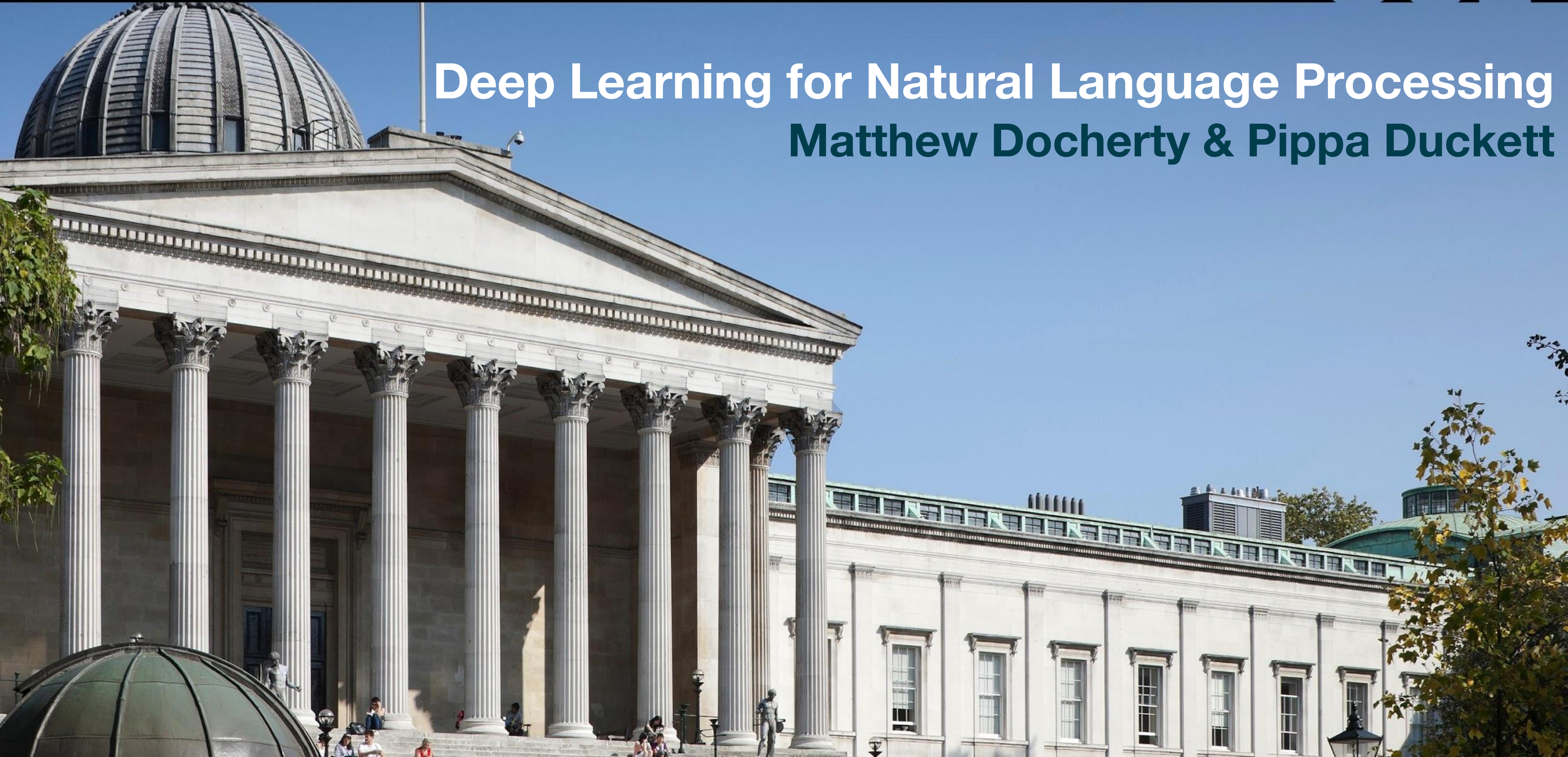
Deep Learning for Natural Language Processing

Matthew Docherty & Pippa Duckett



Who are we?

Matthew Docherty

1st year CDT Data Intensive Science student

Main research: Bayesian deep learning and probabilistic inference for Cosmology

NLP experience: Group project using transformers to improve ASOS search functionality

mdochertyastro.com

Pippa Duckett

1st year CDT Data Intensive Science student

Main research: Tracking and Tagging particles at the LHC (CERN)

NLP experience: Group project using transformers to improve ASOS search functionality

Structure for today

4 I’s:

- Informal
- Interactive
- Introductory
- Intuition-focused → no equations :))

§1 - Text Normalisation

§2 - Basic Text Representation

§3 - NLP Models for sequential Data

NLP Overview

What is NLP?

Train models to understand text and spoken words in much the same way human beings can.

