Building Deep Learning applications with Keras: Recurrent Neural Networks

Felipe Salvatore
https://felipessalvatore.github.io/

Lucas Moura
http://lmoura.me/

June 15, 2018

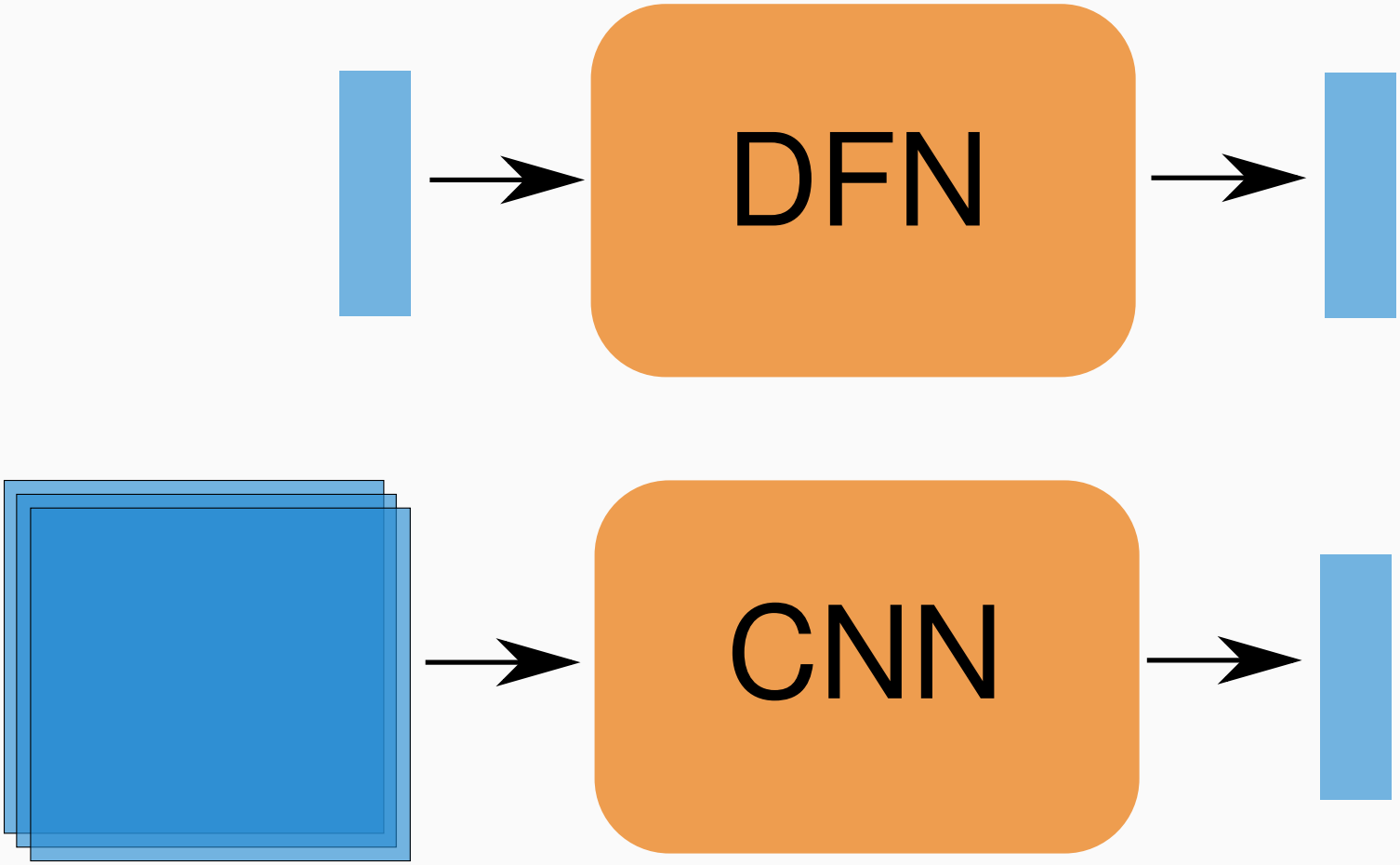
RNN

Basic idea

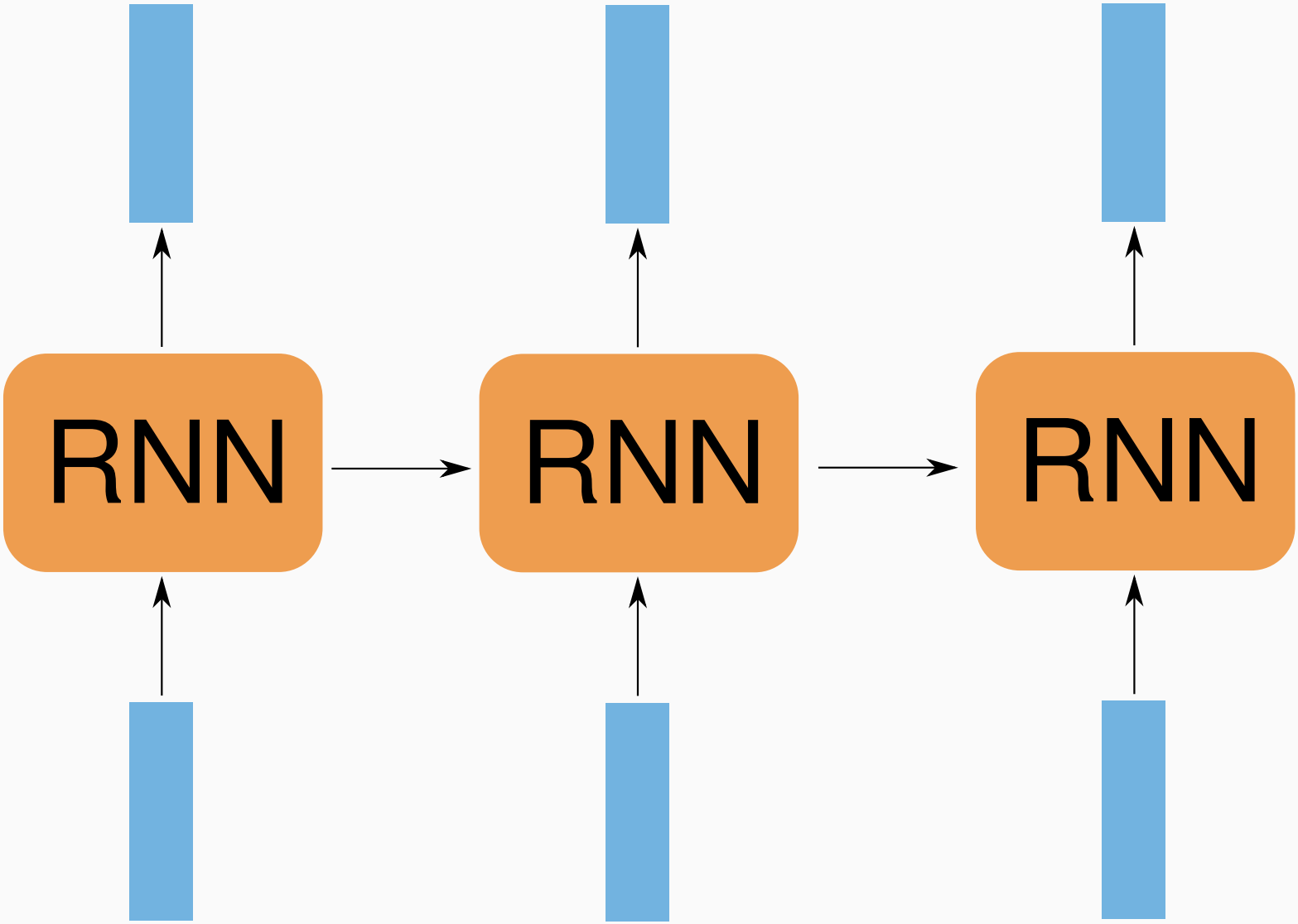
- A Recurrent Neural Network (RNN) allows us to operate over **sequences** of vectors: either sequences in the **input** or the **output**

- This feature differentiate the RNN model from other deep learning architectures such as **Deep Feedforward Network (DFN)** and **Convolutional Neural Network (CNN)**.

