

Anatomia degli LLMs



1. Introduzione all'NLP e ai Language Models



Alessio Miaschi

ItaliaNLP Lab, Istituto di Linguistica Computazionale (CNR-ILC), Pisa

alessio.miaschi@ilc.cnr.it

<https://alemmaschi.github.io/>

<http://www.italianlp.it/alessio-miaschi/>

Struttura del corso

1. **26 maggio:** Introduzione all’NLP e ai Language Models
 2. **30 maggio:** Introduzione ai Transformers + Lezione pratica
 3. **11 giugno:** Pipeline di un Transformer + Lezione pratica
 4. **13 giugno:** Utilizzo dei modelli generativi + Lezione pratica
 5. **16 luglio:** Analisi e Interpretabilità dei Transformers + Lezione Pratica
 6. **18 luglio:** Interpretability e Explainability dei Language Models + Lezione Pratica
-

About me and...



Sono un ricercatore (RTD) presso l'[ItaliaNLP Lab](#) dell'Istituto di Linguistica Computazionale “A. Zampolli” ([CNR-ILC](#), Pisa). Nel 2022 ho conseguito il dottorato in Informatica presso l'Università di Pisa.

I miei interessi di ricerca si collocano principalmente nell'ambito del Natural Language Processing (NLP) e nello studio dei modelli linguistici (Language Models, LM). In particolare, mi occupo dell'interpretabilità dei modelli linguistici di larga scala e della valutazione delle loro rappresentazioni interne, con un'enfasi specifica sulla comprensione delle loro capacità linguistiche.

About me and... the team!



Sono un ricercatore (RTD) presso l'[ItaliaNLP Lab](http://www.italianlp.it) dell'Istituto di Linguistica Computazionale "A. Zampolli" (CNR-ILC, Pisa). Nel 2022 ho conseguito il dottorato in Informatica presso l'Università di Pisa.

I miei interessi di ricerca si collocano principalmente nell'ambito del Natural Language Processing (NLP) e nello studio dei modelli linguistici (Language Models, LM). In particolare, mi occupo dell'interpretabilità dei modelli linguistici di larga scala e della valutazione delle loro rappresentazioni interne, con un'enfasi specifica sulla comprensione delle loro capacità linguistiche.



Istituto di Linguistica
Computazionale
"Antonio Zampolli"



Consiglio Nazionale delle Ricerche

L'**ItaliaNLP Lab** (CNR-ILC) riunisce ricercatori, postdoc e studenti provenienti dai settori della linguistica computazionale, dell'informatica e della linguistica, che lavorano allo sviluppo di risorse e algoritmi per l'elaborazione e la comprensione del linguaggio umano.

Permanent Researchers:

- Felice Dell'Orletta
- Simonetta Montemagni
- Dominique Brunato
- Franco Alberto Cardillo
- Giulia Venturi
- Giulia Benotto

Temporary Researchers:

- Chiara Alzetta
- Alessio Miaschi

Research Fellows:

- Agnese Bonfigli
- Cristiano Ciaccio
- Chiara Fazzone
- Ruben Piperno
- Marta Sartor

PhD Students:

- Luca Dini
- Lucia Domenichelli
- Michele Papucci

+ Master/Undergraduate/Visiting Students

Link al sito: <http://www.italianlp.it/>

Introduzione all'NLP e ai Language Models

Natural Language Processing (NLP)

“L'elaborazione del linguaggio naturale (**NLP**, da Natural Language Processing) è una sottobranca di linguistica, informatica e intelligenza artificiale che tratta l'interazione tra i computer e il linguaggio umano”

Fonte: https://it.wikipedia.org/wiki/Elaborazione_del_linguaggio_naturale

“Natural Language Processing is the set of methods for making human language accessible to computers”

(Jacob Eisenstein)

Natural Language Processing e Linguistica Computazionale

- In linguistica, l'oggetto di studio è il **linguaggio naturale**
 - I metodi computazionali sono utilizzati, come in discipline scientifiche quali la biologia computazionale, ma svolgono solo un ruolo di supporto.
- L'**NLP** è incentrato sul design e l'analisi di algoritmi e metodi di rappresentazione per l'elaborazione del linguaggio naturale

Natural Language Processing e Linguistica Computazionale

- In linguistica, l'oggetto di studio è il **linguaggio naturale**
 - I metodi computazionali sono utilizzati, come in discipline scientifiche quali la biologia computazionale, ma svolgono solo un ruolo di supporto.
- L'**NLP** è incentrato sul design e l'analisi di algoritmi e metodi di rappresentazione per l'elaborazione del linguaggio naturale

Requisiti:

- *Fonetica e fonologia*: comprensione dei “suoni” linguistici
- *Morfologia*: conoscenza delle componenti minime dotate di significato
- *Sintassi*: conoscenza delle relazioni tra parole
- *Semantica*: conoscenza dei significato delle parole e delle loro relazioni
- *Pragmatica*: conoscenza della relazione tra il significato e gli obiettivi e le intenzioni di un parlante

Task e Applicazioni dell'NLP

- Spell Checking, Keyword Search, Finding Synonyms
- Part of Speech Tagging
- Extracting information from a website
- Location, people, temporal expressions
- Classifying text
- Sentiment analysis
- Machine translation
- Complex question answering
- Spoken dialog system

Un po' di storia



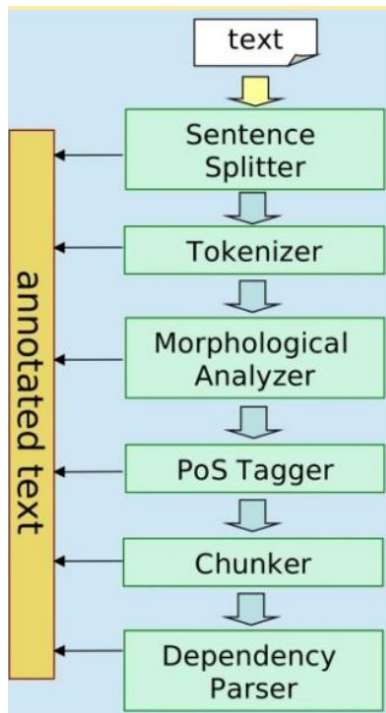
Some Early NLP History

- 1950's:
 - Foundational work: automata, information theory, etc.
 - First speech systems
 - Machine translation (MT) hugely funded by military
 - Toy models: MT using basically word-substitution
 - Optimism!
- 1960's and 1970's: NLP Winter
 - Bar-Hillel (FAHQT) and ALPAC reports kills MT
 - Work shifts to deeper models, syntax
 - ... but toy domains / grammars (SHRDLU, LUNAR)
- 1980's and 1990's: The Empirical Revolution
 - Expectations get reset
 - Corpus-based methods become central
 - Deep analysis often traded for robust and simple approximations
 - *Evaluate everything*
- 2000+: Richer Statistical Methods
 - Models increasingly merge linguistically sophisticated representations with statistical methods, confluence and clean-up
 - *Begin to get both breadth and depth*

Fonte:

https://www.cs.cmu.edu/~arielpro/15381f16/slides/NLP_guest_lecture.pdf

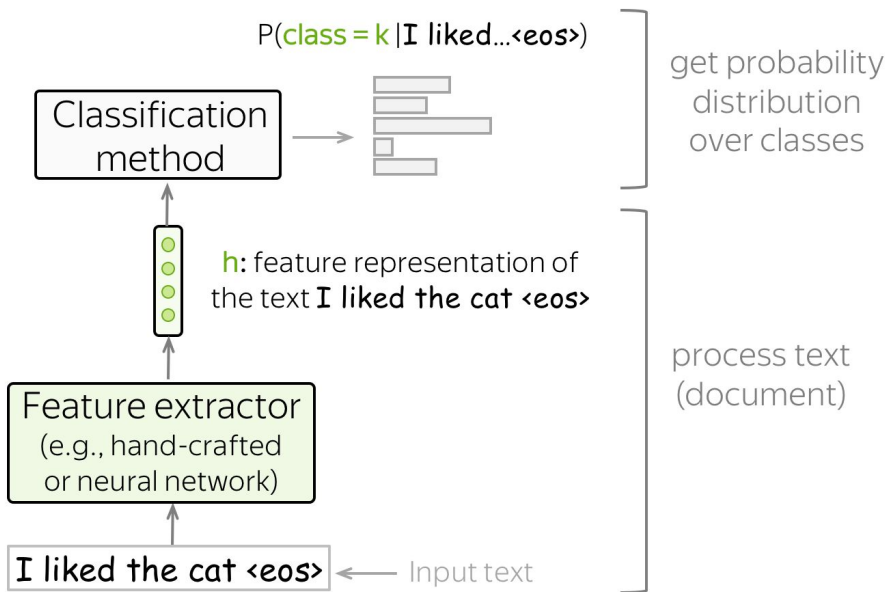
L'NLP Pipeline



La pipeline “tradizionale” prevede l’annotazione automatica di un testo grezzo, al fine di estrarre proprietà dal testo da poter poi utilizzare successivamente per la risoluzione di vari task:

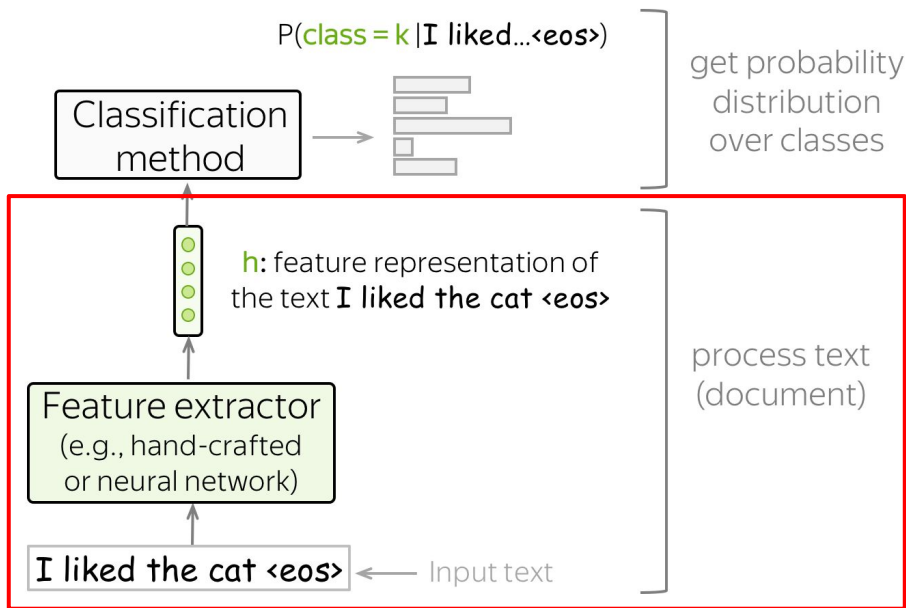
- Sentence splitter: divisione in frasi
- Tokenizer: tokenizzazione, ovvero divisione in token
- Analisi morfologica: analisi automatica della struttura morfologica della frase
- PoS Tagger: annotazione automatica della struttura morfosintattica della frase (i.e. analisi grammaticale)
- Chunking: annotazione delle unità sintattiche minime (e.g. noun phrase, verbal phrase, etc.)
- Dependency parsing: annotazione sintattica delle dipendenze della frase

