Lecture Notes for

Machine Learning in Python

[[👨‍🏫, 👨‍💻, 🐍, 👨‍🔬]]

Professor Eric Larson

Sequential Networks Overview

(In progress) lecture replacing detailed implementation of recurrent networks

Lecture Agenda

- Logistics
  - Grading Update
  - Sequential Networks due **Last Day of Finals**
- Agenda
  - History of Sequential Networks
    - Recurrent Networks to Transformers
    - Word Embeddings

Class Overview, by topic

Table Data Visualization

Numpy, Pandas, Seaborn
Overviews with some in-depth discussion

Dimension Reduction and Image Processing

Scikit-learn, Scikit Image,
Intuition only, Some mathematics

Linear and Logistic Regression

Numpy, Recreate API for Scikit-learn
Detailed mathematics for simple optimization
intuition for advanced optimization

Neural Networks and Back Prop.

Numpy
Detailed mathematics for NN operations

Wide and Deep Networks | **Convolutional Networks** | **Sequential Networks**

Keras, Tensorflow
Intuition, Detailed implement.

Ethics in Language Models

ConceptNet
Case studies

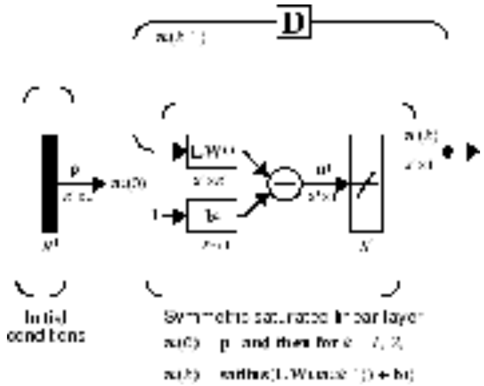
History of Sequential Neural Networks

WHEN YOU TRAIN PREDICTIVE MODELS ON INPUT FROM YOUR USERS, IT CAN LEAK INFORMATION IN UNEXPECTED WAYS.

1980's Recurrent Networks

- Hopfield Network, 1982

John Hopfield, Princeton



Neural Network Design, Hagan, Demuth, Beale, and De Jesus

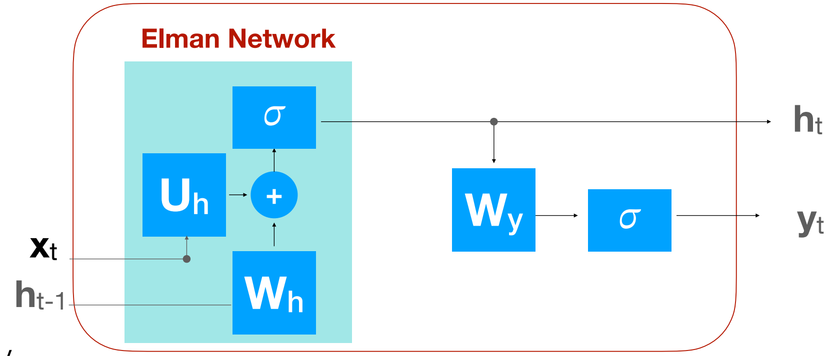
Contribution:
Training with Feedback

- Elman/Jordan Networks, ~1988

Contribution:
Time Steps for Unrolling
Separated output / state

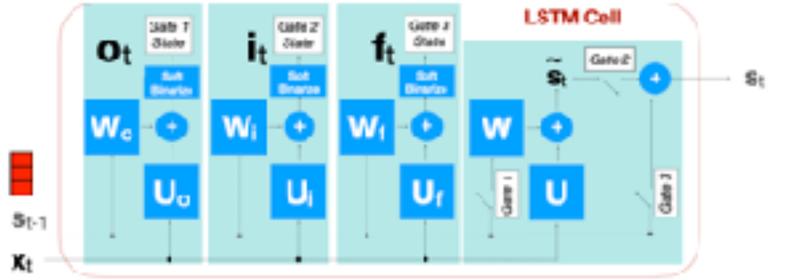
Jeffrey Elman, UCSD

Michael Jordan, Berkeley



1990’s-2000’s Better Recurrent Networks

- Long Short Term Memory, ~1997 - 2010

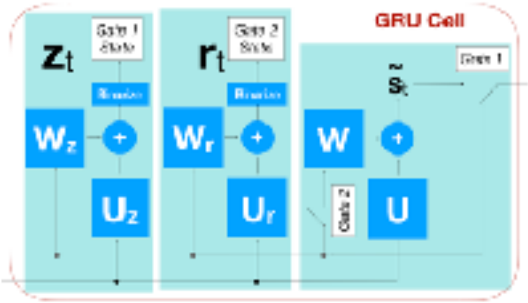


Sepp Hochreiter, Many Universities

Jürgen Schmidhuber, Switzerland

Contribution:
Long Duration Memory and State Vector Separate from Output

- Gated Recurrent Units, ~2014

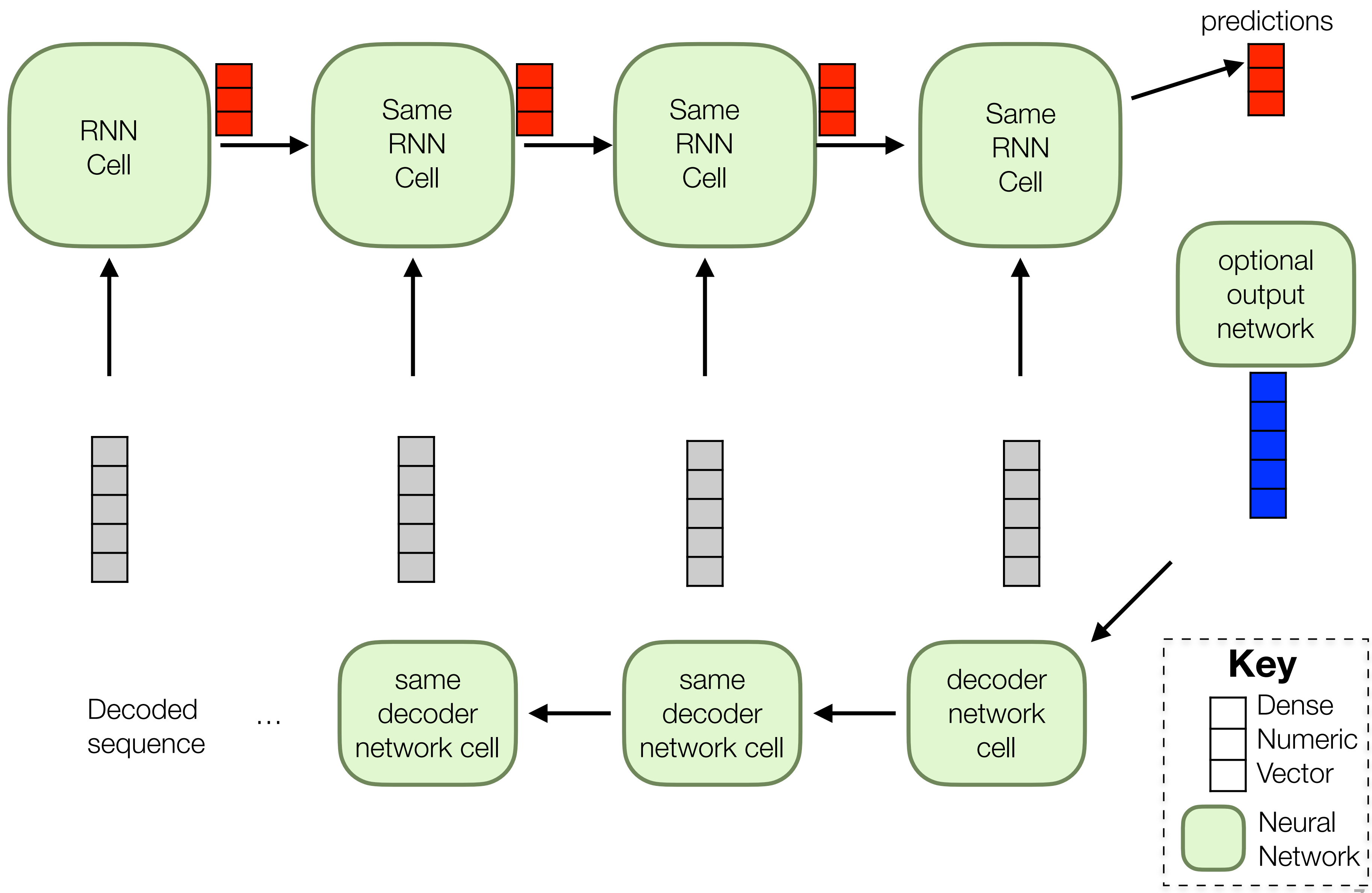


Contribution:
Fewer parameters in RNN

Yoshua Bengio

Kyunghyun Cho, Professor at NYU

General recurrent flow (many to one)



