



7-10 June 2022

Attention? Attention!

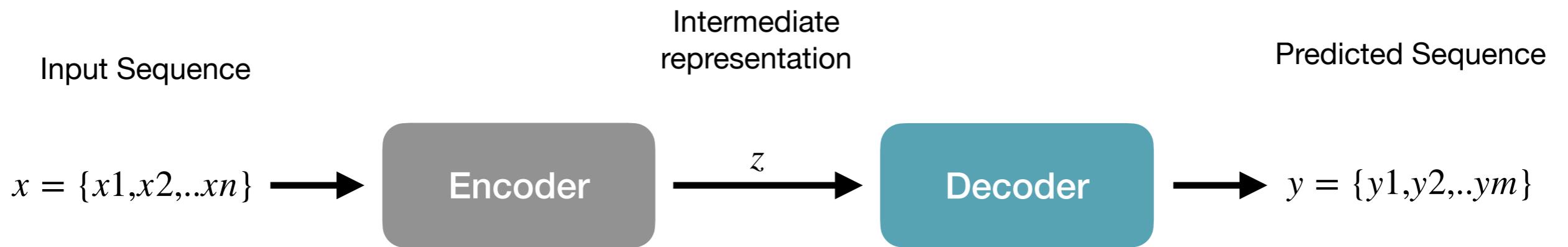
- Attention Mechanism in Deep Learning



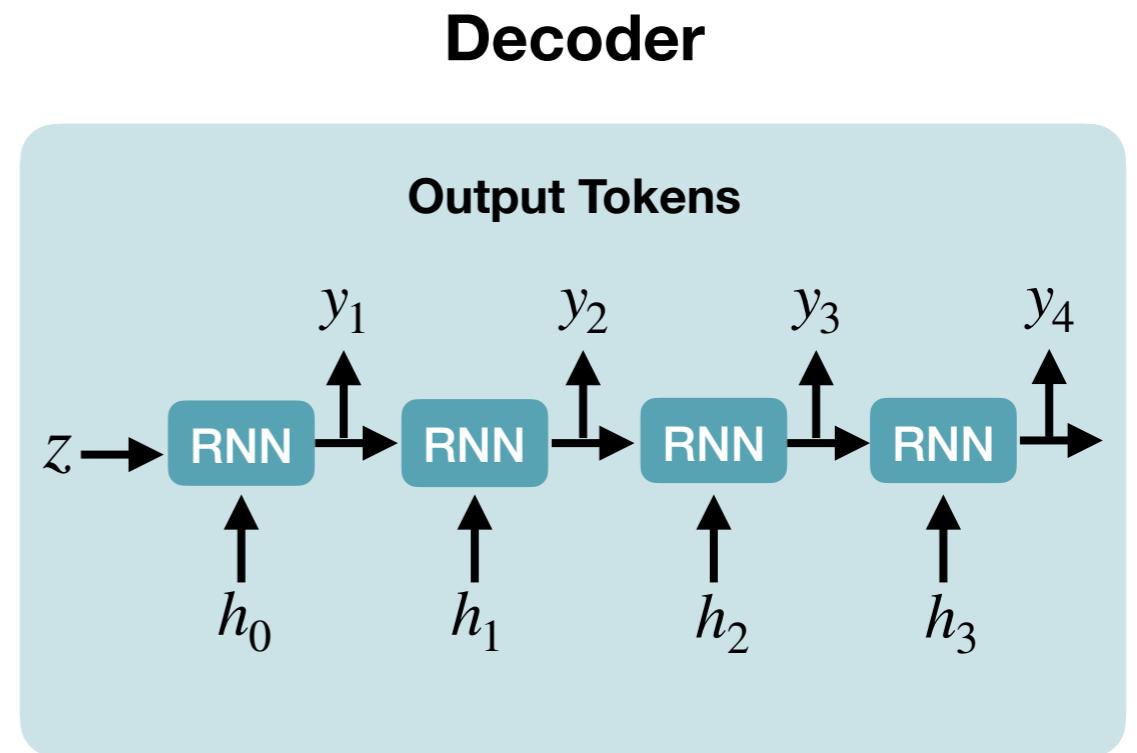
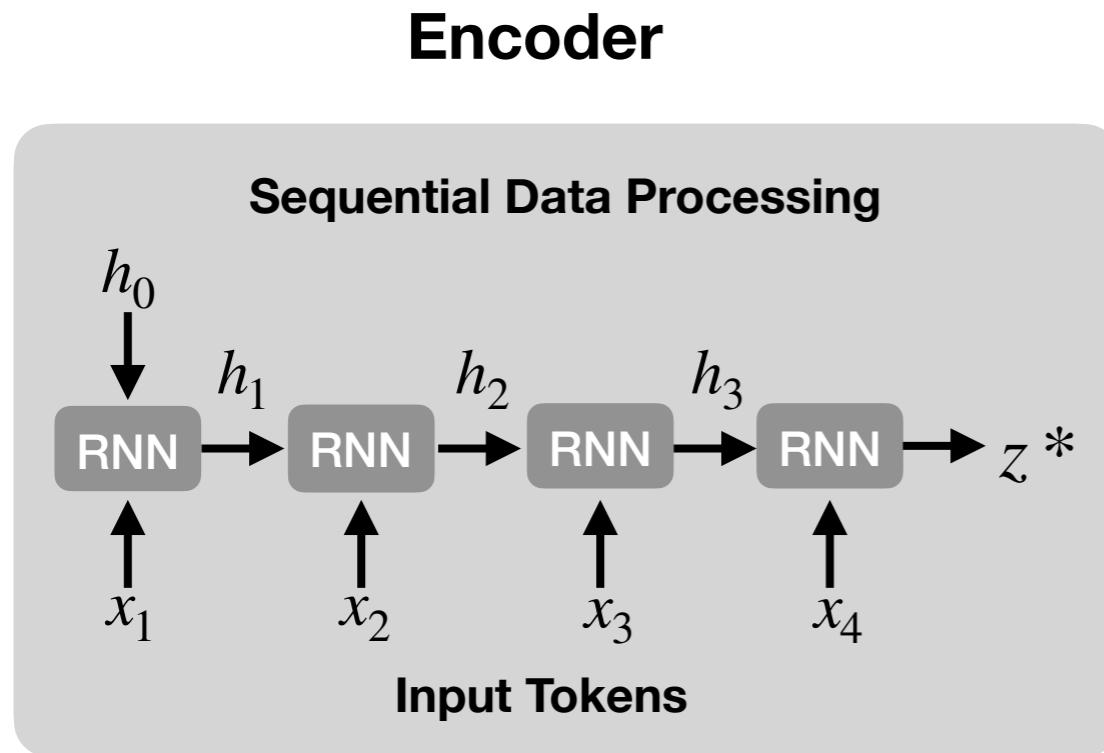
Outline

