Deep Learning & NLP

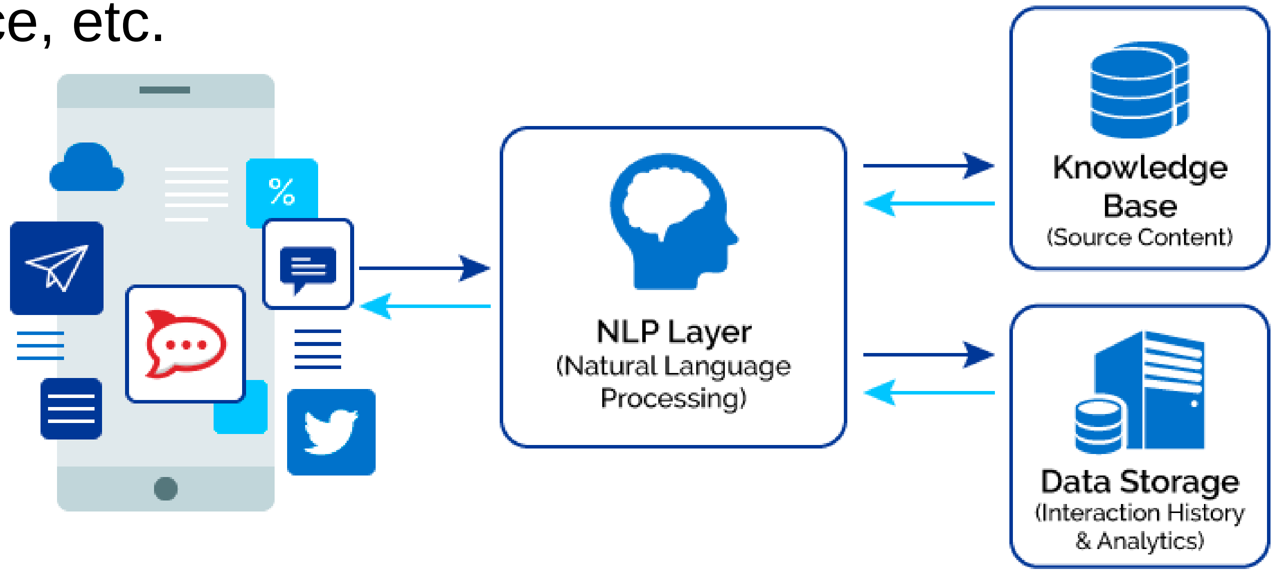
Francesco Pugliese, PhD

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

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

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.



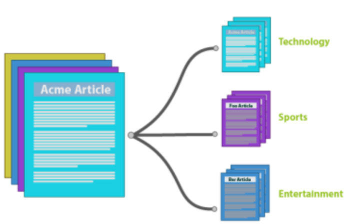
Deep Learning and NLP

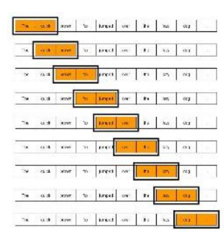
- "Deep Learning" approaches have obtained very high performance across many different **NLP** tasks. These models can often be trained with a **single end-to-end model** and do not require traditional, task-specific feature engineering.
  (*Stanford University School Of Engineering – CS224D*)

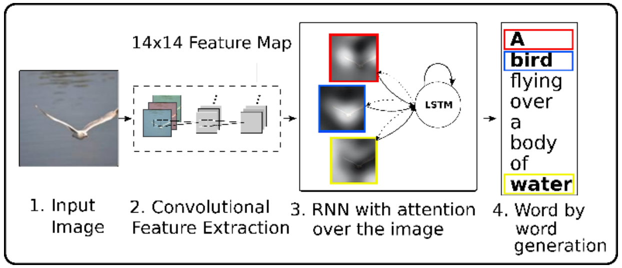
- **Natural language processing** is shifting from statistical methods to **Neural Networks**.

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.



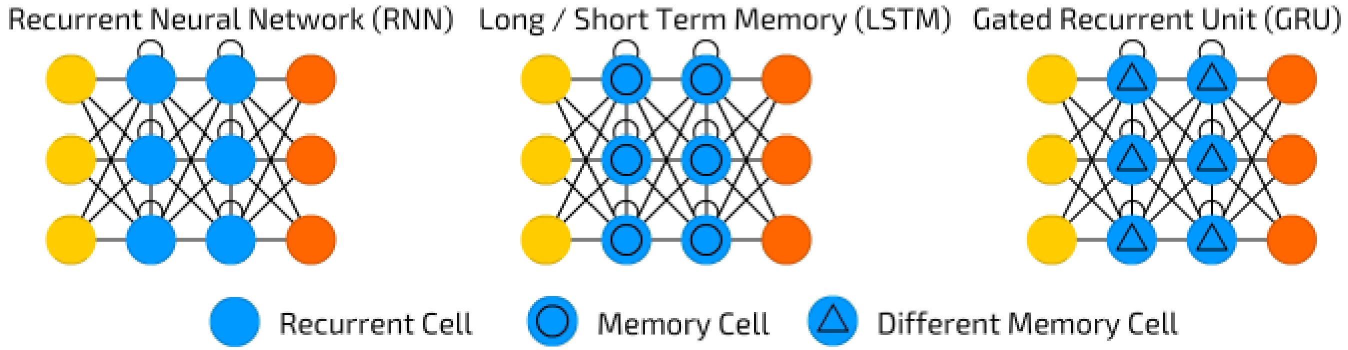


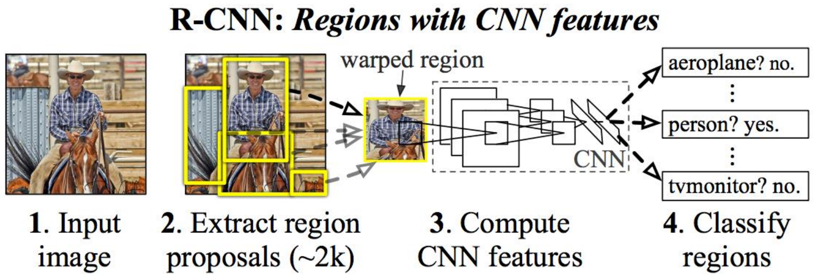
7 NLP applications where Deep Learning achieved «state-of-art» performance

- **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

- **<u>RNN, LSTM, GRU, ConvLstm, RecursiveNN, RNTN, RCNN</u>**
- 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.



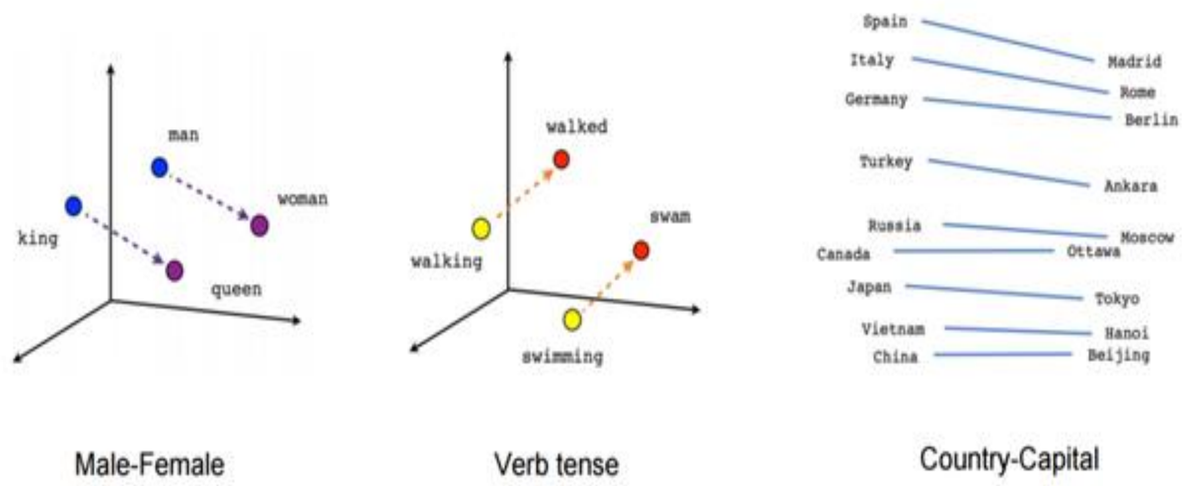


- The **non-linearity of the NN** leads to superior classification accuracy.

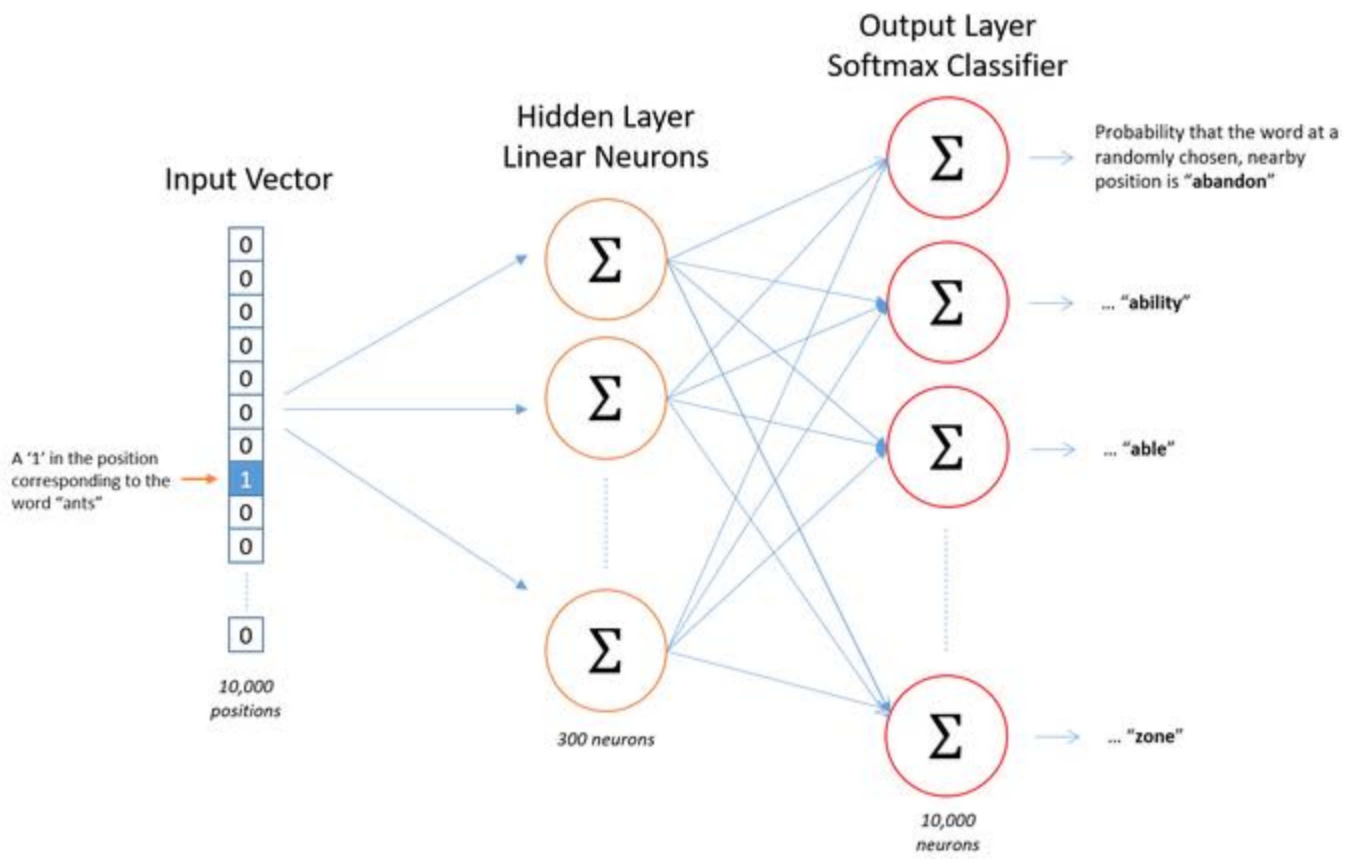
Word Embedding & Language Modeling

- 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.

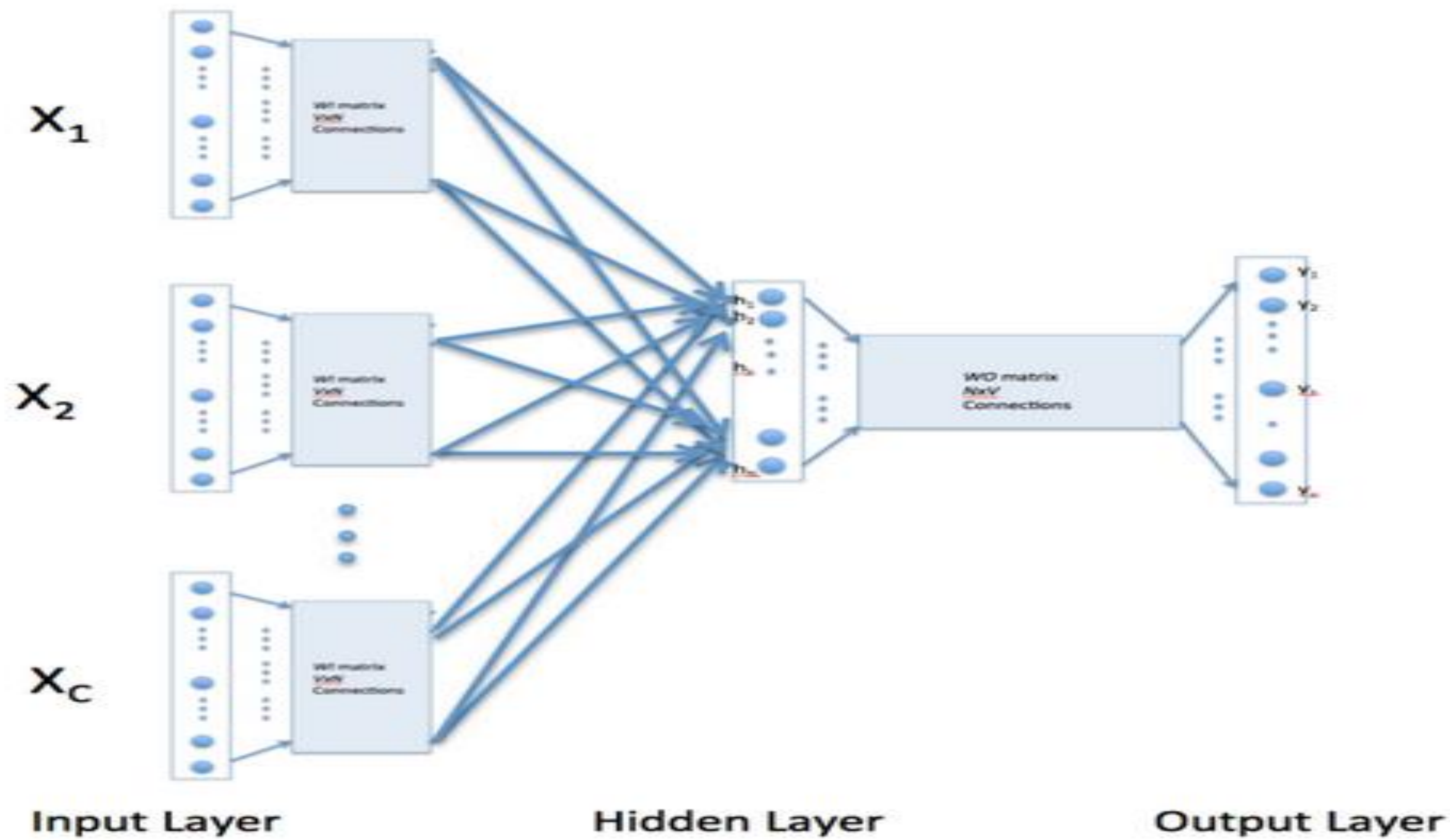
- These vectors are semantically cor-related by metrics like cosine distan-ce



