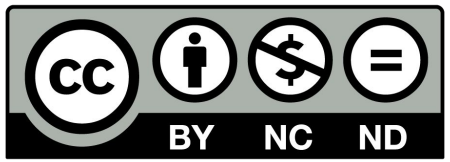
Learn ai

Natural Language Processing

Lesson 13: NLP

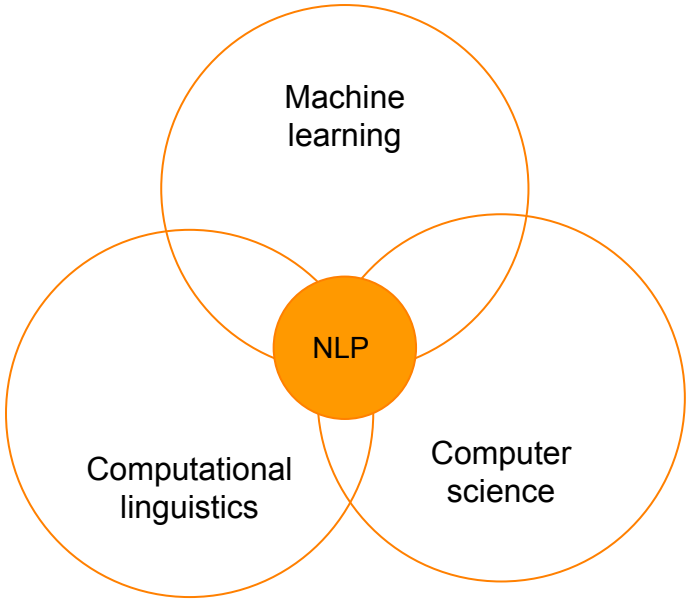
An AI Commons initiative



https://creativecommons.org/licenses/by-nc-nd/4.0/

What is Natural Language Processing?

- NLP is a branch of AI that is concerned with the interaction between humans and computers through a natural language
- The goal is to help computers read, interpret and manipulate human language in a valuable way.
- NLP is automating analysis, generation, and acquisition of natural languages.
- Most NLP techniques depend on ML to bridge the gap between human communication and computer understanding.

