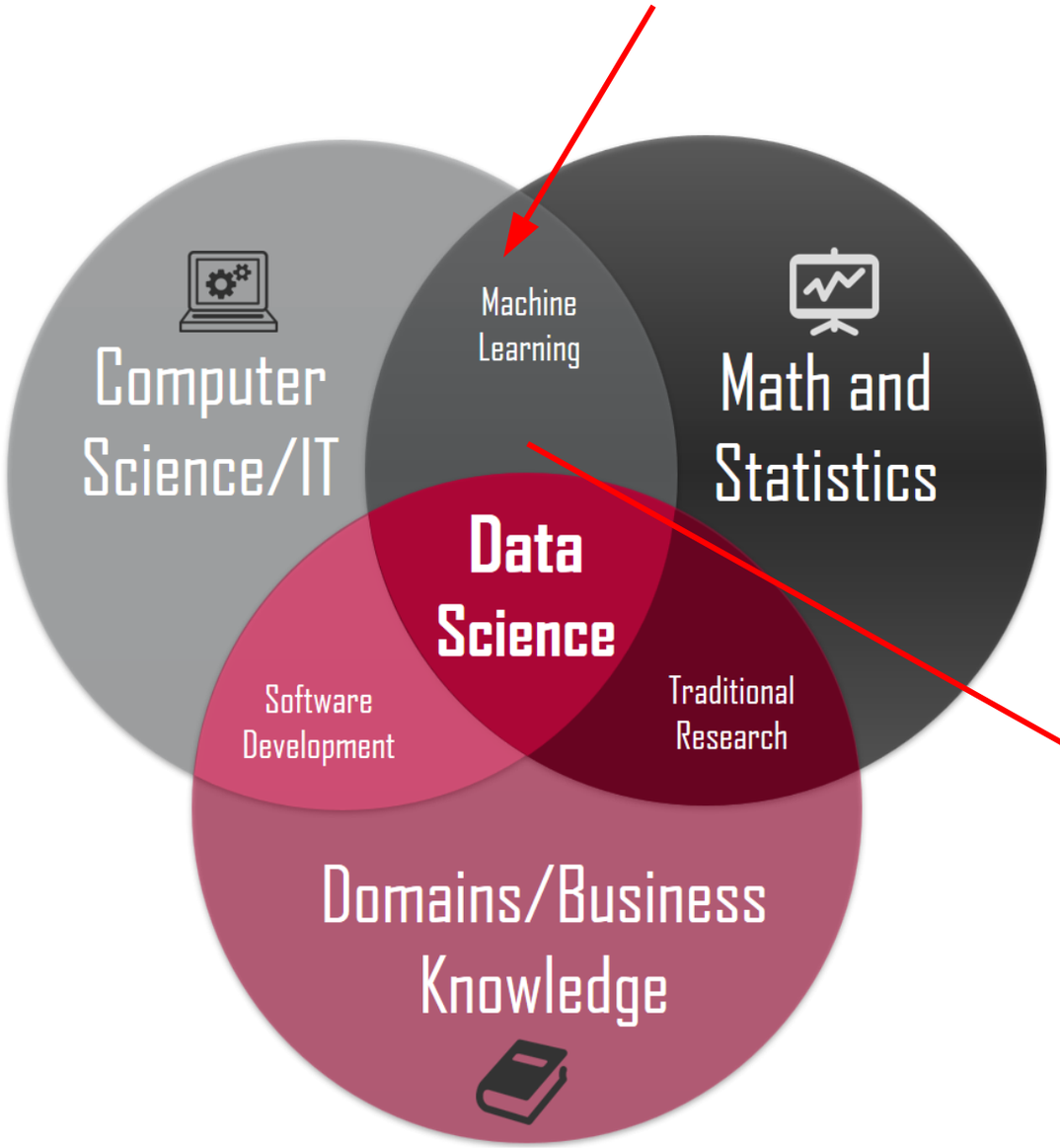
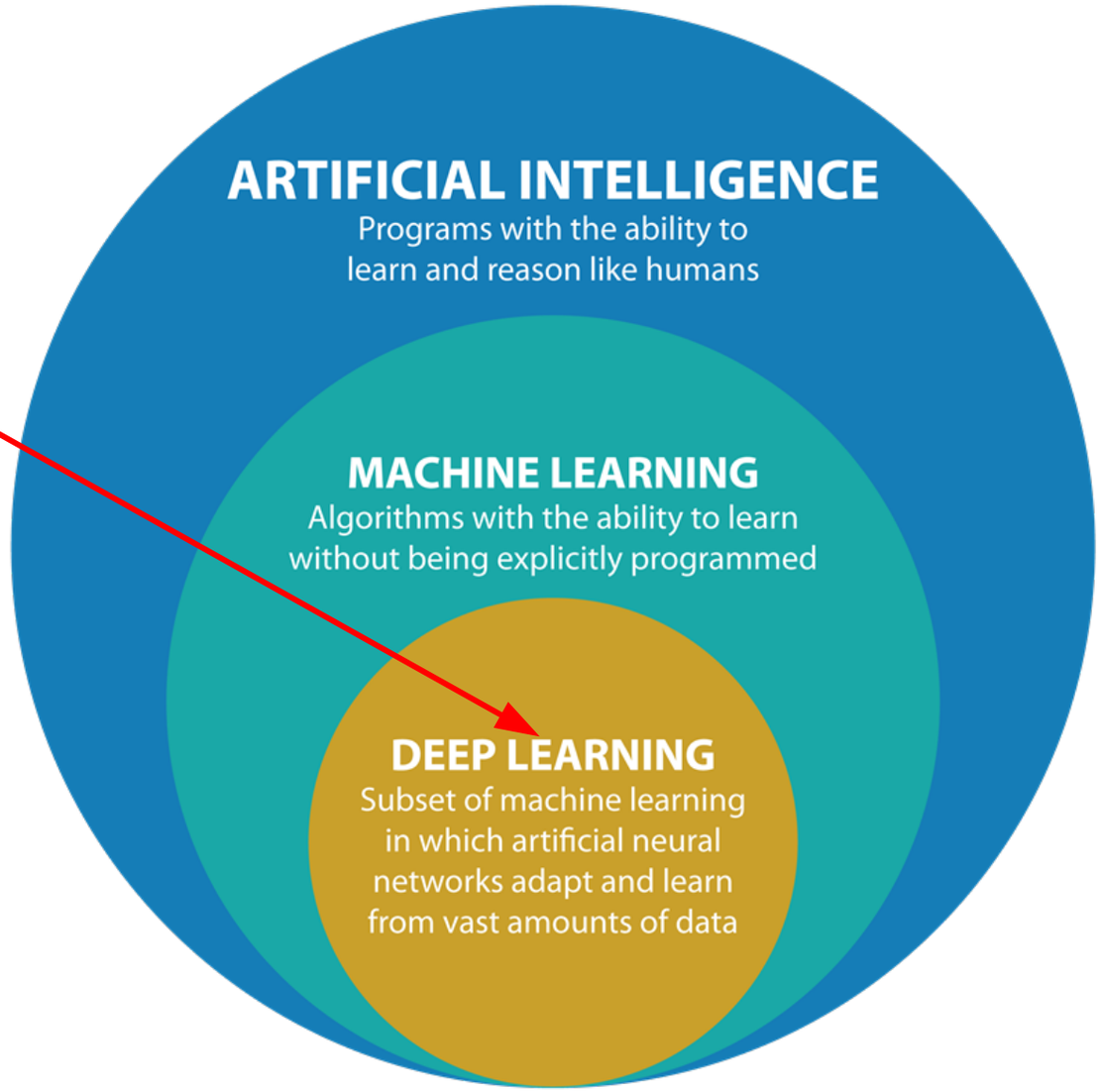
Modelando problemas com dados sequenciais e soluções baseadas em deep learning

Prof. Dr. Dalcimar Casanova
http://www.dalcimar.com

Context


Machine Learning
Computer Science/IT
Math and Statistics
Data Science
Software Development
Traditional Research
Domains/Business Knowledge


ARTIFICIAL INTELLIGENCE
Programs with the ability to learn and reason like humans
MACHINE LEARNING
Algorithms with the ability to learn without being explicitly programmed
DEEP LEARNING
Subset of machine learning in which artificial neural networks adapt and learn from vast amounts of data

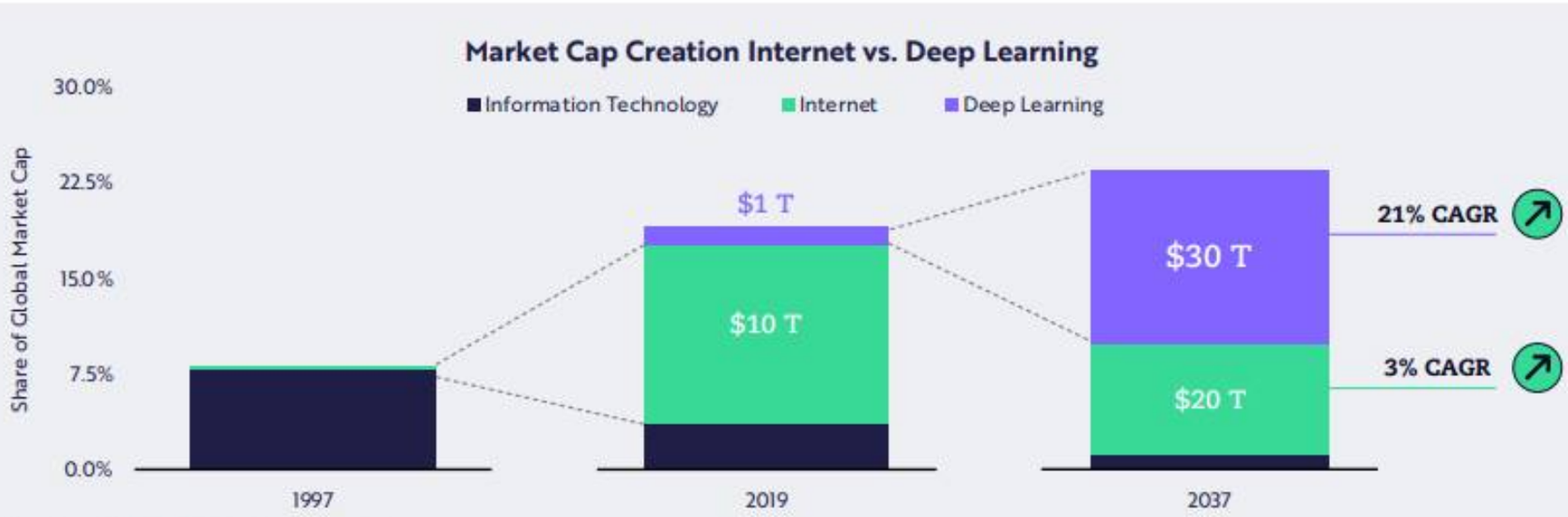
Market size of Deep Learning

BIG IDEAS 2020

Sizing the Opportunity

Deep learning could create more economic value than the Internet. Within two decades the Internet added roughly $10 trillion to the global equity market cap. Since 2012, deep learning has created $1 trillion in market capitalization.

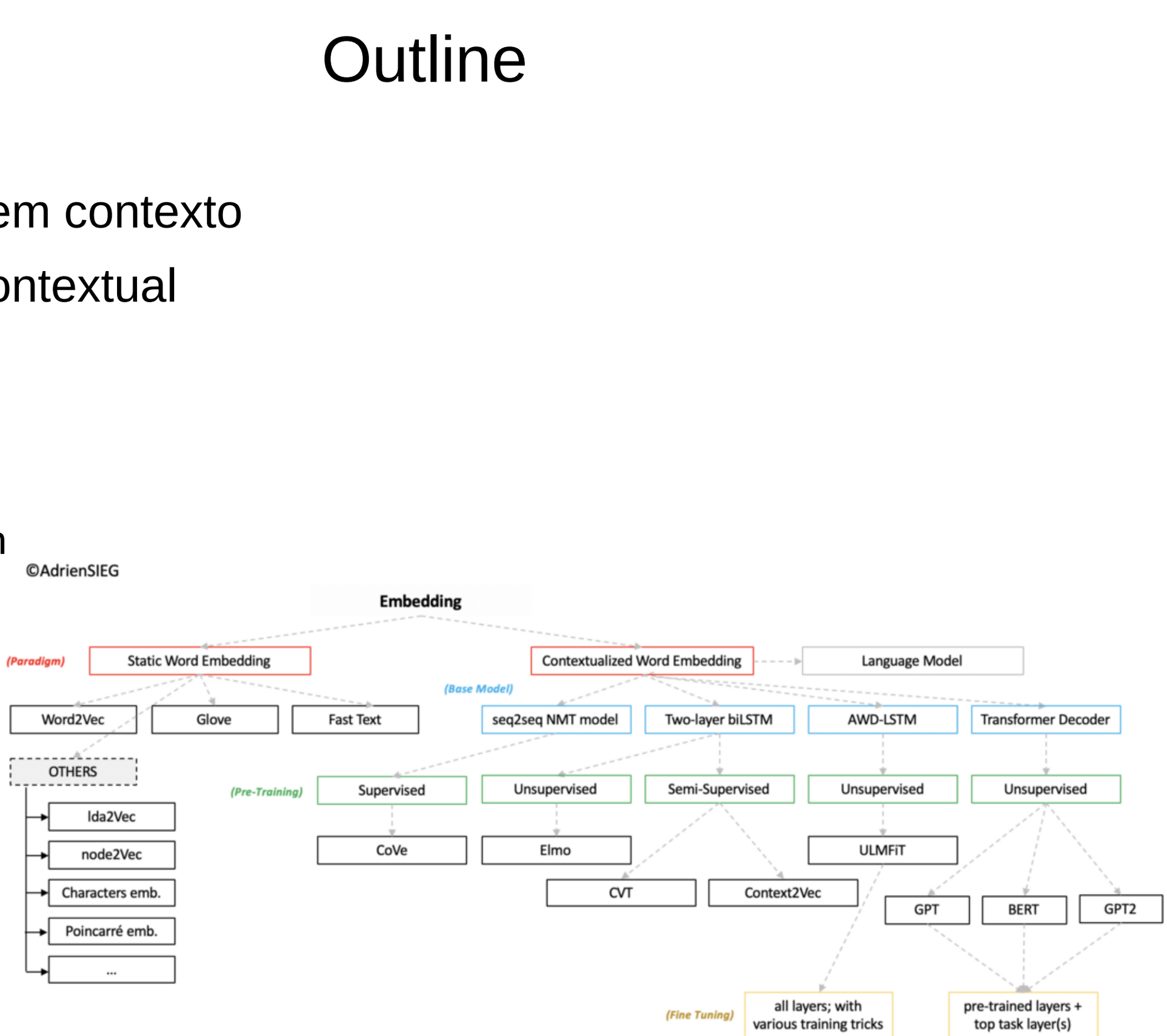
ARK believes it will add $30 trillion by 2037.



Forecasts are inherently limited and cannot be relied upon.
Source: ARK Investment Management LLC, 2019

Outline

- Análise sem contexto
- Análise contextual
  - RNN
  - LSTM
  - GRU
  - Attention
-



GP3 Examples

- https://twitter.com/pavtalk/status/1285410751092416513
- https://twitter.com/sharifshameem/status/1284815412949991425
- https://twitter.com/jsngr/status/1284511080715362304
- https://twitter.com/aquariusacquah/status/1285415144017797126
- https://twitter.com/nutanc/status/1285602813385605120
- https://twitter.com/nutanc/status/1285128268392235010
- https://twitter.com/nutanc/status/1291768602404589569
- https://twitter.com/DonCubed/status/1284908940149395456
- https://twitter.com/nutanc/status/1291364553024892928

A Classic Approach for Text Classification: Bag-of-Words Model

"Raw" training dataset

$\mathbf{x}^{[1]}$ = "The sun is shining"

$\mathbf{x}^{[2]}$ = "The weather is sweet"

$\mathbf{x}^{[3]}$ = "The sun is shining, the weather is sweet, and one and one is two"

x[1] = [6 4 1 3]
x[2] = [6 8 1 5]
x[3] = [6 4 1 3 6 8 1 5 0 2 0 2 1 7]

vocabulary = {
'and': 0,
'is': 1
'one': 2,
'shining': 3,
'sun': 4,
'sweet': 5,
'the': 6,
'two': 7,
'weather': 8,
}

Training set as design matrix

$$\mathbf{X} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 2 & 3 & 2 & 1 & 1 & 1 & 2 & 1 & 1 \end{bmatrix}$$

$\mathbf{y} = [0, 1, 0]$

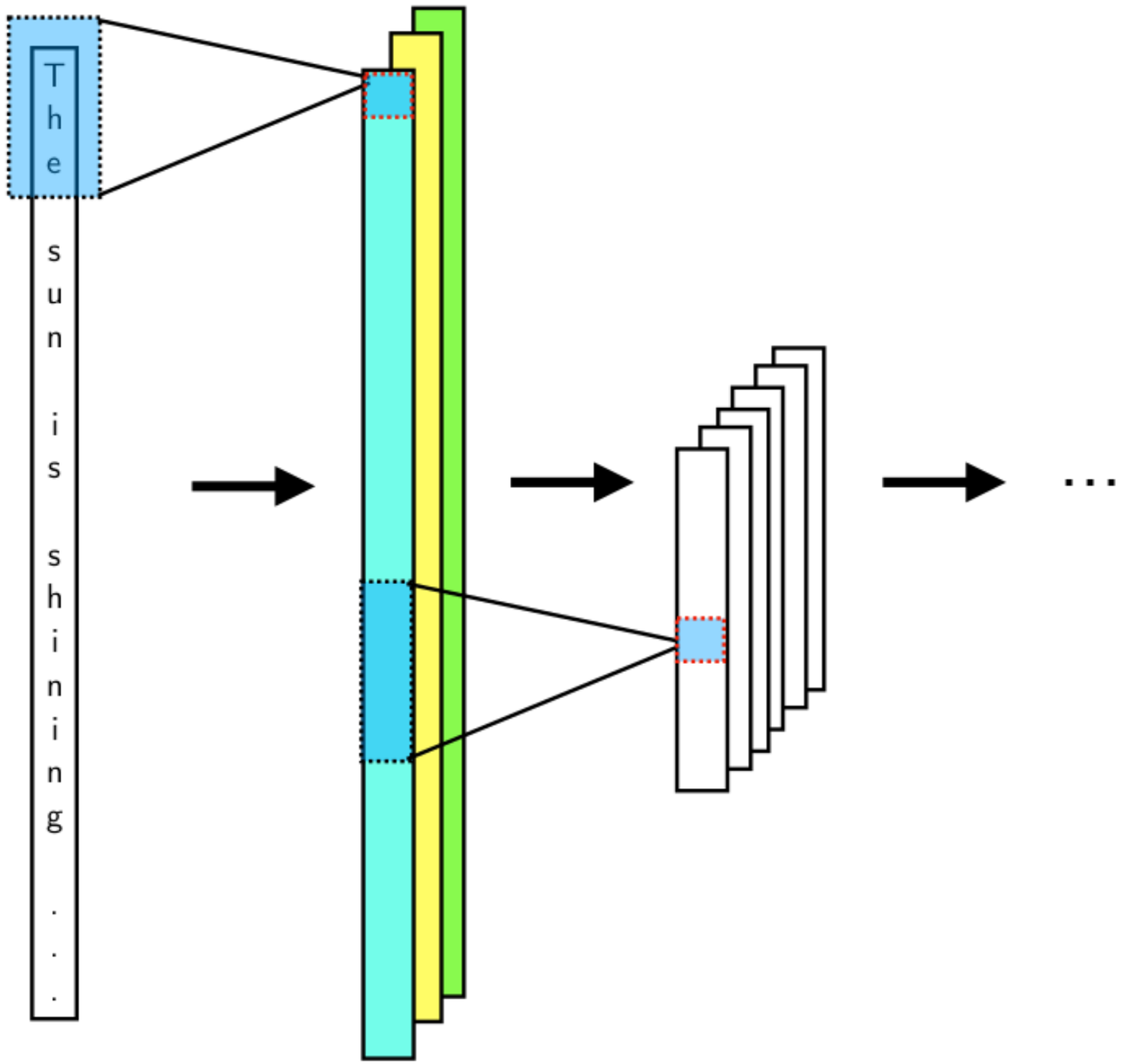
class labels

training

Classifier

(e.g., logistic regression, MLP, ...)