Attention (2016)

$$s_t = AttentionFunction(\mathbf{y}_{i-1}, \mathbf{x}_t) \quad \forall t, \quad 1 \le t \le M$$

$$p_t = \exp(s_t) / \sum_{t=1}^{M} \exp(s_t) \quad \forall t, \quad 1 \le t \le M$$

$$\mathbf{a}_i = \sum_{t=1}^{M} p_t . \mathbf{x}_t$$

- Google

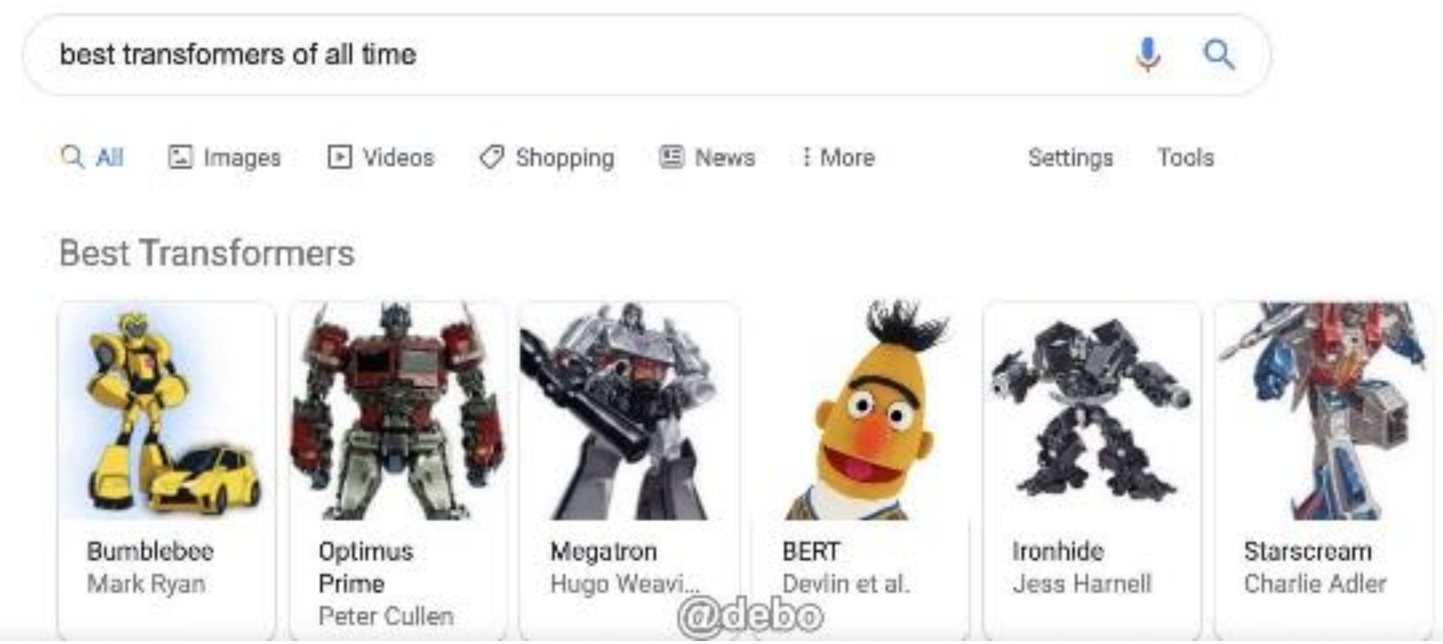
Google Neural Machine Translation:
https://arxiv.org/pdf/1609.08144.pdf
https://medium.com/@Synced/history-and-frontier-of-the-neural-machine-translation-dc981d25422d

Other big advances

- **1D Convolution** to Replace RNN (2015-2018)
- **Attention is All You Need** (2017)
- **Self-attention** (2018)
- **Multi-headed** attention Modern Transformer (2018)
- **BERT**, **GPT-#**, and other LLM etc. (2019-present)