Dall'NLP Pipeline alla risoluzione dei task



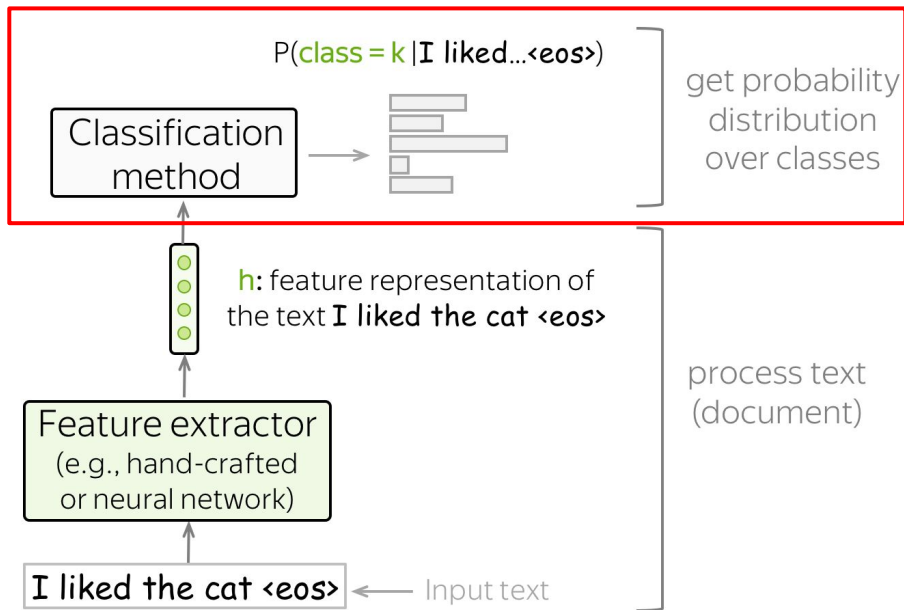
- Prima dell'avvento dei Language Models (LMs) e dei Transformer, l'approccio tradizionale per la risoluzione dei task era strutturato sulla base di queste due componenti:
 - Feature Extractor
 - Classifier

Dall'NLP Pipeline alla risoluzione dei task



- Prima dell'avvento dei Language Models (LMs) e dei Transformer, l'approccio tradizionale per la risoluzione dei task era strutturato sulla base di queste due componenti:
 - **Feature Extractor**
 - **Classifier**

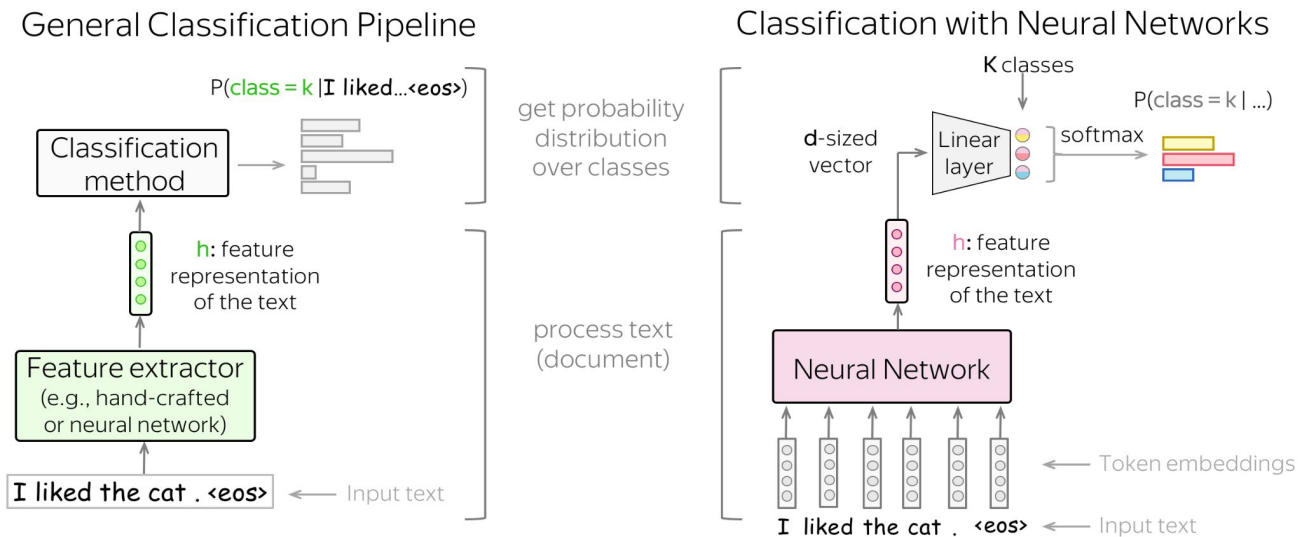
Dall'NLP Pipeline alla risoluzione dei task



- Prima dell'avvento dei Language Models (LMs) e dei Transformer, l'approccio tradizionale per la risoluzione dei task era strutturato sulla base di queste due componenti:
 - Feature Extractor
 - **Classifier**

Dall'NLP Pipeline alle Reti Neurali

- Sfruttando il potenziale delle reti neurali, la rappresentazione delle caratteristiche del testo può essere generata automaticamente sfruttando la struttura della rete neurale



Modelli del linguaggio

Modelli del linguaggio

- Nel contesto di numerosi studi di CL e NLP, si parte dal presupposto che la lingua può essere vista come un *sistema probabilistico*
- Per descrivere e spiegare il funzionamento di un sistema probabilistico è necessario definire un *modello* (probabilistico)
- Un ***modello del linguaggio*** (*language model*), quindi, non è altro che un sistema in grado di assegnare una probabilità a delle sequenze di parole

Modelli del linguaggio

- Ad oggi, i (*Neural*) *Language Model* (NLM) sono considerati lo stato dell'arte per quanto riguarda la risoluzione della maggior parte dei task di *language understanding*:
 - Generazione;
 - Classificazione;
 - Speech recognition;
 - Error/Spell correction;
 - Traduzione automatica;
 - Question Answering;
 - etc.

Language model probabilistici

- Formalmente, una sequenza di parole w_1, \dots, w_n può essere rappresentata come una distribuzione di probabilità:

$$p(w_1, \dots, w_n) = p(w_1)p(w_2|w_1)...p(w_n|w_1, \dots, w_{n-1})$$

Language model probabilistici

- Formalmente, una sequenza di parole w_1, \dots, w_n può essere rappresentata come una distribuzione di probabilità:

$$p(w_1, \dots, w_n) = p(w_1)p(w_2|w_1)\dots p(w_n|w_1, \dots, w_{n-1})$$

- Di conseguenza, la probabilità della parola successiva date quelle precedenti può essere definita come:

$$p(w_n|w_1, \dots, w_{n-1}) = \frac{\text{Count}(w_1, \dots, w_{n-1}, w_n)}{\text{Count}(w_1, \dots, w_{n-1})}$$

Language model probabilistici

- Formalmente, una sequenza di parole w_1, \dots, w_n può essere rappresentata come una distribuzione di probabilità:

$$p(w_1, \dots, w_n) = p(w_1)p(w_2|w_1)...p(w_n|w_1, \dots, w_{n-1})$$

- Di conseguenza, la probabilità della parola successiva date quelle precedenti può essere definita come:

$$P(\textit{the}|\textit{its water is so transparent that}) = \frac{C(\textit{its water is so transparent that the})}{C(\textit{its water is so transparent that})}$$

Language model probabilistici (N-grammi)

- I language model a N-grammi, sfruttando la proprietà di Markov, permettono di approssimare la probabilità di una parola:

$$p(w_i | w_1, \dots, w_{t-1}) \approx p(w_i | w_{i-N}, \dots, w_{i-1})$$

Language model probabilistici (N-grammi)

- I language model a N-grammi, sfruttando la proprietà di Markov, permettono di approssimare la probabilità di una parola:

$$p(w_i | w_1, \dots, w_{t-1}) \approx p(w_i | w_{i-N}, \dots, w_{i-1})$$

- All'aumentare di N , l'approssimazione si fa più precisa, ma la complessità cresce esponenzialmente
- Viceversa, quando $N=1$, il modello ha bisogno di poca informazione, ma le performance sono nettamente inferiori

Language model probabilistici (N-grammi)

Before

$P(\text{I saw a cat on a mat}) =$

$P(\text{I})$

- $P(\text{saw} | \text{I})$
- $P(\text{a} | \text{I saw})$
- $P(\text{cat} | \text{I saw a})$
- $P(\text{on} | \text{I saw a cat})$
- $P(\text{a} | \text{I saw a cat on})$
- $P(\text{mat} | \text{I saw a cat on a})$

After (3-gram)

$P(\text{I saw a cat on a mat}) =$



$P(\text{I})$

- | | | |
|---|---|----------------------------------|
| • $P(\text{saw} \text{I})$ | → | $P(\text{I})$ |
| • $P(\text{a} \text{I saw})$ | → | • $P(\text{saw} \text{I})$ |
| • $P(\text{cat} \text{I saw a})$ | → | • $P(\text{a} \text{I saw})$ |
| • $P(\text{on} \text{I saw a cat})$ | → | • $P(\text{cat} \text{saw a})$ |
| • $P(\text{a} \text{I saw a cat on})$ | → | • $P(\text{on} \text{a cat})$ |
| • $P(\text{mat} \text{I saw a cat on a})$ | → | • $P(\text{a} \text{cat on})$ |
| | | • $P(\text{mat} \text{on a})$ |

ignore use

Language model probabilistici (N-grammi)

