Applied Deep Learning

Dr. Philippe Blaettchen
Bayes Business School (formerly Cass)

www.bayes.city.ac.uk

Learning objectives for today

Goals: Introduce the broad field of NLP and wrap up the module

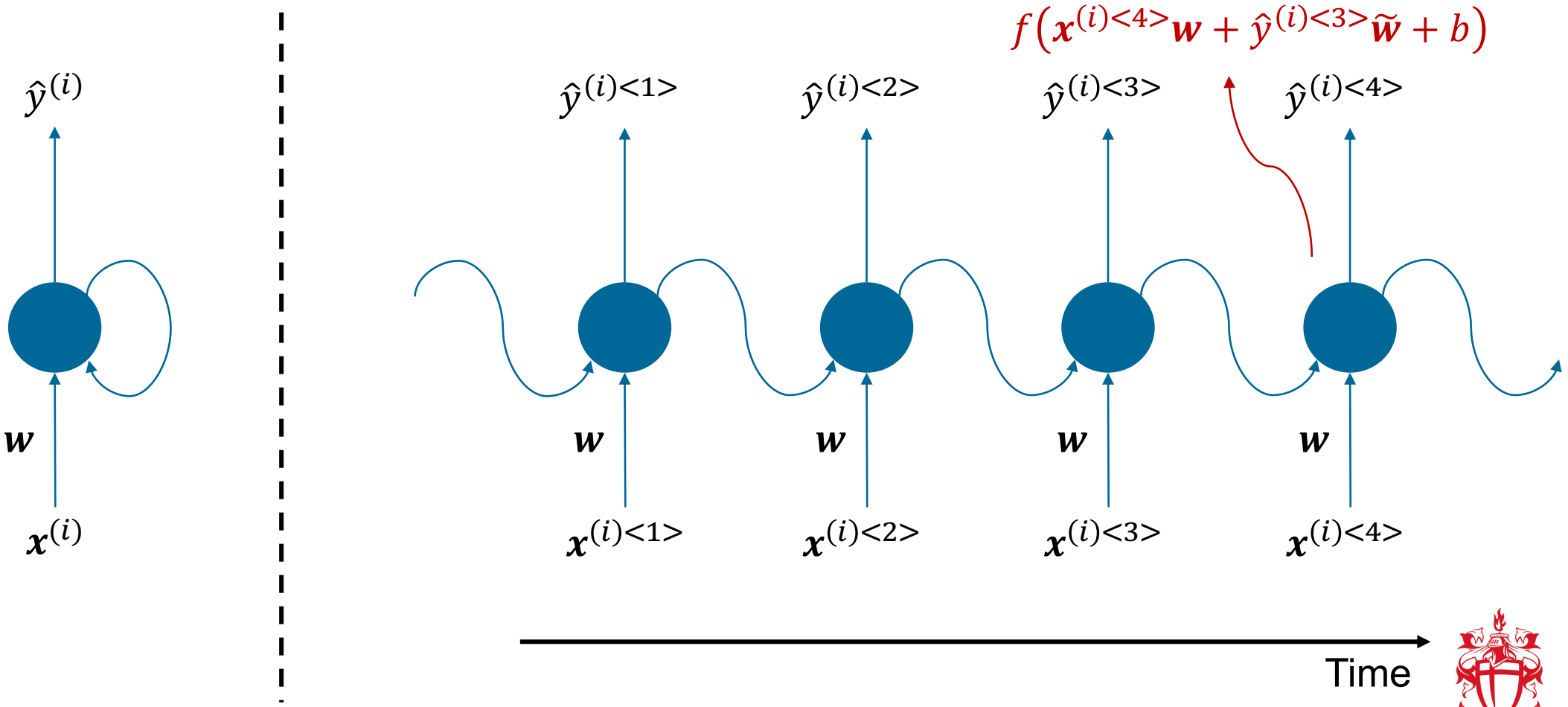
- Understand the basics of NLP and the difference between bag-of-words and sequence models
- Understand the limits of RNNs in analyzing text data, and how attention can be used to improve RNNs (and other networks)
- Learn about transformer architectures and about generative models mor generally

How will we do this?

- We start where we left off with RNNs, then try out a basic NLP examples with a recurrent network
- We then introduce the attention mechanism and the transformer architecture as the state-of-the-art tool to deal with NLP tasks
- We briefly go through generative deep learning models, before wrapping up the module

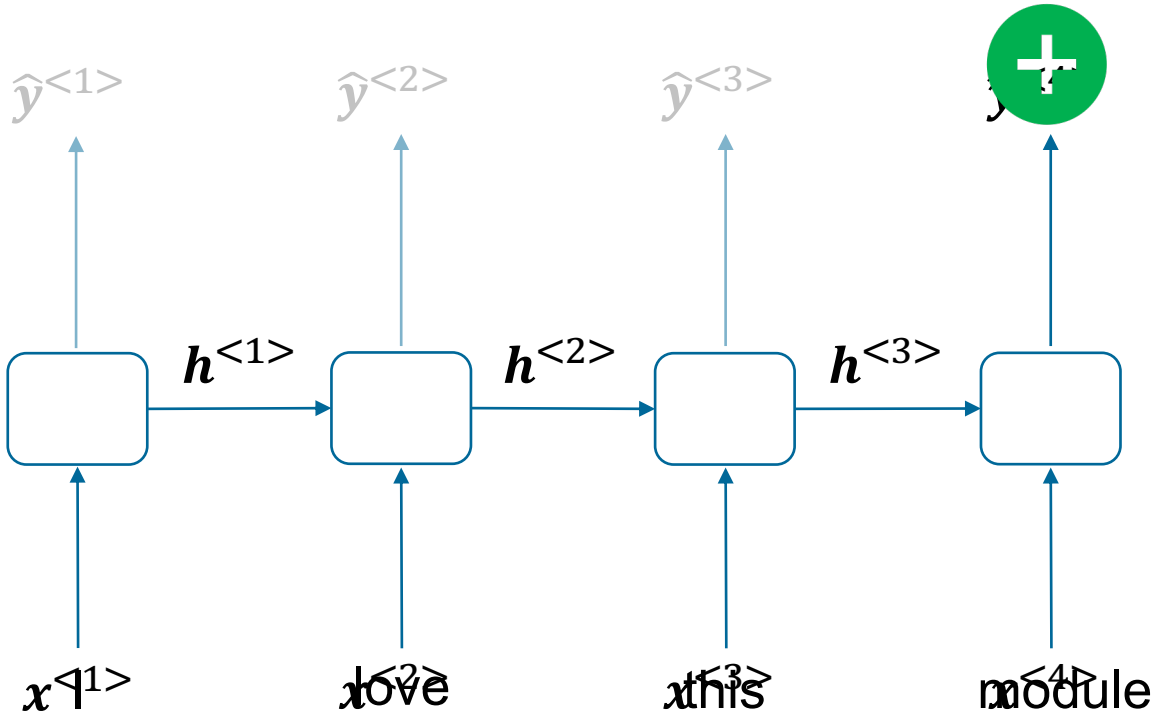
RNNs and their different applications

A recurrent neuron



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Sequence-to-vector networks

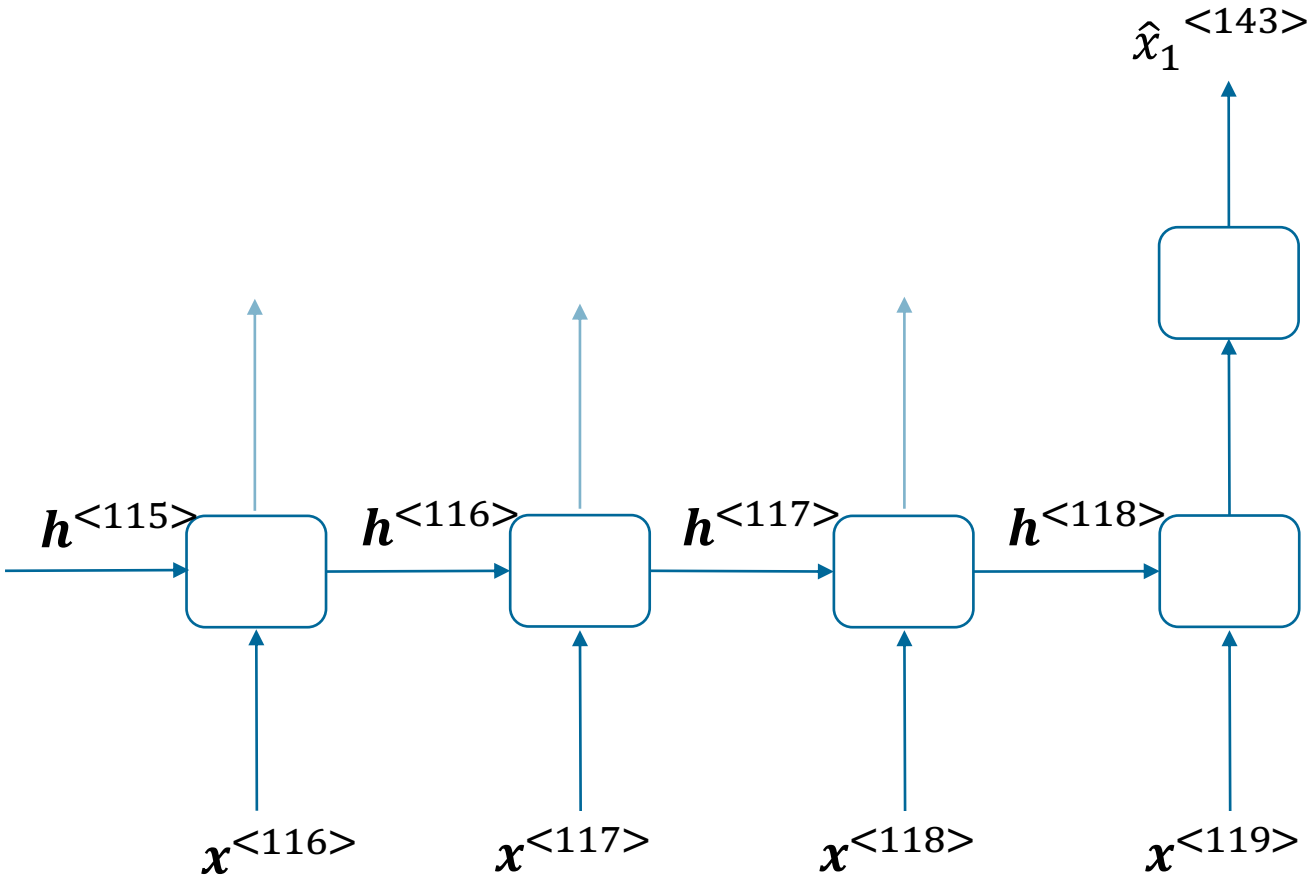


For example:

- Video activity recognition
- DNA sequence probing
- Sentiment classification

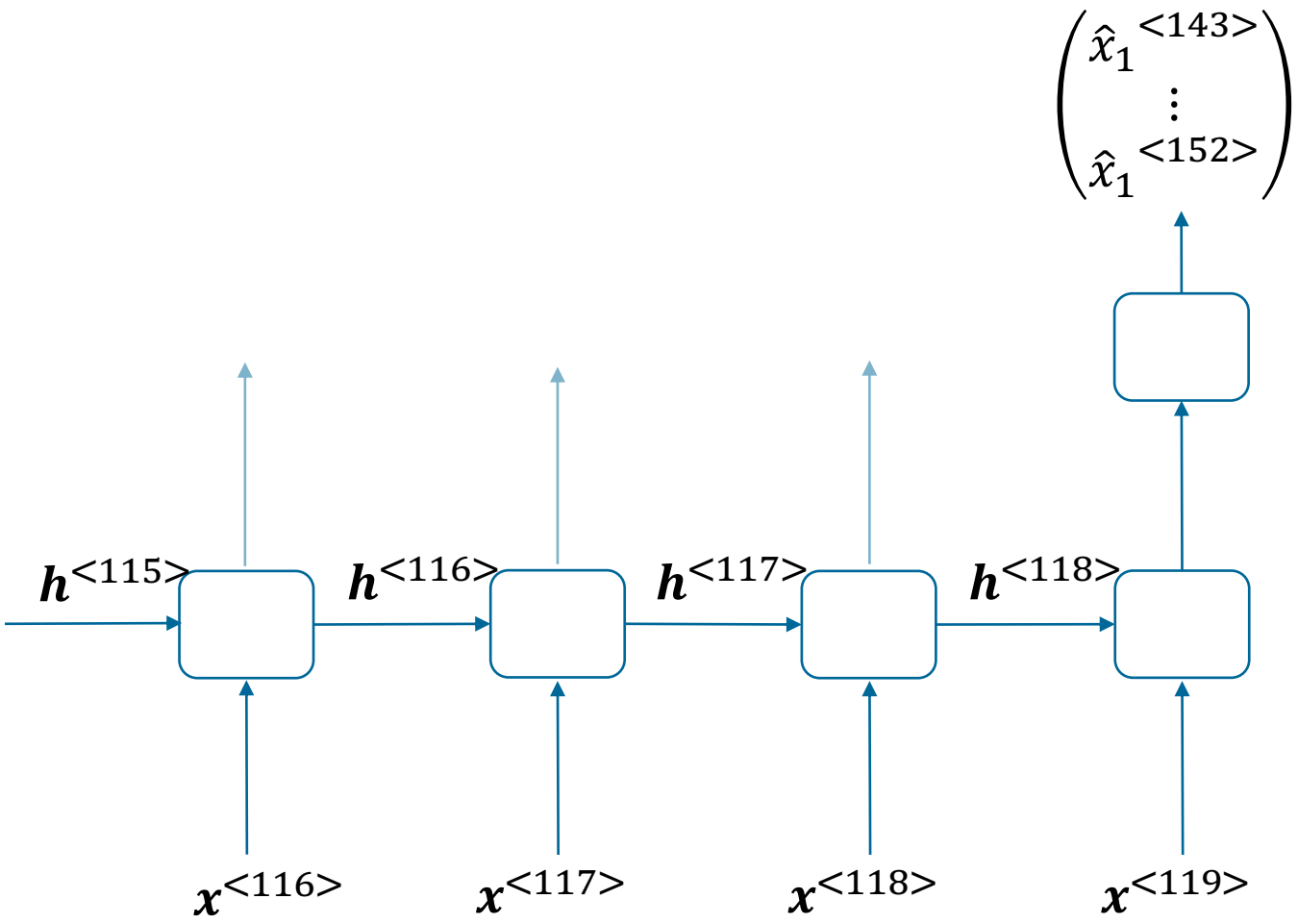
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Sequence-to-vector networks: our example



BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Sequence-to-vector networks: multiple time steps ahead



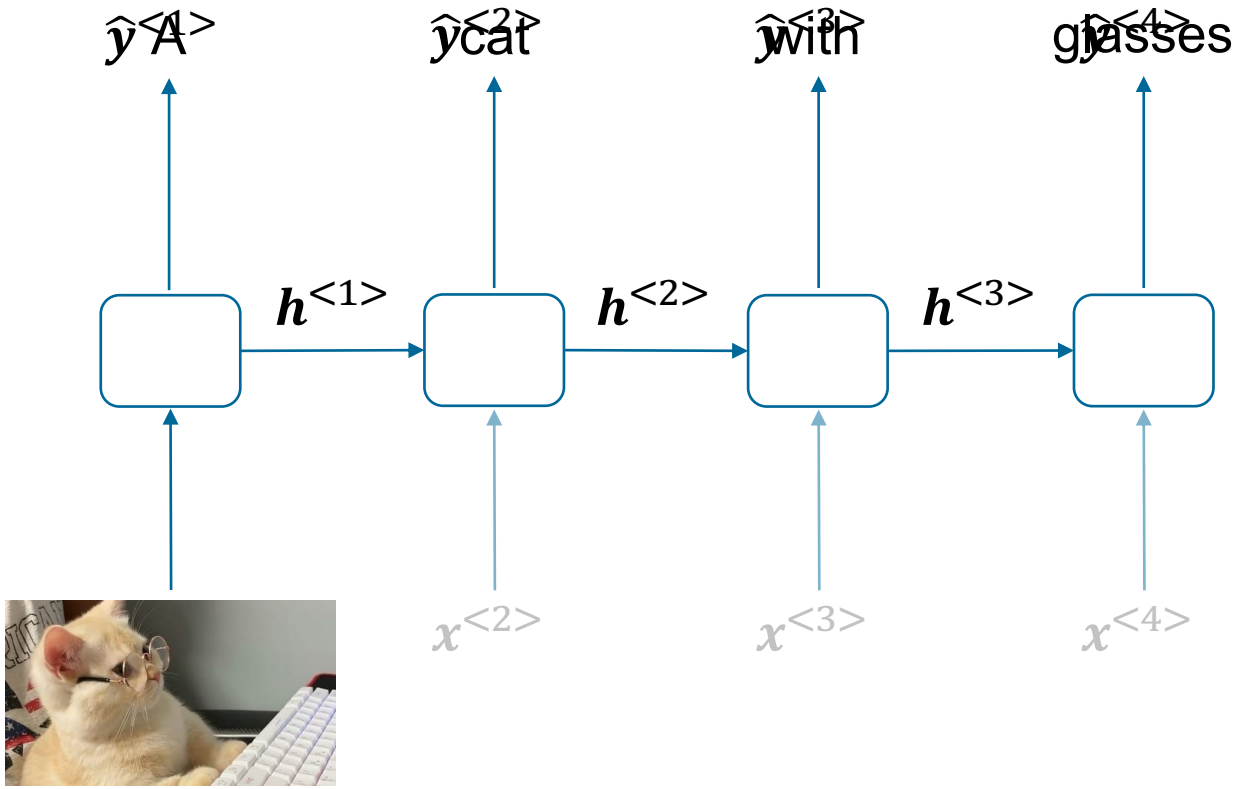
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Let’s use a sequence-to-vector network to forecast multiple time steps ahead

- Go through Part 1 of the Notebook (Colab not needed, but may be useful later)
- Don’t worry about completing the training. Make sure you understand the dataset construction!