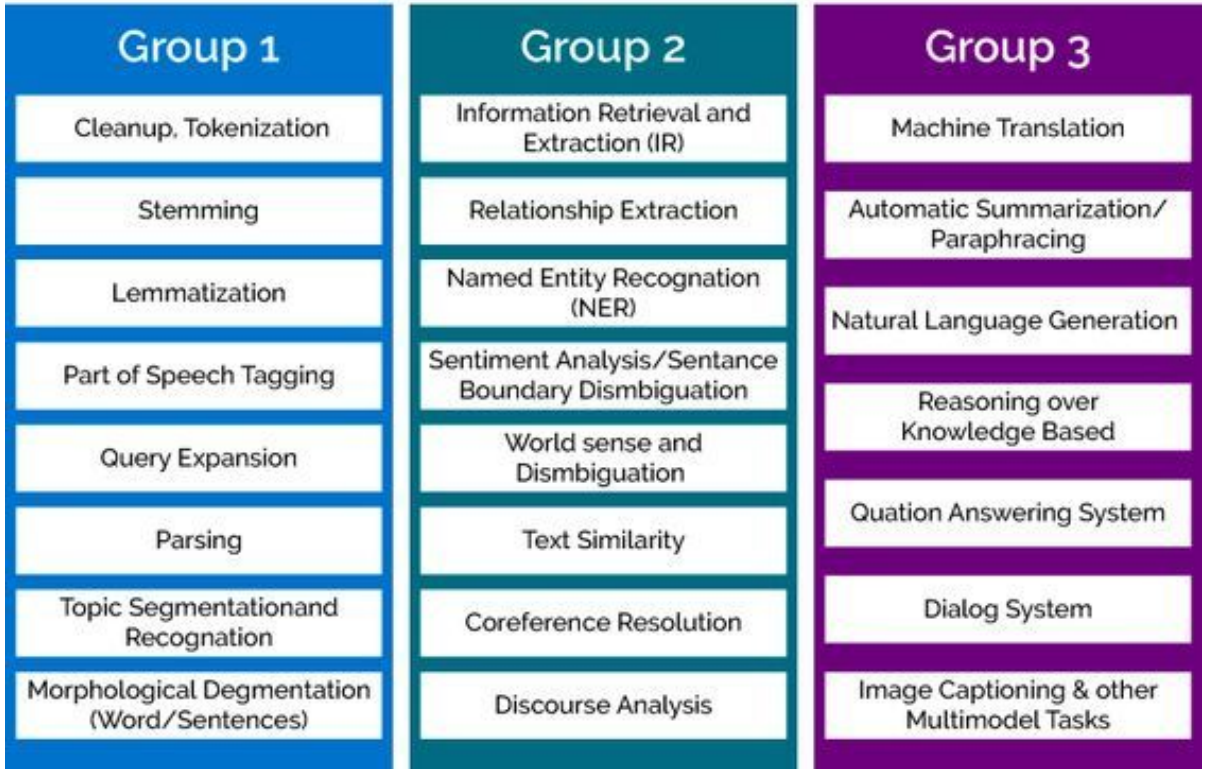


Why should you learn about NLP?

- You are probably reaping the benefits of NLP everyday without even realising it.
- When you use your phone's keyboard predictive text or Google Translate, you are effectively utilising NLP.
- Wouldn't you want to figure out how and why your phone's keyboard accidently suggests the correct word instead of the typo you made while typing?

Applications

- From predictive text to speech recognition, NLP helps us to digitally communicate better.
- We can explore trends in the way natural language is used
- Semantic analysis to understand emotions in text such as customer satisfaction
- Topic analysis to know what advertisements users are interested in based on messages sent
- Various other uses as well, such as recognition and prediction of diseases based on electronic health records and speech of patients.
- Even email filtering and classifying is achieved with NLP by analyzing the text to stop spams.
- Several other diverse applications exist



source

First Step: data processing

Learn ai

- So we have some text like this:

'Twas brillig, and the slithy toves

Did gyre and gimble in the wabe:

All mimsy were the borogoves,

And the mome raths outgrabe.

- How do we do useful NLP stuff with it?

Tokenization

Token: a string that contains one word or punctuation mark

Tokenization: the process of converting raw text to tokens

i.e. a string like

"the quick brown fox jumps over the lazy dogs."

can be tokenized to produce

```
["the", "quick", "brown", "fox", "jumps", "over",
"the", "lazy", "dogs", "."]
```

NLTK's tokenizer

Learn ai

- nltk is a great Python library that does a lot of NLP preprocessing for you

- Tokenization is super simple:

```
import nltk

tokenized_string = nltk.tokenize(string)
```

AI

Representing words

- Great, so we have a list of words, how can we represent them?

Ideas?

Representing words

- It depends a lot on what you want to do with the words

- One form of representation that has been popular recently is a **continuous vector space embedding**