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



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

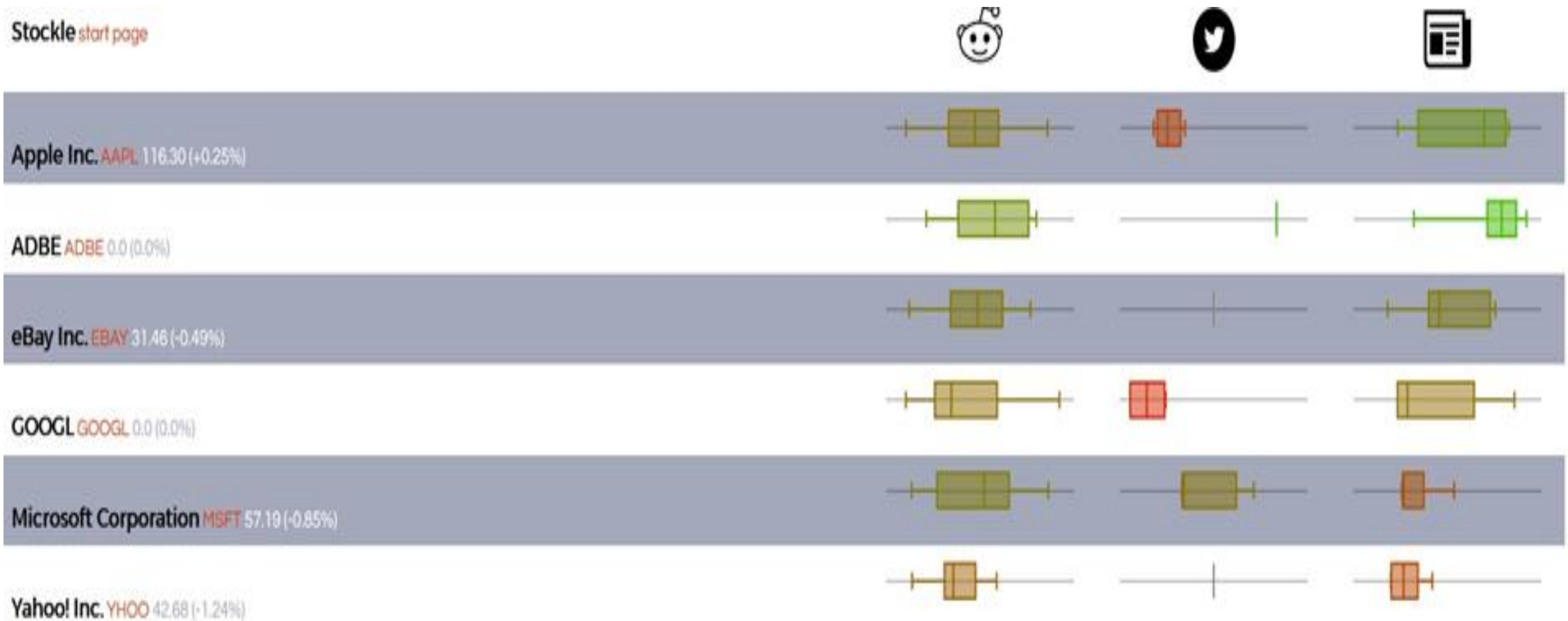


Sentiment Analysis (*Ain, et al. 2017*)

- **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.

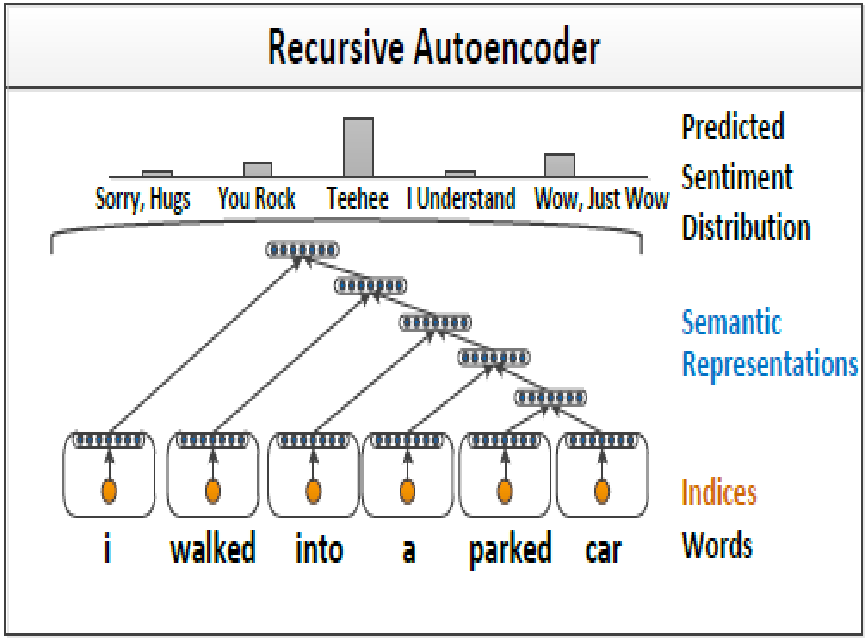
Discovering people opinions, emotions and feelings about a product or service

Sentiment Analysis with Feedback



Recursive Neural Tensor Networks (RecursiveNN) (Socher, R., et al., 2011b)

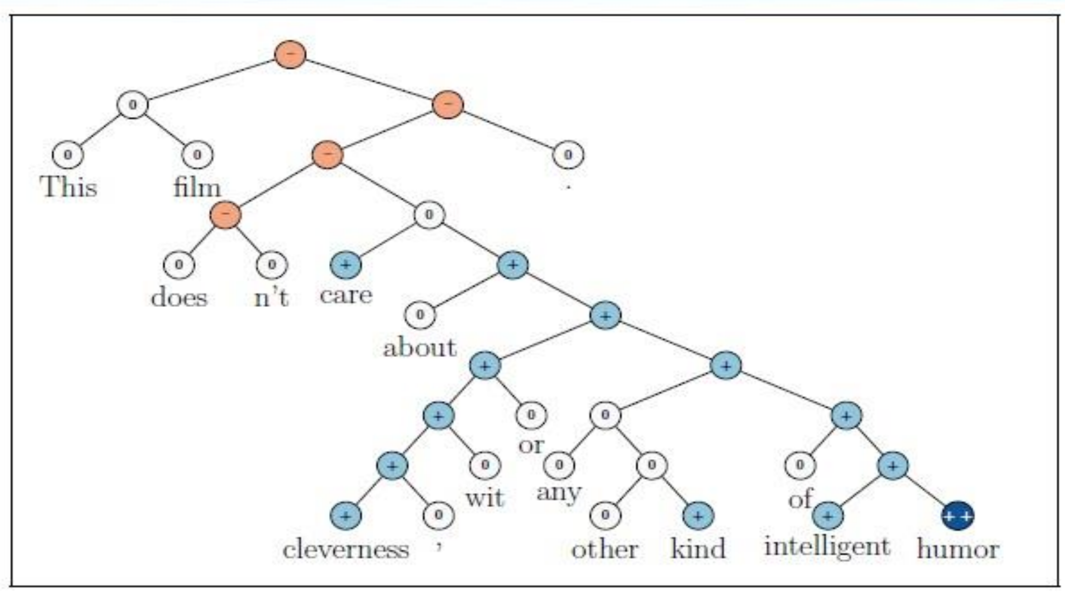
- This models are recursive auto-encoders which learn semantic vector representations of phrases. Word indices (orange) are first mapped into a semantic vector space (blue).
- Then they are recursively merged by the same auto-encoder network into a fixed length sentence representation. The vectors at each node are used as features to predict a distribution over text labels.



Recursive Neural Tensor Networks (RNTN) (Socher, R., et al. 2013)

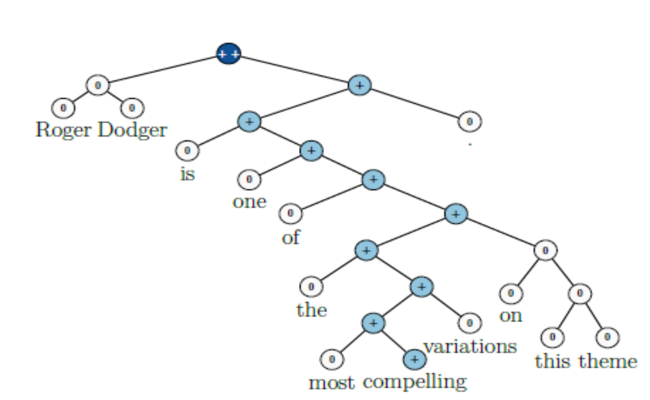
- **The Stanford Sentiment Treebank is the first corpus with fully labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in language.**

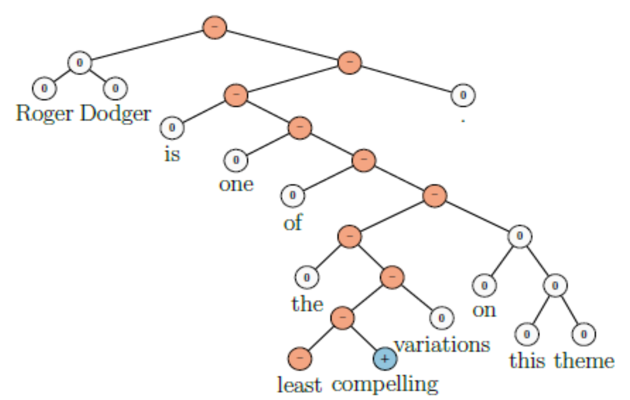
- **RNTNs compute parent vectors in a bottom up fashion using a compositionality function and use node vectors as features for a classifier at that node.**



RNTN – Upside and Downside

- RNTNS are very efficient in terms of constructing sentence representations.
- RNTNs capture the semantics of a sentence via a tree structure. Its performance heavily depends on the performance of the textual tree construction.
- Constructing such a textual tree exhibits a time complexity of at least O(n2), where n is the length of the text.
- RNTNs are unsuitable for modeling long sentences or documents.



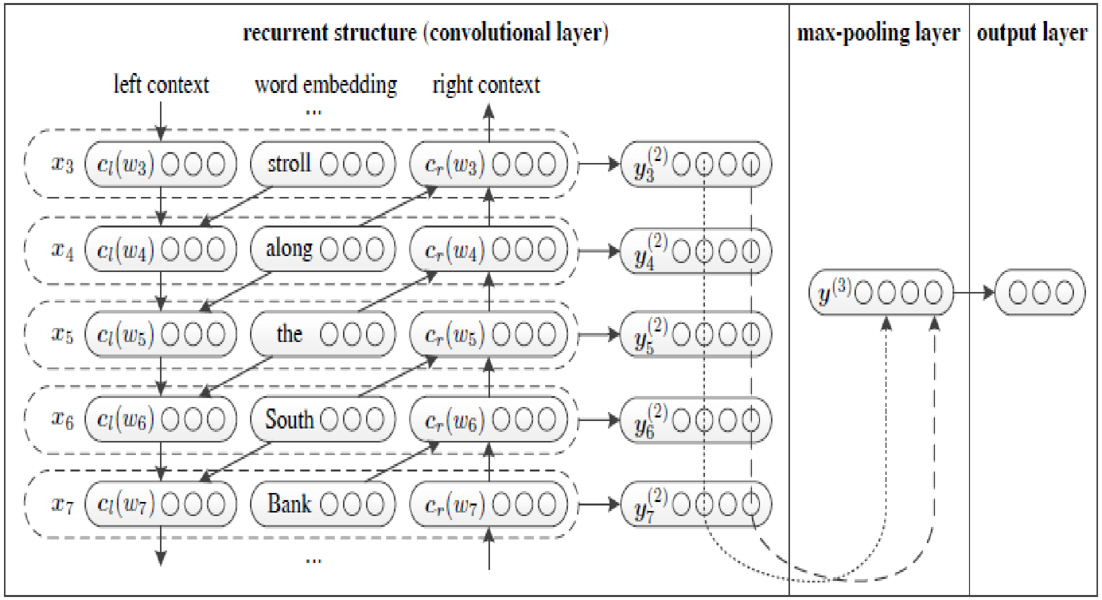


Recurrent Convolutional Neural Networks (RCNN) (*Lai, S., et al. 2015*)

- They adopt a a recurrent structure to **capture contextual information** as far as possible when learning word representations, which may introduce considerably **less noise compared** to traditional window-based neural networks.

- The **bi-directional recurrent structure** of RCNNs.

- **RCNNs** exhibit a time complexity of O(n)



RCNN Equations

- RCNNs exhibit a **time complexity of O(n)** , which is linearly correlated with the length of the text length.

$$c_l(w_i) = f(W^{(l)} c_l(w_{i-1}) + W^{(sl)} e(w_{i-1})) \quad (1)$$

$$c_r(w_i) = f(W^{(r)} c_r(w_{i+1}) + W^{(sr)} e(w_{i+1})) \quad (2)$$

$$x_i = [c_l(w_i); e(w_i); c_r(w_i)] \quad (3)$$

$$y_i^{(2)} = \tanh\left(W^{(2)} x_i + b^{(2)}\right) \quad (4)$$

$$y^{(3)} = \max_{i=1}^{n} y_i^{(2)} \quad (5)$$

$$y^{(4)} = W^{(4)} y^{(3)} + b^{(4)} \quad (6)$$

$$p_i = \frac{\exp\left(y_i^{(4)}\right)}{\sum_{k=1}^{n} \exp\left(y_k^{(4)}\right)} \quad (7)$$

- **7 equations** defining all the Neural Network topology

- **Input length** can be variable

RCNN in Keras

```
class SentimentModelRecConvNet:
    @staticmethod

    def build(input_length, vector_dim):
        hidden_dim_RNN = 200
        hidden_dim_Dense = 100

        embedding = Input(shape=(input_length, vector_dim))

        left_context = LSTM(hidden_dim_RNN, return_sequences = True)(embedding)                    # Equation 1
        # left_contex: batch_size x tweet_length x hidden_state_dim
        right_context = LSTM(hidden_dim_RNN, return_sequences = True, go_backwards = True)(embedding)  # Equation 2
        # right_cntext: come left_contex
        together = concatenate([left_context, embedding, right_context], axis = 2)                 # Equation 3
        semantic = TimeDistributed(Dense(hidden_dim_Dense, activation = "tanh"))(together)         # Equation 4
        pool_rnn = Lambda(lambda x: backend.max(x, axis = 1), output_shape = (hidden_dim_Dense, ))(semantic)  # Equation 5
        pool_rnn_args = Lambda(lambda x: backend.argmax(x, axis=1), output_shape = (hidden_dim_Dense, ))(semantic)

        output = Dense(1, input_dim = hidden_dim_Dense, activation = "sigmoid")(pool_rnn)           # Equations 6, 7

        deepnetwork = Model(inputs=embedding, outputs=output)
        deepnetwork_keywords = Model(inputs=embedding, outputs=pool_rnn_args)

        return [deepnetwork, deepnetwork_keywords]
```

RCNN: Feature Extraction

- RCNNs employ a max-pooling layer that automatically judges which words play key roles in text classification to capture the key components in texts.

- The most important words are the information most frequently selected in the max-pooling layer.

- Contrary to the most positive and most negative phrases in RNTN, RCNN does not rely on a syntactic parser, therefore, the presented n-grams are not typically “phrases”.