What does NLP require

- Processing sequential data
- Representing text of varying length as a vector of fixed length for linear algebra

NLP Examples

Many NLP problems but we are looking at focusing on 2 main types of problems, distinctive by the form of their input and output data

Seq2Vec:

Document Classification - Determining topic from text - Fake news (NLP Group at MIT)

Sentiment Analysis - Determining tone from text - Tripadvisor reviews

Image Generation - Generate an image from input text query - OpenAI DALLE-2

Seq2Seq:

Document Translation - Transform text from one language to another - Google translate

Question & Answer - Conversational answers to user queries - Amazon Alexa

§1 - Text Normalisation

A method of reduction…

- TYPE: *Data preprocessing*
- GOAL: *Reduce Dimensionality & Complexity of Vocabulary*
- WHY: *Lightweight faster computations*
- TRADE-OFF: *Sacrifice information (not always applicable)*

Text Normalisation Techniques

Stop Word Removal

Remove bland and ubiquitous words that are not identifiable and specific to topics

Examples:

"I, The, To, And,…"

Stripping

Stemming

Fast naive rule-based chop off suffix

Low accuracy

Creates non-existent words

Example: ***"studies → studi"***

Lemmatisation

Slower dictionary-based morphology to lemma of word

Higher accuracy

Example ***"studies → study"***

To normalise or not to normalise…

Whether to apply text normalisation techniques is very example-specific and should be thought through

General rule:
Do text normalisation of we only want general idea of sentence but not if we want more tone-based or semantic meaning of the text

<u>seq2vec</u>
Document Classification → ✅
Sentiment Analysis → ❌

<u>seq2seq</u>
Conversational QA → ✅
Translation → ❌

Movie review: "The movie was not good at all."

Text after removal of stop words: "movie good"

§2 - Basic Text Representation

Representation Problem Setup:

- Collection of documents of text = *Corpus*
- All the words across all documents = *Vocabulary*
- Want a want to represent each document as fixed vector

Method 1: One-hot Encoding

Bag of Words One-hot Encoding

- Vocab = entire english language ~*100,000 tokens*
- Dimension of document vector = dim of vocab
- Coefficient of vector dims = number of times word appears in doc
- Possible to do sparse linear algebra in 100,000D

TF-IDF One-hot Encoding

- Coefficient of vectors = formula inverse scaled by number of docs word appears in

Limitations needing solved:

Dimensionality → ❌
Semantic Meaning → ❌

Method 2: Word Embeddings

	queen	*king*	*girl*	*boy*	*rose*	*apple*
gender	0.95	-0.93	0.98	-0.98	0.78	0.01
royal	0.98	0.96	0.11	0.11	0.56	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
flower	0.21	0.03	0.62	-0.11	0.97	0.33
food	0	0	0	0	0.66	0.99

Limitations needing solved:

Dimensionality → ✅

Semantic Meaning → ✅

Word2Vec

Mikolov+ (2013)

Use neural network models trained on fake task to obtain the word vectors.

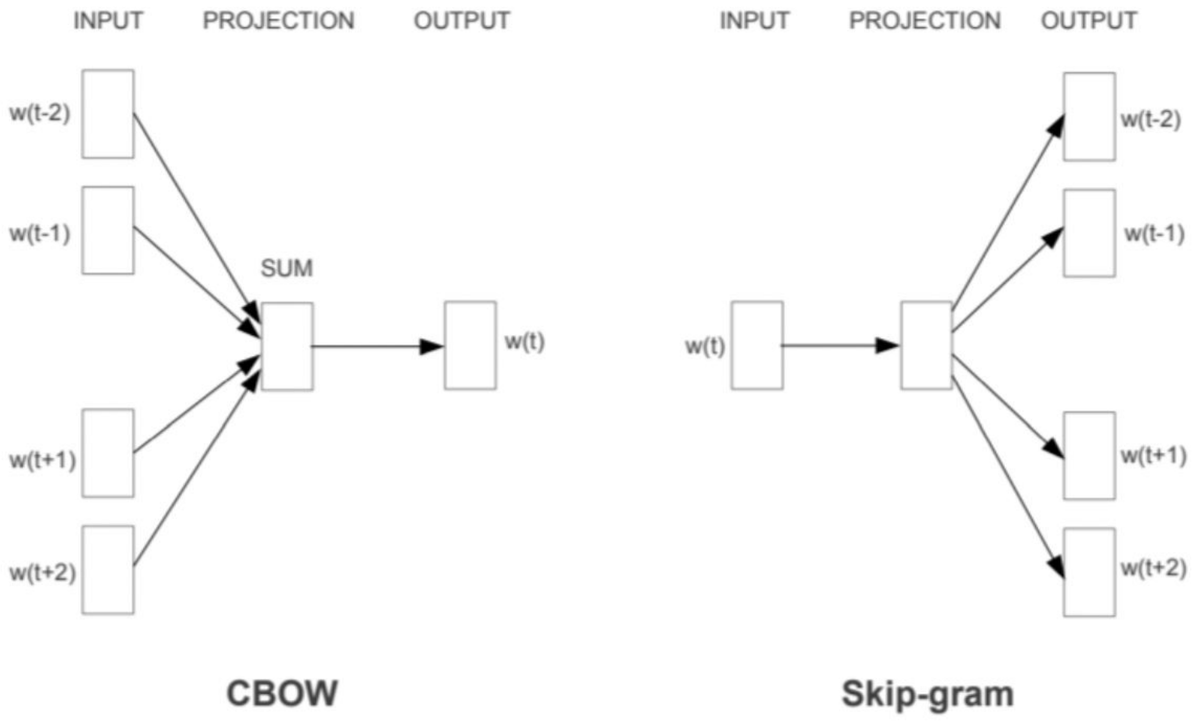
Algorithms for producing word embeddings:

Continuous Bag of Words

- predict target using context

Skip-Gram

- predict context words using target

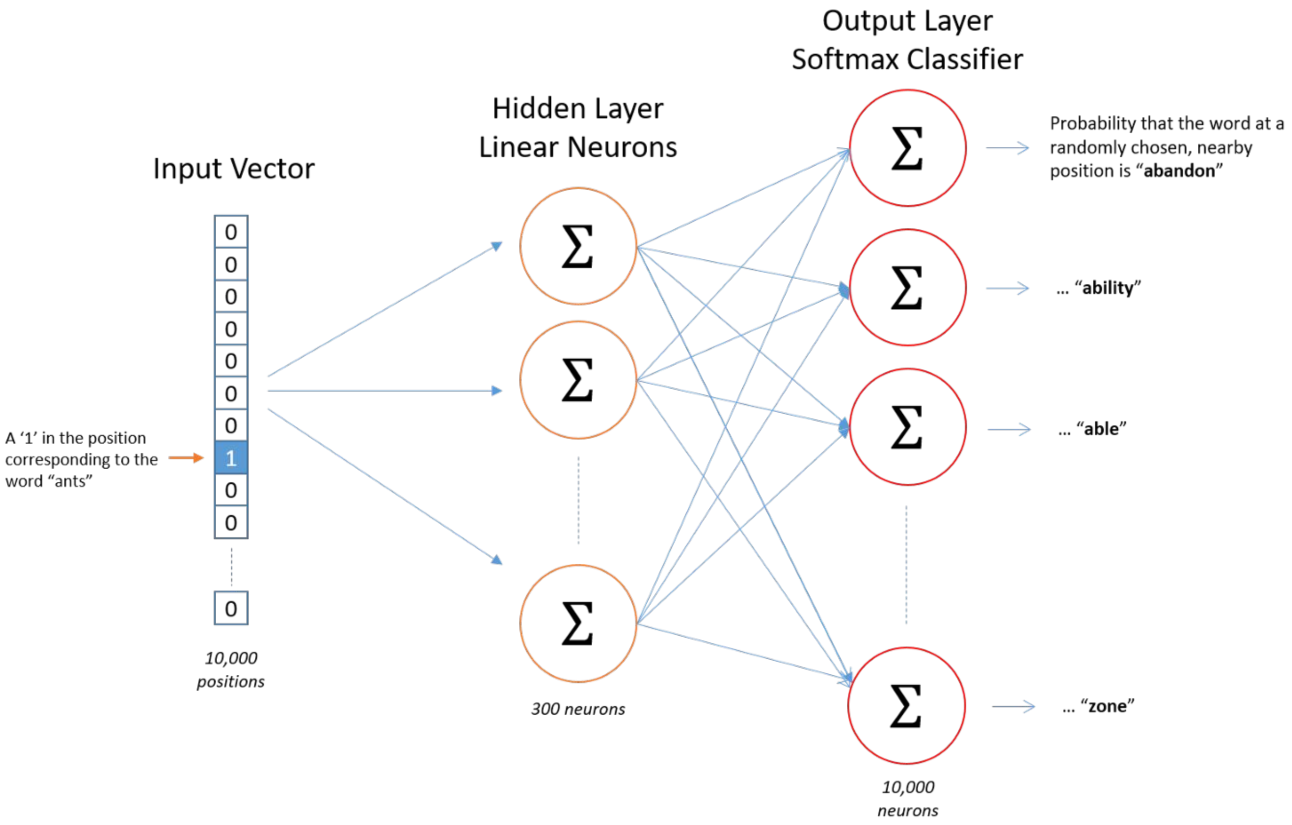


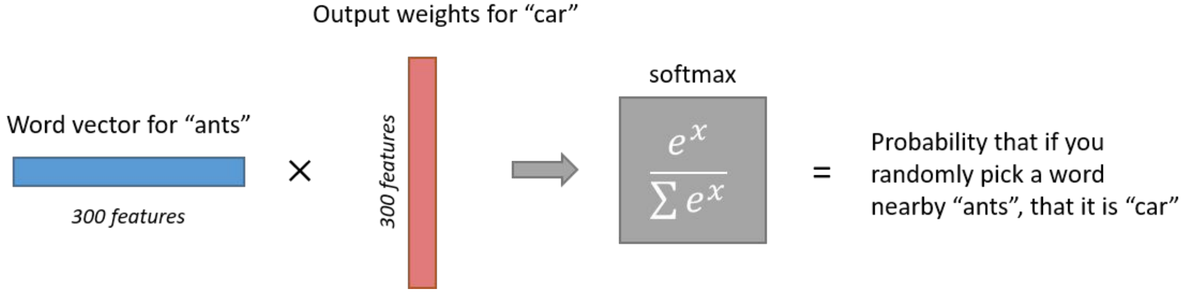
Chablani TDS Word2Vec (skip-gram model): PART 1 - Intuition (2017)

Skip-Gram

Try to predict the context words using the main word:

- Neural network one hidden layer
- Input main word in one-hot encoding
- Weight matrix transforms input into hidden layer
- Outputs to be predicted are the context words (one-hot encoded)
- Weight vector for each output neuron is multiplied against the word vector from the hidden layer
- The model learns by trying to predict the context words.