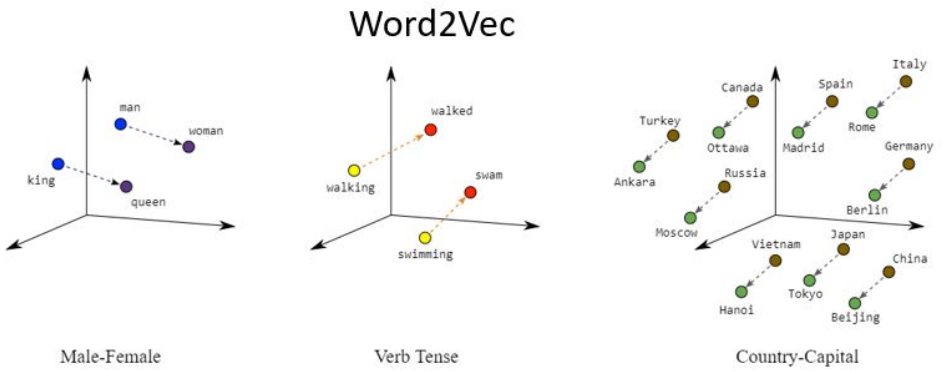
Continuous vector space embeddings

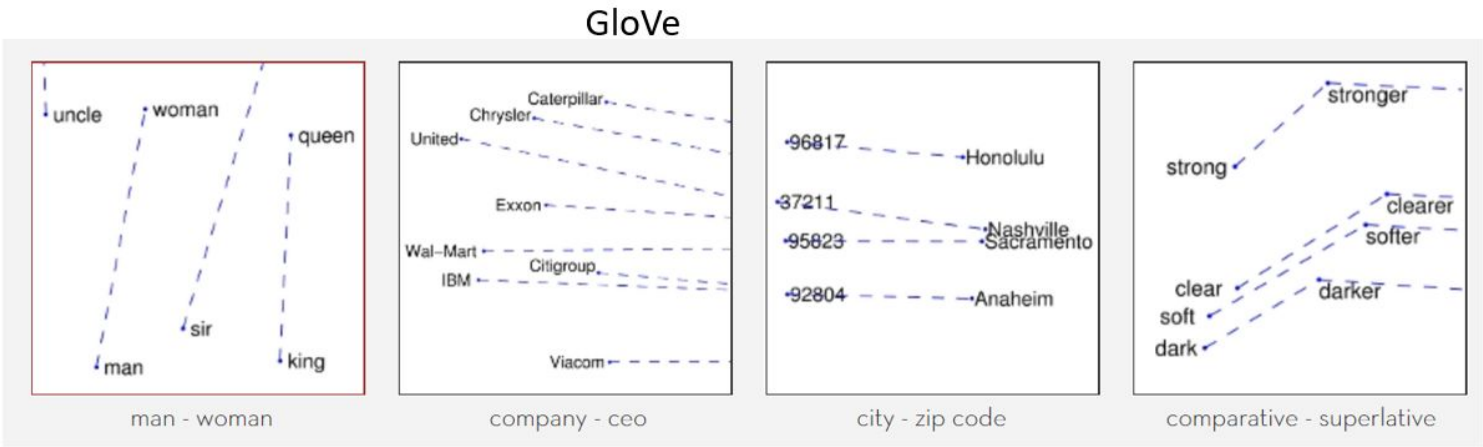
Learn ai

- Word embeddings are 100-300-dimensional...
- Some common training algorithms include
  - Word2Vec
  - GloVe
  - FastText
- These are learned from a large collection of text, called a *corpus*

