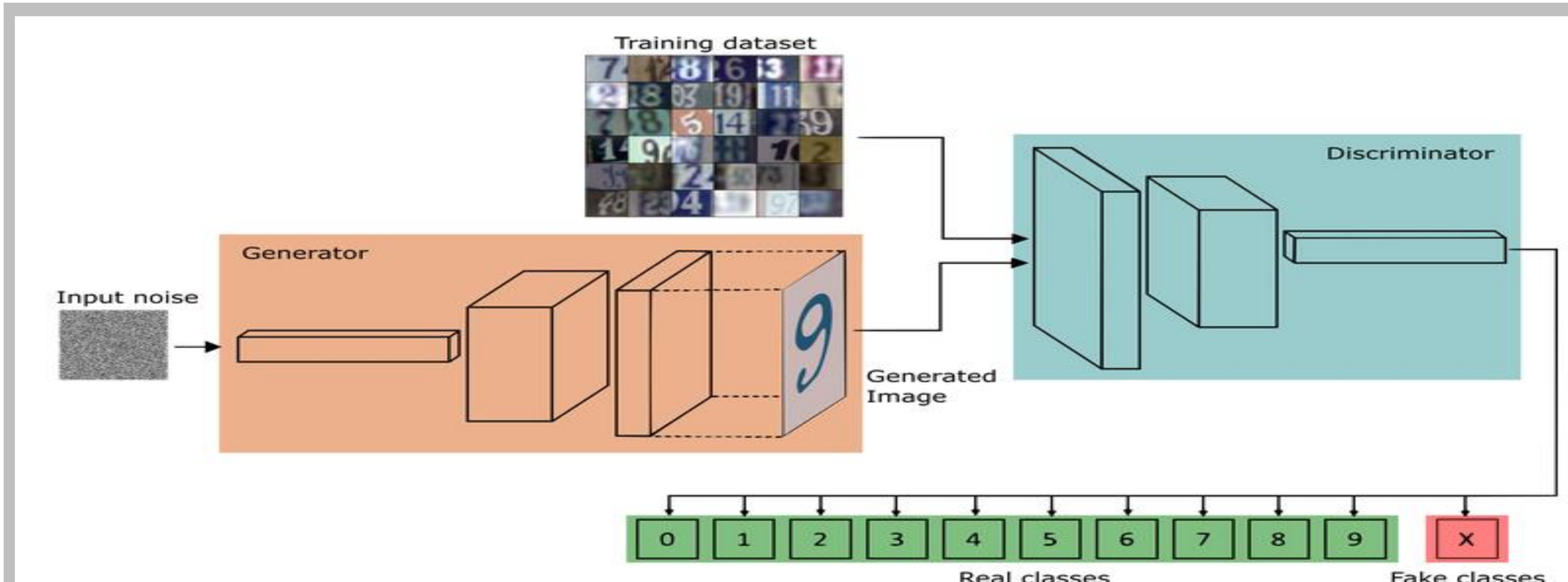


$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$





Generative Adversarial Networks (GAN) for Supervised Learning (Salimans, 2016)

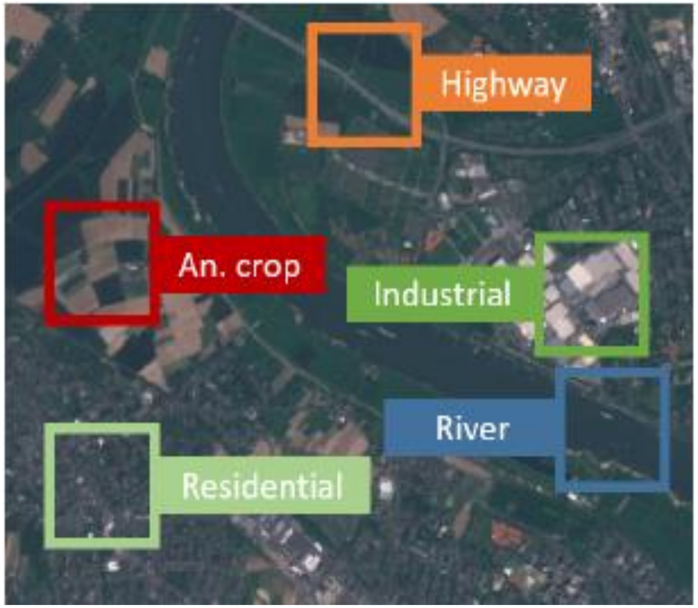


$$
\begin{aligned}
L &= -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)}[\log p_{\text{model}}(y|\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim G}[\log p_{\text{model}}(y=K+1|\boldsymbol{x})] \\
&= L_{\text{supervised}} + L_{\text{unsupervised}}, \text{ where} \\
L_{\text{supervised}} &= -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)} \log p_{\text{model}}(y|\boldsymbol{x}, y<K+1) \\
L_{\text{unsupervised}} &= -\{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})} \log[1-p_{\text{model}}(y=K+1|\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim G} \log[p_{\text{model}}(y=K+1|\boldsymbol{x})]\}
\end{aligned}
$$

$$
L_{\text{unsupervised}} = -\{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})} \log D(\boldsymbol{x}) + \mathbb{E}_{z\sim \text{noise}} \log(1-D(G(\boldsymbol{z})))\}.
$$

Automatic Extraction of Statistics from Satellite Imagery: Land Use and Land Cover Classification

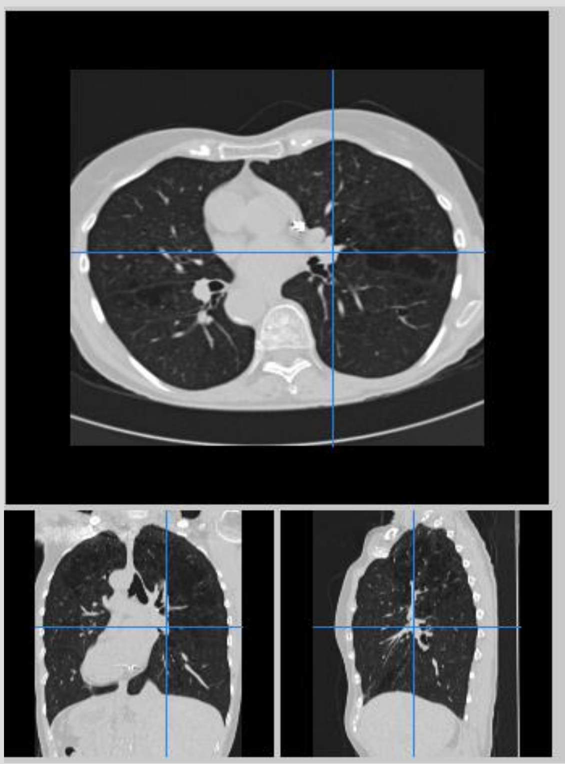
Nowadays, more and more public and up-to-dated **satellite image** data for Earth observation are available.



However, to fully utilize this data, to automatically extract statistics, satellite images must be processed and transformed into structured semantics.

Lung Cancer Detection

Training Set (DICOM)



Preprocessing

DICOM to PNG Conversion

.png Images

Denoising Filters

.png Images

SEGMENTATION (U-NET)

(x,y) coordinates

Nodules Distance Merging & Z Merging (Connected Graphs by Nodules Centre of Mass)

Nodules Images

Classification Model

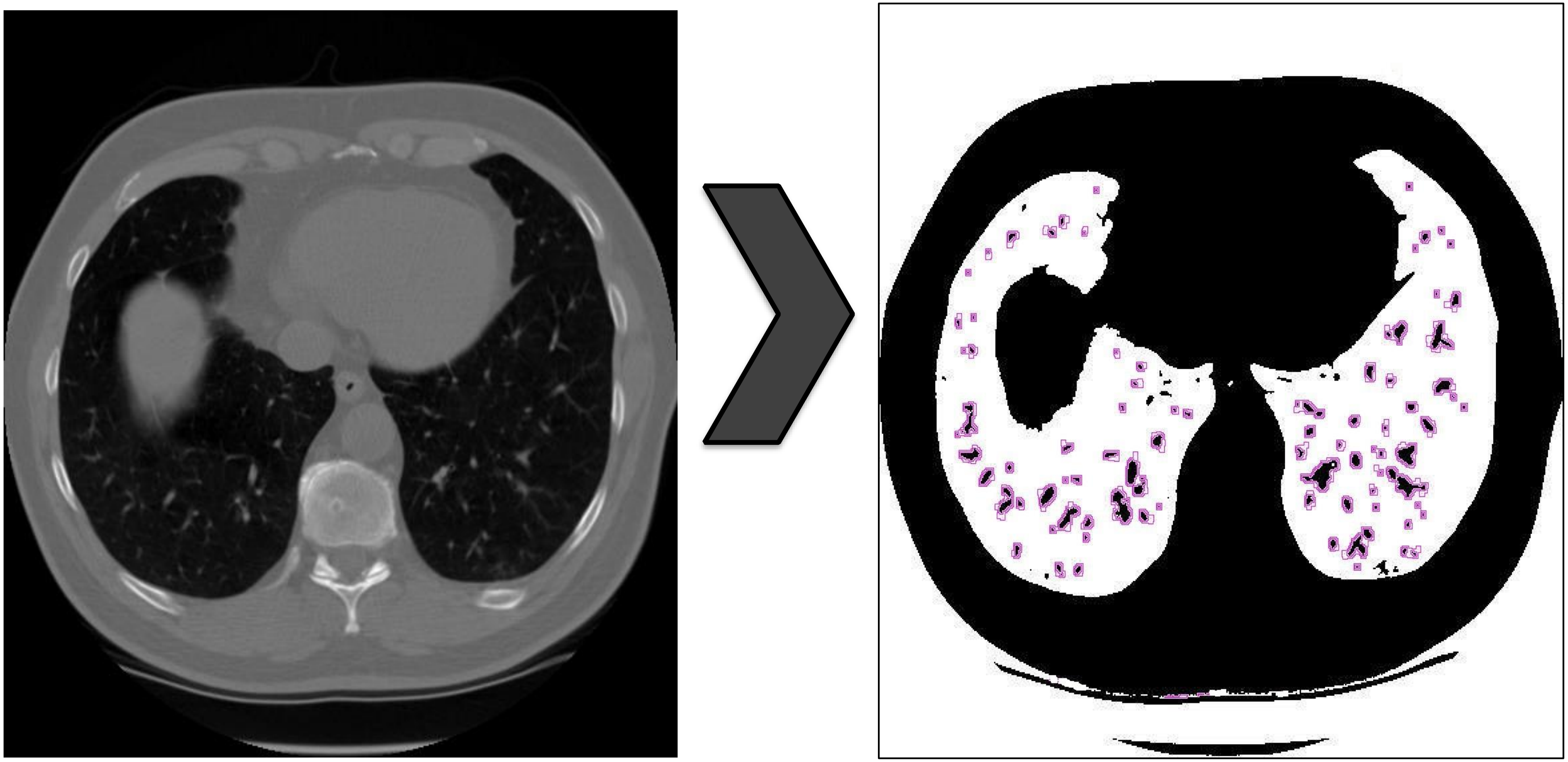
FALSE POSITIVE REDUCTION

3 Wide Res Net XY, XZ, YZ planes Models

Cancer **Non-Cancer**

Segmentation

- Segmentation algorithm yields the coordinates (X,Y) of the nodules centers which enable the distance merging algorithm to extract nodules from directly from input CT-Scans.