- Background
- Types of attention
- Self and multi-head attention
- Practical examples

Sequence to Sequence Learning



High-level view of encoder and decoder



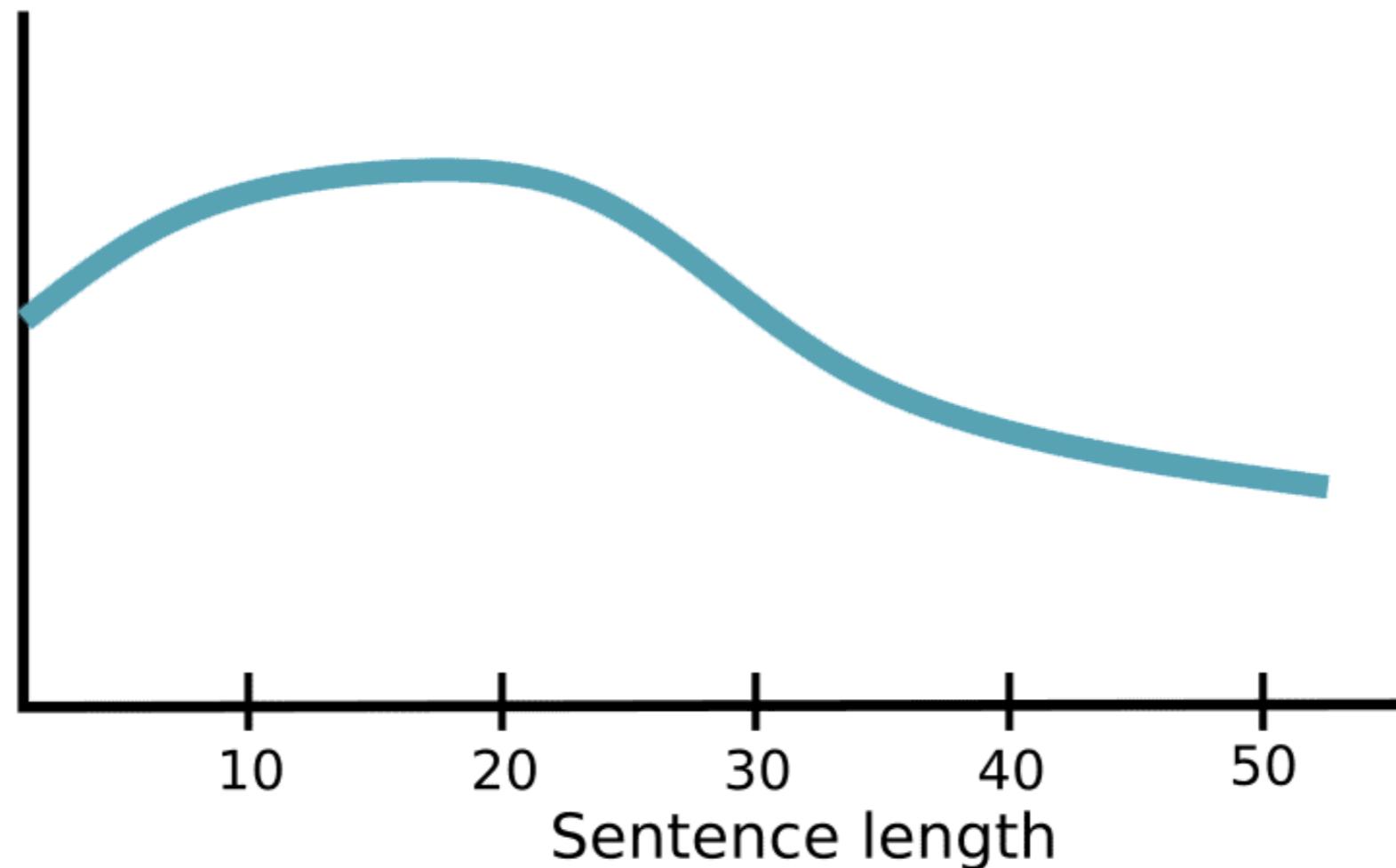
Note: each RNN block **requires** the output of the **previous**