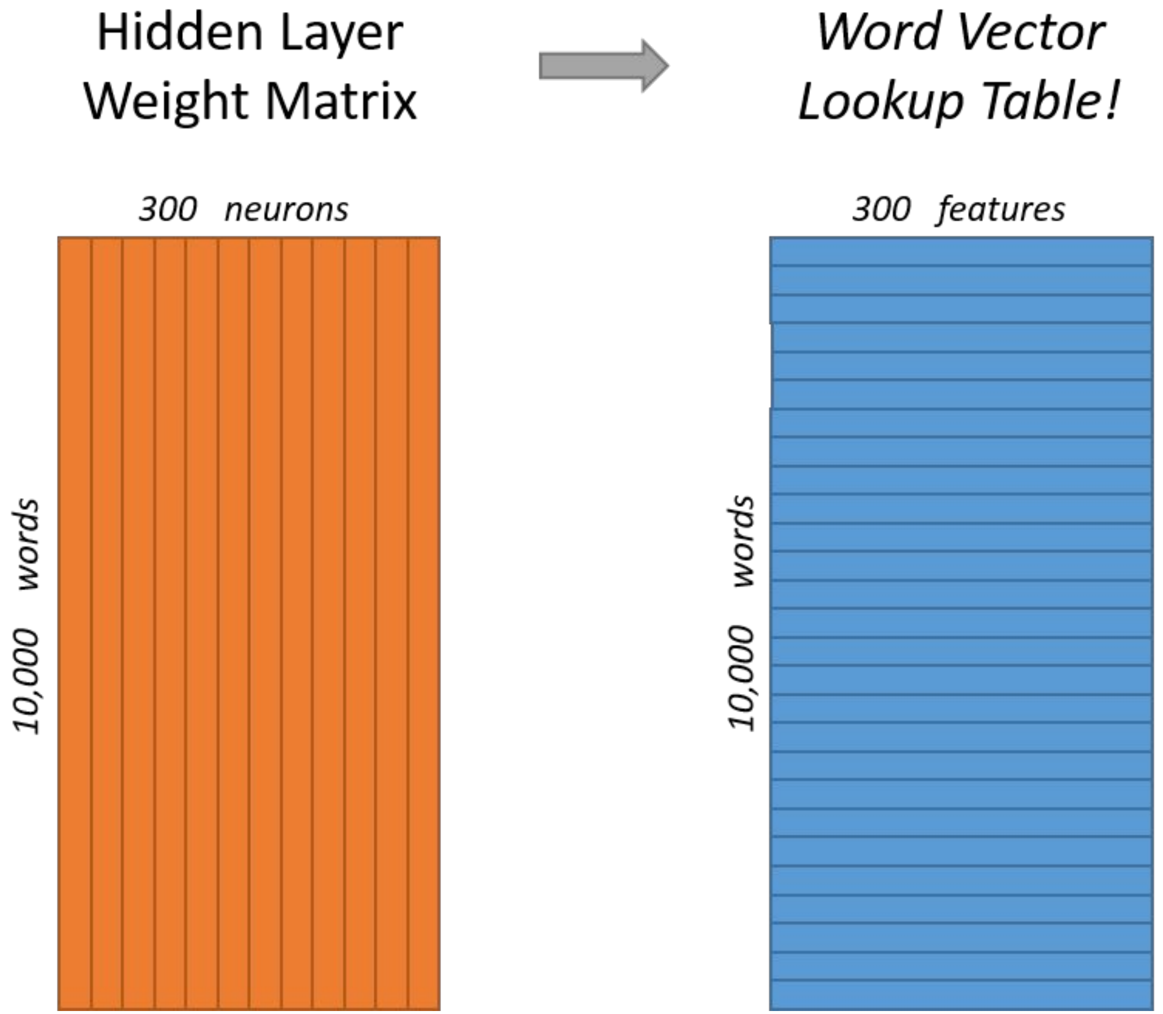




Chablani TDS Word2Vec (skip-gram model): PART 1 - Intuition (2017)

Skip-Gram

- Learnt weight matrix connects the input to the hidden layer
- From this weight matrix the embeddings can be obtained



“You shall know a word by the company it keeps!”

Prof. John Firth
Phonetics dept. UCL
[1965-71]

§3 - NLP Models for Sequential Data

Recap basic word representation methods

In previous section we saw 2 naive approaches for representing text at word level:

1. One-hot Encoding → *Dimensionality* ❌, *Semantic Meaning* ❌
2. Naive sum of word embeddings → *Dimensionality* ✅, *Semantic Meaning* ✅

For more complex/interesting problems, word-level representations themselves have 2 more inherent limitations, lacks ability to encode:

1. *Positional Information* ❌
2. *Contextual Information* ❌

Sequential Motivation - limitations of word-level analysis

Encoding Positional Information

Order matters…

Work to Live…

Live to Work…

Encoding Contextual Information

Context matters…

"If you want to learn an effective full body workout routine optimized for muscle growth, then you need to read this article…"

"Wine body breaks down into three categories: light body, medium body and full body, and a good way to think about the difference between them is the way skim milk, whole milk and cream feel in your mouth…"