Cool properties of word embeddings

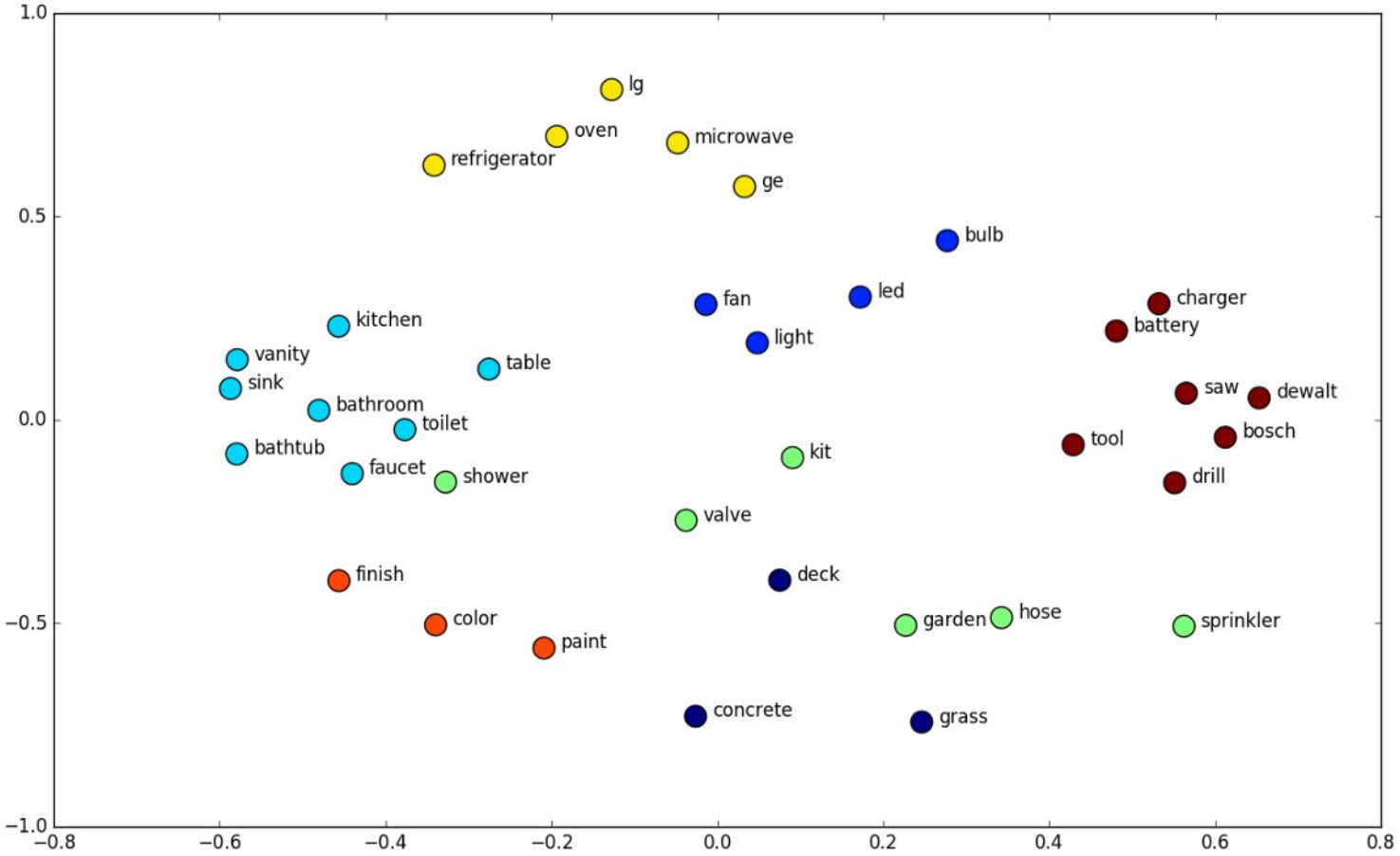
- Analogy: semantic and syntactic analogies can be represented as vector differences.





Cool properties of word embeddings

- Similarity: words that often share contexts are close together in the vector space



Uses for word embeddings

- Representing words in other machine learning models
  - We will see this in more detail when we get to machine translation
- Linguistics and cognitive science applications
  - i.e. using associations between words to predict implicit association test scores (Caliskan et al. 2017)

