ADVANCED SCHOOL AI

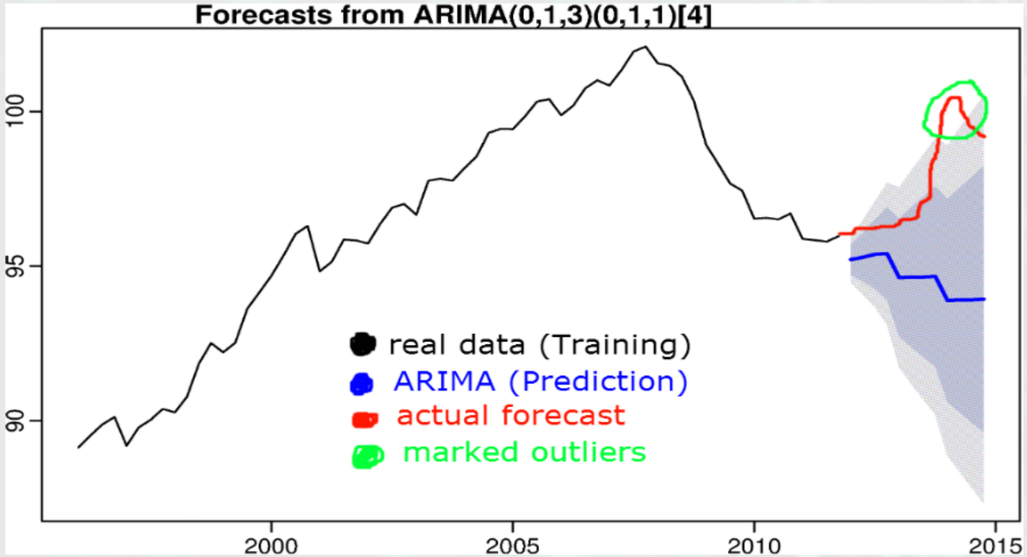
DEEP LEARNING LESSONS

Deep Learning for Natural Language Processing (NLP)

Francesco Pugliese, PhD
Data Scientist at ISTAT
francesco.pugliese@istat.it

Deep Learning for Natural Language Processing (NLP)

Deep Learning for Time-Series Prediction



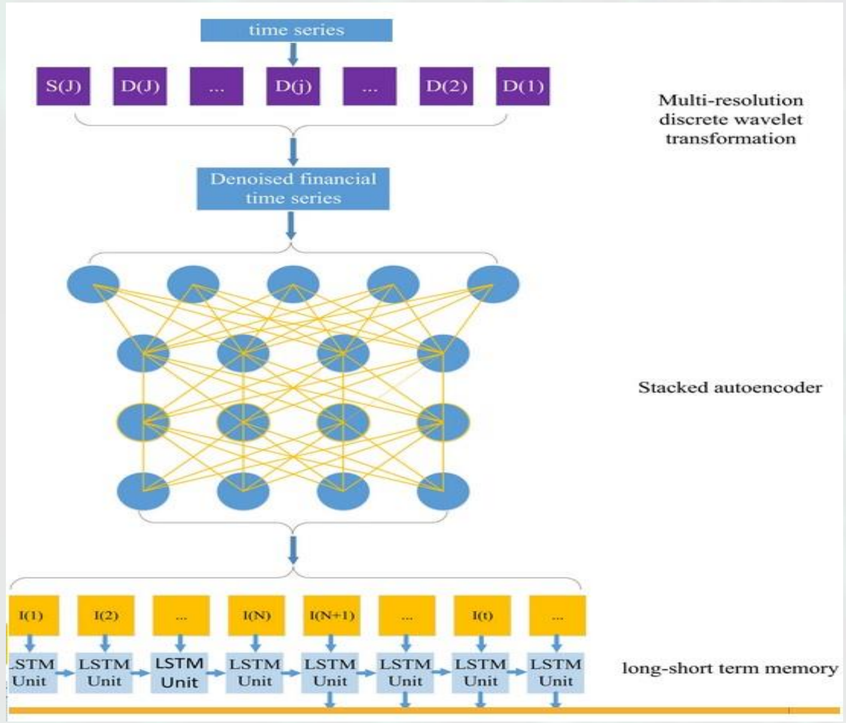
- The application of **Deep Learning** approaches to time-series prediction has received a great deal of attention from both entrepreneurs and researchers. Results show that deep learning models outperform other statistical models in predictive accuracy **(Bao, et al., 2017)**.

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

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

- 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.

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

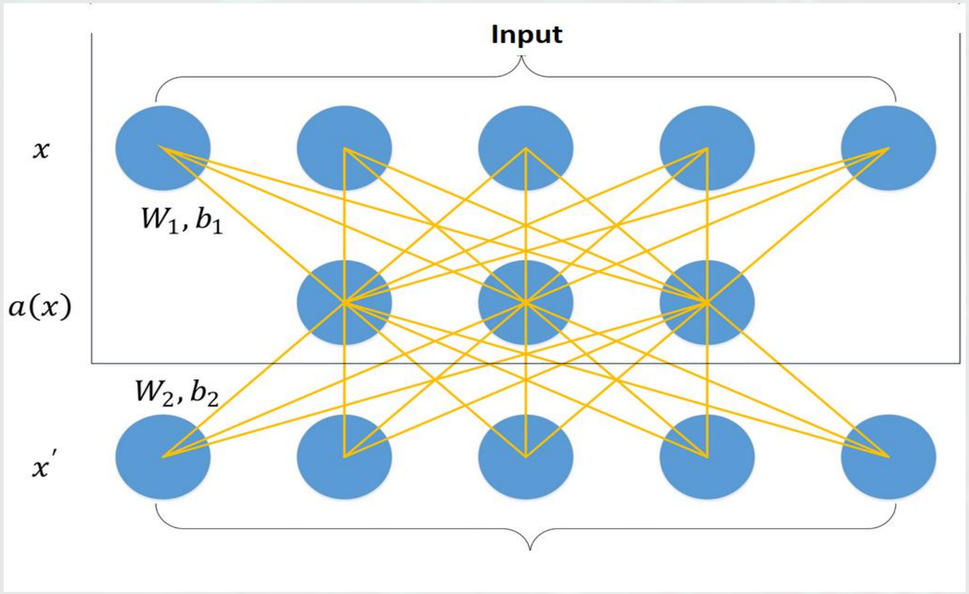


- However, by using ANNs, a priori analysis as ANNs do not require prior knowledge of the time series structure because of their black-box properties **(Nourani, et al., 2009).**

- 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.

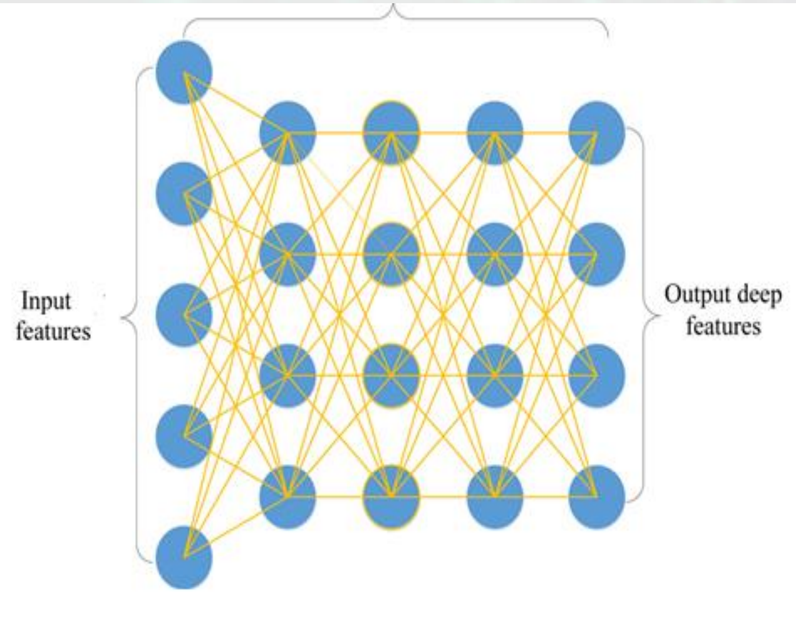
Stacked Auto-encoders (SAEs)