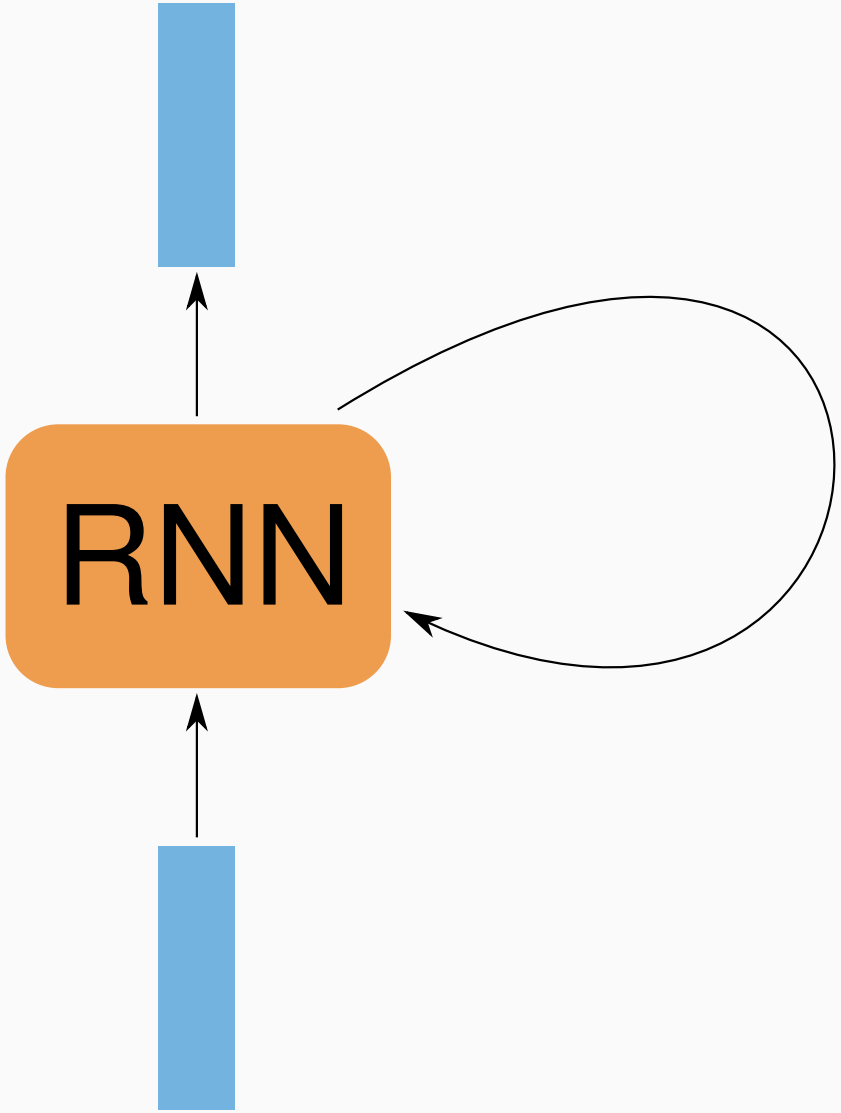
DFN and CNN



RNN: unfold representation



RNN: cyclic representation

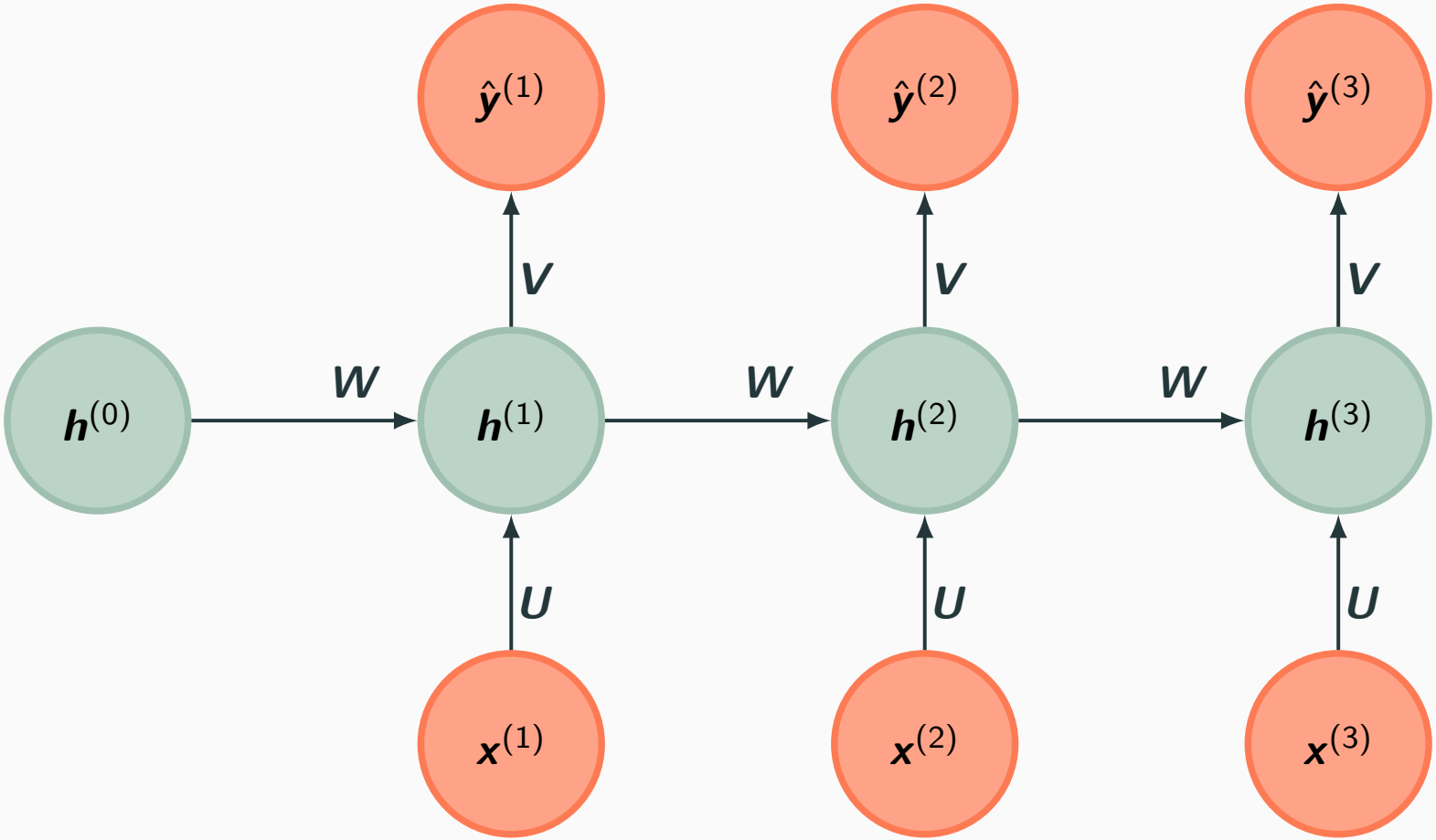


Definition

A RNN is a function f with two inputs:

- An input vector $\boldsymbol{x}$.
- A hidden vector $\boldsymbol{h}$ representing a summary of all past inputs, called state or cell state.

Unfolding the graph



Text data

- RNNs are widely used for processing text data.
- But how can we feed a RNN with text data ?

Text data

I happened across "Bait" on cable one night just as it started and thought, "Eh, why not?" I'm glad I gave it a chance. "Bait" ain't perfect. It suffers from unnecessarily flashy direction and occasional dumbness. But overall, this movie worked. All the elements aligned just right, and they pulled off what otherwise could have been a pretty ugly film. Most of that, I think, is due to Jamie Foxx. I don't know who tagged Foxx for the lead, but whoever it was did this movie a big favor. Believable and amazingly likeable, Foxx glides through the movie, smooth as butter and funnier than hell.

Text data

- Text pre-processing
- Punctuation removal, turn words into lower case ...
- Task dependent

Text data

i happened across bait on cable one night just as it started and thought eh why not? i'm glad i gave it a chance bait ain't perfect it suffers from unnecessarily flashy direction and occasional dumbness but overall this movie worked all the elements aligned just right and they pulled off what otherwise could have been a pretty ugly film most of that i think is due to jamie foxx i don't know who tagged foxx for the lead but whoever it was did this movie a big favor believable and amazingly likeable foxx glides through the movie smooth as butter and funnier than hell

Text data

- Now, let's identify all the unique words in the text
- Let's order this words by the number of times they appear in the text, or their frequency

Text data

unique_words = ['i', 'happened', 'across', 'bait', 'on', 'cable', 'one', 'night', 'just', 'as', 'it', 'started', 'and', 'thought', 'eh', 'why', 'not', "i'm", 'glad', 'gave', ..., 'did', 'big', 'favor', 'believable', 'amazingly', 'likeable', 'glides', 'through', 'smooth', 'butter', 'funnier', 'than', 'hell']

Text data

- We can see that each word has an index in the list.
- For example, the word **happened** has index 2, while the word **night** has index 8
- Now, we will represent each of our words with the index they represent.