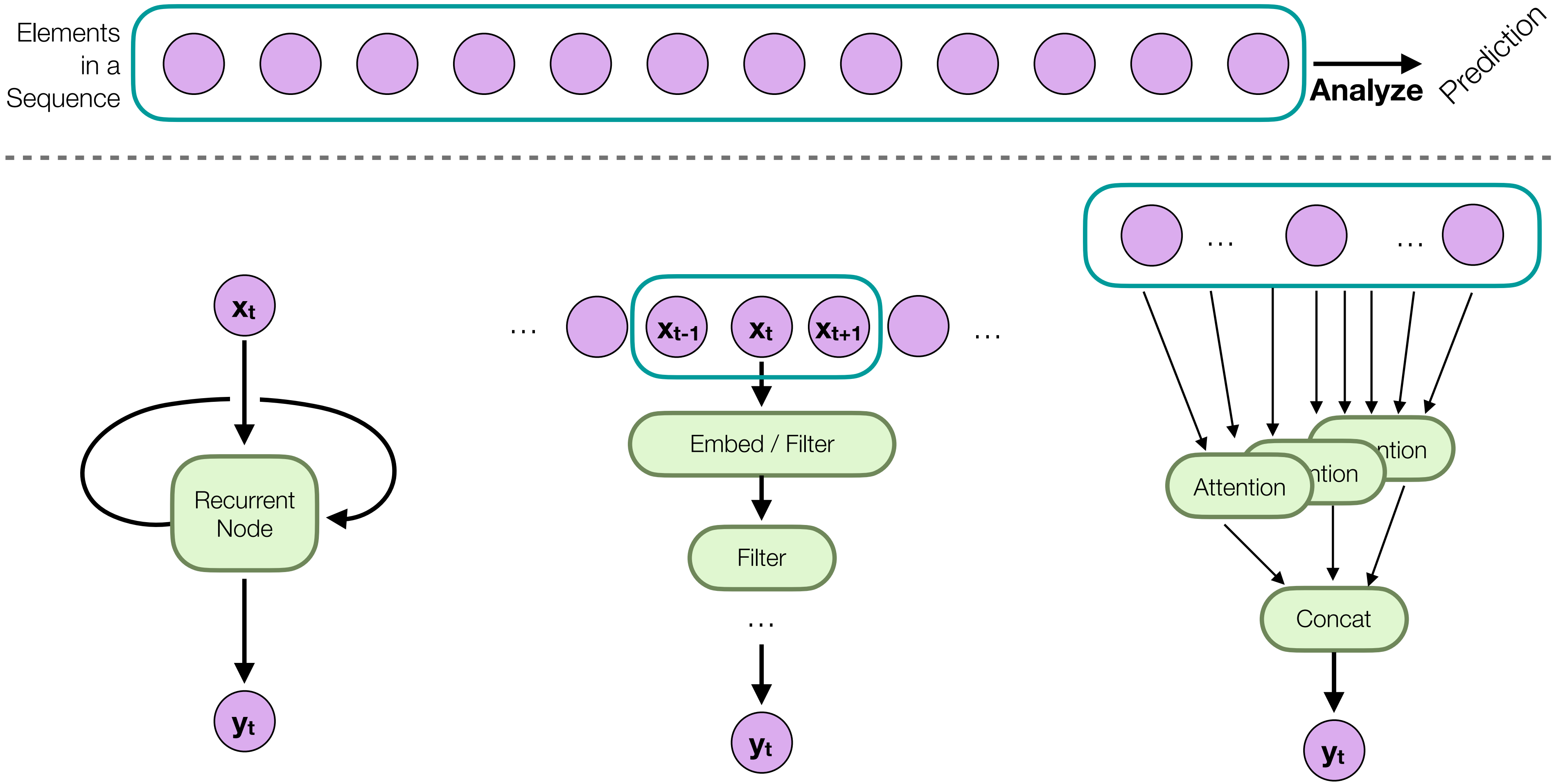
This Course

MLII Course



Overview of Sequential Networks

Sequential Networks Types



recurrent

Update Sequence State
one element at a time

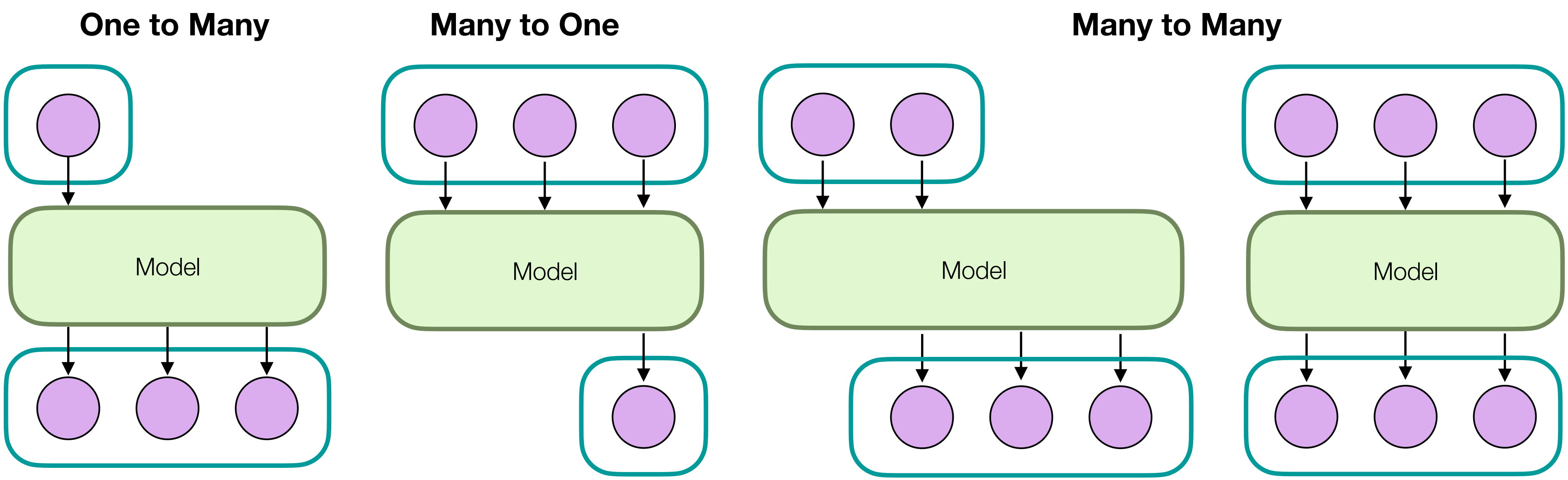
convolutional

Look at groups of Elements
in Parallel

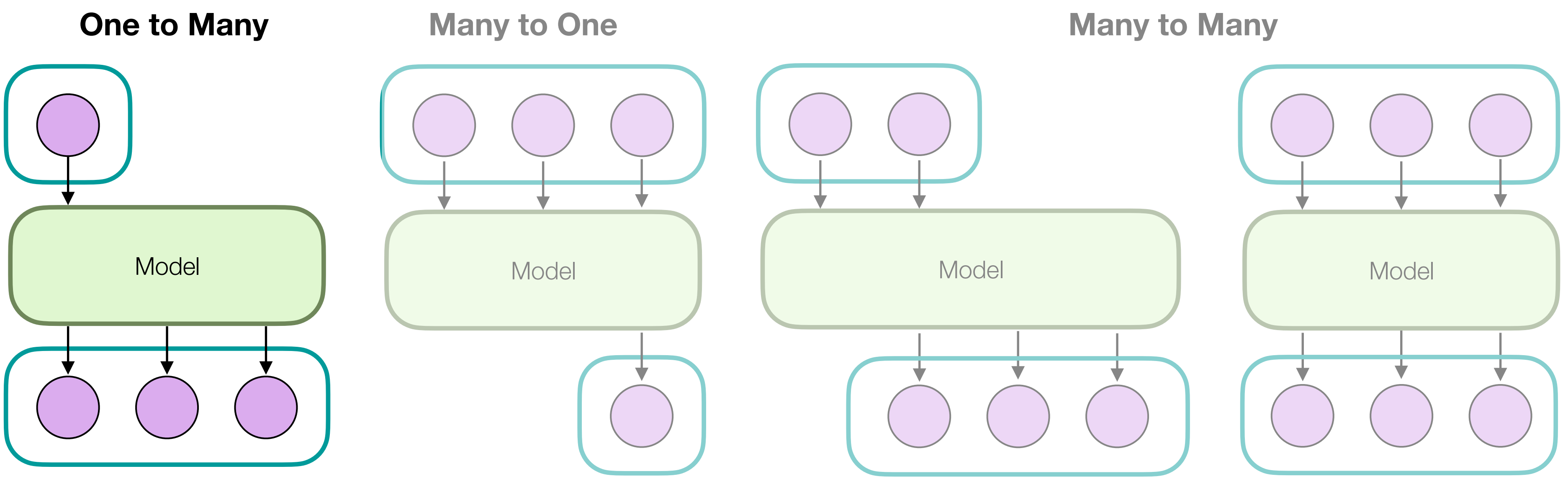
transformer

Everything Everywhere All at Once

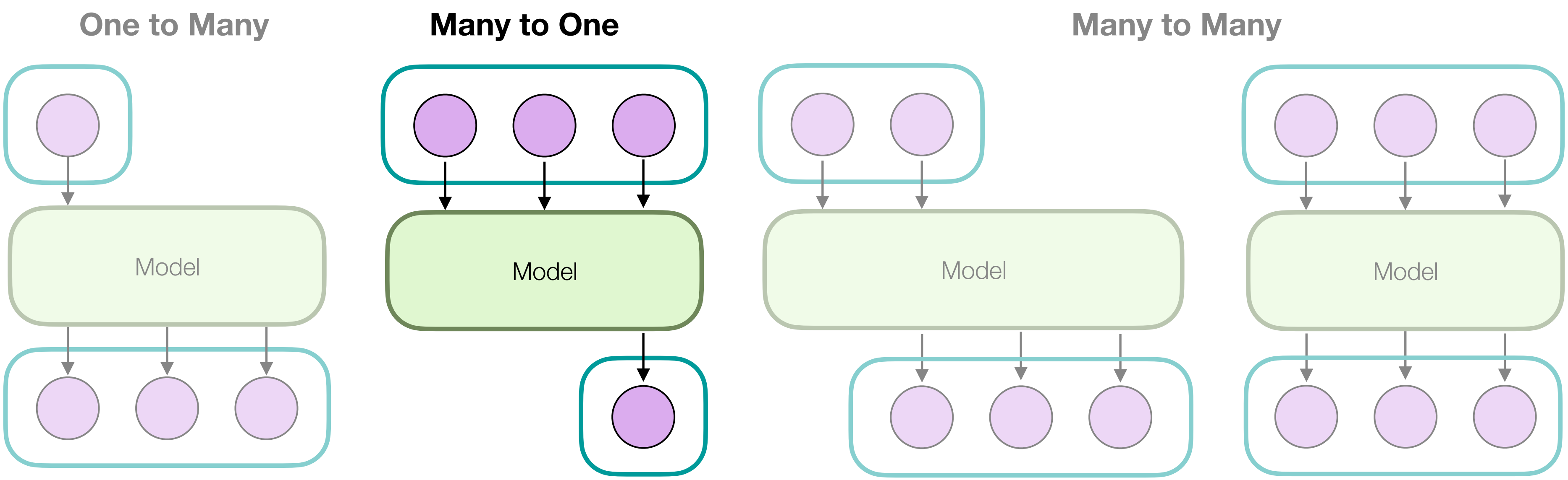
Sequential Networks: Problem Types

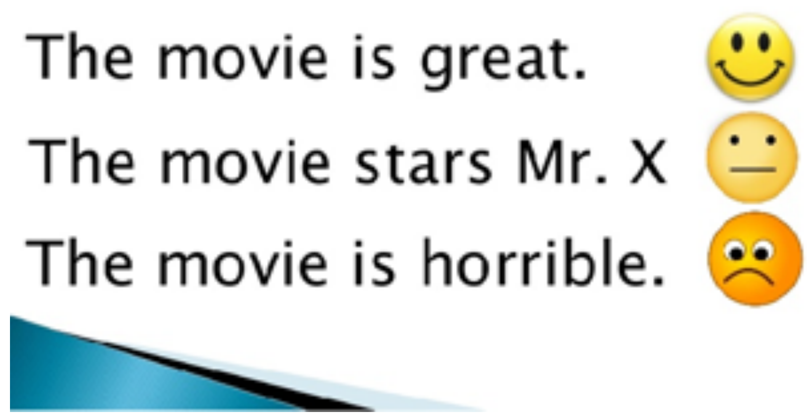


Sequential Networks: Problem Types



Sequential Networks: Problem Types

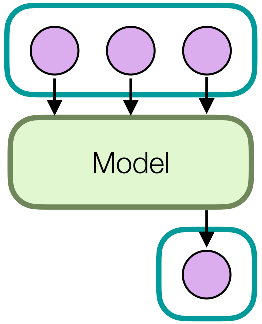




Sequential Networks: Ontology Classification

Eva Ingolf is a well known Icelandic violinist particularly recognized for her authoritative performances of solo works by J. S. Bach. She comes from a leading musical family and her father Ingólfur Guðbrandsson premiered many of the great choral works in Iceland and six of her sisters and brothers are professional musicians who have made an important contribution to the high quality of the musical life in the country.Eva Ingolf currently lives in New York City with her husband Kristinn Sv.