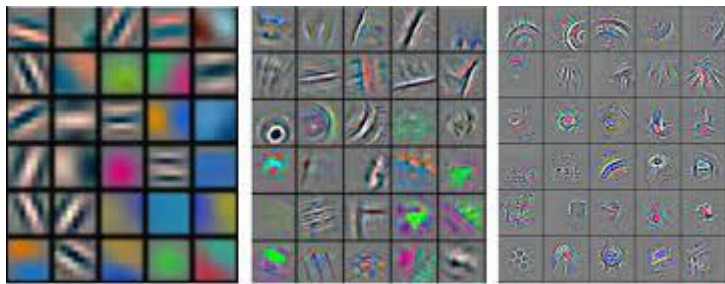
- I language model basati su N-grammi, tuttavia, presentano una serie di limitazioni:
 - A prescindere dal valore assegnato ad N , il modello sarà sempre un'approssimazione della reale distribuzione di probabilità
 - Data la crescita esponenziale della complessità, la scelta di N ricadrà sempre su valori particolarmente bassi (solitamente 2 o 3)
 - Un modello a N-grammi non è in grado di **generalizzare su nuove sequenze di parole**

Word representations

- Le parole possono essere considerate le unità di base di un modello del linguaggio
- Per comprendere una lingua, è anzitutto necessario conoscere il significato delle parole che la compongono
- Per comprendere una determinata lingua, un modello (computazionale) del linguaggio dovrebbe essere in grado di *rappresentare* le parole di quella lingua

Un problema di rappresentazione

- Il *representation learning* è un problema centrale nel contesto dell'Intelligenza Artificiale, delle neuroscienze e della semantica



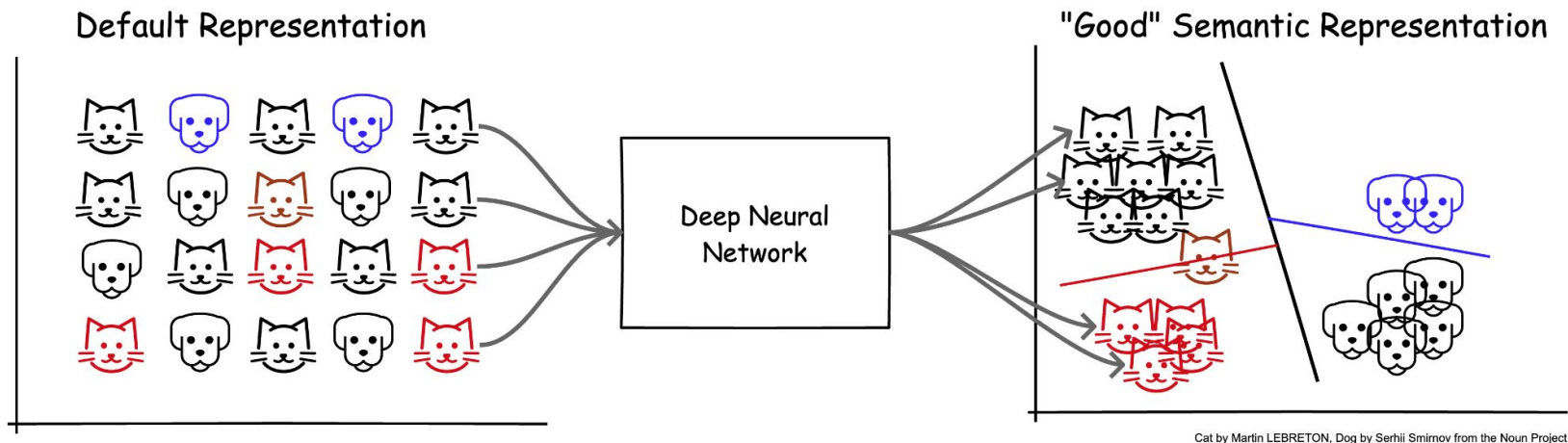
rappresentazione



“scimmia”

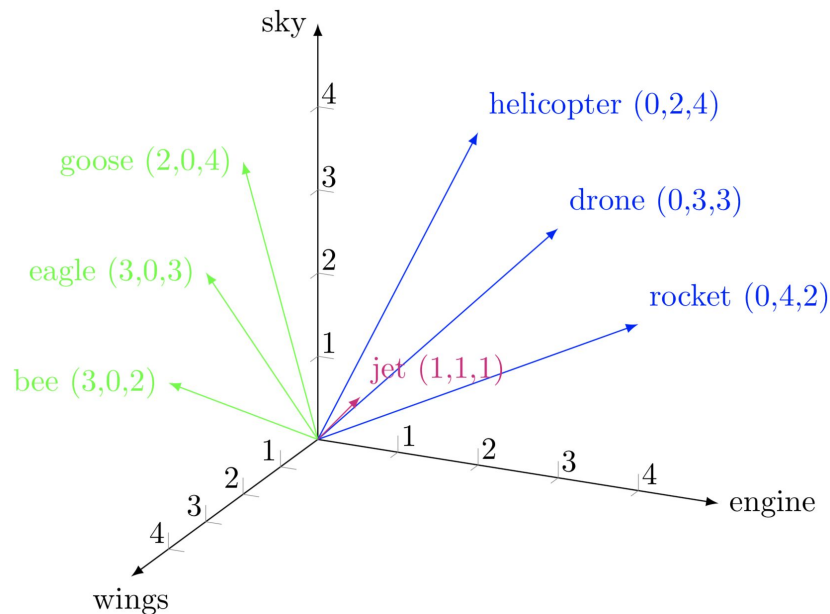
Un problema di rappresentazione

- Il *representation learning* è un problema centrale nel contesto dell'Intelligenza Artificiale, delle neuroscienze e della semantica



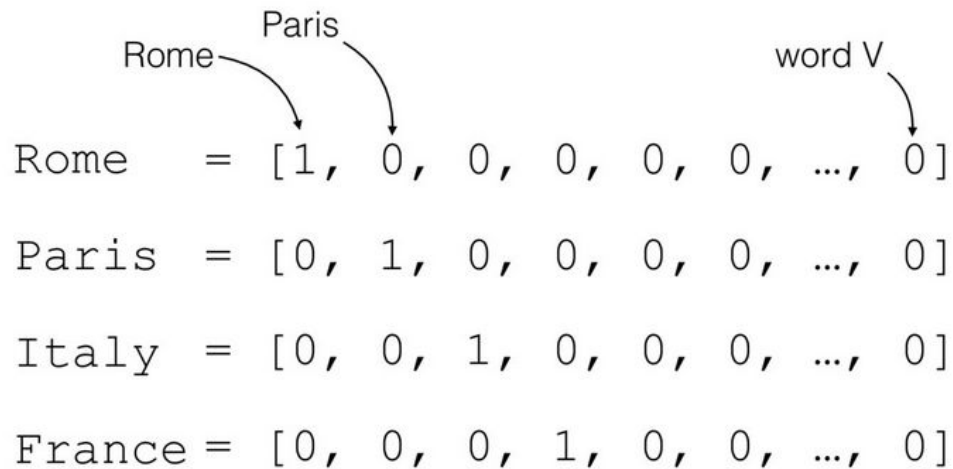
Word representations

- Da un punto di vista computazionale, il metodo più intuitivo per rappresentare una parola è quello di associare ad essa un **vettore di numeri**



One-hot vectors

- Una delle prime tecniche utilizzate è quella di rappresentare le parole tramite *one-hot vectors*



One-hot vectors

- Una delle prime tecniche utilizzate è quella di rappresentare le parole tramite *one-hot vectors*

Rome = [1, 0, 0, 0, 0, 0, ..., 0]

Paris = [0, 1, 0, 0, 0, 0, ..., 0]

Italy = [0, 0, 1, 0, 0, 0, ..., 0]

France = [0, 0, 0, 1, 0, 0, ..., 0]

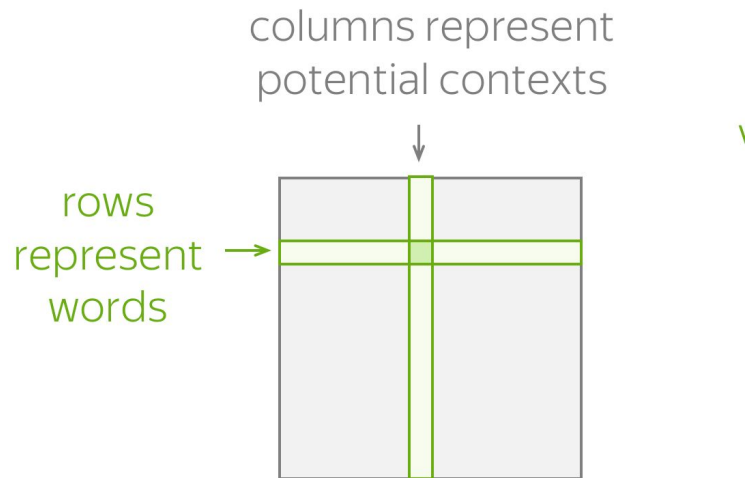
|V|

|V|

Distributional Semantics Hypothesis

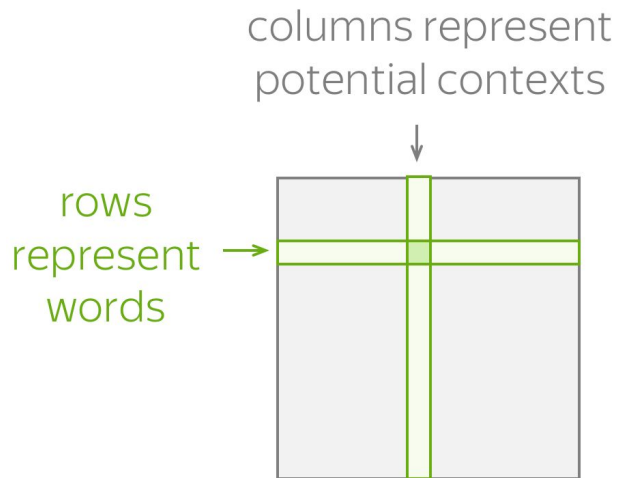
Words that occur in the **same contexts** tend to have **similar meanings** (Harris, 1954)