	RCNN
P	well *worth* the; a *wonderful* movie; even *stinging* at; and *invigorating* film; and *ingenious* entertainment; and *enjoy* .; ’s *sweetest* movie
N	A *dreadful* live-action; Extremely *boring* .; is *n’t* a; ’s *painful* .; Extremely *dumb* .; an *awfully* derivative; ’s *weaker* than; incredibly *dull* .; very *bad* sign;
	RNTN
P	an amazing performance; most visually stunning; wonderful all-ages triumph; a wonderful movie
N	for worst movie; A lousy movie; a complete failure; most painfully marginal; very bad sign

RCNN applied to Extractive Text Summarization

- **Best keywords** lead to **best contextes** ---> **Summarization**

```
Tweet 29: "Giã  avete letto 136 pagine del piano scuola? #Fenomeni #labuonascuola"

Sentiment: -0.95 - -1
Keywords: pagine, avete, fenomeni, piano

Tweet 30: "\"Per l'#aternanza #scuola #lavoro bisogna passare da 11a 100milioni di euro\" #labuonascuola http://t.co/zGAzkni8rv"

Sentiment: -0.81 - -1
Keywords: euro, t, scuola, lavoro



Most significant keywords driving the sentiment decision:

Eccolo

Siamo

Scuola

Giuste

Escluso


Most significant sentences driving the sentiment decision:

...cambierã  solo se noi metteremo al centro...

...solo se noi metteremo al centro la...

...piã¹ grande spettacolo mai visto passodopopasso scuola...

...mai visto passodopopasso scuola labuonascuola...

...nessuno si senta escluso la buona scuola...
```

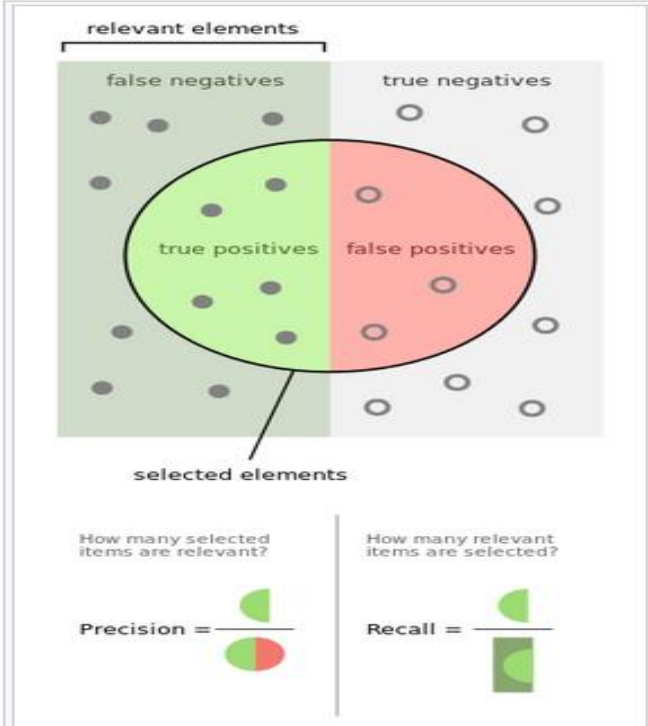
Recurrent Neural Networks are able to understand negations and other things

- Thanks to **word embeddings** semantics **RNNs** can recognize **nagations**, and complex **forms** of **language utterances**.

```
Tweet: This is a bad thing
- Sentiment: -0.72 - -1

Keywords: bad, thing, a, is

Tweet: This is not a bad thing
- Sentiment: 0.46 - +1

Keywords: not, thing, bad, a

Tweet: This is a positive thing
- Sentiment: 0.94 - +1

Keywords: positive, thing, a, is

Tweet: This is a very positive thing
- Sentiment: 0.91 - +1

Keywords: positive, very, thing, a

Tweet: I like Renzi politics
- Sentiment: 0.70 - +1

Keywords: like, renzi, politics, i

Tweet: I don't agree with Renzi Politics
- Sentiment: 0.16 - 0

Keywords: don't, agree, politics, renzi

Tweet: Renzi did a wrong international Politics
- Sentiment: -0.34 - -1

Keywords: wrong, did, renzi, international

Tweet: Renzi did a very good international Politics
- Sentiment: 0.74 - +1

Keywords: did, renzi, good, very

Tweet: Istat is a very good Institute of research
- Sentiment: 0.84 - +1

Keywords: good, very, research, istat

Tweet: Istat is not a good Institute of research - Sentiment: -0.78 - -1

Keywords: not, research, istat, institute
```

Classification Metrics

F-score



Precision and recall

sensitivity, recall, hit rate, or true positive rate (TPR

$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

specificity or true negative rate (TNR)

$$\mathrm{TNR} = \frac{\mathrm{TN}}{N} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

precision or positive predictive value (PPV)

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

negative predictive value (NPV)

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}}$$

miss rate or false negative rate (FNR)

$$\mathrm{FNR} = \frac{\mathrm{FN}}{P} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} = 1 - \mathrm{TPR}$$

fall-out or false positive rate (FPR)

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR}$$

false discovery rate (FDR)

$$\mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TP}} = 1 - \mathrm{PPV}$$

false omission rate (FOR)

$$\mathrm{FOR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TN}} = 1 - \mathrm{NPV}$$

accuracy (ACC)

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{P + N} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

F1 score

is the harmonic mean of precision and sensitivity

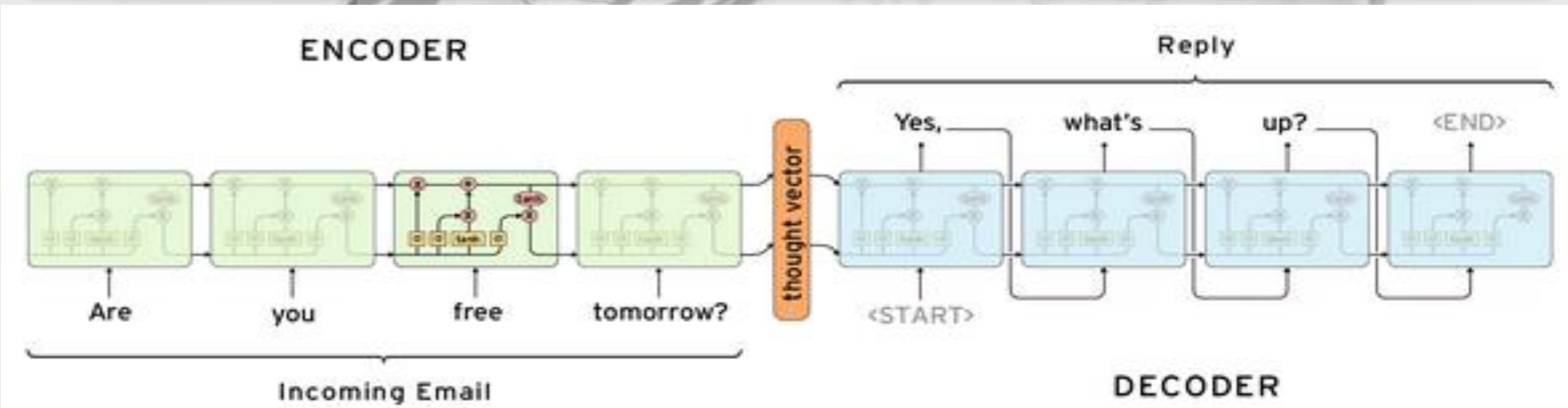
$$F_1 = 2 \cdot \frac{\mathrm{PPV} \cdot \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

Use Case 2: Classification of Cifar 10 with CNN in Keras

Metrics

<table>
<tr><td></td><td></td><td colspan="2">True condition</td><td></td><td colspan="2"></td></tr>
<tr><td></td><td>Total population</td><td>Condition positive</td><td>Condition negative</td><td>Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$</td><td colspan="2">Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$</td></tr>
<tr><td rowspan="2">Predicted condition</td><td>Predicted condition positive</td><td>True positive, Power</td><td>False positive, Type I error</td><td>Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$</td><td colspan="2">False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$</td></tr>
<tr><td>Predicted condition negative</td><td>False negative, Type II error</td><td>True negative</td><td>False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$</td><td colspan="2">Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$</td></tr>
<tr><td></td><td></td><td>True positive rate (TPR), Recall, Sensitivity, probability of detection $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$</td><td>False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$</td><td>Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$</td><td rowspan="2">Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$</td><td rowspan="2">$F_1$ score = $\frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$</td></tr>
<tr><td></td><td></td><td>False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$</td><td>True negative rate (TNR), Specificity (SPC) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$</td><td>Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$</td></tr>
</table>

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.

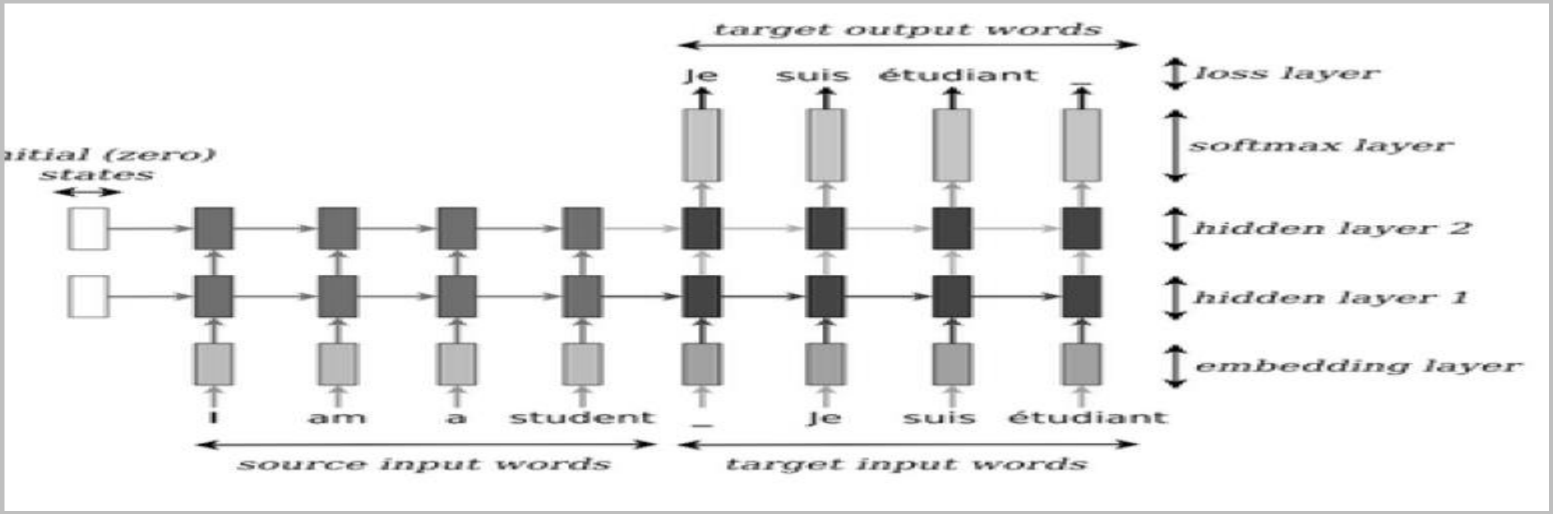
Machine: *is it a windows machine ?*
Machine: *we have linux on your machine .*
Machine: *is it windows xp or windows 7 ?*
Human: *i am using linux*
Machine: *let me take a remote of your machine and check if issue has been pushed properly .*
Human: *ok*
Machine: *please open terminal and run the below 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 (Bahdanau, et., al., 2014).

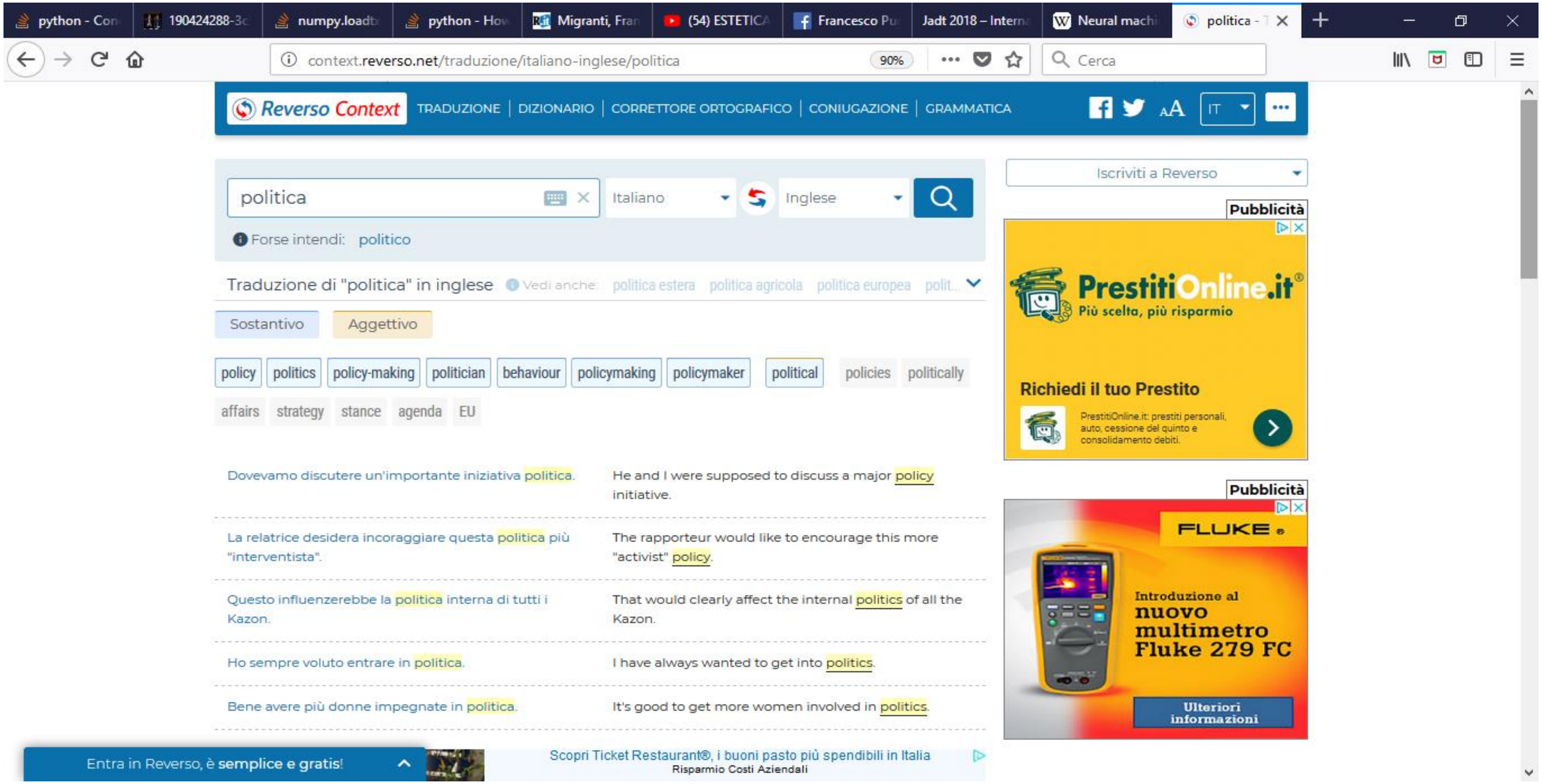
- **Neural machine translation (NMT) is an approach to “machine translation” that uses large ANN to predict the likelihood of a sequence of words, typically modeling entire sentences in a single integrated model (Bahdanau et al., 2014; Luong et Manning, 2016). .**



Multilingual Sentiment Analysis

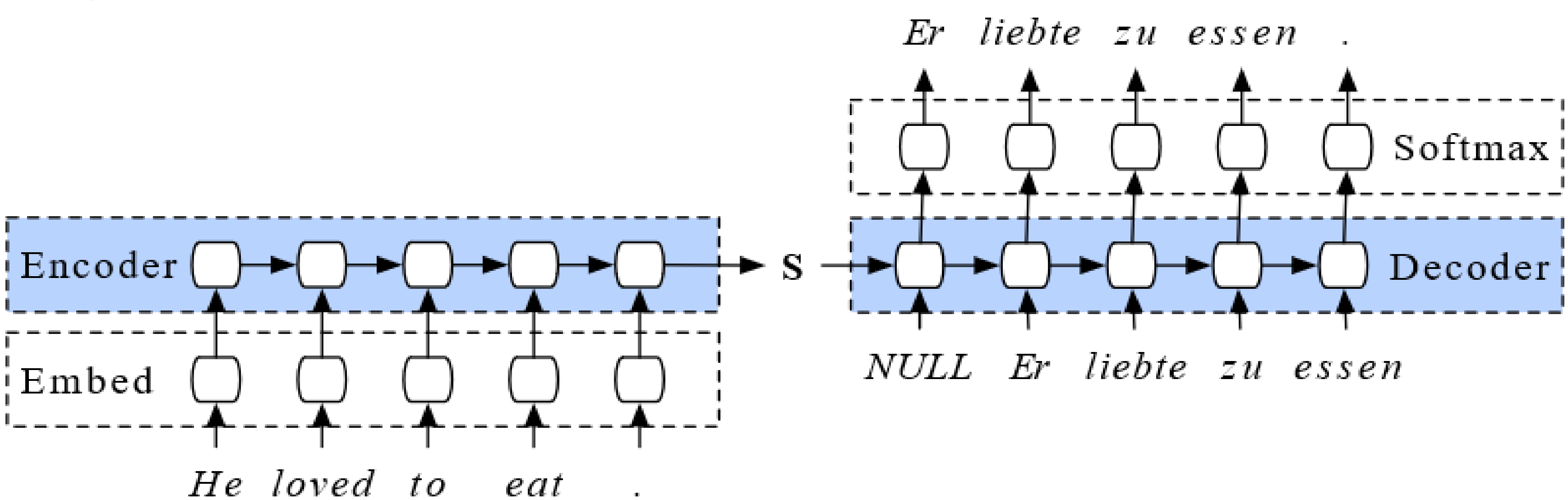
- **During the training stage, the RCNN achieves 84% of accuracy on a validation set (selected at the 20% of the original dataset). On a test set of 380 tweets (provided by Semeval), the model returns around 82% of accuracy on positive tweets and 78% of accuracy on negatives, with an approximative 80% overall on a mixed tweets set.**
- **During the training we determined 3.2 millions of keywords, namely 2 for each tweet, the most important and the second in order of signinificancy.**

Contextual Translations Web-sites



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.