Using pre-trained word embeddings

Learn ai

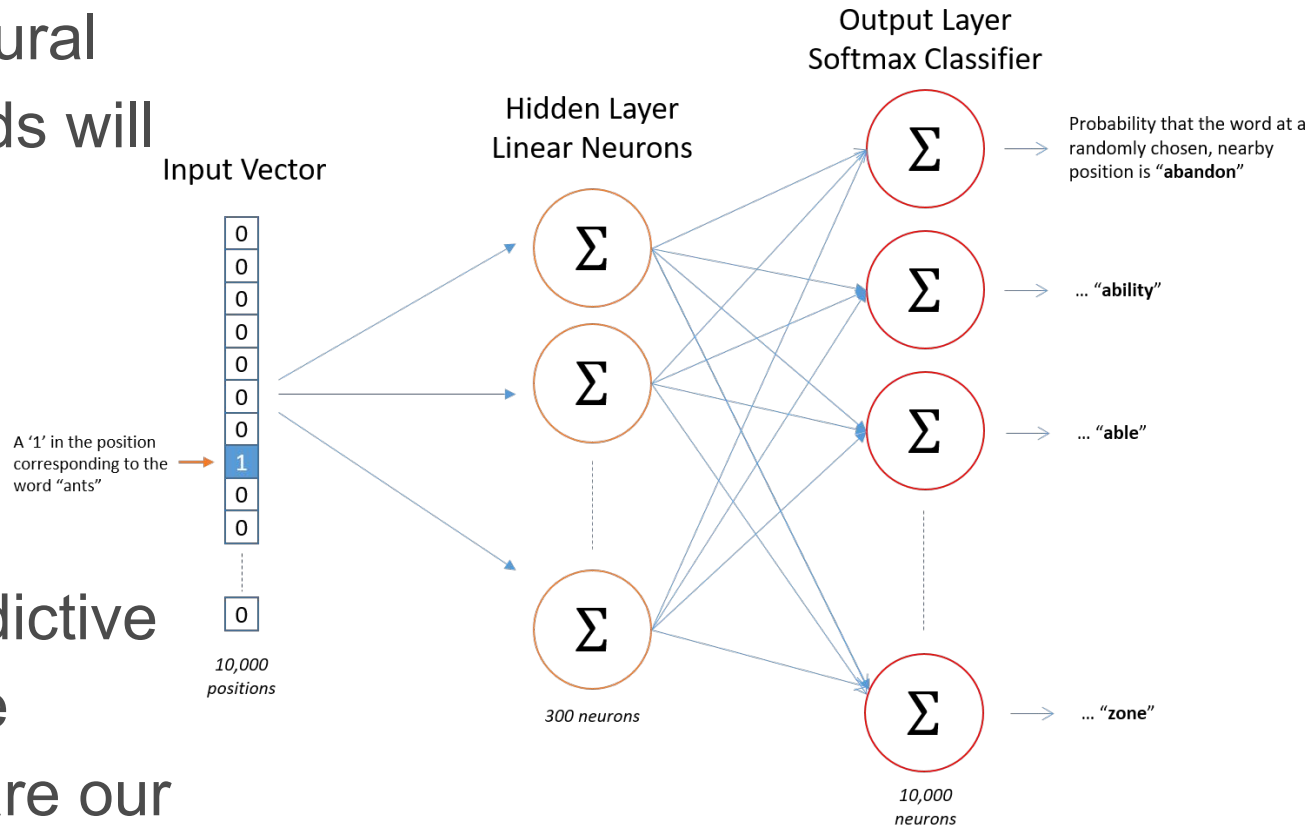
- `gensim` is a library that lets you train and use word embeddings
- Pre-trained word embeddings are available online, such as the Word2Vec embeddings trained on the Google News corpus

```
from gensim.models import KeyedVectors
wv = KeyedVectors.load_word2vec_format("filename'')
```

- Then use `wv` as a dictionary whose keys are words and whose values are vectors

Training word embeddings: Word2Vec

- Word2Vec (the skip-gram version) essentially trains a simple neural network to predict which words will surround a given word
- Words are one-hot encoded, hidden layer is n_words x n_dimensions
- Then, it throws away the predictive model and just holds onto the matrix of weights! The rows are our word embeddings



Machine translation

- The “Holy Grail of NLP”
- The task: take a sentence in one language as input, produce a sentence in another language as output
  - e.g.:

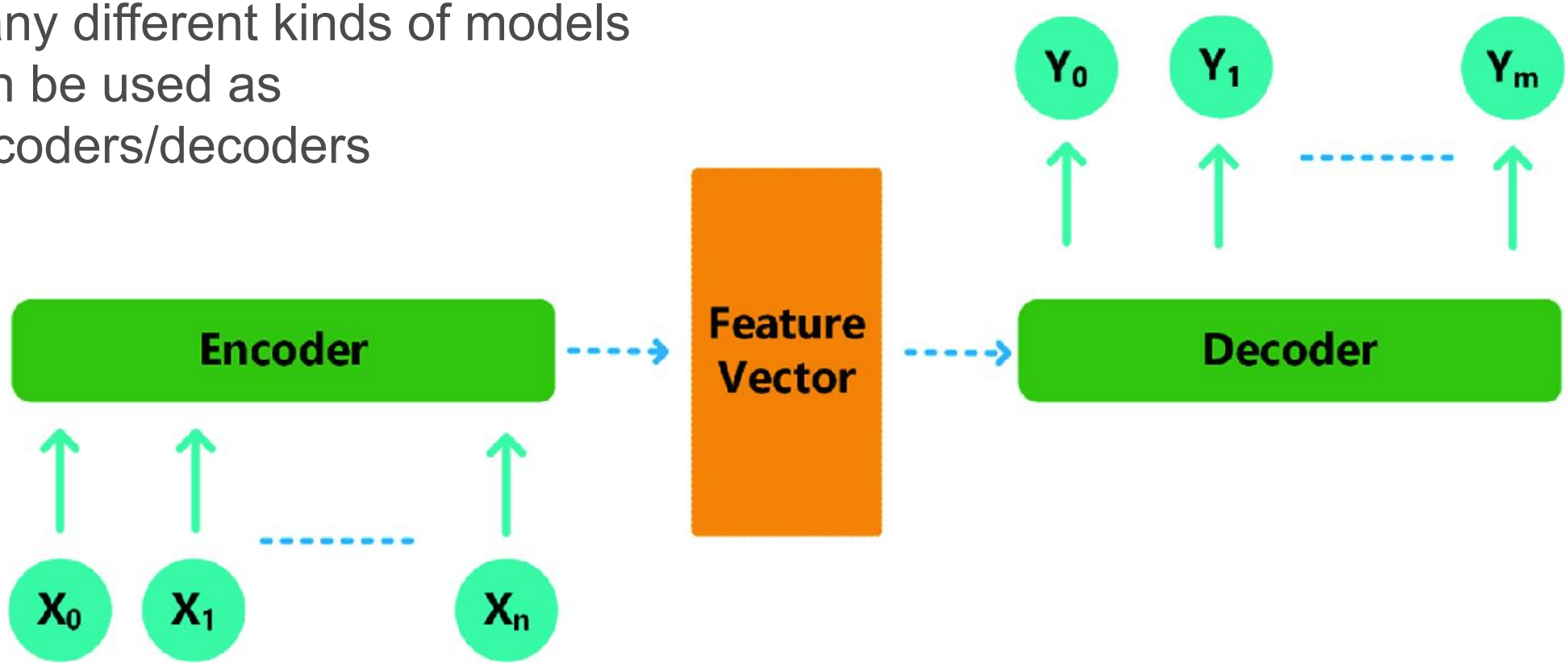
“My hovercraft is full of eels”

(Uzbek)

“Mening havo yostiqli kemam ilonbalig'i bilan to'lgan”