Text data

i happened across bait on cable one night just as it started and thought eh why not? i'm glad i gave it a chance bait ain't perfect it suffers from unnecessarily flashy direction and occasional dumbness but overall this movie worked all the elements aligned just right and they pulled off what otherwise could have been a pretty ugly film most of that i think is due to jamie foxx i don't know who tagged foxx for the lead but whoever it was did this movie a big favor believable and amazingly likeable foxx glides through the movie smooth as butter and funnier than hell

Text data

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1, 20, 11, 21, 22, 4, 23, 24, 11, 25, 26, 27, 28, 29, 13, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 9, 41, 13, 42, 43, 44, 45, 46, 47, 48, 49, 21, 50, 51, 52, 53, 54, 55, 1, 56, 57, 58, 59, 60, 61, 1, 62, 63, 64, 65, 61, 66, 38, 67, 32, 68, 11, 69, 70, 34, 35, 21, 71, 72, 73, 13, 74, 75, 61, 76, 77, 38, 35, 78, 10, 79, 13, 80, 81, 82

Text data

- Why should we transform the words into number like this ?
- We need that because each index will map a word to it embedding representation
- But what is a word embedding ?

Word Embedding

- It is a vectorial representation of a word
- For example, the word **and** could be mapped to [0.1, -0.2, 0.5, 0.6, 0.7]
- Therefore, every word in our text is represented by a vectorial representation

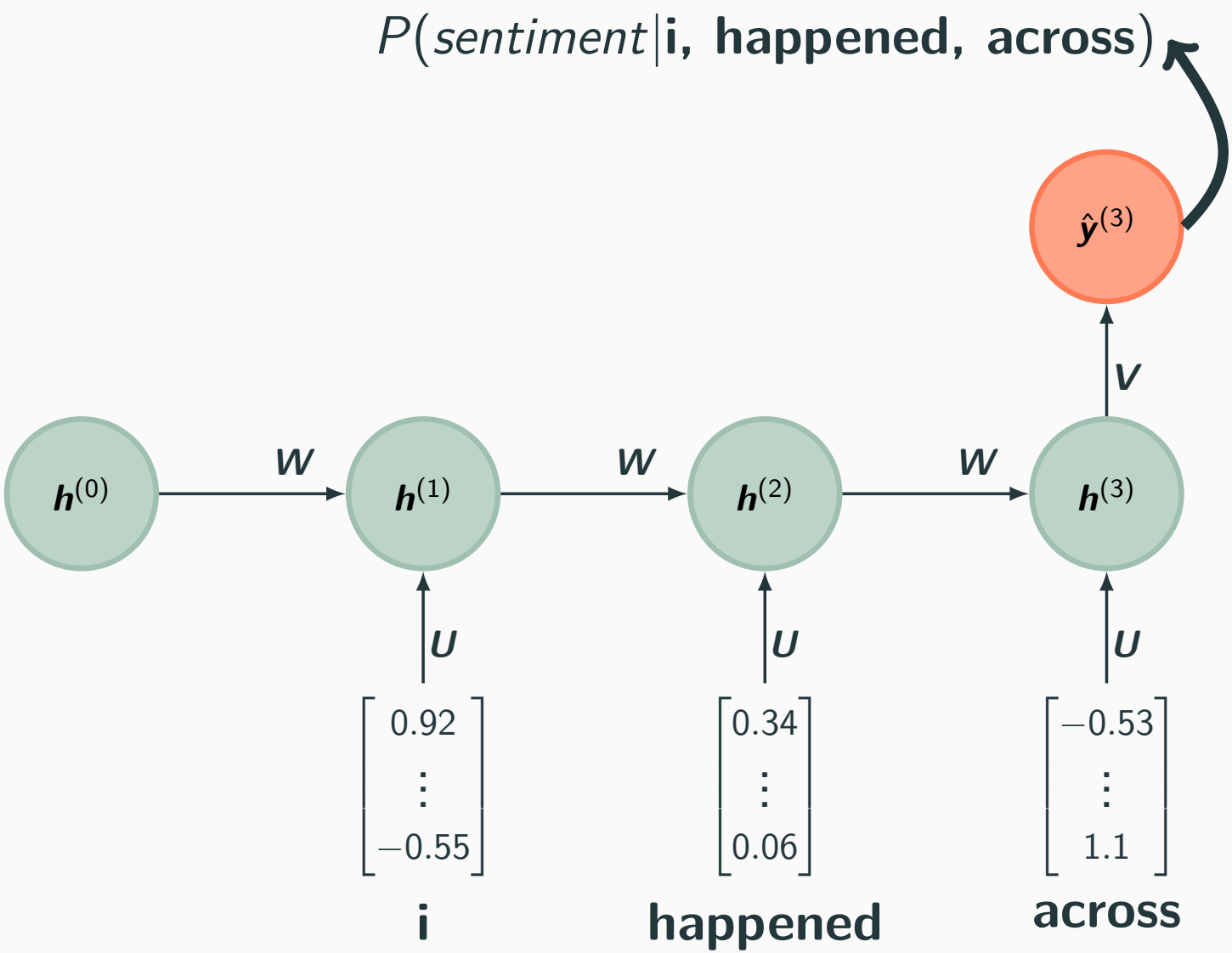
i	0.1	-0.2	0.5	0.8	0.3	-0.1	-0.7	0.8
happened	0.5	0.1	0.2	0.8	0.5	0.9	-0.6	0.1
across	0.9	-0.6	0.3	0.1	0.8	0.4	0.5	0.2
bait	-0.2	0.8	0.5	0.6	0.7	0.5	0.1	-0.3
on	0.8	0.5	0.6	0.3	0.2	0.9	-0.2	0.4
cable	-0.5	0.2	-0.3	0.8	0.3	0.1	0.6	0.5
one	0.1	0.3	-0.9	-0.3	0.8	-0.2	0.7	0.6
night	0.2	0.4	0.3	-0.2	0.9	-0.8	-0.9	-0.8