Metodi Count-based

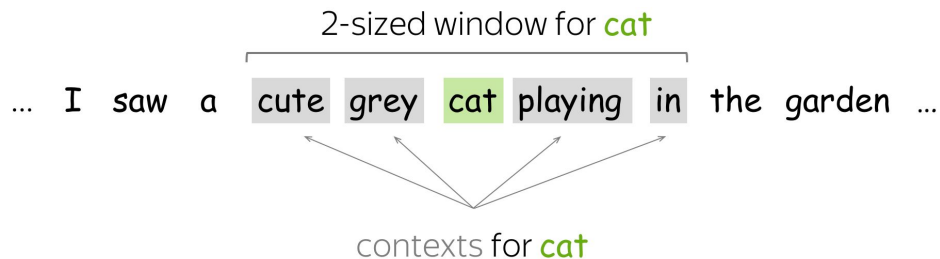


each element says about
the association between a
word and a **context**

Metodi Count-based



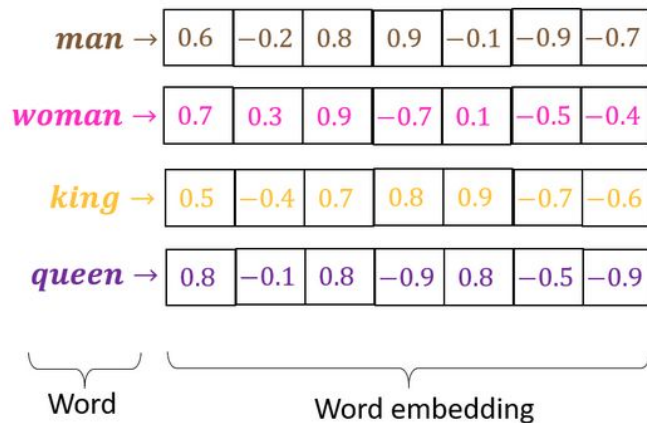
each element says about the association between a **word** and a **context**



Word embeddings

- L'idea è quella di creare **vettori densi** di parole (dimensione $d \ll |V|$) in modo che vettori simili siano associati a parole con significati simili
- Un word embedding è quindi una *rappresentazione distribuita* di una parola

<i>man</i>	→	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
<i>woman</i>	→	0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
<i>king</i>	→	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
<i>queen</i>	→	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9



Word Word embedding

Neural Language Model (NLM)

Le reti neurali artificiali

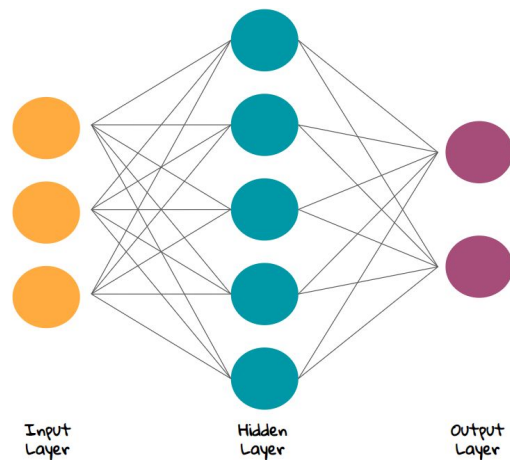
- Nel contesto dell'apprendimento automatico (*machine learning*), una rete neurale artificiale è un modello computazionale composto da **neuroni artificiali**

Le reti neurali artificiali

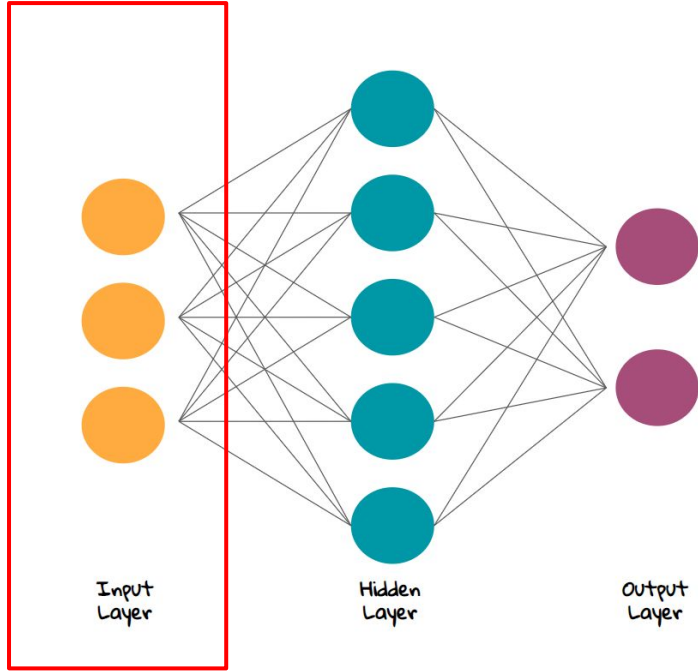
- Nel contesto dell'apprendimento automatico (*machine learning*), una rete neurale artificiale è un modello computazionale composto da **neuroni artificiali**

Ogni rete neurale è composta da:

- uno strato di input (*input layer*)
- uno (o più) strati nascosti (*hidden layers*)
- uno strato di output (*output layer*)

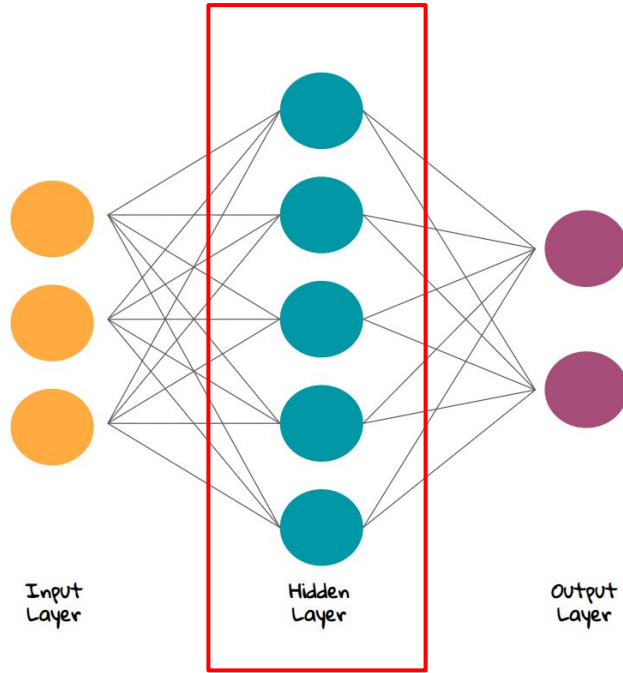


Le reti neurali artificiali



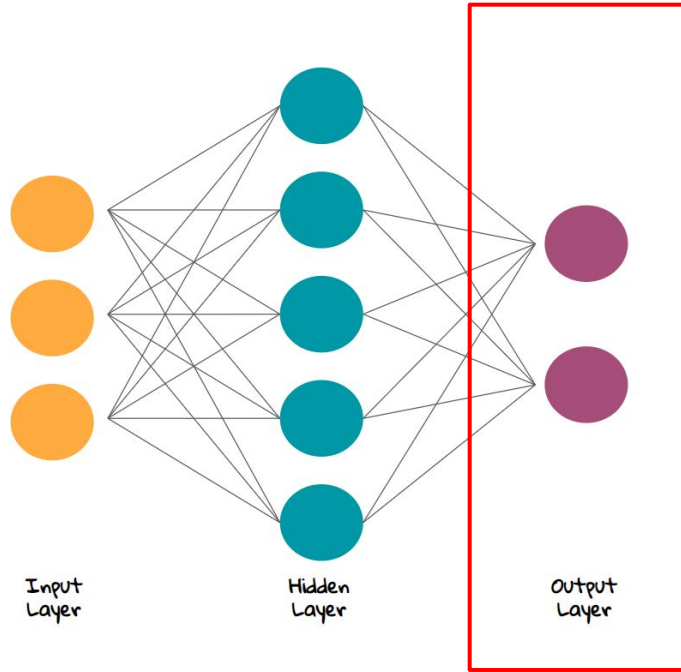
- Dati in input, e.g. immagini, parole, etc