Many to One



Artist

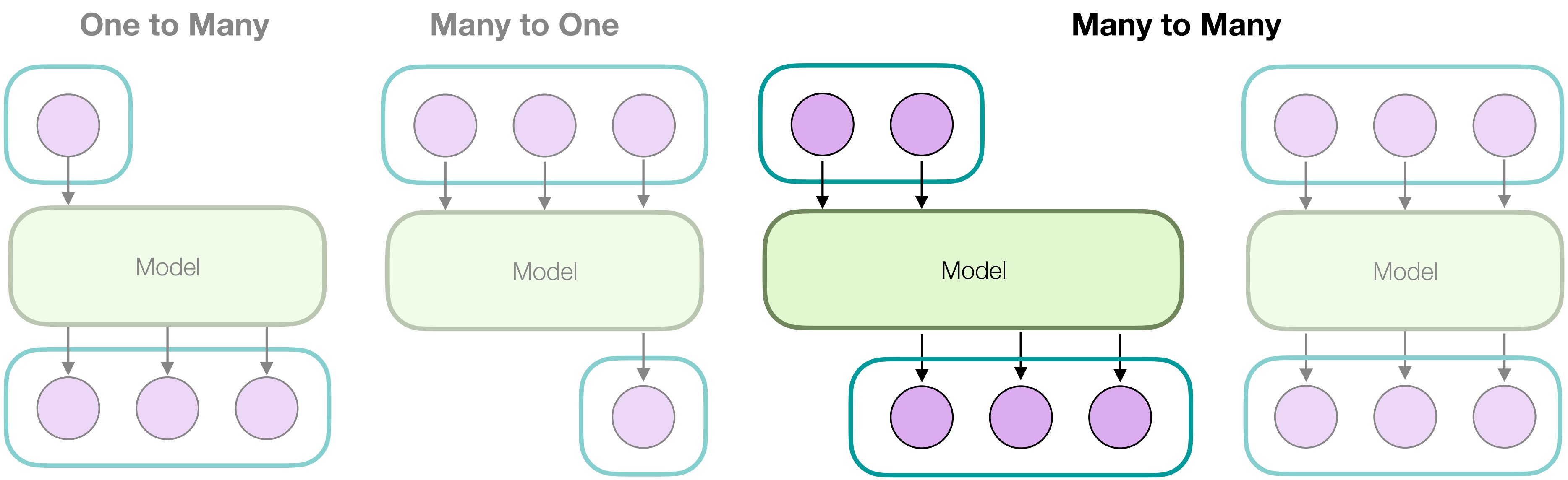
Shaun Norris (born 14 May 1982) is a South African professional golfer.Norris plays on the Sunshine Tour where he has won twice. He won the inaugural Africa Open in 2008 and the Nashua Masters in 2011. He also began playing on the European Tour in 2011 after graduating from qualifying school.

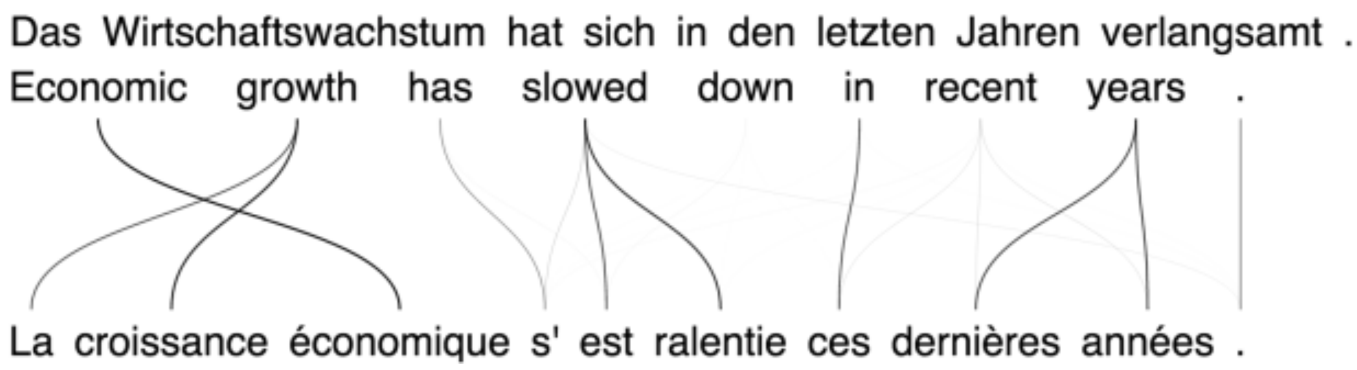
Athlete

Palace Software was a British video game publisher and developer during the 1980s based in London England. It was notable for the Barbarian and Cauldron series of games for 8-bit home computer platforms in particular the ZX Spectrum Amstrad CPC and Commodore 64.