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Vector-to-sequence networks

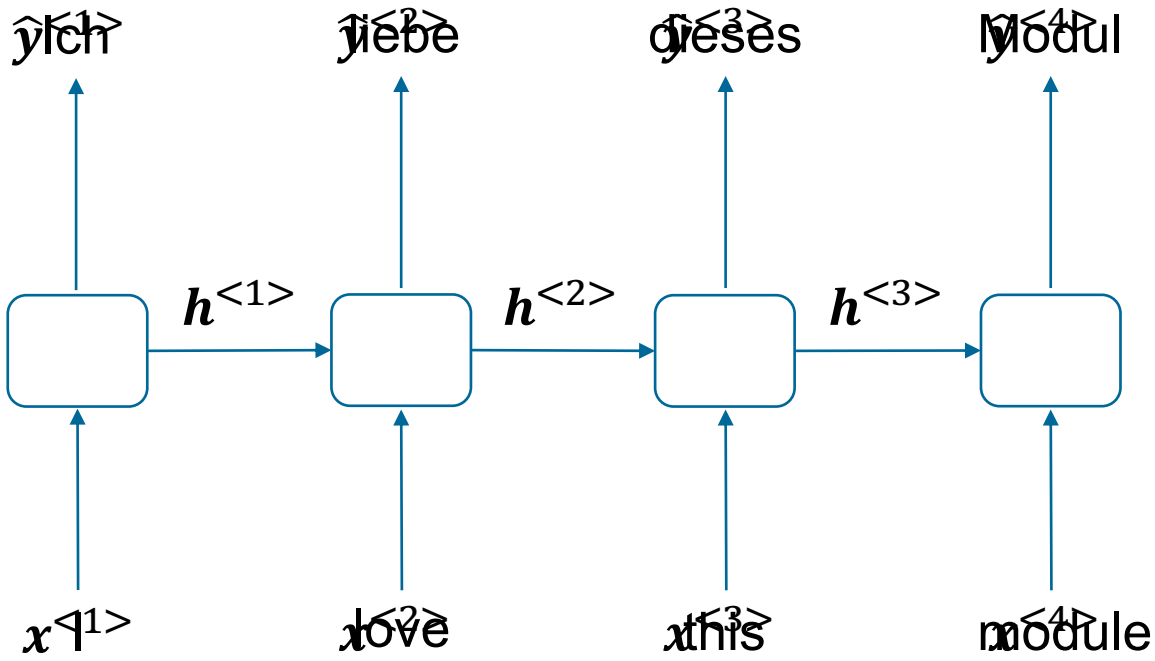


For example:

- Text generation
- Music generation
- Image captions

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Sequence-to-sequence networks

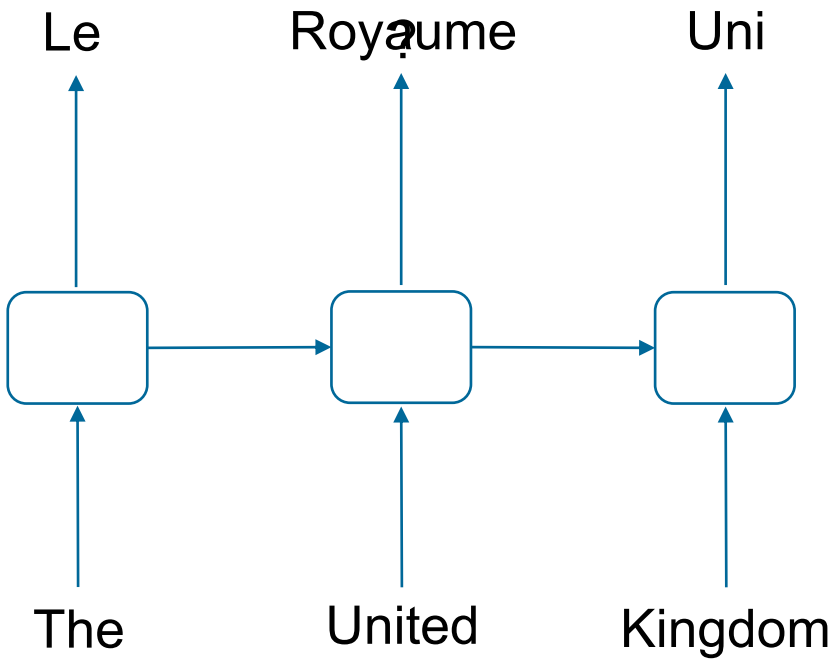


For example:

- Speech recognition
- Translations

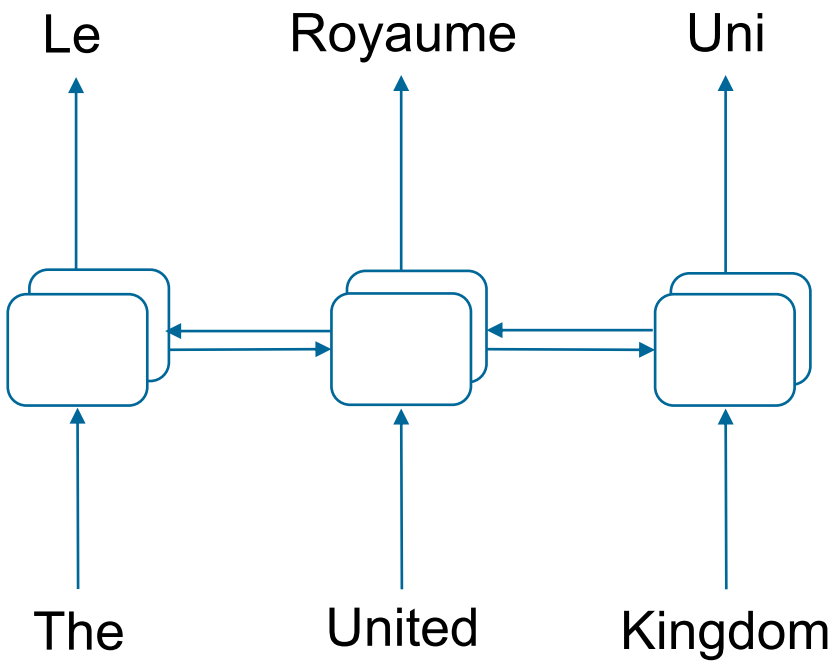
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Bidirectional RNNs – looking into the future



BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

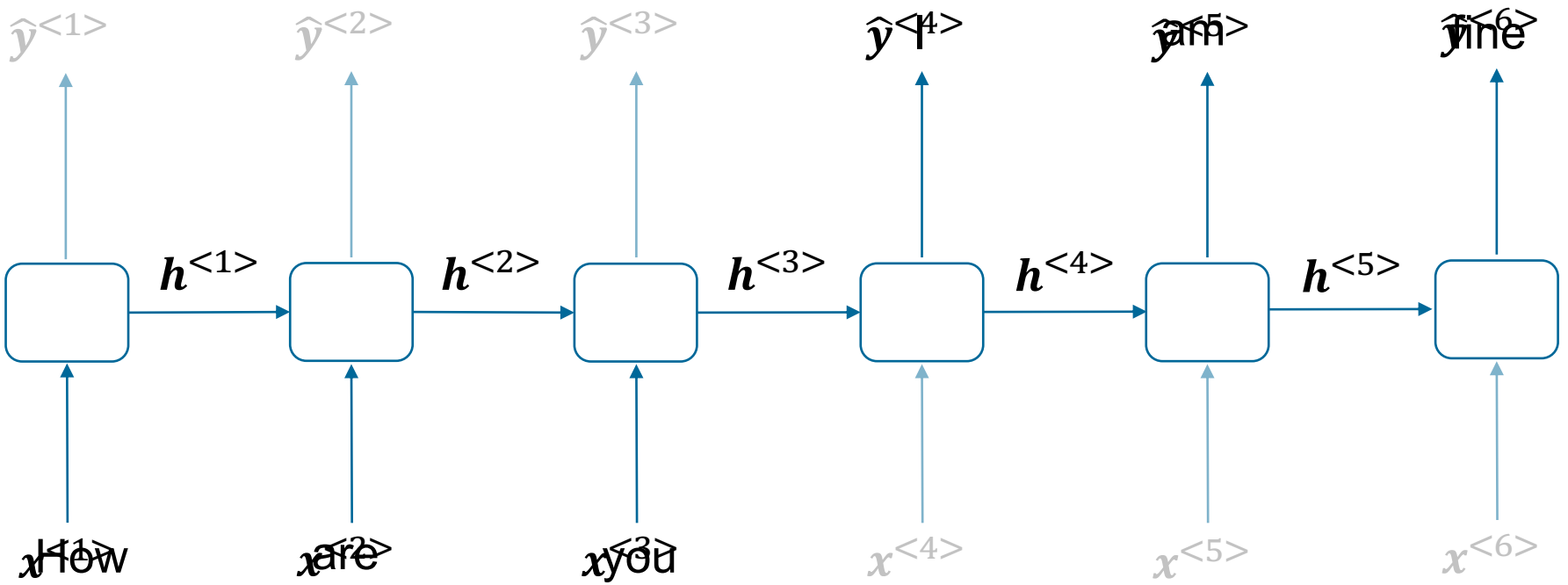
Bidirectional RNNs – looking into the future



For example:

- All sorts of NLP
- Also, in combination with the previous

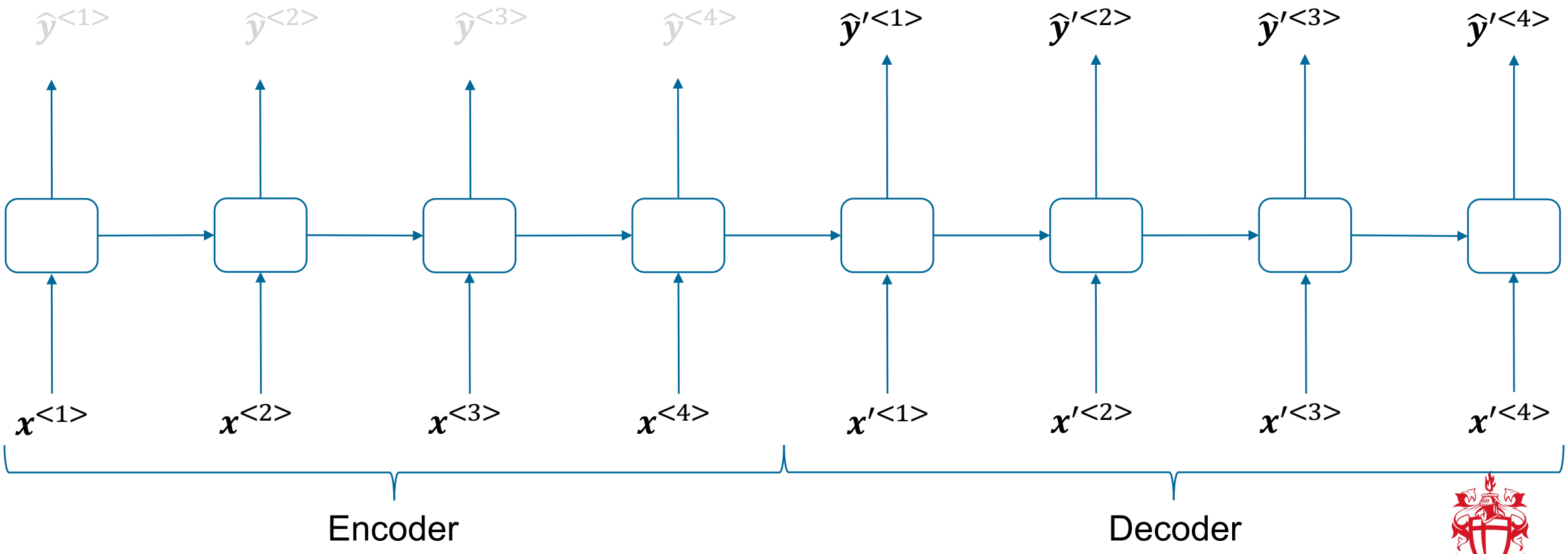
Encoder-decoder networks



For example:

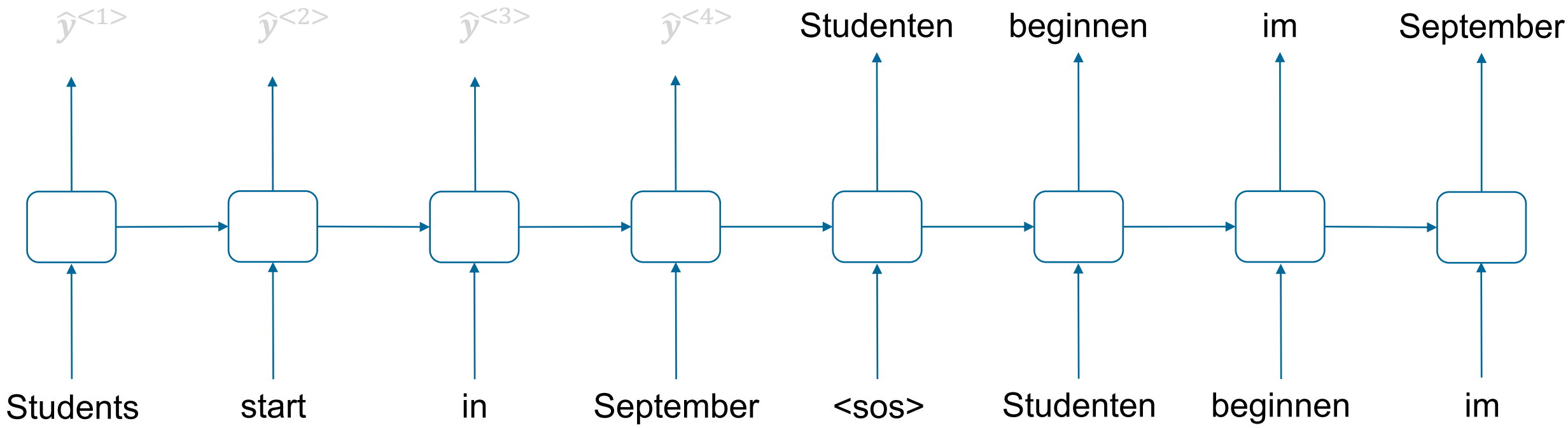
- Translations
- Dialogue

Encoder-decoder networks, for example in translation



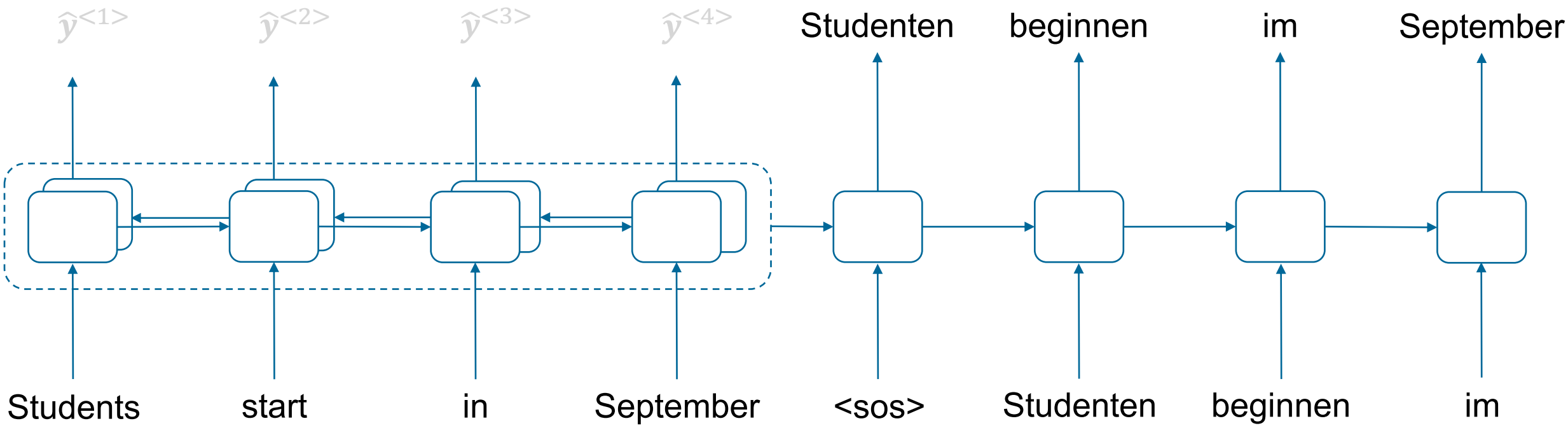
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Encoder-decoder networks, for example in translation



BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Encoder-decoder networks, adding in a bidirectional RNN



BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

A brief intro to Natural Language Processing (NLP)

Typical NLP examples

"To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take Arms against a Sea of troubles,
And by opposing end them: to die, to sleep
No more; and by a sleep, to say we end
The heart-ache, and the thousand natural shocks
That Flesh is heir to? 'Tis a consummation
Devoutly to be wished. To die, to sleep,
To sleep, perchance to Dream; aye, there's the rub,…"

Text classification → Monologue

Content filtering → May contain references to suicide

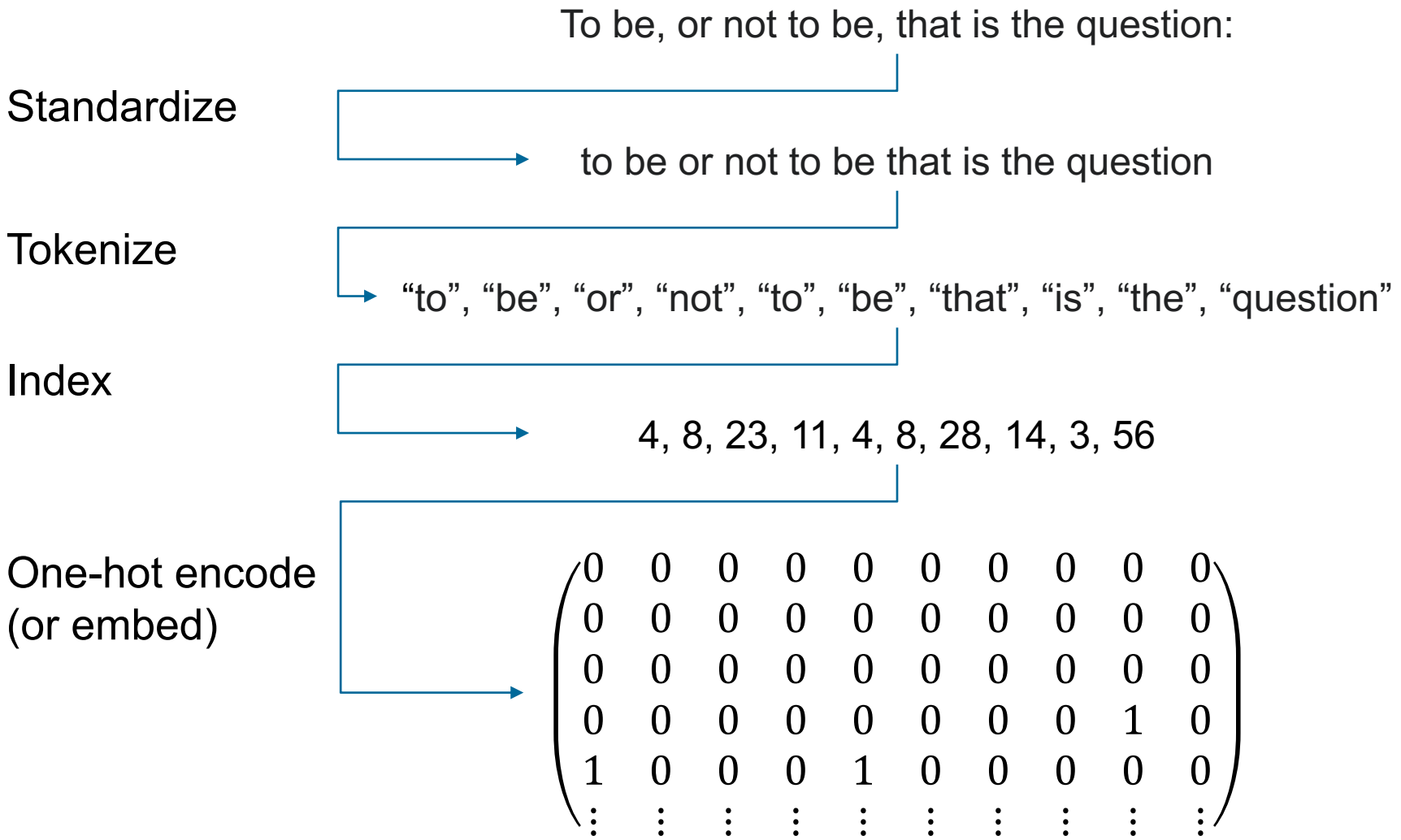
Sentiment analysis → Negative

Language modeling → "For in that sleep of death"

Translation → "Sein oder nicht sein, das ist die Frage…"

Summarization → The speaker contemplates whether it is better to suffer the hardships of life or to end them through death, but the fear of the unknown afterlife causes him to hesitate and remain indecisive.