1D CNNs for text (and other sequence data)



Applications:
Working with Sequential Data

- Text classification
- Speech recognition (acoustic modeling)
- language translation
- ...

Stock market predictions

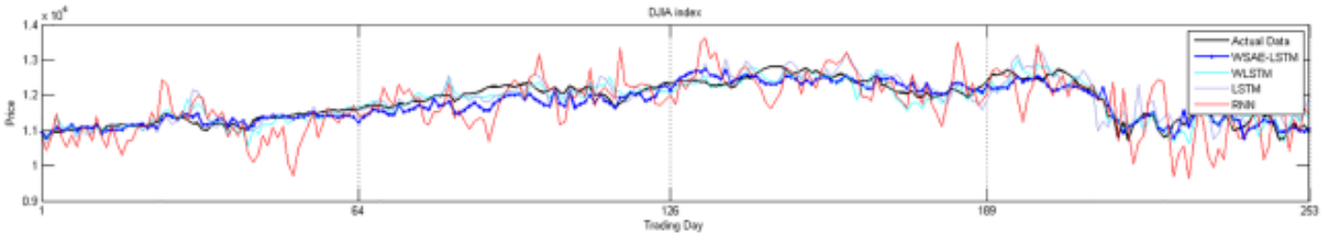
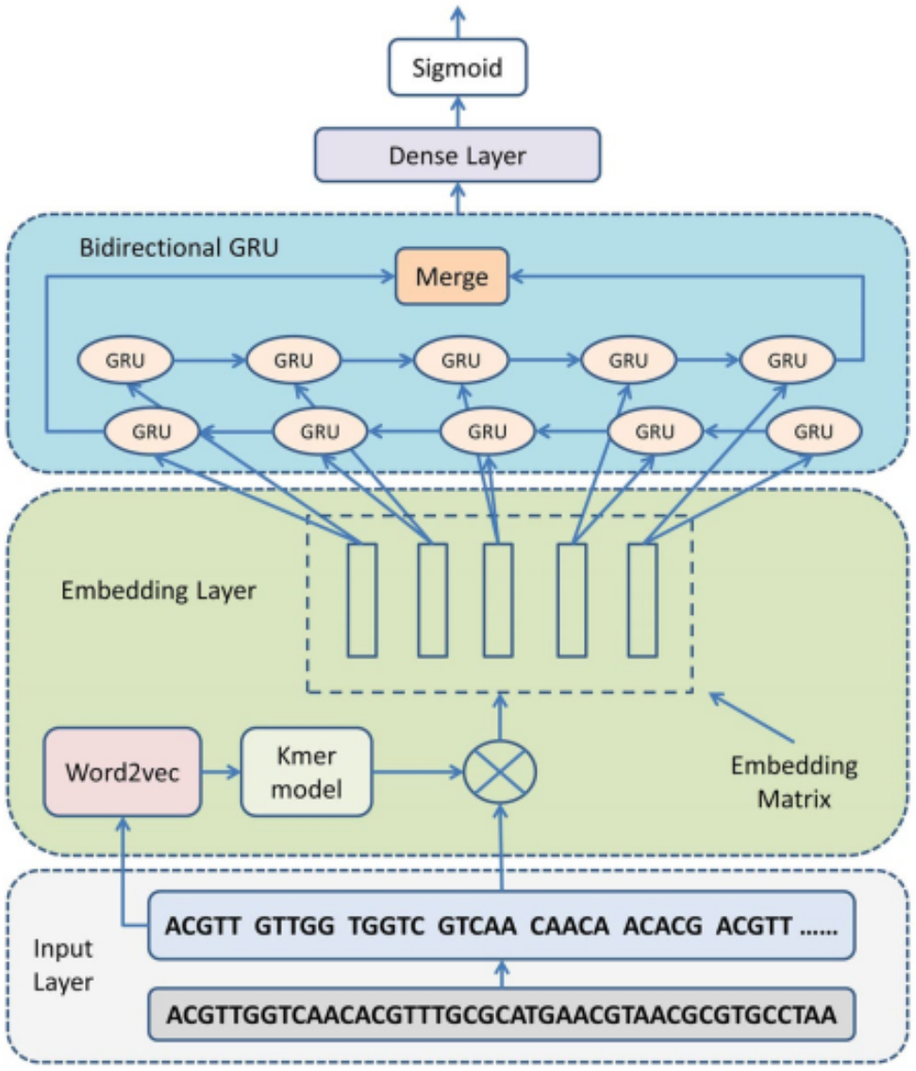


Fig 8. Displays the actual data and the predicted data from the four models for each stock index in Year 1 from 2010.10.01 to 2011.09.30.

https://doi.org/10.1371/journal.pone.0180944.g008

Bao, Wei, Jun Yue, and Yulei Rao. "A deep learning framework for financial time series using stacked autoencoders and long-short term memory." *PloS one* 12, no. 7 (2017): e0180944.



Shen, Zhen, Wenzheng Bao, and De-Shuang Huang. "Recurrent Neural Network for Predicting Transcription Factor Binding Sites." *Scientific reports* 8, no. 1 (2018): 15270.

DNA or (amino acid/protein) sequence modeling

Overview

Networks we used previously: also called feedforward neural networks

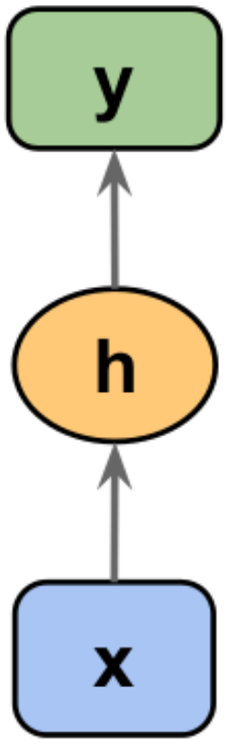
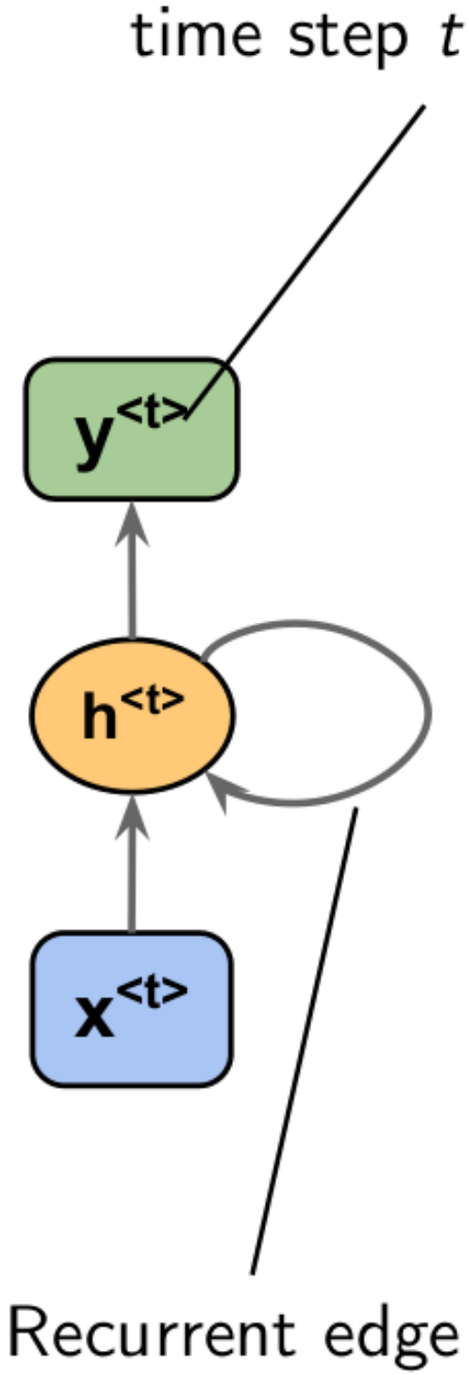


Figure: Sebastian Raschka, Vahid Mirjalili. *Python Machine Learning. 3rd Edition*. Birmingham, UK: Packt Publishing, 2019

Recurrent Neural Network (RNN)



Weight matrices in a single-hidden layer RNN

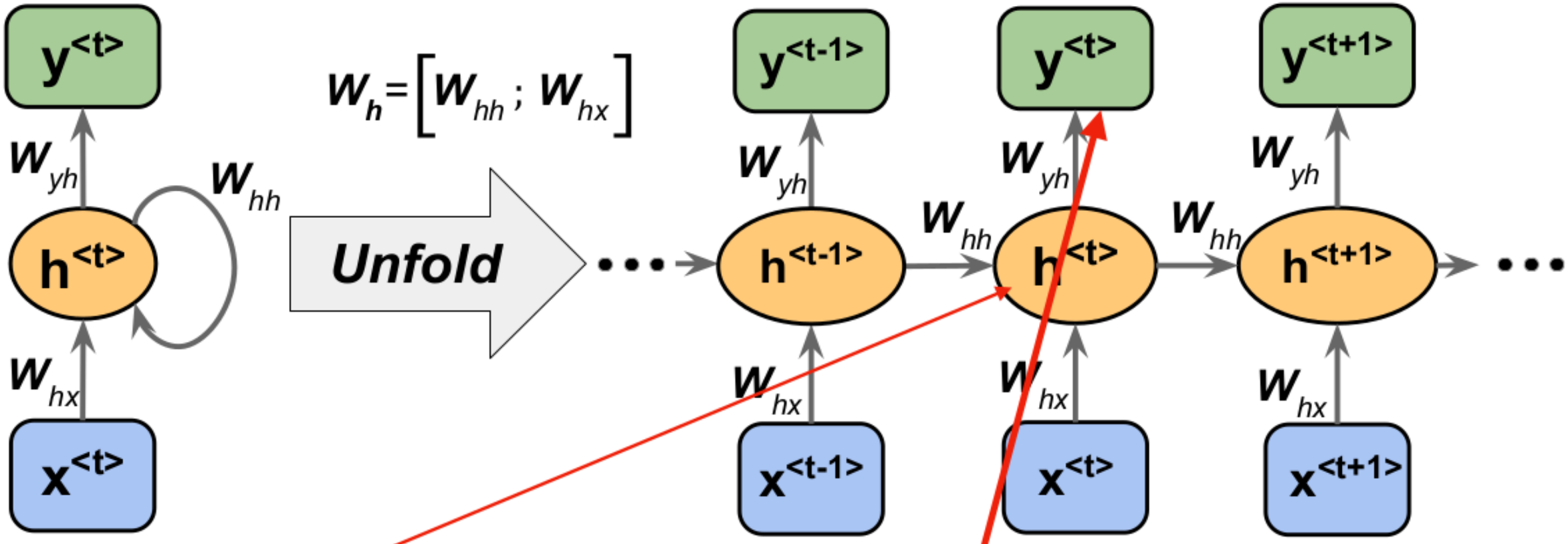


Figure: Sebastian Raschka, Vahid Mirjalili. *Python Machine Learning. 3rd Edition*. Birmingham, UK: Packt Publishing, 2019

Net input:

$$\mathbf{z}_h^{\langle t \rangle} = \mathbf{W}_{hx}\mathbf{x}^{\langle t \rangle} + \mathbf{W}_{hh}\mathbf{h}^{\langle t-1 \rangle} + \mathbf{b}_h$$

Activation:

$$\mathbf{h}^{\langle t \rangle} = \sigma_h\left(\mathbf{z}_h^{\langle t \rangle}\right)$$

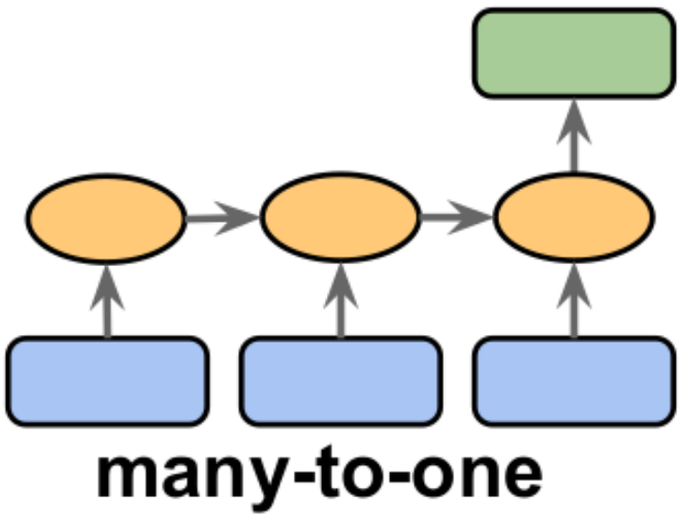
Net input:

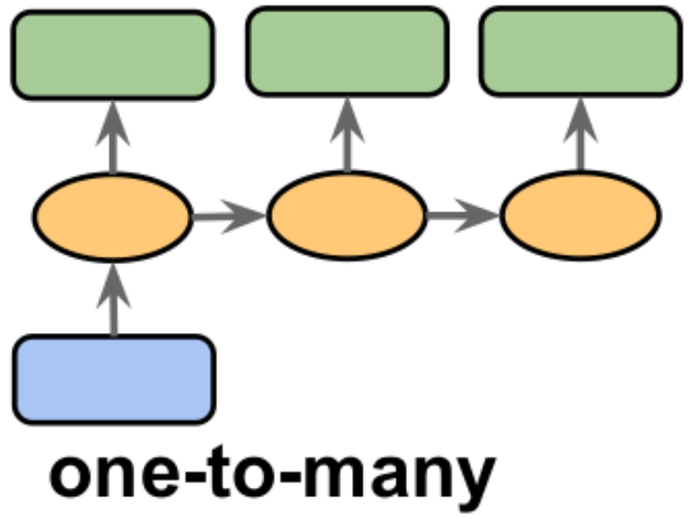
$$\mathbf{z}_y^{\langle t \rangle} = \mathbf{W}_{yh}\mathbf{h}^{\langle t \rangle} + \mathbf{b}_y$$

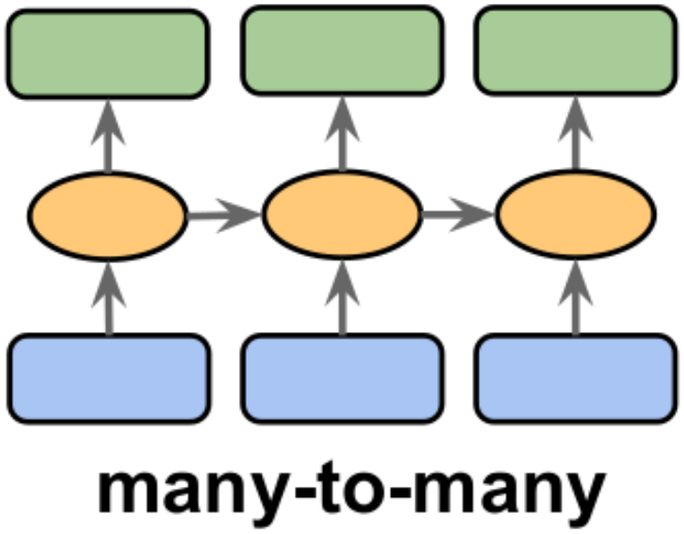
Output:

$$\mathbf{y}^{\langle t \rangle} = \sigma_y\left(\mathbf{z}_y^{\langle t \rangle}\right)$$