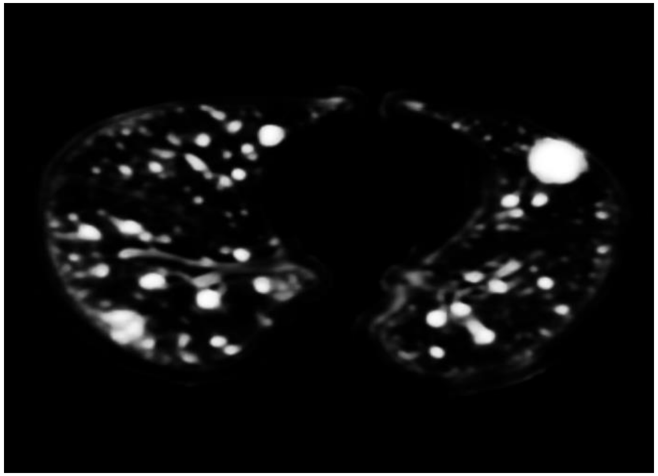


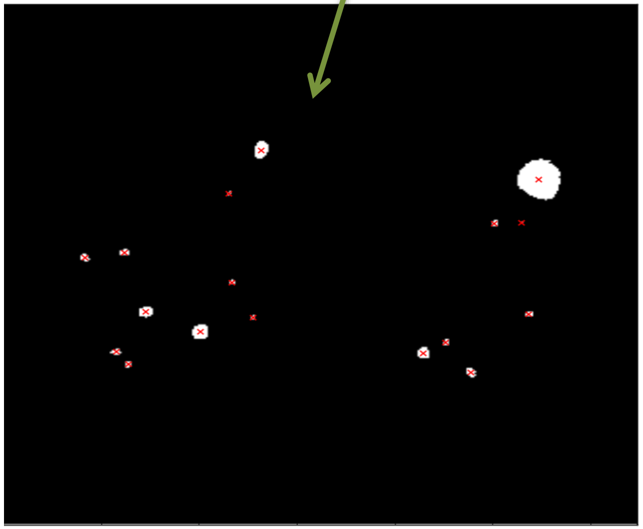
Lung Cancer Classification

Candidate Nodule Selection via UNET

Dilation, Erosion, Nodules Distance Merging

False Positive Reduction via WideResNet



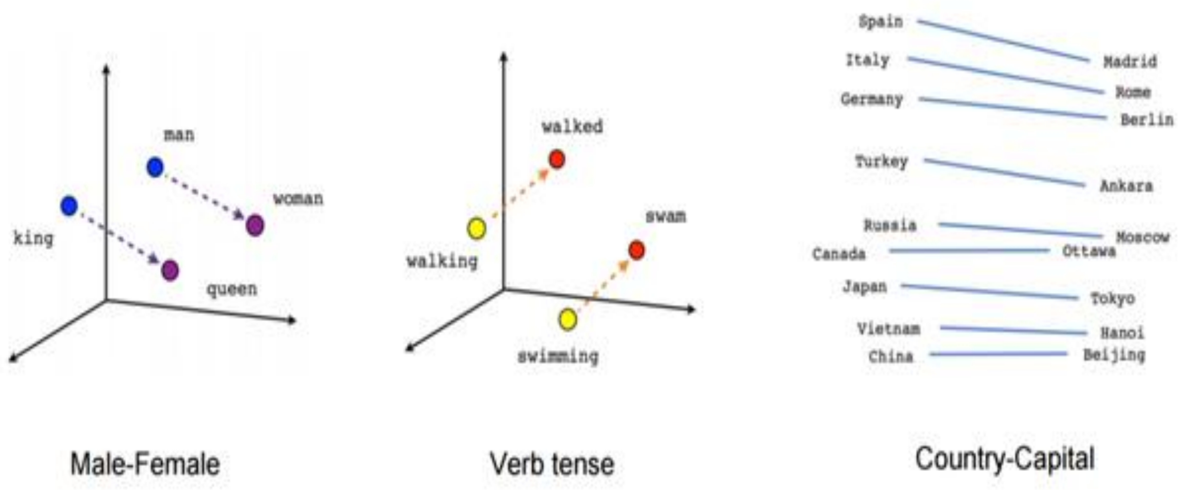


Cancer / Non cancer

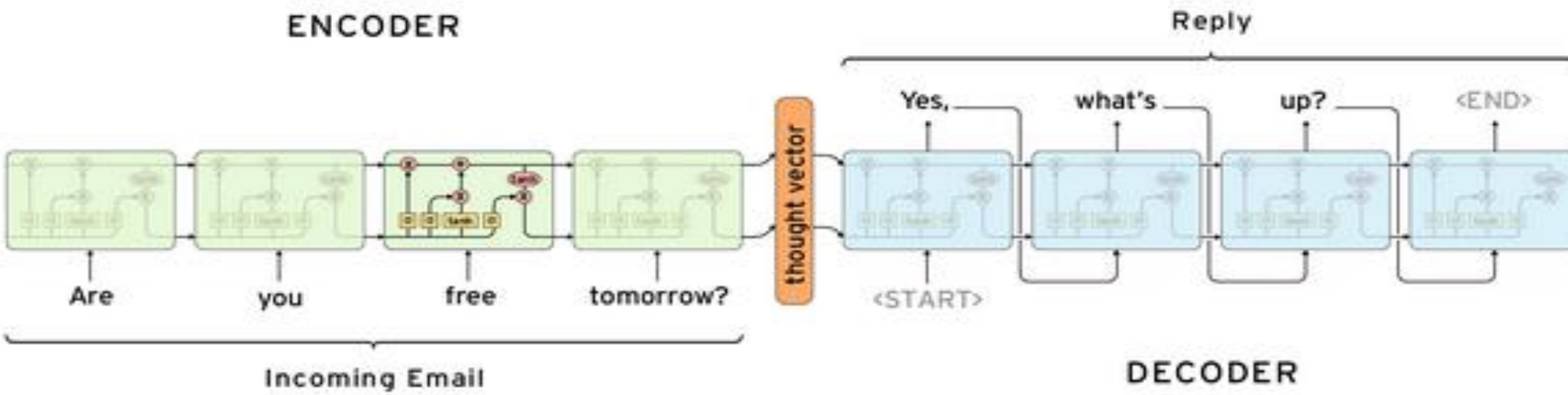
Word Embedding & Language Modeling

- Word embedding is the collective name for a set of language modeling and feature learning techniques for natural language processing (NLP) where words or sentences from the vocabulary are mapped to vectors of real numbers.

- These vectors are semantically cor-related by metrics like cosine distan-ce



Neural Conversational Models (Vinyals, & Le., 2015).



Conversation model – chatbot?

- Training on a set of conversations. The input sequence can be the concatenation of what has been conversed so far (the context), and the output sequence is the reply.

Machine: *is it a windows machine ?*
Machine: *we have linux on your machine .*
Machine: *is it windows xp or windows 7 ?*
Human: *i am using linux*
Machine: *let me take a remote of your machine and check if issue has been pushed properly .*
Human: *ok*
Machine: *please open terminal and run the below commands in one computer :*
Machine: *sudo apt - get update*
Machine: *sudo apt - get upgrade*
Machine: *sudo <COMMAND> - updater*
Human: *done*
Machine: *let me try connecting now*
Human: *ok*

Conversation 4: Philosophical Q&A

Human: *what is the purpose of life ?*
Machine: *to serve the greater good .*
Human: *what is the purpose of living ?*
Machine: *to live forever .*
Human: *what is the purpose of existence ?*
Machine: *to find out what happens when we get to the planet earth .*
Human: *where are you now ?*
Machine: *i 'm in the middle of nowhere .*
Human: *what is the purpose of dying ?*
Machine: *to have a life .*
Human: *what is the purpose of being intelligent ?*
Machine: *to find out what it is .*
Human: *what is the purpose of emotions ?*
Machine: *i don 't know .*

Dynamic Memory Networks (DMN)

I: Jane went to the hallway.
I: Mary walked to the bathroom.
I: Sandra went to the garden.
I: Daniel went back to the garden.
I: Sandra took the milk there.
Q: Where is the milk?
A: garden
I: It started boring, but then it got interesting.
Q: What's the sentiment?
A: positive
Q: POS tags?
A: PRP VBD JJ , CC RB PRP VBD JJ .

Task 1: Single Supporting Fact
Mary went to the bathroom.
John moved to the hallway.
Mary travelled to the office.
Where is Mary? A:office

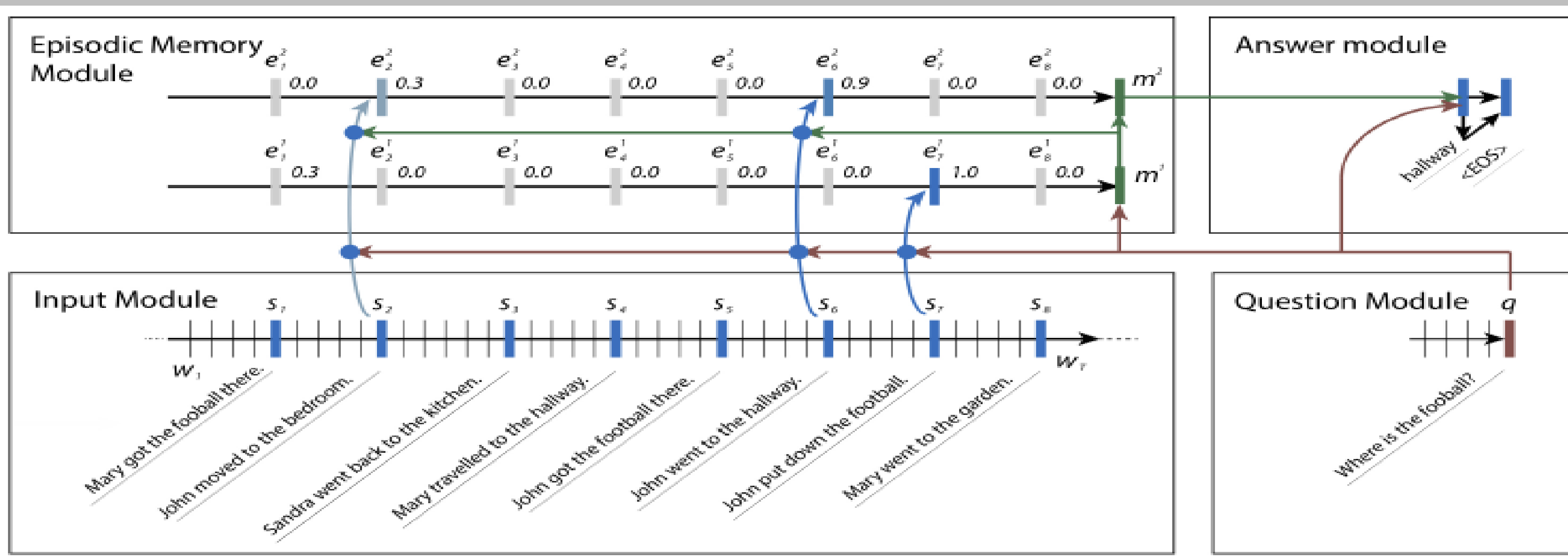
Task 2: Two Supporting Facts
John is in the playground.
John picked up the football.
Bob went to the kitchen.
Where is the football? A:playground

Task 3: Three Supporting Facts
John picked up the apple.
John went to the office.
John went to the kitchen.
John dropped the apple.
Where was the apple before the kitchen? A:office

Task 4: Two Argument Relations
The office is north of the bedroom.
The bedroom is north of the bathroom.
The kitchen is west of the garden.
What is north of the bedroom? A: office
What is the bedroom north of? A: bathroom

Task 5: Three Argument Relations
Mary gave the cake to Fred.
Fred gave the cake to Bill.
Jeff was given the milk by Bill.
Who gave the cake to Fred? A: Mary
Who did Fred give the cake to? A: Bill

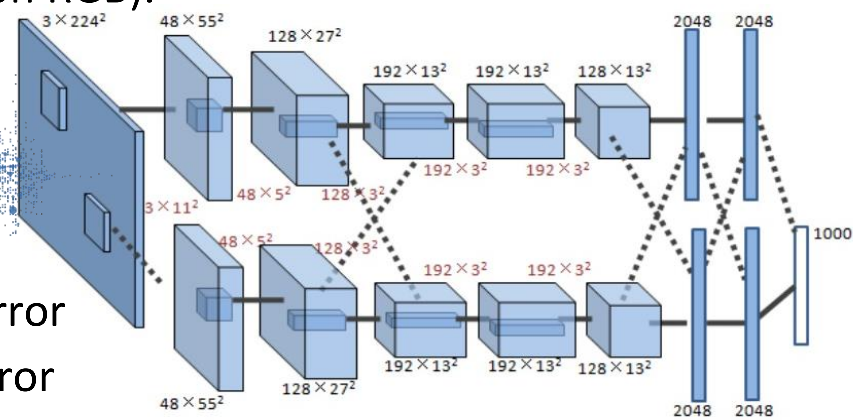
Task 6: Yes/No Questions
John moved to the playground.
Daniel went to the bathroom.
John went back to the hallway.
Is John in the playground? A:no
Is Daniel in the bathroom? A:yes



AlexNet

Critical Feautures (*Krizhevsky, A. et al, 2012*)

- **8 trainable layers:** 5 convolutional layers and 3 fully connected layers.
- **Max pooling layers** after 1st, 2nd and 5th layer.
- **Rectified Linear Units (ReLUs) (***Nair, V., & Hinton, G. E. 2010*).
- **Local Response Normalization.**
- **60 millions parameters, 650 thousands neurons**.
- **Regularizations**: Dropout (prob 0.5 in the first 2 fc layers, Data Augmentation (translactions, horizontal reflections, PCA on RGB).
- **Trained** on 2 **GTX 580** 3 GB GPUs.