Neural Machine Translation

adottare un vocabolario condiviso è un suggerimento perfetto su come scrivere frasi comprensibili
adopt a shared vocabulary is a perfect suggestion on how to write understandable sentences

un altro suggerimento su come scrivere frasi semplici: evita le negazioni inutili
another suggestion about how to write simple sentences : avoid unnecessary <unk>

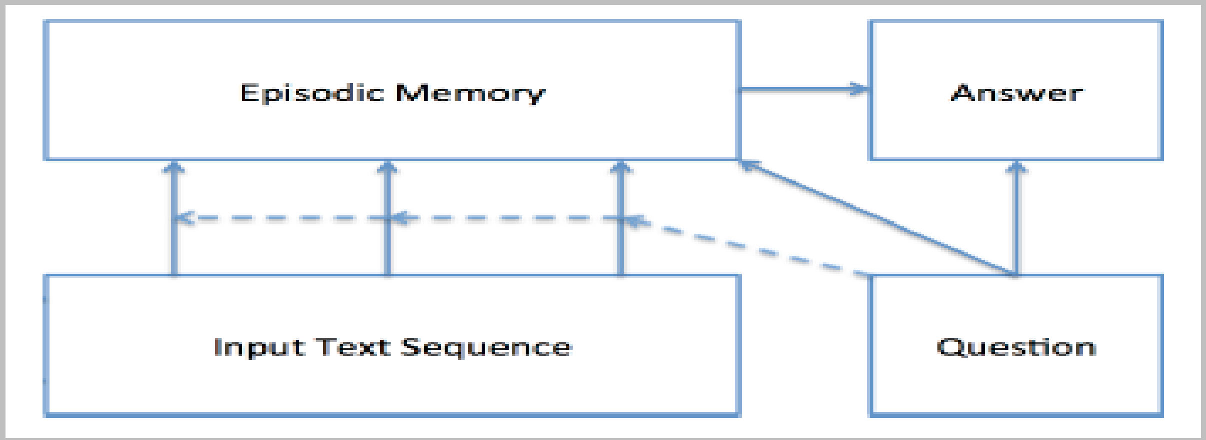
quasi 90 persone sono morte per una tempesta tropicale nelle filippine
nearly 90 people died for a tropical storm in the philippines

Figure 3. Some translations from Italian to English by means of the neural model trained by us.

- We have tested the English RCNN model on the same italian SENTIPOLC 2016 test-set translated into English by our neural machine translation model. Results highlight a boost of performance : **78%** of accuracy on the test set versus the **43%** of the Italian trained RCNN model proving our strategy of stacking NMT and RCNN models is successful.

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).



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

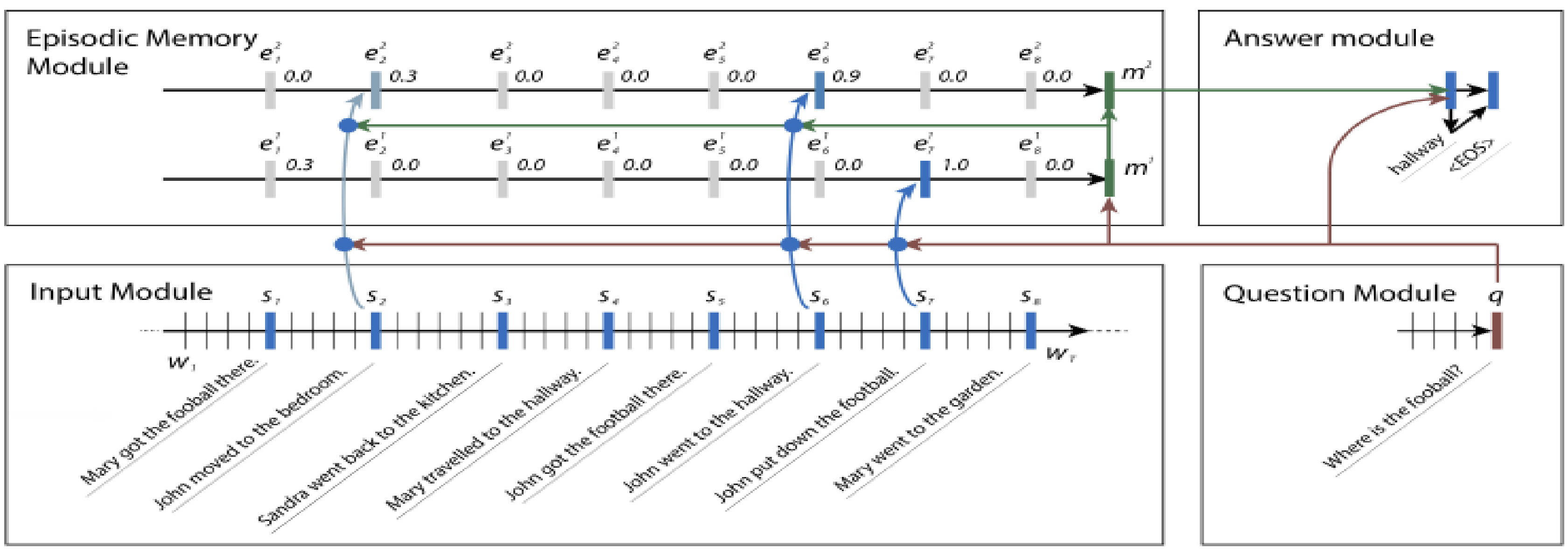
Task 2: Two Supporting Facts
John is in the playground.
John picked up the football.
Bob went to the kitchen.
Where is the football? A:playground

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

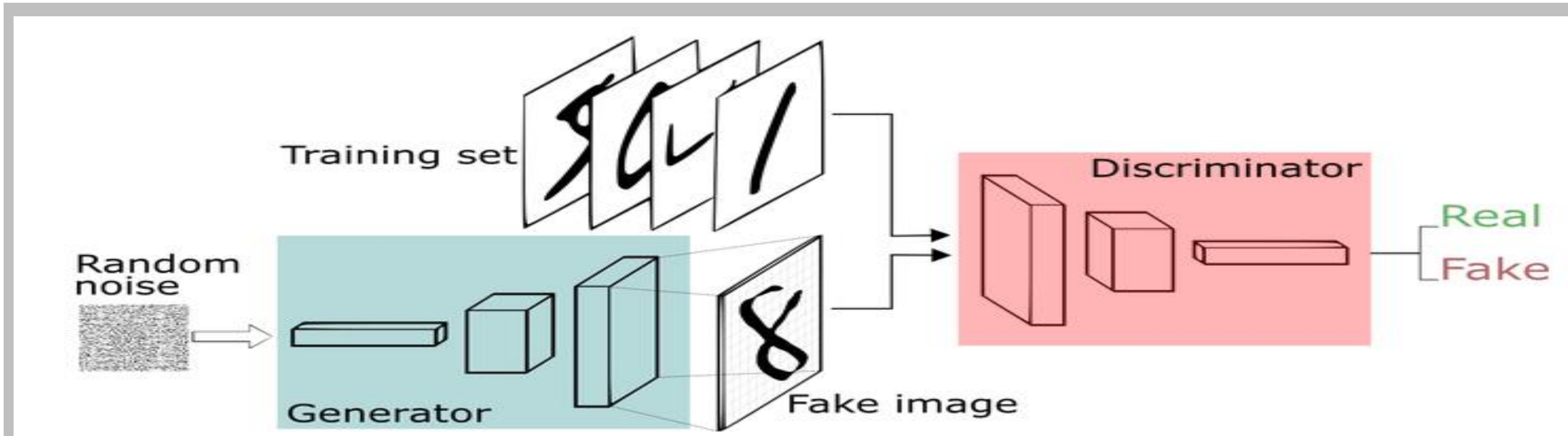
Task 4: Two Argument Relations
The office is north of the bedroom.
The bedroom is north of the bathroom.
The kitchen is west of the garden.
What is north of the bedroom? A: office
What is the bedroom north of? A: bathroom

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

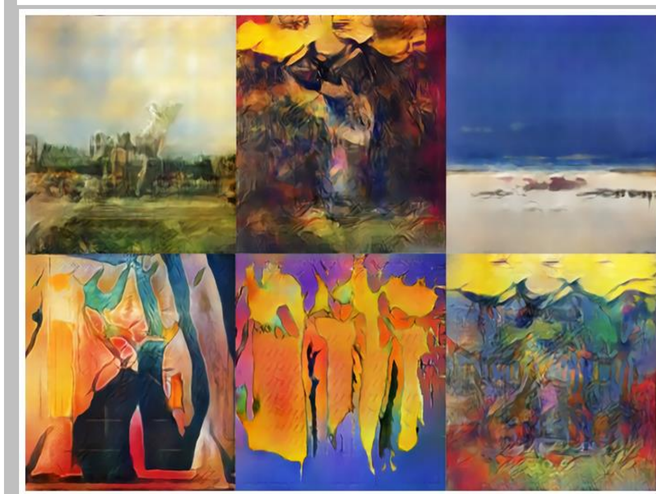
Task 6: Yes/No Questions
John moved to the playground.
Daniel went to the bathroom.
John went back to the hallway.
Is John in the playground? A:no
Is Daniel in the bathroom? A:yes

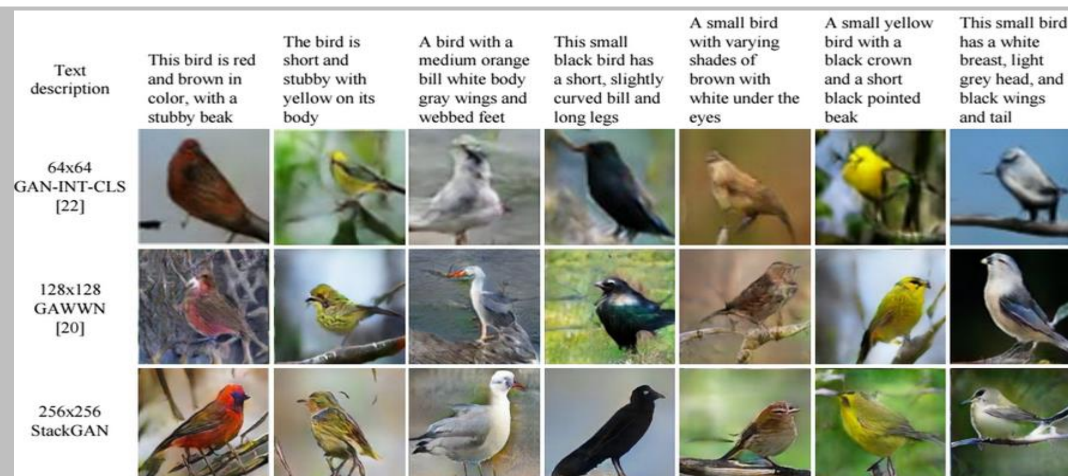


Generative Adversarial Networks (GAN) (Goodfellow, et al.,2014)