Note: predictions **MUST** be performed **sequentially**

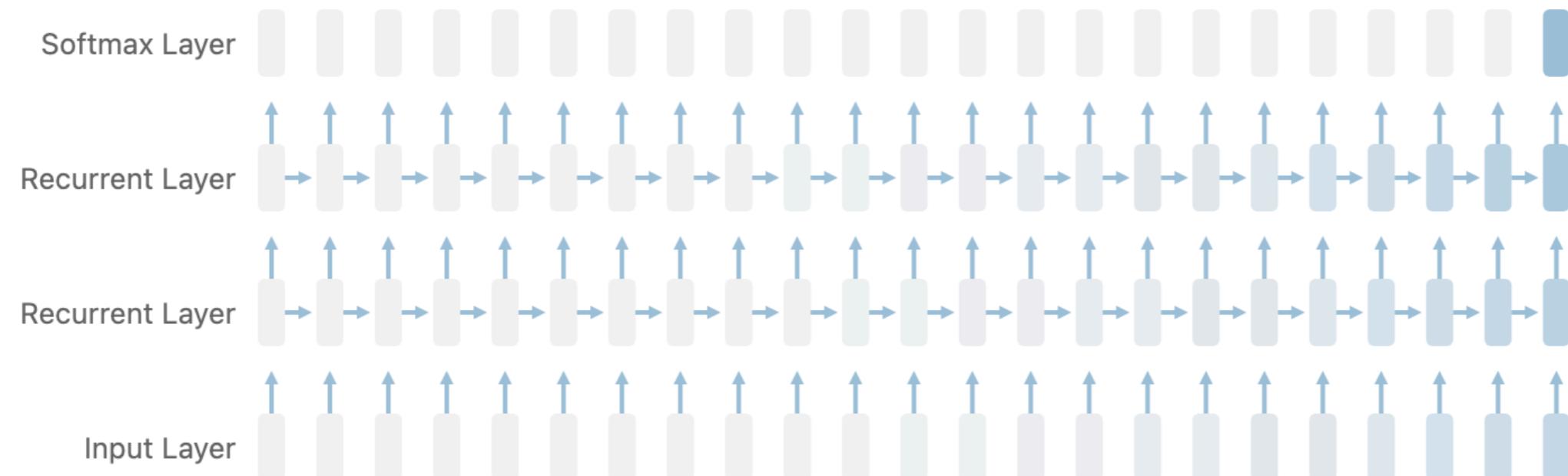
* Encoded sequence representation



Memorization in RNNs



Visualizing memorization in RNNs



Vanishing Gradient: where the contribution from the earlier steps becomes insignificant in the gradient for the RNN.