Typical data preprocessing for NLP

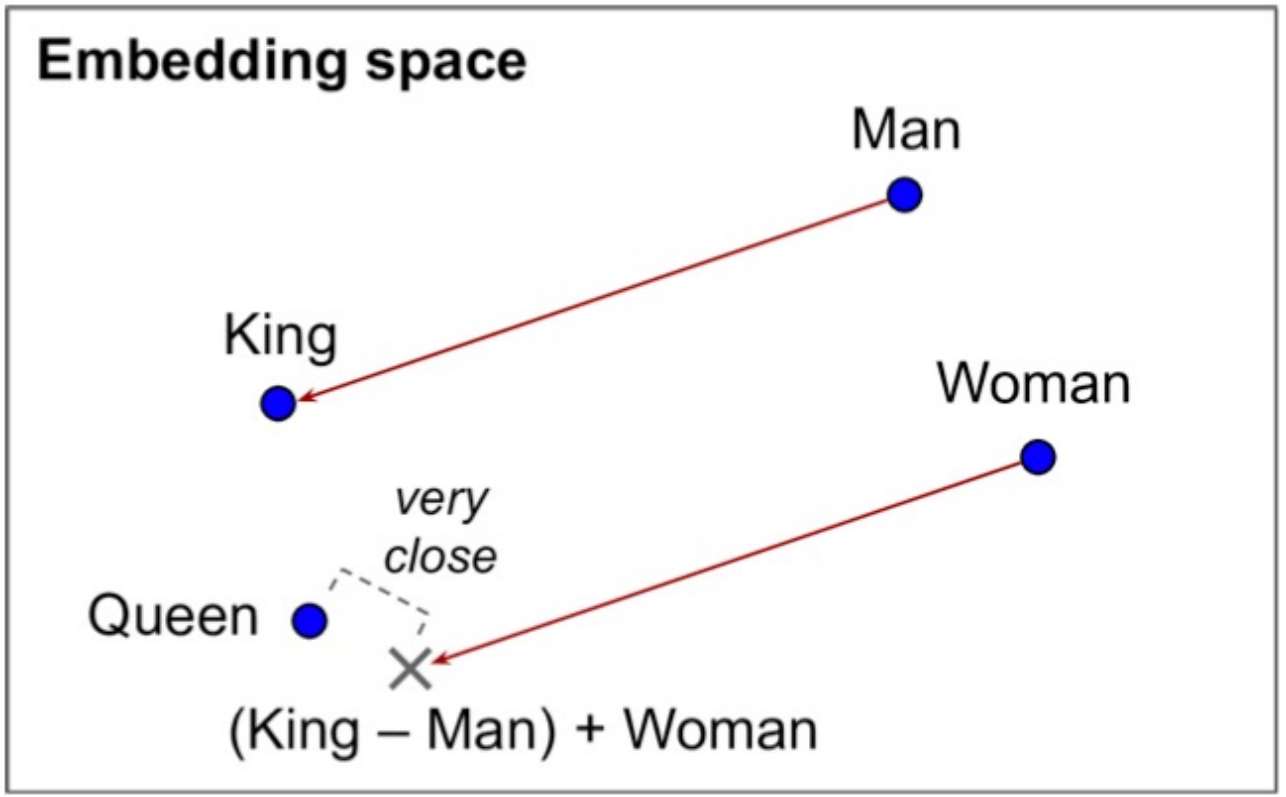


BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The issue with one-hot encoding

- With one-hot encoding, we assume that different tokens are all independent
- However, words like “movie” and “film” are very similar in meaning and mostly interchangeable
  - They’re clearly not independent
- We want to encode words to represent their semantic relationship
  - “Word embeddings”
  - This also allows us to have fewer dimensions

Word embeddings



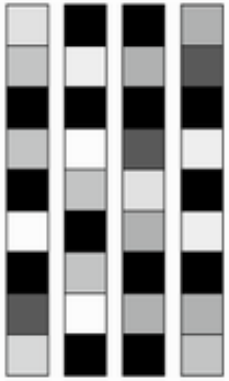
Let's take a look: https://projector.tensorflow.org

Source: Géron

One-hot encoding vs. word embedding

One-hot encoding:

- Sparse
- High-dimensional
- Hardcoded



Word embedding:

- Dense
- Lower-dimensional
- Learned from data

Source: Chollet

Using word embedding

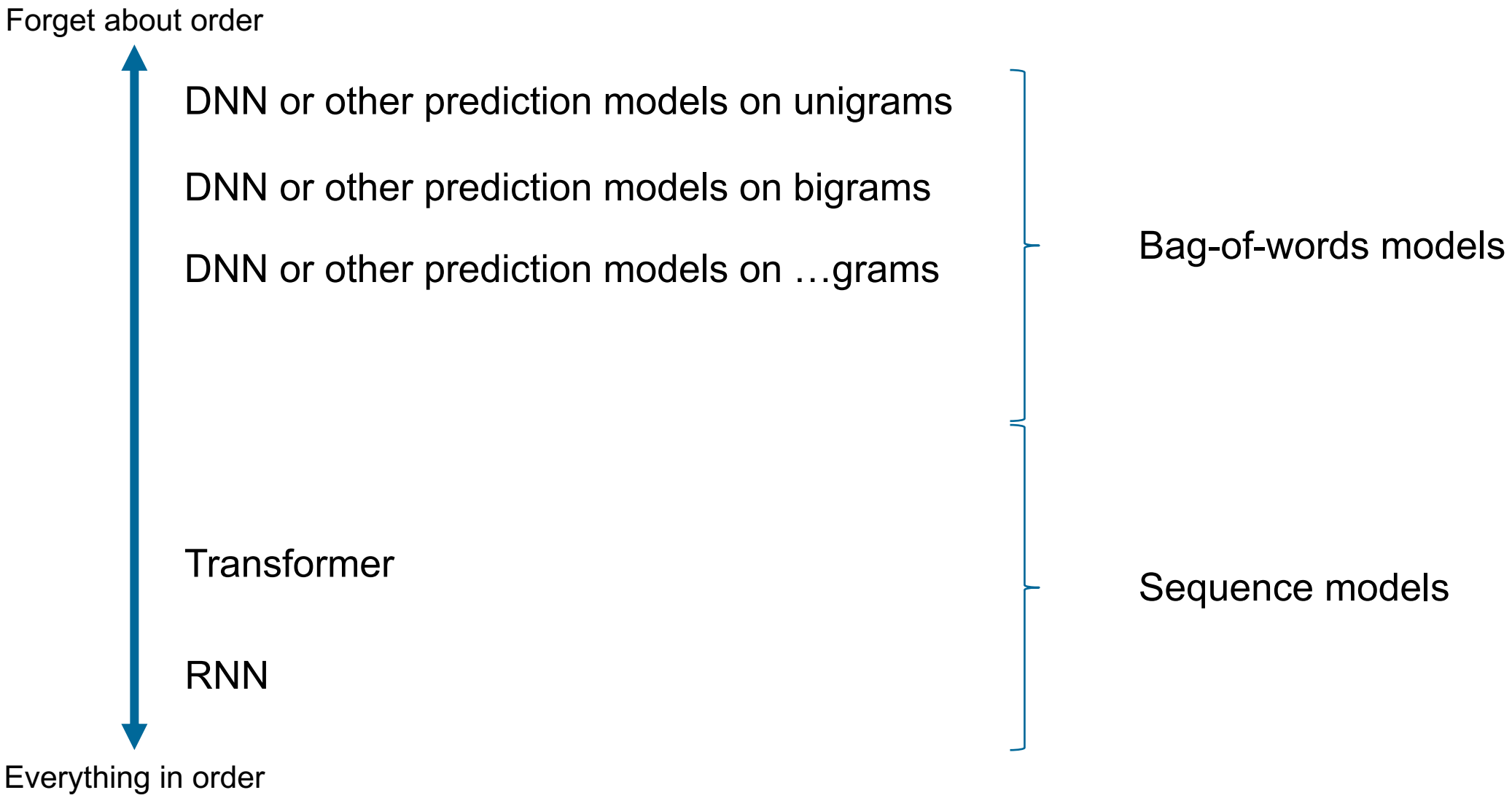
	Learn embeddings (embedding layer)	Use pre-trained embeddings
+	Ideal word embedding task-specific	Usually, quite flexible
−	Adds to training time Requires sufficient data	May not be sufficiently specific

Representing groups of words

- Clearly, order matters in natural language

- There is no canonical order as with time series:
  - “And I go out” ≈ “And out I go”
  - “This order makes sense” ≠ “Sense this order makes”
  - “The United Kingdom” ≠ “Le Royaume Uni”
  - …

Different ways of dealing with order



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

N-grams

Original sentence: to be or not to be

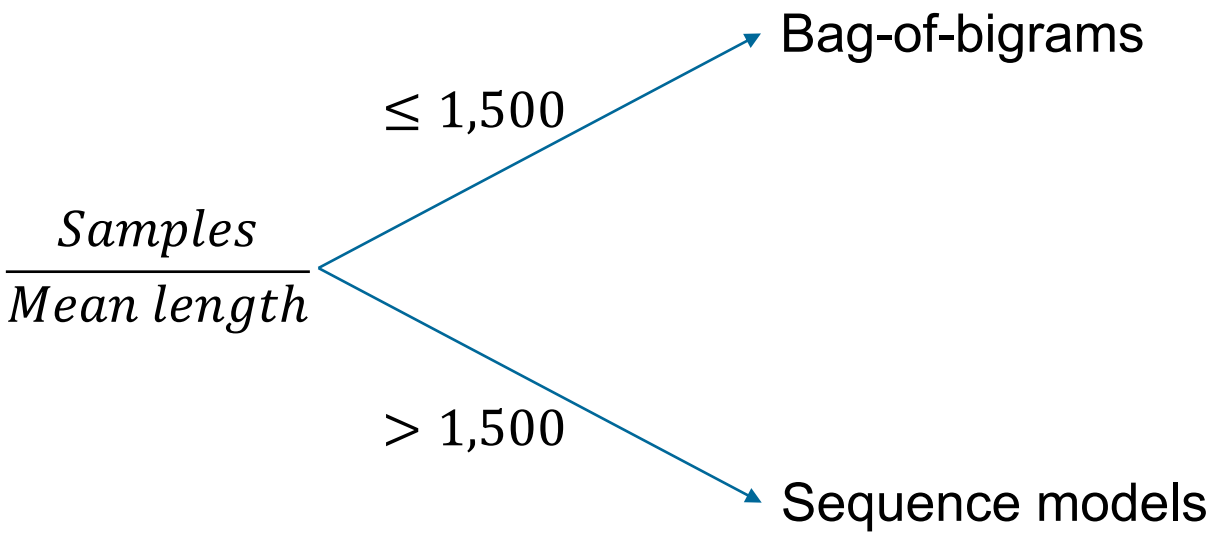
Unigrams: {“to”, “be”, “or”, “not”}

Bigrams: {“to”, “be”, “or”, “not”, “to be”, “be or”, “or not”, “not to”}

Let’s try out both on a simple NLP example

- Go through Parts 2.1-2.4 of the Notebook
- Don’t worry about completing the training – test accuracies are given when process takes long
- Make sure you understand the different input formats and can answer the discussion questions

Bag-of-words or sequence models? A heuristic for text classification

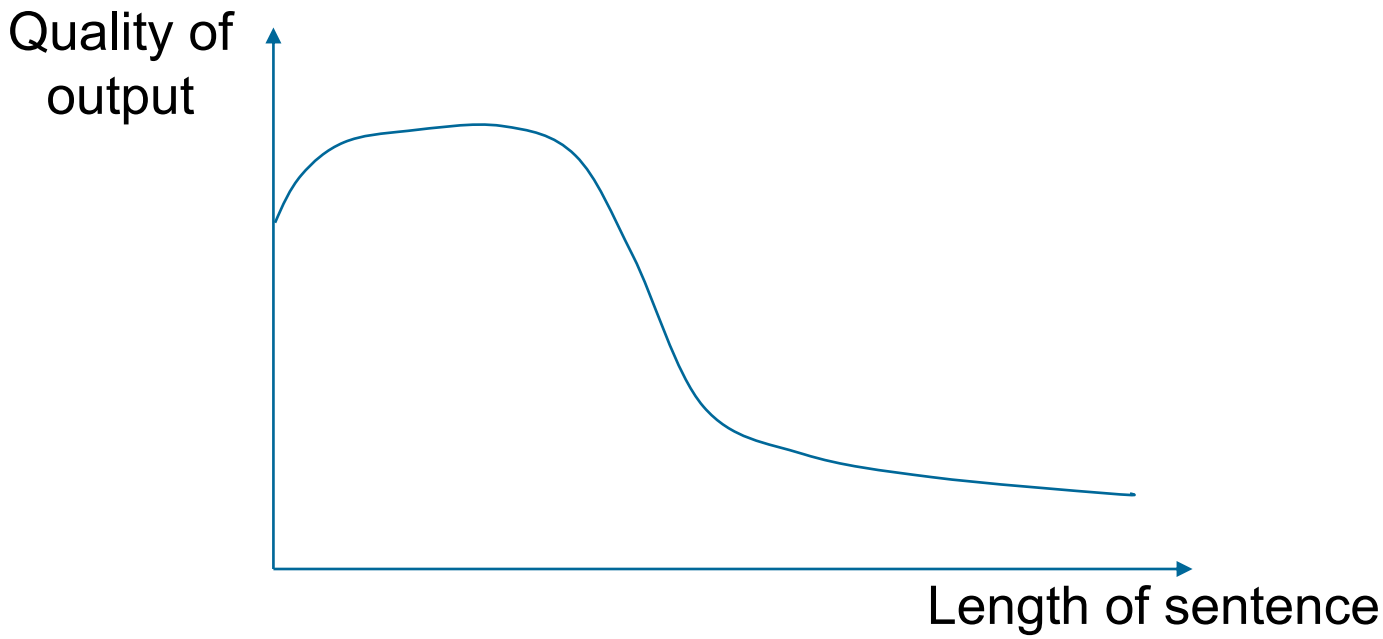


Source: Chollet

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Attention mechanism

Why RNNs are not enough



The basic idea

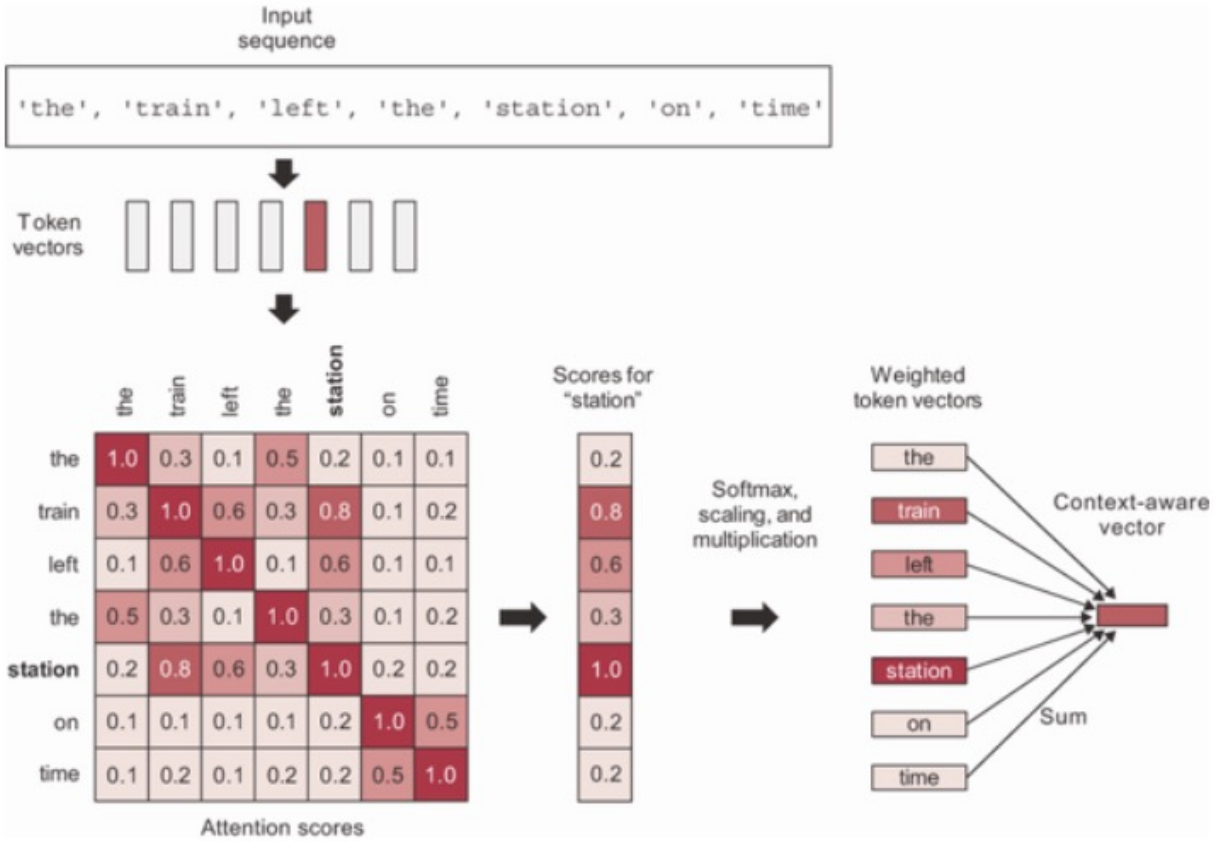
- Not all input information is equally important for your task

- Instead, “pay more attention” to some features, and “pay less attention” to others
  - E.g., “the train left the station on time”

- Like embeddings, attention mechanism captures semantic relationship, however:
  - Word embedding: each word has fixed position (and, thus, fixed relationship to others)
  - Attention: relationship with other words is context-specific

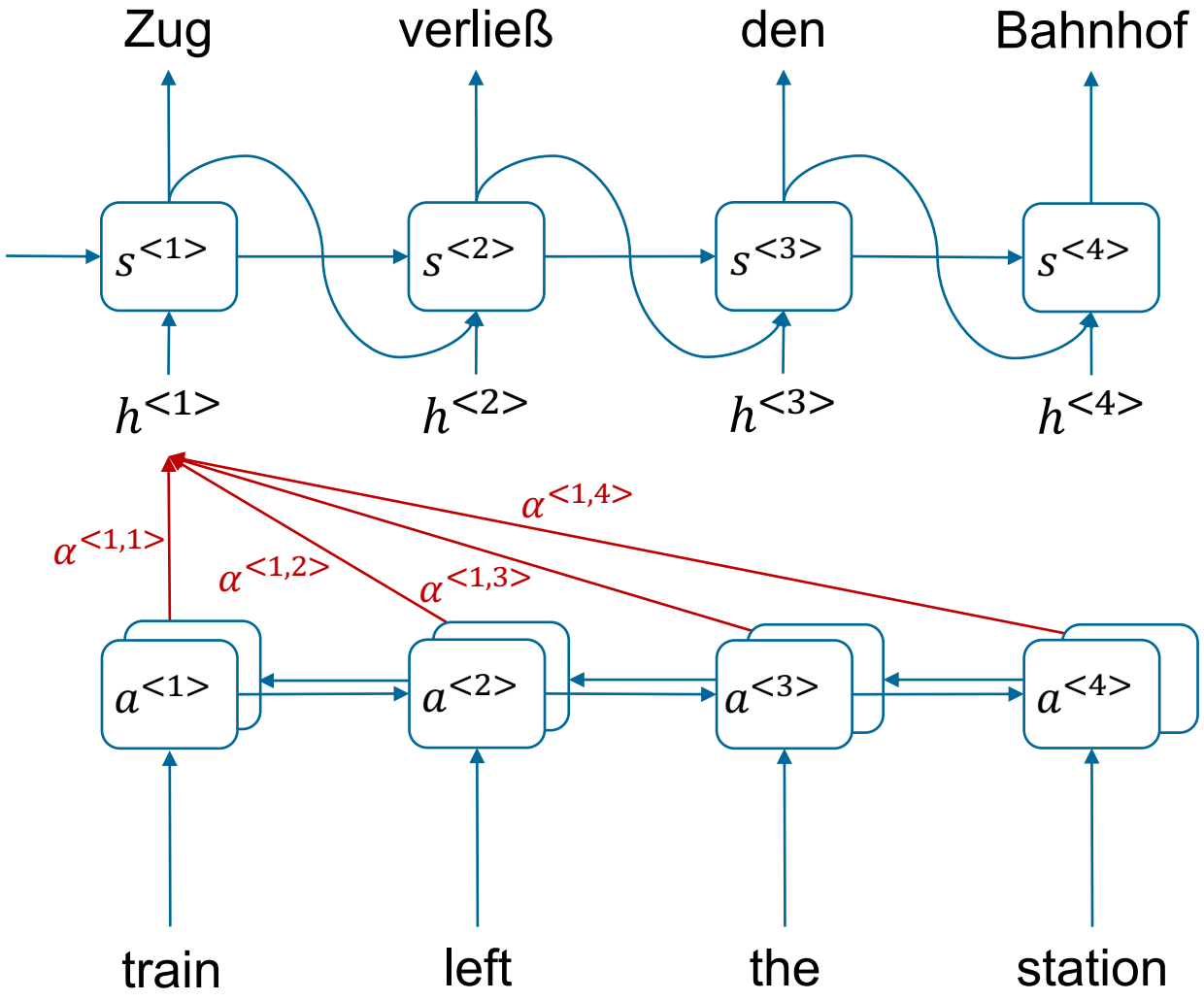
- E.g., think of the word “stand”.
  - Where would you put it relative to “pedestal” in an embedding space?
  - Where would you put it relative to “view” in an embedding space?