Different Types of Sequence Modeling Tasks







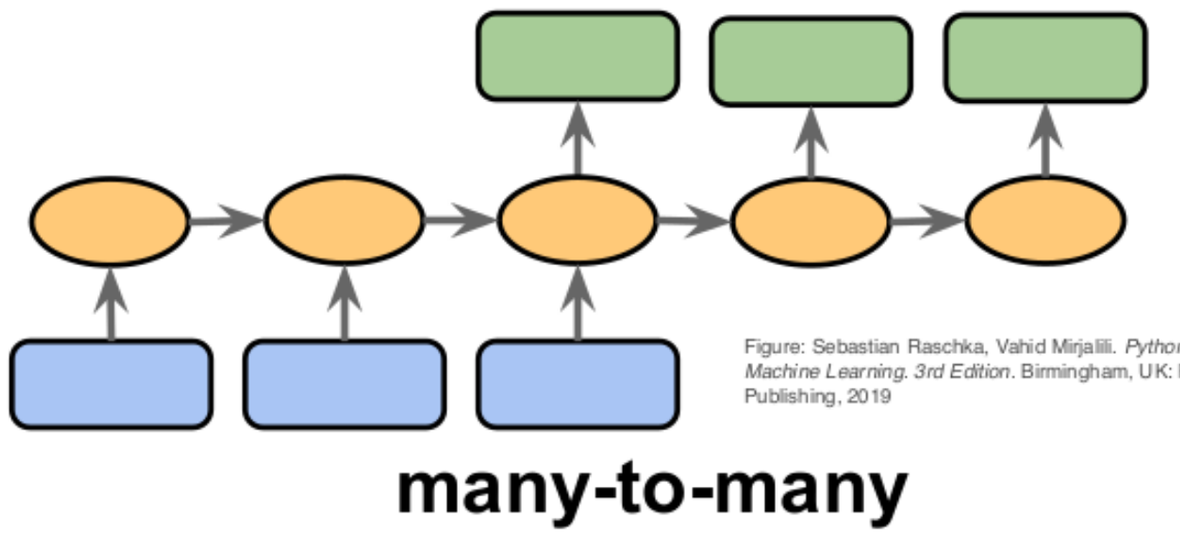
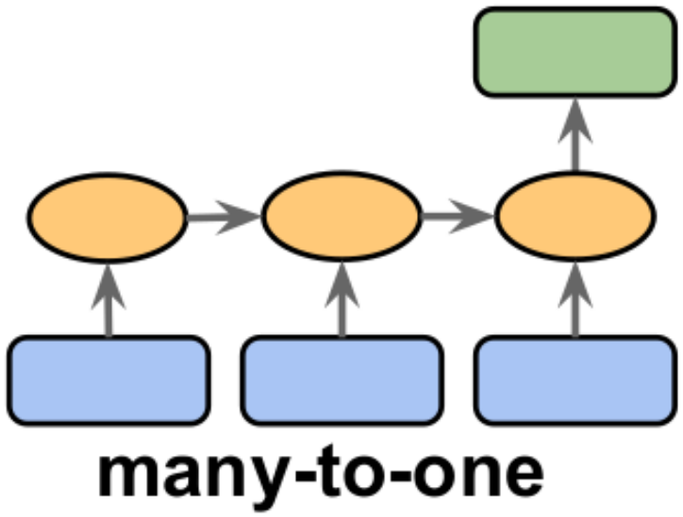


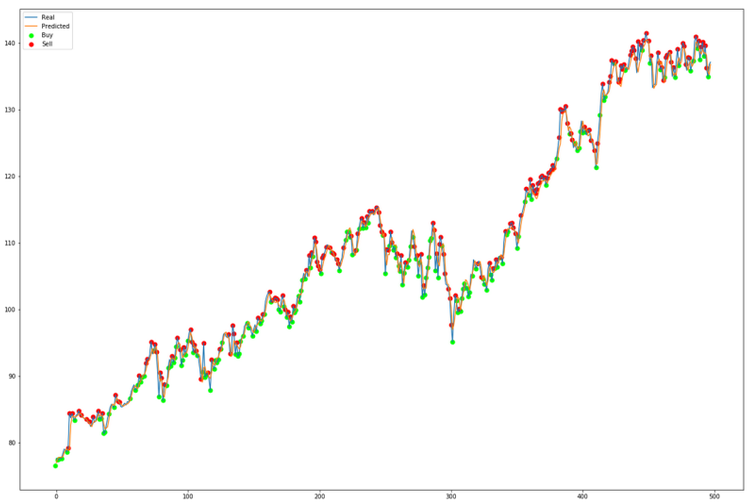
Figure: Sebastian Raschka, Vahid Mirjalili. *Python Machine Learning. 3rd Edition*. Birmingham, UK: Packt Publishing, 2019

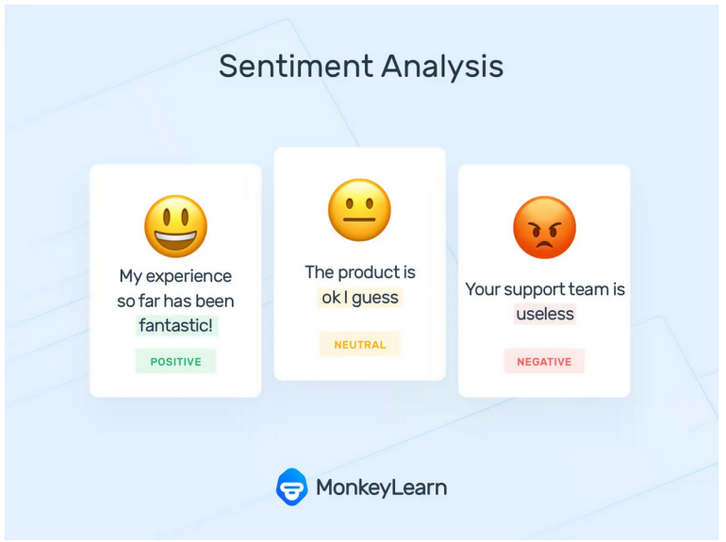
Figure based on:

The Unreasonable Effectiveness of Recurrent Neural Networks by Andrej Karpathy (http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

Different Types of Sequence Modeling Tasks

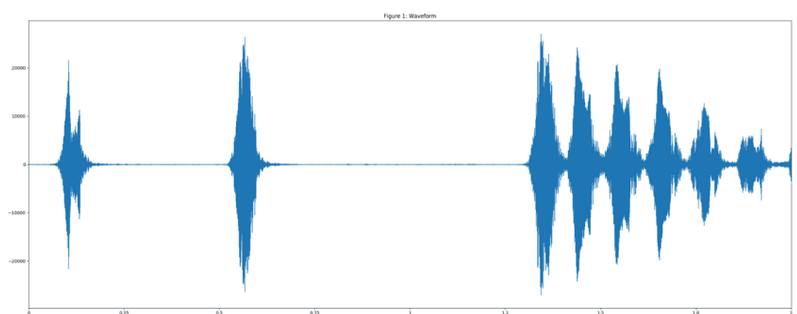




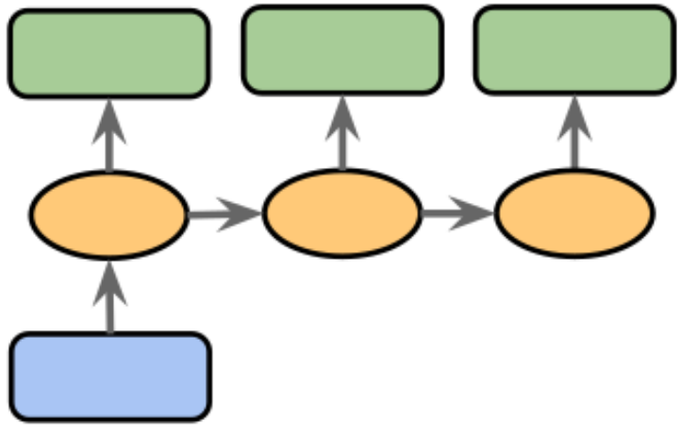


Many-to-one: The input data is a sequence, but the output is a fixed-size vector, not a sequence.

Ex.: sentiment analysis, the input is some text, and the output is a class label.



Different Types of Sequence Modeling Tasks



One-to-many: Input data is in a standard format (not a sequence), the output is a sequence.

Ex.: Image captioning, where the input is an image, the output is a text description of that image

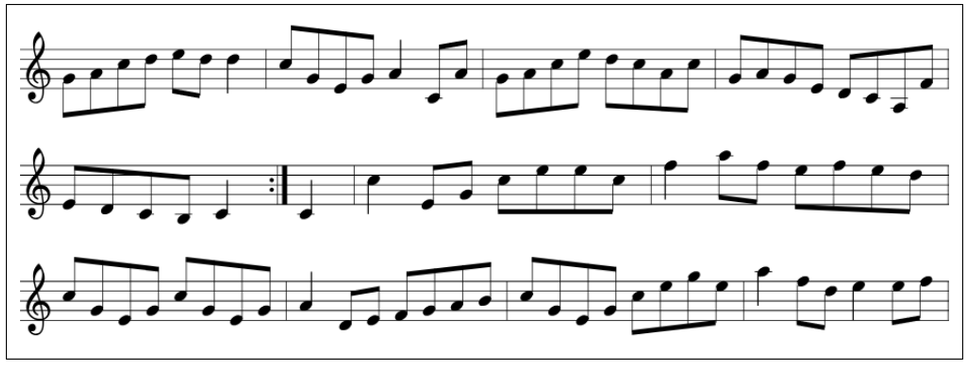
"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

https://www.youtube.com/watch?v=Xq9FJfP3zTI

https://www.youtube.com/watch?v=RByCiOLlxug

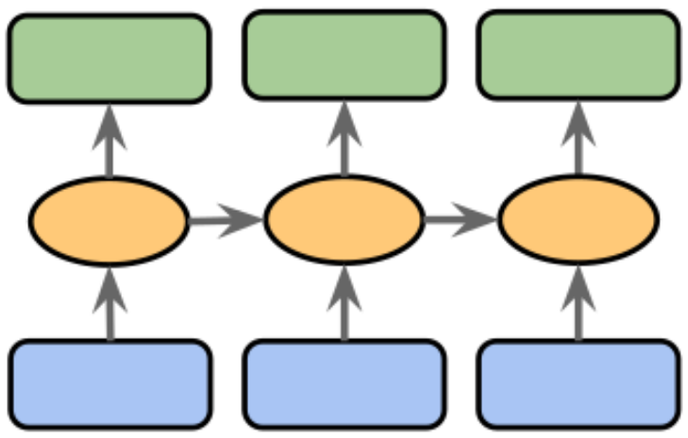


Different Types of Sequence Modeling Tasks

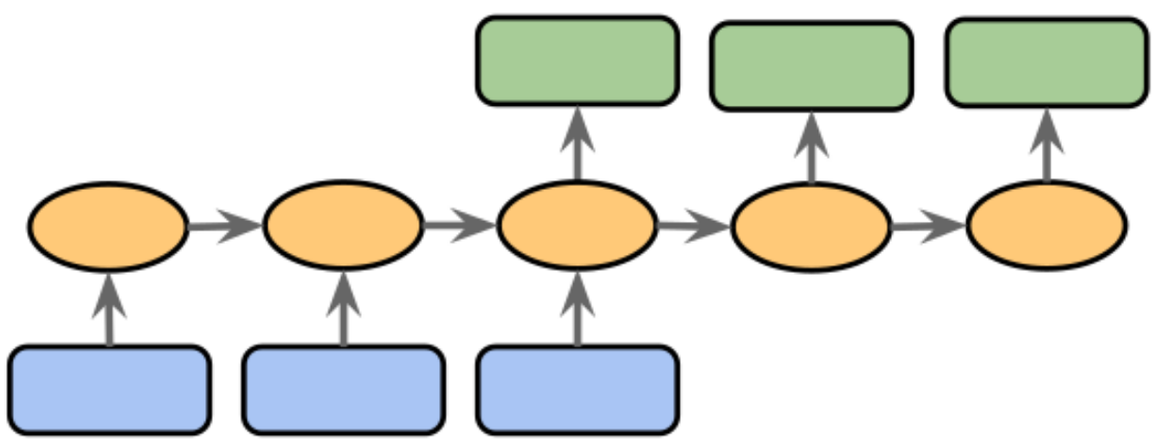
Many-to-many: Both inputs and outputs are sequences. Can be direct or delayed.

Ex.: Video-captioning, i.e., describing a sequence of images via text (direct).

Translating one language into another (delayed)