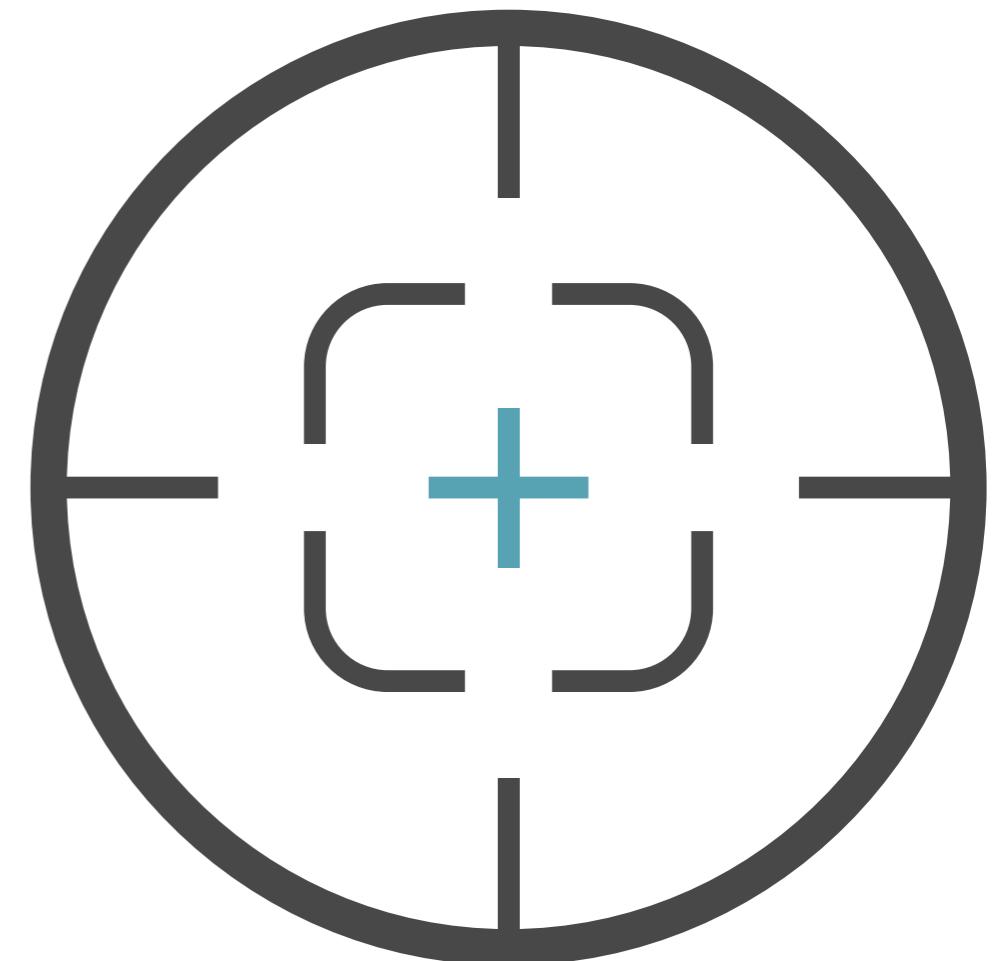
Memory is attention through time.

~ Alex Graves 2020*

*Alex Graves is a research scientist at DeepMind.

Human attention

“Humans seldom utilize all available inputs to complete a task.”





Attention Types

- Soft attention
- Hard attention
- Global attention
- Local attention
- Self-attention

Attention Types - Soft vs. Hard Attention

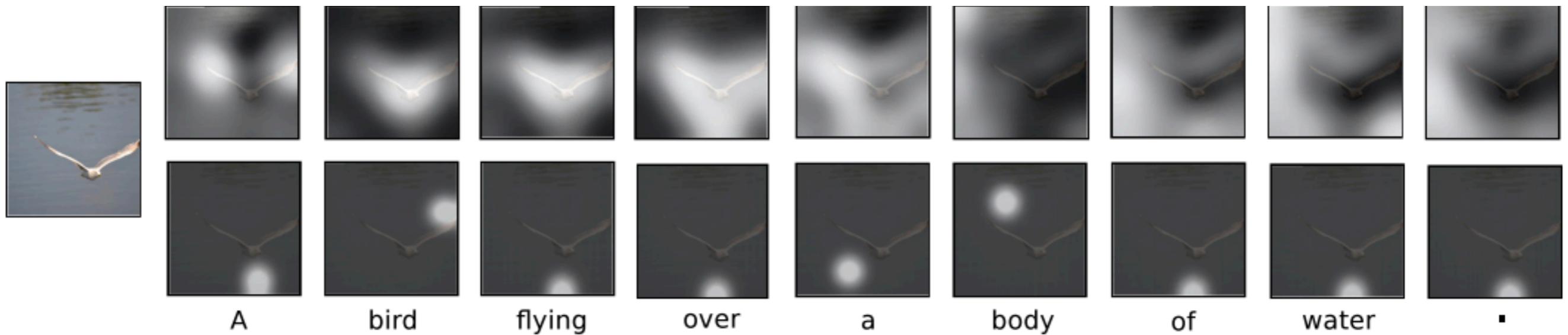


Image source: Fig. 6(b) in Xu et al. 2015
Paper source: <http://proceedings.mlr.press/v37/xuc15.pdf>

Sources

Soft Attention

Attention score is used as weights in the weighted average context vector calculation. This is a differentiable function.

$$\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t] = \sum_{i=1}^L \alpha_{t,i} \mathbf{a}_i$$

"a" represents encoder/input hidden states, "α" represents the attention scores, "s_{t,i}" is a one-hot variable with "1" if "i-th" location is to be selected.