Visualizing the attention mechanism



Source: Chollet

Attention in a translation seq-to-seq network

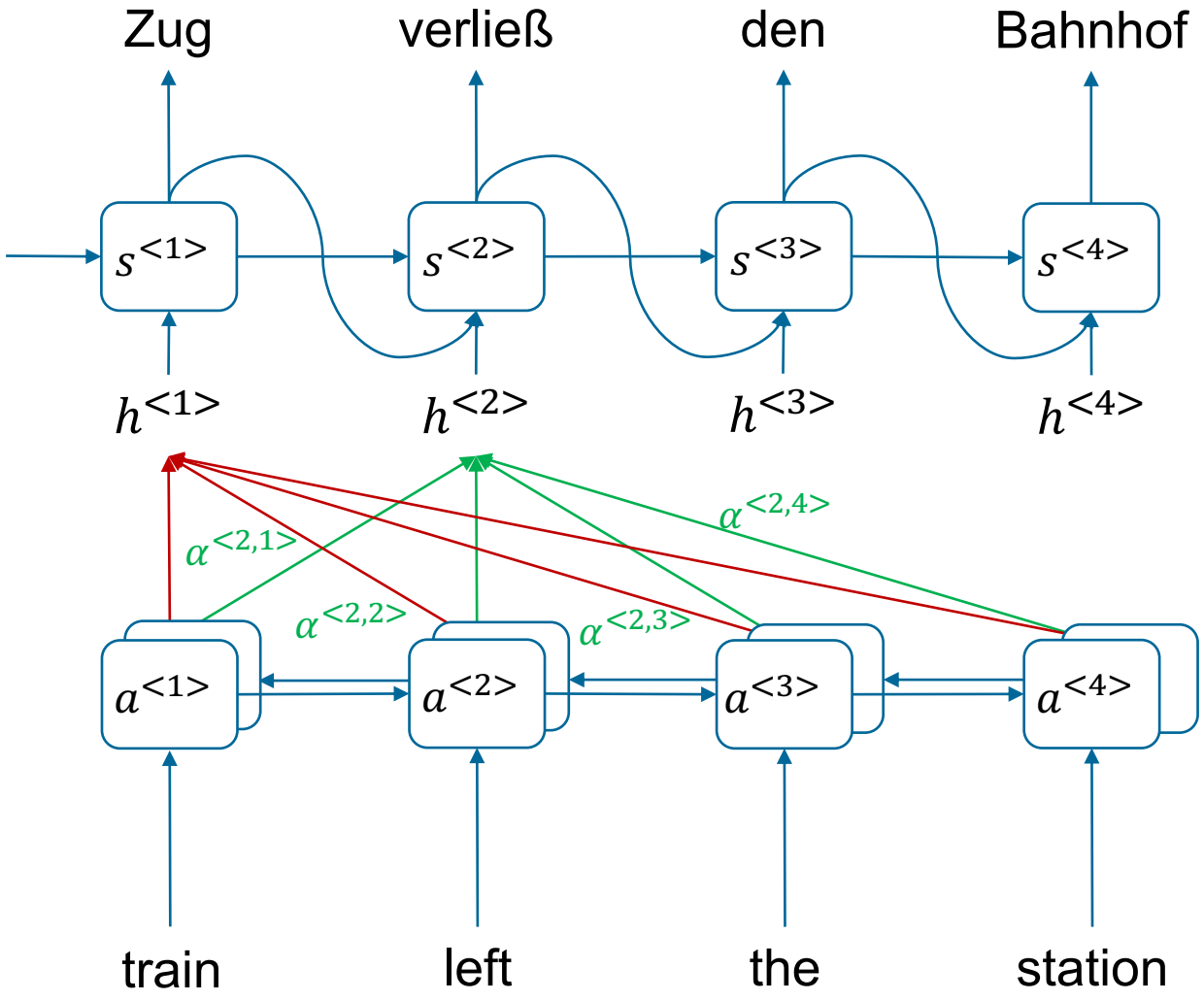


$h^{<1>} = \sum_t \alpha^{<1,t>} a^{<t>} = \alpha^{<1,1>} a^{<1>} + \alpha^{<1,2>} a^{<2>} + \alpha^{<1,3>} a^{<3>} + \alpha^{<1,4>} a^{<4>}$

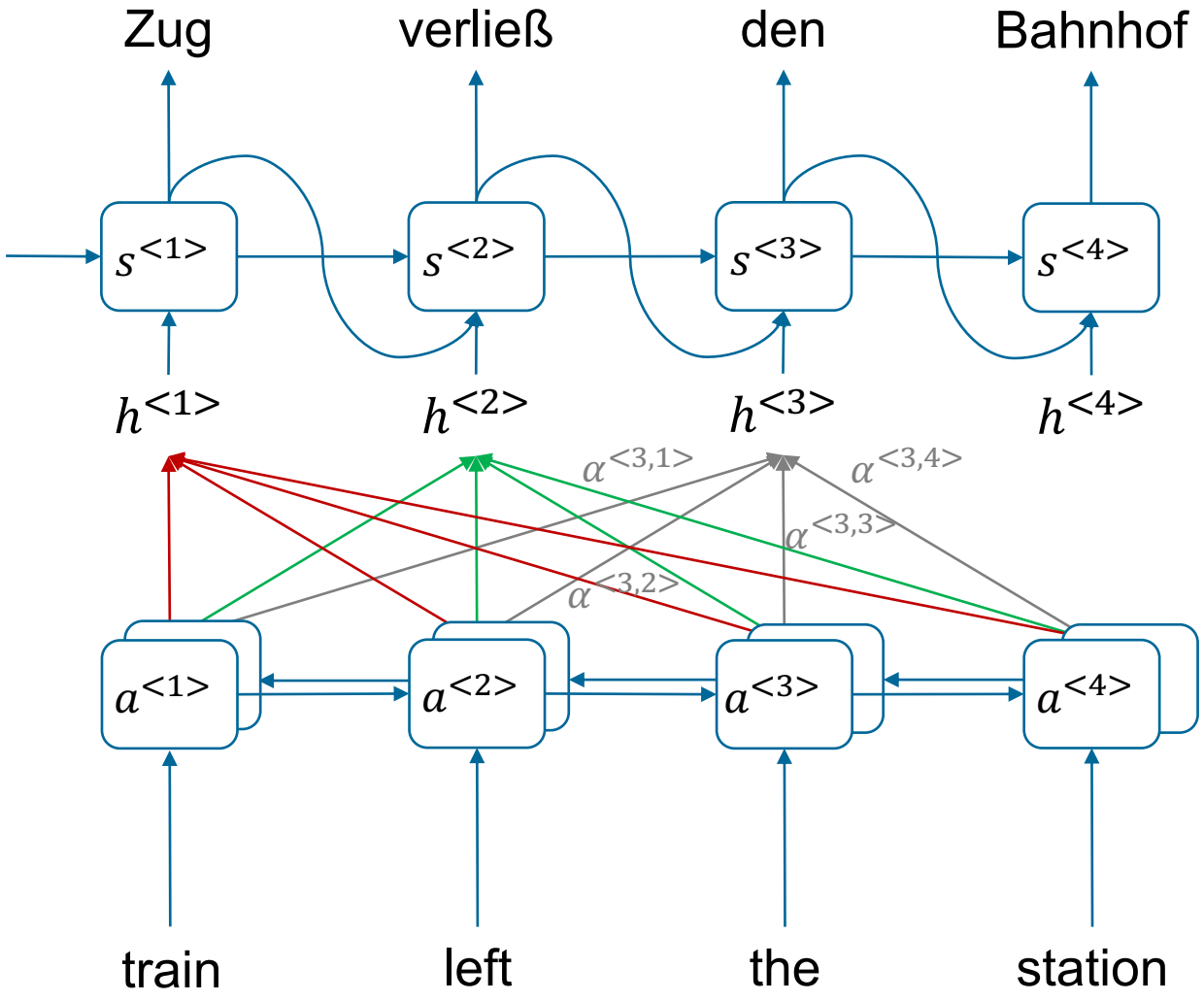
$\sum_t \alpha^{<1,t>} = 1$

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

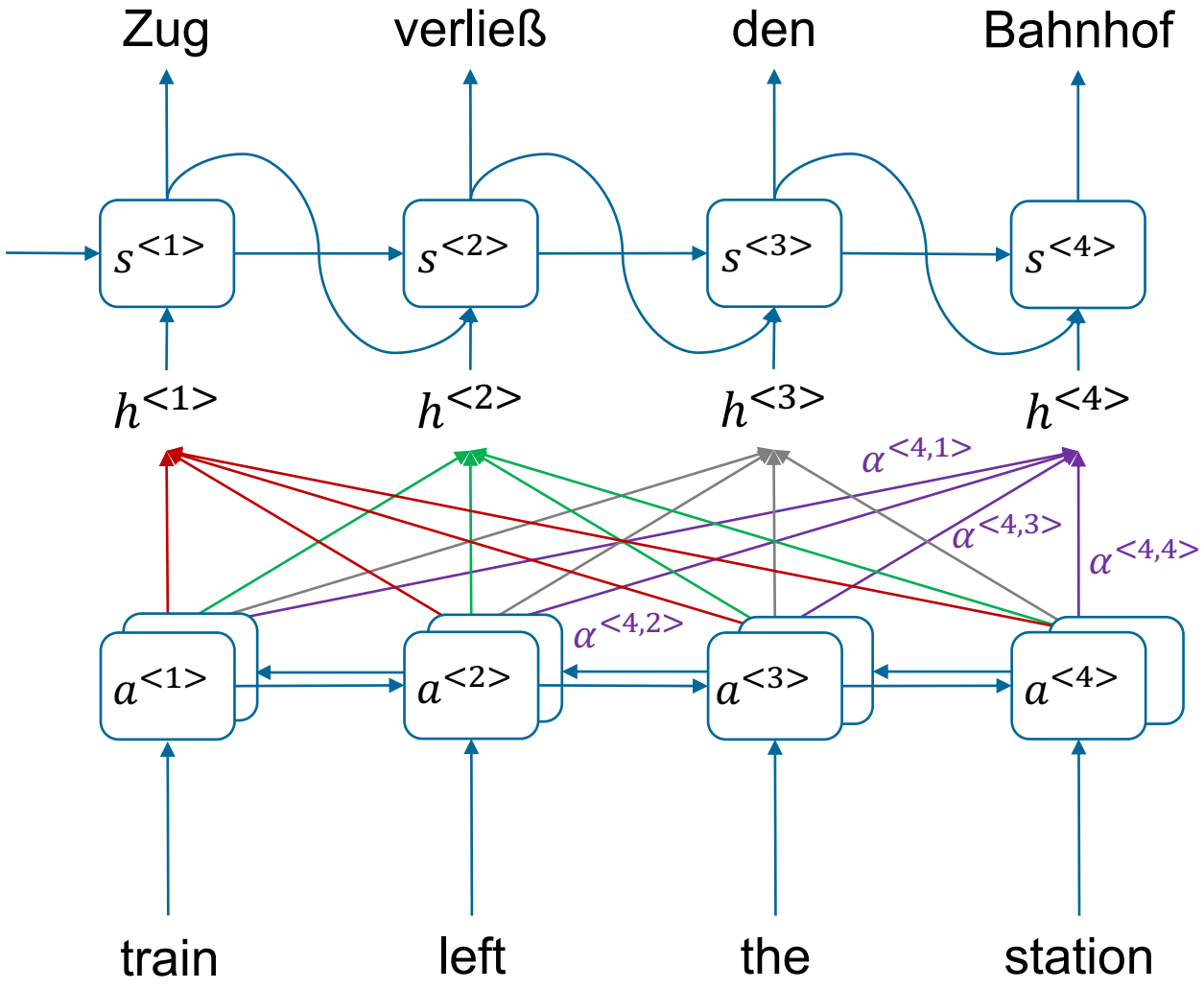
Attention in a translation seq-to-seq network



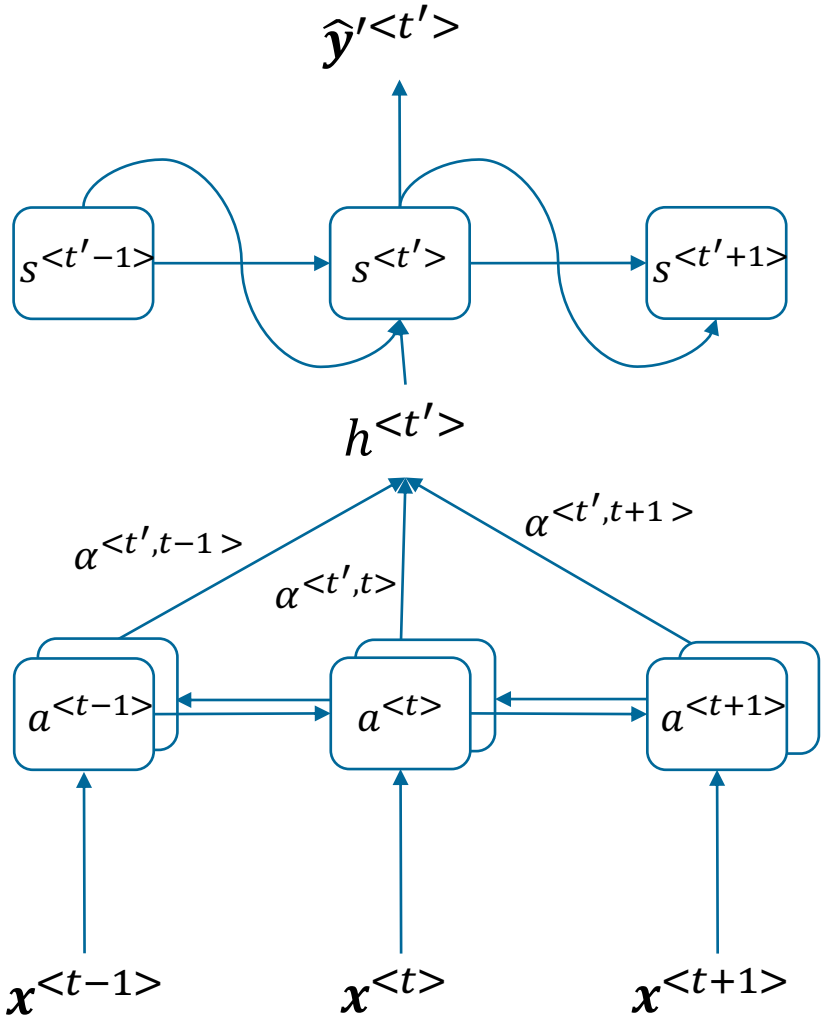
Attention in a translation seq-to-seq network

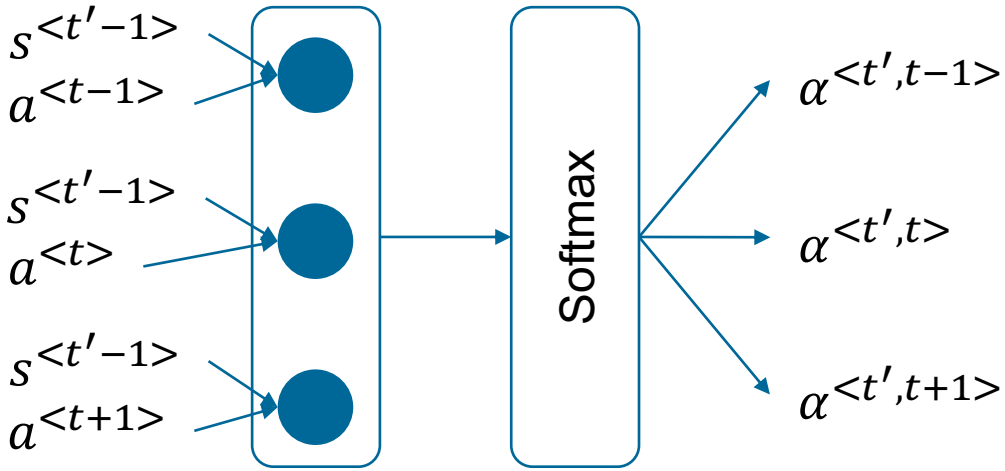


Attention in a translation seq-to-seq network



Computing attention





BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Generalized self-attention

```
outputs = sum(inputs * pairwise_scores(inputs, inputs))
```

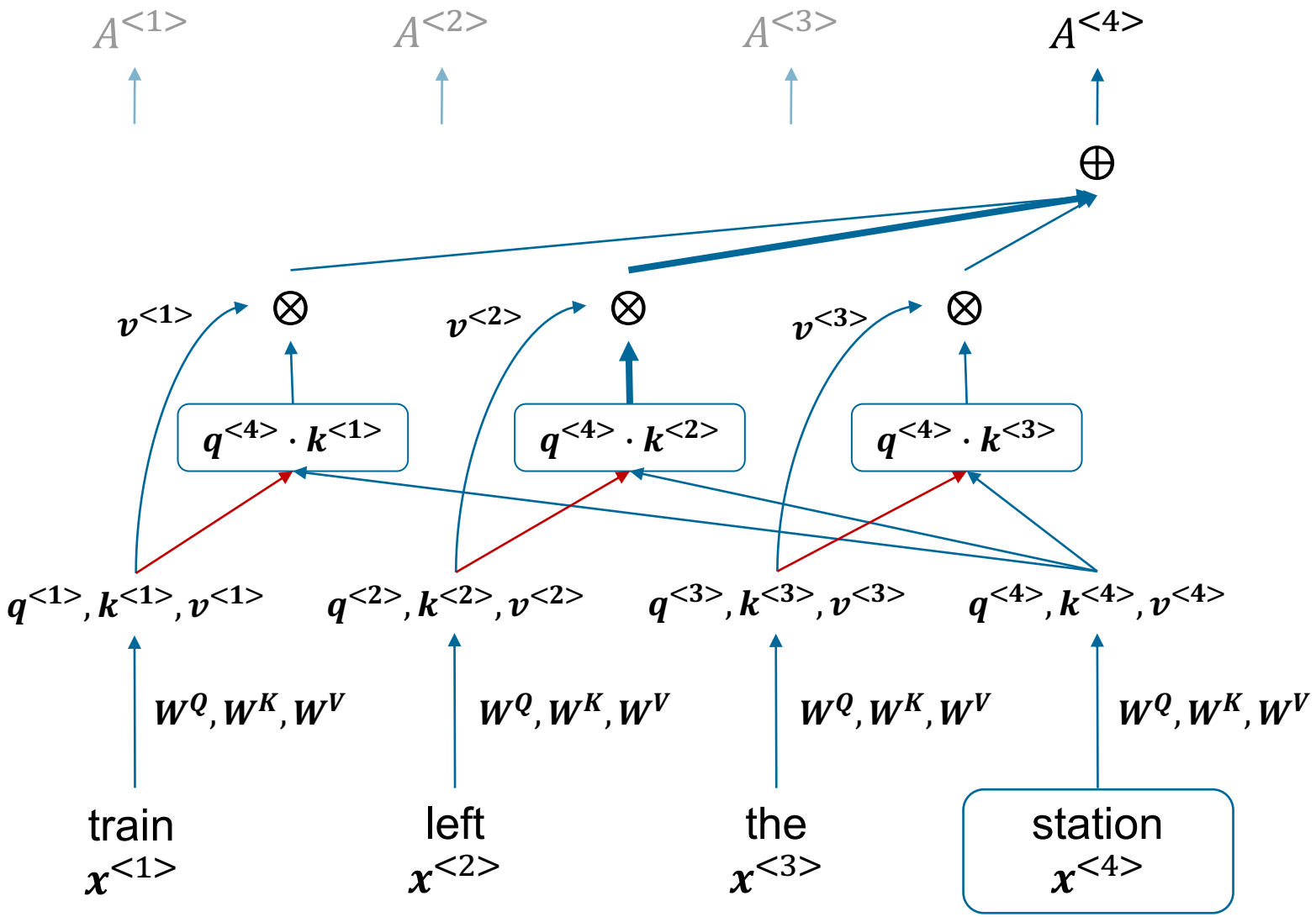
```
outputs = sum(values * pairwise_scores(query, key))
```

Generalized self-attention

We say $A(Q, K, V)$ is the attention-based representation of a word

Query: What happened here?