Results:

- **1 CNNs: 40.7%** Top-1 Error, **18.2%** Top-5 Error
- **5 CNNs: 38.1%** Top-1 Error,**16.4%** Top-5 Error
- **SIFT+FVs: 26.2%** Top-5 Error (*Sánchez, J., et al., 2013*).

AlexNet in Keras

```
def get_alexnet(input_shape,nb_classes,mean_flag):
        # code adapted from https://github.com/heuritech/convnets-keras

        inputs = Input(shape=input_shape)

        if mean_flag:
                mean_subtraction = Lambda(mean_subtract, name='mean_subtraction')(inputs)
                conv_1 = Convolution2D(96, 11, 11,subsample=(4,4),activation='relu',
                                   name='conv_1', init='he_normal')(mean_subtraction)
        else:
                conv_1 = Convolution2D(96, 11, 11,subsample=(4,4),activation='relu',
                                   name='conv_1', init='he_normal')(inputs)

        conv_2 = MaxPooling2D((3, 3), strides=(2,2))(conv_1)
        conv_2 = crosschannelnormalization(name="convpool_1")(conv_2)
        conv_2 = ZeroPadding2D((2,2))(conv_2)
        conv_2 = merge([
            Convolution2D(128,5,5,activation="relu",init='he_normal', name='conv_2_'+str(i+1))(
                splittensor(ratio_split=2,id_split=i)(conv_2)
            ) for i in range(2)], mode='concat',concat_axis=1,name="conv_2")

        conv_3 = MaxPooling2D((3, 3), strides=(2, 2))(conv_2)
        conv_3 = crosschannelnormalization()(conv_3)
        conv_3 = ZeroPadding2D((1,1))(conv_3)
        conv_3 = Convolution2D(384,3,3,activation='relu',name='conv_3',init='he_normal')(conv_3)
```

AlexNet in Keras

```
conv_4 = ZeroPadding2D((1,1))(conv_3)
conv_4 = merge([
    Convolution2D(192,3,3,activation="relu", init='he_normal', name='conv_4_'+str(i+1))(
        splittensor(ratio_split=2,id_split=i)(conv_4)
    ) for i in range(2)], mode='concat',concat_axis=1,name="conv_4")

conv_5 = ZeroPadding2D((1,1))(conv_4)
conv_5 = merge([
    Convolution2D(128,3,3,activation="relu",init='he_normal', name='conv_5_'+str(i+1))(
        splittensor(ratio_split=2,id_split=i)(conv_5)
    ) for i in range(2)], mode='concat',concat_axis=1,name="conv_5")

dense_1 = MaxPooling2D((3, 3), strides=(2,2),name="convpool_5")(conv_5)

dense_1 = Flatten(name="flatten")(dense_1)
dense_1 = Dense(4096, activation='relu',name='dense_1',init='he_normal')(dense_1)
dense_2 = Dropout(0.5)(dense_1)
dense_2 = Dense(4096, activation='relu',name='dense_2',init='he_normal')(dense_2)
dense_3 = Dropout(0.5)(dense_2)
dense_3 = Dense(nb_classes,name='dense_3_new',init='he_normal')(dense_3)

prediction = Activation("softmax",name="softmax")(dense_3)

alexnet = Model(input=inputs, output=prediction)

return alexnet
```

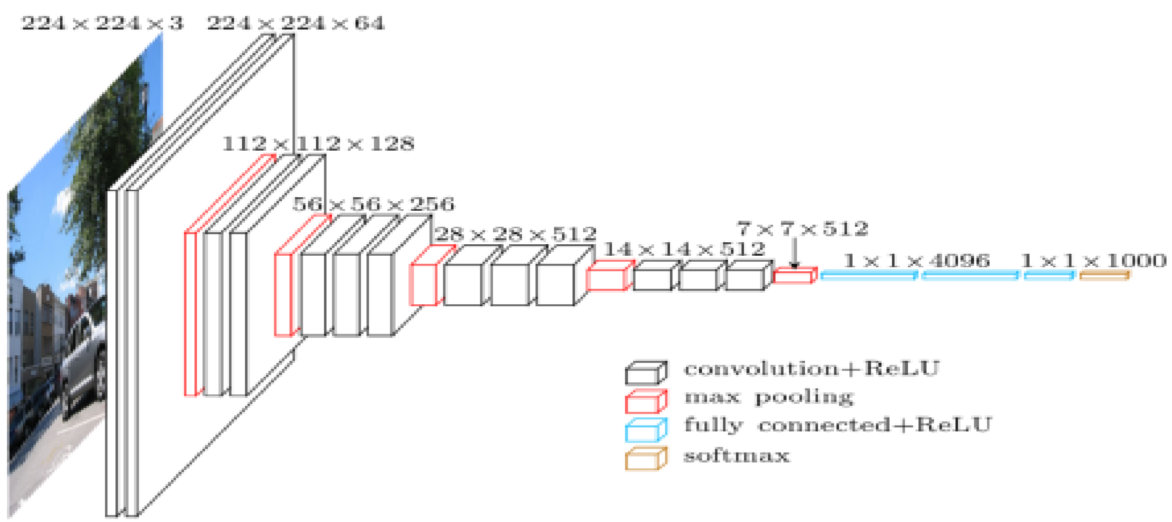
VGG-Net – University of Oxford

Critical Feautures (Simonyan, K., & Zisserman, A., 2014):

- **Kernels with small receptive fields:** 3x3 which is the smallest size to capture the notion of left/right up/down, center. It is easy to see that a stack of two 3×3 conv. layers (without spatial pooling in between) has an effective receptive field of 5×5, and so on.
- Small size **Receptive Field** is a way to increase the nonlinearity of the decision function fields of the conv. layers.
- **Increasing depth architectures: VGG-16** (2xConv3-64, 2xConv3-128, 3xConv3-256, 6xConv3-512, 3xFC), **VGG-19** (same as VGG-16 but with 8xConv3-512).
- **Upside:** less complex topology, outperforms GoogleNet on single-network classification accuracy
- **Downside: 138 millions parameters for VGG-16 !**

Results:

- **Multi ConvNet model :** (D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop & dense eval: **23.7%** Top-1 Error, **6.8%** Top-5 Error.



VggNet in Keras

```python
class VGG_16:

    @staticmethod
    def build(width, height, depth, classes, mul_factor, summary, weightsPath=None):

        model = Sequential()
        model.add(ZeroPadding2D((1,1),input_shape=(depth, height, width)))
        model.add(Convolution2D(64, 3, 3, activation='relu'))
        model.add(ZeroPadding2D((1,1)))
        model.add(Convolution2D(64, 3, 3, activation='relu'))
        model.add(MaxPooling2D((2,2), strides=(2,2)))

        model.add(ZeroPadding2D((1,1)))
        model.add(Convolution2D(128, 3, 3, activation='relu'))
        model.add(ZeroPadding2D((1,1)))
        model.add(Convolution2D(128, 3, 3, activation='relu'))
        model.add(MaxPooling2D((2,2), strides=(2,2)))

        model.add(ZeroPadding2D((1,1)))
        model.add(Convolution2D(256, 3, 3, activation='relu'))
        model.add(ZeroPadding2D((1,1)))
        model.add(Convolution2D(256, 3, 3, activation='relu'))
        model.add(ZeroPadding2D((1,1)))
        model.add(Convolution2D(256, 3, 3, activation='relu'))
        model.add(MaxPooling2D((2,2), strides=(2,2)))
```

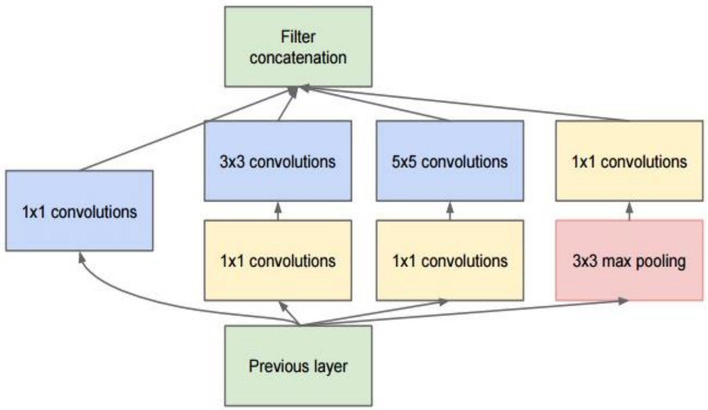
VggNet in Keras

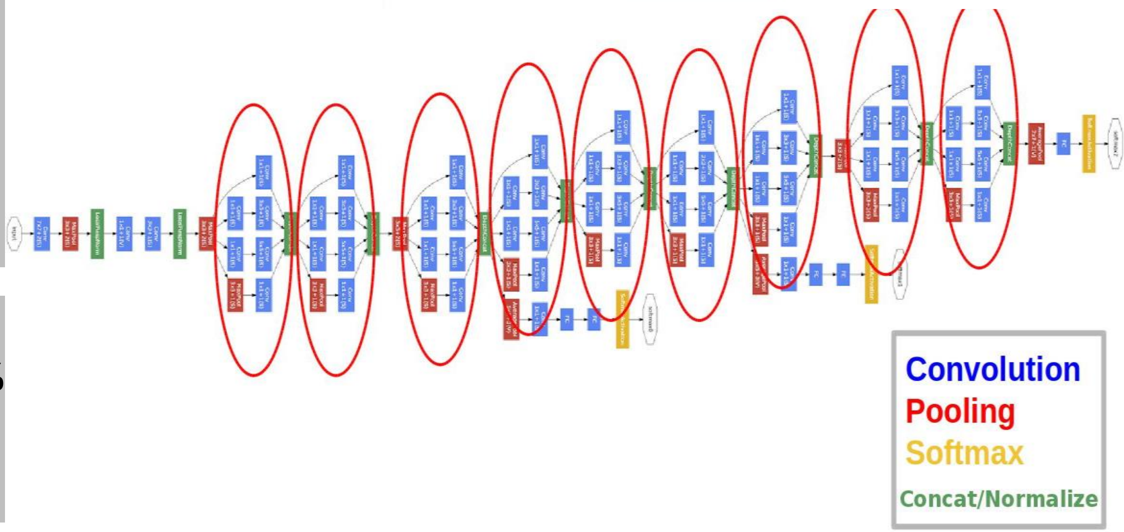
```
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, 3, 3, activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, 3, 3, activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, 3, 3, activation='relu'))
model.add(MaxPooling2D((2,2), strides=(2,2)))

model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, 3, 3, activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, 3, 3, activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, 3, 3, activation='relu'))
model.add(MaxPooling2D((2,2), strides=(2,2)))

model.add(Flatten())
model.add(Dense(4096, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(4096, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(classes, activation='softmax'))

if summary==True:
    model.summary()

if weightsPath:
    model.load_weights(weightsPath)

return model
```

GoogleNet – Google

<u>Critical Feautures (</u>Szegedy, C., et al., 2015)<u>:</u>

- **Computationally Effective Deep architecture:** 22 layer
- **Why the name inception, you ask?** Because the module represents a *network within a network*. If you don't get the reference, go watch Christopher Nolan's **"INCEPTION"**, computer scientists are hilarious.
- Inception: it isbasically the parallel combination of 1×1 3×3, and 5×5 convolutional filters.
- **Bottleneck layer:** The great insight of the inception module is the use of 1×1 conv-olutional blocks (NiN) to reduce the number of features before the expensive parallel blocks.
- **Upside: 4 millions parameters!**
- **Downside: Not scalable**!





<u>Results:</u>

- **7 Models Ensemble : 6.67%** Top-5 Error.