Hard Attention

Attention score is used as the probability of the i-th location getting selected. We could use a simple argmax to make the selection, but it is not differentiable and so complex techniques are employed.

$$\begin{aligned}\hat{\mathbf{z}}_t &= \sum_i s_{t,i} \mathbf{a}_i \\ p(s_{t,i} = 1 \mid s_{j < t}, \mathbf{a}) &= \alpha_{t,i} \\ \tilde{s}_t^n &\sim \text{Multinoulli}_L(\{\alpha_i^n\})\end{aligned}$$

Source: Xu et al. 2015,
<https://towardsdatascience.com/attention-in-neural-networks-e66920838742>

Attention Types - Global vs. Local Attention

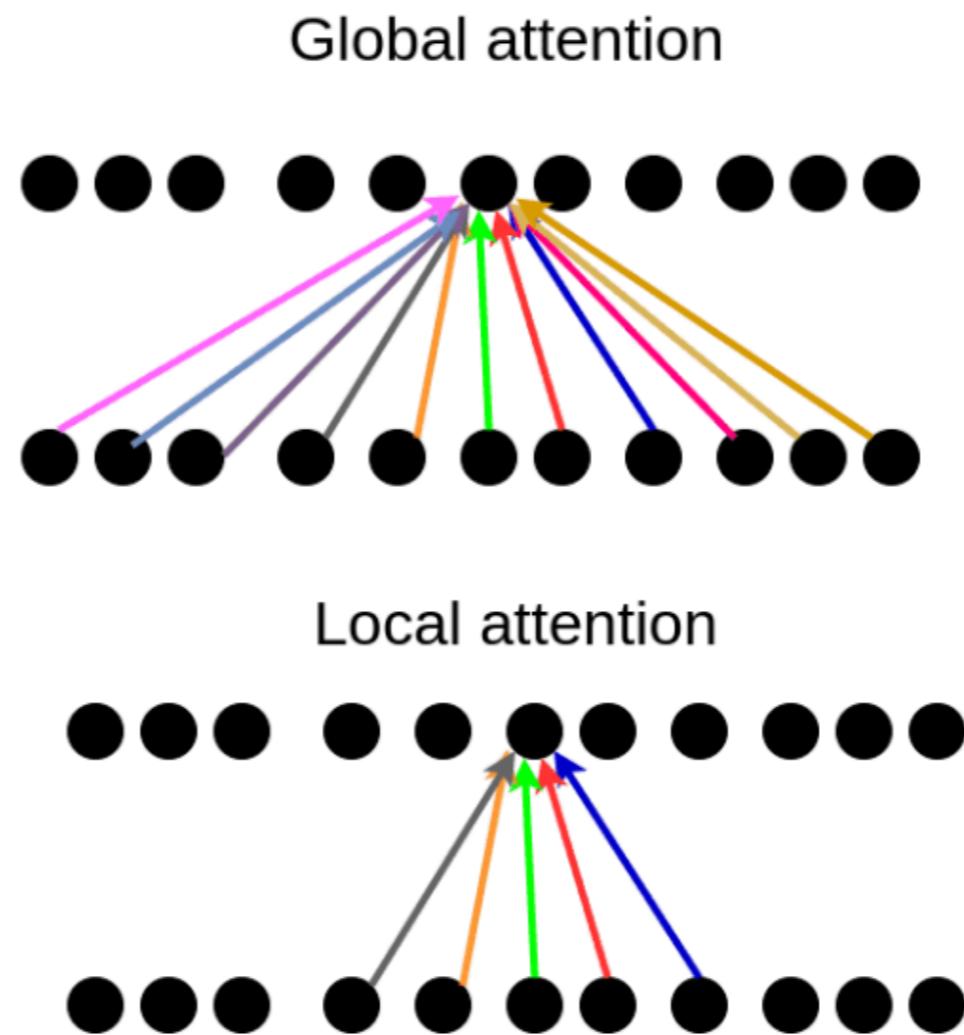


Image source: <https://theaisummer.com/attention/>
Paper source: <https://arxiv.org/pdf/1508.04025.pdf>

Self-Attention



Self-Attention

Probability Score Matrix

	This	movie	was	not	bad
This	0.65	0.1	0.05	0.05	0.15
movie	0.1	0.5	0.1	0.05	0.25
was	0.05	0.2	0.45	0.1	0.2
not	0.2	0.1	0.05	0.05	0.6
bad	0.05	0.1	0.05	0.2	0.6

Softmax (Attention) equation

Self-Attention Basics

- Input: sequence of tensors $x_1, x_2, \dots x_t$
- Output: sequence of tensors, each one a weighted sum of the input sequence $y_1, y_2, \dots y_t$, $y_i = \sum_j w_{ij} x_j$