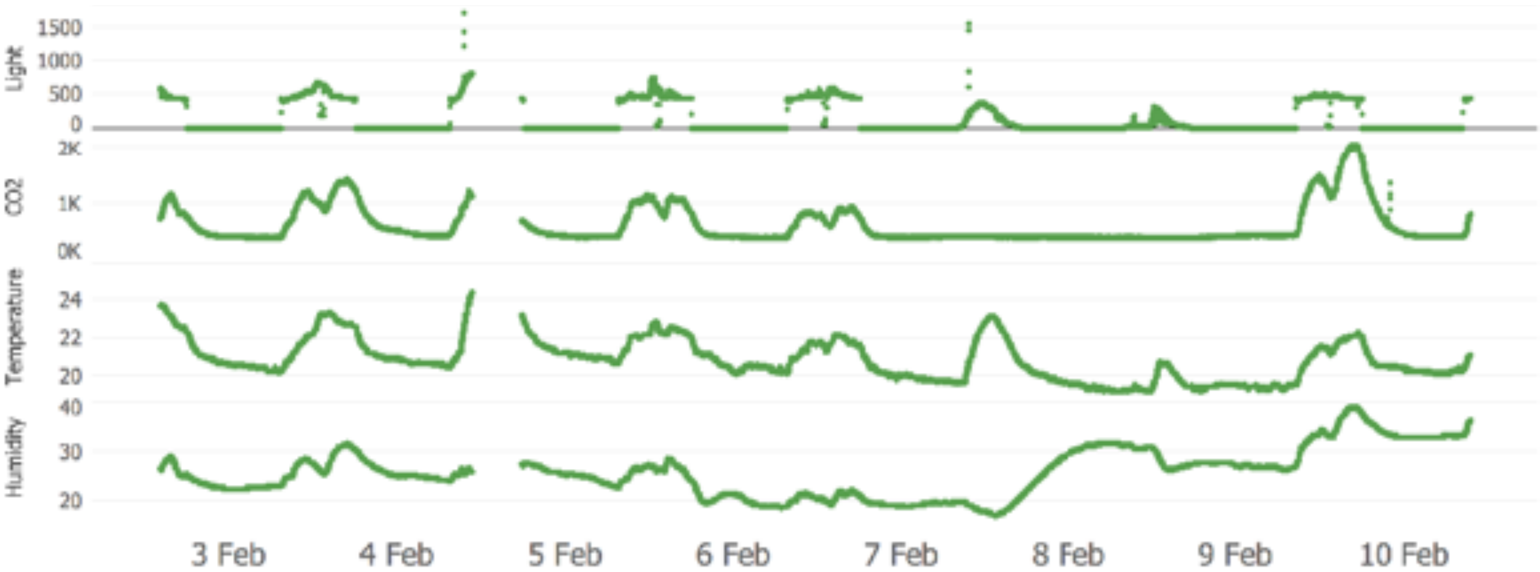
Company

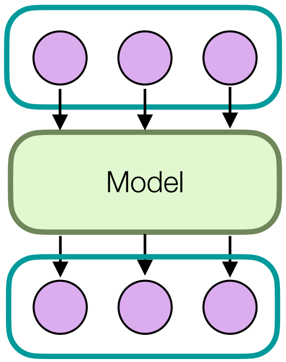
Sequential Networks: Problem Types





Sequential Networks: Problem Types

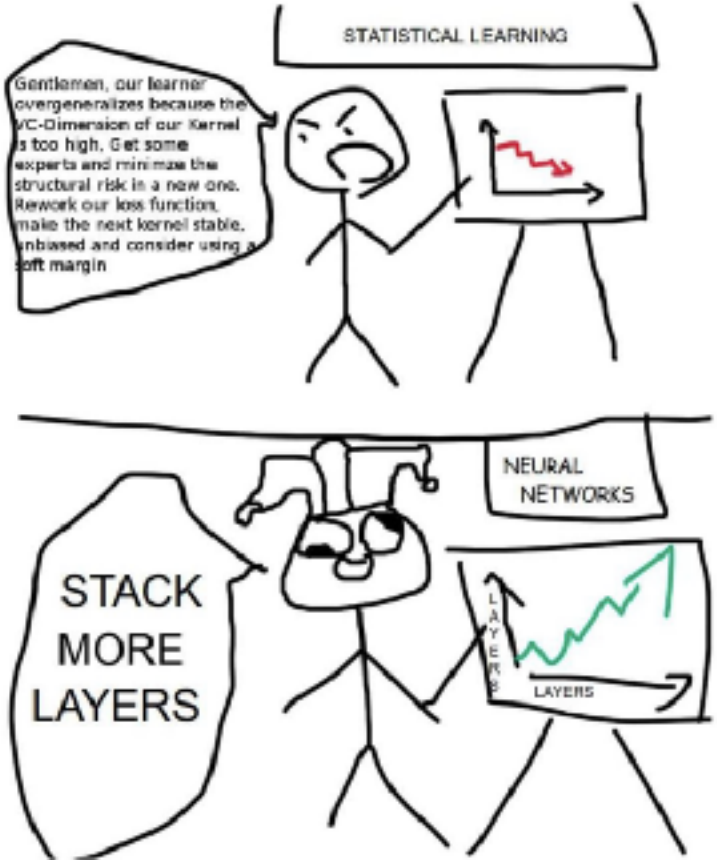




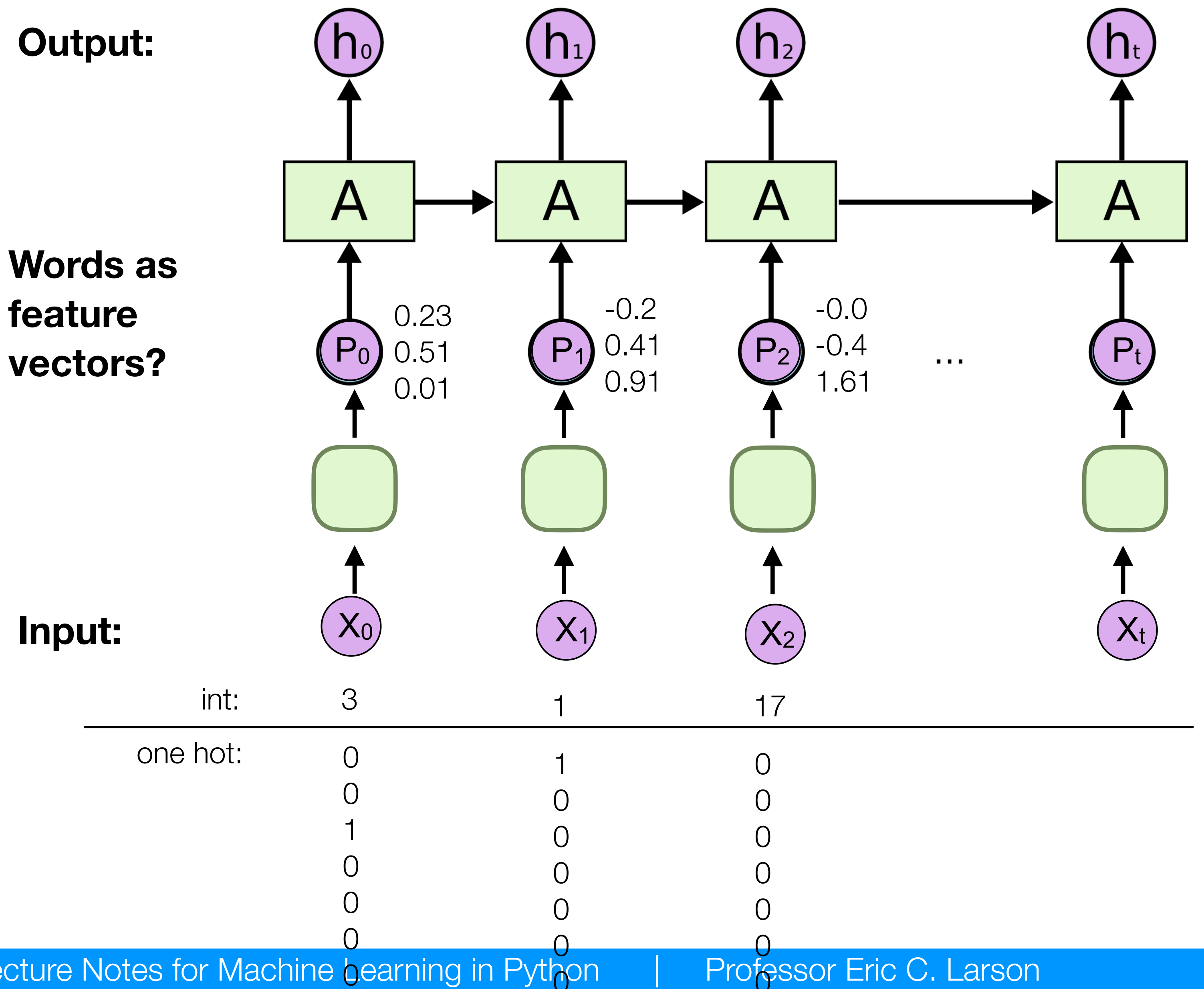
Many to Many

sequence to sequence

Word Embeddings

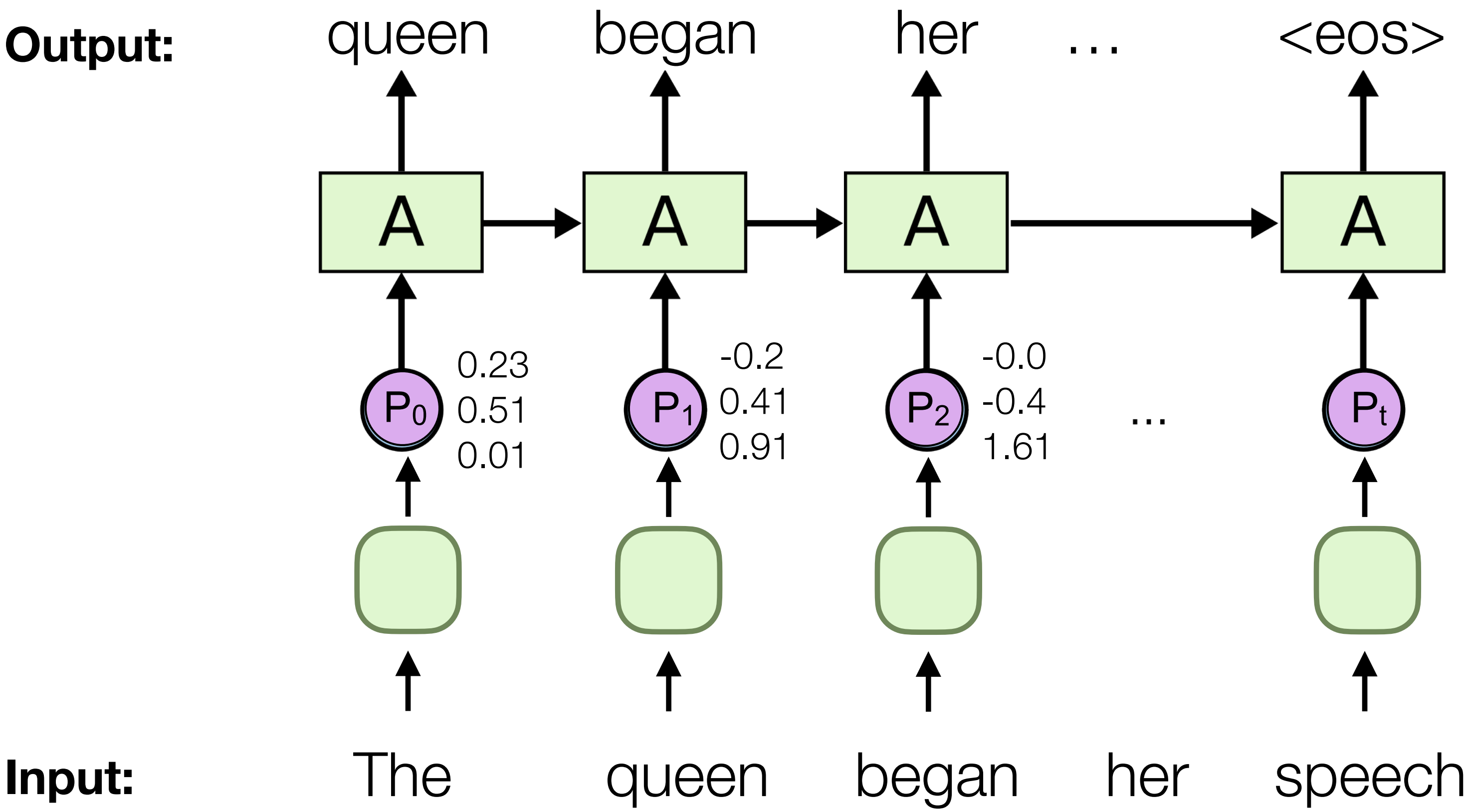


Word Embeddings (like Wide/Deep)



Word Embeddings: Training

- many training options exist
  - a popular option, next word prediction



Word Embeddings

- Many are pre-trained for you!!