Encoder / Decoder architecture

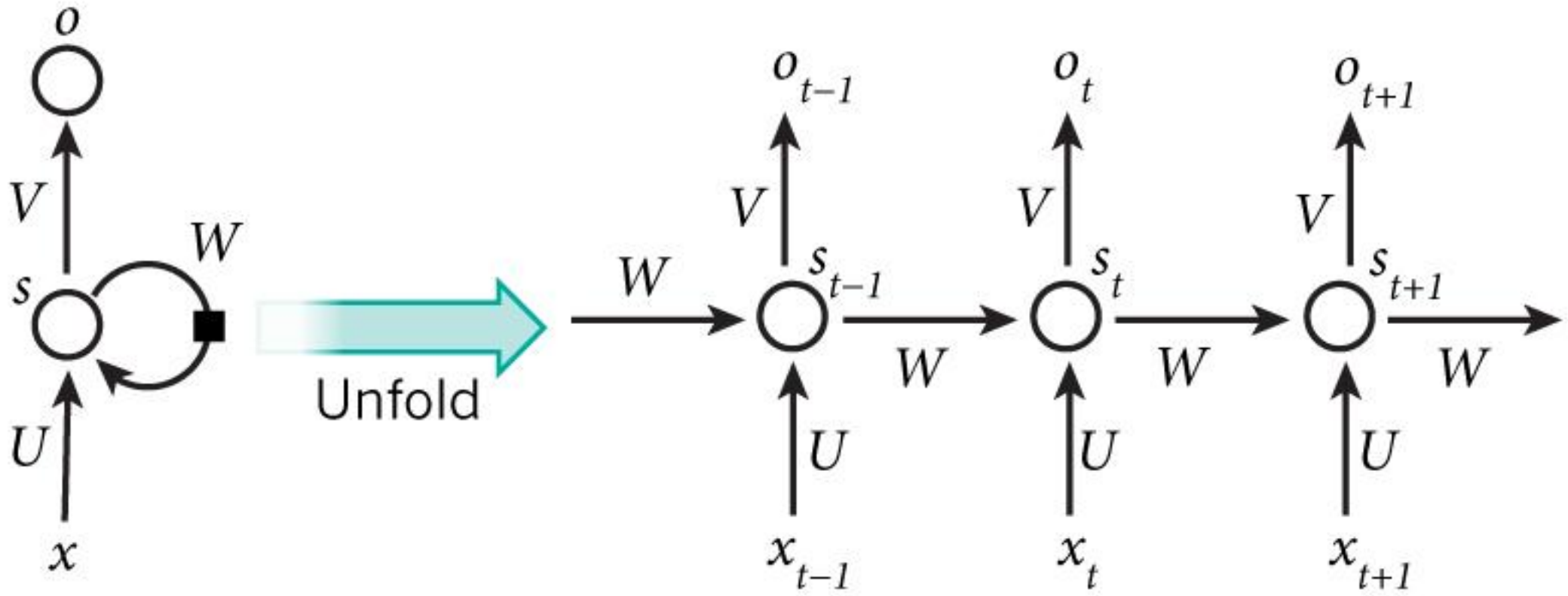
- A general framework for models that map sequences to sequences
- Encoder maps sequences to (“thought”) vectors
- Decoder maps vectors to sequences
- Many different kinds of models can be used as encoders/decoders



RNNs

- One common type of network used as encoders / decoders is the Recurrent Neural Network (RNN)
- Add a form of memory to neural nets
- Key idea behind RNNs: A neuron's activation at time `t` is an input to the same neuron at time `t+1`