$\boldsymbol{k}^{<1>}$: Object
$\boldsymbol{k}^{<2>}$: Action
$\boldsymbol{k}^{<3>}$: Article

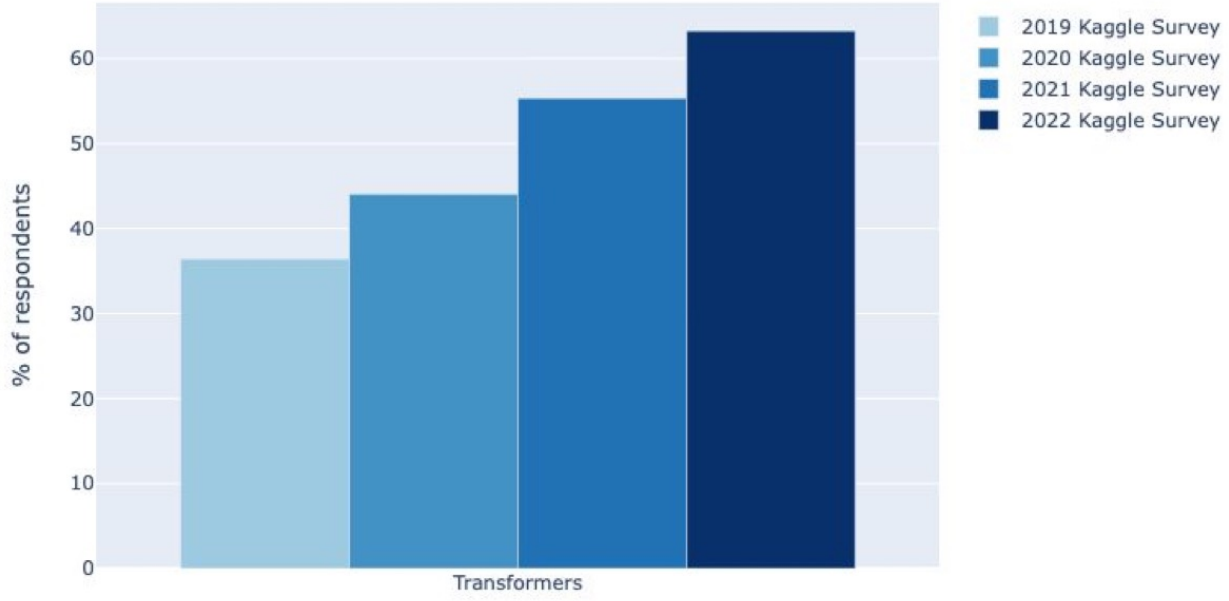


BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Attention is all you need

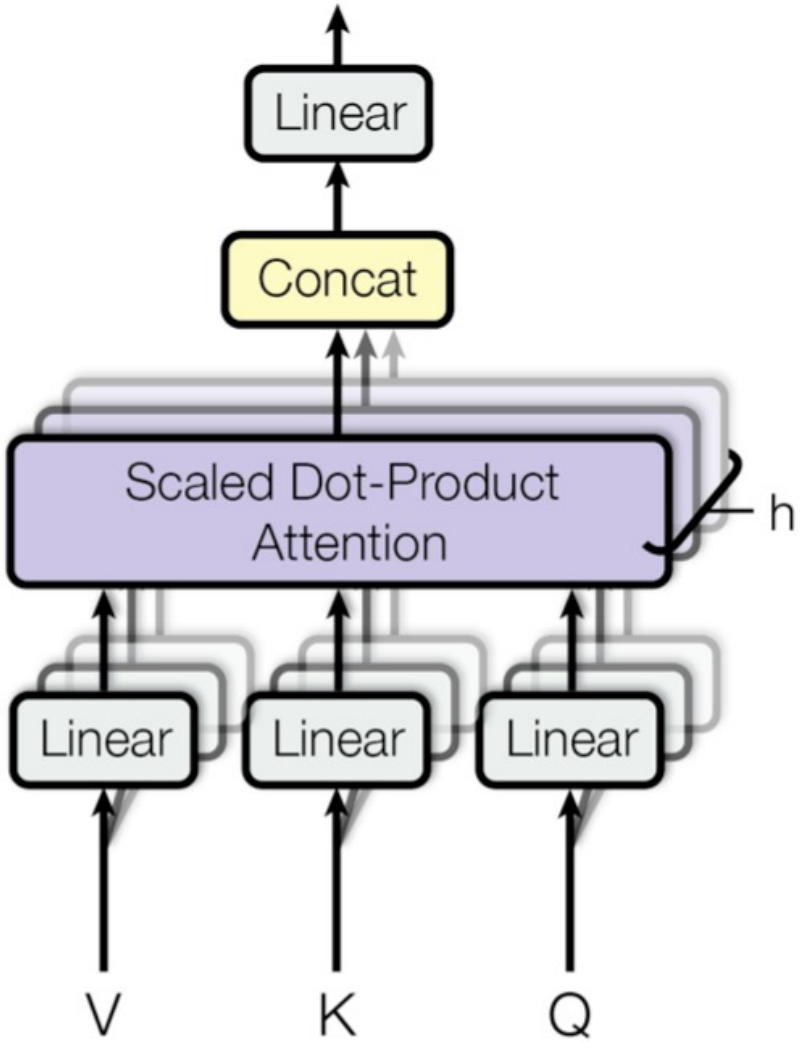
An overview

- 2017 paper by researchers at Google – probably most groundbreaking paper in last decade of machine learning
- Use attention exclusively (and add in some sequence information in another way)
- Motivation: recurrence cannot be parallelized by its very nature, drastically limiting NLP models



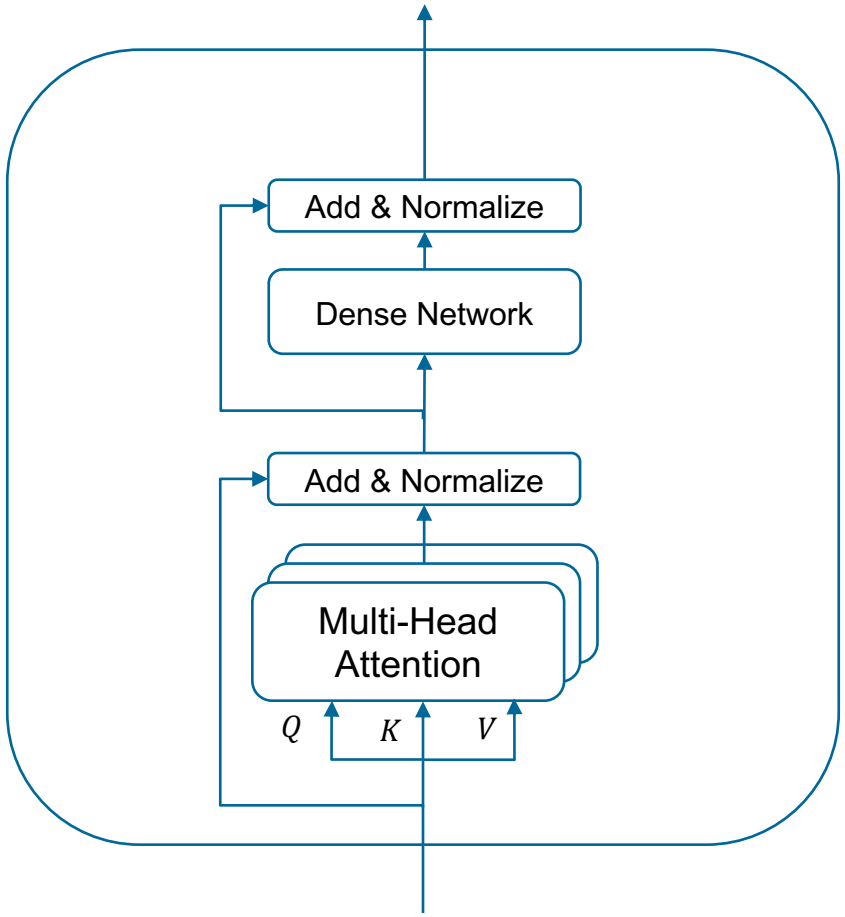
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Multi-head attention layer



Source: Vaswani

The transformer encoder



train left the station

Positional encoding added to the word embedding – a first attempt

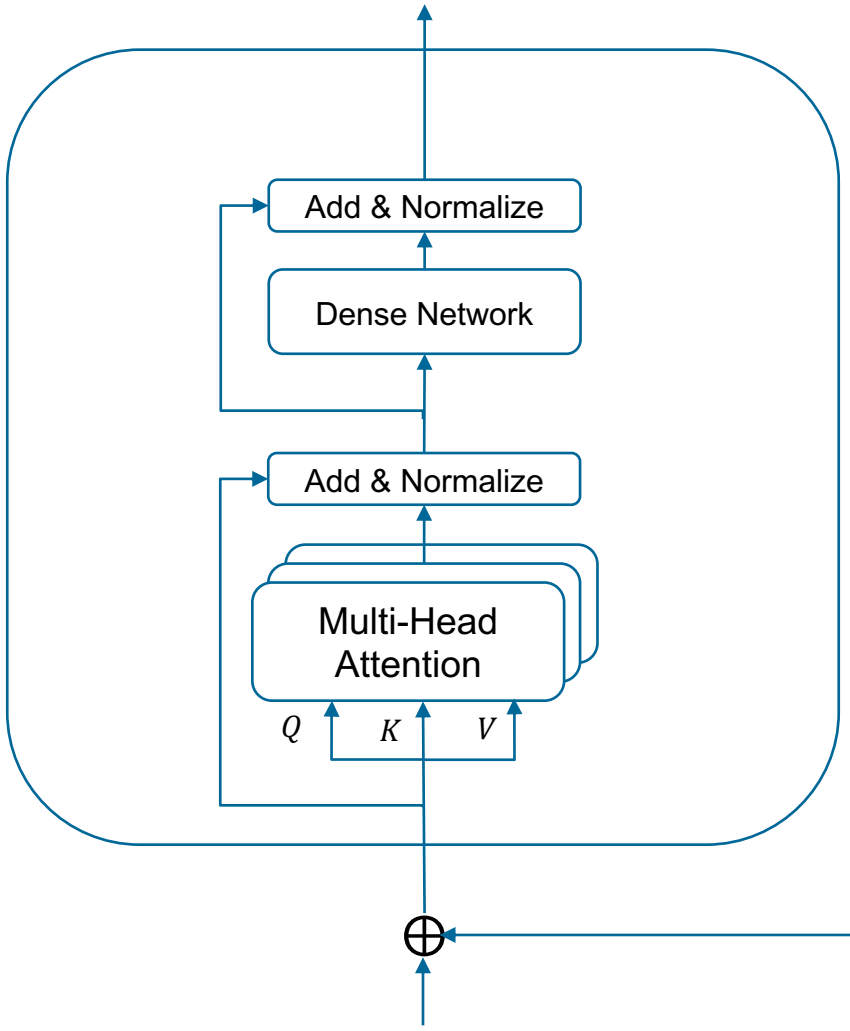
train	+	Position encoding	=	New vector
$\begin{pmatrix} -0.21 \\ 0.45 \\ -0.84 \\ 0.13 \end{pmatrix}$		$\begin{pmatrix} 4 \\ 4 \\ 4 \\ 4 \end{pmatrix}$		$\begin{pmatrix} 3.79 \\ 4.45 \\ 3.16 \\ 4.13 \end{pmatrix}$

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Positional encoding added to the word embedding – a second attempt

- Learn an embedding of the position
- I.e., create a layer that maps the position space [0,…, sequence length] to a dense space and train it

The transformer encoder with positional encoding



Let’s repeat the previous example, using a Transformer encoder

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Positional encoding added to the word embedding – using sinusoids

$$PE_{pos,i} = \begin{cases} \sin\left(\frac{pos}{10000^{i/d}}\right), if\ i\ is\ even \\ \cos\left(\frac{pos}{10000^{(i-1)/d}}\right), if\ i\ is\ odd \end{cases}$$

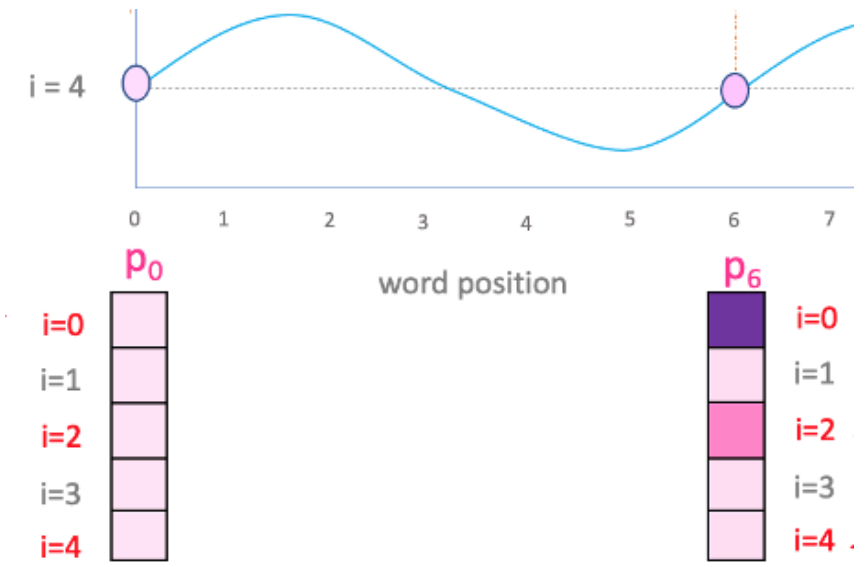
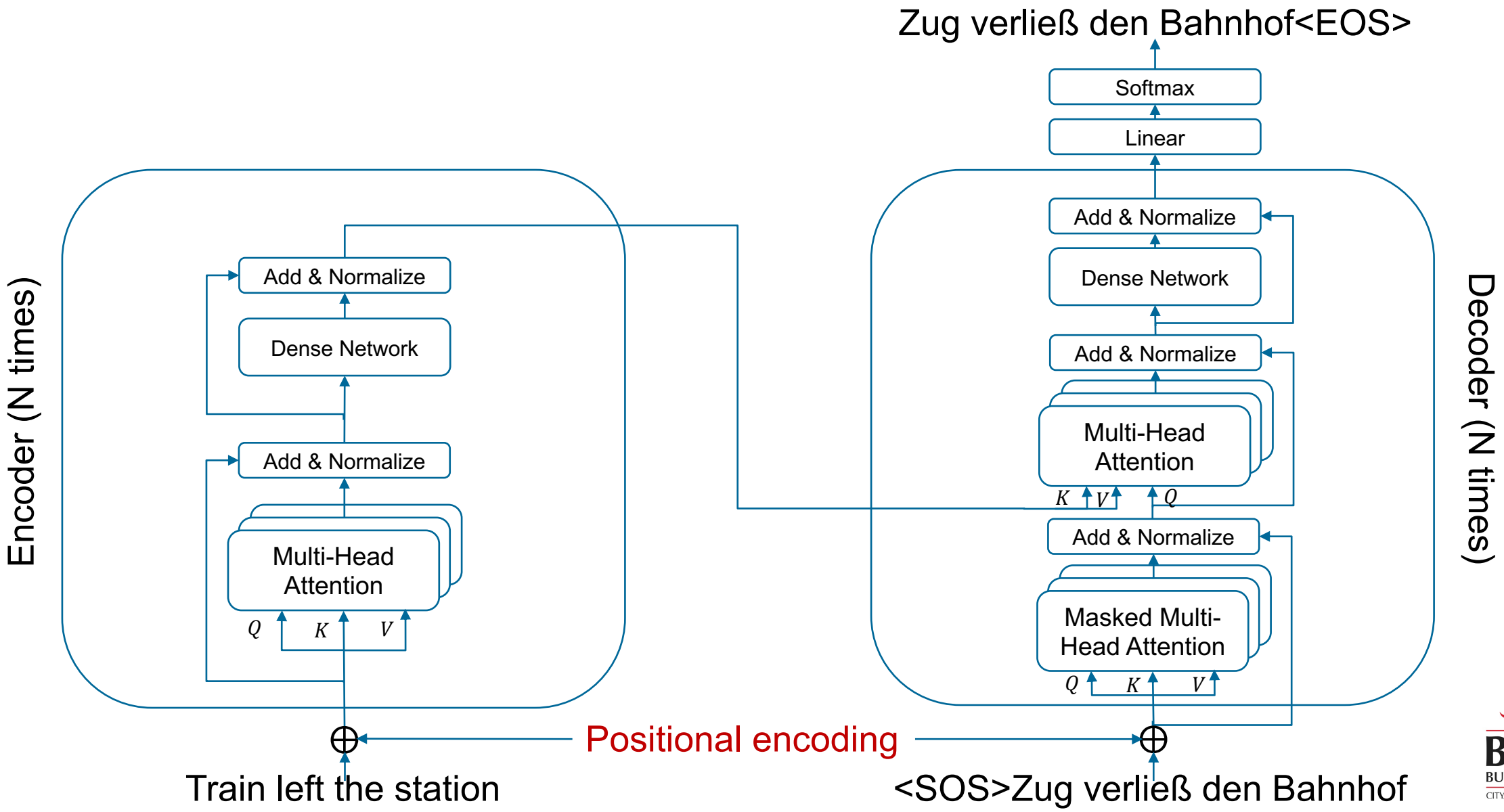


Image source: https://www.youtube.com/watch?v=dichIcUZfOw

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Transformer-based seq-to-seq

Adding a decoder



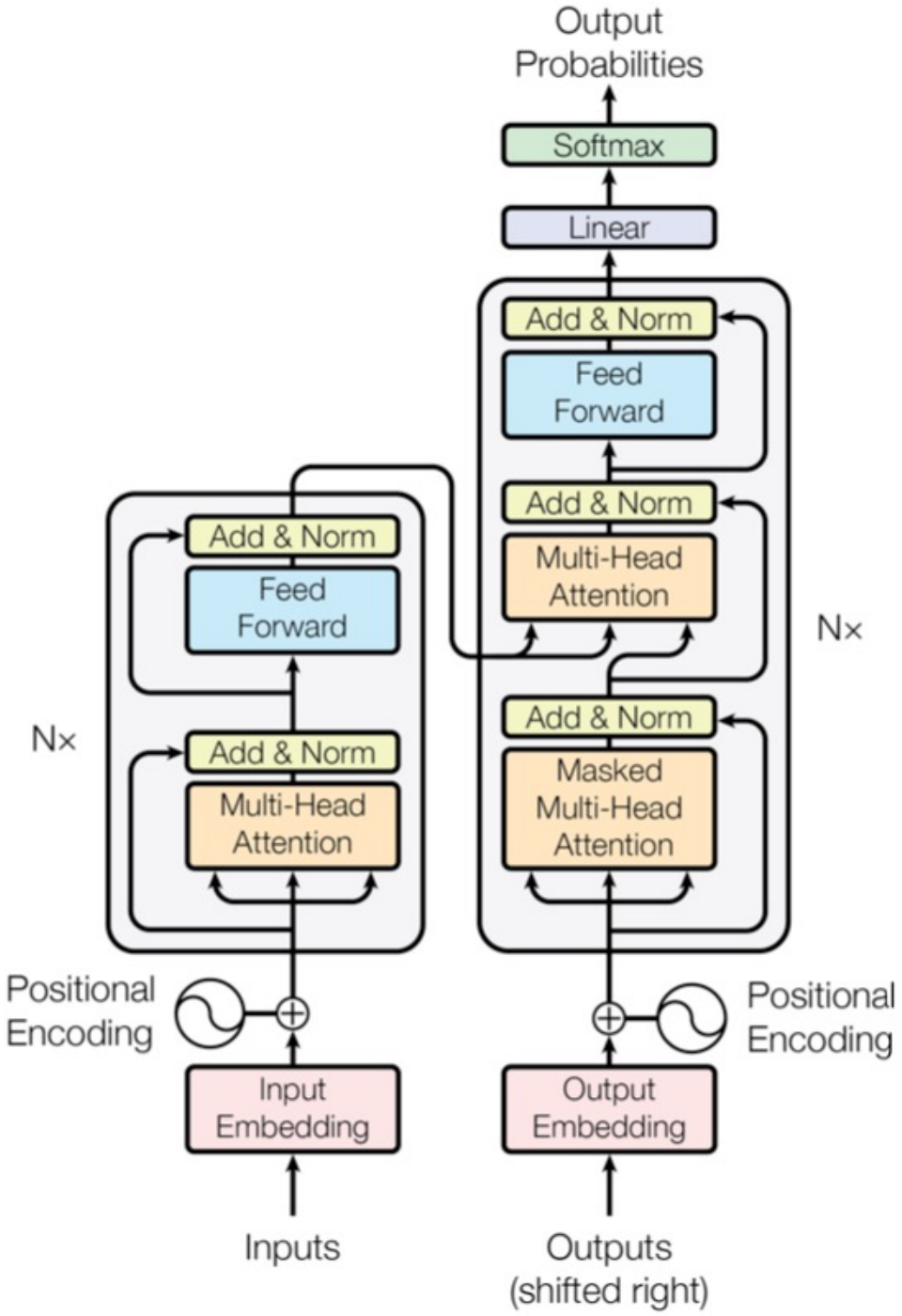
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

An example why position matters in a translation task

Students start in September, have three terms with classes, finish a project, and successfully complete their degree the following summer.

Students start **the following summer**, have three terms with classes, finish a project, and successfully complete their degree **in September**.

Putting it together



Source: Vaswani

The 🤗 Hugging Face library

- AI platform and open-source community designed to share models, model weights, datasets, etc.
- Most accessible library of pre-trained transformer models
  - Generally, with model-specific pre-processing functionality
- Originally built for PyTorch, but works increasingly well with TensorFlow

Using existing transformer models

- Go through Part 3 of the Notebook
- Don’t worry about executing Part 3.2, as this will take quite some time. Make sure you understand what is happening here
- Play around with the examples in 3.3 and try answering the coding question there.