GloVe

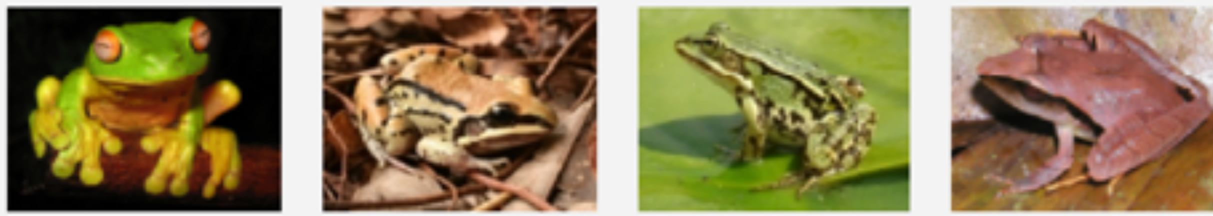
Global Vectors for Word Representation

Highlights

1. Nearest neighbors

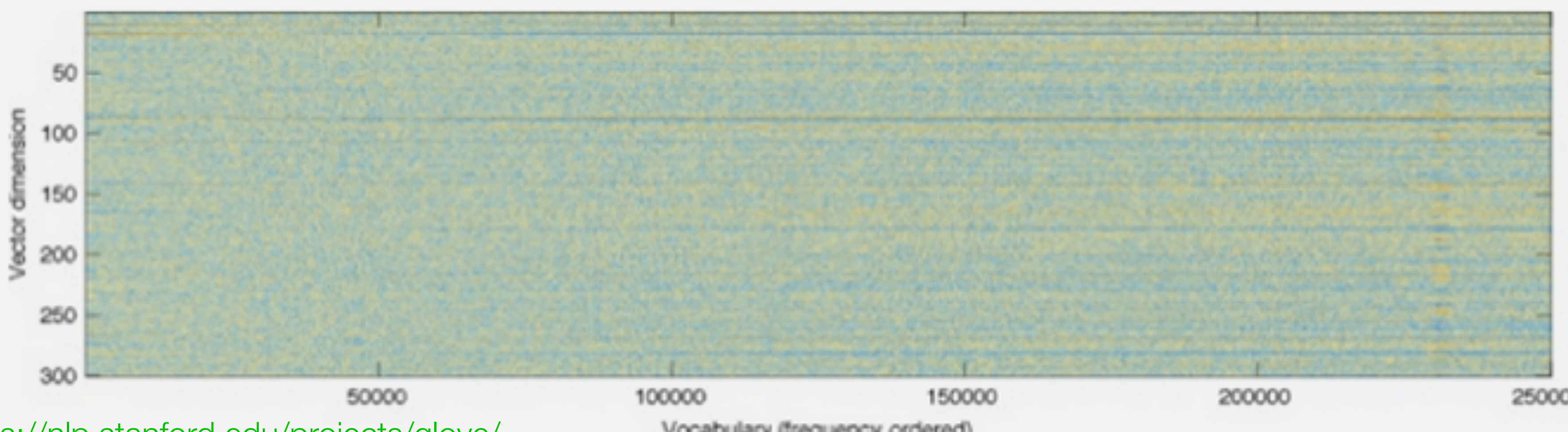
The Euclidean distance (or cosine similarity) between two word vectors provides an effective method for measuring the linguistic or semantic similarity of the corresponding words. Sometimes, the nearest neighbors according to this metric reveal rare but relevant words that lie outside an average human's vocabulary. For example, here are the closest words to the target word *frog*:

0. *frog*
1. frogs
2. toad
3. litoria
4. leptodactylidae
5. rana
6. lizard
7. eleutherodactylus



3. litoria 4. leptodactylidae 5. rana 7. eleutherodactylus

GloVe produces word vectors with a marked banded structure that is evident upon visualization:



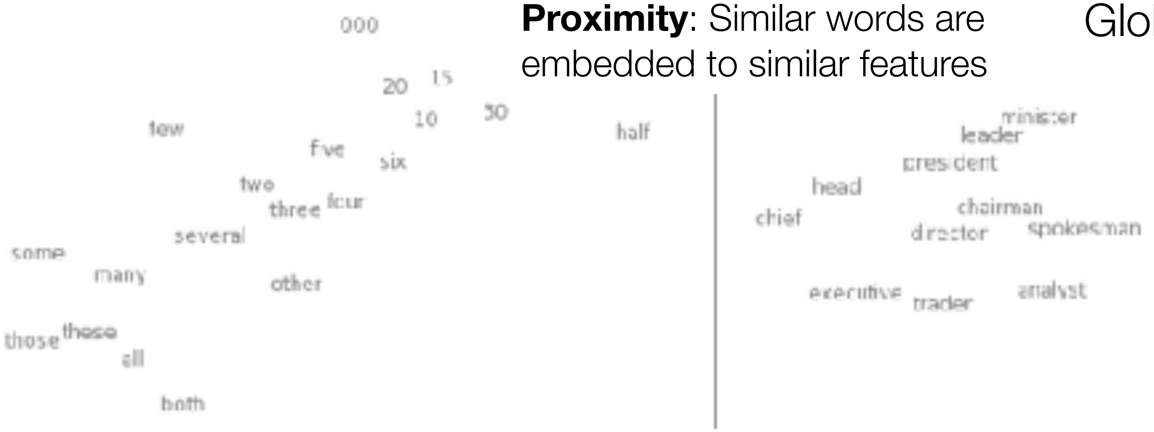
https://nlp.stanford.edu/projects/glove/

Word Embeddings: proximity

GloVe

Proximity: Similar words are embedded to similar features

Global Vectors for Word Representation



t-SNE visualizations of word embeddings. Left: Number Region; Right: Jobs Region. From Turian *et al.* (2010), see complete image.

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

What words have embeddings closest to a given word? From Collobert *et al.* (2011)

The **chairman** called the **meeting** to order.

The **director** called the **conference** to order.

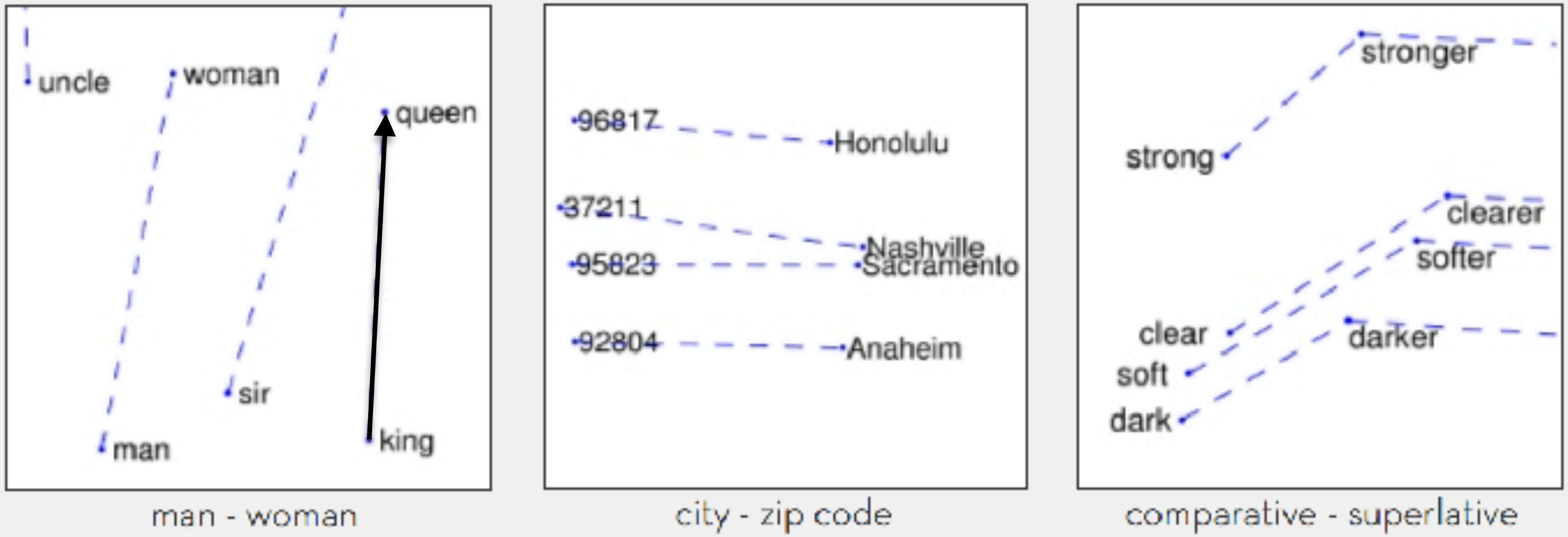
The **chief** called the **council** to order.

http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/

Word Embeddings: Analogy

GloVe

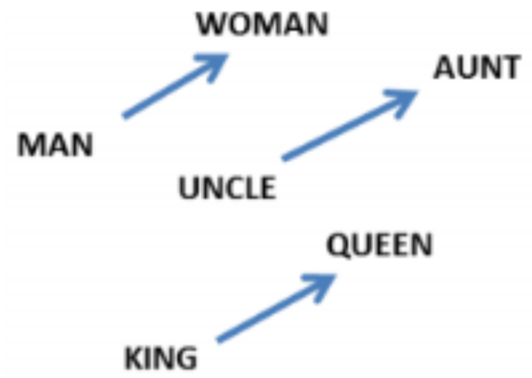
Global Vectors for Word Representation



each axis **might** encode a different type of relationship

https://nlp.stanford.edu/projects/glove/ http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/

Word Embeddings: Analogy



From Mikolov *et al.* (2013a)