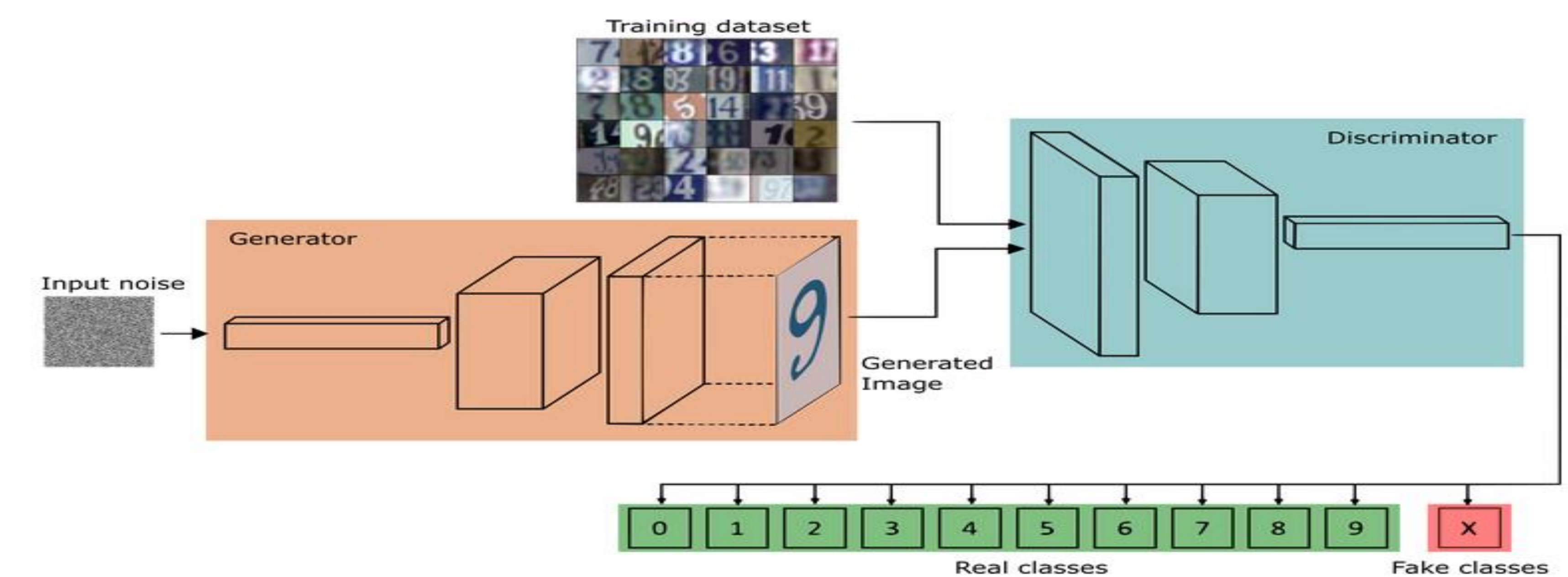


$$\min_{G} \max_{D} V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$





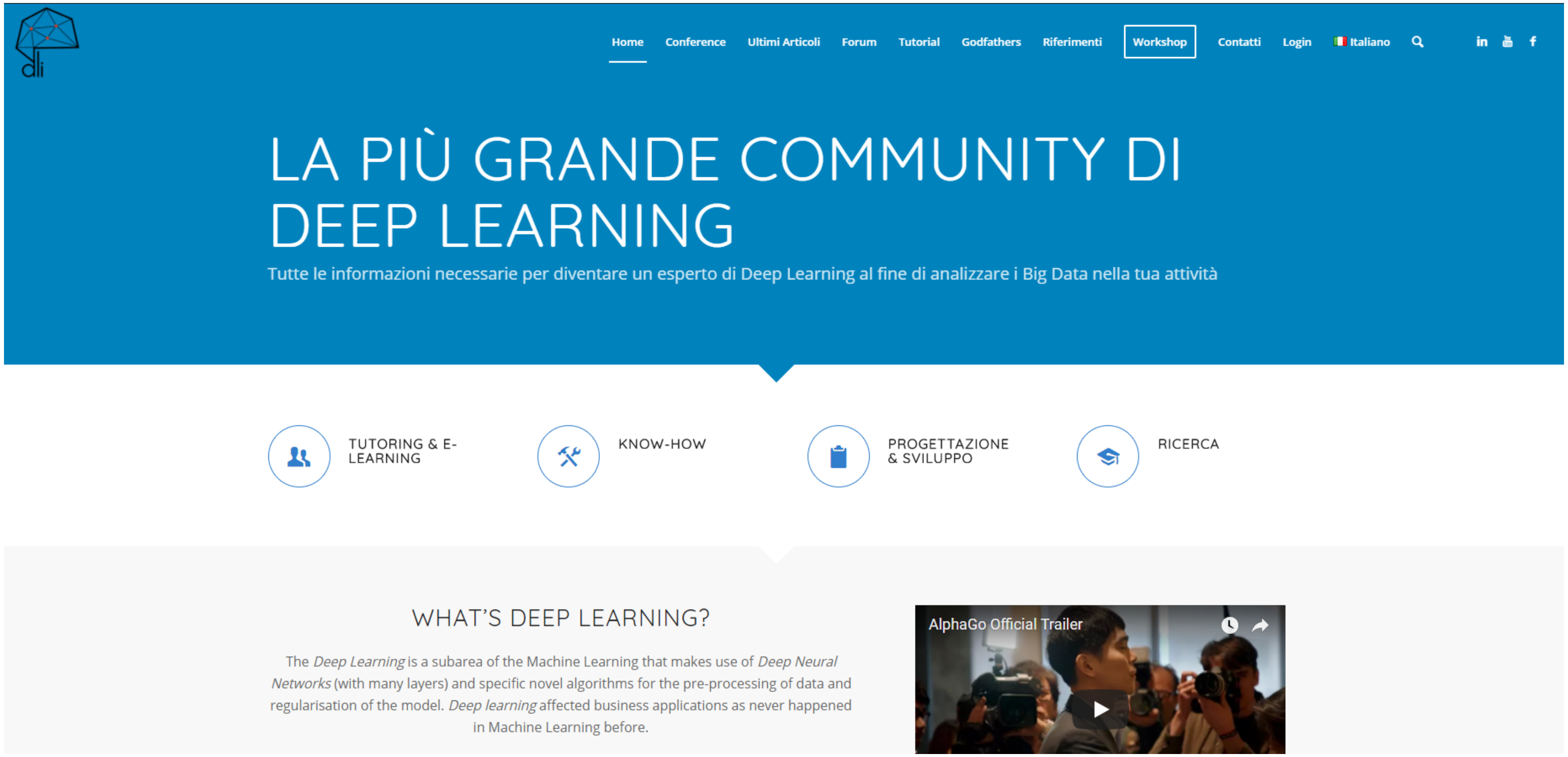
Generative Adversarial Networks (GAN) for Supervised Learning (Salimans, 2016)



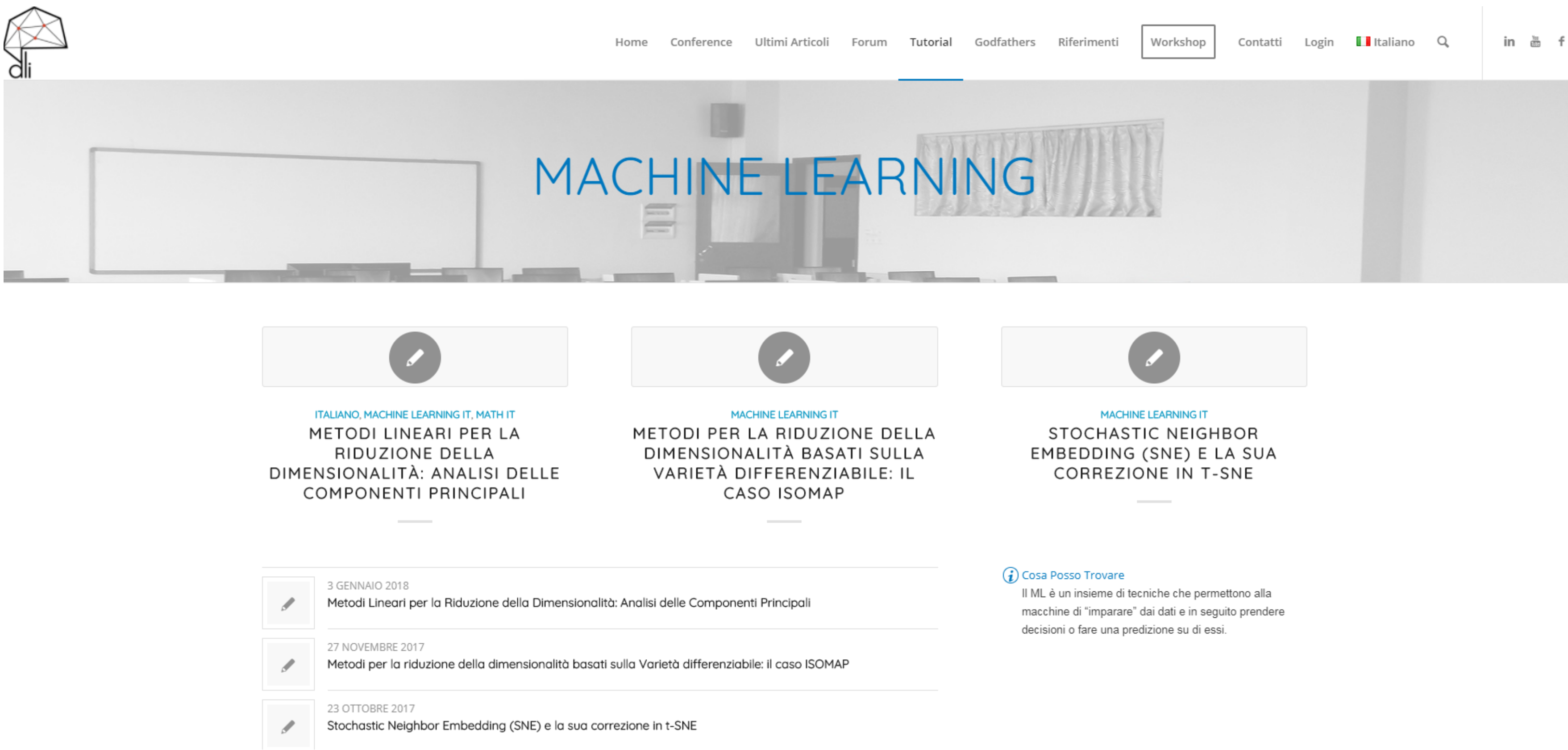
$$L = -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)}[\log p_{\text{model}}(y|\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim G}[\log p_{\text{model}}(y = K+1|\boldsymbol{x})]$$
$$= L_{\text{supervised}} + L_{\text{unsupervised}}, \text{ where}$$
$$L_{\text{supervised}} = -\mathbb{E}_{\boldsymbol{x},y\sim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

The Deep Learning Italia Project

- A competence-sharing web-site designed **exclusively** for **Deep Learning**
- An **e-learning platform** for the disclosure of Deep Learning
- A **collector of professionals** around Deep Learning topics
- In the next future it will become a **complete development suite for Deep Learning**

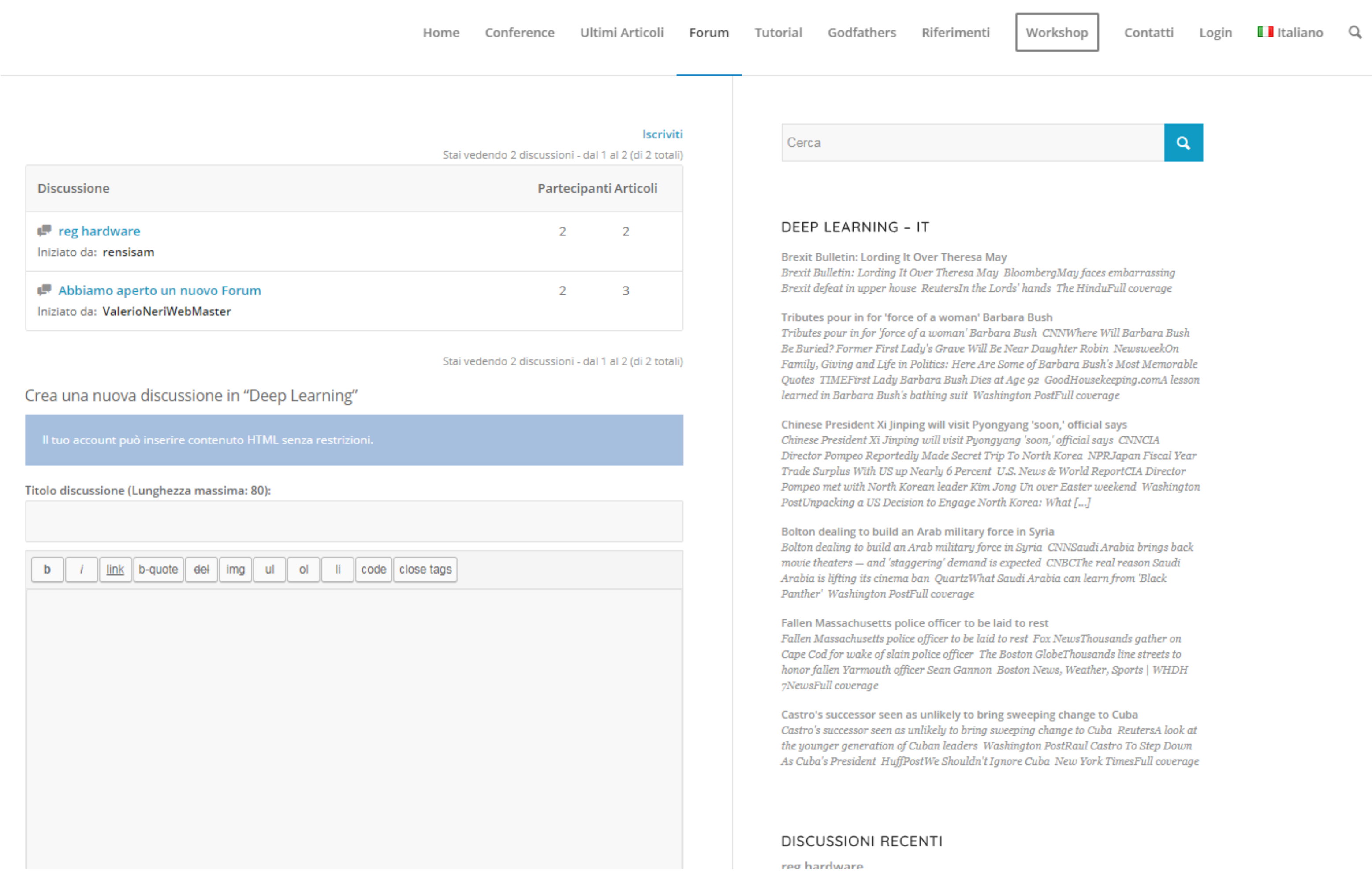


Tutorials

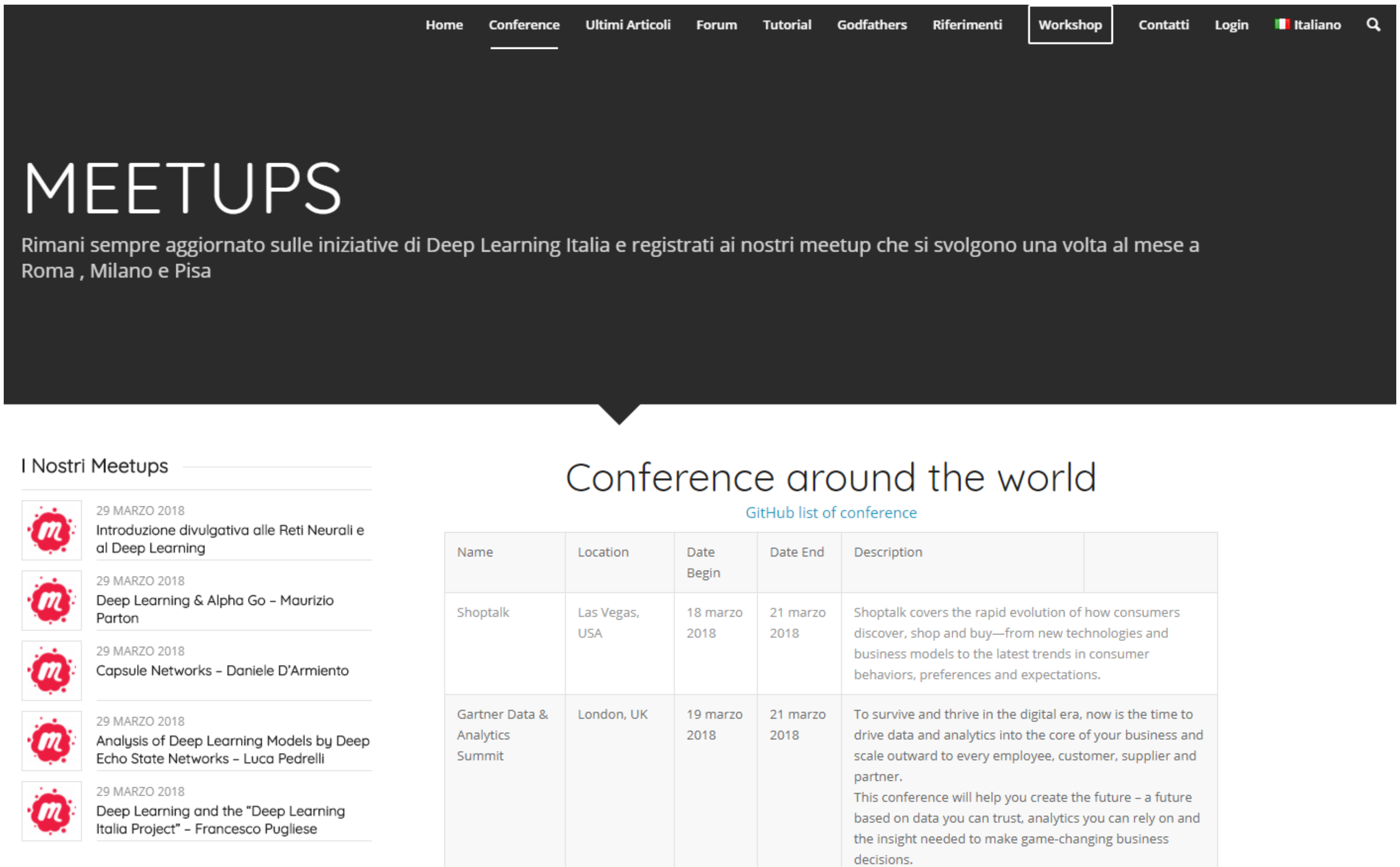


Home
Conference
Ultimi Articoli
Forum
Tutorial
Godfathers
Riferimenti
Workshop
Contatti
Login
Italiano
MACHINE LEARNING
ITALIANO, MACHINE LEARNING IT, MATH IT
METODI LINEARI PER LA RIDUZIONE DELLA DIMENSIONALITÀ: ANALISI DELLE COMPONENTI PRINCIPALI
MACHINE LEARNING IT
METODI PER LA RIDUZIONE DELLA DIMENSIONALITÀ BASATI SULLA VARIETÀ DIFFERENZIABILE: IL CASO ISOMAP
MACHINE LEARNING IT
STOCHASTIC NEIGHBOR EMBEDDING (SNE) E LA SUA CORREZIONE IN T-SNE
3 GENNAIO 2018
Metodi Lineari per la Riduzione della Dimensionalità: Analisi delle Componenti Principali
27 NOVEMBRE 2017
Metodi per la riduzione della dimensionalità basati sulla Varietà differenziabile: il caso ISOMAP
23 OTTOBRE 2017
Stochastic Neighbor Embedding (SNE) e la sua correzione in t-SNE
Cosa Posso Trovare
Il ML è un insieme di tecniche che permettono alla macchine di "imparare" dai dati e in seguito prendere decisioni o fare una predizione su di essi.