- According to recent studies, better approximation to nonlinear functions can be generated by stacked deep learning models than those models with a more shallow structure.
- A Single layer **Auto-Encoder (AE)** is a three-layer neural network. The first layer and the third layer are the input layer and the reconstruction layer with k units, respectively. The second layer is the hidden layer with n units, which is designed to generate the deep feature for this single layer AE.
- The aim of training the single layer AEE is to minimize the error between thee input vector and the reconstruction vector by **gradient descent.**

$$a(x) = f(\mathbf{W}_1 x + b_1)$$
$$x' = f(\mathbf{W}_2 a(x) + b_2)$$

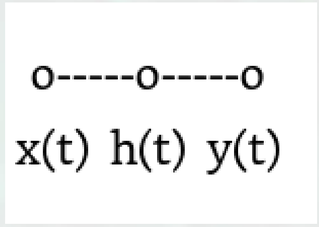
Stacked Auto-encoders (SAEs): 4 Auto-encoders



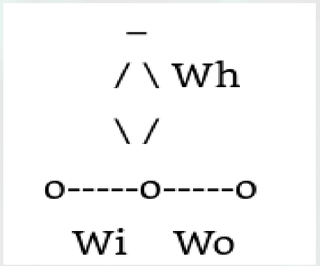
- Stacked auto-encoders (SAEs) are constructed by stacking a sequence of single-layer AEs layer by layer **(Bengio Y, et. Al. 2007).**
- After training the first single-layer auto-encoder, the reconstruction layer of the first single layer auto-encoder is removed (included weights and biases), and the **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 particulary non-stationary **(Ramsey, (1999) .**

Recurrent Neural Networks (RNNs) : Elman's Architecture

Simple Feed Forward Artificial Neural Network (MLP)



Recurrent Neural Network (Elman's Architecture)



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

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

- **RMSE (Root Mean Square Error):** Represents the sample standard deviation of the differences between predicted values and observed values.

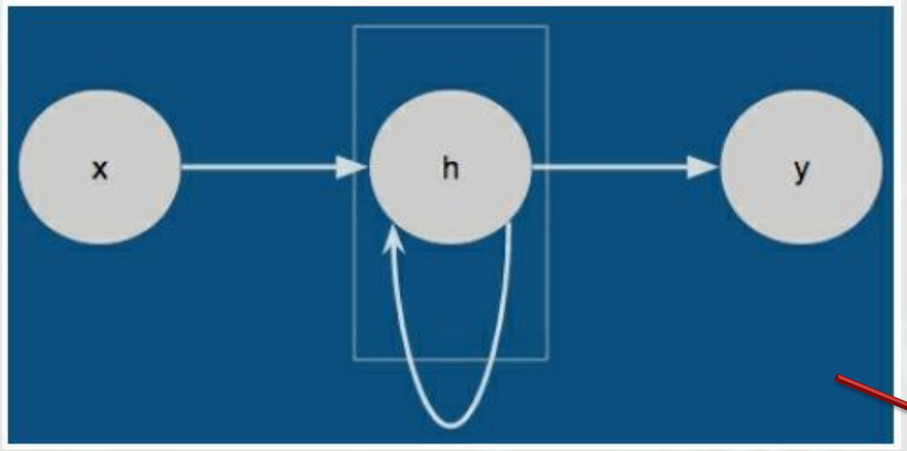
- **MAPE (Mean Absolute Percentage Error):** Measures the size of the error in percentage terms. Most people are comfortable thinking in percentage terms, making the MAPE easy to interpret.

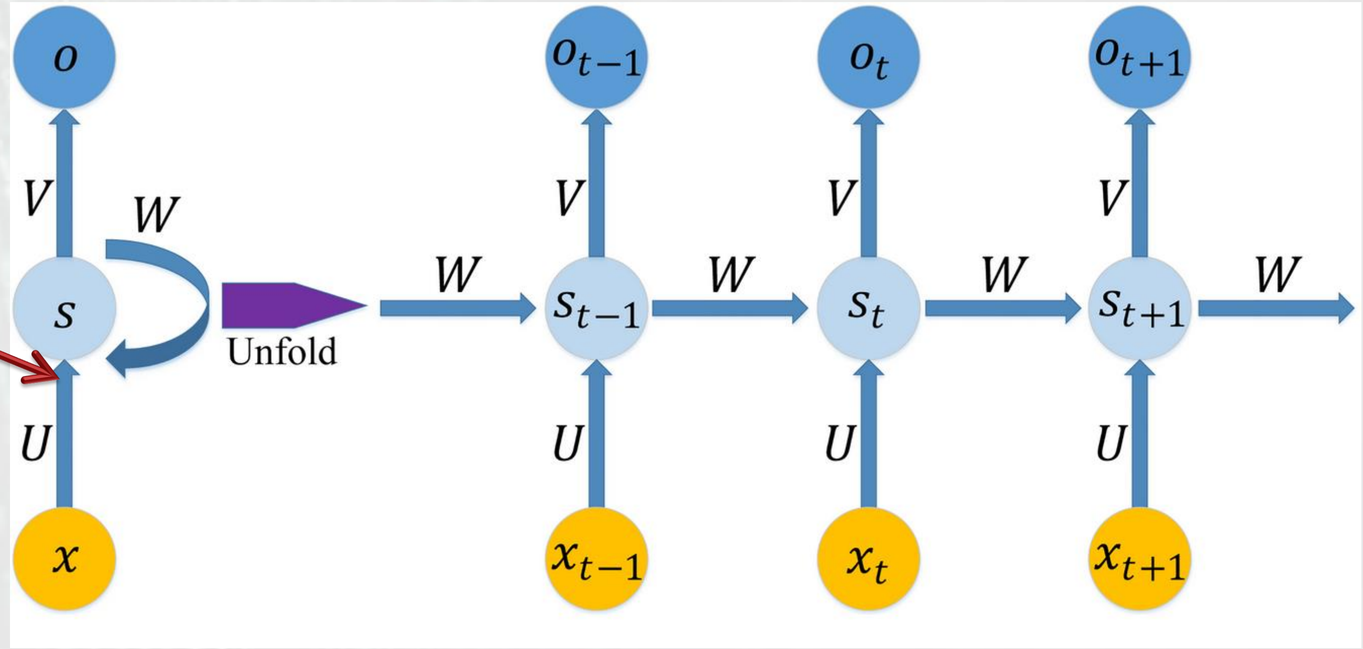
- Thanks to its recursive formulation, RNNs are not limited by the **Markov assumption** for sequence modeling:

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

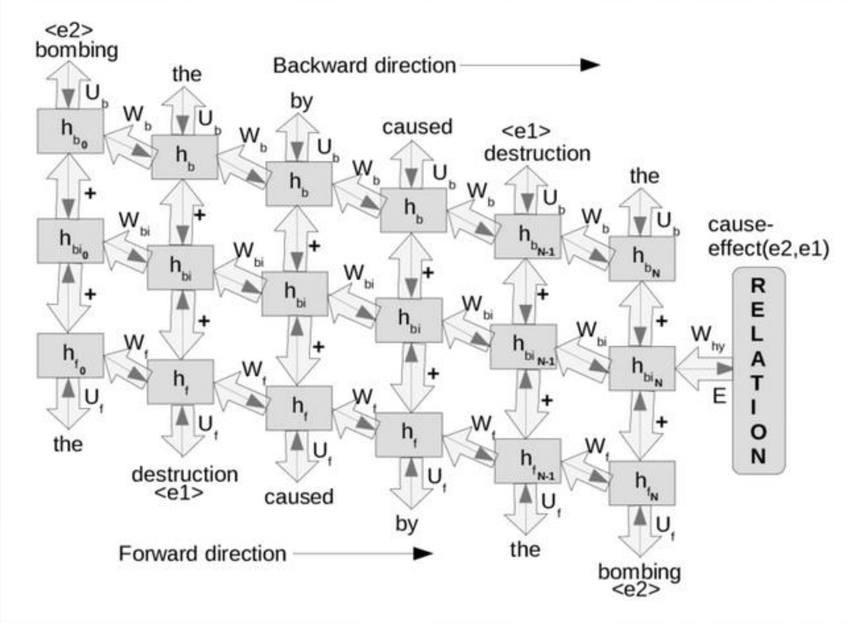
Unfolding RNNs

- Although RNN models the time series well, it is hard to learn long-term dependencies because of the vanishing gradient problem in **Back-Propagation Through Time (BPTT) (Palangi H, et al., 2016).**