GloVe

Global Vectors for Word Representation

$$W(\text{``woman''}) - W(\text{``man''}) \simeq W(\text{``aunt''}) - W(\text{``uncle''})$$

$$W(\text{``woman''}) - W(\text{``man''}) \simeq W(\text{``queen''}) - W(\text{``king''})$$

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

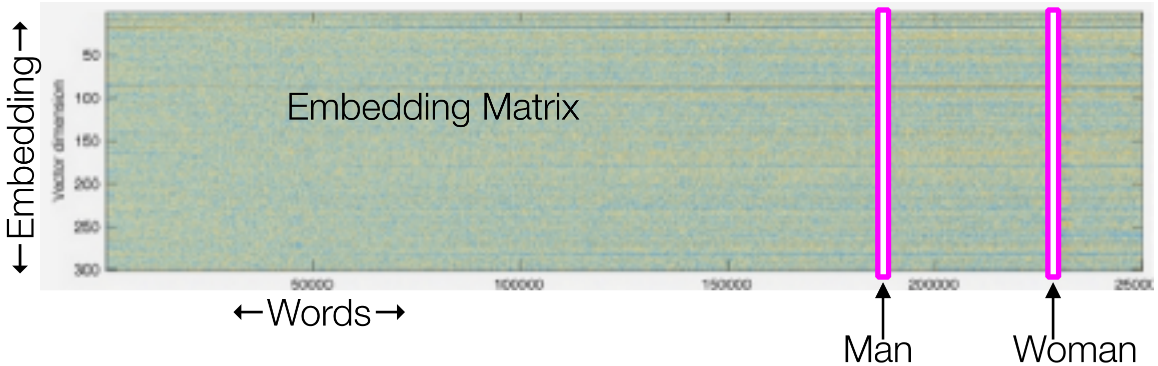
Relationship pairs in a word embedding. From Mikolov *et al.* (2013b).

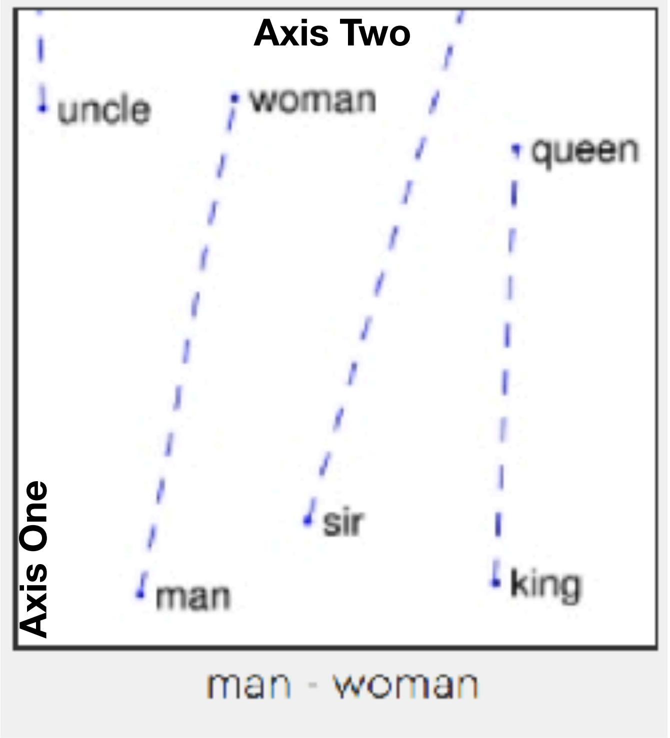
https://nlp.stanford.edu/projects/glove/

http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/

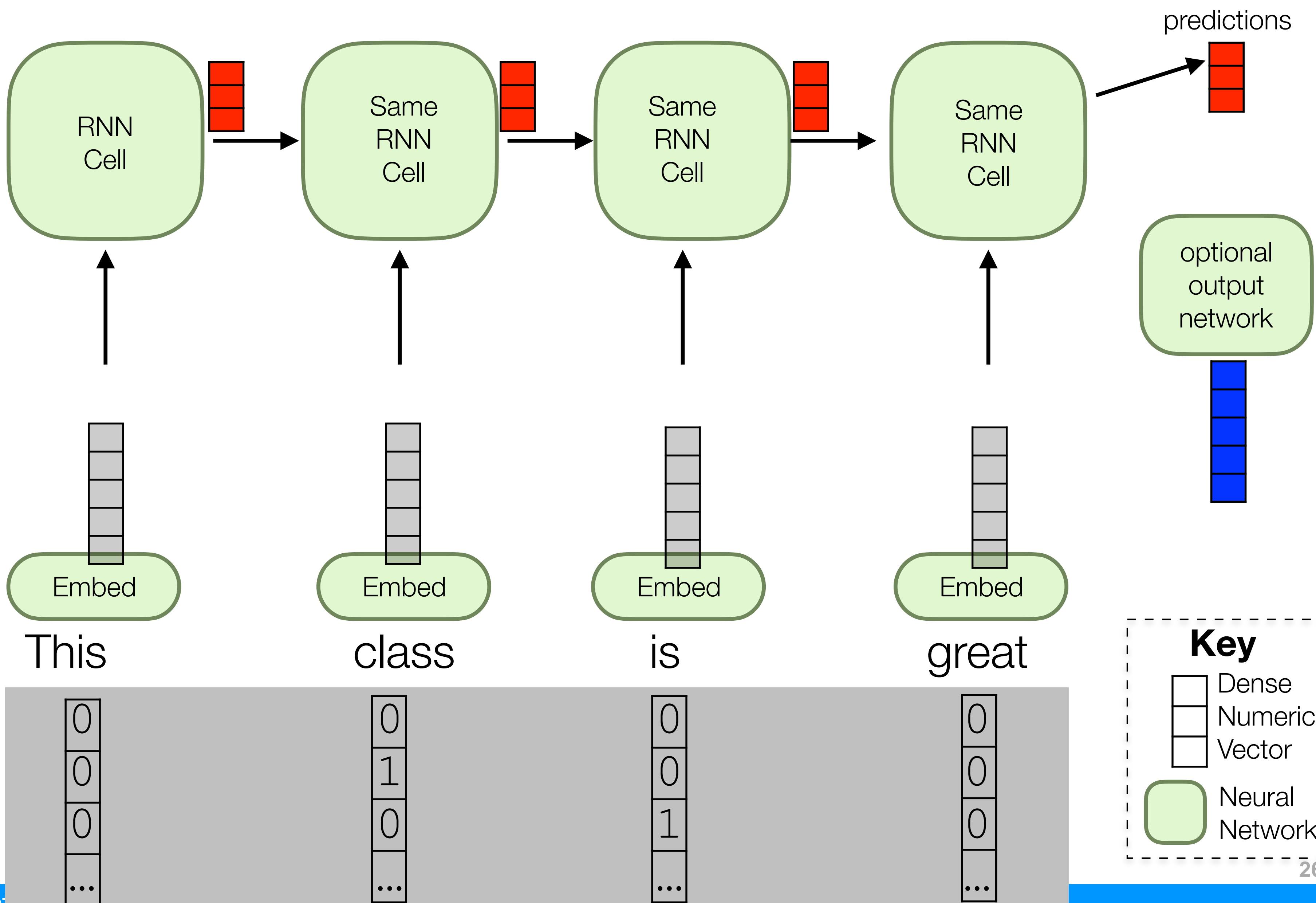
Self Test: Analogy

- Each axis on the **embedding plot** below is:
  - A. a weight inside the embedding matrix
  - B. a weighted average of weights inside the embedding layer
  - C. the average of the one hot encoding for a word
  - D. an output of the embedding matrix



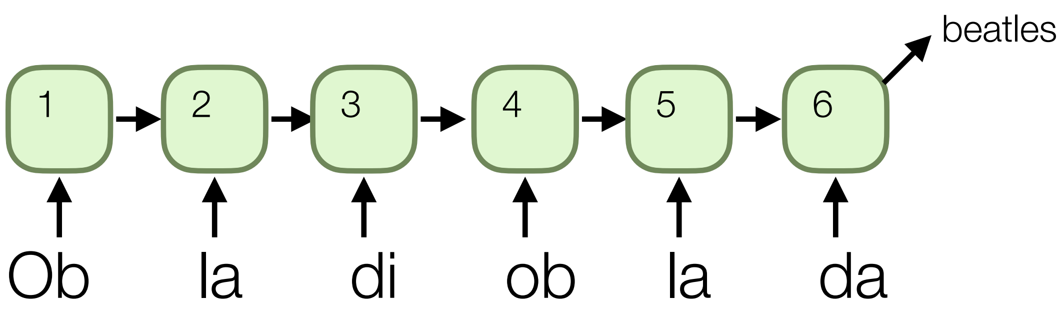


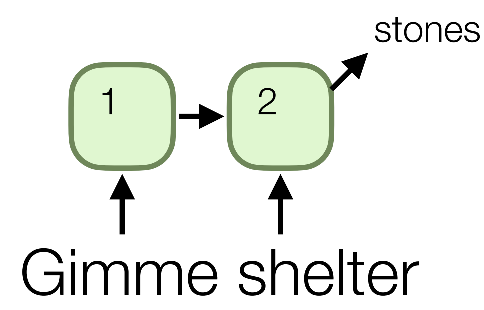
Recurrent flow with embeddings



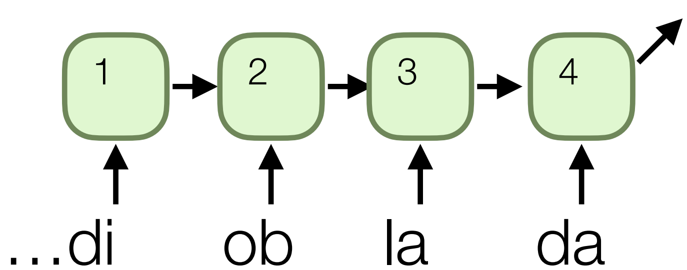
Different length input documents?