Community


Home
Conference
Ultimi Articoli
Forum
Tutorial
Godfathers
Riferimenti
Workshop
Contatti
Login
Italiano
Iscriviti
Stai vedendo 2 discussioni - dal 1 al 2 (di 2 totali)
Discussione
Partecipanti Articoli
reg hardware
Iniziato da: rensisam
2
2
Abbiamo aperto un nuovo Forum
Iniziato da: ValerioNeriWebMaster
2
3
Stai vedendo 2 discussioni - dal 1 al 2 (di 2 totali)
Crea una nuova discussione in “Deep Learning”
Il tuo account può inserire contenuto HTML senza restrizioni.
Titolo discussione (Lunghezza massima: 80):
b
i
link
b-quote
del
img
ul
ol
li
code
close tags
Cerca
DEEP LEARNING – IT
Brexit Bulletin: Lording It Over Theresa May
Brexit Bulletin: Lording It Over Theresa May BloombergMay faces embarrassing Brexit defeat in upper house ReutersIn the Lords' hands The HinduFull coverage
Tributes pour in for 'force of a woman' Barbara Bush
Tributes pour in for 'force of a woman' Barbara Bush CNNWhere Will Barbara Bush Be Buried? Former First Lady's Grave Will Be Near Daughter Robin NewsweekOn Family, Giving and Life in Politics: Here Are Some of Barbara Bush's Most Memorable Quotes TIMEFirst Lady Barbara Bush Dies at Age 92 GoodHousekeeping.comA lesson learned in Barbara Bush's bathing suit Washington PostFull coverage
Chinese President Xi Jinping will visit Pyongyang 'soon,' official says
Chinese President Xi Jinping will visit Pyongyang 'soon,' official says CNNCIA Director Pompeo Reportedly Made Secret Trip To North Korea NPRJapan Fiscal Year Trade Surplus With US up Nearly 6 Percent U.S. News & World ReportCIA Director Pompeo met with North Korean leader Kim Jong Un over Easter weekend Washington PostUnpacking a US Decision to Engage North Korea: What […]
Bolton dealing to build an Arab military force in Syria
Bolton dealing to build an Arab military force in Syria CNNSaudi Arabia brings back movie theaters — and 'staggering' demand is expected CNBCThe real reason Saudi Arabia is lifting its cinema ban QuartzWhat Saudi Arabia can learn from 'Black Panther' Washington PostFull coverage
Fallen Massachusetts police officer to be laid to rest
Fallen Massachusetts police officer to be laid to rest Fox NewsThousands gather on Cape Cod for wake of slain police officer The Boston GlobeThousands line streets to honor fallen Yarmouth officer Sean Gannon Boston News, Weather, Sports | WHDH 7NewsFull coverage
Castro's successor seen as unlikely to bring sweeping change to Cuba
Castro's successor seen as unlikely to bring sweeping change to Cuba ReutersA look at the younger generation of Cuban leaders Washington PostRaul Castro To Step Down As Cuba's President HuffPostWe Shouldn't Ignore Cuba New York TimesFull coverage
DISCUSSIONI RECENTI
reg hardware

Meetup & Conferences



Home Conference Ultimi Articoli Forum Tutorial Godfathers Riferimenti Workshop Contatti Login Italiano

MEETUPS

Rimani sempre aggiornato sulle iniziative di Deep Learning Italia e registrati ai nostri meetup che si svolgono una volta al mese a Roma , Milano e Pisa

I Nostri Meetups

29 MARZO 2018
Introduzione divulgativa alle Reti Neurali e al Deep Learning

29 MARZO 2018
Deep Learning & Alpha Go – Maurizio Parton

29 MARZO 2018
Capsule Networks – Daniele D'Armiento

29 MARZO 2018
Analysis of Deep Learning Models by Deep Echo State Networks – Luca Pedrelli

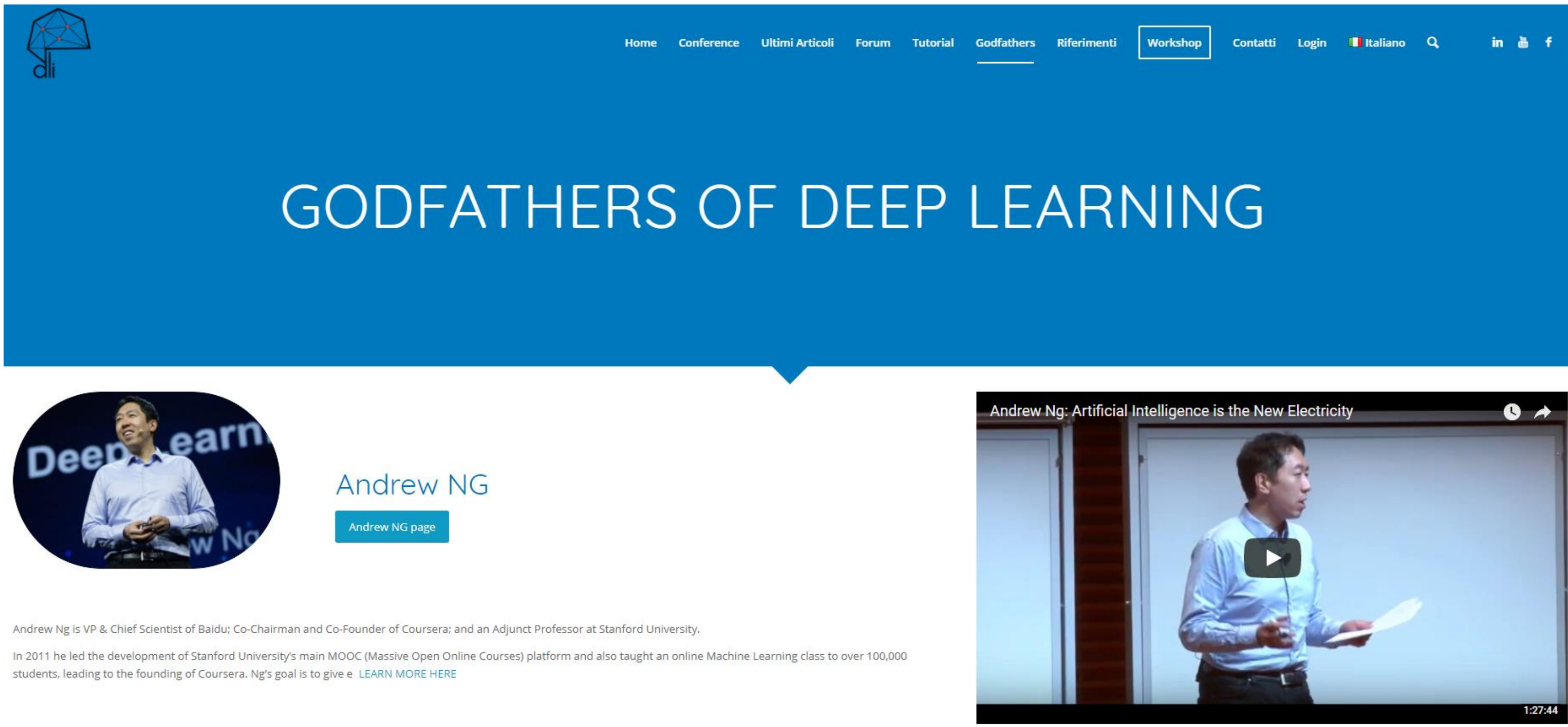
29 MARZO 2018
Deep Learning and the "Deep Learning Italia Project" – Francesco Pugliese

Conference around the world

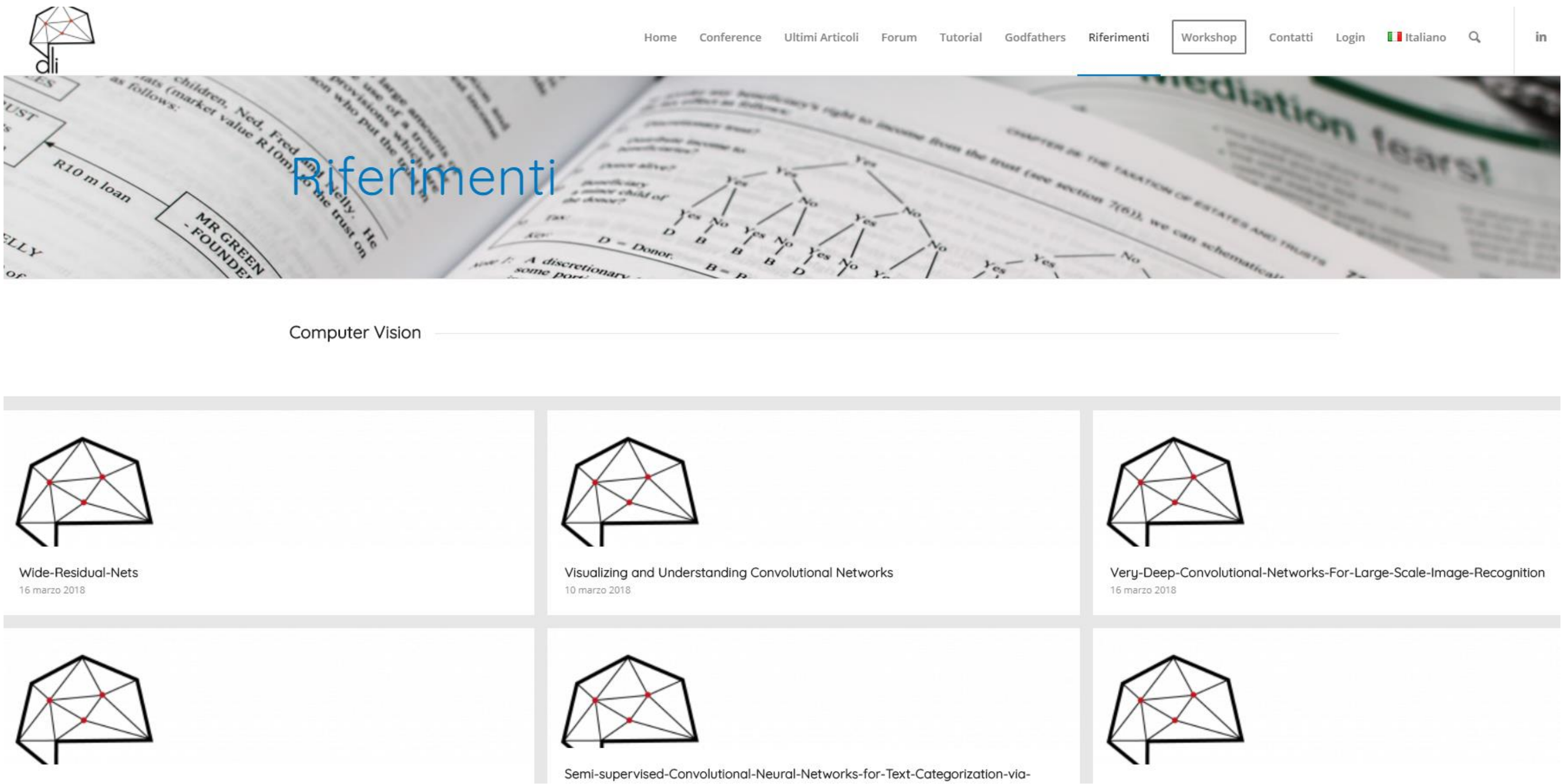
GitHub list of conference

Name	Location	Date Begin	Date End	Description
Shoptalk	Las Vegas, USA	18 marzo 2018	21 marzo 2018	Shoptalk covers the rapid evolution of how consumers discover, shop and buy—from new technologies and business models to the latest trends in consumer behaviors, preferences and expectations.
Gartner Data & Analytics Summit	London, UK	19 marzo 2018	21 marzo 2018	To survive and thrive in the digital era, now is the time to drive data and analytics into the core of your business and scale outward to every employee, customer, supplier and partner. This conference will help you create the future – a future based on data you can trust, analytics you can rely on and the insight needed to make game-changing business decisions.

Goodfathers

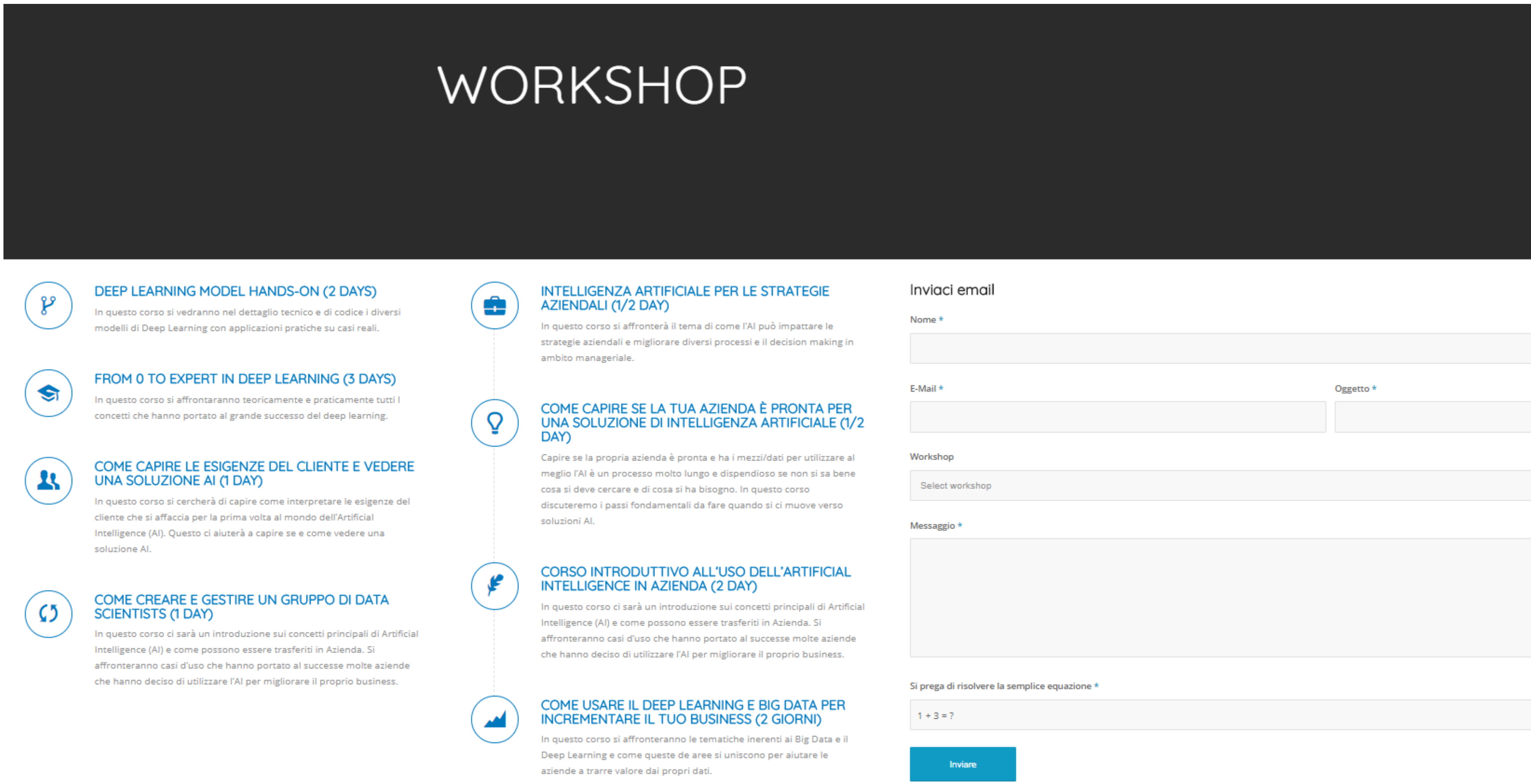


REFERENCES



dli
Home
Conference
Ultimi Articoli
Forum
Tutorial
Godfathers
Riferimenti
Workshop
Contatti
Login
Italiano
in
Riferimenti
Computer Vision
Wide-Residual-Nets
16 marzo 2018
Visualizing and Understanding Convolutional Networks
10 marzo 2018
Very-Deep-Convolutional-Networks-For-Large-Scale-Image-Recognition
16 marzo 2018
Semi-supervised-Convolutional-Neural-Networks-for-Text-Categorization-via-

Workshop



WORKSHOP

DEEP LEARNING MODEL HANDS-ON (2 DAYS)

In questo corso si vedranno nel dettaglio tecnico e di codice i diversi modelli di Deep Learning con applicazioni pratiche su casi reali.