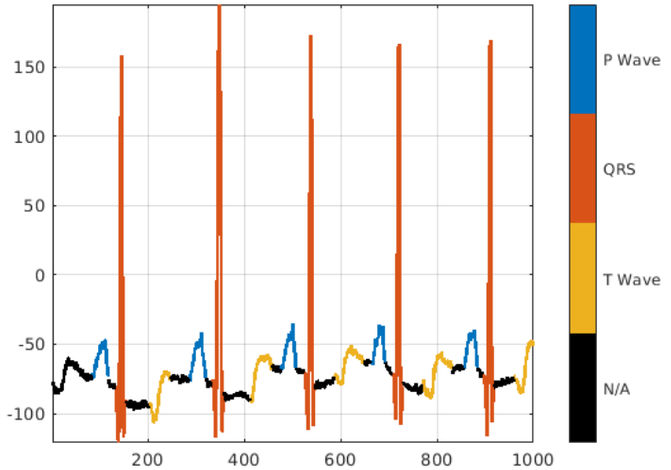


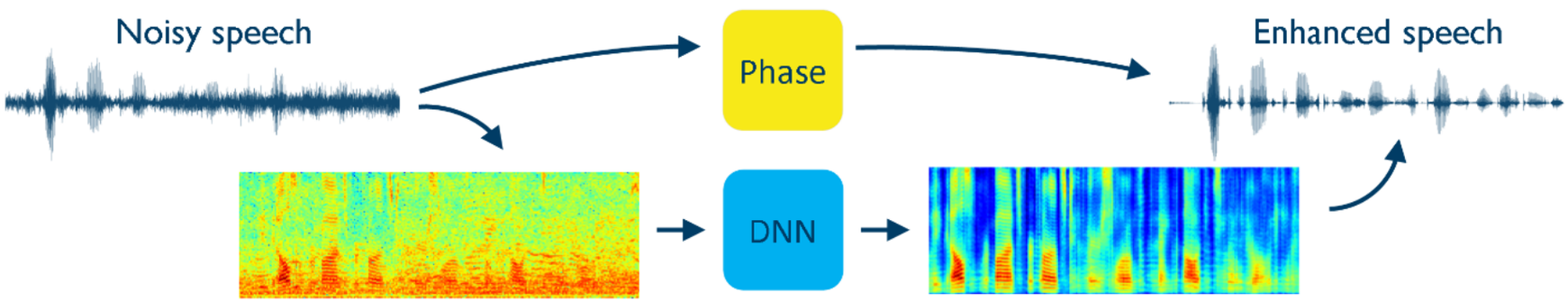
many-to-many

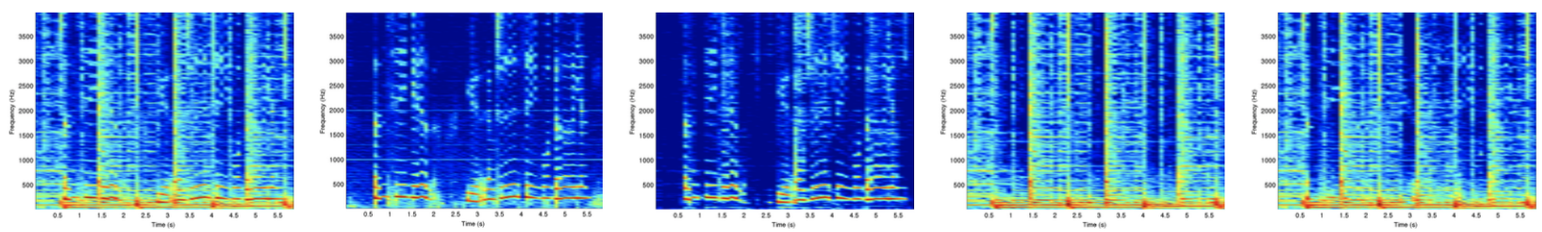


many-to-many

Many-to-many direct

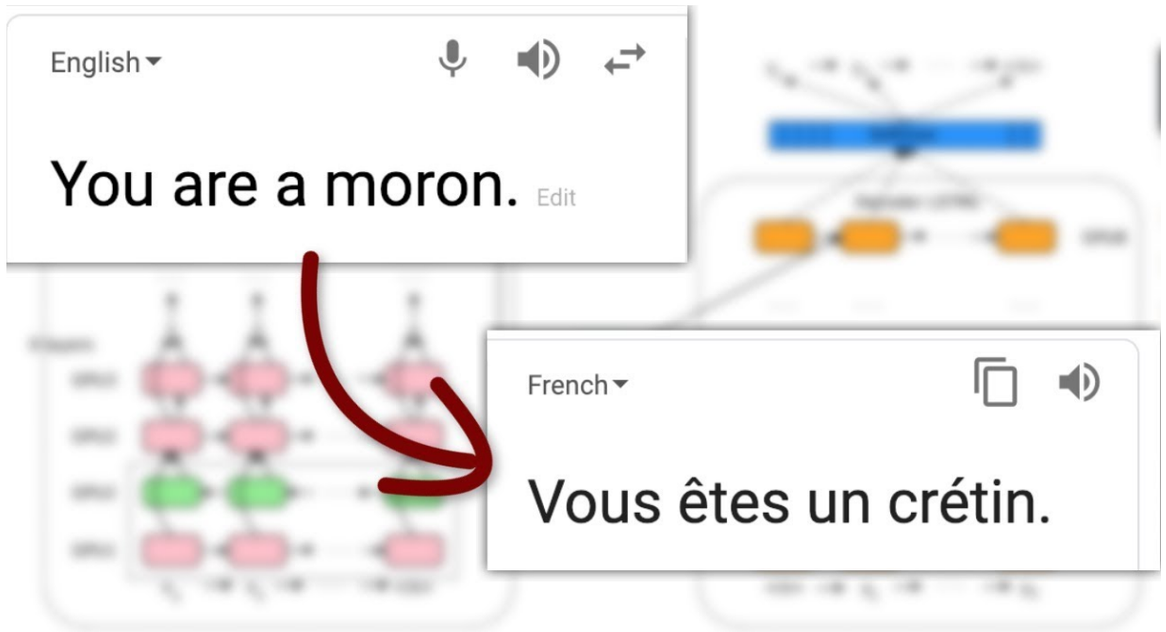


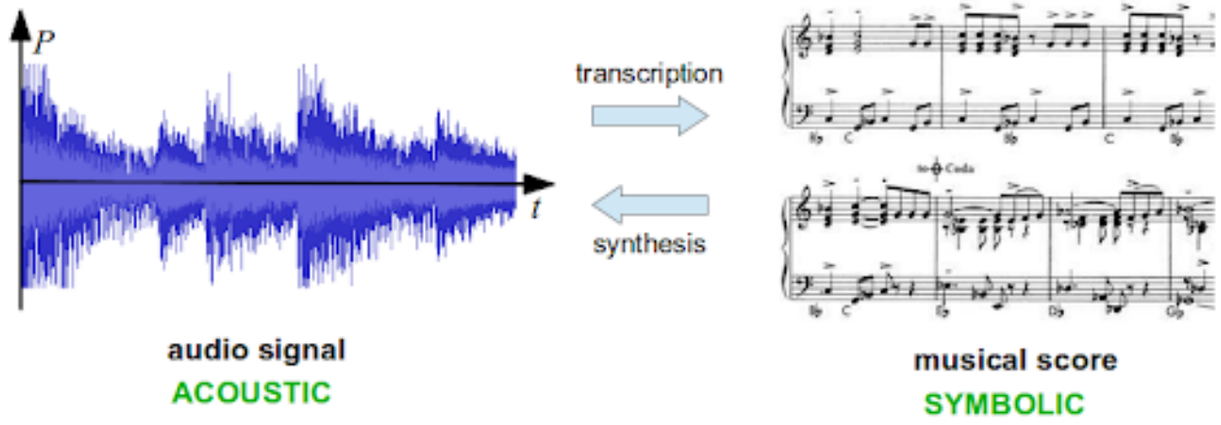


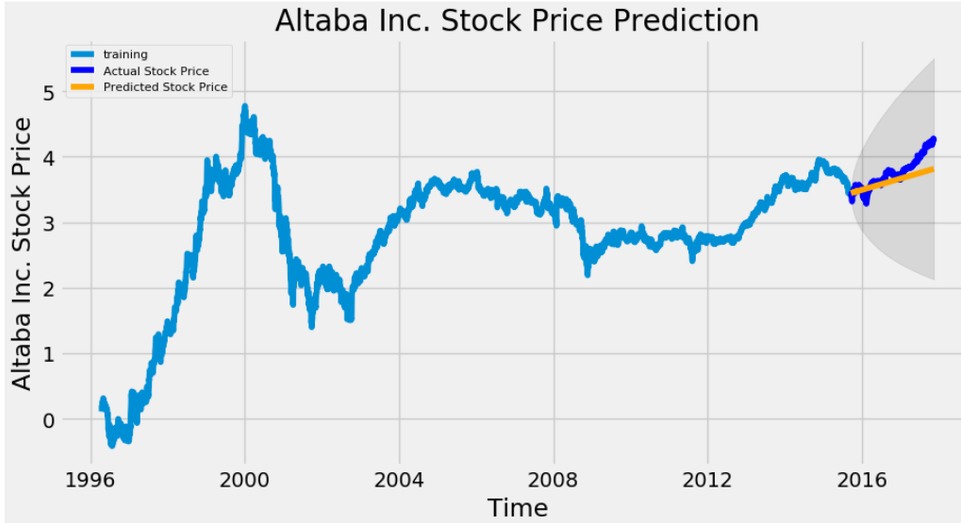


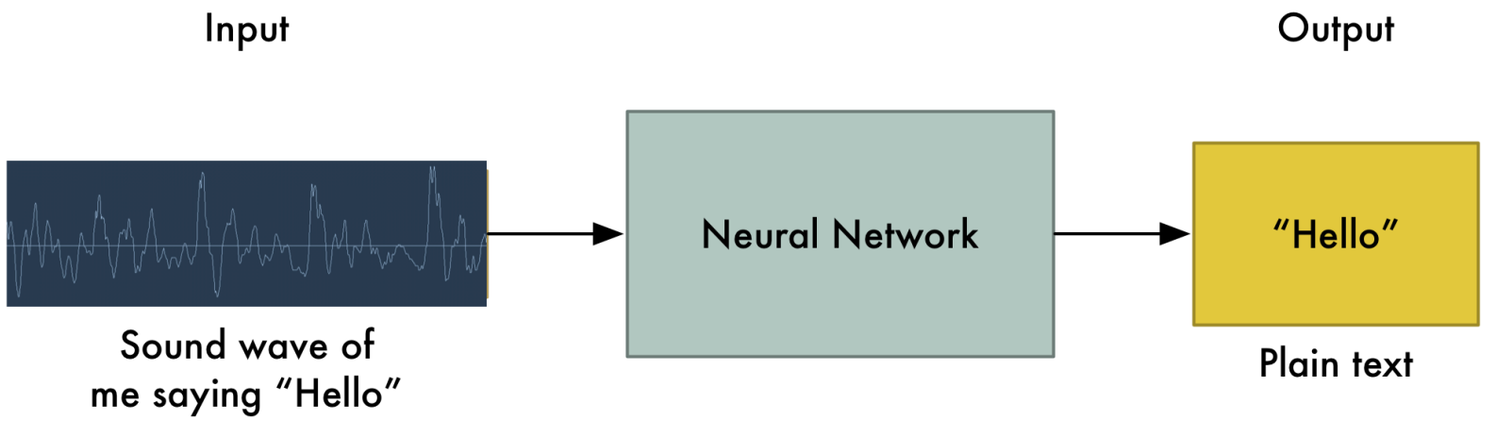
(a) Mixutre (b) Clean vocal (c) Recovered vocal (d) Clean music (e) Recovered music

Many-to-many delayed

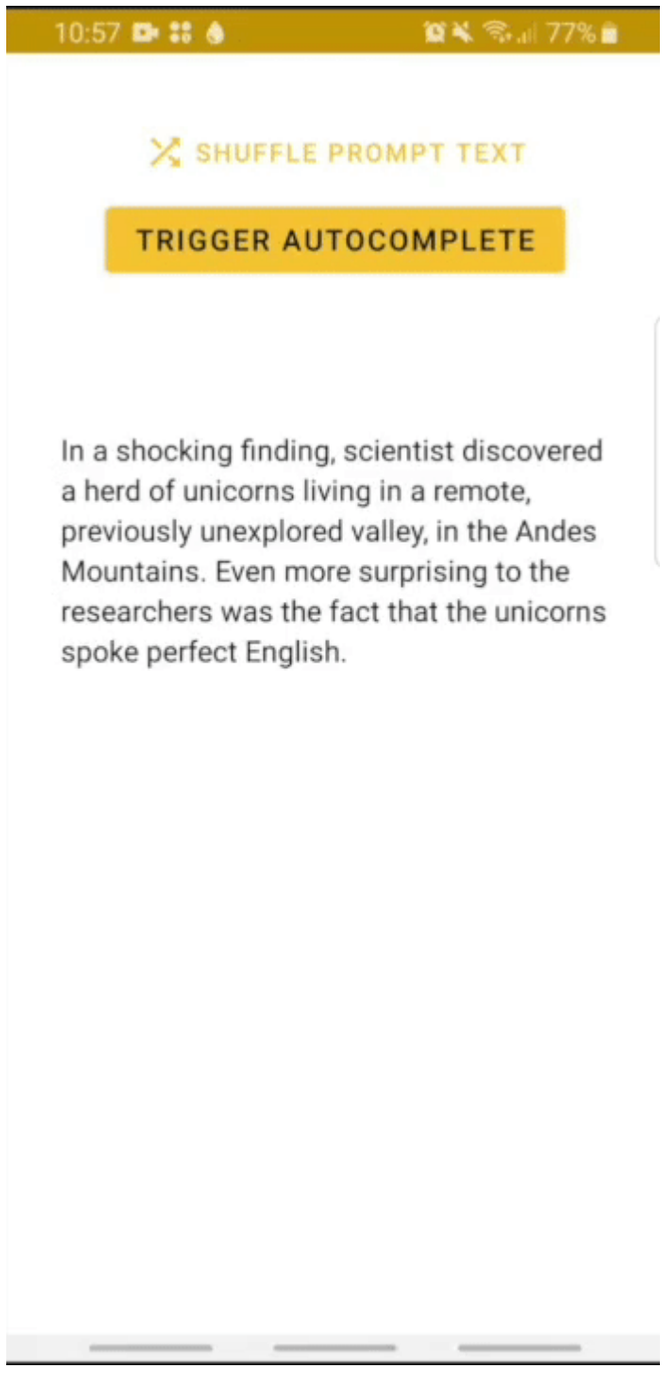


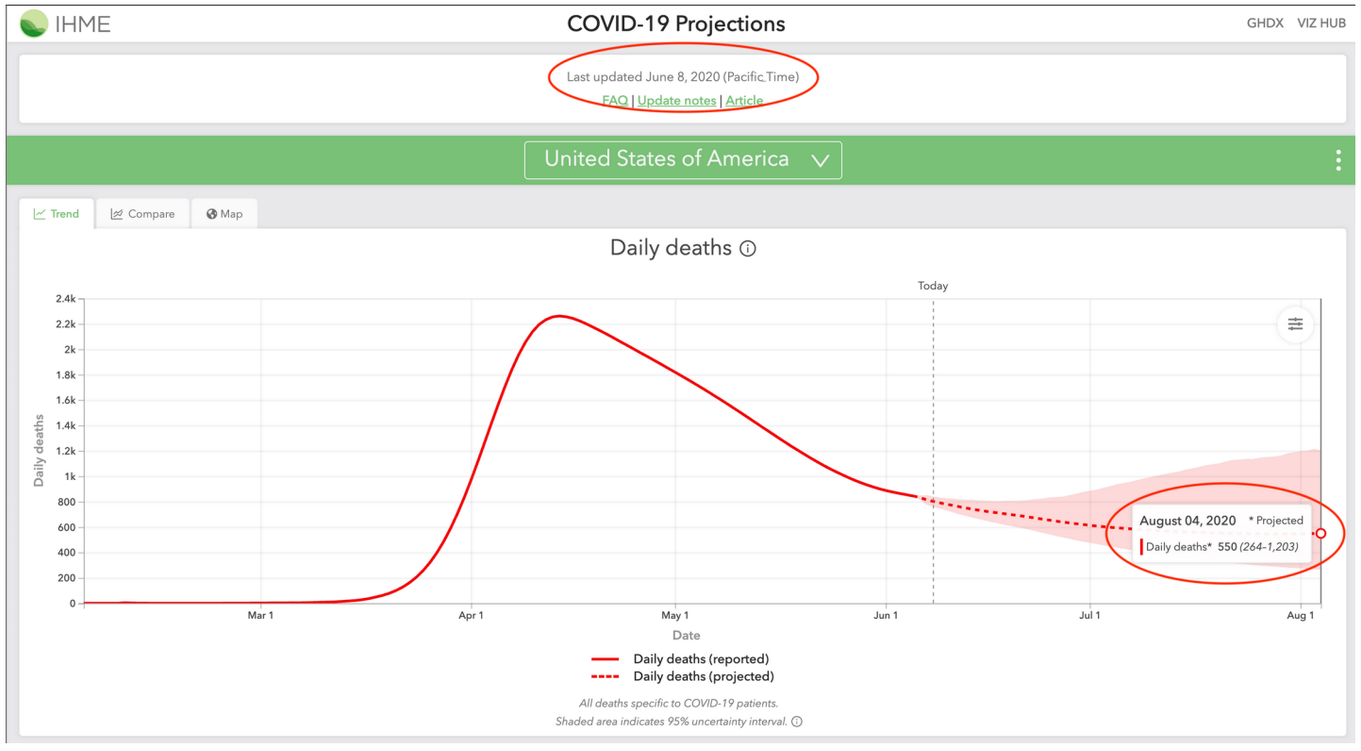






Many-to-many delayed





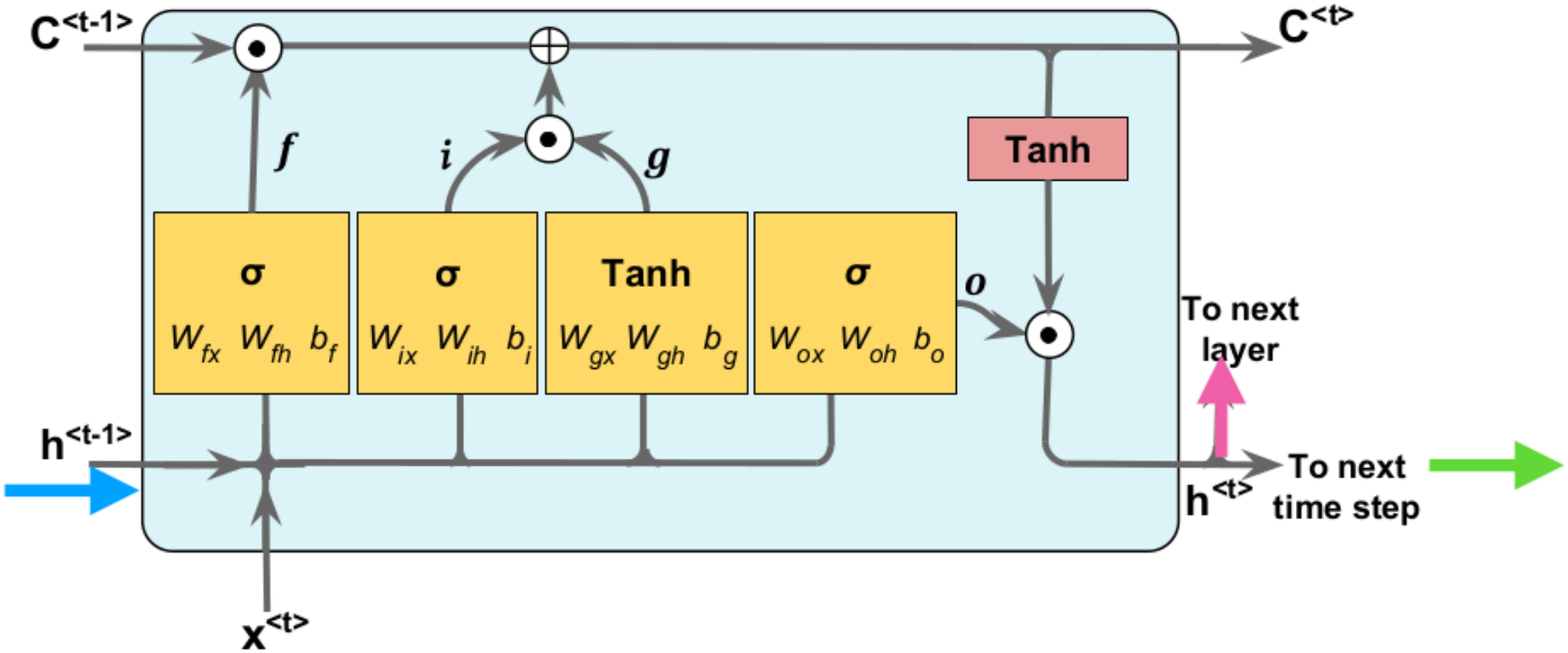
Solutions to the vanishing/exploding gradient problems

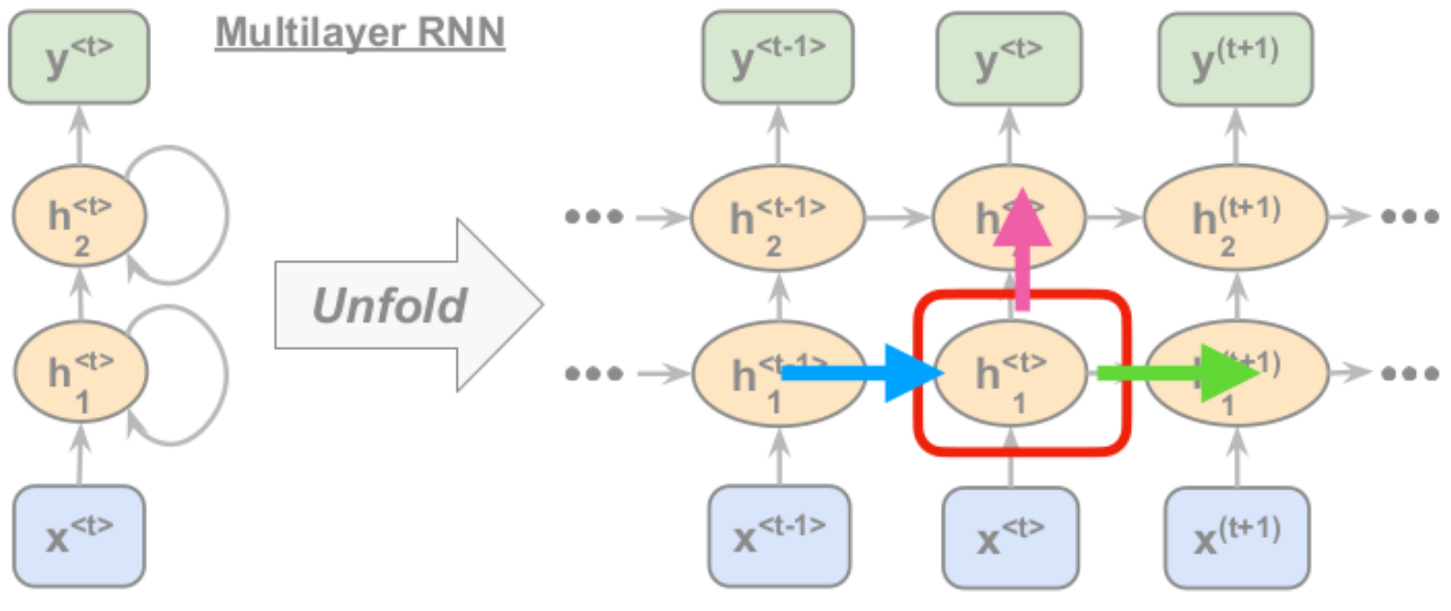
1) Gradient Clipping: set a max value for gradients if they grow to large (solves only exploding gradient problem)

2) Truncated backpropagation through time (TBPTT)
   - simply limits the number of time steps the signal can backpropagate each forward pass. E.g., even if the sequence has 100 elements/steps, we may only backpropagate through 20 or so

3) Long short-term memory (LSTM) -- uses a memory cell for modeling long-range dependencies and avoid vanishing gradient problems

Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9, no. 8 (1997): 1735-1780.