Le reti neurali artificiali



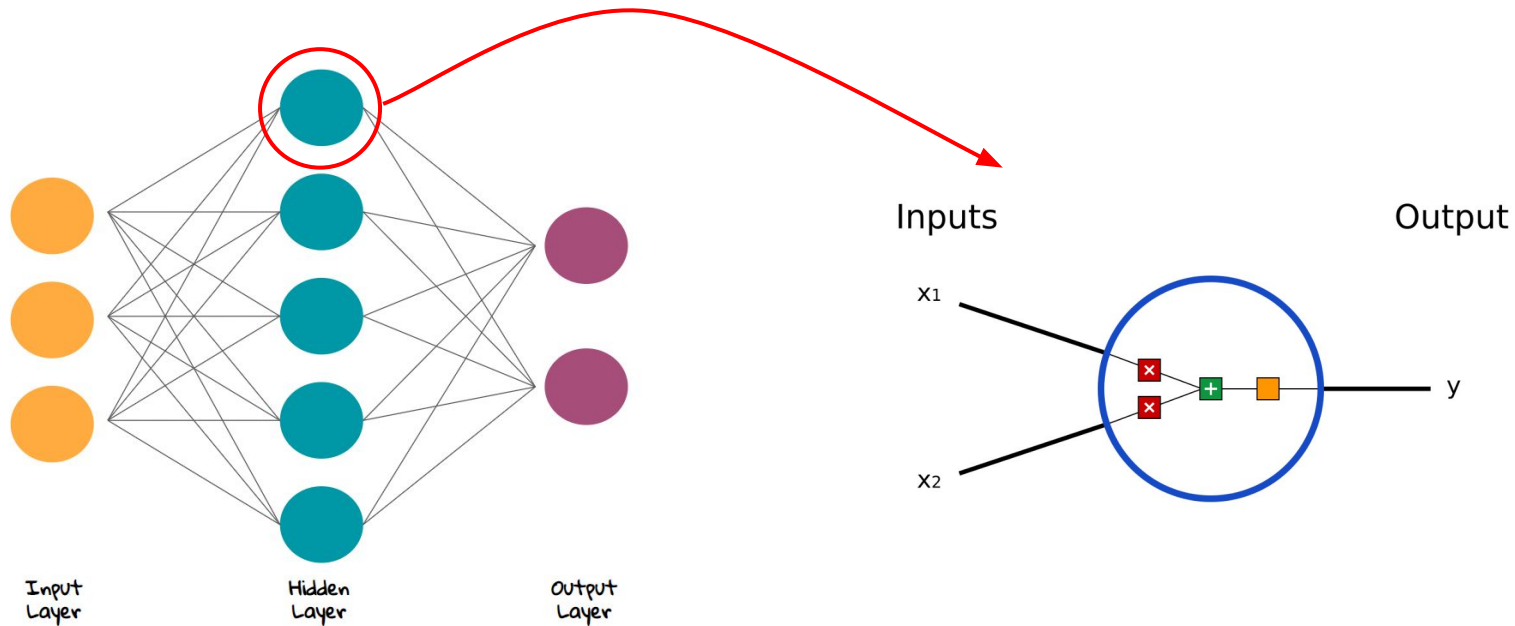
- Rappresentazioni interne

Le reti neurali artificiali

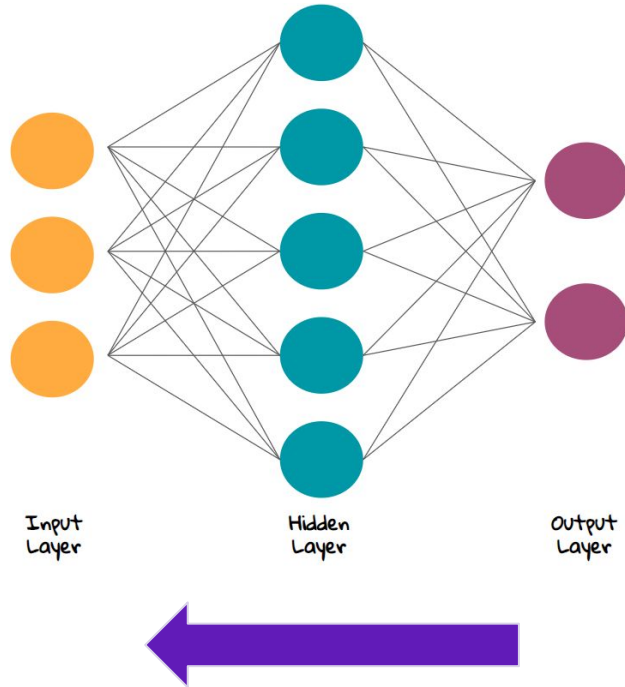


- Output, e.g. l'immagine contiene o non contiene un gatto

Le reti neurali artificiali

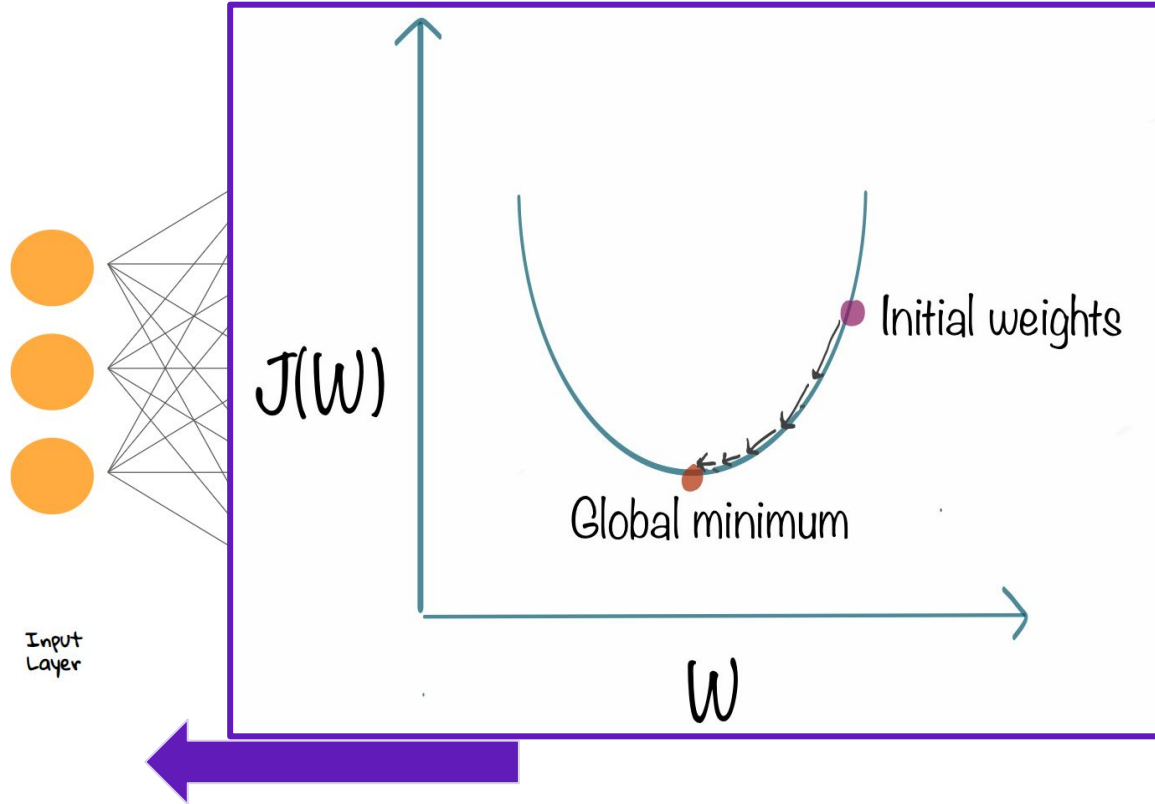


Le reti neurali artificiali



- **Backpropagation:** algoritmo di addestramento della rete

Le reti neurali artificiali



ation: algoritmo
mento della rete

Loss function

- Le reti neurali vengono addestrati per prevedere distribuzioni di probabilità sulle classi
- Intuitivamente, a ogni passo si massimizza la probabilità che il modello predica la classe corretta
- La funzione di loss standard è la **cross-entropy loss**
- Dati:

$$p^* = (0, \dots, 0, 1, 0, \dots) \quad \text{distribuzione target}$$

$$p = (p_1, \dots, p_K) \quad \text{distribuzione del modello}$$

Loss function

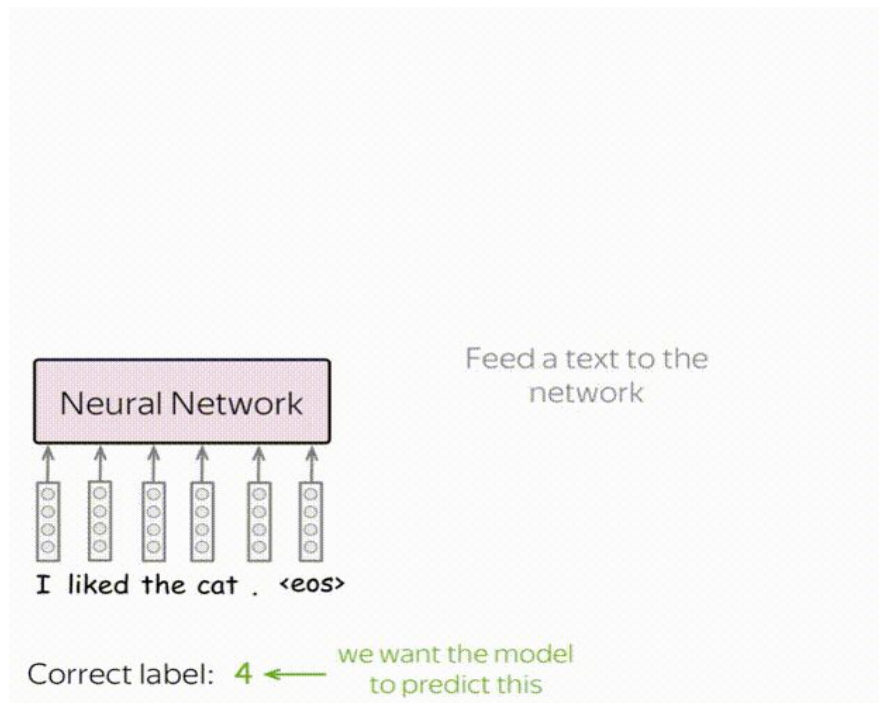
- Le reti neurali vengono addestrati per prevedere distribuzioni di probabilità sulle classi
- Intuitivamente, a ogni passo si massimizza la probabilità che il modello predica la classe corretta
- La funzione di loss standard è la **cross-entropy loss**
- Dati:

$p^* = (0, \dots, 0, 1, 0, \dots)$ distribuzione target

$p = (p_1, \dots, p_K)$ distribuzione del modello

$$Loss(p^*, p) = -p^* \log(p) = - \sum_{i=1}^K p_i^* \log(p_i)$$

Loss function



Neural Language Model (NLM)

- UN NLM è una rete neurale addestrata per approssimare la funzione di **language modeling**

Neural Language Model (NLM)

- UN NLM è una rete neurale addestrata per approssimare la funzione di **language modeling**
- Un modello del linguaggio probabilistico (**LM**) definisce la probabilità di una frase $s = [w_1, w_2, \dots, w_n]$ come:

$$P(s) = \prod_{i=1}^N P(w_i | w_1, w_2, \dots, w_{i-1})$$

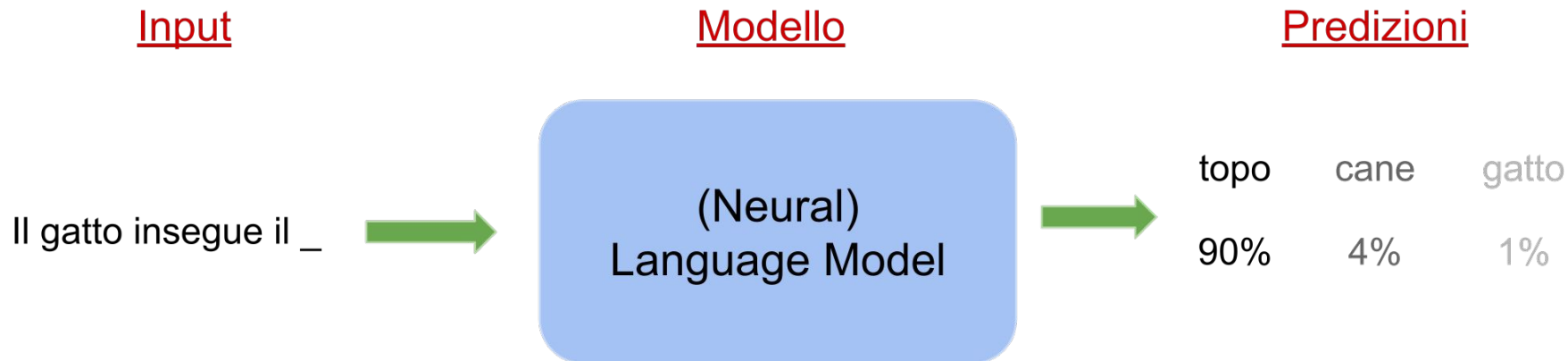
Neural Language Model (NLM)

- UN NLM è una rete neurale addestrata per approssimare la funzione di **language modeling**
- Un modello del linguaggio probabilistico (**LM**) definisce la probabilità di una frase $s = [w_1, w_2, \dots, w_n]$ come:

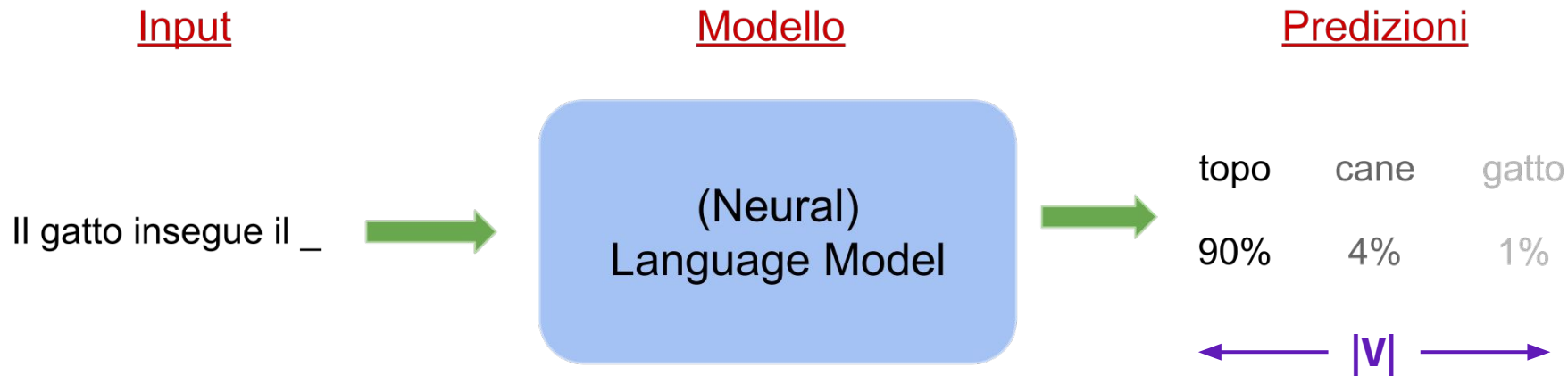
$$P(s) = \prod_{i=1}^N P(w_i | w_1, w_2, \dots, w_{i-1})$$

- **Bengio et al. (2003)** proposero un modello in grado di risolvere tale funzione ricorrendo all'architettura di una rete neurale → **Neural Probabilistic Language Model**

Neural Language Model (NLM)



Neural Language Model (NLM)



Neural Language Model (NLM)

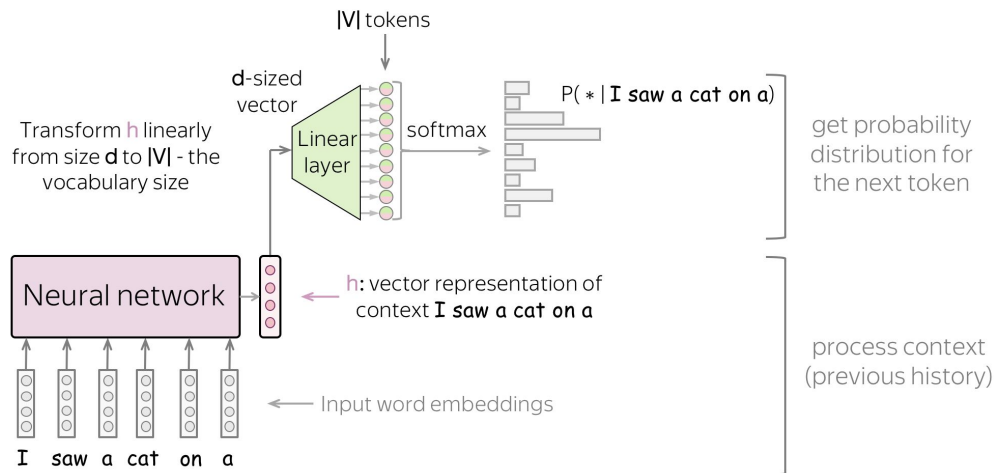
$$Loss(p^*, p) = -p^* \log(p) = -\sum_{i=1}^K p_i^* \log(p_i)$$

Neural Language Model (NLM)

$$Loss(p^*, p) = -p^* \log(p) = - \sum_{i=1}^{\cancel{K}^{|V|}} p_i^* \log(p_i)$$

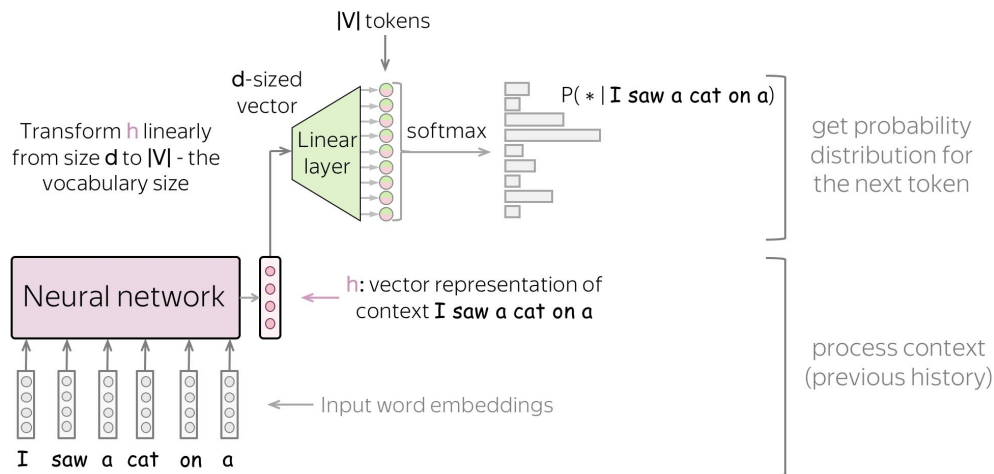
Neural Language Model (NLM)

$$Loss(p^*, p) = -p^* \log(p) = - \sum_{i=1}^{|V|} p_i^* \log(p_i)$$

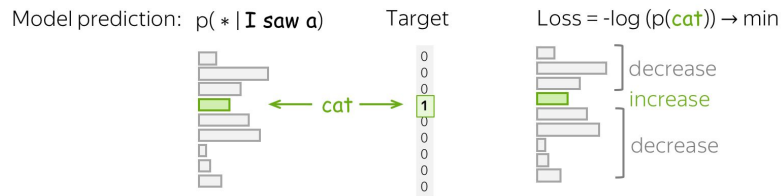


Neural Language Model (NLM)

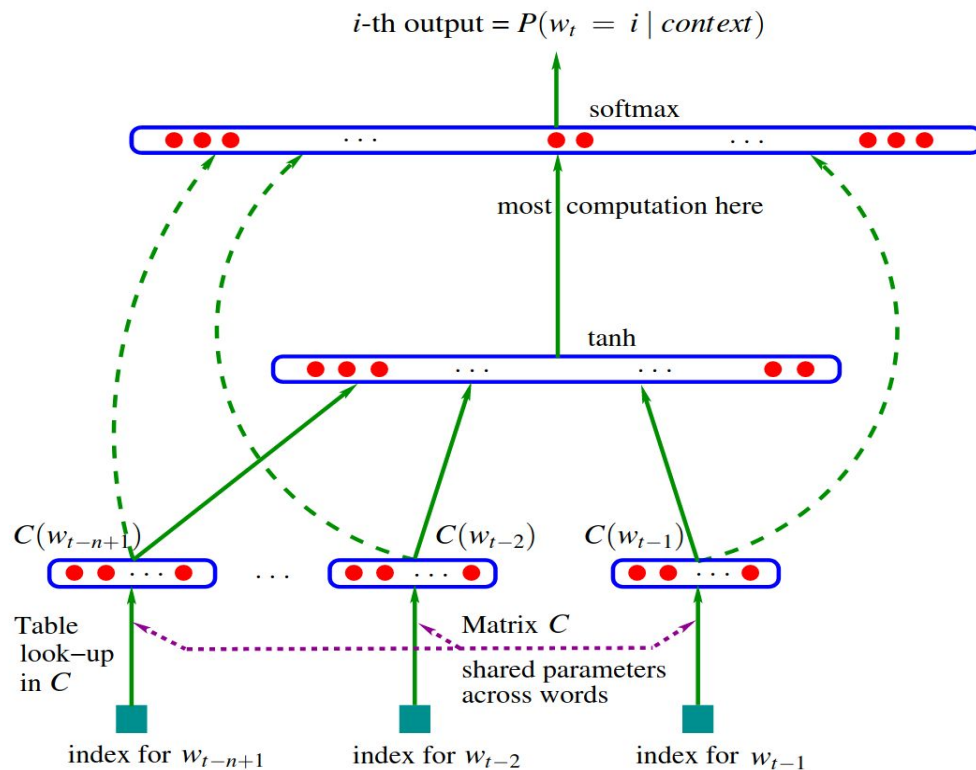
$$Loss(p^*, p) = -p^* \log(p) = - \sum_{i=1}^{|V|} p_i^* \log(p_i)$$



we want the model to predict this
 ↓
 Training example: **I saw a cat** on a mat <eos>

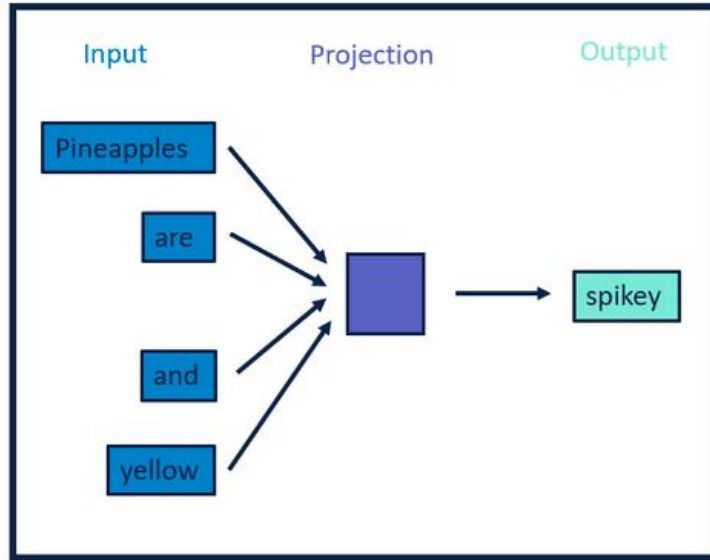


Neural Language Model (NLM)