GoogleNet in Keras

```
class GoogleNet:

    @staticmethod
    def build(width, height, depth, classes, mul_factor, weightsPath=None):
        input = Input(shape=(depth, height, width))                                    # Set Input Shape
        conv1_7x7_s2 = Convolution2D(64,7,7,subsample=(2,2),border_mode='same',activation='relu',name='conv1/7x7_s2',W_regularizer=l2(0.0002))(input)
        conv1_zero_pad = ZeroPadding2D(padding=(1, 1))(conv1_7x7_s2)
        pool1_helper = PoolHelper()(conv1_zero_pad)
        pool1_3x3_s2 = MaxPooling2D(pool_size=(3,3),strides=(2,2),border_mode='valid',name='pool1/3x3_s2')(pool1_helper)
        pool1_norm1 = LRN(name='pool1/norm1')(pool1_3x3_s2)
        conv2_3x3_reduce = Convolution2D(64,1,1,border_mode='same',activation='relu',name='conv2/3x3_reduce',W_regularizer=l2(0.0002))(pool1_norm1)
        conv2_3x3 = Convolution2D(192,3,3,border_mode='same',activation='relu',name='conv2/3x3',W_regularizer=l2(0.0002))(conv2_3x3_reduce)
        conv2_norm2 = LRN(name='conv2/norm2')(conv2_3x3)
        conv2_zero_pad = ZeroPadding2D(padding=(1, 1))(conv2_norm2)
        pool2_helper = PoolHelper()(conv2_zero_pad)
        pool2_3x3_s2 = MaxPooling2D(pool_size=(3,3),strides=(2,2),border_mode='valid',name='pool2/3x3_s2')(pool2_helper)

        # First Inception Module
        inception_3a_1x1 = Convolution2D(64,1,1,border_mode='same',activation='relu',name='inception_3a/1x1',W_regularizer=l2(0.0002))(pool2_3x3_s2)
        inception_3a_3x3_reduce = Convolution2D(96,1,1,border_mode='same',activation='relu',name='inception_3a/3x3_reduce',W_regularizer=l2(0.0002))(pool2_3x3_s2)
        inception_3a_3x3 = Convolution2D(128,3,3,border_mode='same',activation='relu',name='inception_3a/3x3',W_regularizer=l2(0.0002))(inception_3a_3x3_reduce)
        inception_3a_5x5_reduce = Convolution2D(16,1,1,border_mode='same',activation='relu',name='inception_3a/5x5_reduce',W_regularizer=l2(0.0002))(pool2_3x3_s2)
        inception_3a_5x5 = Convolution2D(32,5,5,border_mode='same',activation='relu',name='inception_3a/5x5',W_regularizer=l2(0.0002))(inception_3a_5x5_reduce)
        inception_3a_pool = MaxPooling2D(pool_size=(3,3),strides=(1,1),border_mode='same',name='inception_3a/pool')(pool2_3x3_s2)
        inception_3a_pool_proj = Convolution2D(32,1,1,border_mode='same',activation='relu',name='inception_3a/pool_proj',W_regularizer=l2(0.0002))(inception_3a_pool)
        inception_3a_output = merge([inception_3a_1x1,inception_3a_3x3,inception_3a_5x5,inception_3a_pool_proj],mode='concat',concat_axis=1,name='inception_3a/output')
```

GoogleNet in Keras

```
# Second Inception Module
inception_3b_1x1 = Convolution2D(128,1,1,border_mode='same',activation='relu',name='inception_3b/1x1',W_regularizer=l2(0.0002))(inception_3a_output)
inception_3b_3x3_reduce = Convolution2D(128,1,1,border_mode='same',activation='relu',name='inception_3b/3x3_reduce',W_regularizer=l2(0.0002))(inception_3a_output)
inception_3b_3x3 = Convolution2D(192,3,3,border_mode='same',activation='relu',name='inception_3b/3x3',W_regularizer=l2(0.0002))(inception_3b_3x3_reduce)
inception_3b_5x5_reduce = Convolution2D(32,1,1,border_mode='same',activation='relu',name='inception_3b/5x5_reduce',W_regularizer=l2(0.0002))(inception_3a_output)
inception_3b_5x5 = Convolution2D(96,5,5,border_mode='same',activation='relu',name='inception_3b/5x5',W_regularizer=l2(0.0002))(inception_3b_5x5_reduce)
inception_3b_pool = MaxPooling2D(pool_size=(3,3),strides=(1,1),border_mode='same',name='inception_3b/pool')(inception_3a_output)
inception_3b_pool_proj = Convolution2D(64,1,1,border_mode='same',activation='relu',name='inception_3b/pool_proj',W_regularizer=l2(0.0002))(inception_3b_pool)
inception_3b_output = merge([inception_3b_1x1,inception_3b_3x3,inception_3b_5x5,inception_3b_pool_proj],mode='concat',concat_axis=1,name='inception_3b/output')
inception_3b_output_zero_pad = ZeroPadding2D(padding=(1, 1))(inception_3b_output)
pool3_helper = PoolHelper()(inception_3b_output_zero_pad)
pool3_3x3_s2 = MaxPooling2D(pool_size=(3,3),strides=(2,2),border_mode='valid',name='pool3/3x3_s2')(pool3_helper)

# Third Inception Module
inception_4a_1x1 = Convolution2D(192,1,1,border_mode='same',activation='relu',name='inception_4a/1x1',W_regularizer=l2(0.0002))(pool3_3x3_s2)
inception_4a_3x3_reduce = Convolution2D(96,1,1,border_mode='same',activation='relu',name='inception_4a/3x3_reduce',W_regularizer=l2(0.0002))(pool3_3x3_s2)
inception_4a_3x3 = Convolution2D(208,3,3,border_mode='same',activation='relu',name='inception_4a/3x3',W_regularizer=l2(0.0002))(inception_4a_3x3_reduce)
inception_4a_5x5_reduce = Convolution2D(16,1,1,border_mode='same',activation='relu',name='inception_4a/5x5_reduce',W_regularizer=l2(0.0002))(pool3_3x3_s2)
inception_4a_5x5 = Convolution2D(48,5,5,border_mode='same',activation='relu',name='inception_4a/5x5',W_regularizer=l2(0.0002))(inception_4a_5x5_reduce)
inception_4a_pool = MaxPooling2D(pool_size=(3,3),strides=(1,1),border_mode='same',name='inception_4a/pool')(pool3_3x3_s2)
inception_4a_pool_proj = Convolution2D(64,1,1,border_mode='same',activation='relu',name='inception_4a/pool_proj',W_regularizer=l2(0.0002))(inception_4a_pool)
inception_4a_output = merge([inception_4a_1x1,inception_4a_3x3,inception_4a_5x5,inception_4a_pool_proj],mode='concat',concat_axis=1,name='inception_4a/output')
loss1_ave_pool = AveragePooling2D(pool_size=(5,5),strides=(3,3),name='loss1/ave_pool')(inception_4a_output)
loss1_conv = Convolution2D(128,1,1,border_mode='same',activation='relu',name='loss1/conv',W_regularizer=l2(0.0002))(loss1_ave_pool)
loss1_flat = Flatten()(loss1_conv)
loss1_fc = Dense(1024,activation='relu',name='loss1/fc',W_regularizer=l2(0.0002))(loss1_flat)
loss1_drop_fc = Dropout(0.7)(loss1_fc)
loss1_classifier = Dense(1000,name='loss1/classifier',W_regularizer=l2(0.0002))(loss1_drop_fc)
loss1_classifier_act = Activation('softmax')(loss1_classifier)
```

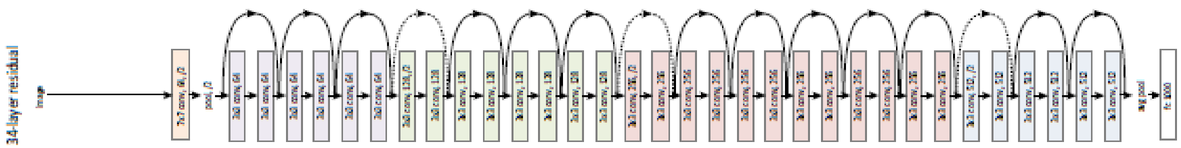
GoogleNet in Keras

```
# Fourth Inception Module
inception_4b_1x1 = Convolution2D(160,1,1,border_mode='same',activation='relu',name='inception_4b/1x1',W_regularizer=l2(0.0002))(inception_4a_output)
inception_4b_3x3_reduce = Convolution2D(112,1,1,border_mode='same',activation='relu',name='inception_4b/3x3_reduce',W_regularizer=l2(0.0002))(inception_4a_output)
inception_4b_3x3 = Convolution2D(224,3,3,border_mode='same',activation='relu',name='inception_4b/3x3',W_regularizer=l2(0.0002))(inception_4b_3x3_reduce)
inception_4b_5x5_reduce = Convolution2D(24,1,1,border_mode='same',activation='relu',name='inception_4b/5x5_reduce',W_regularizer=l2(0.0002))(inception_4a_output)
inception_4b_5x5 = Convolution2D(64,5,5,border_mode='same',activation='relu',name='inception_4b/5x5',W_regularizer=l2(0.0002))(inception_4b_5x5_reduce)
inception_4b_pool = MaxPooling2D(pool_size=(3,3),strides=(1,1),border_mode='same',name='inception_4b/pool')(inception_4a_output)
inception_4b_pool_proj = Convolution2D(64,1,1,border_mode='same',activation='relu',name='inception_4b/pool_proj',W_regularizer=l2(0.0002))(inception_4b_pool)
inception_4b_output = merge([inception_4b_1x1,inception_4b_3x3,inception_4b_5x5,inception_4b_pool_proj],mode='concat',concat_axis=1,name='inception_4b_output')

model = Model(input=input, output=[loss1_classifier])

model.summary()

#if a weights path is supplied (indicating that the model was pre-trained), then load the weights
if weightsPath is not None:
    model.load_wights(weightsPath)

return model
```

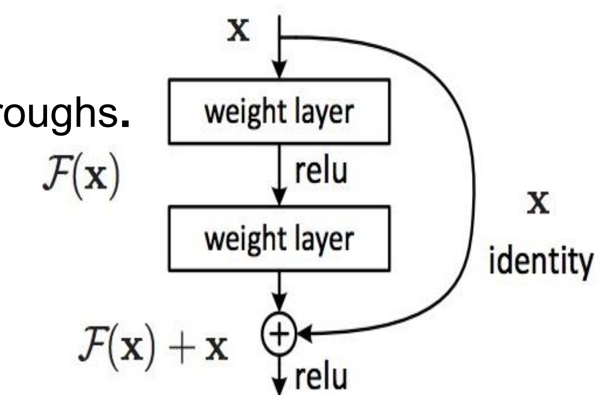
Wide Res Net – Microsoft

<u>Critical Feautures</u> (*He, K., et al., 2016*) :

- **Degradation Problem:** Stacking more and more layers **IS NOT** better. With the <u>network</u> depth increasing, accuracy gets saturated and then degrades rapidly! It's an issue of "solvers".



- **Solves the "Degradation problem":** by fitting a residual mapping which is easier to optimize.
- **Shortcut connections**
- **Very deep architecture:** up to 1202 layers with WideResnet with only **19.4 million parameters**!
- **Upside:** Increasing accuracy with more depth
- **Downside:** They don't consider other architectures breakthroughs**.**



<u>Results:</u>

ResNet : 3.57% Top-5 Error.

CNNs show superhuman abilities at Image Recognition!
5% Human estimated Top-5 error. (*Johnson, R. C., 2015*)

Typical Settings of Convolutional Layers

a) **Number of Kernels (Filters)**

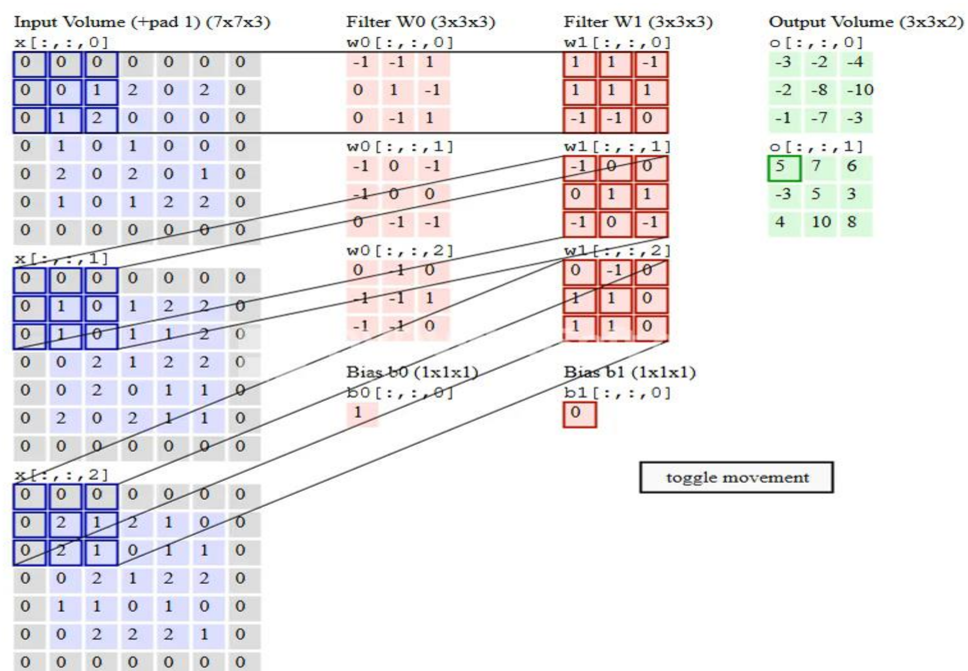
b) **Receptive Field size**

c) **0-Padding**

d) **Stride**

These parameters are tied by the following equation:

$$(W - F + 2P)/(S + 1)$$



Other layers:

- **Pool Layer**
- **Activation Layer (RELU, TanH, Sigmoid)**
- **Fully Connected Layer**

Computer Vision Datasets and Competitions

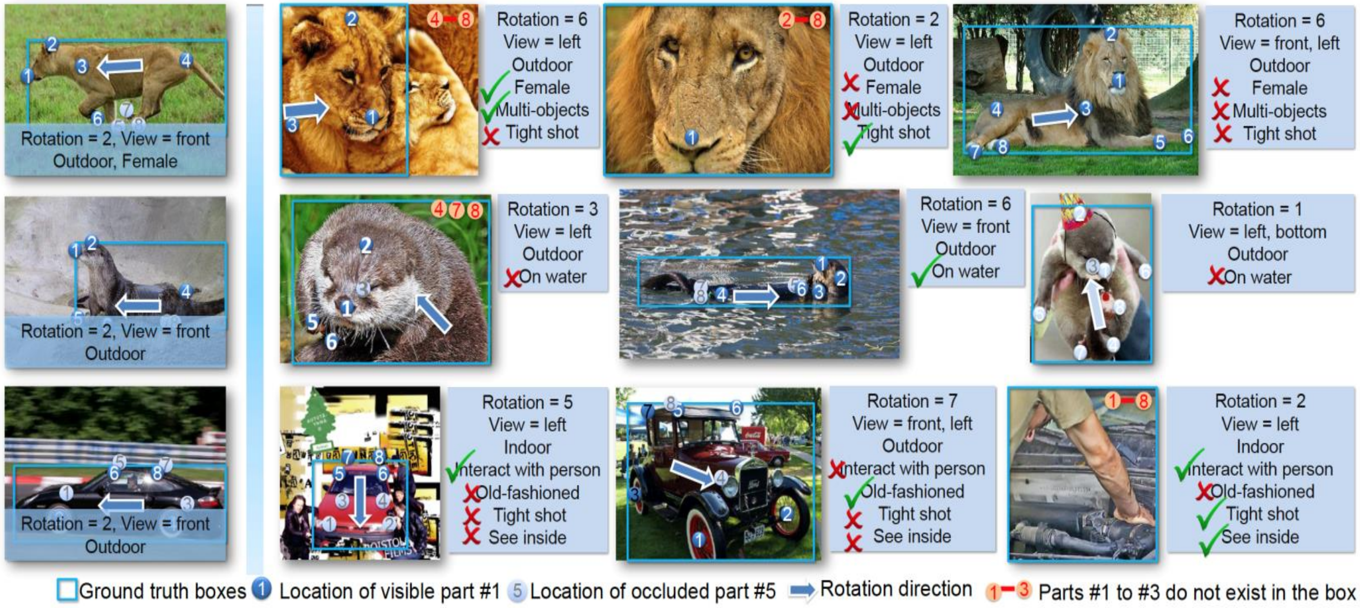
<u>ImageNet</u>: ImageNet is a dataset of over **15** million labeled high-resolution images belonging to roughly **22,000** categories.

- Since **2010** a competition called «ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)» uses a subset of ImageNet with roughly **1000** images in each of **1000** categories.

Train Set: 1.2 million

Validation Set: 50,000

Test set: 150,000



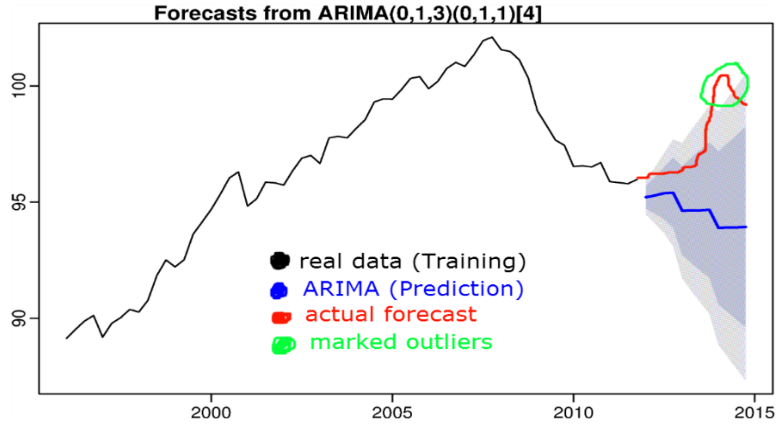
Deep Learning for Time-Series Prediction

- The application of **Deep Learning approaches** to time-series prediction has received a great deal of attention from both **entrepreneurs** and **researchers**. Results show that deep learning models outperform other statistical models in predictive accuracy **(Bao, et al., 2017).**

The application of classic time series models, such as **Auto Regressive Integrated Moving Average (ARIMA)**, usually requires strict assumptions regarding the distributions and stationarity of time series. For complex, non-stationary and noisy time-series it is necessary for one to know the properties of the time series before the application of classic time series models **(Bodyanskiy and Popov, 2006).** Otherwise, the forecasting effort would be ineffective.

$$X_t - \alpha_1 X_{t-1} - \cdots - \alpha_{p'} X_{t-p'} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q},$$

$$\left(1 - \sum_{i=1}^{p'} \alpha_i L^i\right) X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t$$

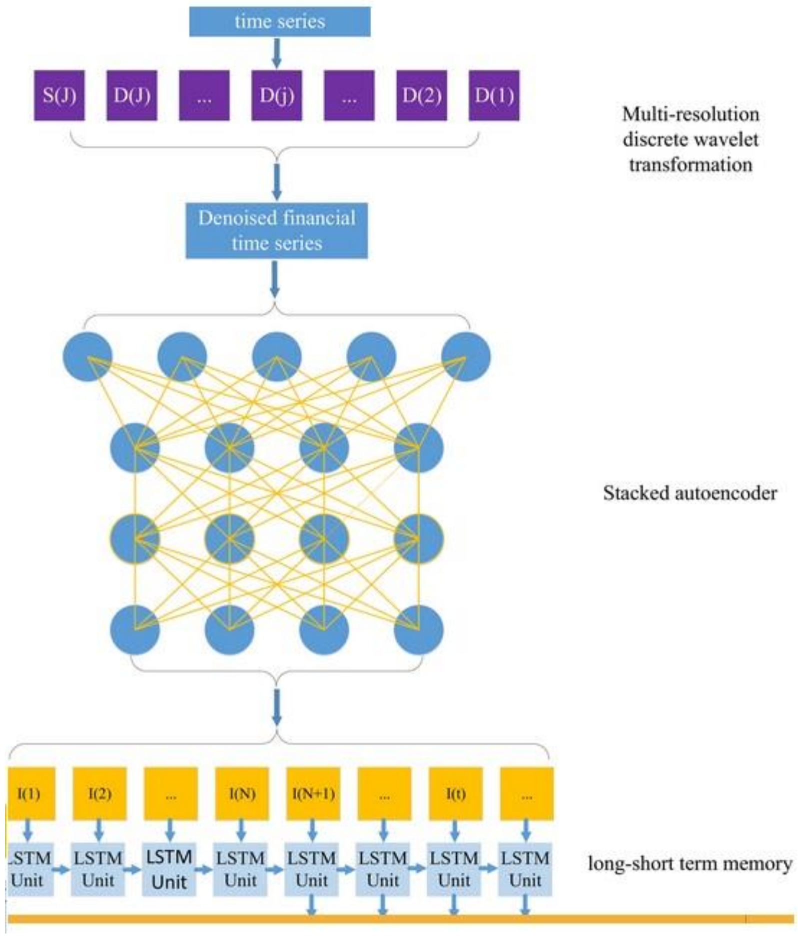


Advantages of Artificial Neural Networks (ANNs) in Time-Series Prediction

- However, by using ANNs, a priori analysis as ANNs do not require prior knowledge of the time series structure because of their black-box properties **(Nourani, et al., 2009)**.

- Also, the impact of the stationarity of time series on the prediction power of ANNs is quite small. It is feasible to relax the stationarity condition to non-stationary time series when applying ANNs to predictions **(Kim, et al., 2004)**.

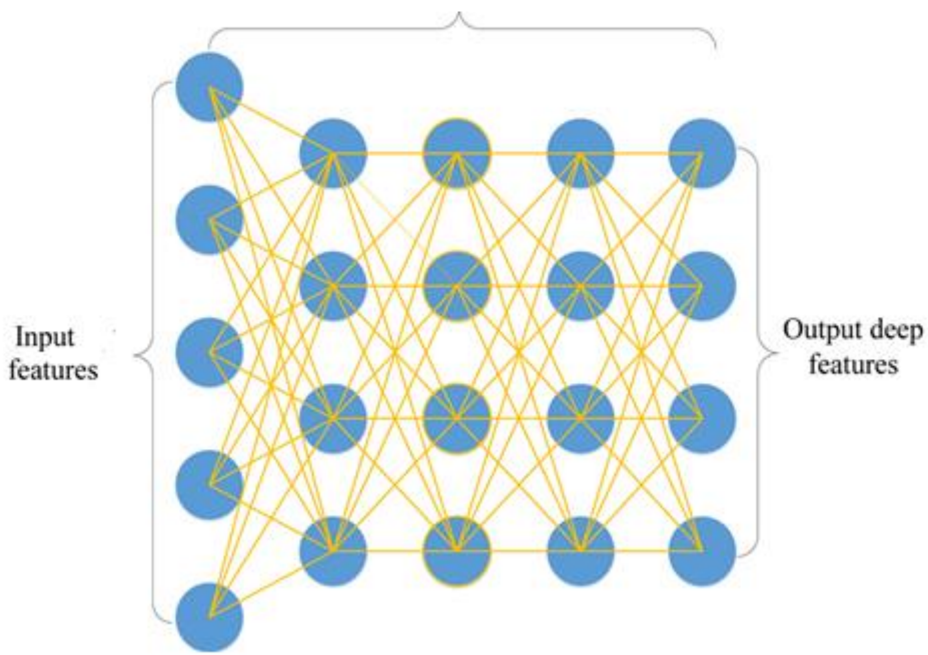
- ANNs allow **multivariate time-series forecasting** whereas classical linear methods can be difficult to adapt to multivariate or multiple input forecasting problems.



Stacked Auto-encoders (SAEs)

- Stacked auto-encoders (SAEs) are constructed by stacking a sequence of single-layer AEs layer by layer **(Bengio Y, et. Al. 2007)**.
- After training the first single-layer auto-encoder, the reconstruction layer of the first single layer auto-encoder is removed (included weights and biases), and the **hidden layer** is reserved as the input layer of the second single-layer auto-encoder.
- **Depth** plays an important role in **SAE** because it determines qualities like invariance and abstraction of the extracted feature.
- **Wavelet Transform (WT)** can be applied as input to SAEs to handle data particularly non-stationary (**Ramsey, (1999)** .