C<t-1>
C<t>
f
i
g
Tanh
σ
σ
Tanh
σ
o
W_fx W_fh b_f
W_ix W_ih b_i
W_gx W_gh b_g
W_ox W_oh b_o
To next layer
h<t-1>
To next time step
h<t>
x<t>

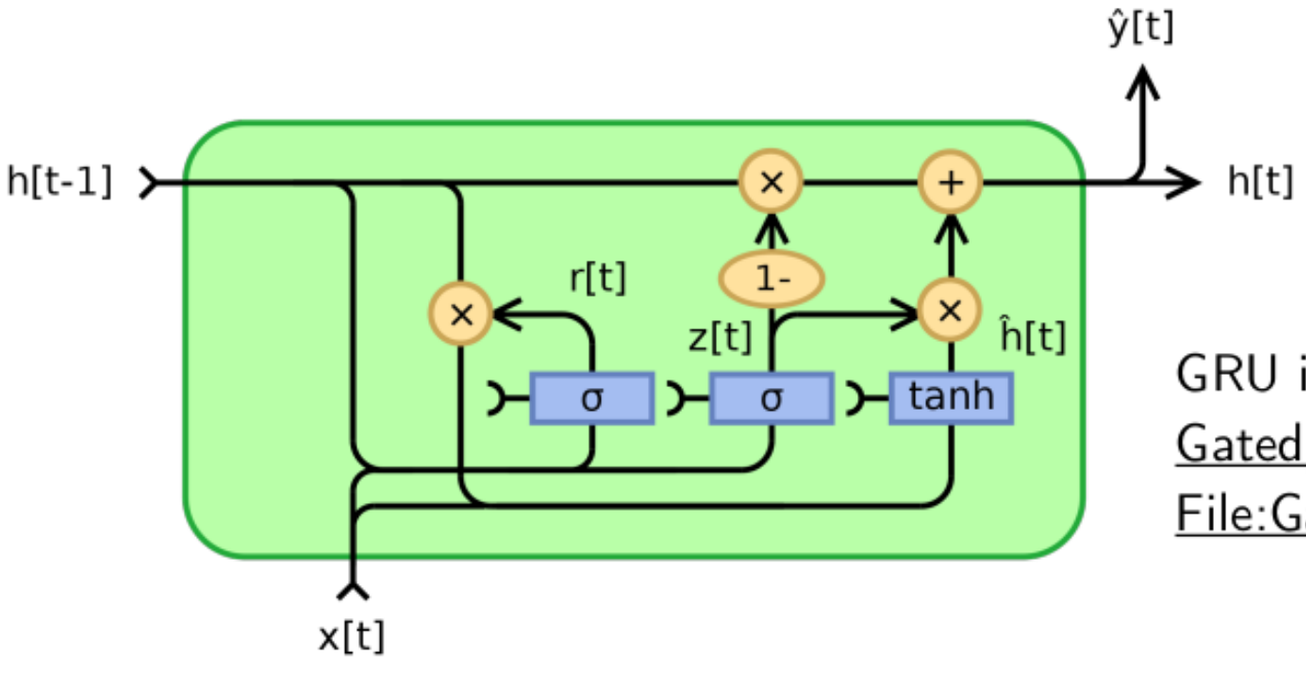

Multilayer RNN
y<t>
h2<t>
h1<t>
x<t>
Unfold
y<t-1>
y<t>
y(t+1)
h2<t-1>
h2(t+1)
h1<t-1>
h1<t>
h1(t+1)
x<t-1>
x<t>
x(t+1)

Long-short term memory (LSTM)

- Still popular and widely used today
- A recent, related approach is the Gated Recurrent Unit (GRU)

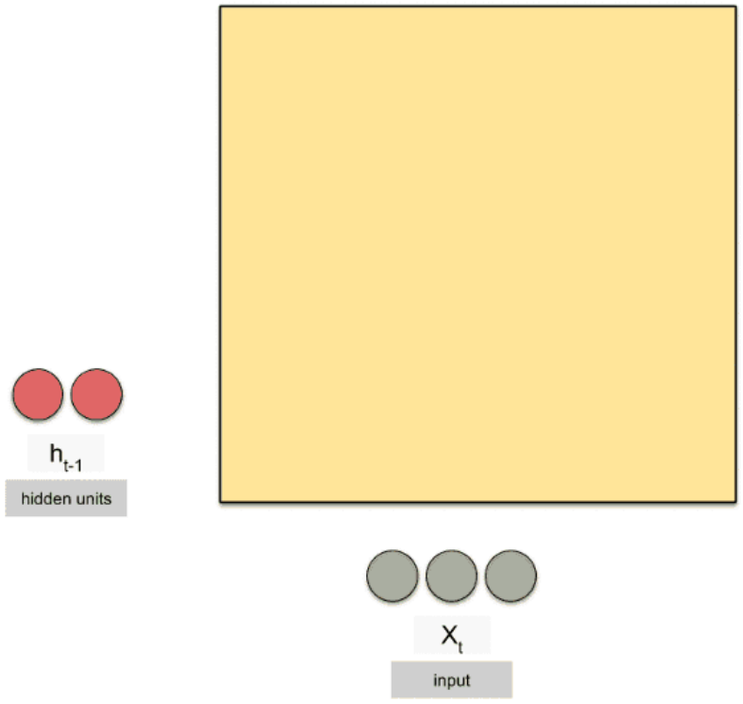
  Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." *arXiv preprint arXiv:1406.1078* (2014).
- Nice article exploring LSTMs and comparing them to GRUs

  Jozefowicz, Rafal, Wojciech Zaremba, and Ilya Sutskever. "An empirical exploration of recurrent network architectures." In *International Conference on Machine Learning*, pp. 2342-2350. 2015.



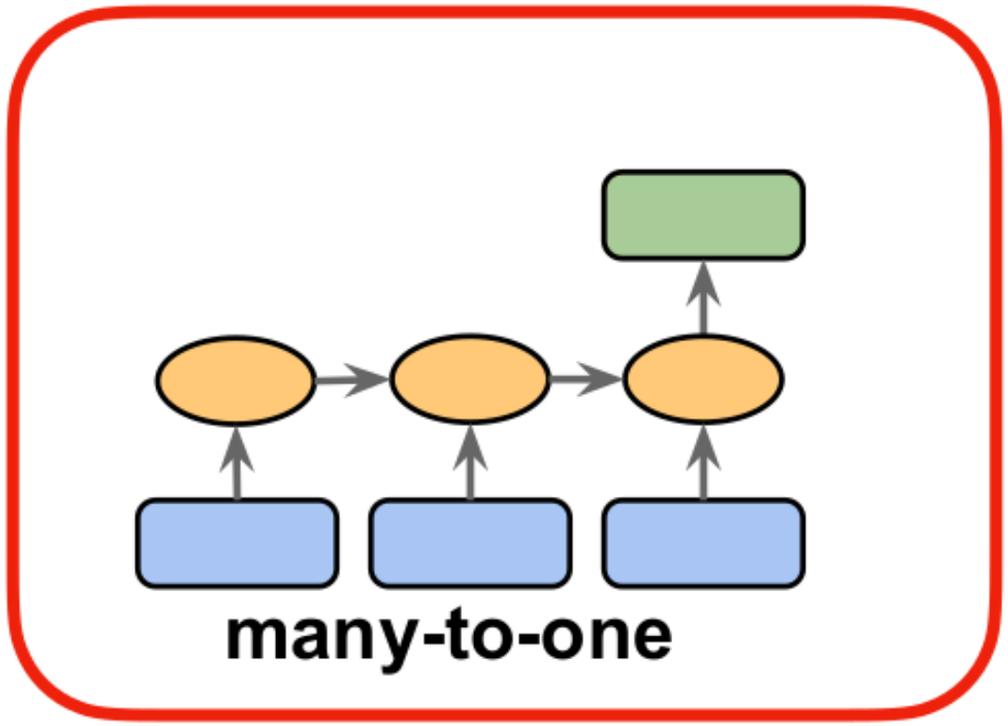
GRU image source: https://en.wikipedia.org/wiki/Gated_recurrent_unit#/media/File:Gated_Recurrent_Unit,_base_type.svg

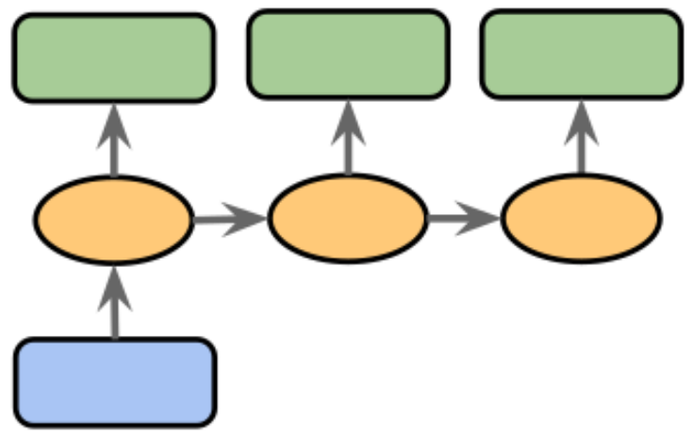
RNN x LSTM

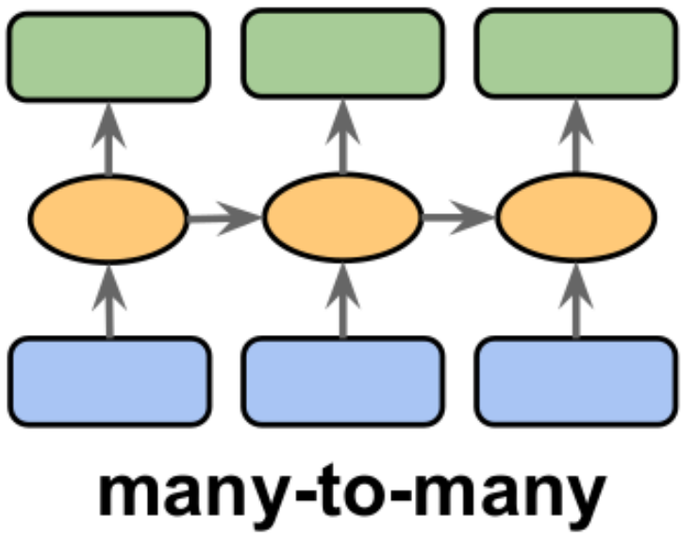

h_{t-1}
hidden units
X_t
input

C_{t-1}
cell state
h_{t-1}
hidden state / units
X_t
input

Different Types of Sequence Modeling Tasks







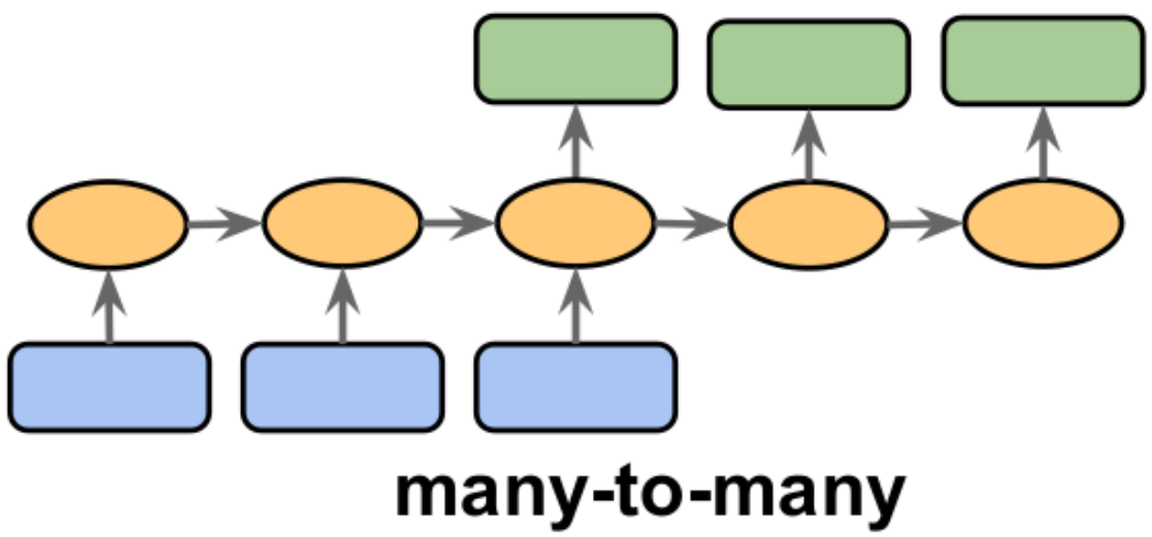


Figure based on:

The Unreasonable Effectiveness of Recurrent Neural Networks by Andrej Karpathy (http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

Many-to-One

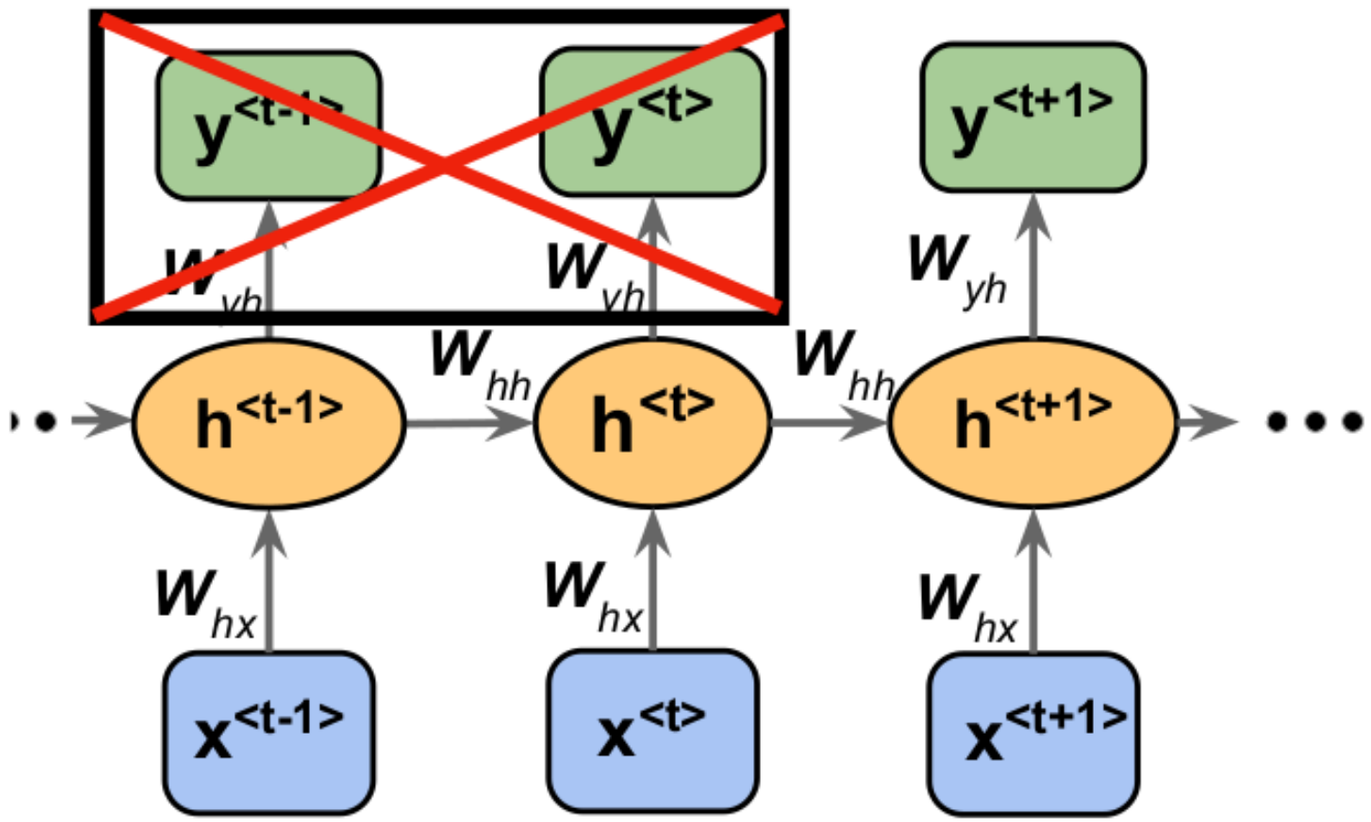
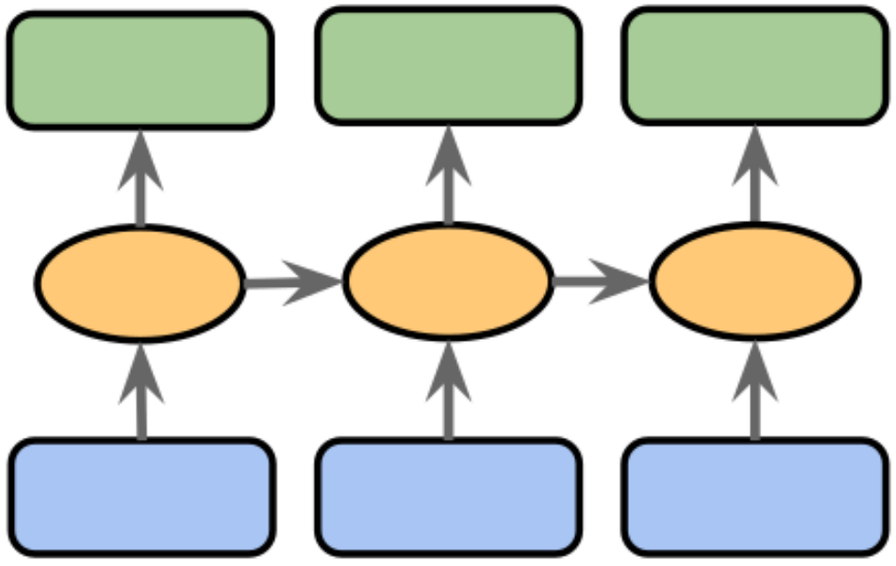
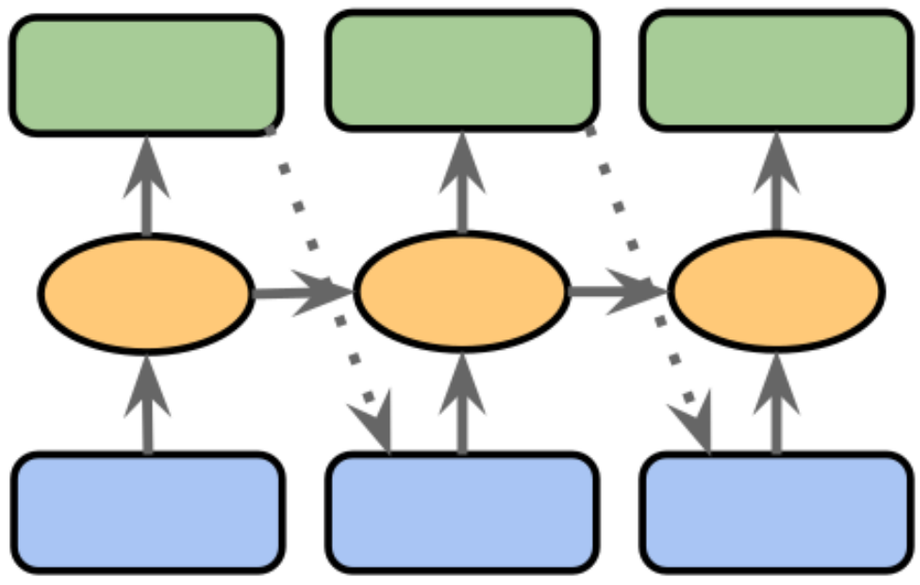