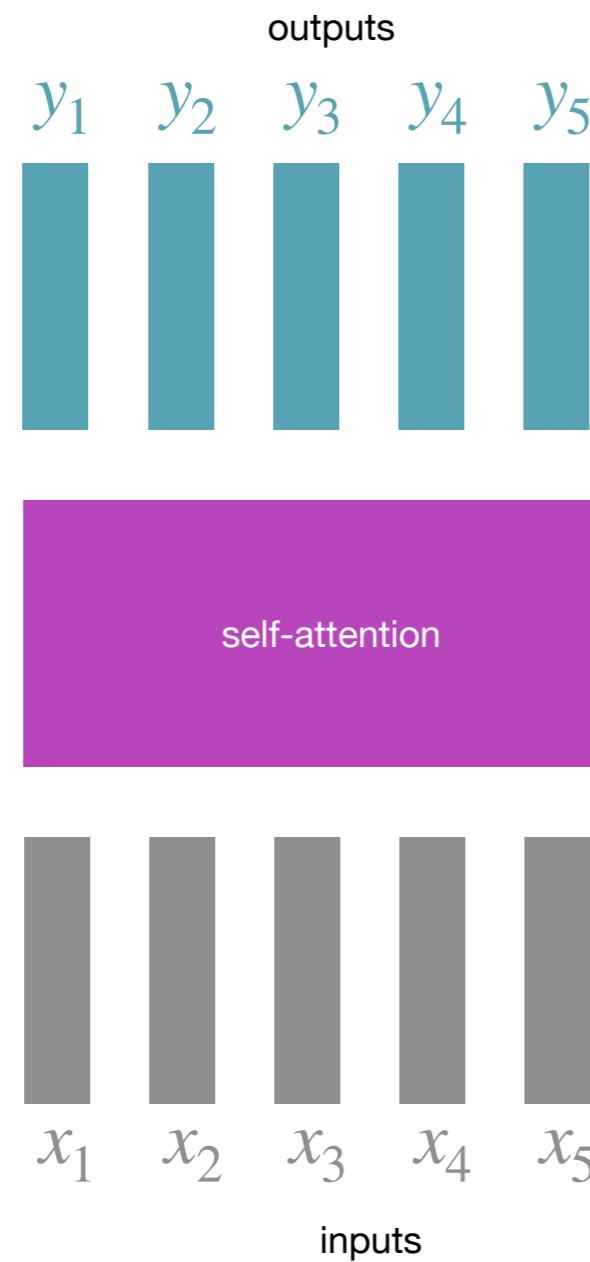
$$w_{ij}' = x_i^\top x_j$$

dot product

$$w_{ij} = \frac{\exp w_{ij}'}{\sum_j \exp w_{ij}'}$$

weighted average

Self-Attention Basics