Generative models

Generative models

→ Learn an implicit distribution from data

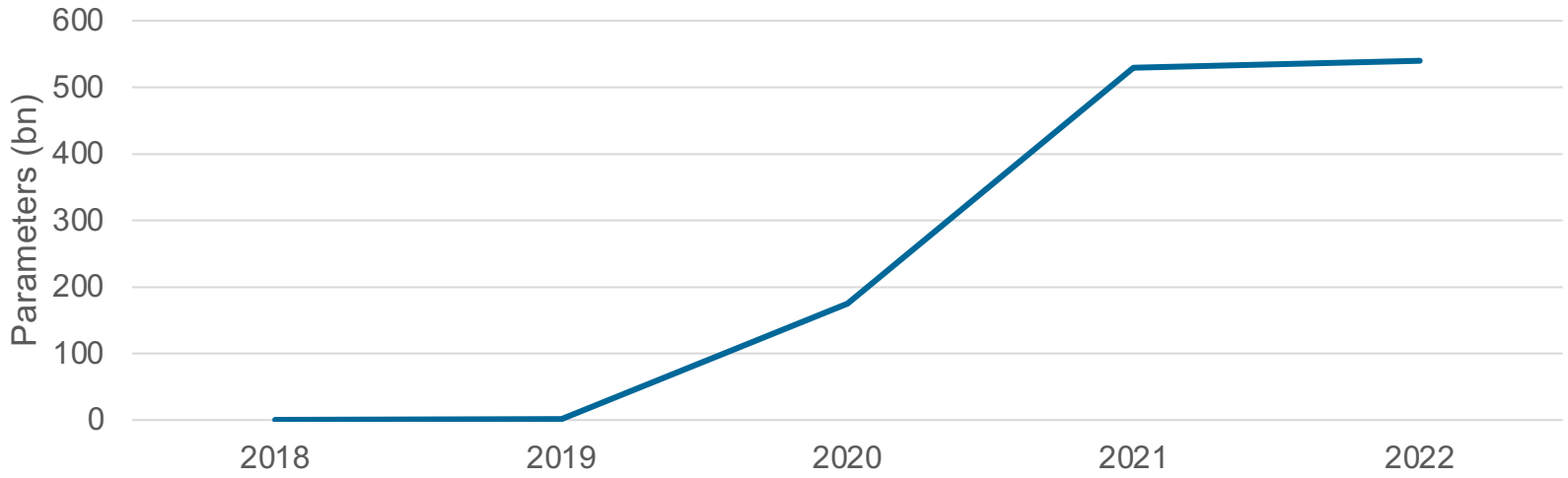
Source: Rosca & Donahue

Creating content with deep learning

- Models can learn the statistical structure of perception, language, art, …
- We can then sample from the latent space learned by the models to create new language output, new images, new pieces of art,…
- Keep in mind: there is no meaning associated with what our models do, it's only mathematical operations

Large language models

- Language model: able to generate text word-for-word
  - More precisely: any neural network that can model the probability of the next token, given all the previous ones
  - Needs to learn latent space of language

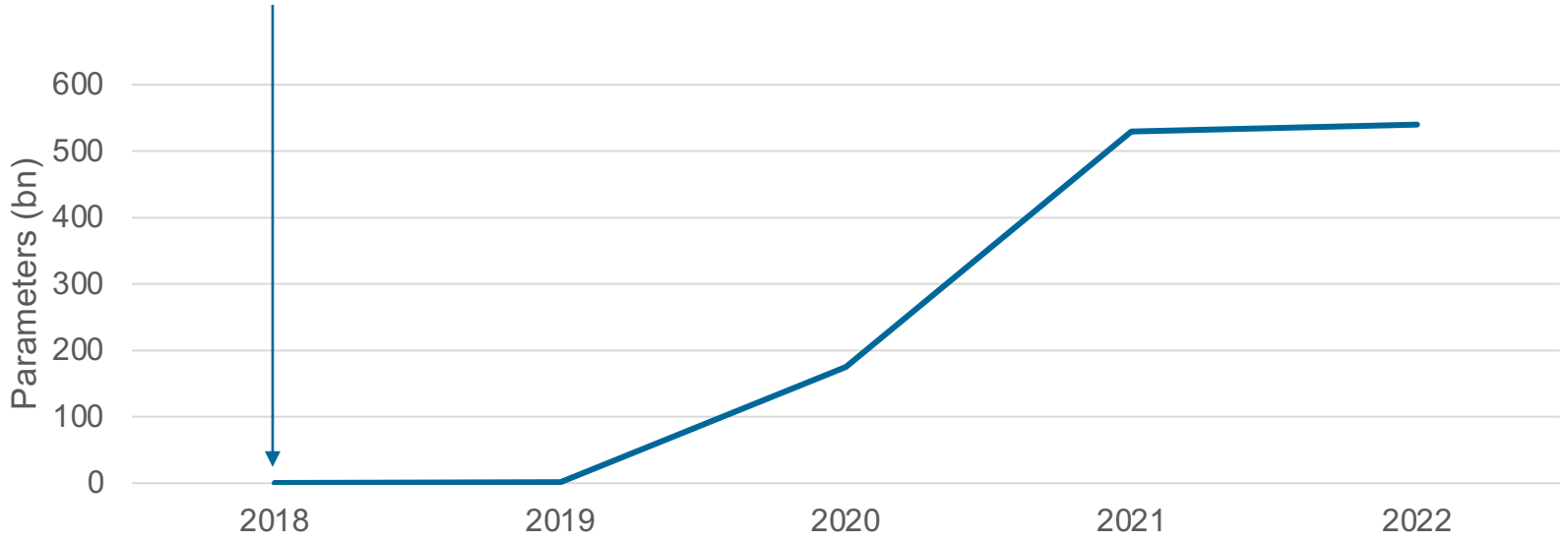


BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Large language models

BERT:

- Google
- 340 mio parameters
- 3.3 bn words
- Open

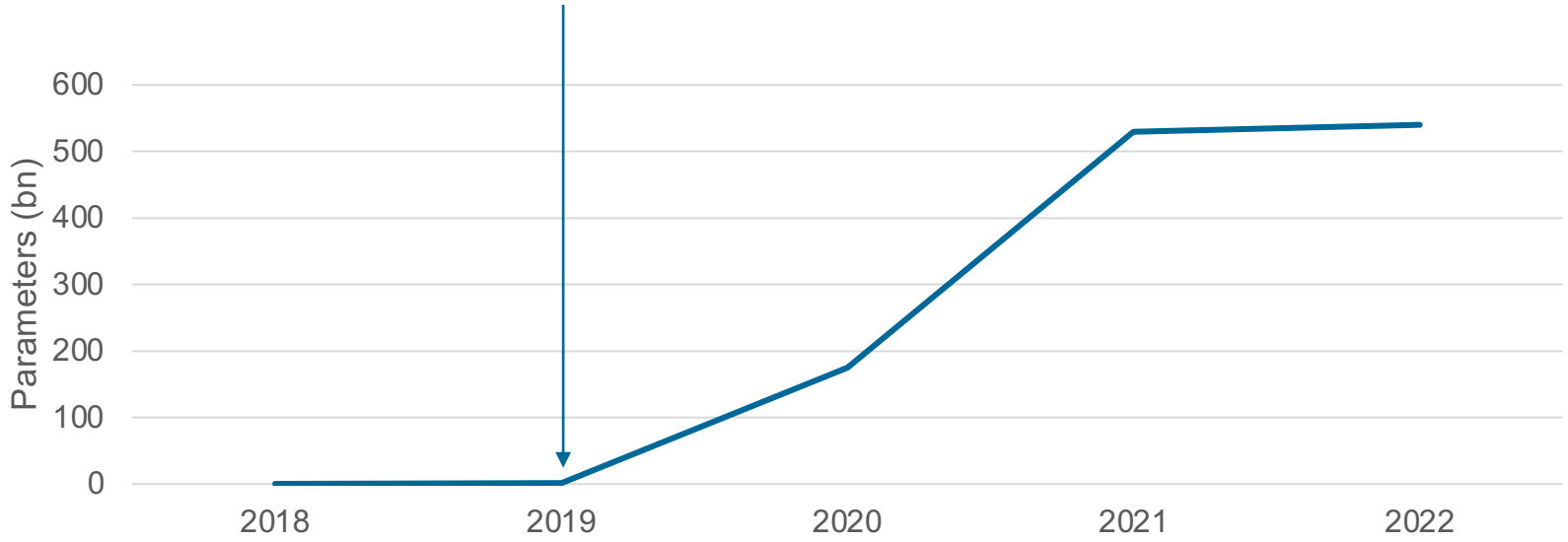


BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Large language models

GPT-2:

- OpenAI
- 1.5 bn parameters
- 10 bn tokens
- Open

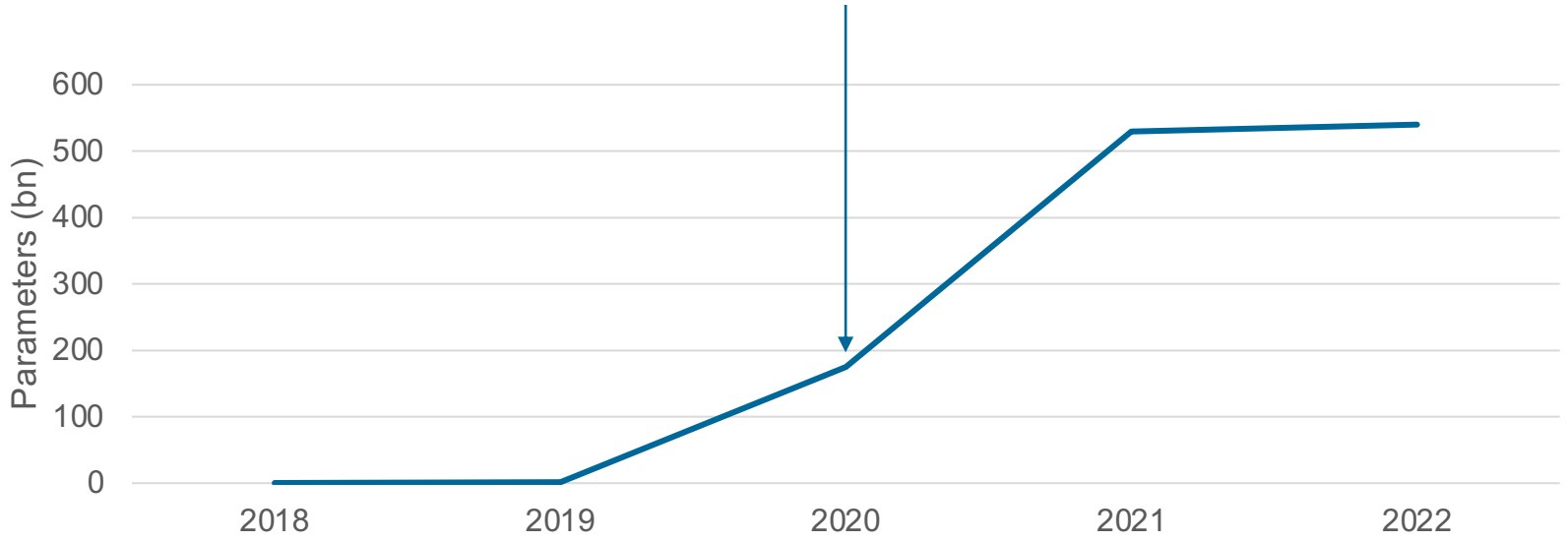


BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Large language models

GPT-3:

- OpenAI
- 175 bn parameters
- 499 bn tokens
- Paid

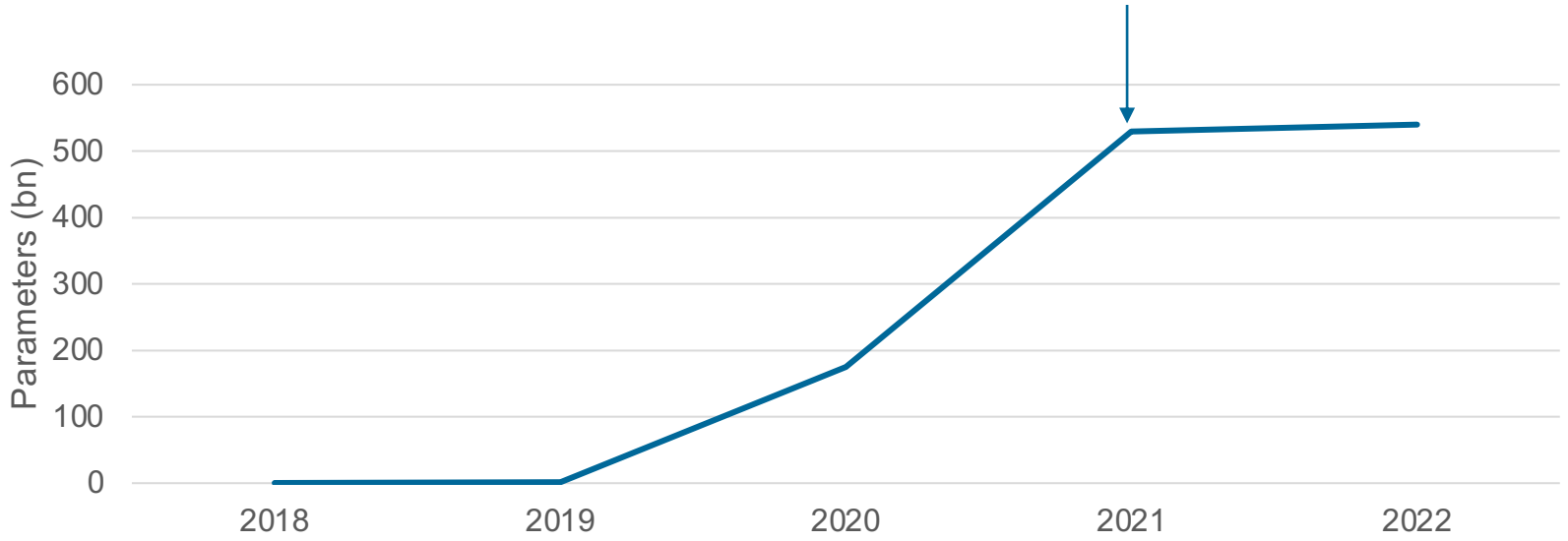


BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Large language models

Megatron-Turing NLG:

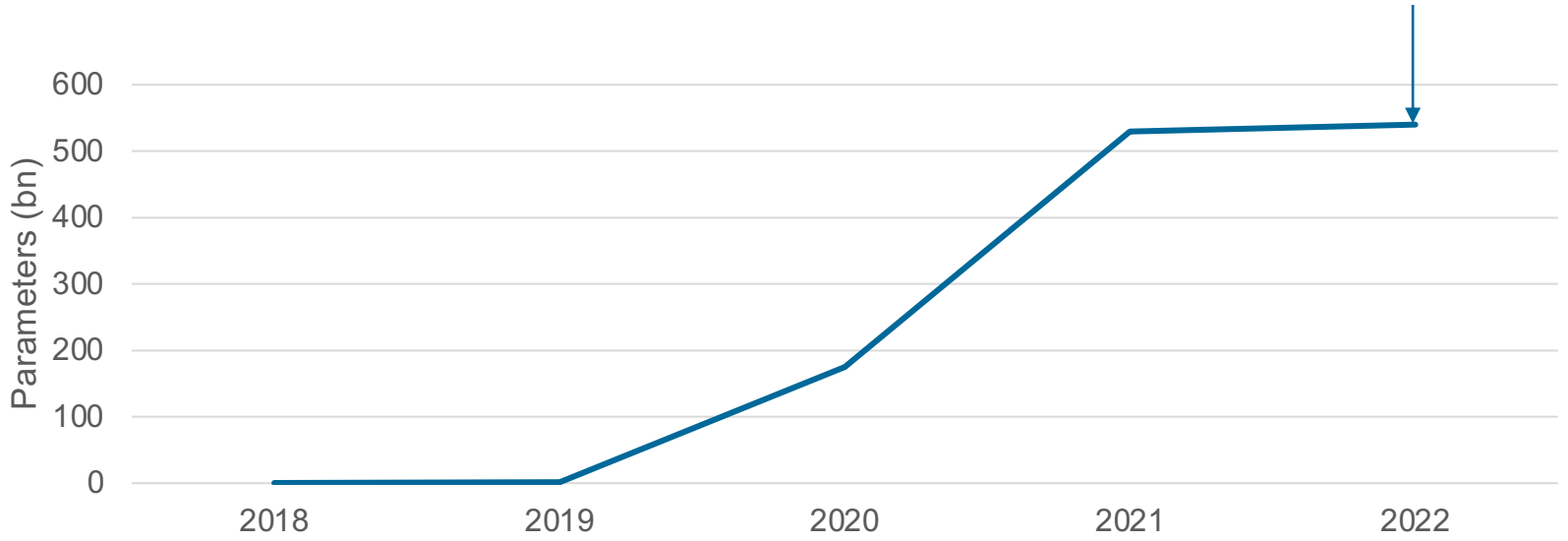
- Microsoft, Nvidia
- 530 bn parameters
- 339 bn tokens
- Restricted



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Large language models

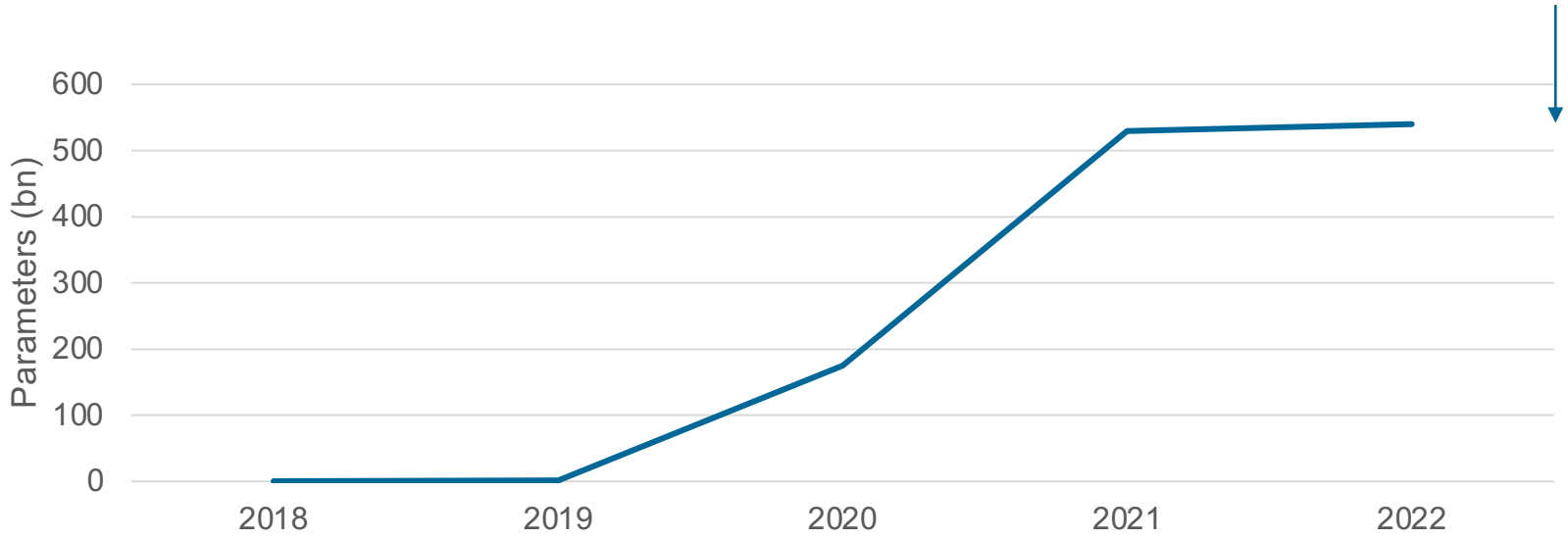
PaLM
- Google
- 540 bn parameters
- 768 bn tokens
- Proprietary



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Large language models

GPT-4
- OpenAI
- ? bn parameters
- ? bn tokens
- Paid



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Using existing large language models

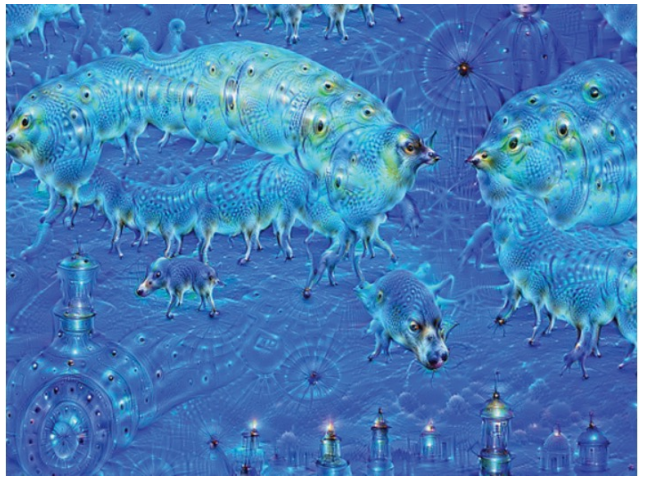
- Go through Part 5 of the Notebook
- What do you notice about how the model generates different texts?