Back Propagation Through Time (BPTT)

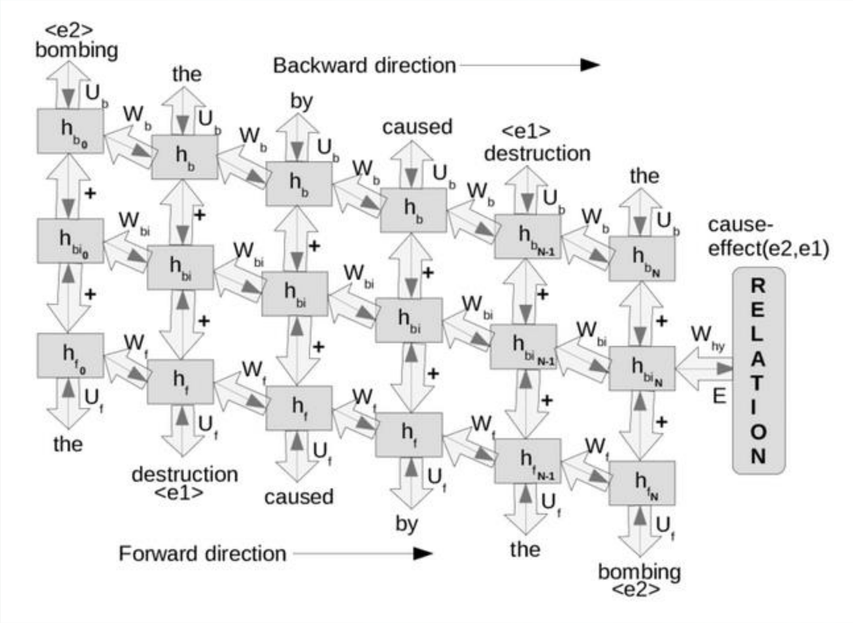


- In BPTT updating weights is going to look exactly the same.
- We can prove that the derivative of the loss function “Cross-Entropy” passes through the derivative of and the Softmax.
- Things are going to be multiplied together over and over again, due to the chain rule of calculus:

$$d[W_h{}^T h(t-1)] / dW_h$$

- The result is that gradients go down through the time (vanishing gradient problem) or they get very large very quickly (exploding gradient problem)
- RRNNs, GRUs, LSTMs solve the gradient problems with BPTT

Back Propagation Through Time (BPTT)



$$y(t)=\text{softmax}(W_o^T h(t))$$

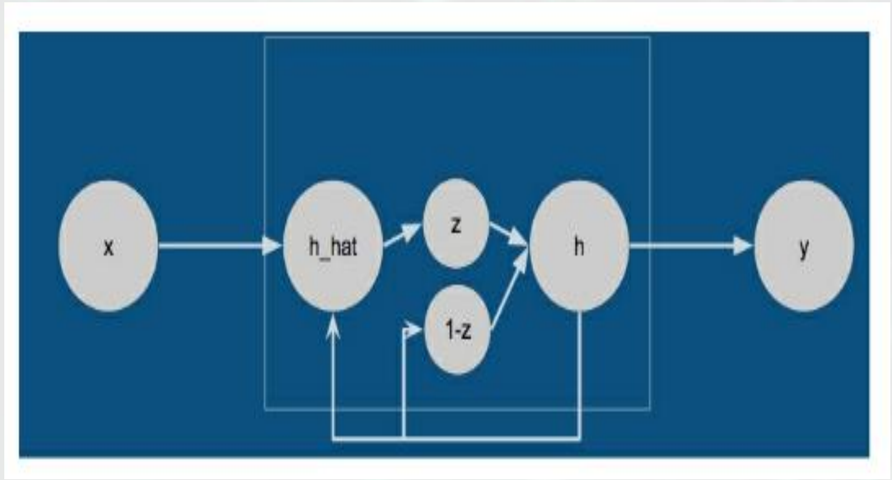
$$y(t)=\text{softmax}(W_o^T f(W_h^T h(t-1)+W_x^T x(t)))$$

$$y(t)=\text{softmax}(W_o^T f(W_h^T f(W_h^T h(t-2)+W_x^T x(t-1))+W_x^T x(t)))$$

$$y(t)=\text{softmax}(W_o^T f(W_h^T f(W_h^T f(W_h^T h(t-3)+W_x^T x(t-2))+W_x^T x(t-1))+W_x^T x(t)))$$

- We drop the bias in ordcer to display things simply

Rated Recurrent Neural Networks (RRNs)



- The idea is to weight f(x, h(t-1)), which is the output of a simple RNN and h(t-1) which is the previous state **(Amari, et al., 1995).**
- We add a rating operation between what would have been the output of a simple RNN and the previous output value.
- This new operation can be seen as a gate since it takes a value between 0 an 1, and the other gate has to take 1 minus that value
- This is a gate that is choosing between 2 things: a) taking on the old value or taking the new value. As result we get a mixture of both.

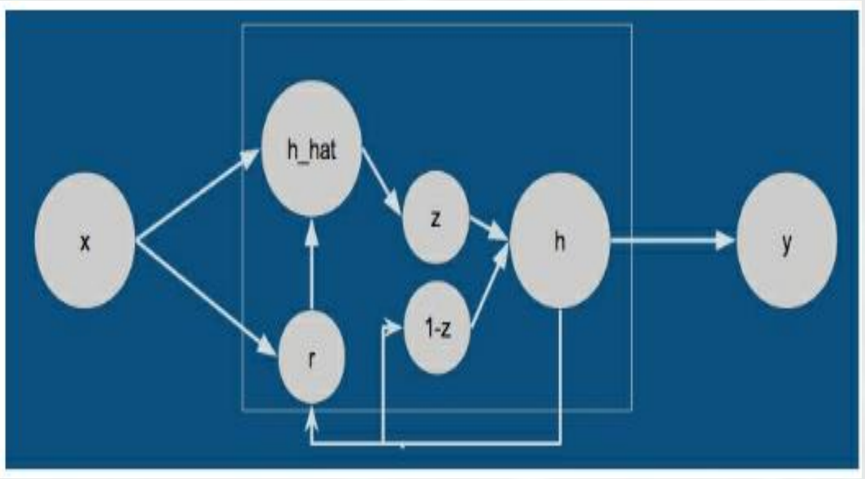
$$h\_hat(t) = f(x(t)W_x + h(t-1)W_h + b_h)$$

$$z(t) = sigmoid(x(t)W_{xz} + h(t-1)W_{hz} + b_z)$$

$$h(t) = (1 - z(t)) * h(t-1) + z(t) * h\_hat(t)$$

- **Z(t) is called the “rate”**

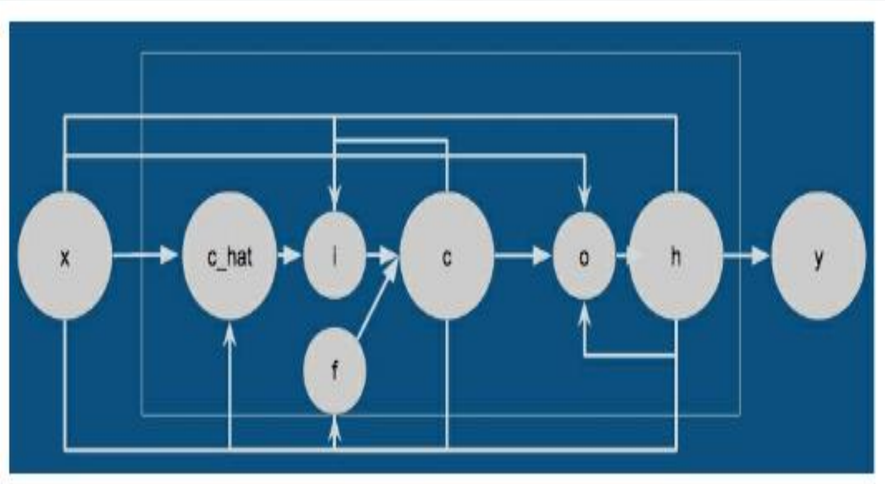
Gated Recurrent Neural Networks (GRUs)