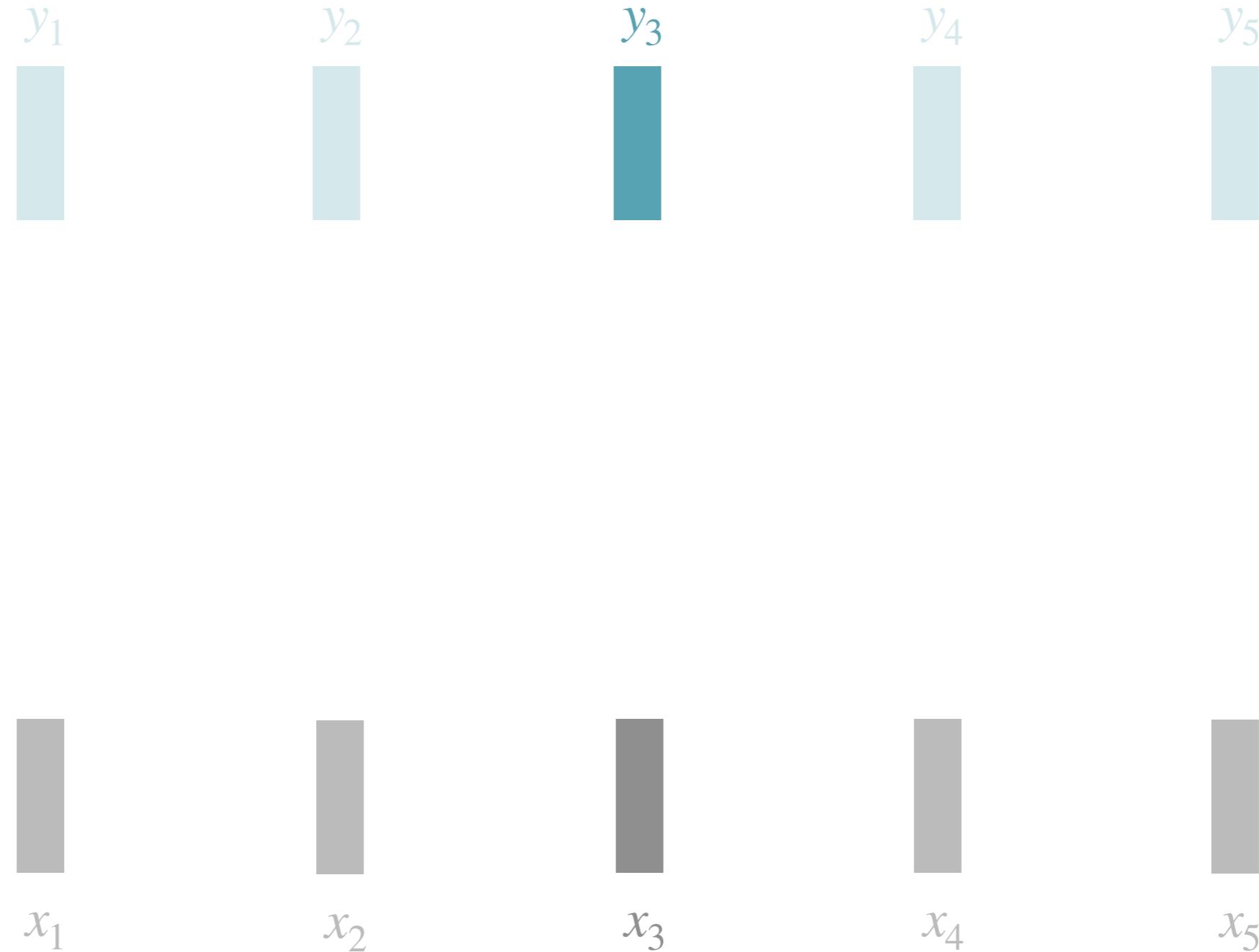
Self-Attention Basics

$$y_i = \sum_j w_{ij} x_j$$

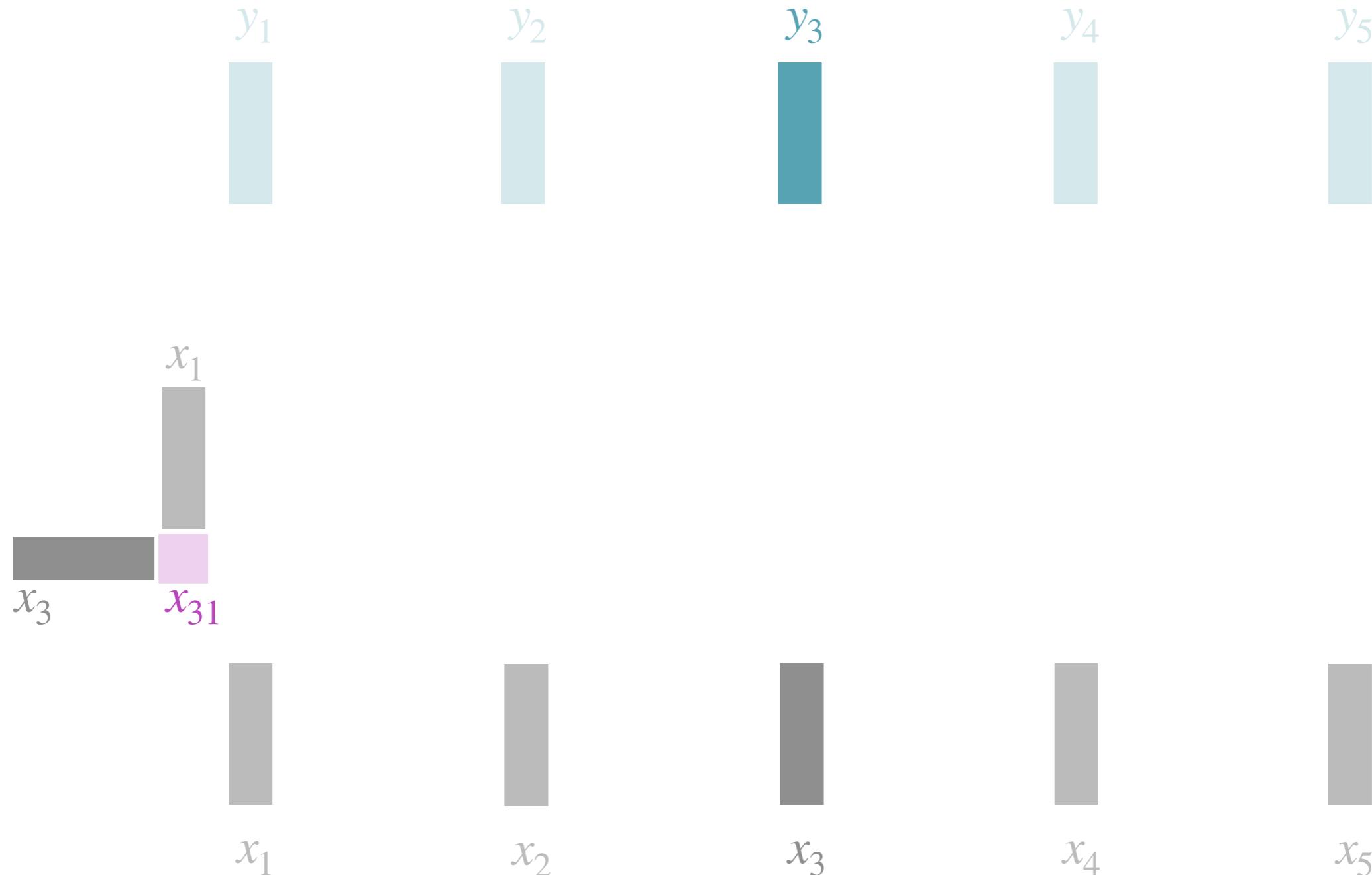
$$w'_{ij} = \mathbf{x}_i^\top \mathbf{x}_j$$