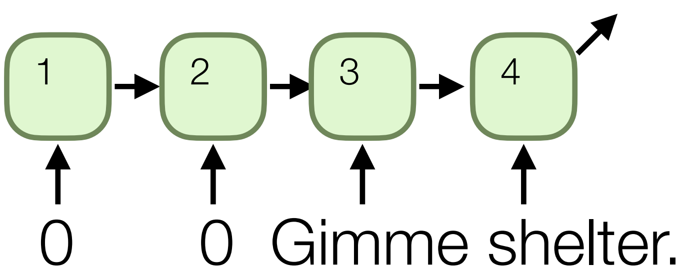
- option A: dynamic length sequences





- option B: padding/clipping



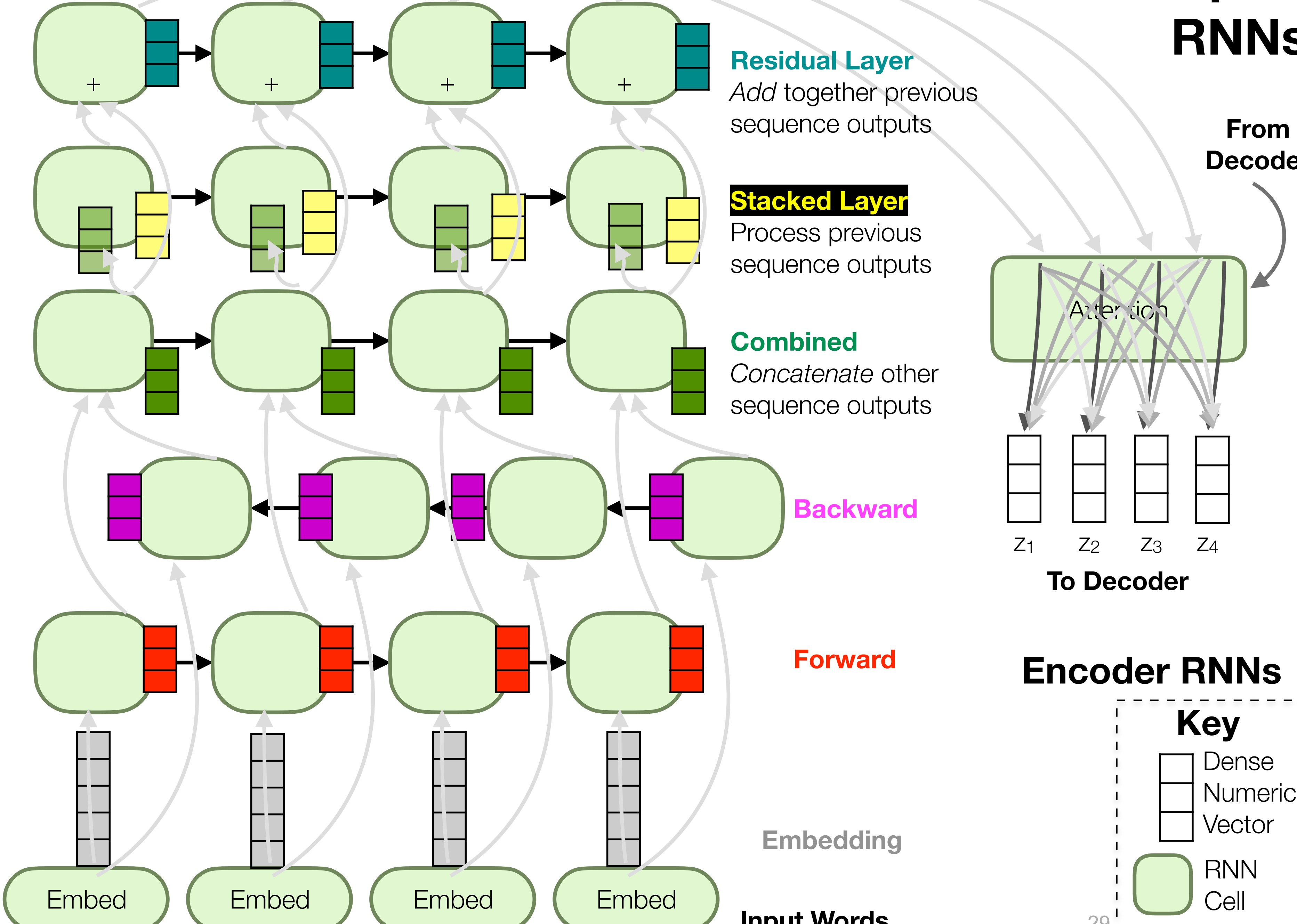


- main difference:
  speed based on computation graph design

Self Test

- The main reason **dynamic length is slow** is because:
  - A. the computation graph must be updated
  - B. weights must be tied together for each recurrent/sequential node differently
  - C. the embedding matrix cannot be applied in parallel to each word
  - D. no reason: dynamic length is actually faster

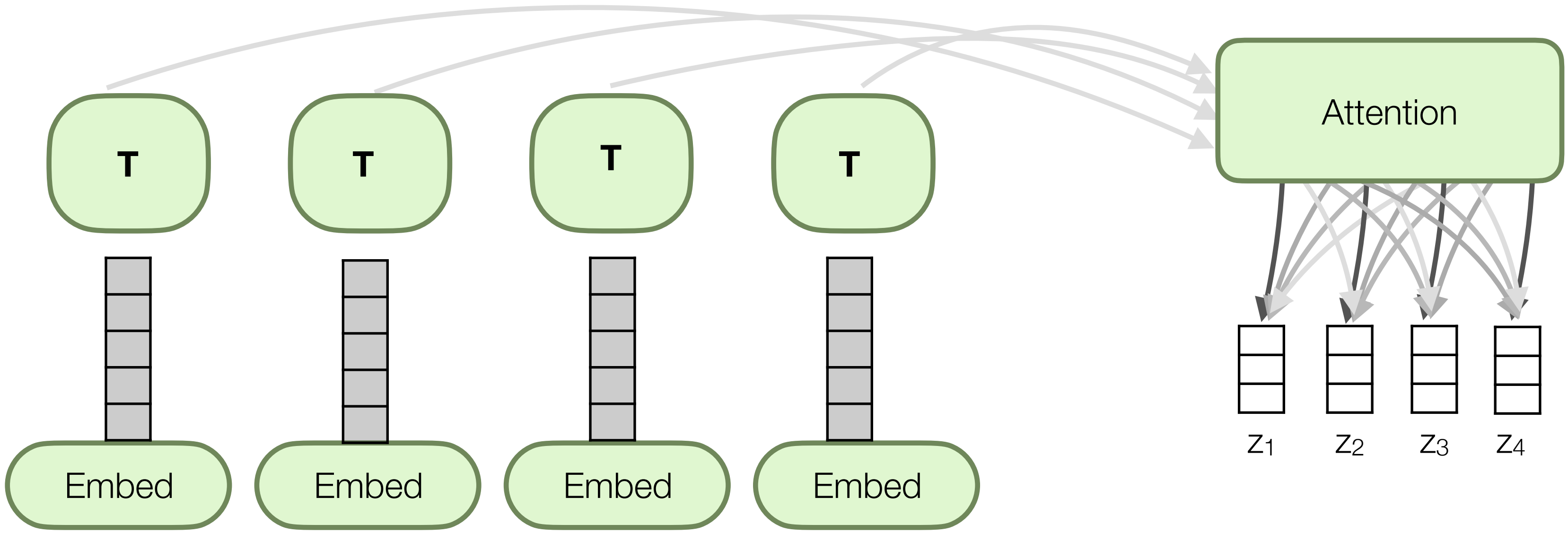
More Complex RNNs



Encoder RNNs

Transformers Intuition

- Recurrent networks track state using an “updatable” state vector, but this takes processing iterative
- Attention mechanism (in RNNs) already takes a weighted sum of state vectors to generate new token in a decoder
- … so why not just use attention on a transformation of the embedding vectors? **Do away with the recurrent state vector all together?**



Lecture Notes for

Machine Learning in Python

[👨‍🏫, 👨‍💻, 🐍, 👨‍🔬]

Professor Eric Larson

Sequential Networks Overview