- Gated Reccurrent Units were introduced in 2014 and are a simpler versione of LSTM. They have less parameters but same concepts **(Chung, et al., 2014).**
- Recent research has also show that the accuracy between **LSTM** and **GRU** is comparable and even better with the GRUs in some cases.
- In **GRUs** we add one more gate with regard to RRNNs: the “reset gate r(t)” controlling how much of the previous hidden we will consider when we create a new candidate hidden value. In other words, it can “reset” the hidden value.
- The old gate of RRNNs is now called “update gate z(t)” balancing previous hidden values and new candidate hidden value for the new hidden value.

$$r_t = \sigma(x_t W_{xr} + h_{t-1} W_{hr} + b_r)$$
$$z_t = \sigma(x_t W_{xz} + h_{t-1} W_{hz} + b_z)$$
$$\hat{h}_t = g(x_t W_{xh} + (r_t \odot h_{t-1}) W_{hh} + b_h)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t.$$

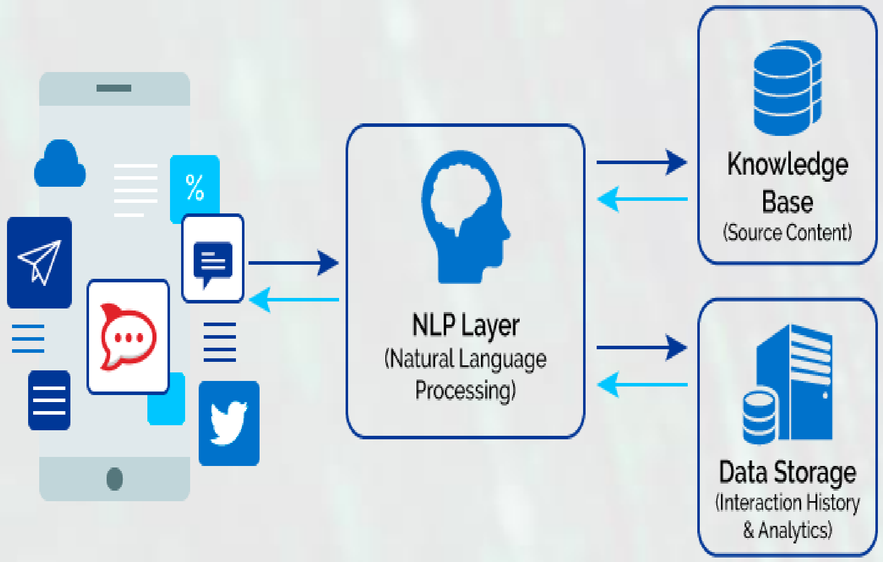
Long-Short Term Memories (LSTMs)



- **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.

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

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).

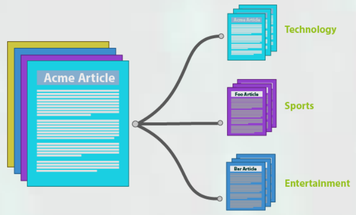
- **NLP** is everywhere because people communicate most everything: web search, advertisement, emails, customer service, etc.

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)

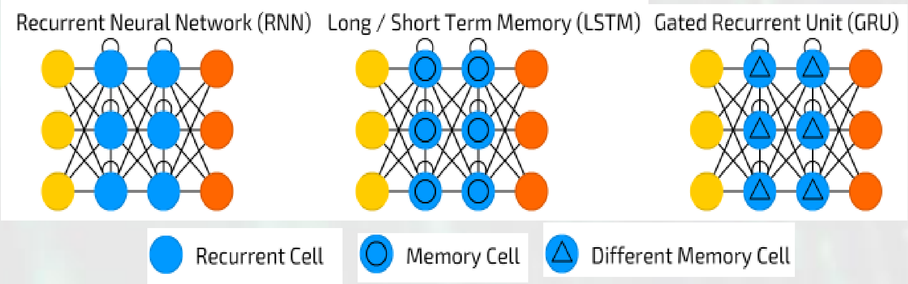
- **Natural language Processing** is shifting from statistical methods to **Neural Networks**.

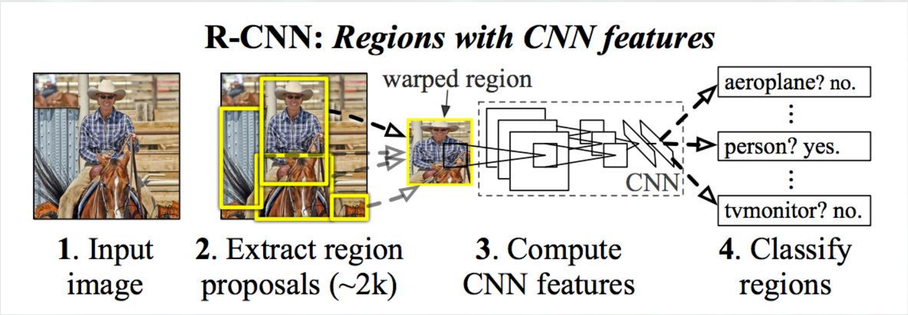
7 NLP applications where Deep Learning achieved «state-of-art» performance



- **1 Text Classification:** Classifying the topic or theme of a document (i.e. Sentiment Analysis).
- **2 Language Modeling:** Predict the next word given the previous words. It is fundamental for other tasks.
- **3 Speech Recognition:** Mapping an acoustic signal containing a spoken natural language utterance into the corresponding sequence of words intended by the speaker.
- **4 Caption Generation:** Given a digital image, such as a photo, generate a textual description of the contents of the image.
- **5 Machine Translation:** Automatic translation of text or speech from one language to another, is one [of] the most important applications of NLP.
- **6 Document Summarization:** It is the task where a short description of a text document is created.
- **7 Question Answering:** 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

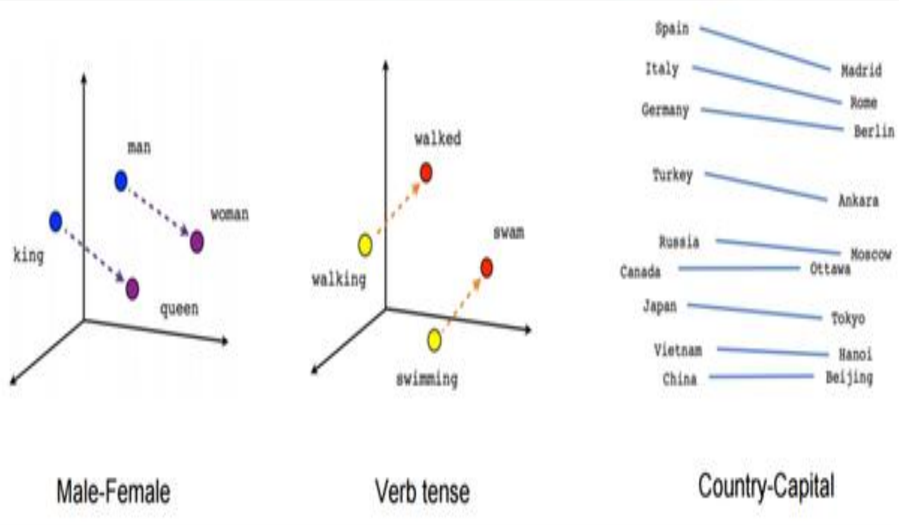




- **RNN, LSTM, GRU, ConvLstm, RecursiveNN, RNTN, RCNN**
- The modus operandi for text classification involves the use of a pre-trained **word embedding** for **representing words** and a **deep neural networks** for learning how to **discriminate documents** on classification problems.