Figure: Sebastian Raschka, Vahid Mirjalili. *Python Machine Learning. 3rd Edition*. Birmingham, UK: Packt Publishing, 2019



many-to-many

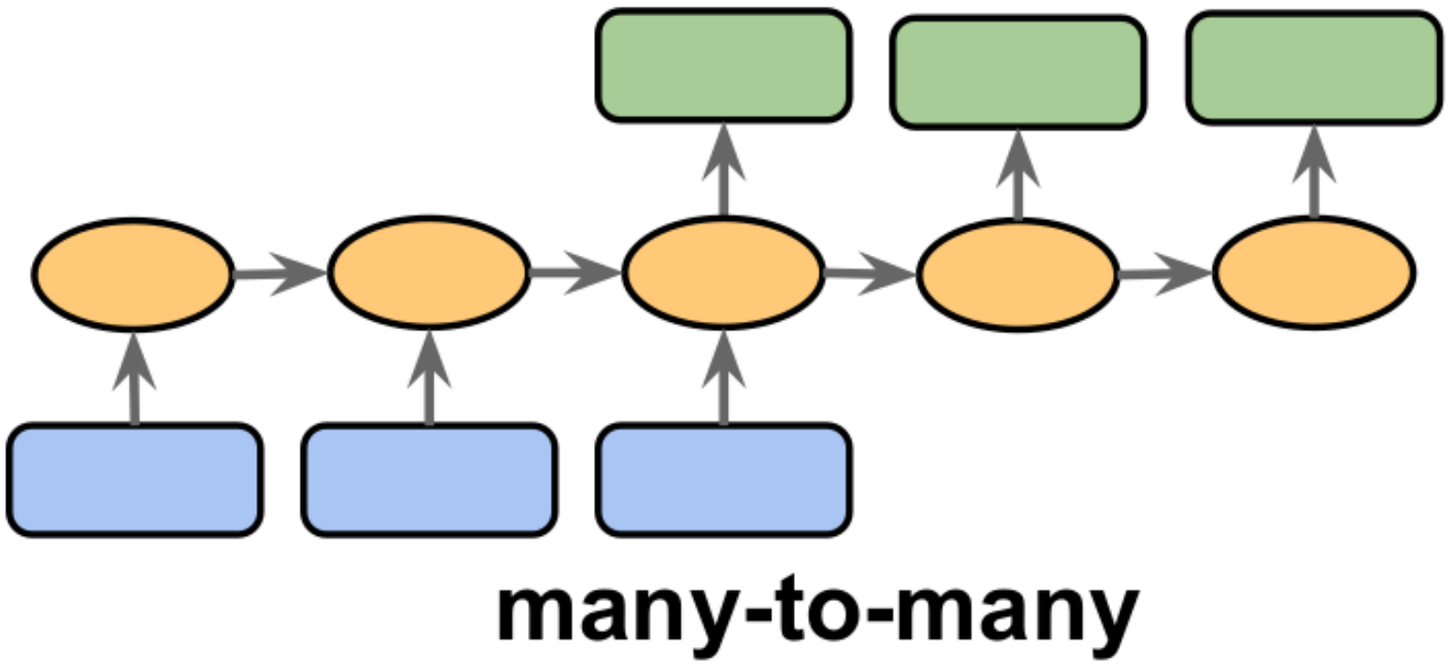
"training"



~~many~~-to-many

"one"

"generating new text"



seq2seq:
Chatbot
Machine Translation
Question Answering
Abstract Text Summarization
Text Generation

Translation with a Sequence to Sequence Network and Attention (English to French)

Figure based on:

The Unreasonable Effectiveness of Recurrent Neural Networks by Andrej Karpathy (http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

Attention Mechanism

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

Assign attention weight to each word to know how much "attention" the model should pay to each word (i.e., for each word, the network learns a "context")

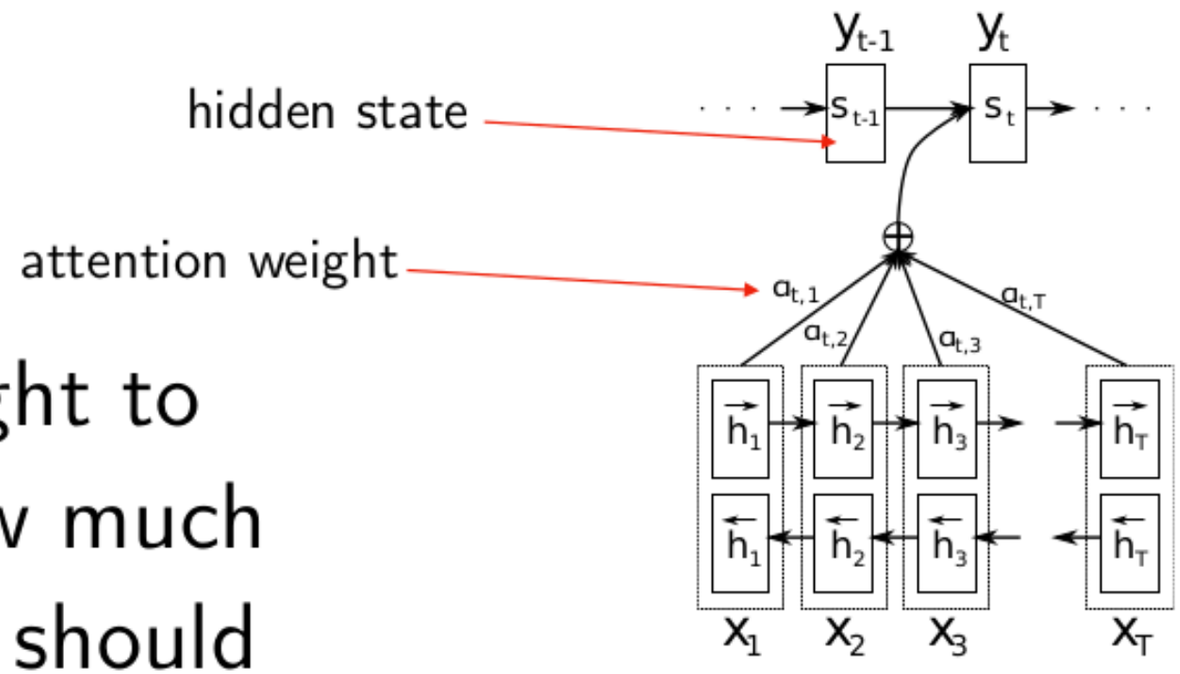


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Attention Mechanism

- While standard **RNN** architectures have led to **incredible breakthroughs** in NLP they suffer from a variety of challenges. While **in theory** they can capture **long term dependencies** they **tend to struggle modeling longer sequences**, this is still an open problem.

Self-Attention Mechanism

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

https://arxiv.org/abs/1706.03762

A Basic Version of Self-Attention

Input sequence:

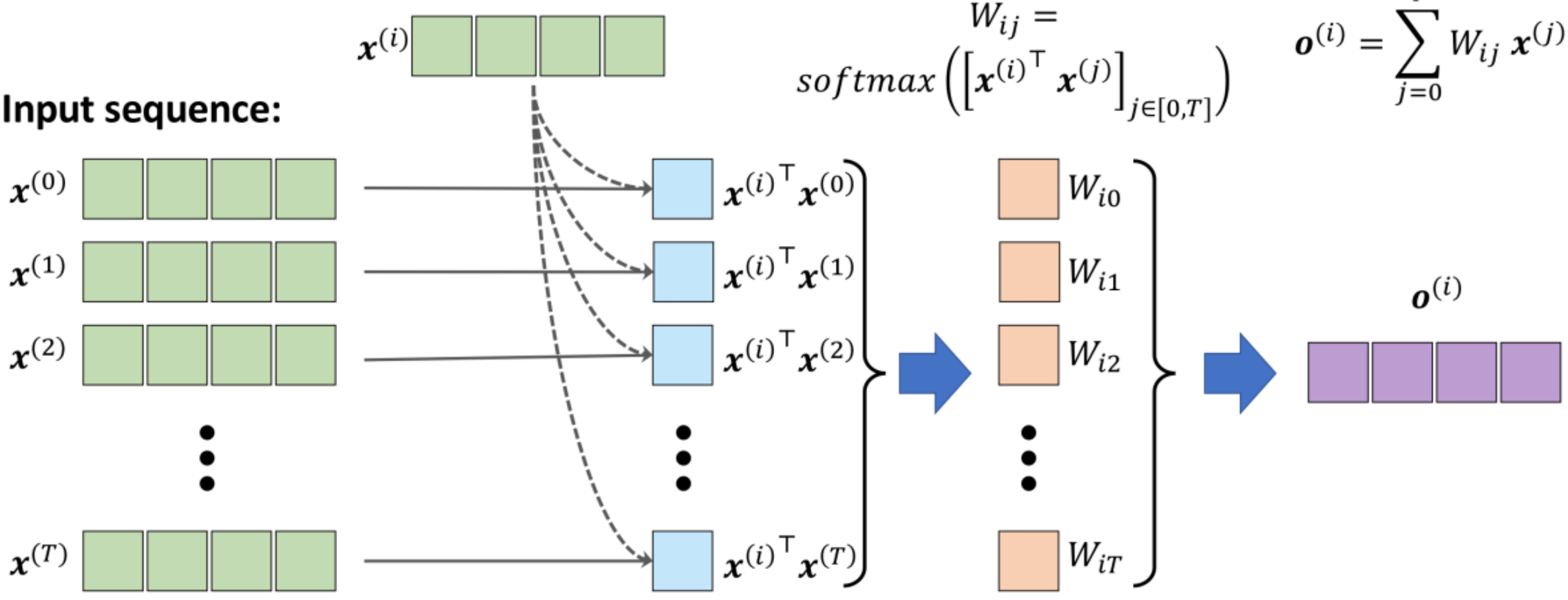
Step 1: Compute dot products

Step 2: Normalize using softmax

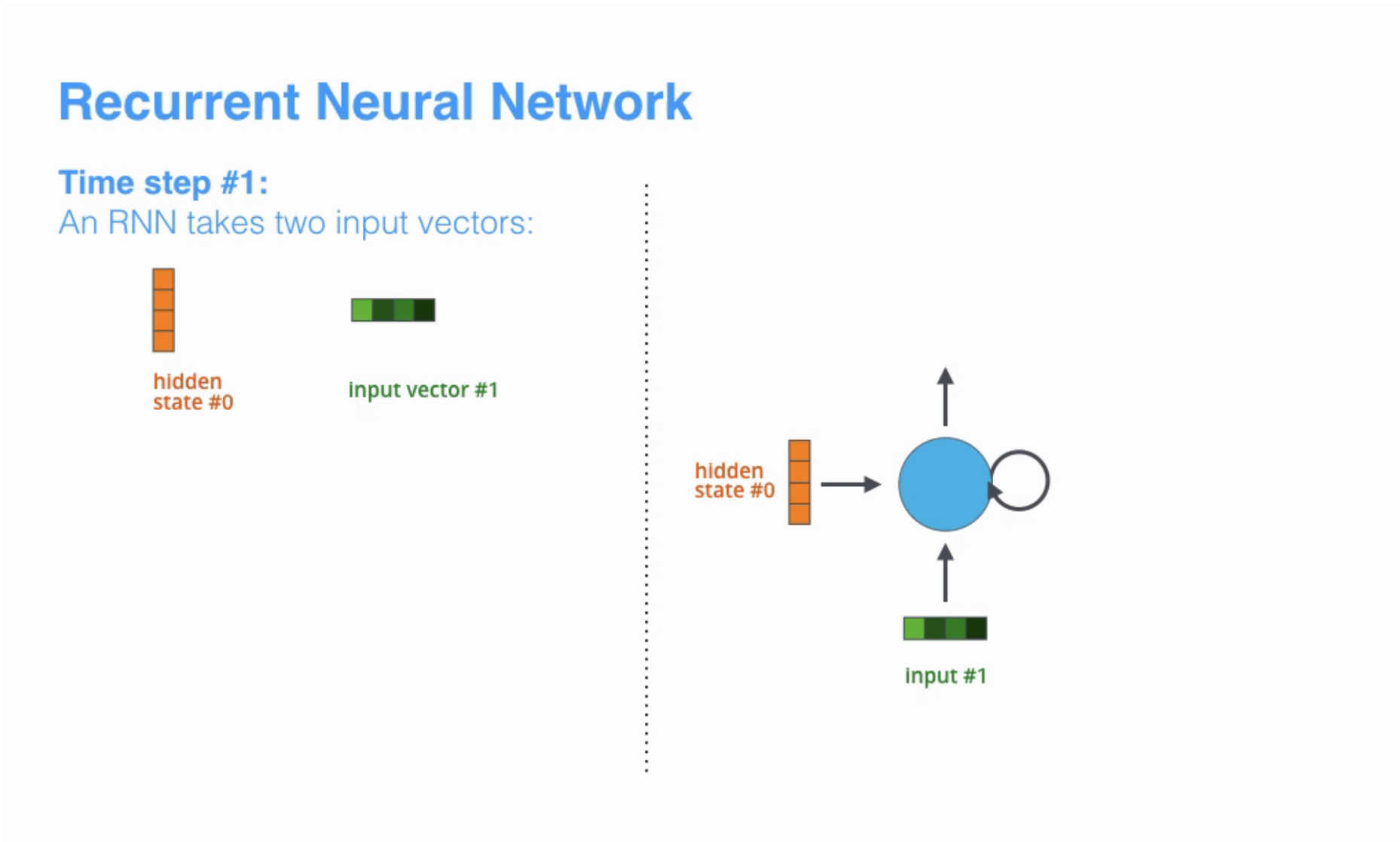
$$W_{ij} = softmax\left(\left[\boldsymbol{x}^{(i)\top}\boldsymbol{x}^{(j)}\right]_{j\in[0,T]}\right)$$

Step 3: Compute output

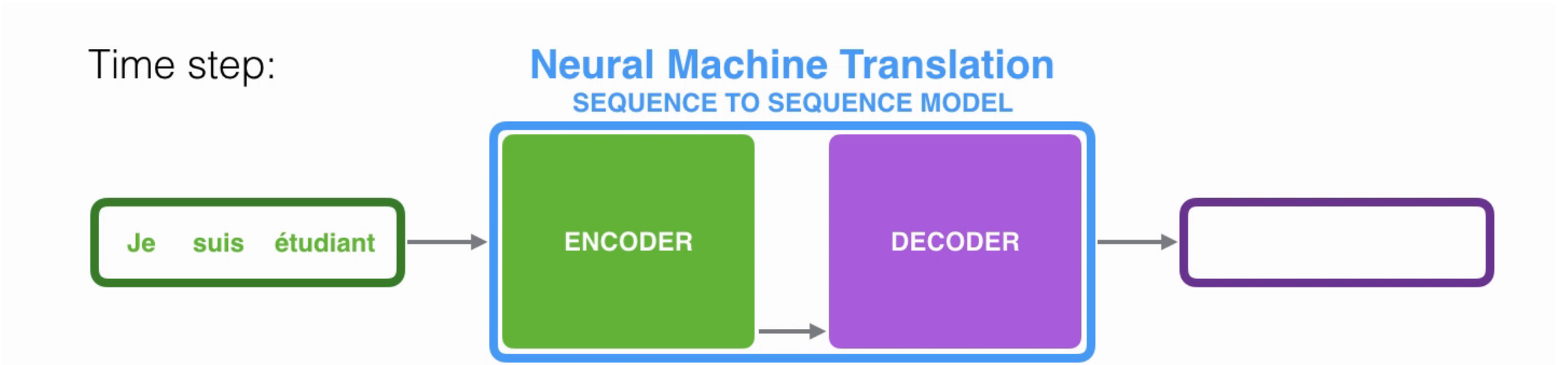
$$\boldsymbol{o}^{(i)} = \sum_{j=0}^{T} W_{ij}\,\boldsymbol{x}^{(j)}$$