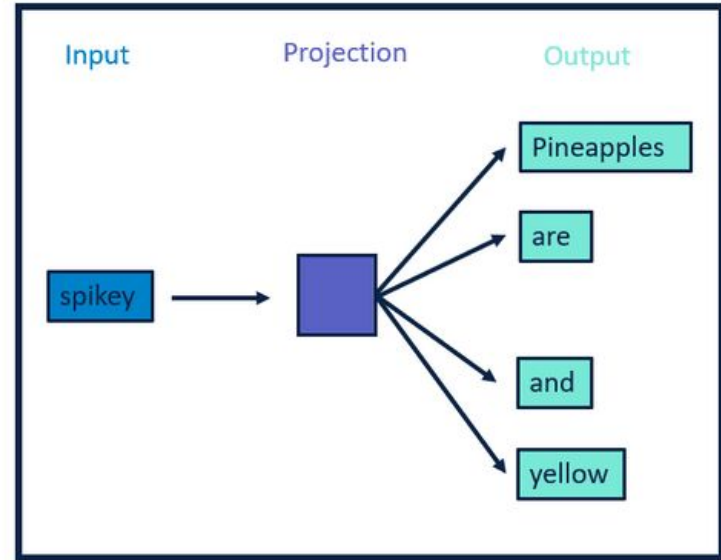


Dalle FFNN ai seq2seq Models

word2vec (Mikolov et al., 2013)

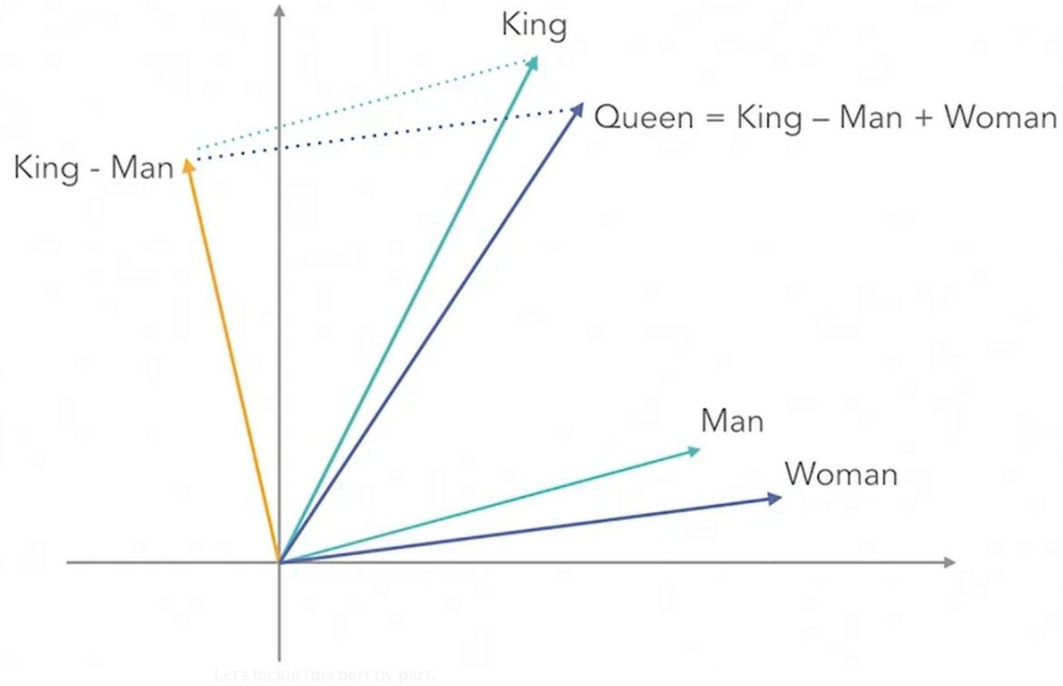


CBOW



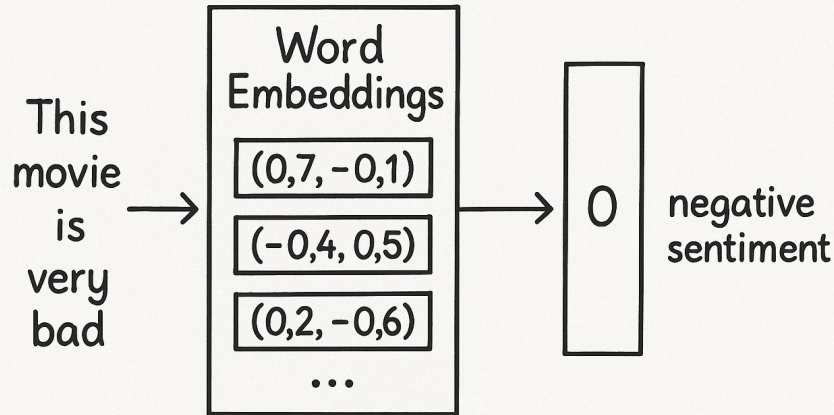
Skip-gram

word2vec (Mikolov et al., 2013)



link: <https://code.google.com/archive/p/word2vec/>

word2vec (Mikolov et al., 2013)



- Gli embeddings di word2vec venivano utilizzati come “input data” di un altro sistema di ML per la risoluzione di task di NLP
 - e.g. sentiment analysis, text classification

word2vec (Mikolov et al., 2013)

Principali limiti:

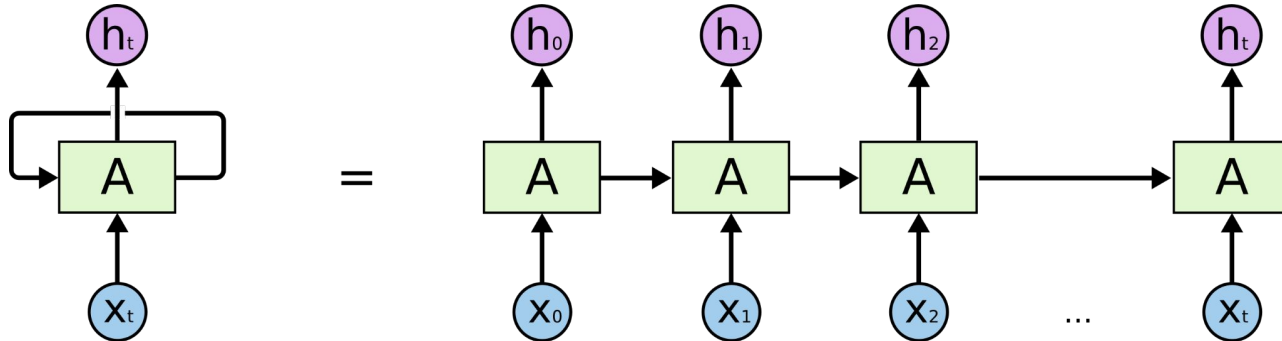
- Le dimensioni della “fixed-window” non sono mai abbastanza grandi → bisogno di più contesto
- La posizione delle parole non è presa in considerazione
- Nessun concetto temporale

Reti neurali ricorrenti (RNN)

- Per sopperire ai limiti delle FFNN, tra il 2013 e il 2014 cominciano a diventare più frequenti i modelli del linguaggio basati su reti neurali ricorrenti (e.g. simple RNNs, LSTMs)

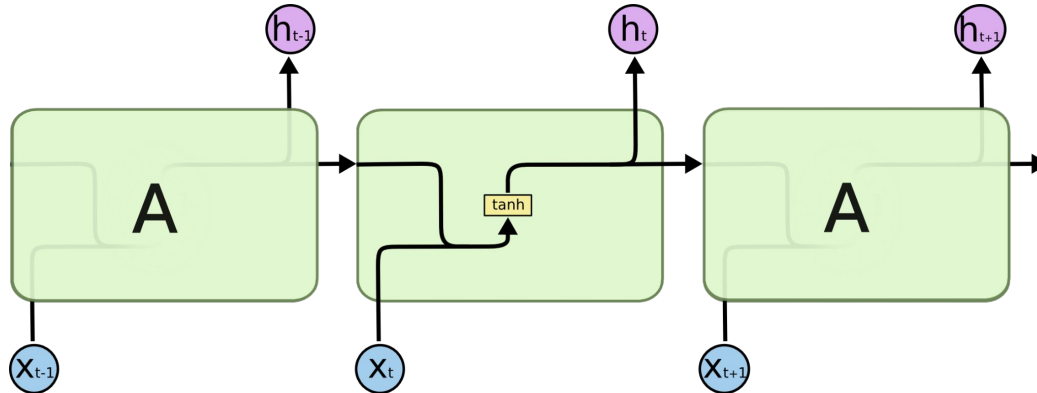
Reti neurali ricorrenti (RNN)

- Per sopperire ai limiti delle FFNN, tra il 2013 e il 2014 cominciano a diventare più frequenti i modelli del linguaggio basati su reti neurali ricorrenti (e.g. simple RNNs, LSTMs)
- Una RNN è una rete neurale in cui le connessioni tra i nodi possono creare un ciclo, consentendo all'output di alcuni nodi di influenzare l'input successivo

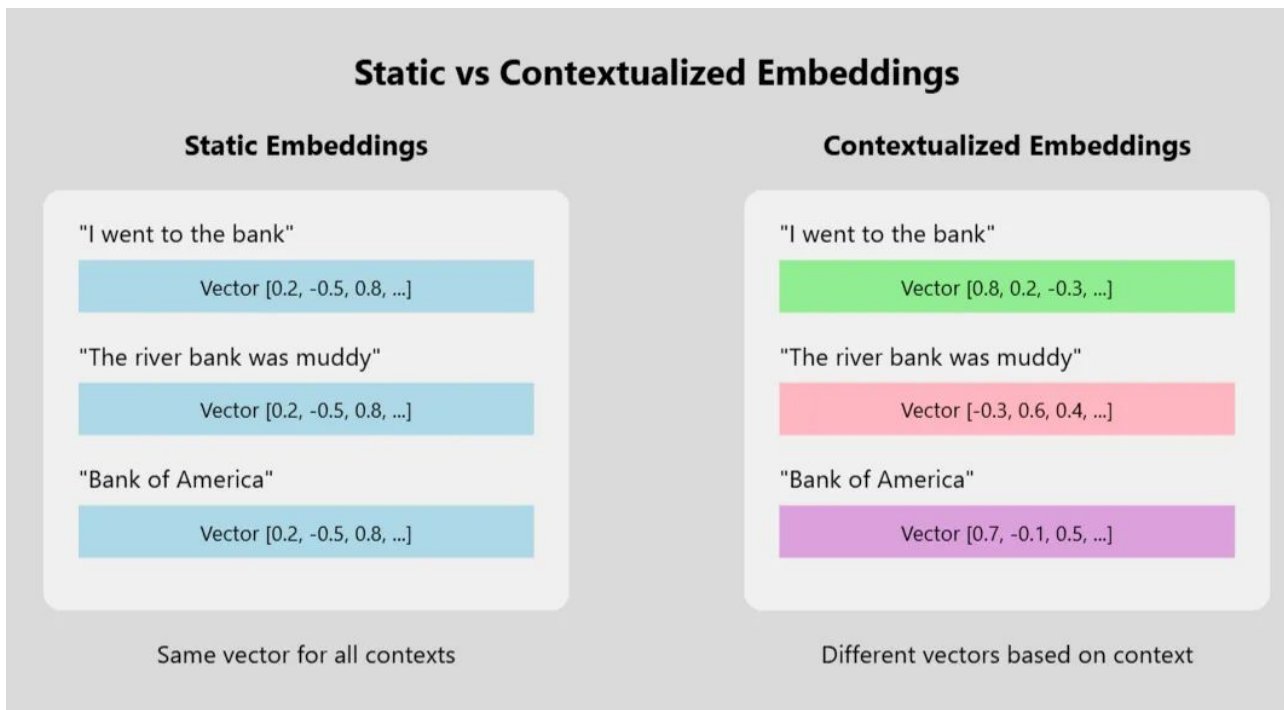


Reti neurali ricorrenti (RNN)