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

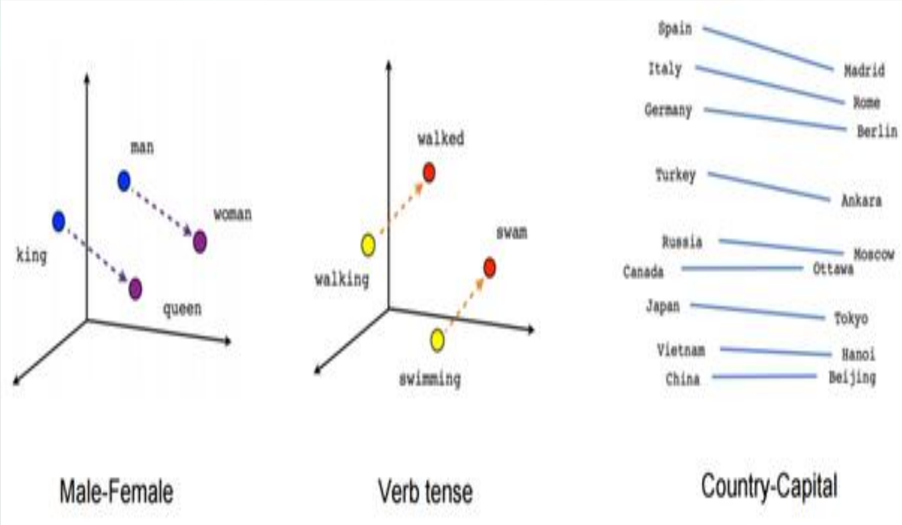
Word Embedding & Language Modeling

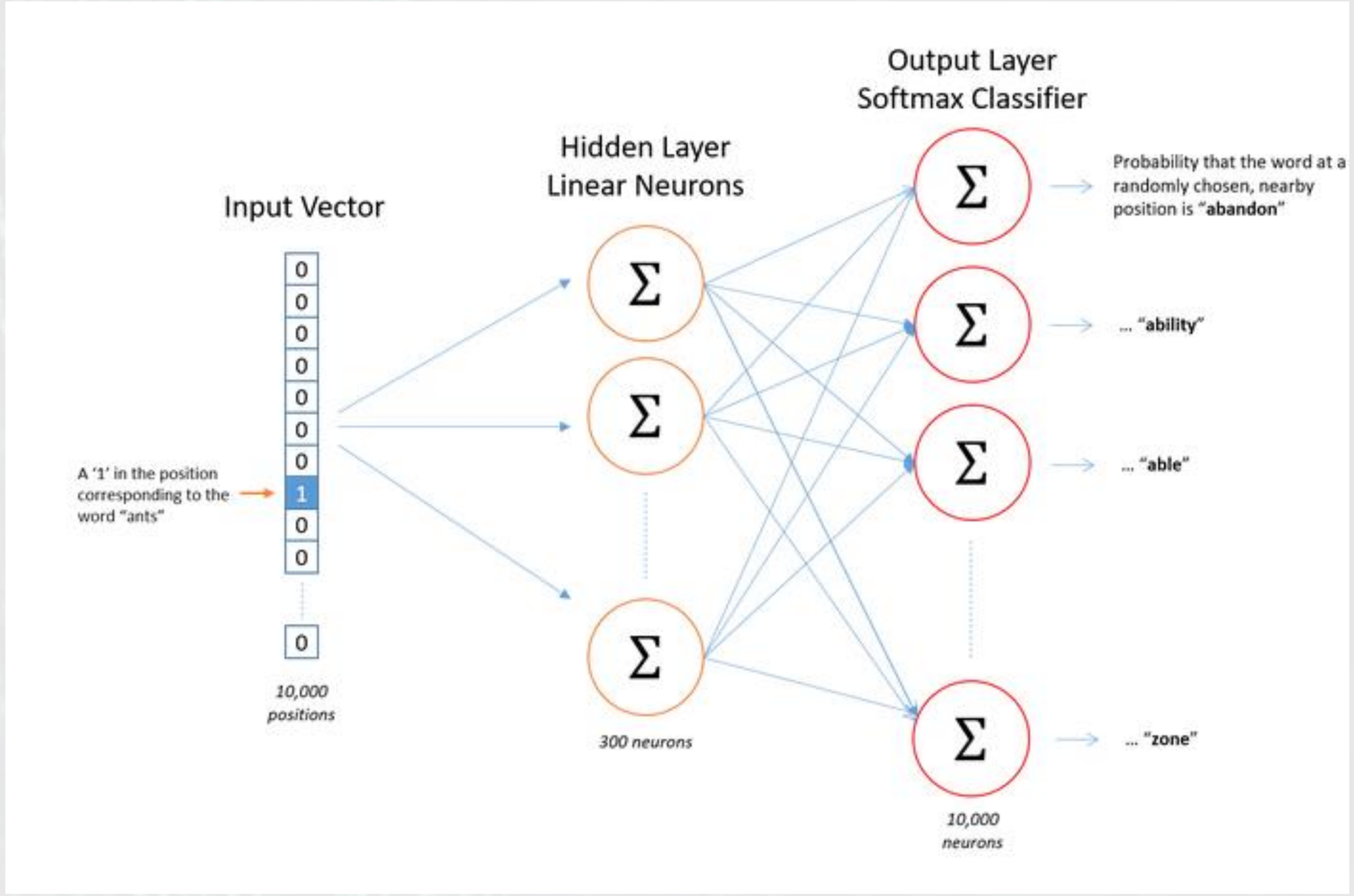


- **Word embedding** is the collective name for a set of language modeling and feature learning techniques for natural language processing (**NLP**) where words or sentences from the vocabulary are mapped to vectors of real numbers.

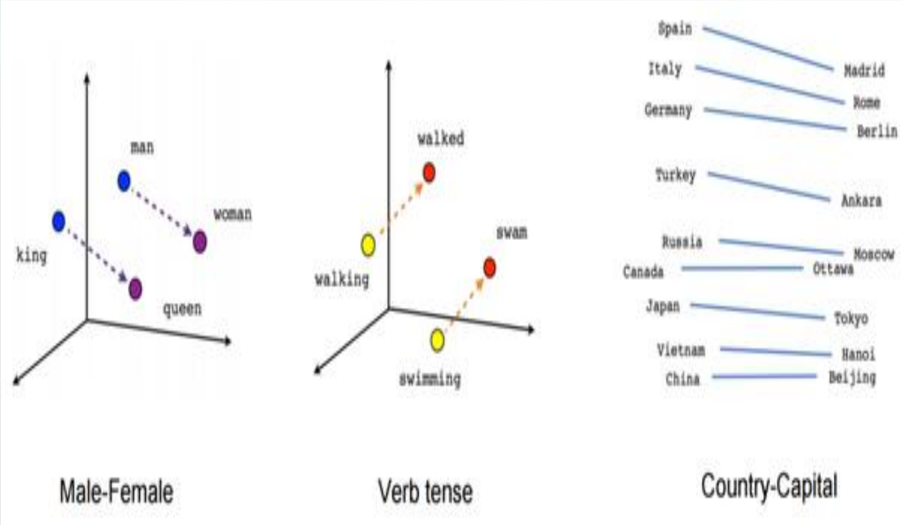
- These vectors are semantically correlated by metrics like **cosine distance.**