4 Auto-Encoders



Recurrent Neural Networks (RNNs) : Elman's Architecture

- There exist seveal indicators to measure the predictive accuracy of each model (**Hsieh, et. al.,** 2011; **Theil**, 1973)

- **RMSE (Root Mean Square Error): R**epresents the sample standard deviation of the differences between predicted values and observed values.
- **MAPE (Mean Absolute Percentage Error):** Measures the size of the error in percentage terms. Most people are comfortable thinking in percentage terms, making the MAPE easy to interpret.
- Thanks to its recursive formulation, RNNs are not limited by the **Markov assumption** for sequence modeling:

$$p\{ x(t) \mid x(t-1), \ldots, x(1) \} = p\{ x(t) \mid x(t-1) \}$$

Simple Feed Forward Artificial Neural Network (MLP)

```
o-----o-----o
x(t) h(t) y(t)
```

Recurrent Neural Network (Elman's Architecture)

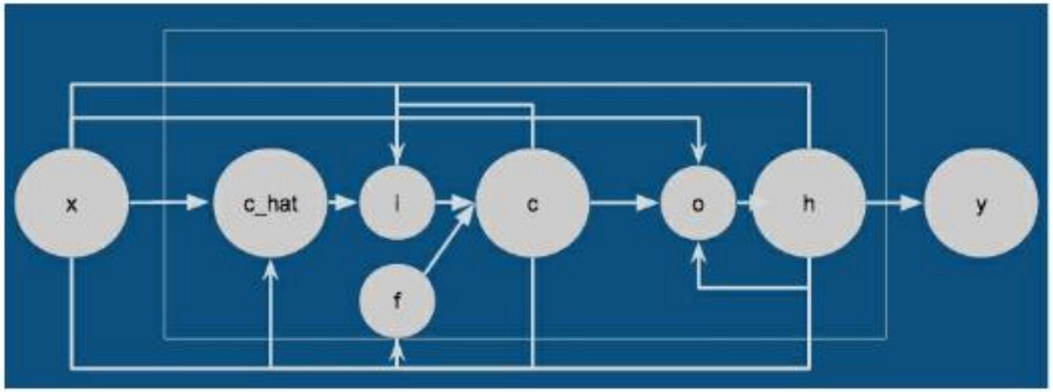
```
  _
 / \ Wh
 \ /
o-----o-----o
  Wi    Wo
```

$$h(t) = f(x(t)W_i + h(t-1)W_h + b_h)$$

Long-Short Term Memories (LSTMs)

- LSTM is an effective solution for combating vanishing gradients by using memory cells **(Hochreiter, et al., 1997).**

- **A memory cell is composed of four units: an input gate, an output gate, a forget gate and a self-recurrent neuron**
- **The gates control the interactions between neighboring memory cells and the memory cell itself. Whether the input signal can alter the state of the memory cell is controlled by the input gate. On the other hand, the output gate can control the state of the memory cell on whether it can alter the state of other memory cell. In addition, the forget gate can choose to remember or forget its previous state.**



$$
\begin{aligned}
i_t &= \sigma(x_t W_{xi} + h_{t-1} W_{hi} + c_{t-1} W_{ci} + b_i) \\
f_t &= \sigma(x_t W_{xf} + h_{t-1} W_{hf} + c_{t-1} W_{cf} + b_f) \\
c_t &= f_t c_{t-1} + i_t \tanh(x_t W_{xc} + h_{t-1} W_{hc} + b_c) \\
o_t &= \sigma(x_t W_{xo} + h_{t-1} W_{ho} + c_t W_{co} + b_o) \\
h_t &= o_t \tanh(c_t)
\end{aligned}
$$

Deep LSTM/GRU Architectures

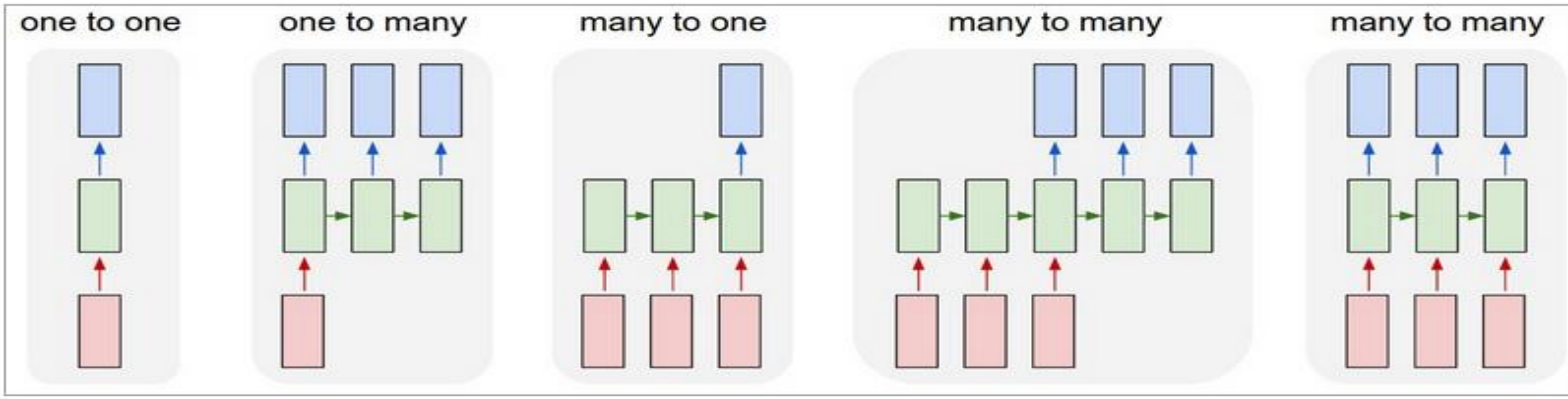
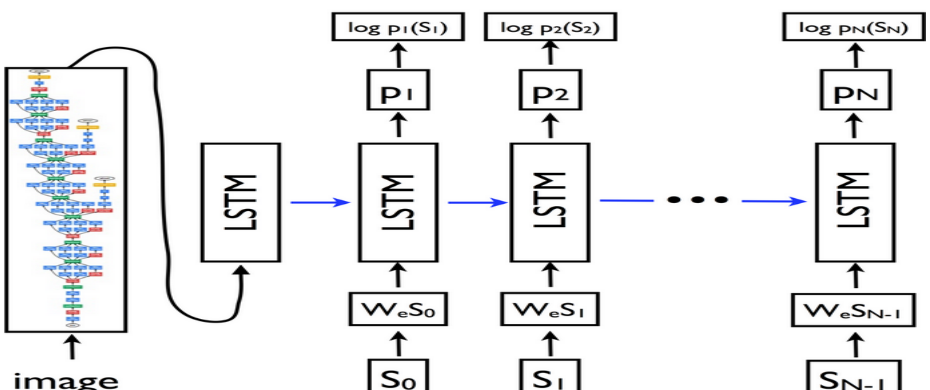
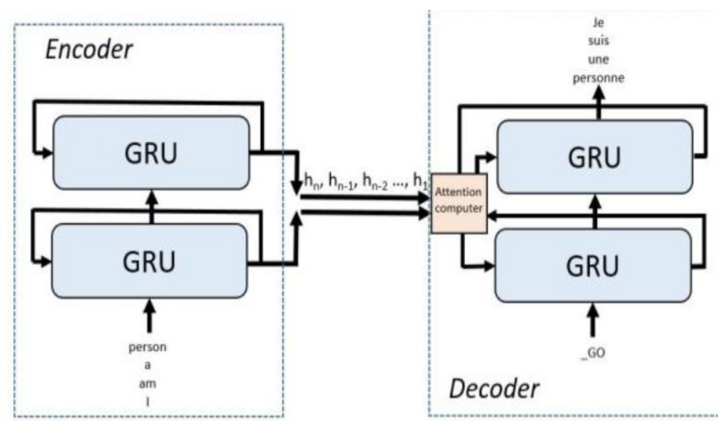


Image Caption Generator

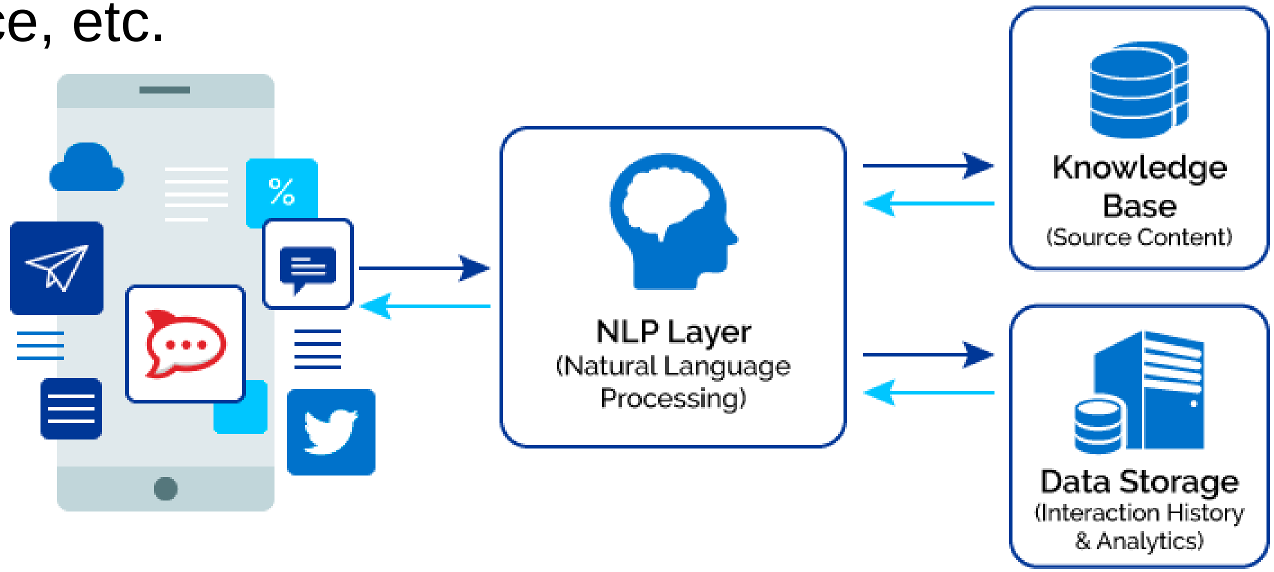


Seq2seq model



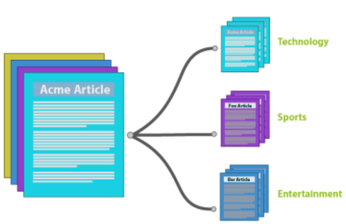
Textual Big Data alias The problem of the Natural Languale Processing - NLP

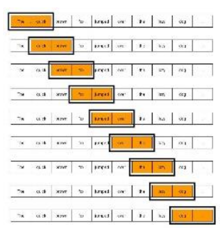
- Understanding **complex language utterances is one of the hardest challenge** for Artificial Intelligence (AI) and Machine Learning (ML).
- **NLP** is everywhere because people communicate most everything: web search, advertisement, emails, customer service, etc.

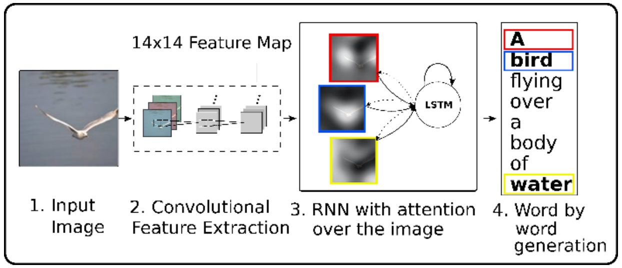


7 NLP applications where Deep Learning achieved «state-of-art» performance

- **1 <u>Text Classification</u>**: Classifying the topic or theme of a document (i.e. **Sentiment <u>Analysis</u>**).
- **2 <u>Language Modeling</u>**: Predict the **next word given the previous words**. It is fundamental for other tasks.
- **3 <u>Speech Recognition</u>**: Mapping an **acoustic signal** containing a spoken natural language utterance into the corresponding sequence of words intended by the speaker.
- **4 <u>Caption Generation</u>**: Given a **digital image**, such as a photo, generate a **textual description** of the contents of the image.