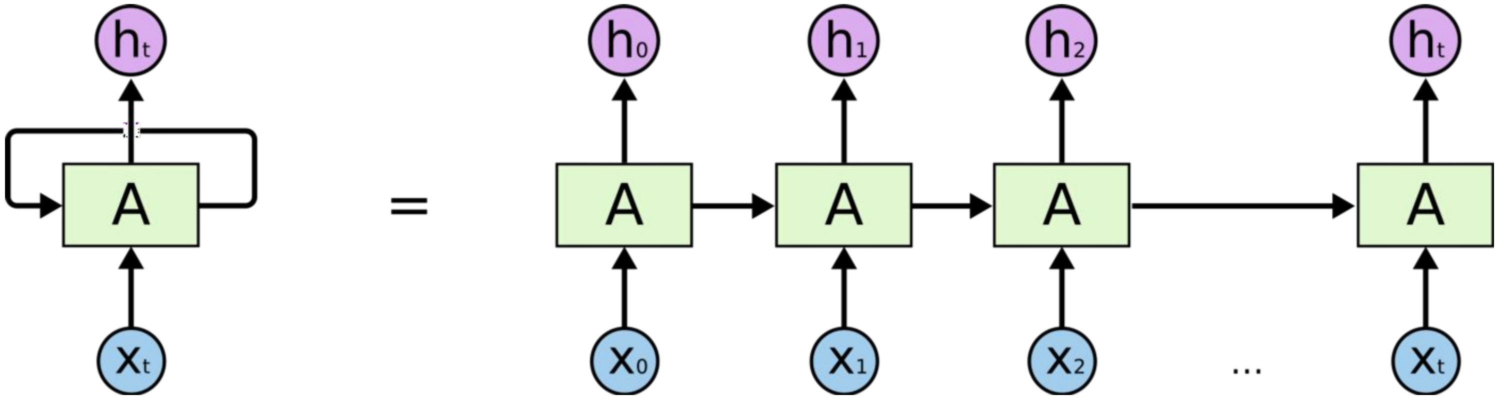
"'Double Ironfist' Fullbody is a Marine lieutenant commander serving under Rear Admiral Hina. He was a Marine Headquarters lieutenant at the start of the series, but was demoted to seaman recruit as a condition when he helped the former Black Cat pirate, Jango, get acquitted for his past crimes (in the anime, it was after getting into a fight with Sanji on the Baratie)..."

Solve Limitations

Pivot from word-level representations to sequential representations using RNNs…

RNNS

Temizel+ (2020)



- Family of models for processing sequential data (what CNN is to 2d image data)
- Sequential input data = one token per time step
- Network parameters A shared across each time step crucial for generalising to inputs of varying length (left diag ^^^)
- Sequential approach such that the output of a time step is based on activation between network weights and context from the output of all other time steps (right diag ^^^)

Limitations:

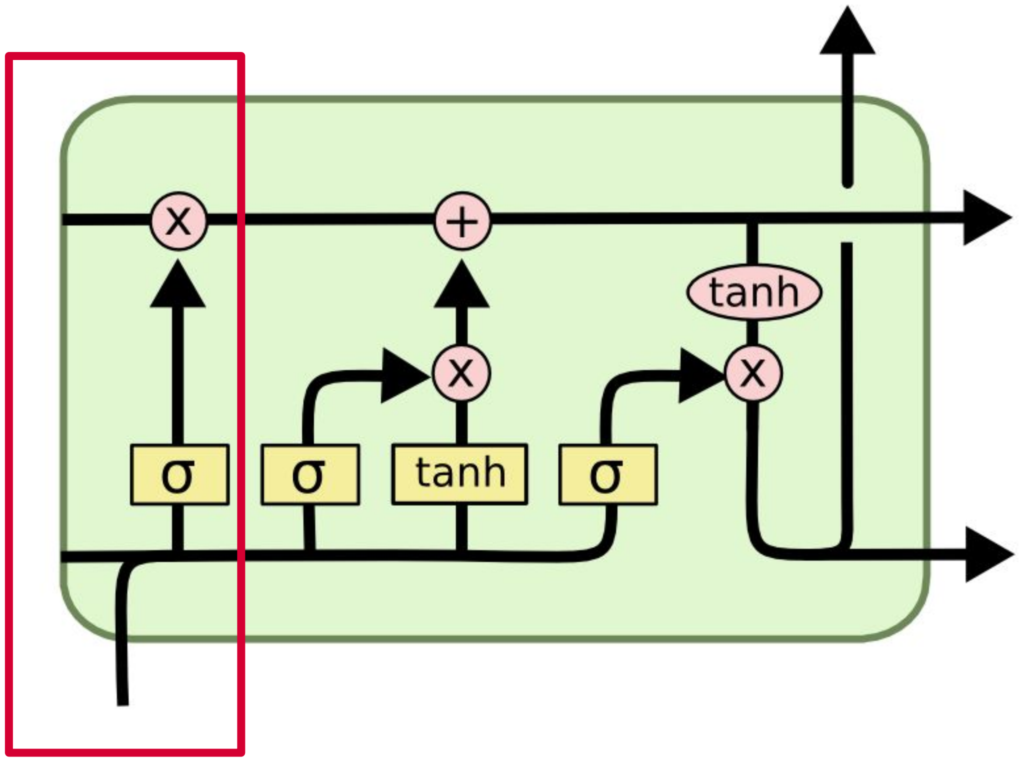
- Sequential models are slow to train (need to process each time step)
- Basic RNNs specifically are hard to train due to vanishing gradients from short memory