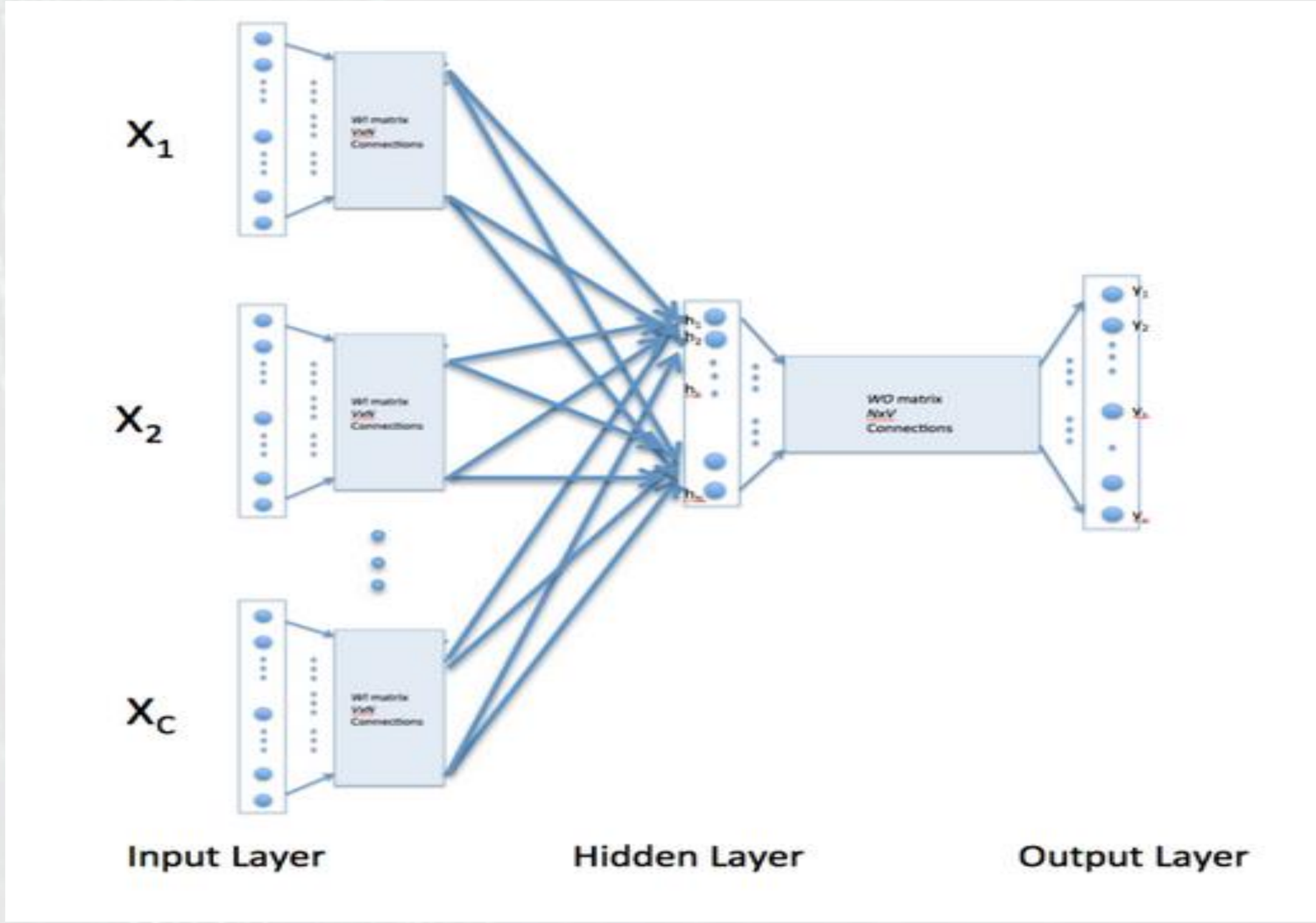
Skip-Gram Model (Mikolov, et. al., 2013)





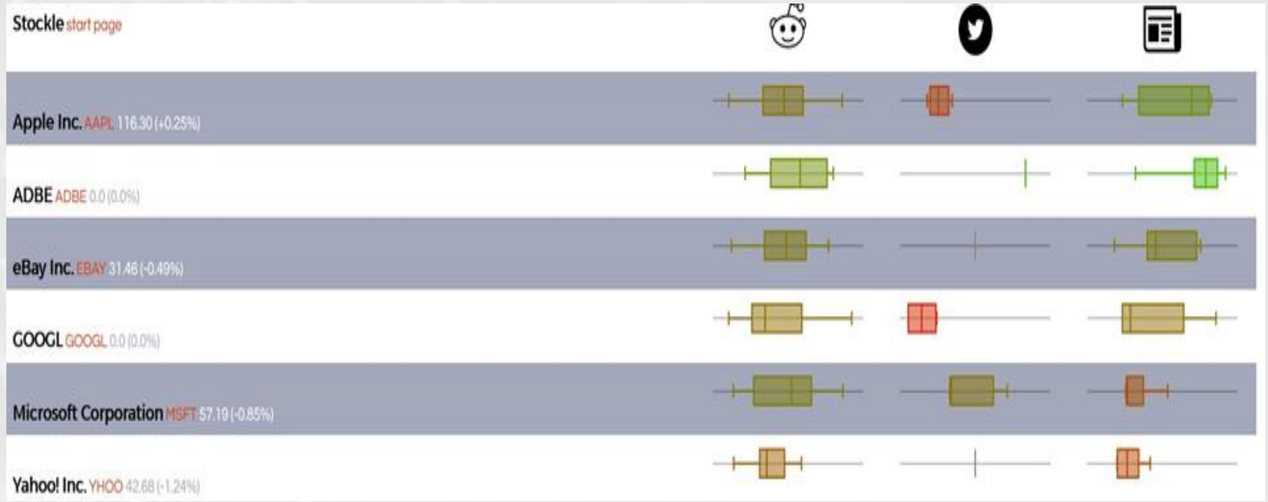
C-BOW Model (Bow, et al., 2003).



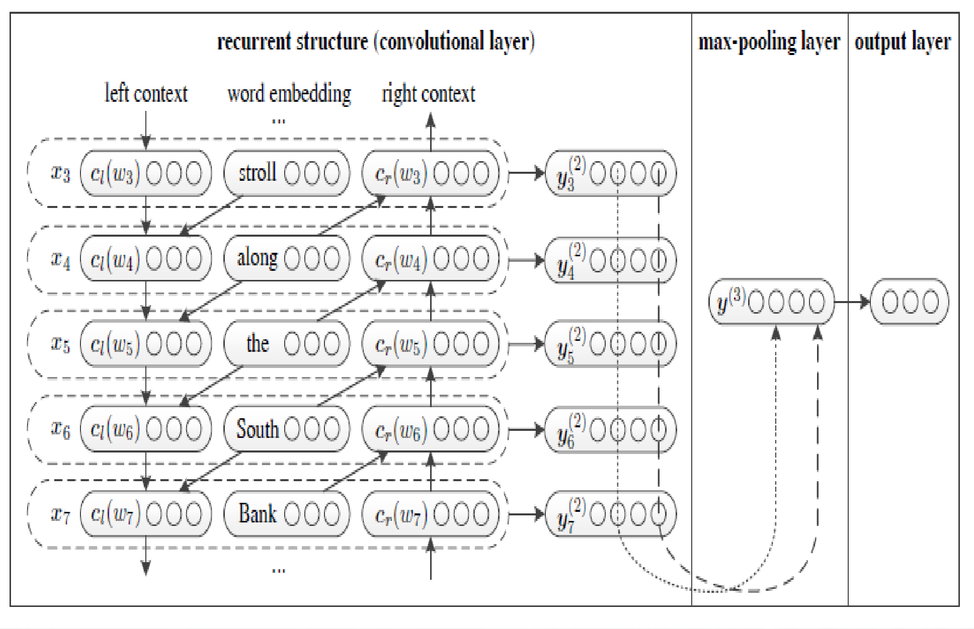


Sentiment Analysis (Ain, et al. 2017)

- **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.