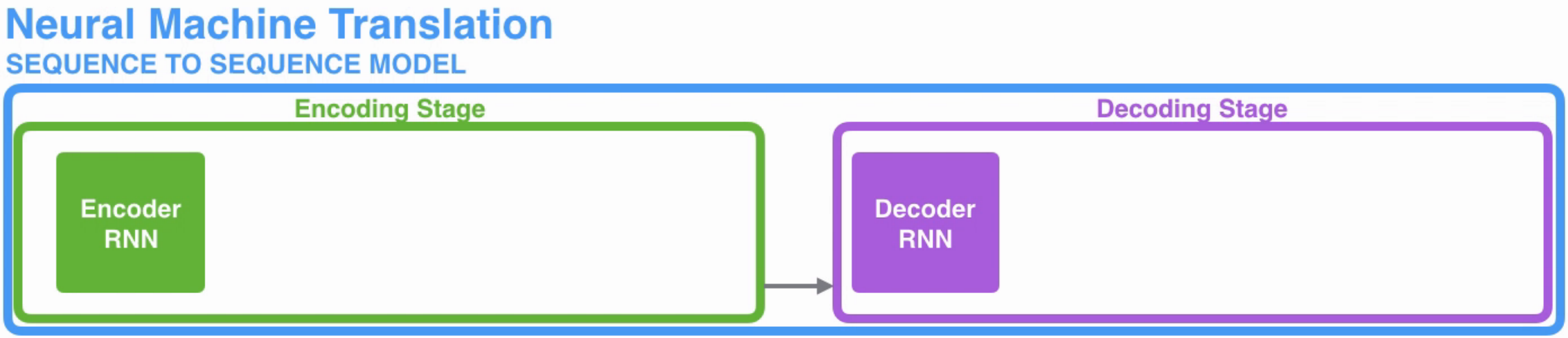
Attention Mechanism



Attention Mechanism



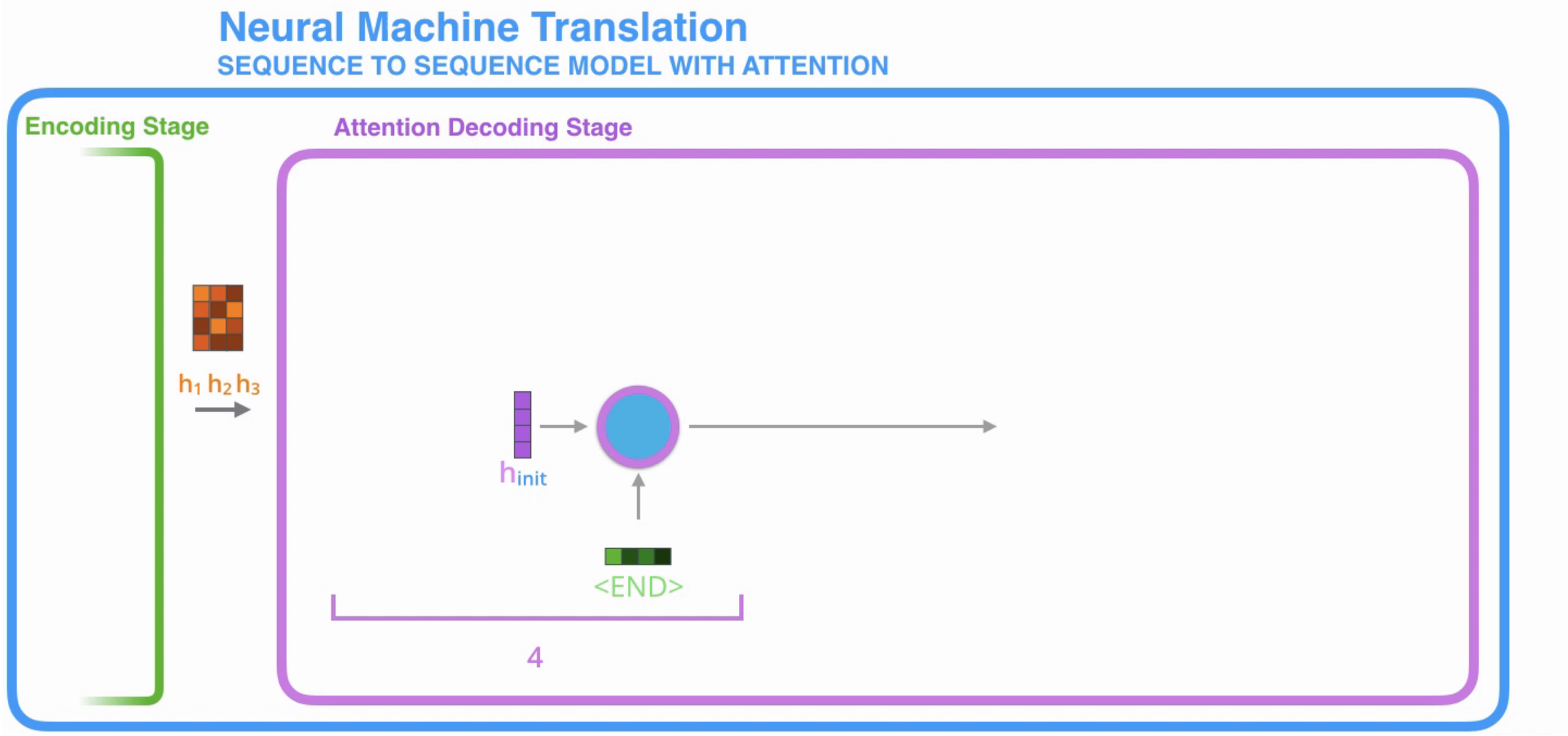
Attention Mechanism



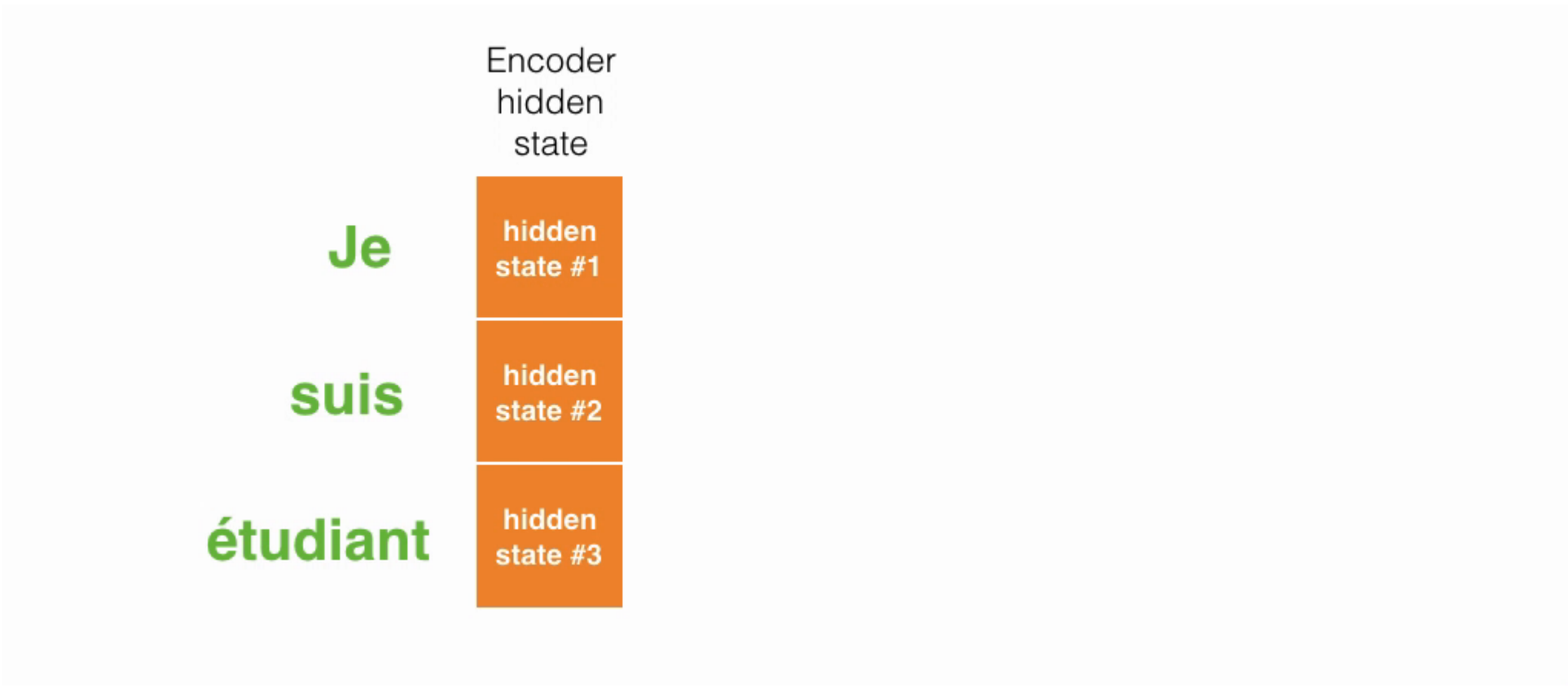
Time step: 7

Attention Mechanism

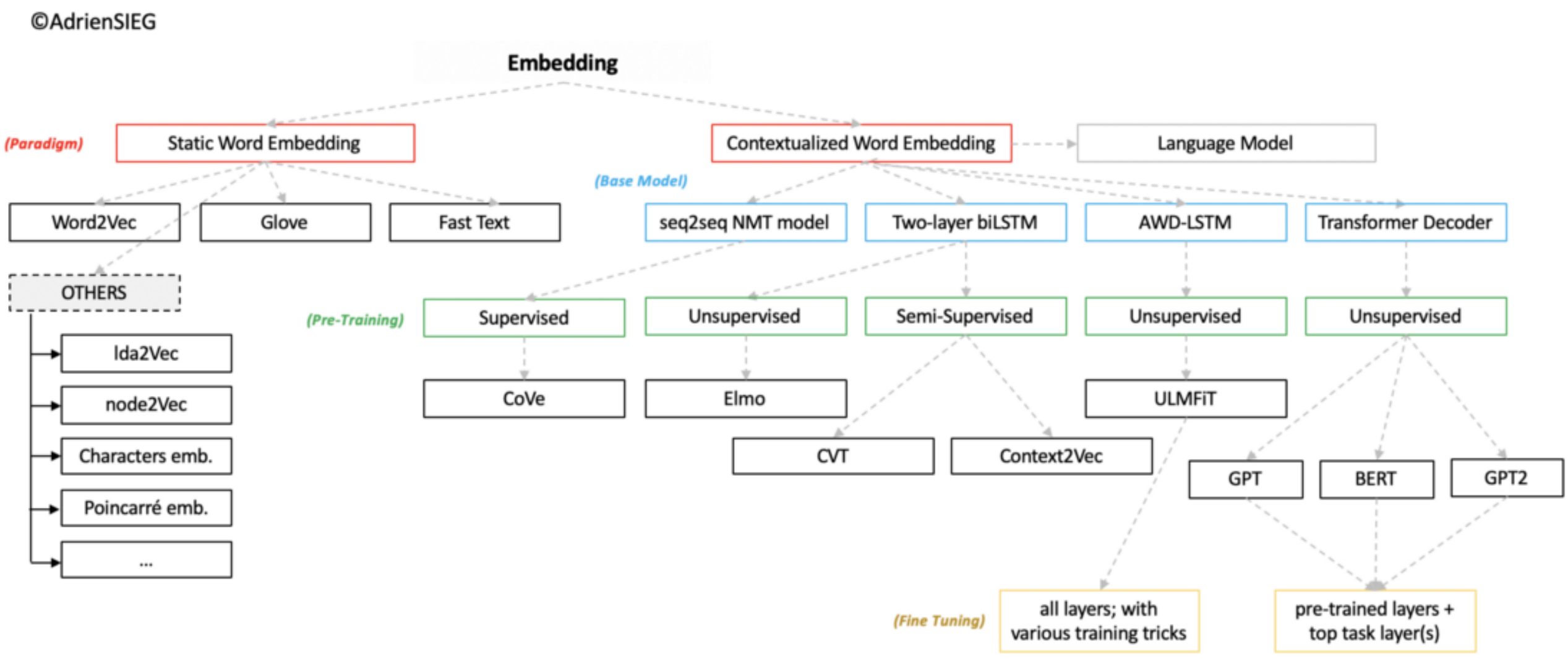
Attention Mechanism



Attention Mechanism

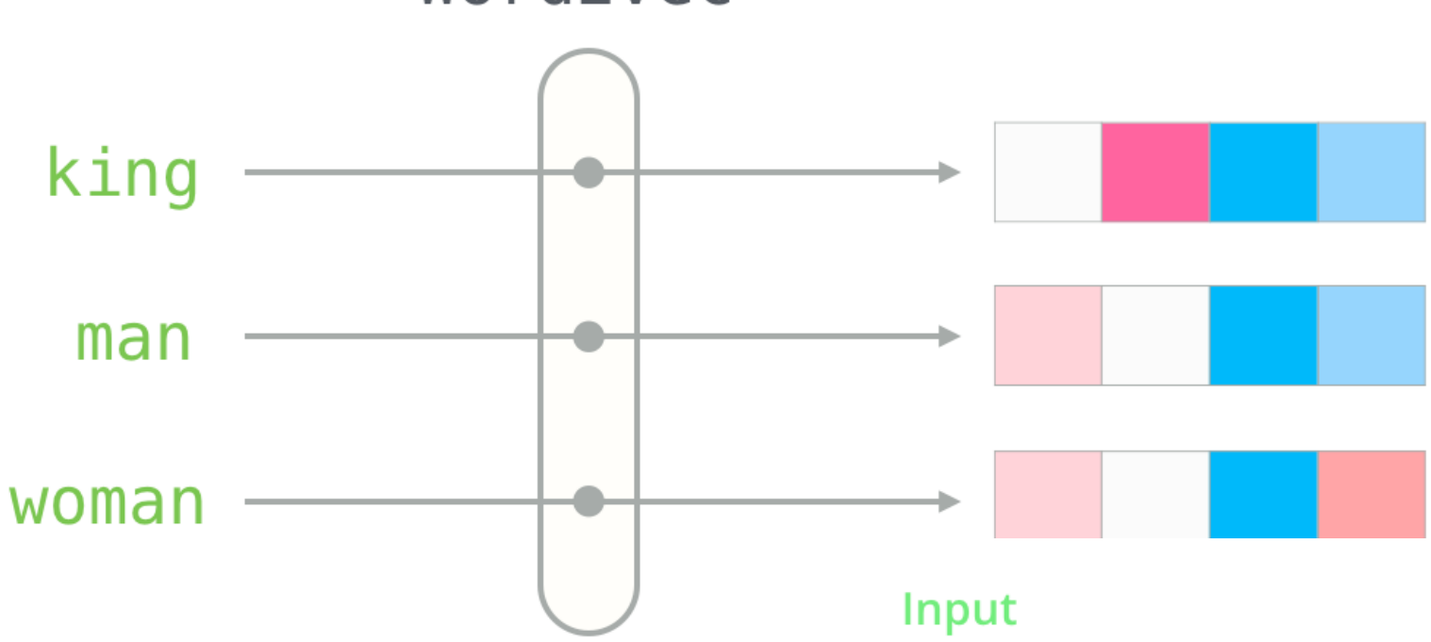


NLP Models


©AdrienSIEG
Embedding
(Paradigm)
Static Word Embedding
Contextualized Word Embedding
Language Model
(Base Model)
Word2Vec
Glove
Fast Text
seq2seq NMT model
Two-layer biLSTM
AWD-LSTM
Transformer Decoder
OTHERS
lda2Vec
node2Vec
Characters emb.
Poincarré emb.
...
(Pre-Training)
Supervised
Unsupervised
Semi-Supervised
Unsupervised
Unsupervised
CoVe
Elmo
ULMFiT
CVT
Context2Vec
GPT
BERT
GPT2
(Fine Tuning)
all layers; with
various training tricks
pre-trained layers +
top task layer(s)

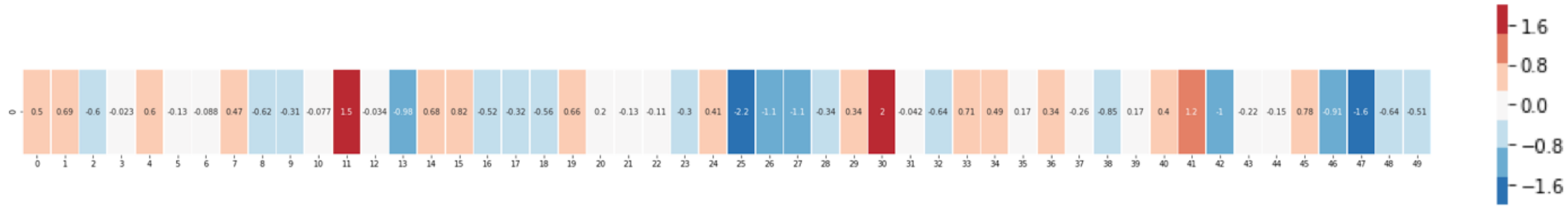
Word2vec

- Word embeddings pretrained on large amounts of unlabeled data via algorithms such as **word2vec** and **GloVe** are used to initialize the **first layer** of a neural network, the rest of which is then trained on data of a particular task



Word Embeddings

- King = [0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 , -0.13498 , -0.08813 , 0.47377 , -0.61798 , -0.31012 , -0.076666, 1.493 , -0.034189, -0.98173 , 0.68229 , 0.81722 , -0.51874 , -0.31503 , -0.55809 , 0.66421 , 0.1961 , -0.13495 , -0.11476 , -0.30344 , 0.41177 , -2.223 , -1.0756 , -1.0783 , -0.34354 , 0.33505 , 1.9927 , -0.04234 , -0.64319 , 0.71125 , 0.49159 , 0.16754 , 0.34344 , -0.25663 , -0.8523 , 0.1661 , 0.40102 , 1.1685 , -1.0137 , -0.21585 , -0.15155 , 0.78321 , -0.91241 , -1.6106 , -0.64426 , -0.51042]



“king”