Limitations solved:

Positional → ✅
Contextual → ⌛

Long Short Term Memory RNNs

Klinkachorn+ (2019)



- LSTMs improves memory length by allowing information that is not important to be forgot.
- Typical network in a single time step A is show in diag, keeping it intuition-focused only concerned with the forget gate
- Allows useless info to be dropped and so giving higher weight to time step outputs that are crucial for context

Limitations:
- Sequential models are still slow to train (need to process each time step)
- Vanishing gradients are circumvented using LSTMs

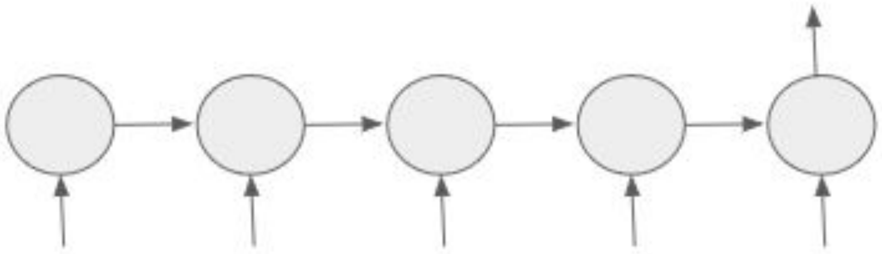
Limitations solved:

Positional → ✅
Contextual → ⌛⌛⌛

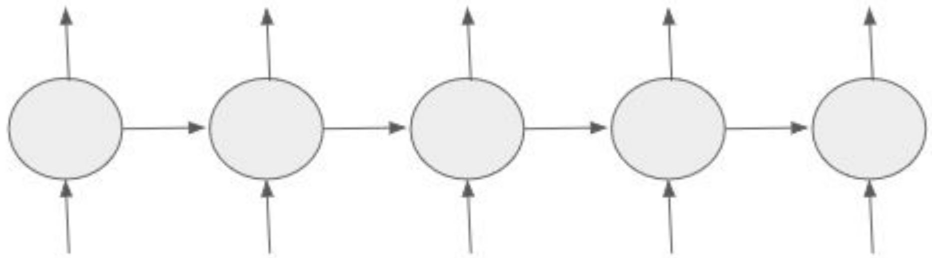
RNN Architectures

Arbel (2019) DDI: Attention in RNNs (2019)