FROM 0 TO EXPERT IN DEEP LEARNING (3 DAYS)

In questo corso si affronteranno teoricamente e praticamente tutti i concetti che hanno portato al grande successo del deep learning.

COME CAPIRE LE ESIGENZE DEL CLIENTE E VEDERE UNA SOLUZIONE AI (1 DAY)

In questo corso si cercherà di capire come interpretare le esigenze del cliente che si affaccia per la prima volta al mondo dell'Artificial Intelligence (AI). Questo ci aiuterà a capire se e come vedere una soluzione AI.

COME CREARE E GESTIRE UN GRUPPO DI DATA SCIENTISTS (1 DAY)

In questo corso ci sarà un introduzione sui concetti principali di Artificial Intelligence (AI) e come possono essere trasferiti in Azienda. Si affronteranno casi d'uso che hanno portato al successe molte aziende che hanno deciso di utilizzare l'AI per migliorare il proprio business.

INTELLIGENZA ARTIFICIALE PER LE STRATEGIE AZIENDALI (1/2 DAY)

In questo corso si affronterà il tema di come l'AI può impattare le strategie aziendali e migliorare diversi processi e il decision making in ambito manageriale.

COME CAPIRE SE LA TUA AZIENDA È PRONTA PER UNA SOLUZIONE DI INTELLIGENZA ARTIFICIALE (1/2 DAY)

Capire se la propria azienda è pronta e ha i mezzi/dati per utilizzare al meglio l'AI è un processo molto lungo e dispendioso se non si sa bene cosa si deve cercare e di cosa si ha bisogno. In questo corso discuteremo i passi fondamentali da fare quando si ci muove verso soluzioni AI.

CORSO INTRODUTTIVO ALL'USO DELL'ARTIFICIAL INTELLIGENCE IN AZIENDA (2 DAY)

In questo corso ci sarà un introduzione sui concetti principali di Artificial Intelligence (AI) e come possono essere trasferiti in Azienda. Si affronteranno casi d'uso che hanno portato al successe molte aziende che hanno deciso di utilizzare l'AI per migliorare il proprio business.

COME USARE IL DEEP LEARNING E BIG DATA PER INCREMENTARE IL TUO BUSINESS (2 GIORNI)

In questo corso si affronteranno le tematiche inerenti ai Big Data e il Deep Learning e come queste de aree si uniscono per aiutare le aziende a trarre valore dai propri dati.

Inviaci email

Nome *

E-Mail *

Oggetto *

Workshop

Select workshop

Messaggio *

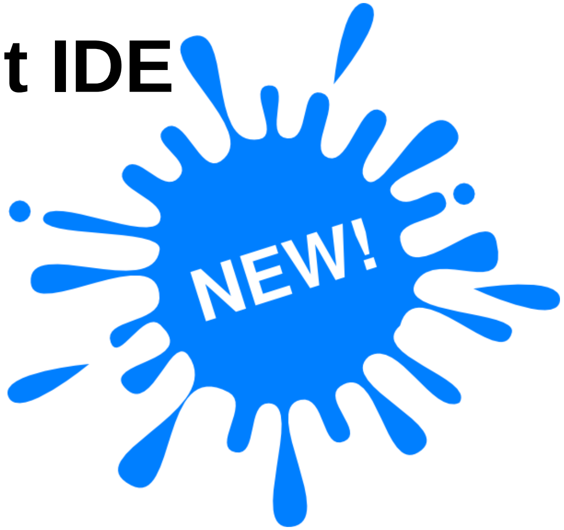
Si prega di risolvere la semplice equazione *

1 + 3 = ?

Inviare

New Features

- Deep Learning **Development IDE**
- **Repository** of Datasets



USEFUL Links

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

WebSite:

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

Use Case 1: Regression with Multilayer Perceptrons

- The problem that we will look at in this tutorial is the [Boston house price dataset].

Pre-processing

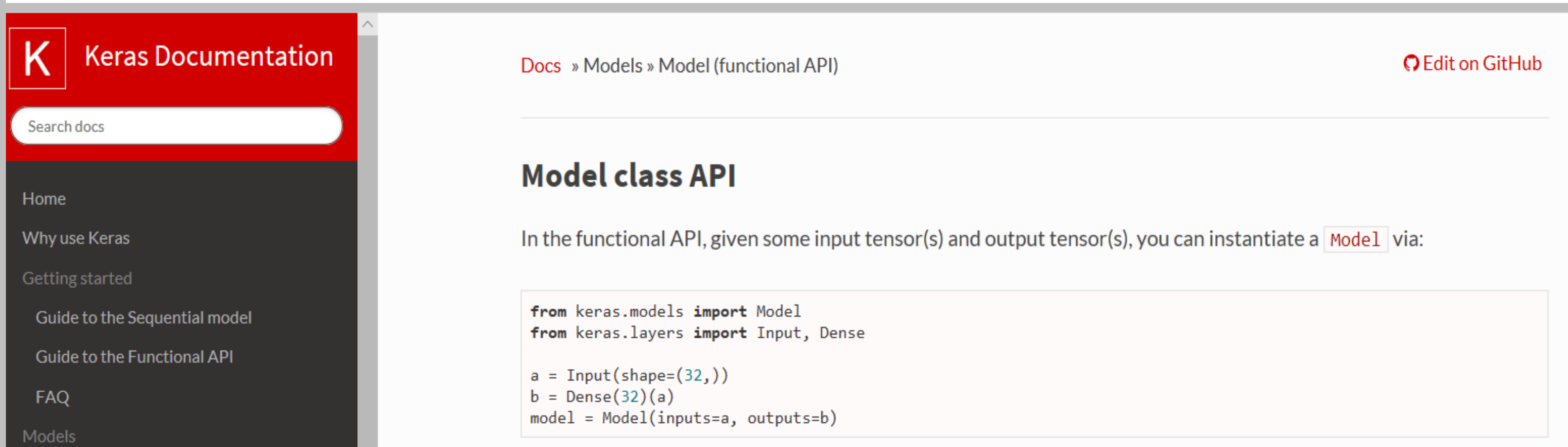
```
import numpy
import pandas
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasRegressor
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
```

```
# load dataset
dataframe = pandas.read_csv("housing.csv", delim_whitespace=True, header=None)
dataset = dataframe.values
# split into input (X) and output (Y) variables
X = dataset[:,0:13]
Y = dataset[:,13]
```

Use Case 1: Regression with Multilayer Perceptrons

Model Definition

```
# define base model
def baseline_model():
        # create model
        model = Sequential()
        model.add(Dense(13, input_dim=13, kernel_initializer='normal', activation='
        model.add(Dense(1, kernel_initializer='normal'))
        # Compile model
        model.compile(loss='mean_squared_error', optimizer='adam')
        return model
```



Use Case 1: Regression with Multilayer Perceptrons

Model Fitting

```
# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)
# evaluate model with standardized dataset
estimator = KerasRegressor(build_fn=baseline_model, epochs=100, batch_size=5, verbo
```

Wrappers for the Scikit-Learn API

You can use `Sequential` Keras models (single-input only) as part of your Scikit-Learn workflow via the wrappers found at `keras.wrappers.scikit_learn.py`.

There are two wrappers available:

`keras.wrappers.scikit_learn.KerasClassifier(build_fn=None, **sk_params)`, which implements the Scikit-Learn classifier interface,

`keras.wrappers.scikit_learn.KerasRegressor(build_fn=None, **sk_params)`, which implements the Scikit-Learn regressor interface.

Use Case 1: Regression with Multilayer Perceptrons

Model Validation

```
kfold = KFold(n_splits=10, random_state=seed)
results = cross_val_score(estimator, X, Y, cv=kfold)
print("Results: %.2f (%.2f) MSE" % (results.mean(), results.std()))
```

```
Baseline: 31.64 (26.82) MSE
```

Model Standardization

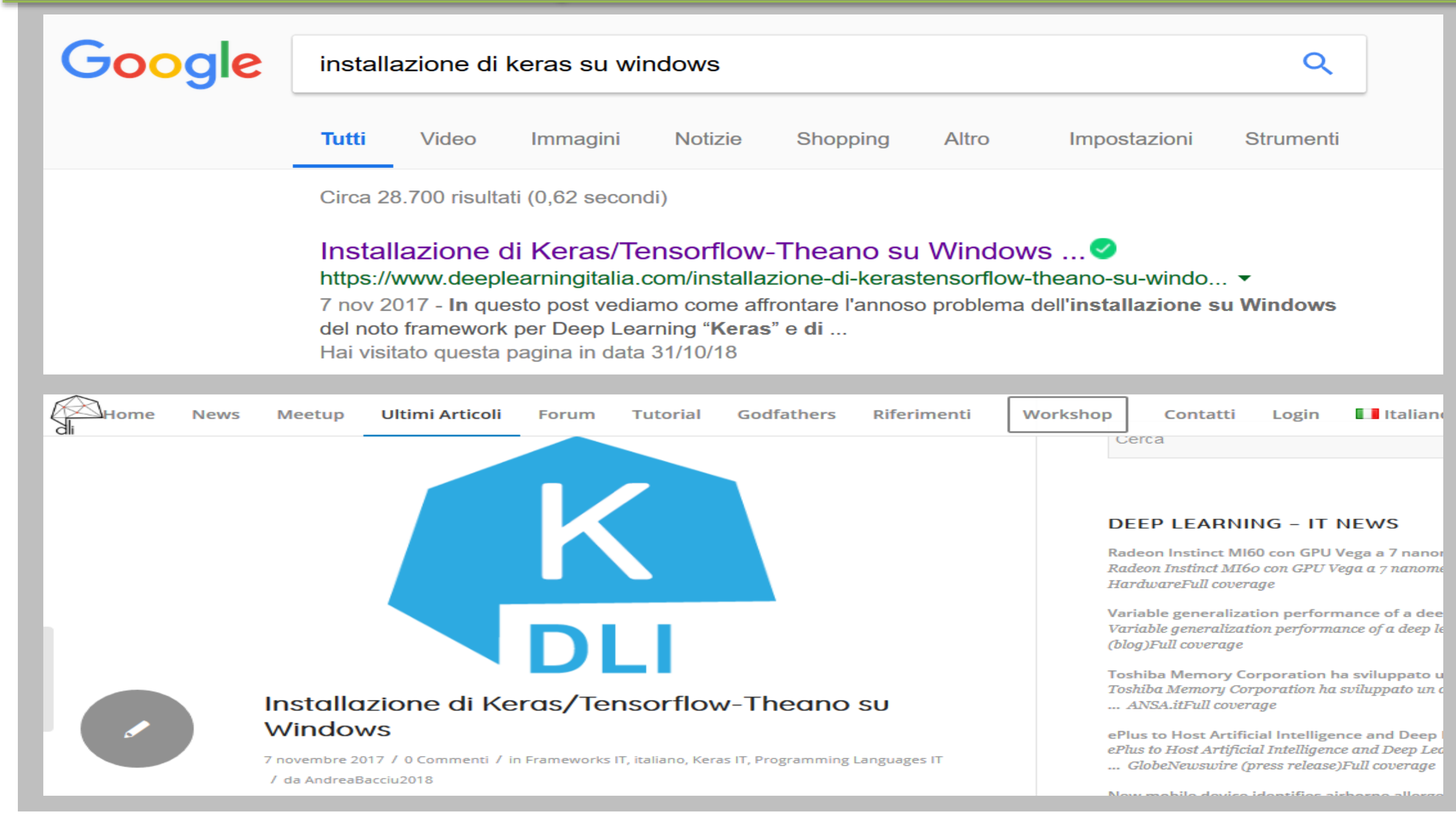
```
# evaluate model with standardized dataset
numpy.random.seed(seed)
estimators = []
estimators.append(('standardize', StandardScaler()))
estimators.append(('mlp', KerasRegressor(build_fn=baseline_model, epochs=50, batch_
pipeline = Pipeline(estimators)
kfold = KFold(n_splits=10, random_state=seed)
results = cross_val_score(pipeline, X, Y, cv=kfold)
print("Standardized: %.2f (%.2f) MSE" % (results.mean(), results.std()))
```