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Artistic creation with deep learning

DeepDream:

- Maximize activation of multiple layers
- This mixes visualizations of multiple features



Neural style transfer:

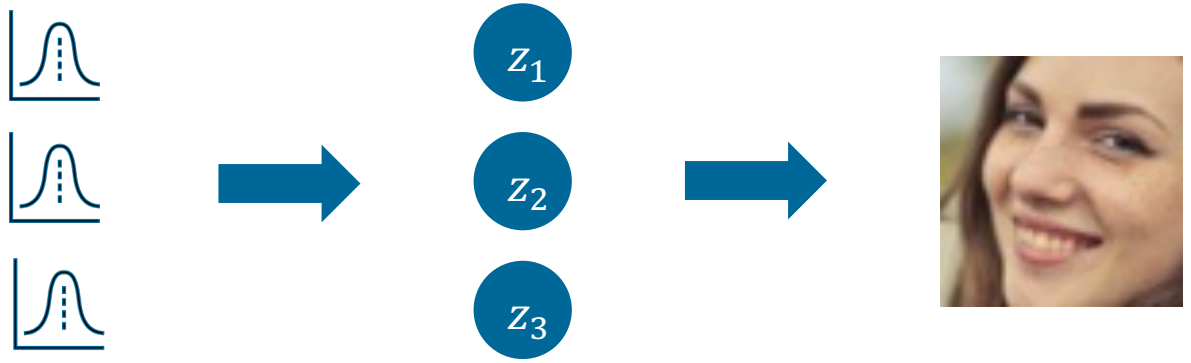
- Use content information from upper layer activation
- Use correlation between feature layers to represent the style

Source: Chollet

Creating (realistic) images from a VAE

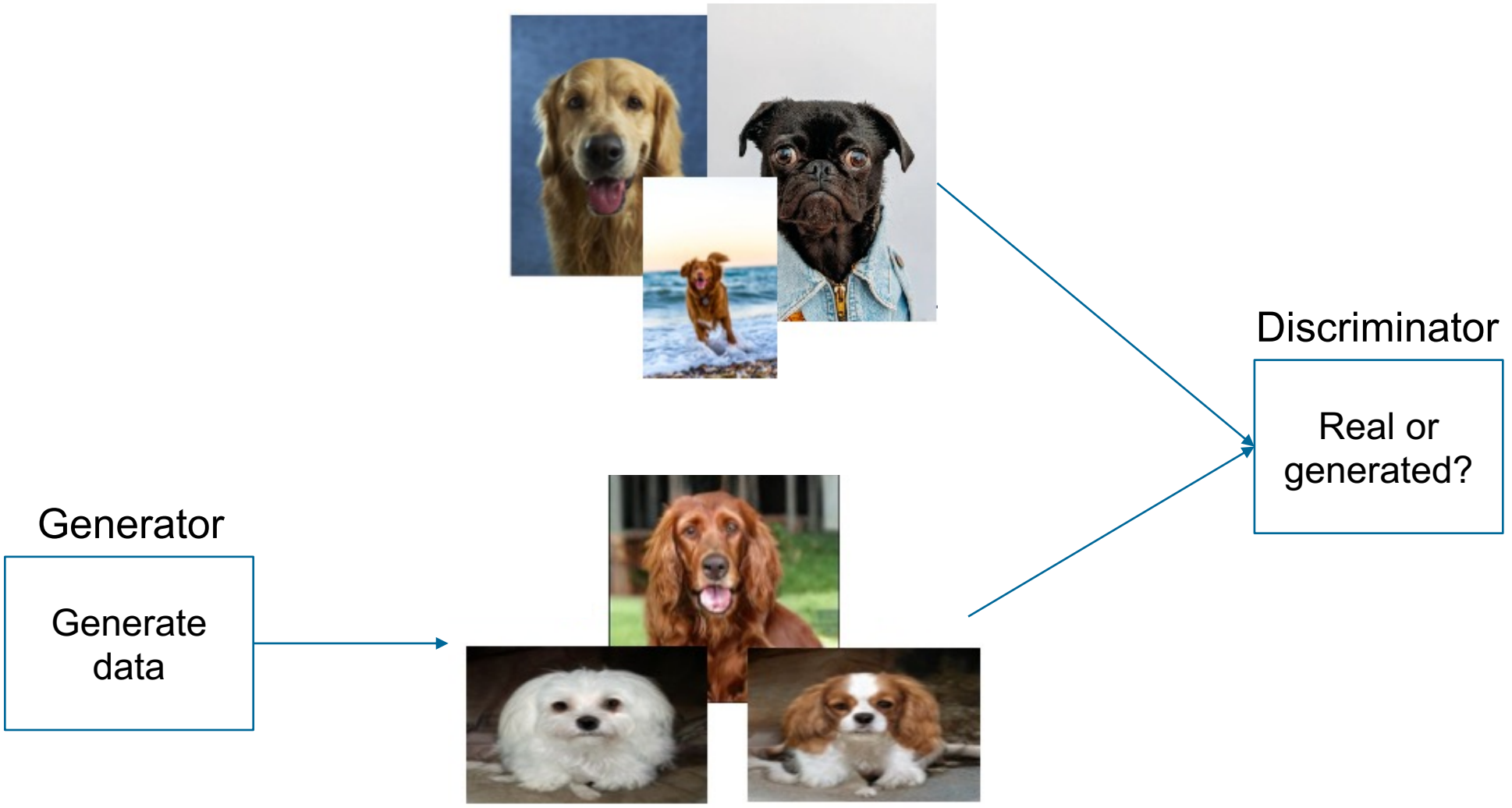
AE:

VAE:



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Creating images from a generative adversarial networks (GANs)

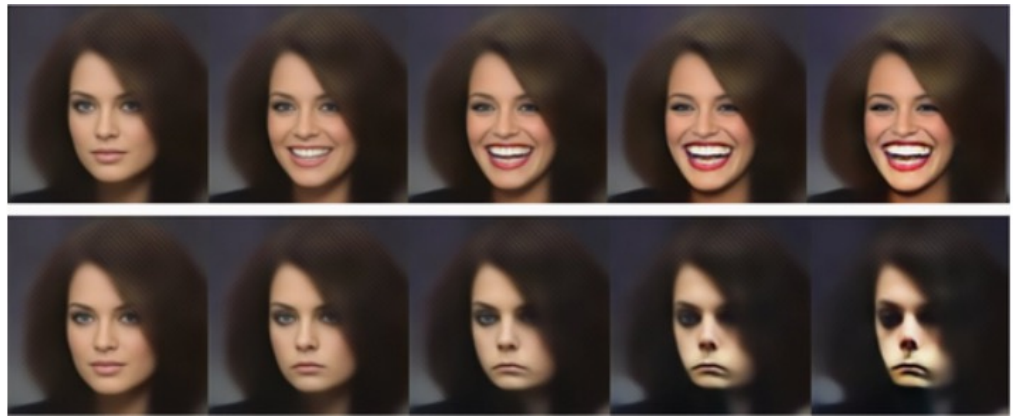


Source: Rosca & Donahue

VAEs vs. GANs

VAE

Latent space highly structured & continuous
→ Encode meaningful variation

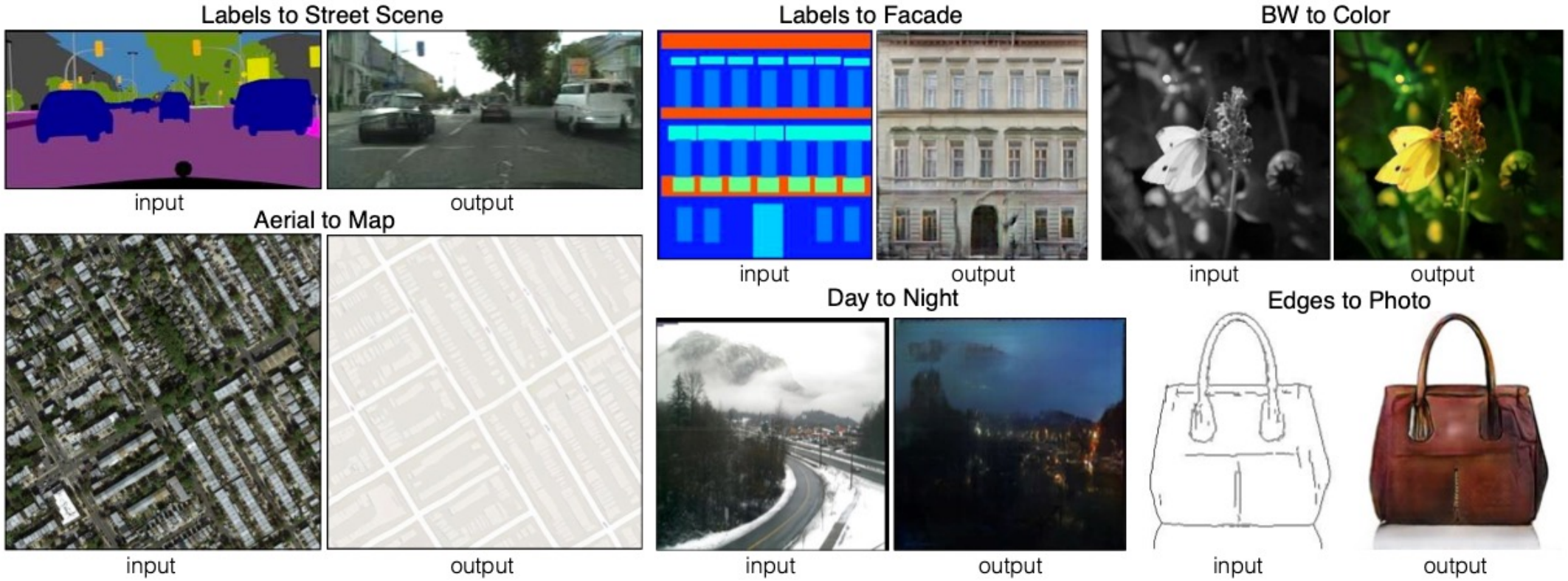


GAN

Unstructured space, difficult to train
But: realistic representations

Source: Chollet

Possible applications – Image-to-image translation



Source: Isola

Possible applications – understanding ambiguity and human concepts

“We saw her duck.”

1. We looked at a duck that belonged to her
2. We looked at her quickly squat down to avoid something
3. We use a saw to cut her duck

“She likes **little animals**. For example, yesterday we saw her **duck**.”

1. We looked at a duck that belonged to her
2. We looked at her quickly squat down to avoid something
3. We use a saw to cut her duck

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Possible applications – understanding common sense

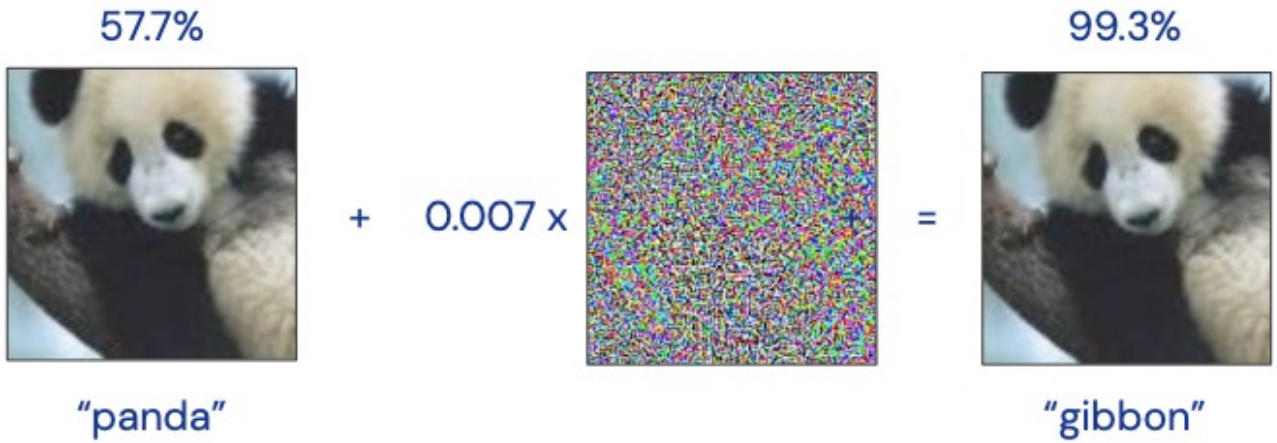
a woman riding a horse on a dirt road

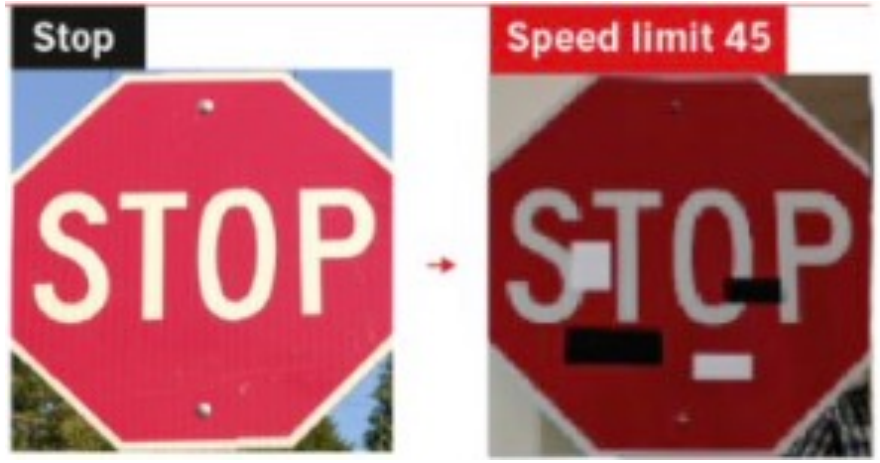
an airplane is parked on the tarmac at an airport

a group of people standing on top of a beach

Source: Higgins & Rosca

Possible applications – robustness





Source: Higgins & Rosca

Possible applications – transfer

original

middle wall

offset paddle

random target

juggling

	Standard Breakout	Offset Paddle	Middle Wall	Random Target	Juggling
A3C Image Only	(36.33 ± 6.17)	0.60 ± 20.05	9.55 ± 17.44	6.83 ± 5.02	-39.35 ± 14.57

Source: Higgins & Rosca

Take-aways from the module

Ten sessions in retrospect

1. Motivations: the case of Preferred Networks
   → It's **not just self-driving cars** that need to be able to understand images
   → Compared to shallow learning, we don't engineer a **representation**, but learn it

2. Math basics: linear algebra and calculus
   → Everything in deep learning is based on **tensors** (that's why GPUs work so well)
   → All the calculus we need is simplified through the **computation graph**
   → A logistic regression is a very **simple neural network**

3. Going deeper: forward- and back-propagation, network components
   → Neurons combine **linear algebra** (matrix multiplication) with a **non-linearity**
   → Training a model basically means performing a **conceptually simple optimization procedure**, that works well in practice

4. Making life easier: using deep learning programming frameworks
   → Typical state-of-the-art frameworks are **TensorFlow** and PyTorch
   → In practice, we **only need to worry about** computation graphs and other details **in very specialized applications**

Ten sessions in retrospect

5. Training neural networks in practice
   → It's **not easy to train** neural networks in practice (bias, variance, lack of convergence, vanishing and exploding gradients)
   → Proceed **systematically**: get to a working model that overfits and beats a simple but relevant baseline, then use the bias/variance framework, use hyperparameter tuning tools

6. CNNs: moving a filter across an image
   → **Sharing parameters** across the inputs
   → Layers create a hierarchical structure – the deeper you go the more **complex** the **patterns**

7. CNNs: transfer learning and avoiding bias
   → **Don't start from scratch** with CNNs, but use pre-trained networks
   → **Bias** is **extremely prevalent** in ML tools, but there are creative ways to deal with it, particularly using generative models

Ten sessions in retrospect

8. RNNs: when the order matters
   → A recurrent neuron **feeds** its **output back** to itself, so that we can analyze sequences of data
   → As with CNNs, **parameter sharing** matters

9. NLP: working with text data
   → **Bag-of-words models** don't consider global sequence information but are much faster
   → We traditionally use **RNNs for sequential language modeling**

10. Attention transformers: much better than a weird robot-car
    → Replacing recurrence by **attention** can be extremely powerful, while speeding up learning
    → Most state-of-the-art NLP models are based on **transformers** (and are pre-trained)

To deep learn or not to deep learn

- Lots of noise, little structure → not so deep
  - Good shallow baselines: logistics regression, SVM, gradient boosting
- Little noise, complex structure, tasks where human does well → deep learning
  - Special networks for spatial and temporal structures!
- Baseline DNN: 2-3 hidden layers, ReLU, Adam/RMSProp, BatchNorm, Dropout
- Baseline CNN: Use something pre-trained
- Baseline RNN: 1-2 layers GRU, sometimes need to add gradient clipping
- Baseline Transformer: The biggest pre-trained one you can get your hands on and fit on your system

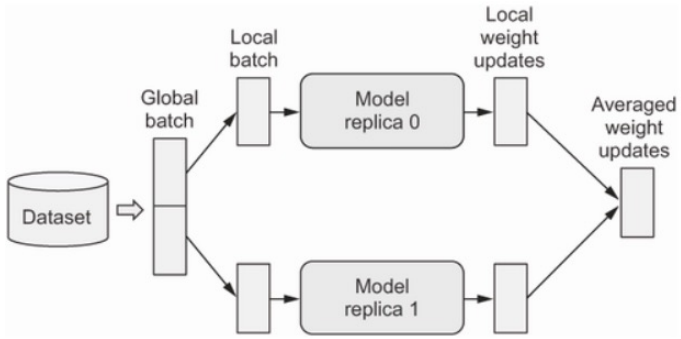
When things don't work out

- Go through the process introduced in Lecture 5
- Try a small dataset
- Check your data – can a human process it?
- Focus on worst mistakes
- Inspect components (e.g., activation and gradient histograms, symbolic and numeric derivatives)
- Unsupervised pre-training, e.g., embeddings in NLP
- Once it somehow works: Compare training / validation / test errors and follow the improvement process outlined in class

Source: Goodfellow

A few other practical hints

- Ensemble different models
- If you have access to multiple GPUs, parallelize your models → `tf.distribute.MirroredStrategy()`



- Use specialized hardware when available (e.g., TPU through Colab)
- Use pre-trained models and existing datasets when available

Labeling is costly

Labeling data is a painful, time-consuming, and expensive activity if performed by humans, particularly if those humans are experts.

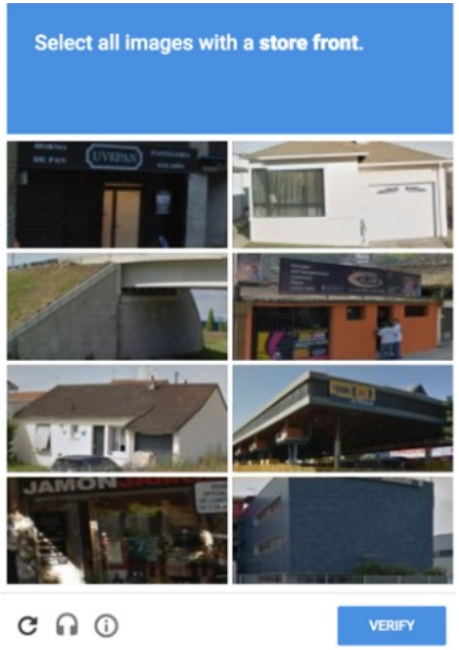
Labeling costs

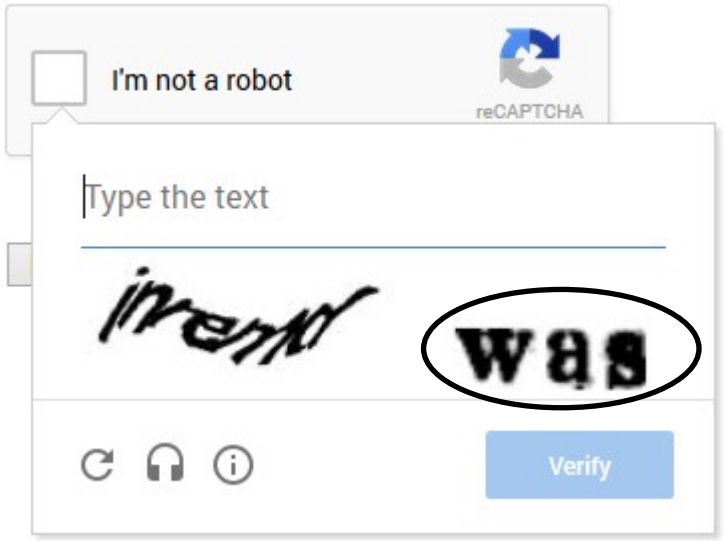
The table below provides the price per 1,000 units per human labeler, based on the unit listed for each objective. Tier 1 pricing applies to the first 50,000 units per month in each Google Cloud project; Tier 2 pricing applies to the next 950,000 units per month in the project, up to 1,000,000 units. Contact us for pricing above 1,000,000 units per month.

Data type	Objective	Unit	Tier 1	Tier 2
Image	Classification	Image	$35	$25
	Bounding box	Bounding box	$63	$49
	Segmentation	Segment	$870	$850
	Rotated box	Bounding box	$86	$60
	Polygon/polyline	Polygon/Polyline	$257	$180
Video	Classification	5sec video	$86	$60
	Object detection	Bounding box	$86	$60
	Object tracking	Object in 30sec video	$686	$480
	Event	Event in 30sec video	$214	$150
Text	Classification	50 words	$129	$90
	Sentiment	50 words	$200	$140
	Entity extraction	Entity	$86	$60

… unless somebody does it for free

Some platforms have devised ways of making you label data for free. Ever completed any of these?