3-neuron RNN unrolled over time



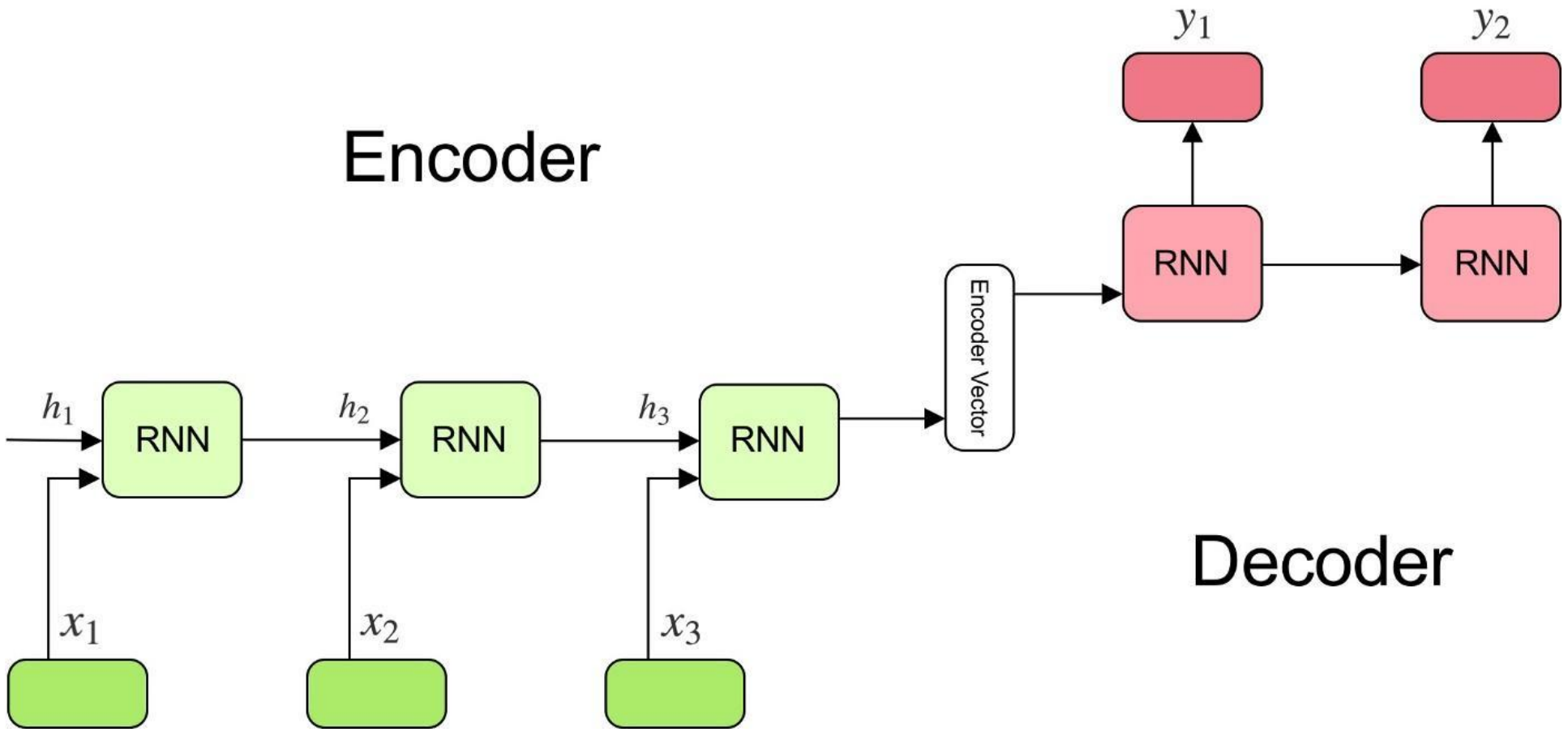
RNN-based encoder-decoder

Learn ai

- Encode using an RNN
  - Sentence feeds into an RNN, processed one token at a time
  - The state of the hidden layer after processing the last token is used as the encoded vector
- Decode using another RNN
  - Start off using the encoded vector as a hidden state
  - Predict the next character one at a time

AI

Encoder/Decoder visualization



Creating an RNN in Keras

Learn ai

- SimpleRNN layer class lets you easily create RNNs
- Can be used just like any other layer class (i.e. Dense)

```
model = Sequential([…, SimpleRNN(…), …])
```

References

- A simple introductory guide to NLP:

  A Simple Introduction to Natural Language Processing | by Dr. Michael J. Garbade

- Various applications of NLP:

  Your Guide to Natural Language Processing (NLP) | by Diego Lopez Yse

- Free resources to learn NLP for beginners:

  8 Free resources for Beginners to Learn Natural Language Processing

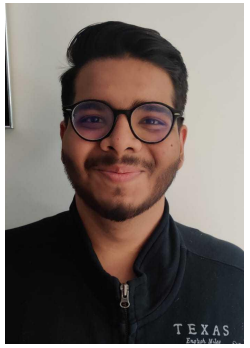
Contributors

Mustafa Imam

Addison Weatherhead

Isha Sharma

Kevin Zhu



Pranjal Kumar

Gilles Fayad