Table 1: Word Embedding representation

1	0.1	-0.2	0.5	0.8	0.3	-0.1	-0.7	0.8
2	0.5	0.1	0.2	0.8	0.5	0.9	-0.6	0.1
3	0.9	-0.6	0.3	0.1	0.8	0.4	0.5	0.2
4	-0.2	0.8	0.5	0.6	0.7	0.5	0.1	-0.3
5	0.8	0.5	0.6	0.3	0.2	0.9	-0.2	0.4
6	-0.5	0.2	-0.3	0.8	0.3	0.1	0.6	0.5
7	0.1	0.3	-0.9	-0.3	0.8	-0.2	0.7	0.6
8	0.2	0.4	0.3	-0.2	0.9	-0.8	-0.9	-0.8

Table 2: Word Embedding index representation

The language model: graph



Benefits of Word Embeddings

- We capture semantic information of words
- Words with similar meanings will have similar vectorial representations
- It a format our RNNs can understand

Creating Word Embedding

- We can use pre-trained word embeddings, which means that someone has already created them for us:
  - **Stanford English Word Embeddings**: https://nlp.stanford.edu/projects/glove/
  - **ICMC Portuguese Word Embeddings**: http://www.nilc.icmc.usp.br/embeddings
- Or we can train these embeddings inside our model, meaning that during our training, we will train both the model and the embedding matrix.

Back to RNNs

- **Pro:** Read sequential data like a human do, can be used to a wide range of sequential data, like time series.
- **Cons:** Slow to train due to sequence processing, short memory

Short Memory

- Unfortunetely, RNNs can't capture long relations due to its archictecture.
- However, we have nowadays more advances types od Recurrent Neural Networks that can handle that issue, such as a Long Short Term Memory (LSTM)

LSTM

- LSTM allow the model to learn what forget and what to keep from their past.
- They can better handle long term dependecies in data.

Time to practice

But let's now see what RNN can do in practice

References I


I. Goodfellow, Y. Bengio, and A. Courville.
***Deep Learning*.**
MIT Press, 2017.