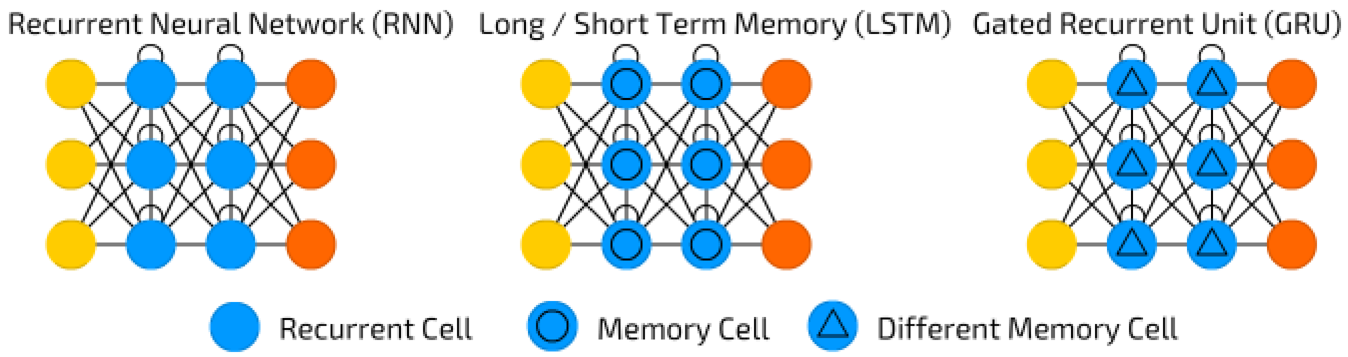


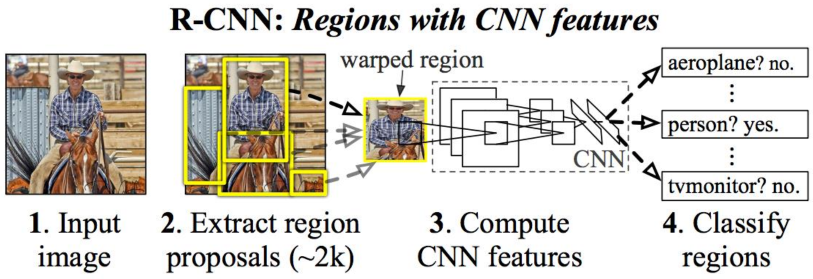
7 NLP applications where Deep Learning achieved «state-of-art» performance

- **5 <u>Machine Translation</u>**: Automatic translation of text or speech from one language to another, is one [of] the most important applications of NLP.
- **6 <u>Document Summarization</u>**: It is the task where a short description of a text document is created.
- **7 <u>Question Answering</u>**: It is the task where the system tries to answer a user query that is formulated in the form of a question by returning the appropriate noun phrase such as a location, a person, or a date. (i.e. Who killed President Kennedy? Oswald)

Text Classification Models

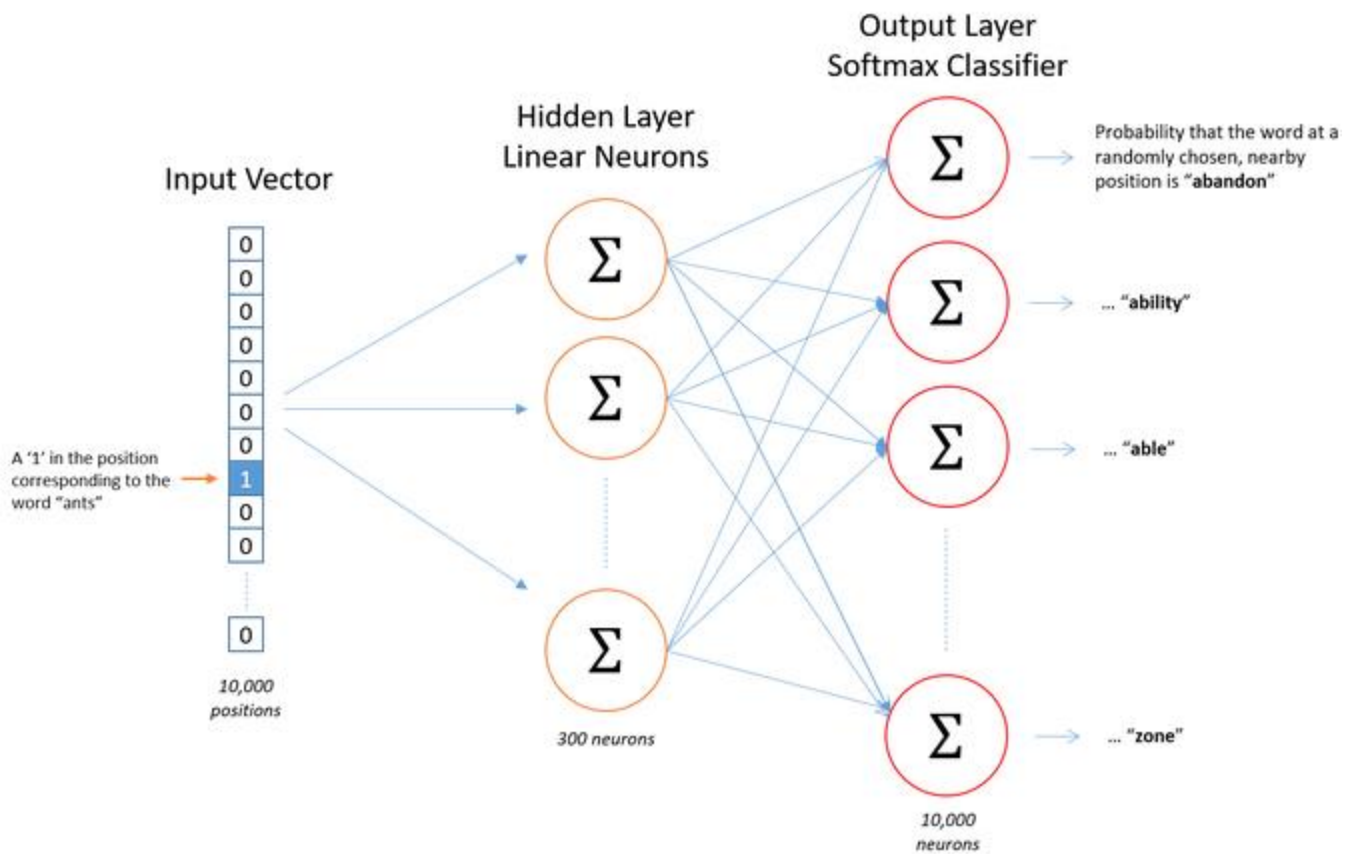
- **RNN, LSTM, GRU, ConvLstm, RecursiveNN, RNTN, RCNN**
- The modus operandi for text classification involves the use of a pre-trained **word embedding** for **representing words** and a **deep neural networks** for **learning how to discriminate documents** on classification problems.



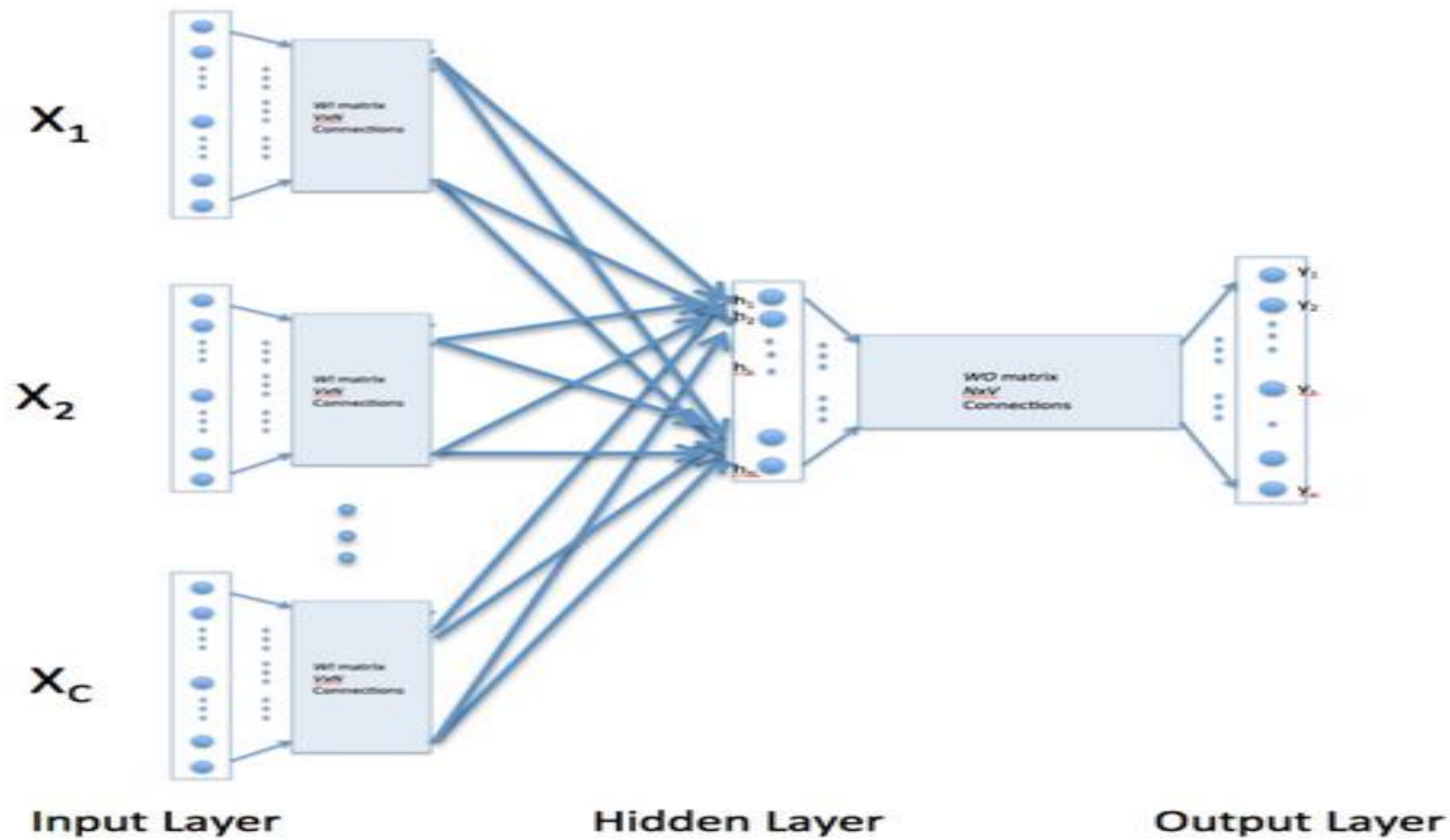


- The **non-linearity of the NN** leads to superior classification accuracy.

Skip-Gram Model (Mikolov, et. al., 2013)



C-BOW Model (Bow, et al., 2003).



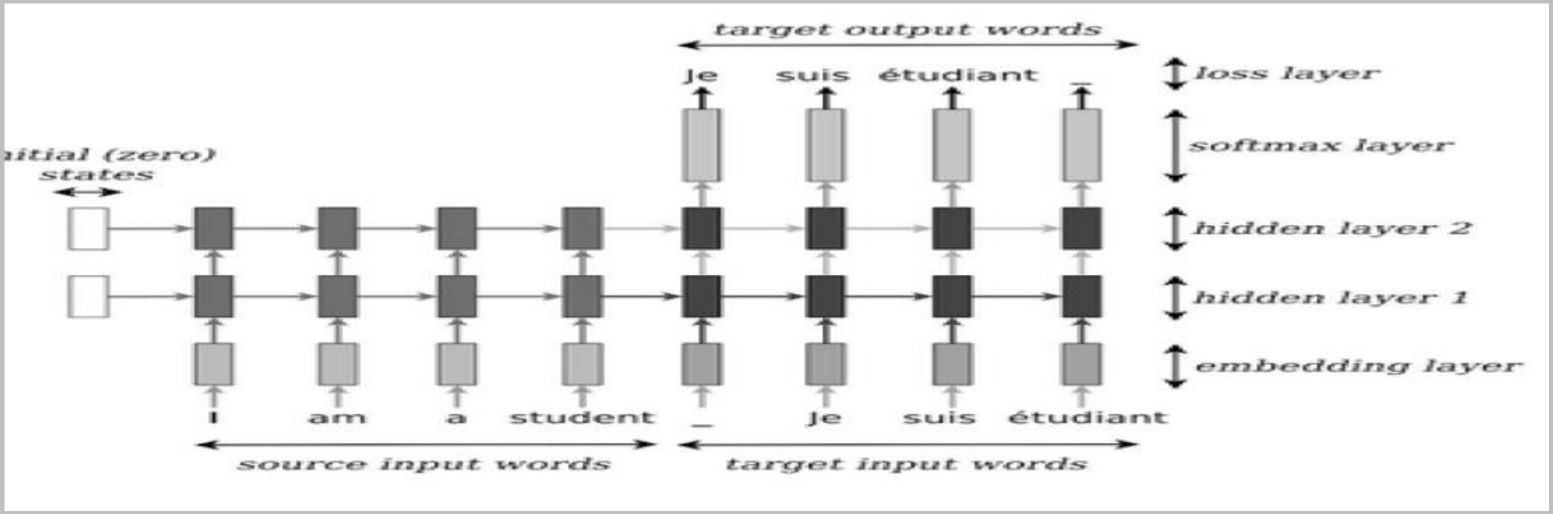
Sentiment Analysis (*Ain, et al. 2017*)

- **Sentiments** of users that are expressed on the web has great influence on the readers, product vendors and politicians.
- **Sentiment Analysis** refers to text organization for the classification of mind-set or feelings in different manners such as negative, positive, favorable, unfavorable, thumbs up, thumbs down, etc. Thanks to DL, the SA can be visual as well.