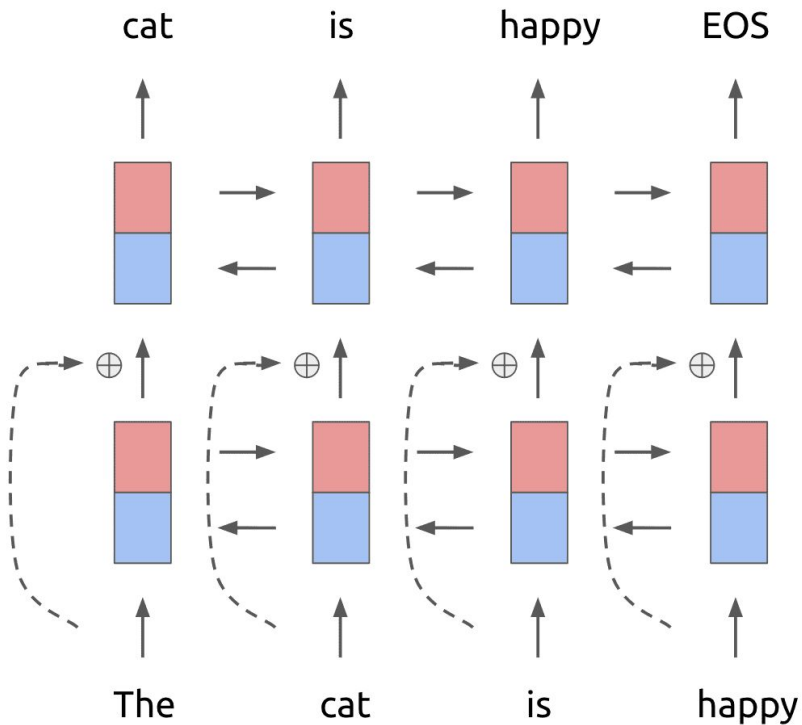
- Per sopperire ai limiti delle FFNN, tra il 2013 e il 2014 cominciano a diventare più frequenti i modelli del linguaggio basati su reti neurali ricorrenti (e.g. simple RNNs, LSTMs)
- Una RNN è una rete neurale in cui le connessioni tra i nodi possono creare un ciclo, consentendo all'output di alcuni nodi di influenzare l'input successivo



Non contextual vs. Contextual Embeddings



ELMo (Peters et al., 2018)



Peters et al. (2018). Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, New Orleans.

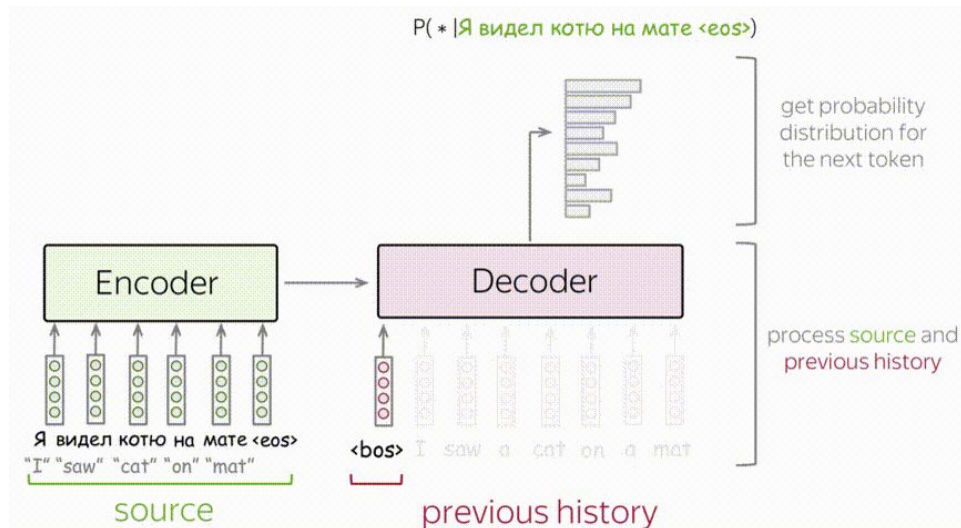
Link: <https://aclanthology.org/N18-1202/>

ELMo explained:

<https://www.mihaileric.com/posts/deep-contextualized-word-representations-elmo/>

Sequence to Sequence (seq2seq) Models

- Modelli “*Sequence to Sequence*” in grado di mappare una data sequenza in input **A**, in una sequenza di output **B**
- Successo soprattutto nel contesto della **Machine Translation**



Sequence to Sequence (seq2seq) Models

- Modelli “
in grado
sequenz
sequenz

Limiti:

Encoder builds a representation of the source and gives it to the decoder

Encoder

Я видел котю на мате <eos>
“I” “saw” “cat” “on” “mat”

Source sentence

Target sentence

I saw a cat on a mat <eos>

Decoder

Decoder uses this source representation to generate the target sentence

e <eos>)

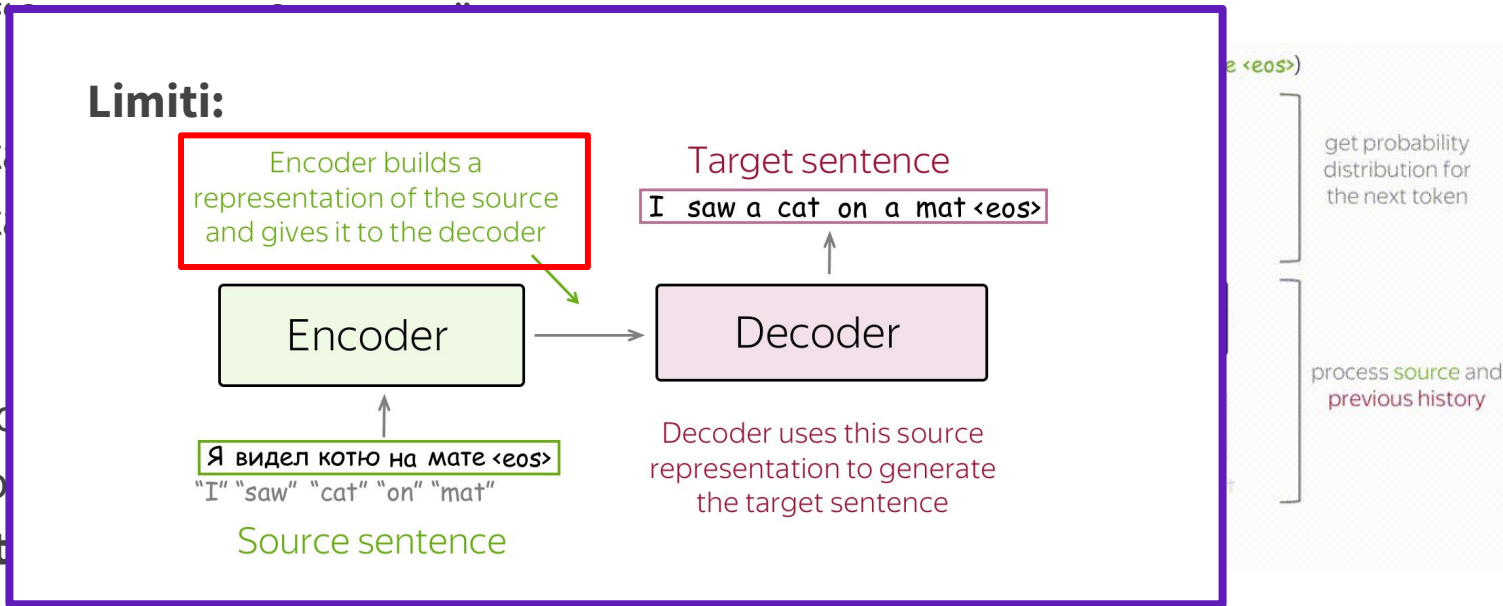
get probability distribution for the next token

process source and previous history

- Successo
contesto
Translat

Sequence to Sequence (seq2seq) Models

- Modelli “
in grado
sequenz
sequenz
- Successo
contesto
Translat



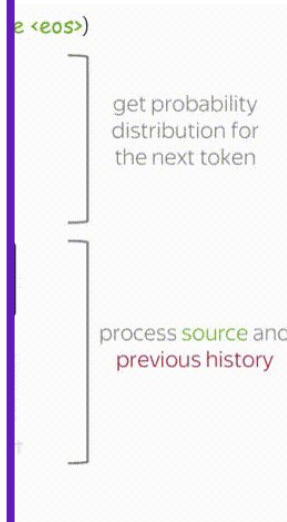
Sequence to Sequence (seq2seq) Models

- Modelli “seq2seq”
in grado di
sequenziare
sequenziare
- Successo
contesto
Traduzione

Limiti:

- Prior research has shown that, roughly speaking, the more such steps decisions require, the harder it is for a recurrent network to learn how to make those decisions.
- The sequential nature of RNNs also makes it more difficult to fully take advantage of modern fast computing devices such as TPUs and GPUs

Da: [Transformer: A Novel Neural Network Architecture for Language Understanding](https://arxiv.org/abs/1609.08144)



Da Sequence to Sequence (seq2seq) and Attention, NLP Course, Lena Voita,
https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html