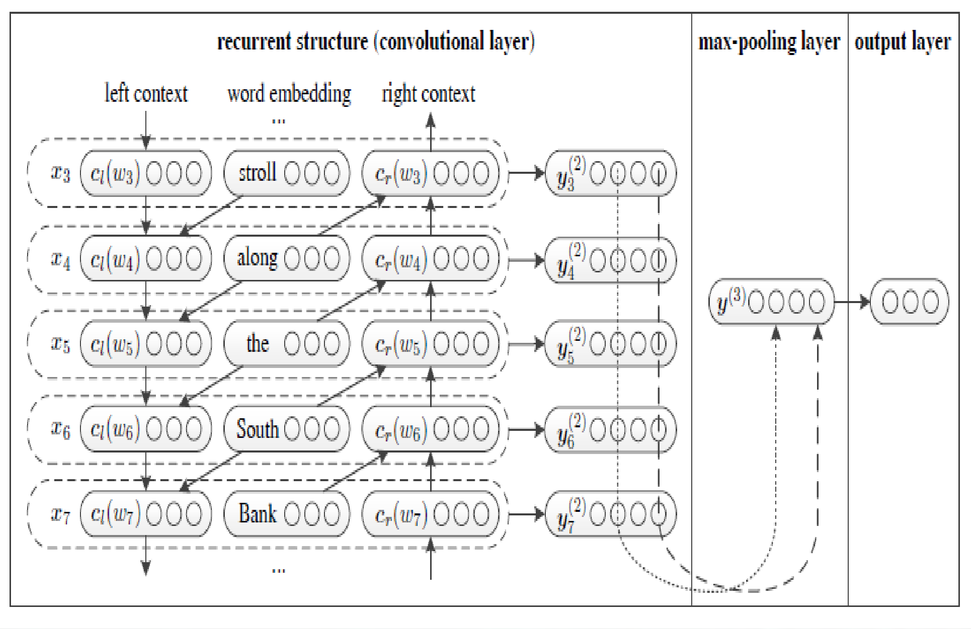


Recurrent Convolutional Neural Networks (RCNN) (Lai, S., et al. 2015)



- They adopt 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).**

Recurrent Convolutional Neural Networks (RCNN) Equations



- RCNNs exhibit a **time complexity of O(n)** , which is linearly correlated with the length of the text length.
- **7 equations** defining all the Neural Network topology
- **Input length** can be **variable**

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

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

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

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

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

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

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

Recurrent Neural Networks are able to understand negations and other things

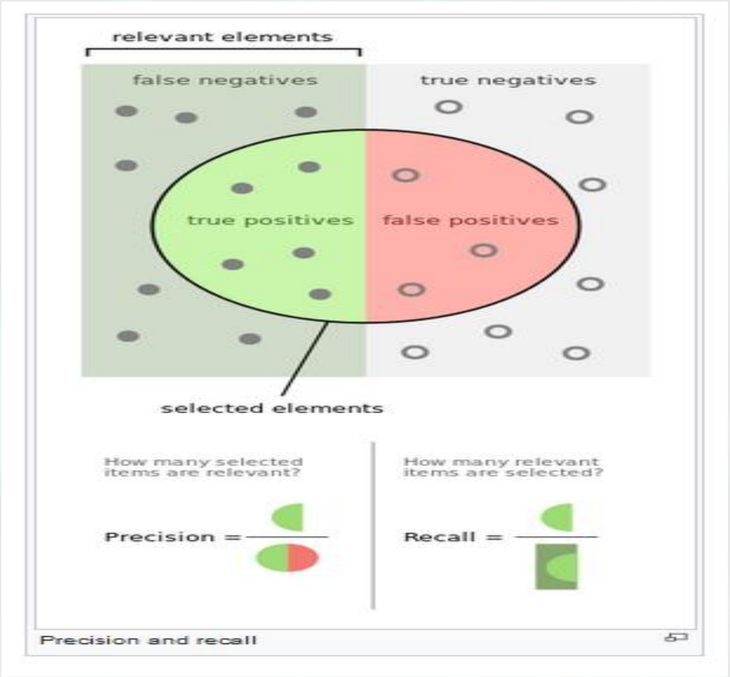
	RCNN
P	well *worth* the; a *wonderful* movie; even *stinging* at; and *invigorating* film; and *ingenious* entertainment; and *enjoy* .; 's *sweetest* movie
N	A *dreadful* live-action; Extremely *boring* .; is *n't* a; 's *painful* .; Extremely *dumb* .; an *awfully* derivative; 's *weaker* than; incredibly *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

- 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



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

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

Deep LSTM/GRU Architectures

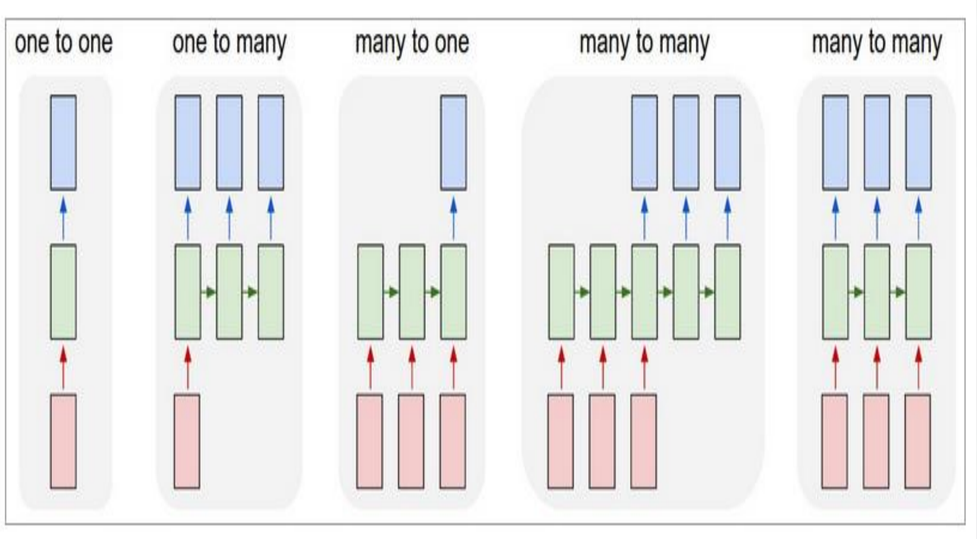
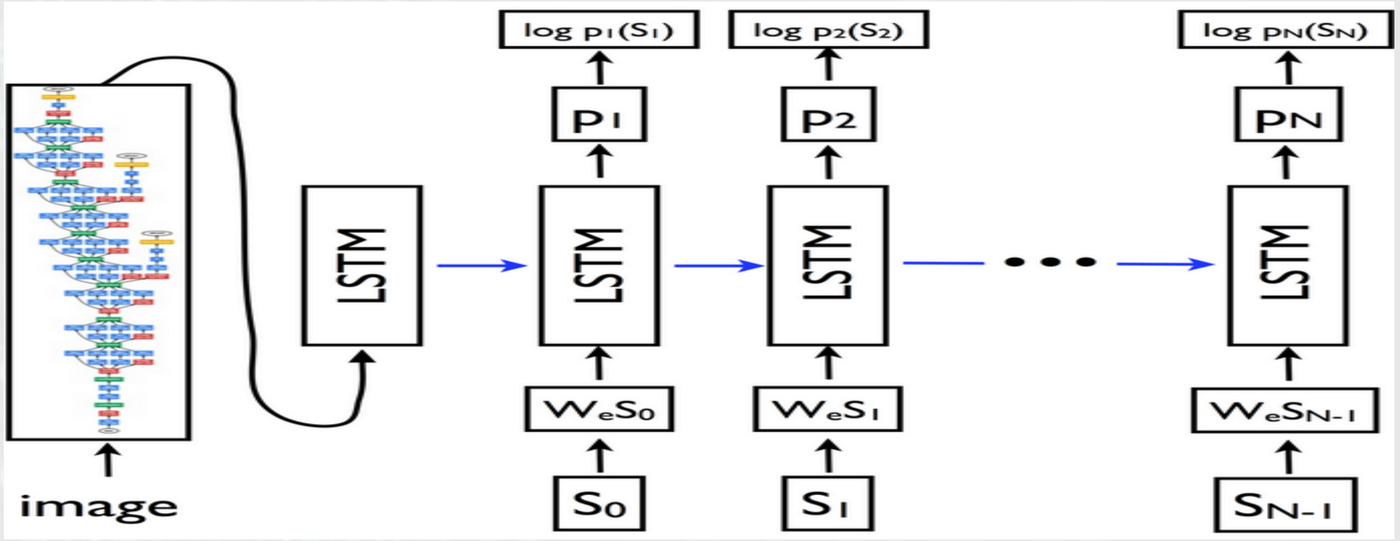
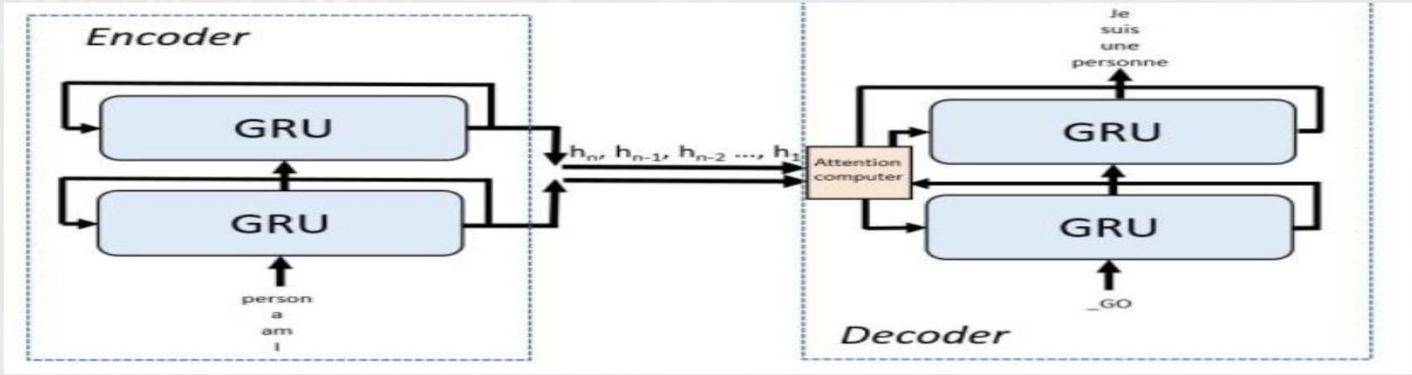