Seq2Vec

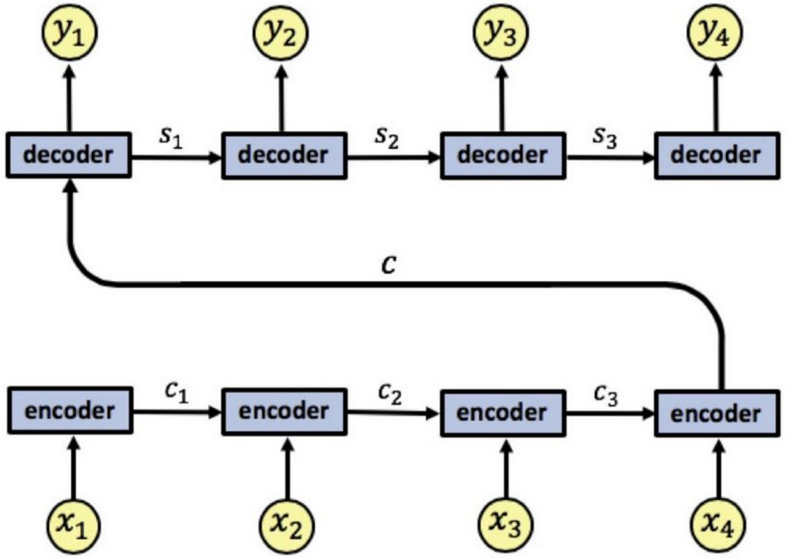


Seq2Seq



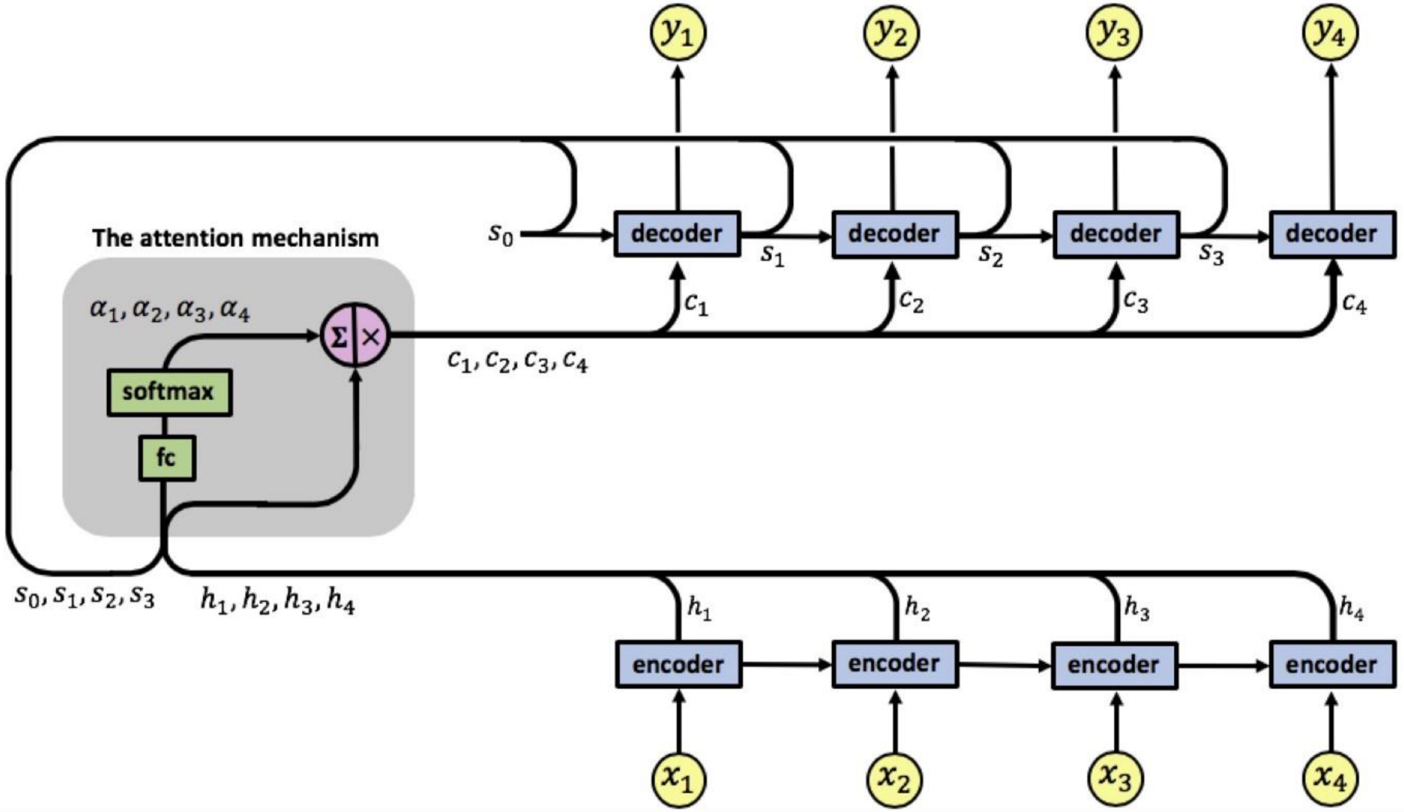
Basic

Encoder-Decoder



Attention Mechanisms in RNNs

Arbel (2019) DDI: Attention in RNNs (2019)



- Attention mechanism between encoder-decoder, allows **all encoder time steps** to be input to decoder
- **Improves memory** by by applying attention weights to every time step based on **contextual importance**
- *Limitation* - **RNNs are still slow to train** due to sequential nature of data (can't parallelise)

Limitations solved:

Positional → ✅
Contextual → ✅

Vaswani+ (2017)

Transformers

Designed to process sequential input data

Unlike RNNs, transformers process the entire input all at once

Key Components:

Positional Encodings

- inject positional information into the embeddings
  → can process entire sequences in parallel
- scale the speed and capacity of sequential models

Self Attention

- track the relations between words across very long text sequences

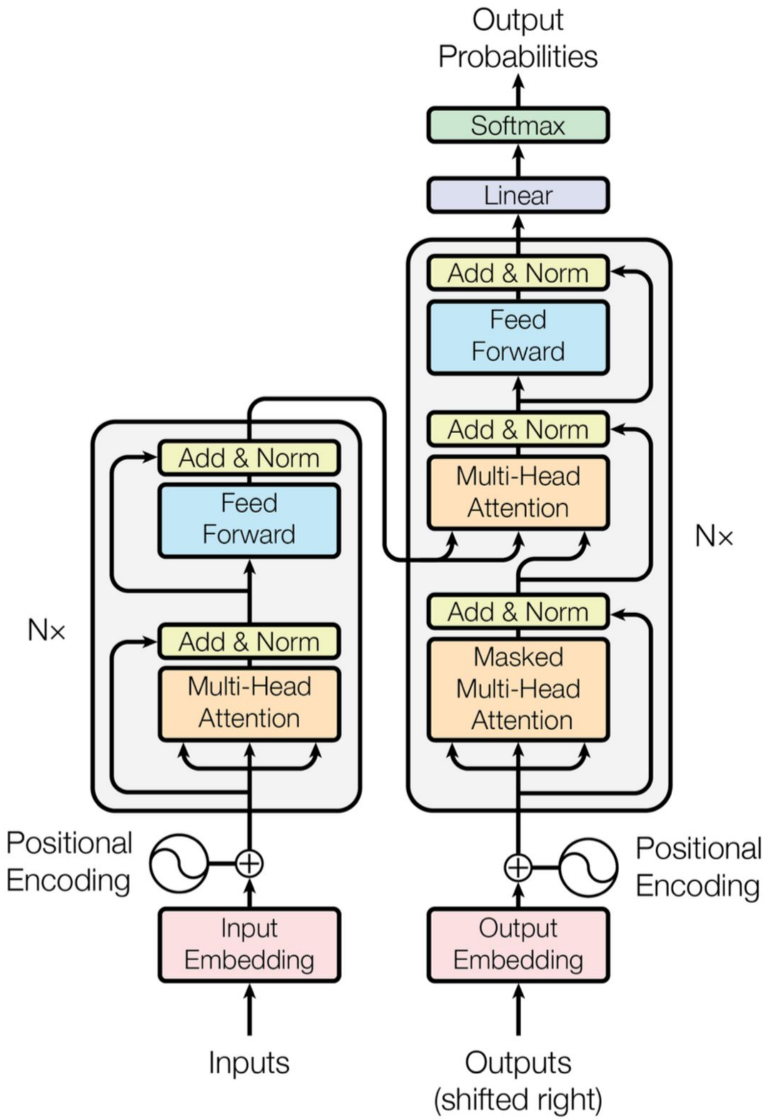


Figure 1: The Transformer - model architecture.