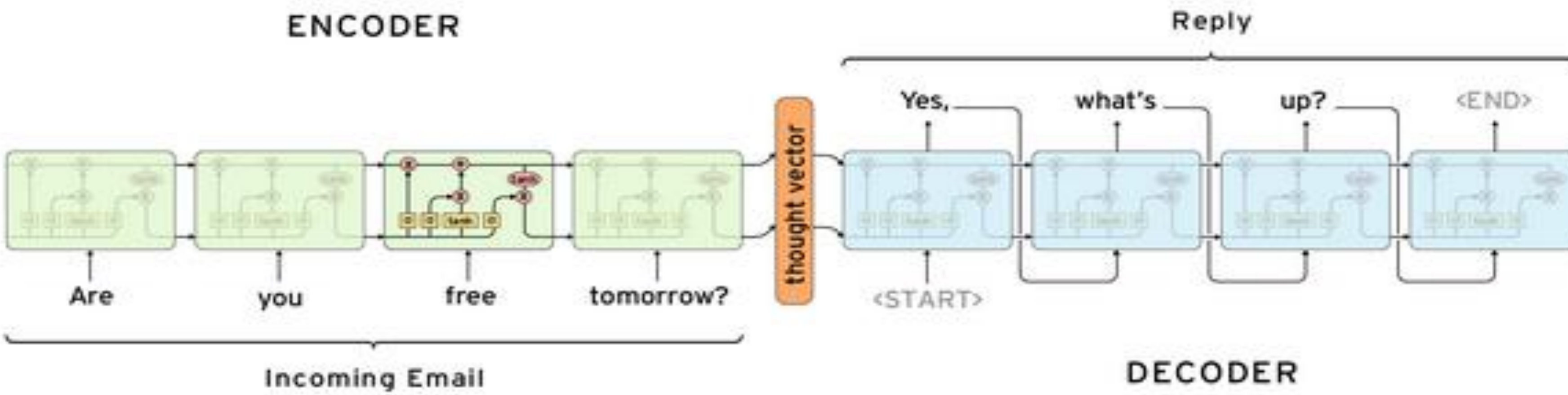
Discovering people opinions, emotions and feelings about a product or service

Neural Machine Translation (Bahdanau, et., al., 2014).

- **Neural machine translation (NMT) is an approach to “machine translation” that uses large ANN to predict the likelihood of a sequence of words, typically modeling entire sentences in a single integrated model (Bahdanau et al., 2014; Luong et Manning, 2016). .**



Neural Conversational Models (Vinyals, & Le., 2015).



Conversation model – chatbot?

- Training on a set of conversations. The input sequence can be the concatenation of what has been conversed so far (the context), and the output sequence is the reply.

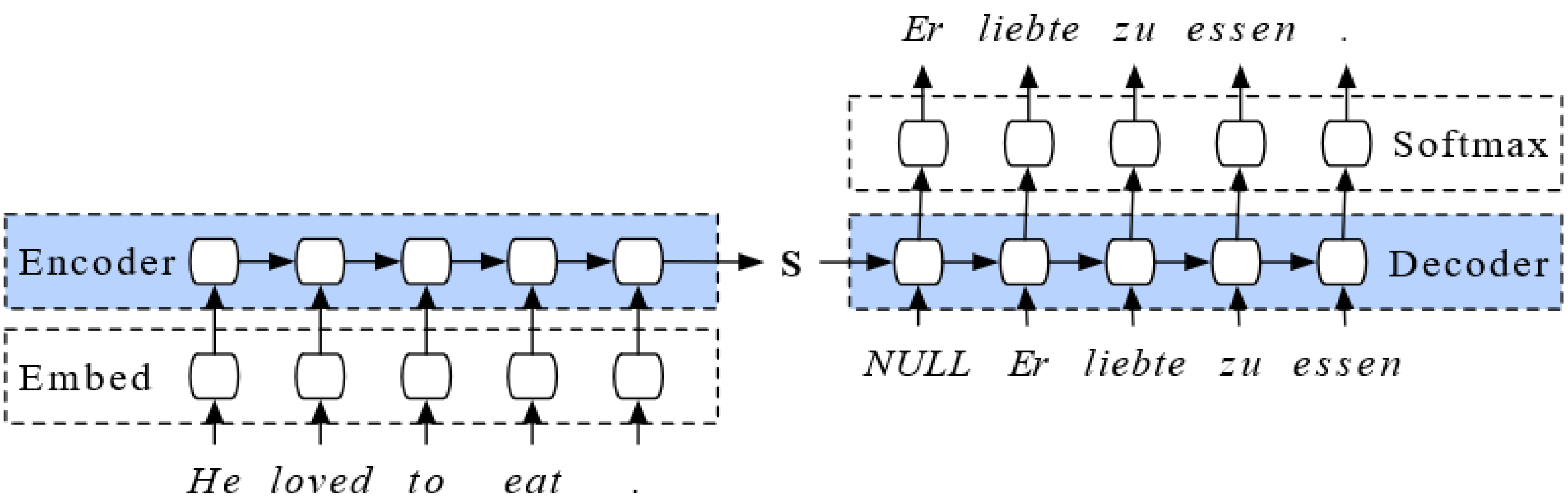
Machine: *is it a windows machine ?*
Machine: *we have linux on your machine .*
Machine: *is it windows xp or windows 7 ?*
Human: *i am using linux*
Machine: *let me take a remote of your machine and check if issue has been pushed properly .*
Human: *ok*
Machine: *please open terminal and run the below commands in one computer :*
Machine: *sudo apt - get update*
Machine: *sudo apt - get upgrade*
Machine: *sudo <COMMAND> - updater*
Human: *done*
Machine: *let me try connecting now*
Human: *ok*

Conversation 4: Philosophical Q&A

Human: *what is the purpose of life ?*
Machine: *to serve the greater good .*
Human: *what is the purpose of living ?*
Machine: *to live forever .*
Human: *what is the purpose of existence ?*
Machine: *to find out what happens when we get to the planet earth .*
Human: *where are you now ?*
Machine: *i 'm in the middle of nowhere .*
Human: *what is the purpose of dying ?*
Machine: *to have a life .*
Human: *what is the purpose of being intelligent ?*
Machine: *to find out what it is .*
Human: *what is the purpose of emotions ?*
Machine: *i don 't know .*

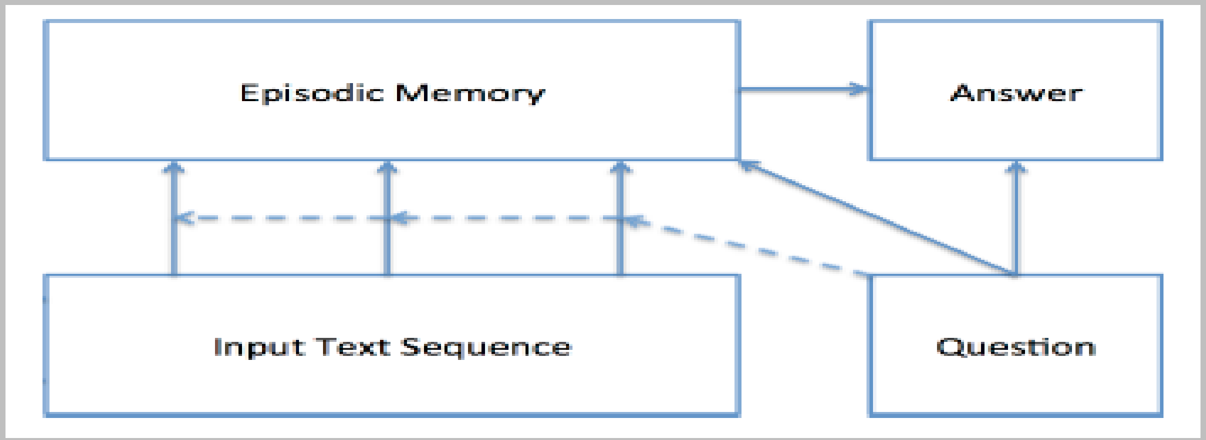
Neural Machine Translation

Input sentence:	Translation (PBMT):	Translation (GNMT):	Translation (human):
李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.



Dynamic Memory Networks (Kumar, et al., 2016).

- Dynamic Memory Networks (DMN) are a recurrent neural network architecture which processes input sequences and questions, forms episodic memories, and generates relevant answers. The DMN can be trained end-to-end and obtains state-of-the-art results on question answering (Facebook's bAbI dataset), text classification for sentiment analysis (Stanford Sentiment Treebank) and sequence modeling for part-of-speech tagging (WSJ-PTB).



USEFUL Links

- **EMOS PROJECT – Prof. Agostino di Ciaccio**

WebSite:
http://ec.europa.eu/eurostat/web/european-statistical-system/emos

REFERENCES

Chellappa, R. (2012). Mathematical statistics and computer vision. *Image and Vision Computing*, *30*(8), 467-468.

Tsukahara, J. S., Harrison, T. L., & Engle, R. W. (2016). The relationship between baseline pupil size and intelligence. *Cognitive Psychology*, *91*, 109-123.

Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, *195*(1), 215-243.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929-1958.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807-814).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Sánchez, J., Perronnin, F., Mensink, T., & Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, *105*(3), 222-245.

REFERENCES

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Abate, T., Van Huis, A., & Ampofo, J. K. O. (2000). Pest management strategies in traditional agriculture: an African perspective. *Annual review of entomology*, *45*(1), 631-659.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, *2*(1), 1-127

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, *17*(3), 37

Hinton, G. E., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. Neural Comput. 18, 1527–1554. doi: 10.1162/neco.2006.18.7.1527

Holland, J., H., (1975). Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor.

REFERENCES

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989)
Backpropagation applied to handwritten zip code recognition. Neural Comput., 1(4):541{551.

McCulloch, W., S., and Pitts, W., (1943).“A logical calculus of the ideas immanent in nervous activity, *Bulletin of mathematical biology*, *5*(4), 115-133.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81-106.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-9).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).

Johnson, R. C. (2015). Microsoft, Google beat humans at image recognition. *EE Times*.

Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 234-241). Springer International Publishing.

REFERENCES

Scott, A. J., & Knott, M. (1974). A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, 507-512.

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one, 12(7), e0180944.

Bodyanskiy Y, Popov S. (2006) Neural network approach to forecasting of quasiperiodic financial time series. Eur J Oper Res. 175(3):1357–66.

Nourani V, Komasi M, Mano A. (2009) A multivariate ANN-wavelet approach for rainfall-runoff modeling. Water Resources Management. 2009;23(14):2877.

Kim TY, Oh KJ, Kim C, Do JD. (2004) Artificial neural networks for non-stationary time series. Neurocomputing.

Vinyals, O., & Le, Q. (2015). A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kim TY, Oh KJ, Kim C, Do JD. Artificial neural networks for non-stationary time series. Neurocomputing. 2004;61(C):439–47.

REFERENCES

Hsieh TJ, Hsiao HF, Yeh WC. (2011) Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. Applied Soft Computing. 2011;11(2):2510–25. 61(C):439–47.

Sutskever I, Hinton GE (2008). Deep, narrow sigmoid belief networks are universal approximators. Neural Computation. 20(11):2629–36. pmid:18533819

.

Roux NL, Bengio Y. (2010) Deep Belief Networks Are Compact Universal Approximators. Neural Computation.22(8):2192–207.

Bengio Y, Lamblin P, Popovici D, Larochelle H. (2007) Greedy layer-wise training of deep networks. Advances in neural information processing systems. 19:153.

Ramsey JB. (1999) The contribution of wavelets to the analysis of economic and financial data. Philosophical Transactions of the Royal Society B Biological Sciences. 357(357):2593–606.

Palangi H, Ward R, Deng L. (2016) Distributed Compressive Sensing: A Deep Learning Approach. IEEE Transactions on Signal Processing. 64(17):4504–18.

Amari, S. I., Cichocki, A., & Yang, H. H. (1995, December). Recurrent neural networks for blind separation of sources. In *Proc. Int. Symp. NOLTA* (pp. 37-42).

REFERENCES

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Hochreiter S, Schmidhuber J. Long Short-Term Memory (1997). Neural Computation. 1997;9(8):1735–80. pmid:9377276

.

Bliemel F. Theil's (1973) Forecast Accuracy Coefficient: A Clarification. Journal of Marketing Research. 10(4):444.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., ... & Socher, R. (2016, June). Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning* (pp. 1378-1387).

Bow, C., Hughes, B., & Bird, S. (2003, July). Towards a general model of interlinear text. In *Proceedings of EMELD workshop* (pp. 11-13).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

REFERENCES

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In *Advances in Neural Information Processing Systems* (pp. 2234-2242).

.

https://www.thewayofmeditation.com.au/dalai-lama-turns-science-on-its-head

Master Executive di II Livello

BIG DATA ANALYSIS AND BUSINESS INTELLIGENCE

Grazie

Nome Docente

Francesco Pugliese, PhD

Italian National Institute of Statistics, Division "Information and Application Architecture", Directorate for methodology and statistical design

Email **Francesco Pugliese** : francesco.pugliese@istat.it

fondazione INUIT TOR VERGATA