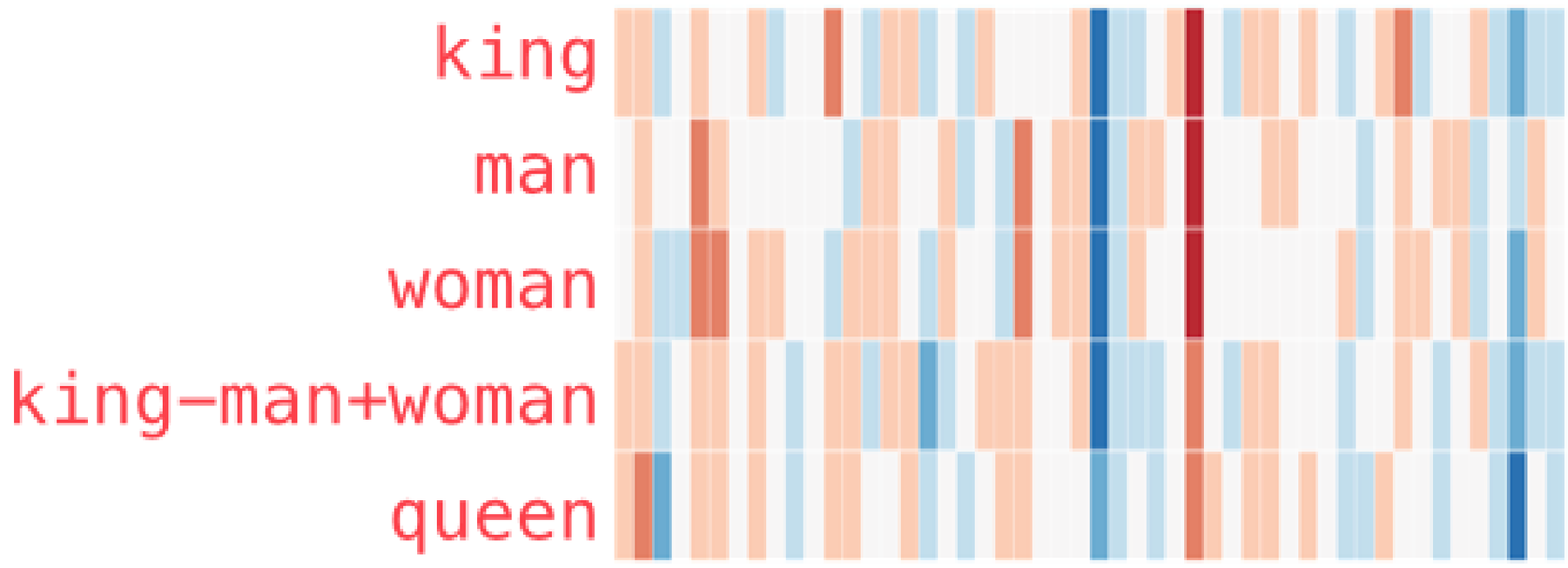
“Man”

“Woman”

Word Embeddings

- GloVe vector trained on Wikipedia

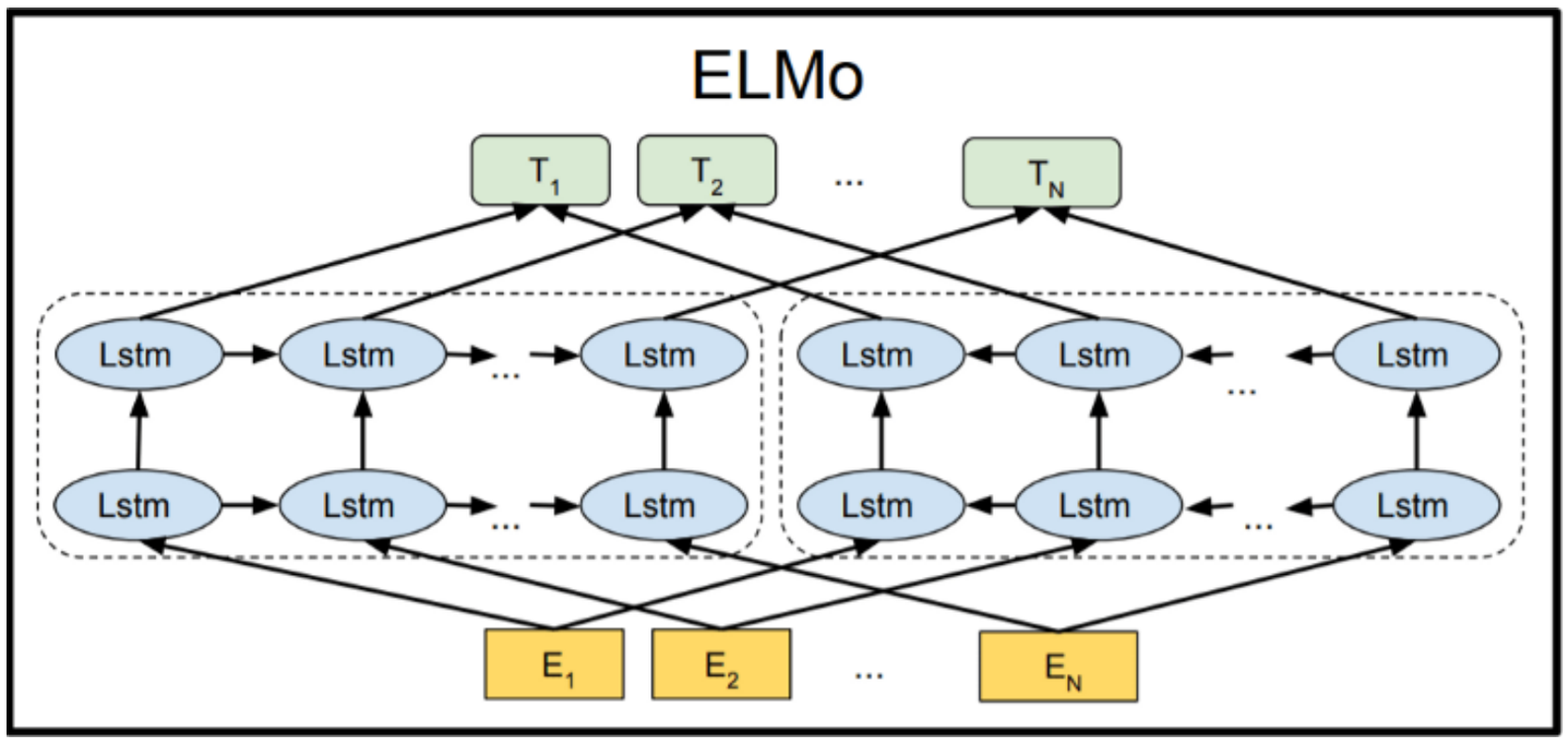
king − man + woman ~= queen



ELMO and ULMFiT

- However, an embedding like **Word2Vec** will give the same vector for “bank” in both the contexts.
  - That’s valuable information we are losing

- ELMo was the NLP community’s response to the problem of Polysemy — same words having different meanings based on their context

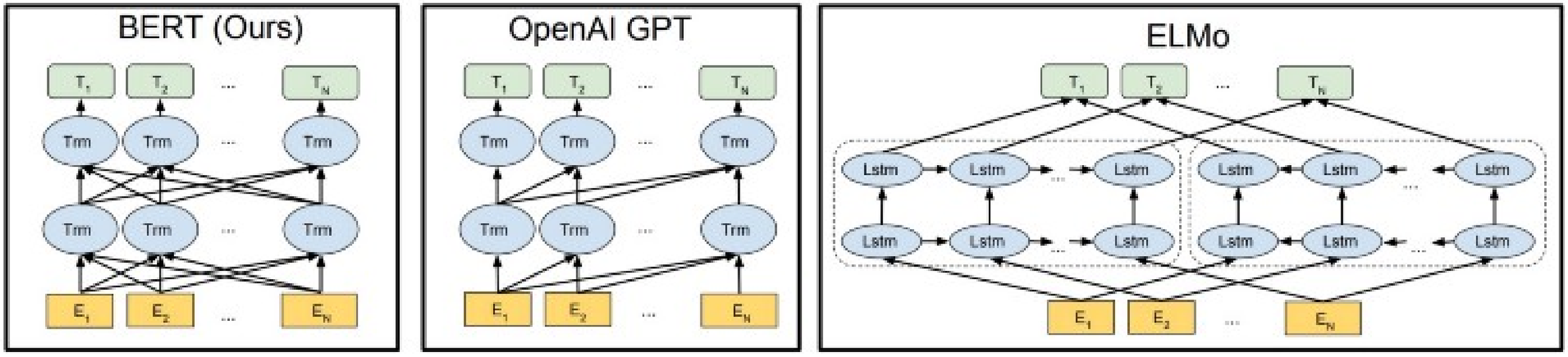


• OpenAI’s GPT

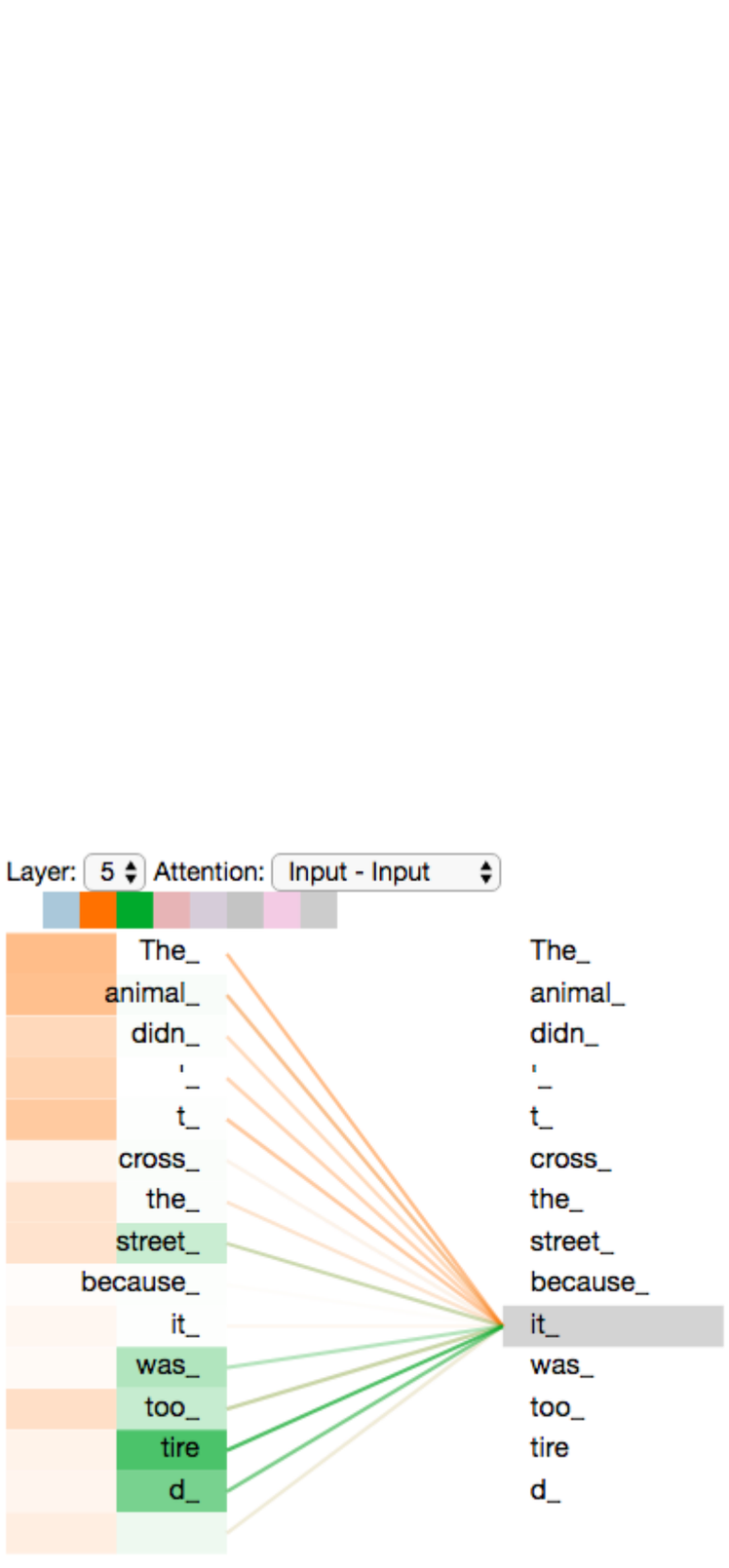
- OpenAI’s GPT extended the methods of pre-training and fine-tuning that were introduced by **ULMFiT** and **ELMo**.
  - GPT essentially replaced the LSTM-based architecture for Language Modeling with a **Transformer-based architecture**.

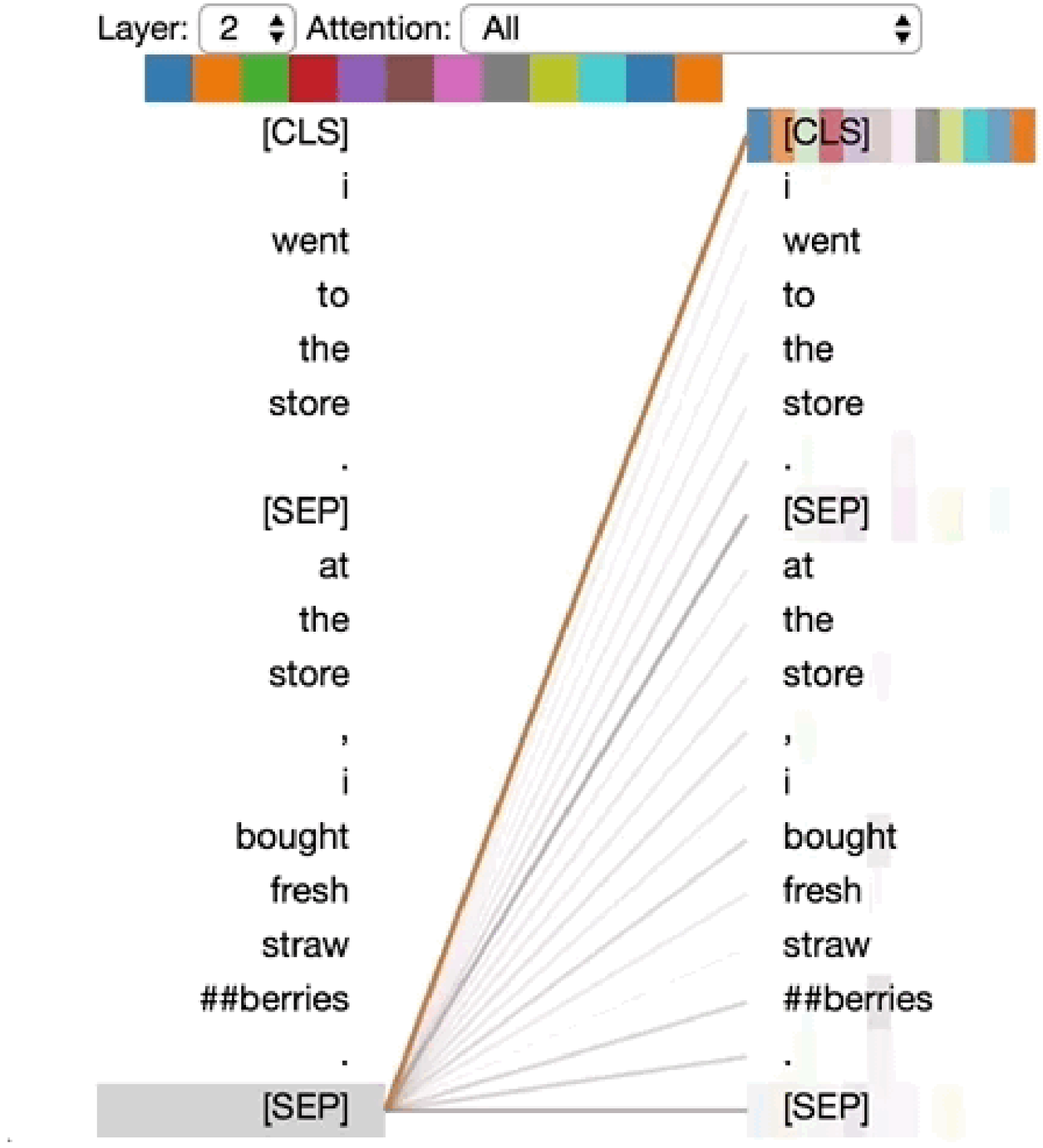
BERT

- BERT is designed as a deeply bidirectional model
  - What makes BERT different from OpenAI GPT (a left-to-right Transformer) and ELMo (a concatenation of independently trained left-to-right and right- to-left LSTM), is that the model's architecture is a deep bidirectional Transformer encoder.



BERT (attention)





Examples GPT 3

- GPT-3 is trained using next word prediction, just the same as its GPT-2 predecessor.
  - GPT-3 demonstrates that a language model trained on enough data can solve NLP tasks **that it has never encountered**. That is, GPT-3 studies the model as a general solution for many downstream jobs without fine-tuning.
  - The cost of AI is increasing exponentially. Training GPT-3 would cost over **$4.6M** using a Tesla V100 cloud instance..
  - GPT-3 comes in eight sizes, ranging from 125M to **175B parameters**
  - **a single RTX 8000, assuming 15 TFLOPS, would take 665 years to run**

- Text Generation
  - This is GPT's rockstar application

- General NLP Tasks
  - Although writing a new article is cool, the killer feature of GPT-3 is the ability to be 're-programmed' for general NLP tasks without any finetuning

Pretrained models

- **Big changes** are underway in the world of Natural Language Processing (NLP).

- The long reign of word vectors as NLP's core representation technique has seen an exciting new line of challengers emerge: **ELMo**, **ULMFiT**, **BERT**, **GPT** and the other **OpenAI transformers**.
- These works made headlines by demonstrating that **pretrained language** models can be used to achieve state-of-the-art results on a **wide range of NLP tasks**.
  - Such methods herald a watershed moment: they may have the **same wide-ranging impact on NLP as pretrained ImageNet models had on computer vision**.