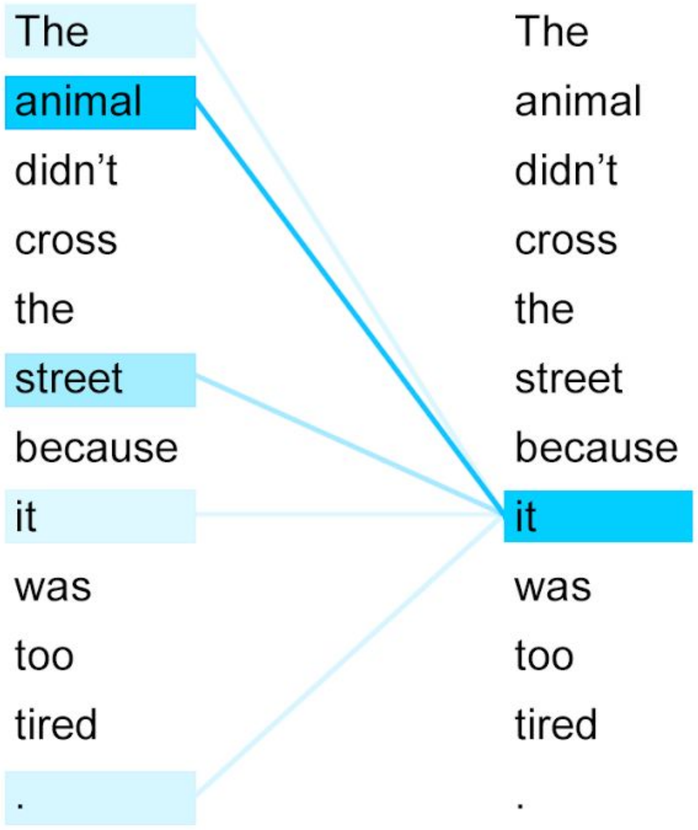
Uszkoreit (2017) Transformer: A Novel Neural Network Architecture for Language Understanding

Self Attention

- Allows a neural network to understand a word in the context of the words around it
- How much attention to give each other word when considering specific word
- High attention score for two words that are relevant to each other in the sentence.

Self-attention help neural networks:

- disambiguate words
- do part-of-speech tagging
- learn semantic roles



Ben Dickson (2022) Can large language models be democratized?

LLMs

- Additional training parallelization of transformers
  → allows training on larger datasets
  → large pretrained transformer models
- Trained on enormous amounts of data
  (10s of gigabytes)
- Hundreds of millions of parameters

Ben Dickson (2022) Can large language models be democratized?

LLMs

Why They're Good:

- Single model used for lots of tasks

  (QA, summarization, text generation, translation)

- Performance scales with parameters and data

- Good predictions with just a few labeled examples (few-shot learning)

LLMs - Transfer Learning

Pretraining

- train a model from scratch
- weights randomly initialized, training starts with no prior knowledge
- requires v large corpus of data
- training can take up to several weeks

Fine-tuning

- additional training on pretrained language model with a dataset specific to your task
- Eg. leverage a pretrained model trained on the English language → fine-tune it on an arXiv corpus → science/research-based model.

Why Fine-tuning (Transfer Learning):

- Take advantage of knowledge acquired by the initial model during pretraining
- Pretrained model was already trained on lots of data

  → fine-tuning requires less data for decent results

  → lower time & resources to get good results

  → lower financial, & environmental costs

  → achieve better results than training from scratch if have limited data/resources

LLMs - Examples

Model	*Company*	*Main Corpus*	*Training Task*	*Parameters*
USE	Google Research	SNLI	SR	85m
BERT-L	Google AI Language	Wiki	QA	340m
OPT175B	Meta AI Research	Common Crawl	QA	175B
GPT-3	OpenAI	Common Crawl	Next Word Prediction	175B

Summary

- Introductory overview of NLP and deep learning techniques → real world examples

- Introduced optional data-preprocessing → reduce dimensionality

- Introduced naive word representations **one-hot encoding** and ***embeddings*** which solve *dimensionality* and *semantic* limitations → cannot encode *positional* or *contextual* information

- Introduced **RNNs** which solve *positional* and *contextual* limitations using **LSTMs** and/or **attention mechanisms** → slow to train due to sequential nature

- Introduce **transformers** that overcome this sequential limitation → multi-headed self-attention (parallelisable)

- This training efficiency of transformers allows them to be used within **large language models** with hundreds of billions of parameters which can be used to solve hard NLP tasks leveraging t*ransfer learning* → examples of current SoTA LLMs