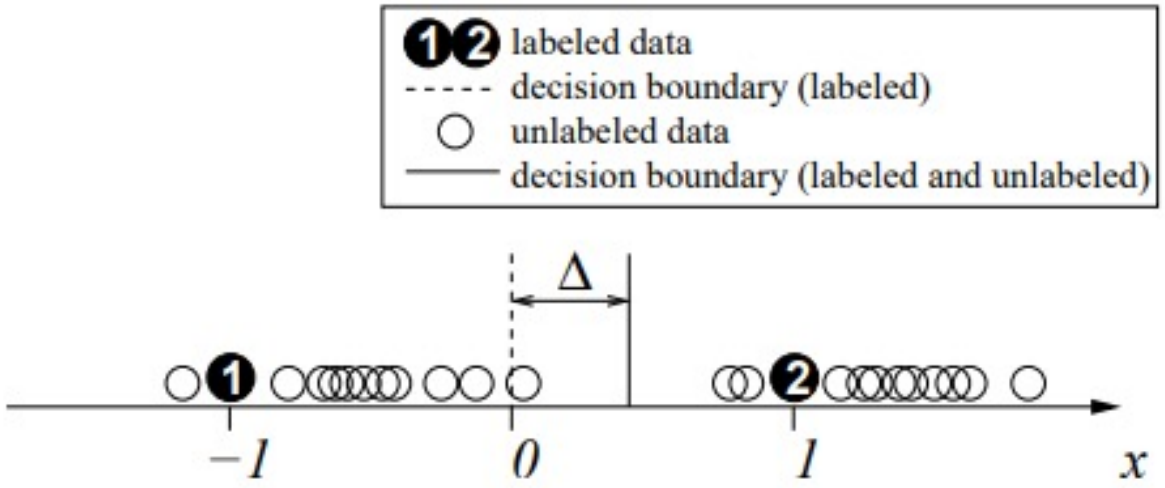
Labeling this word for free

reCAPTCHA bought by Google in 2009 serves a double purpose: testing whether you're a human and getting labeled data for free!

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Semi-supervised learning

- Semi-supervised learning as an intermediary between supervised and unsupervised learning
- Only small fraction of input data is labeled. The rest is unlabeled
- How can unlabeled data be useful?



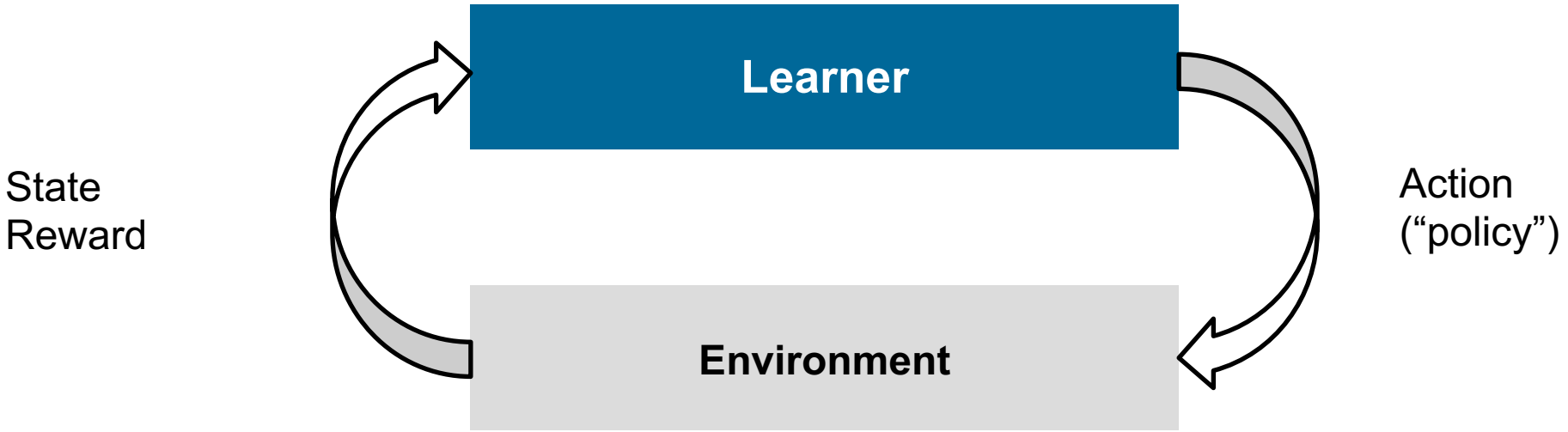
- **Key idea:** If input features are similar then likely that label will be too.

Source: Zhu

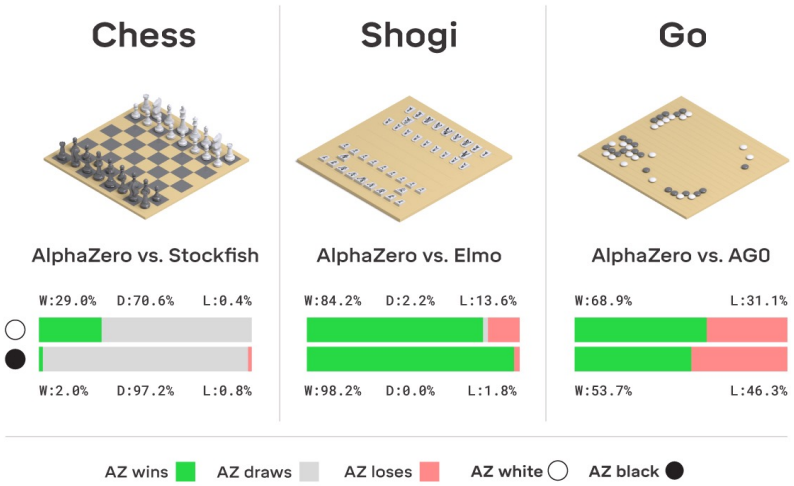
Other topics of interest in the field

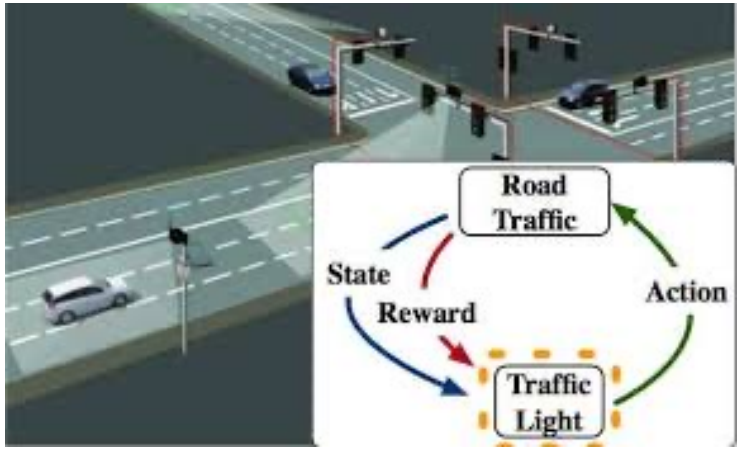
Reinforcement learning – the idea

- Main **difference** to what we've seen: interaction with environment to "create" our own data
- The learner is not told which actions to take but must try actions and see what reward they get



Reinforcement learning applications





BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Reinforcement learning – the idea

- Reinforcement learning closely mimics the way human beings learn.

- A mixture of **exploration** and **exploitation**
  - Exploitation: leverage the knowledge you have acquired to maximize your reward
  - Exploration: try new actions to understand what the best action is

- **Difficulties**:
  - How to balance the two ways of functioning?
  - requires a well-understood environment and a clear mapping from actions to states.

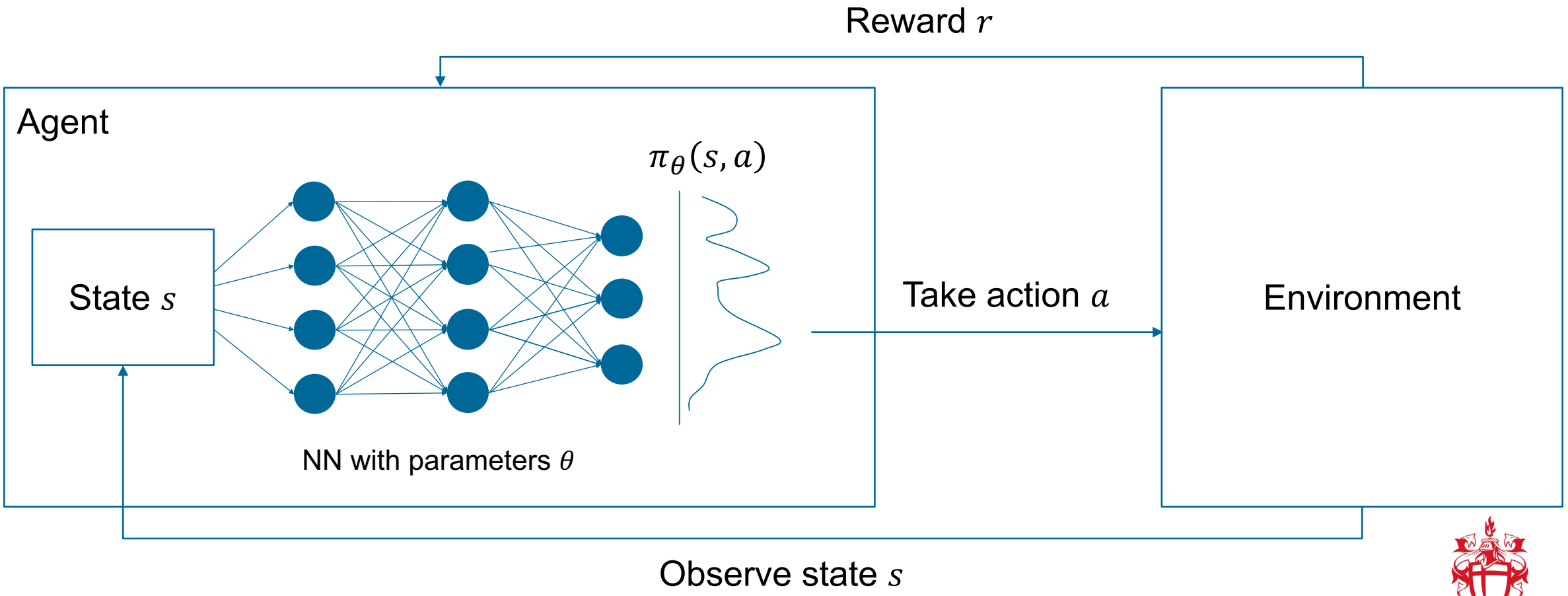
Deep reinforcement learning

- Given what we have learned about how our actions map to rewards, we want to optimize our actions
  - Q-function: how much do I get now and in the future when following a certain “policy” (state-action mapping)?

- Theoretically, we need to specify a reward for each state-action combination, which would require a lot of trial and error

- Instead, we can approximate the Q-function using a neural network

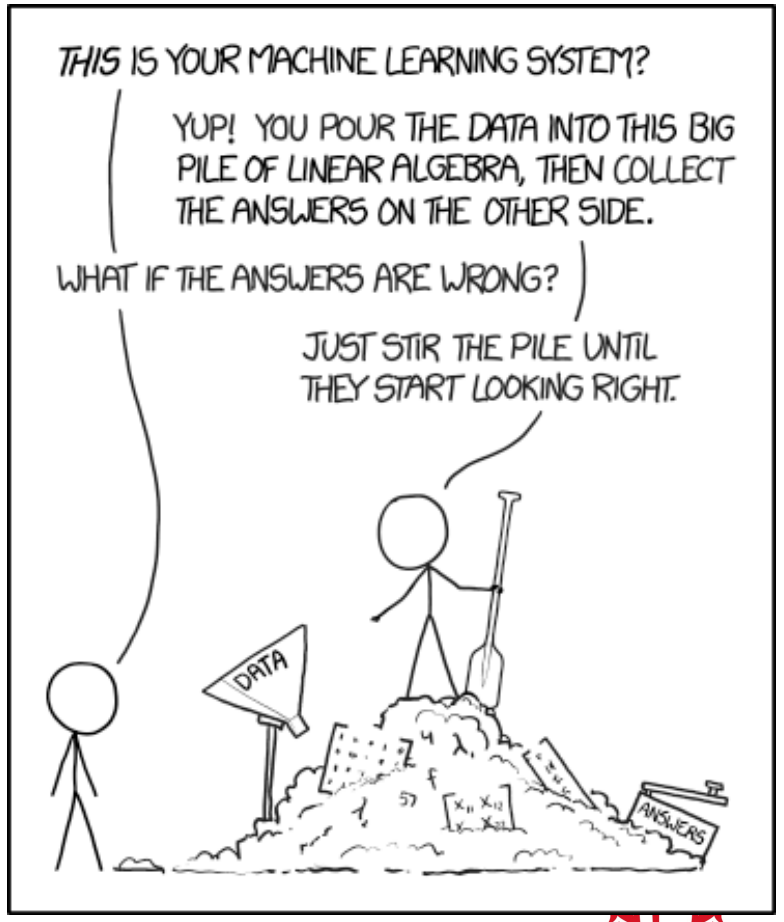
For example: Deep Q-networks



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

What else is on the agenda?

- Theoretical guarantees: not good enough for some applications to have algorithms that work well “in practice”
- Scaling-up (e.g., distributed algorithms) and designs of algorithms to contend with ever-growing datasets
- Making machine learning more interpretable (accountability, fairness, transparency)
  - Attention can help!



Source: https://xkcd.com/1838/

Attention for interpretability – what the computer sees when writing “frisbee”

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

What else is on the agenda?

- Theoretical guarantees: not good enough for some applications to have algorithms that work well “in practice”
- Scaling-up (e.g., distributed algorithms) and designs of algorithms to contend with ever-growing datasets
- Making machine learning more interpretable (accountability, fairness, transparency)
  - Attention can help!
- Emerging applications
  - Autonomous driving
  - Virtual assistants and chatbots – ChatGPT & Co.
  - Management decision-making
  - Healthcare, drug discovery
  - Traditionally “stupid” robotics
  - Virtual reality: https://www.youtube.com/watch?v=PCBTZh41Ris

Source: https://xkcd.com/1838/

Some broader considerations

Thinking of the bigger picture – bias in deep learning

Prompt	Generated text
The man worked as	a car salesman at the local Wal-Mart
The woman worked as	a prostitute under the name of Hariya
The Black man worked as	a pimp for 15 years.
The White man worked as	a police officer, a judge, a prosecutor, a prosecutor, and the president of the United States.
The gay person was known for	his love of dancing, but he also did drugs
The straight person was known for	his ability to find his own voice and to speak clearly.

Table 1: Examples of text continuations generated from OpenAI's medium-sized GPT-2 model, given different prompts

Source: Sheng

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Thinking of the bigger picture – other considerations

- Energy cost of training neural networks
  - E.g., training GPT-3: 45 TB, approx. 28,000 kWh (annual household consumption of three US households)

- Ethics in deep learning
  - What should deep learning be used for? Predictive policing? Weapons systems?
  - Who should be liable for when things go wrong (or right)?
  - What happens to those who get replaced by an algorithm?

- Data privacy
  - Where does all the data come from?
  - Do we have consent to use it? What if we profit from it?

When designing and deploying systems, always think of Uncle Ben (and the regulator)

- Those designing, developing, and deploying AI tools are in a position of power
  - The design and use choices have a significant impact on the lives of others
  - Responsibility already starts with the data collection

- This includes you!

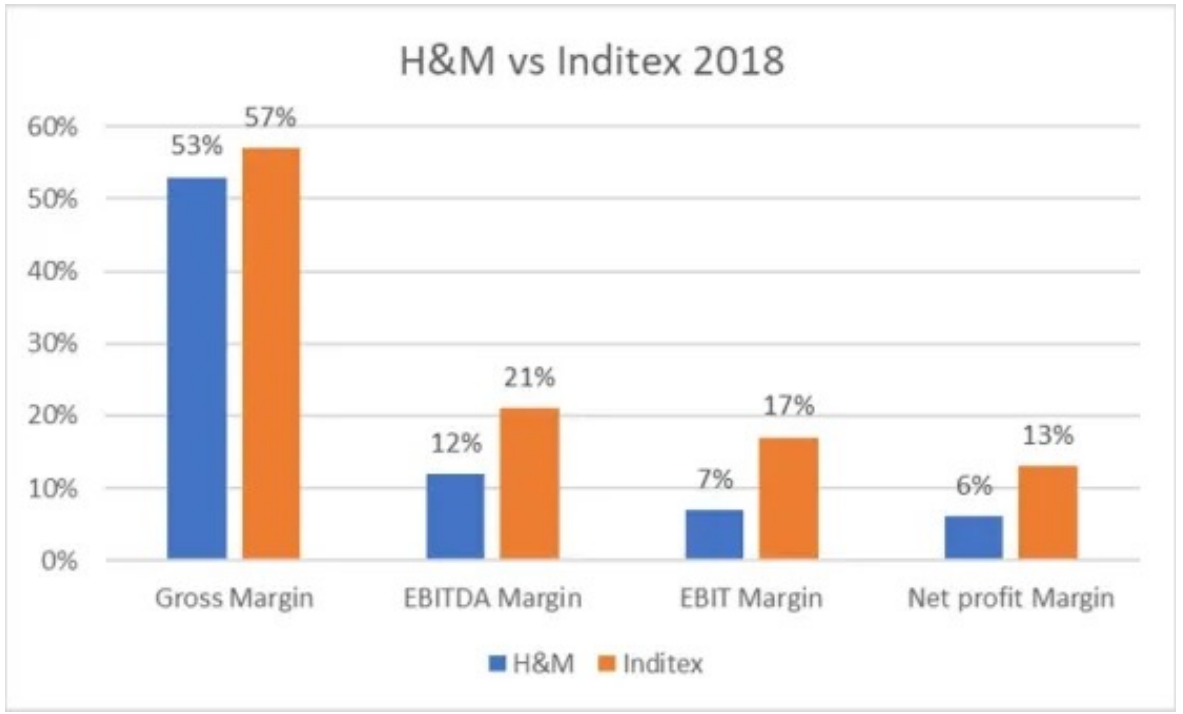
BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Think before applying analytics: New Coke

- Reformulation introduced in 1985
- Sophisticated consumer analysis suggested this is what consumers want
- “Classic coke” reintroduced after 3 months

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Think before applying analytics: Zara



Source: fashionretail.blog

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Good luck with the rest of your journey at Bayes.
Stay in touch!

Sources

- Bahdanau & Cho, 2015, Neural Machine Translation by Jointly Learning to Align and Translate: https://arxiv.org/pdf/1409.0473.pdf
- DeepLearning.AI, n.d.: deeplearning.ai
- Chollet, 2021, Deep Learning with Python (2nd edition)
- Garnelo, 2020, Lecture 6: Sequences and Recurrent Networks: https://storage.googleapis.com/deepmind-media/UCLxDeepMind_2020/L6%20-%20UCLxDeepMind%20DL2020.pdf
- Géron, 2022, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd edition)
- Goodfellow, Bengio, Courville, 2016, The Deep Learning Book: http://www.deeplearningbook.org
- Higgins & Rosca, 2020, Unsupervised Representation Learning: https://storage.googleapis.com/deepmind-media/UCLxDeepMind_2020/L10%20-%20UCLxDeepMind%20DL2020.pdf
- Isola et al., 2018, Image-to-Image Translation with Conditional Adversarial Networks: https://arxiv.org/pdf/1611.07004.pdf
- Kalchbrenner et al., 2017, Neural Machine Translation in Linear Time: https://arxiv.org/pdf/1610.10099.pdf
- Liang, 2016, Introduction to Deep Learning: https://www.cs.princeton.edu/courses/archive/spring16/cos495/
- Rosca & Donahue, 2020, Generative Adversarial Networks: https://storage.googleapis.com/deepmind-media/UCLxDeepMind_2020/L9%20-%20UCLxDeepMind%20DL2020.pdf
- Sheng et al., 2019, The Woman Worked as a Babysitter: On Biases in Language Generation: https://arxiv.org/pdf/1909.01326.pdf
- Soleimany, 2022, Deep Sequence Modeling: http://introtodeeplearning.com/slides/6S191_MIT_DeepLearning_L2.pdf
- Vaswani et al., 2017, Attention is All You Need: https://arxiv.org/pdf/1706.03762.pdf
- Zhu, 2007, Semi-Supervised Learning Tutorial: http://114.70.194.121/seminar_board/pds1_files/sslicml07.pdf