Use Case 1: Regression with Multilayer Perceptrons

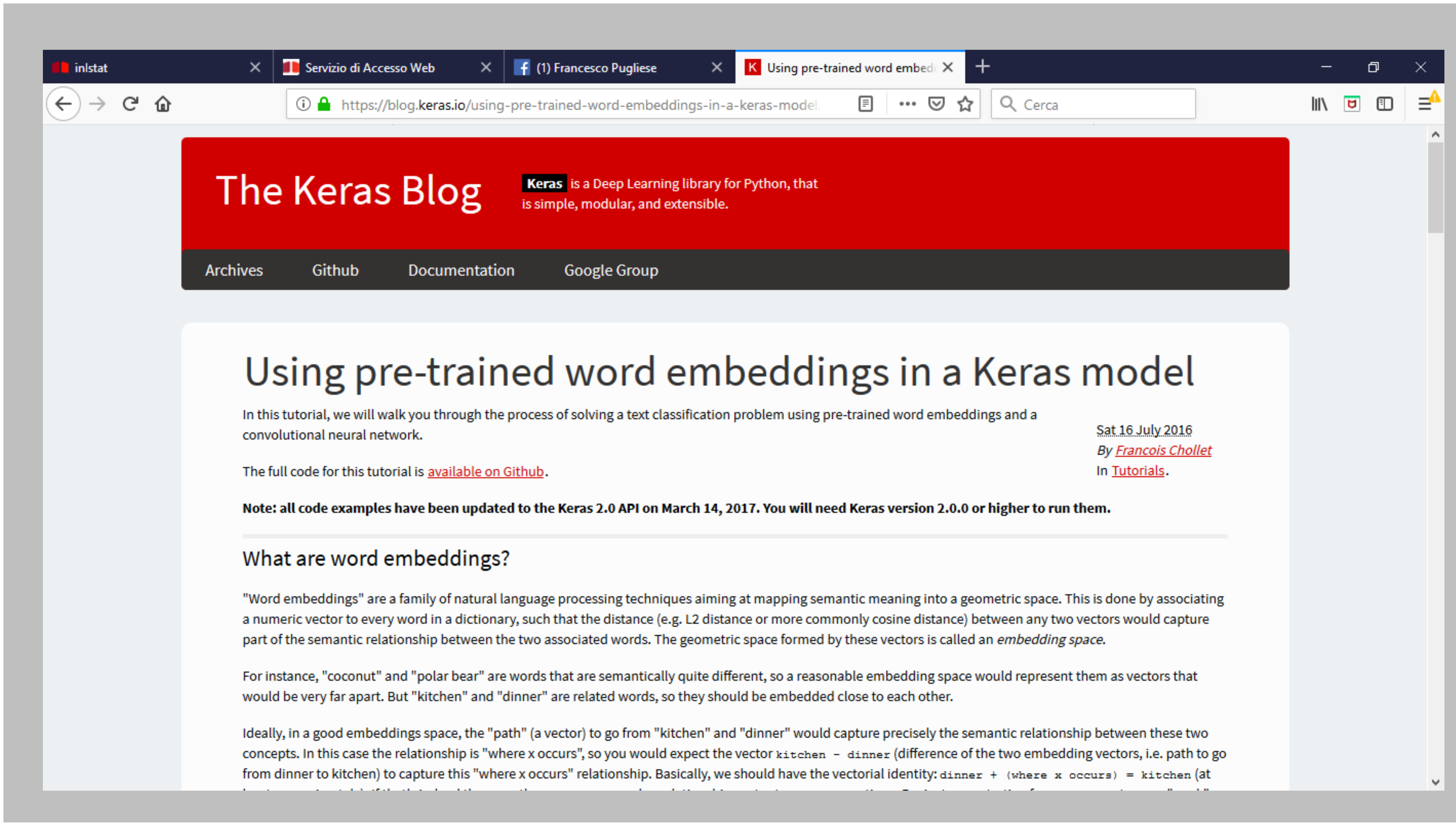
Wider Multilayer Neural Network

```
# define wider model
def wider_model():
        # create model
        model = Sequential()
        model.add(Dense(20, input_dim=13, kernel_initializer='normal', activation='
        model.add(Dense(1, kernel_initializer='normal'))
        # Compile model
        model.compile(loss='mean_squared_error', optimizer='adam')
        return model
```

Exercise 1: Text Classifier in Python/Keras



Exercise 1: Text Classifier in Python/Keras



Exercise 1: Text Classifier in Python/Keras

inIstat | Servizio di Accesso Web | (1) Francesco Pugliese | Using pre-trained word embed

https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model

Cerca

Preparing the text data

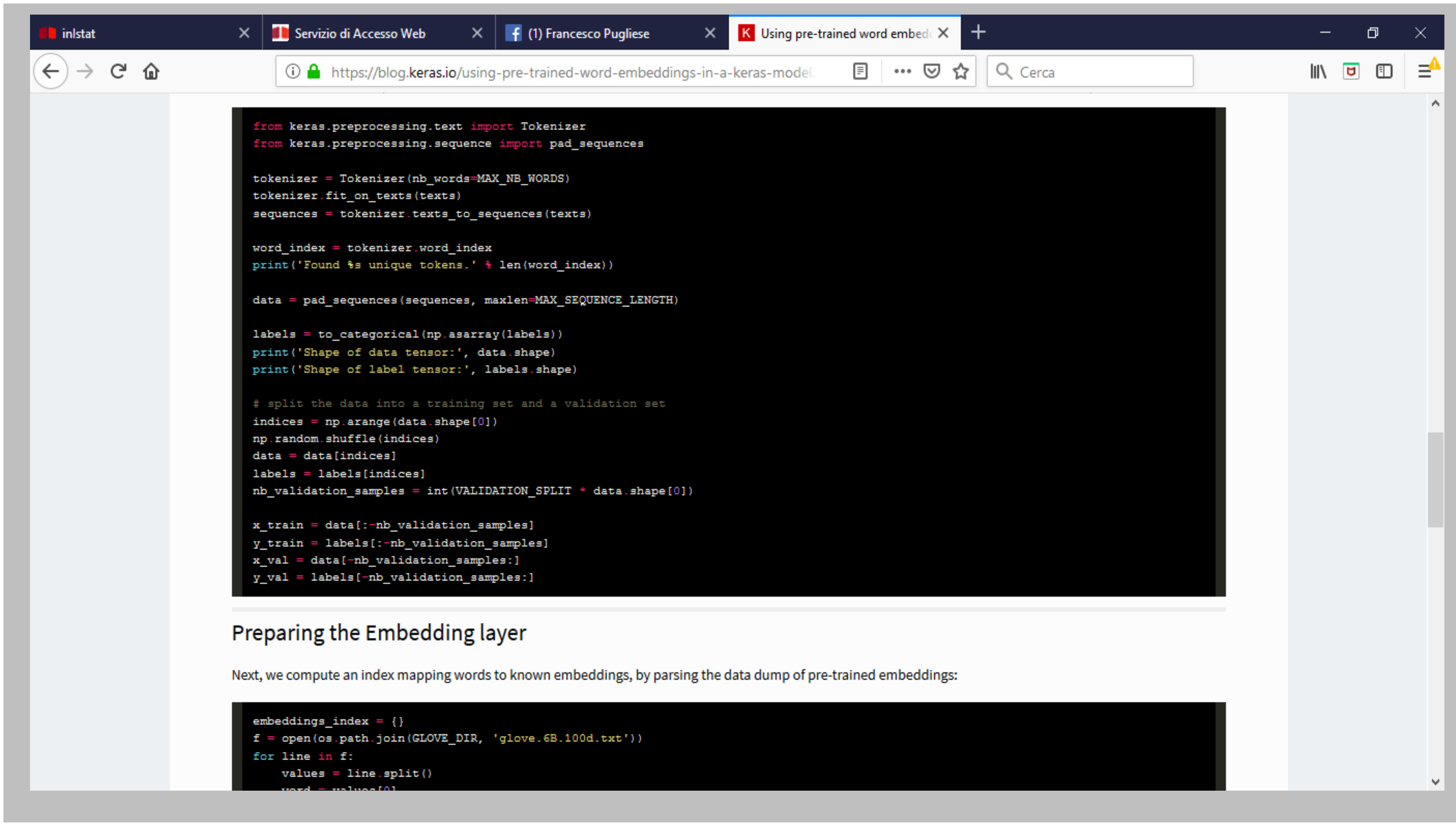
First, we will simply iterate over the folders in which our text samples are stored, and format them into a list of samples. We will also prepare at the same time a list of class indices matching the samples:

```
texts = []  # list of text samples
labels_index = {}  # dictionary mapping label name to numeric id
labels = []  # list of label ids
for name in sorted(os.listdir(TEXT_DATA_DIR)):
    path = os.path.join(TEXT_DATA_DIR, name)
    if os.path.isdir(path):
        label_id = len(labels_index)
        labels_index[name] = label_id
        for fname in sorted(os.listdir(path)):
            if fname.isdigit():
                fpath = os.path.join(path, fname)
                if sys.version_info < (3,):
                    f = open(fpath)
                else:
                    f = open(fpath, encoding='latin-1')
                t = f.read()
                i = t.find('\n\n')  # skip header
                if 0 < i:
                    t = t[i:]
                texts.append(t)
                f.close()
                labels.append(label_id)

print('Found %s texts.' % len(texts))
```

Then we can format our text samples and labels into tensors that can be fed into a neural network. To do this, we will rely on Keras utilities `keras.preprocessing.text.Tokenizer` and `keras.preprocessing.sequence.pad_sequences`.

```
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
```

Exercise 1: Text Classifier in Python/Keras




```python
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences

tokenizer = Tokenizer(nb_words=MAX_NB_WORDS)
tokenizer.fit_on_texts(texts)
sequences = tokenizer.texts_to_sequences(texts)

word_index = tokenizer.word_index
print('Found %s unique tokens.' % len(word_index))

data = pad_sequences(sequences, maxlen=MAX_SEQUENCE_LENGTH)

labels = to_categorical(np.asarray(labels))
print('Shape of data tensor:', data.shape)
print('Shape of label tensor:', labels.shape)

# split the data into a training set and a validation set
indices = np.arange(data.shape[0])
np.random.shuffle(indices)
data = data[indices]
labels = labels[indices]
nb_validation_samples = int(VALIDATION_SPLIT * data.shape[0])

x_train = data[:-nb_validation_samples]
y_train = labels[:-nb_validation_samples]
x_val = data[-nb_validation_samples:]
y_val = labels[-nb_validation_samples:]
```

Preparing the Embedding layer

Next, we compute an index mapping words to known embeddings, by parsing the data dump of pre-trained embeddings:

```python
embeddings_index = {}
f = open(os.path.join(GLOVE_DIR, 'glove.6B.100d.txt'))
for line in f:
    values = line.split()
```

Exercise 1: Text Classifier in Python/Keras

https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model

Preparing the Embedding layer

Next, we compute an index mapping words to known embeddings, by parsing the data dump of pre-trained embeddings:

```
embeddings_index = {}
f = open(os.path.join(GLOVE_DIR, 'glove.6B.100d.txt'))
for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index[word] = coefs
f.close()

print('Found %s word vectors.' % len(embeddings_index))
```

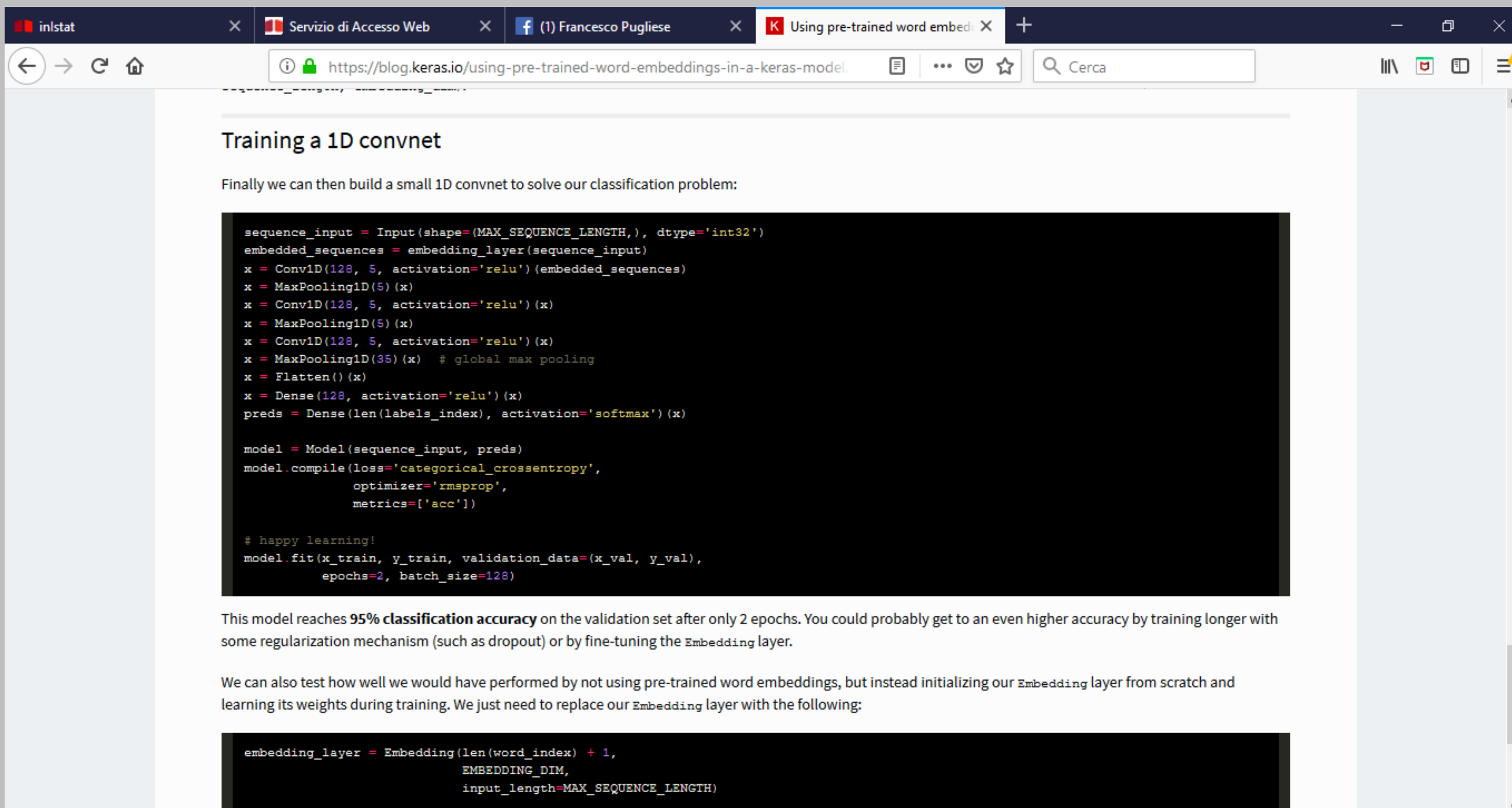
At this point we can leverage our `embedding_index` dictionary and our `word_index` to compute our embedding matrix:

```
embedding_matrix = np.zeros((len(word_index) + 1, EMBEDDING_DIM))
for word, i in word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        # words not found in embedding index will be all-zeros.
        embedding_matrix[i] = embedding_vector
```

We load this embedding matrix into an `Embedding` layer. Note that we set `trainable=False` to prevent the weights from being updated during training.

```
from keras.layers import Embedding

embedding_layer = Embedding(len(word_index) + 1,
                            EMBEDDING_DIM,
                            weights=[embedding_matrix],
                            input_length=MAX_SEQUENCE_LENGTH,
                            trainable=False)
```

Exercise 1: Text Classifier in Python/Keras



Training a 1D convnet

Finally we can then build a small 1D convnet to solve our classification problem:

```
sequence_input = Input(shape=(MAX_SEQUENCE_LENGTH,), dtype='int32')
embedded_sequences = embedding_layer(sequence_input)
x = Conv1D(128, 5, activation='relu')(embedded_sequences)
x = MaxPooling1D(5)(x)
x = Conv1D(128, 5, activation='relu')(x)
x = MaxPooling1D(5)(x)
x = Conv1D(128, 5, activation='relu')(x)
x = MaxPooling1D(35)(x)  # global max pooling
x = Flatten()(x)
x = Dense(128, activation='relu')(x)
preds = Dense(len(labels_index), activation='softmax')(x)

model = Model(sequence_input, preds)
model.compile(loss='categorical_crossentropy',
              optimizer='rmsprop',
              metrics=['acc'])

# happy learning!
model.fit(x_train, y_train, validation_data=(x_val, y_val),
          epochs=2, batch_size=128)
```

This model reaches **95% classification accuracy** on the validation set after only 2 epochs. You could probably get to an even higher accuracy by training longer with some regularization mechanism (such as dropout) or by fine-tuning the `Embedding` layer.

We can also test how well we would have performed by not using pre-trained word embeddings, but instead initializing our `Embedding` layer from scratch and learning its weights during training. We just need to replace our `Embedding` layer with the following:

```
embedding_layer = Embedding(len(word_index) + 1,
                            EMBEDDING_DIM,
                            input_length=MAX_SEQUENCE_LENGTH)
```

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).

.

AKNOWLEDGEMENTS

THANK YOU

FOR YOUR ATTENTION

Francesco Pugliese