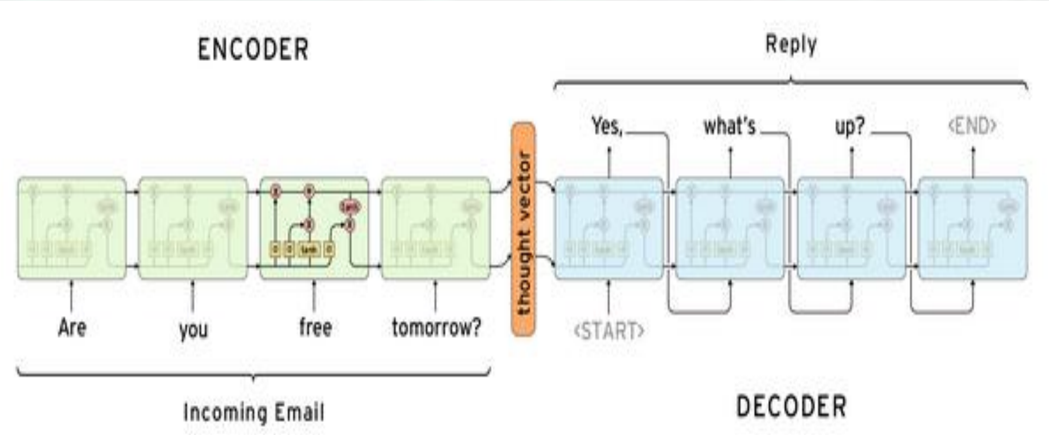
Image Caption Generator



Seq2seq model



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.

Conversation 4: Philosophical Q&A

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*

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 .*

References

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*.

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).

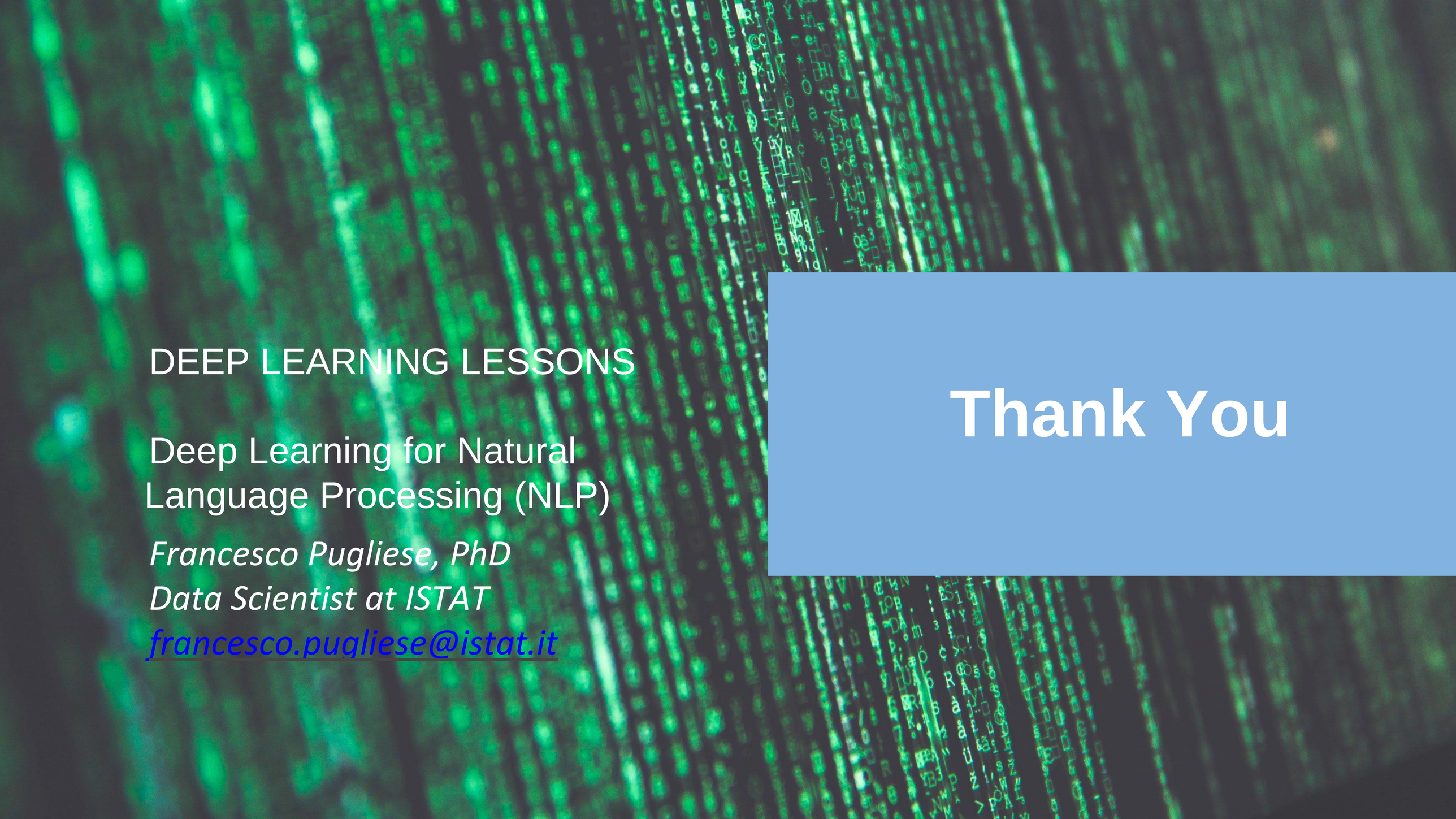
Bow, C., Hughes, B., & Bird, S. (2003, July). Towards a general model of interlinear text. In *Proceedings of EMELD workshop* (pp. 11-13).

References

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).

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).



DEEP LEARNING LESSONS

Deep Learning for Natural Language Processing (NLP)

Francesco Pugliese, PhD
Data Scientist at ISTAT
francesco.pugliese@istat.it

Thank You