$$w_{ij} = \frac{\exp w'_{ij}}{\sum_j \exp w'_{ij}}$$

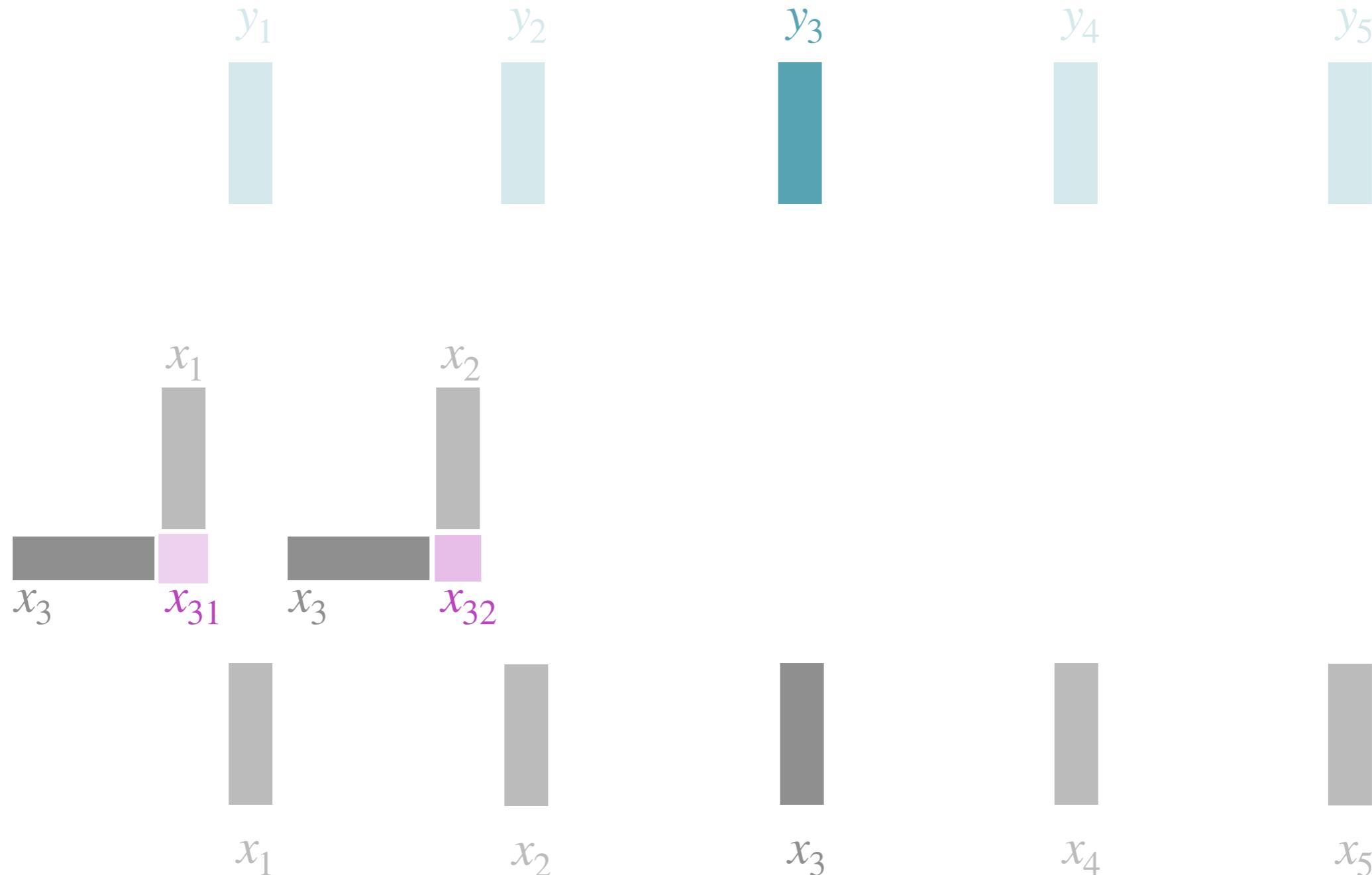
Self-Attention Basics



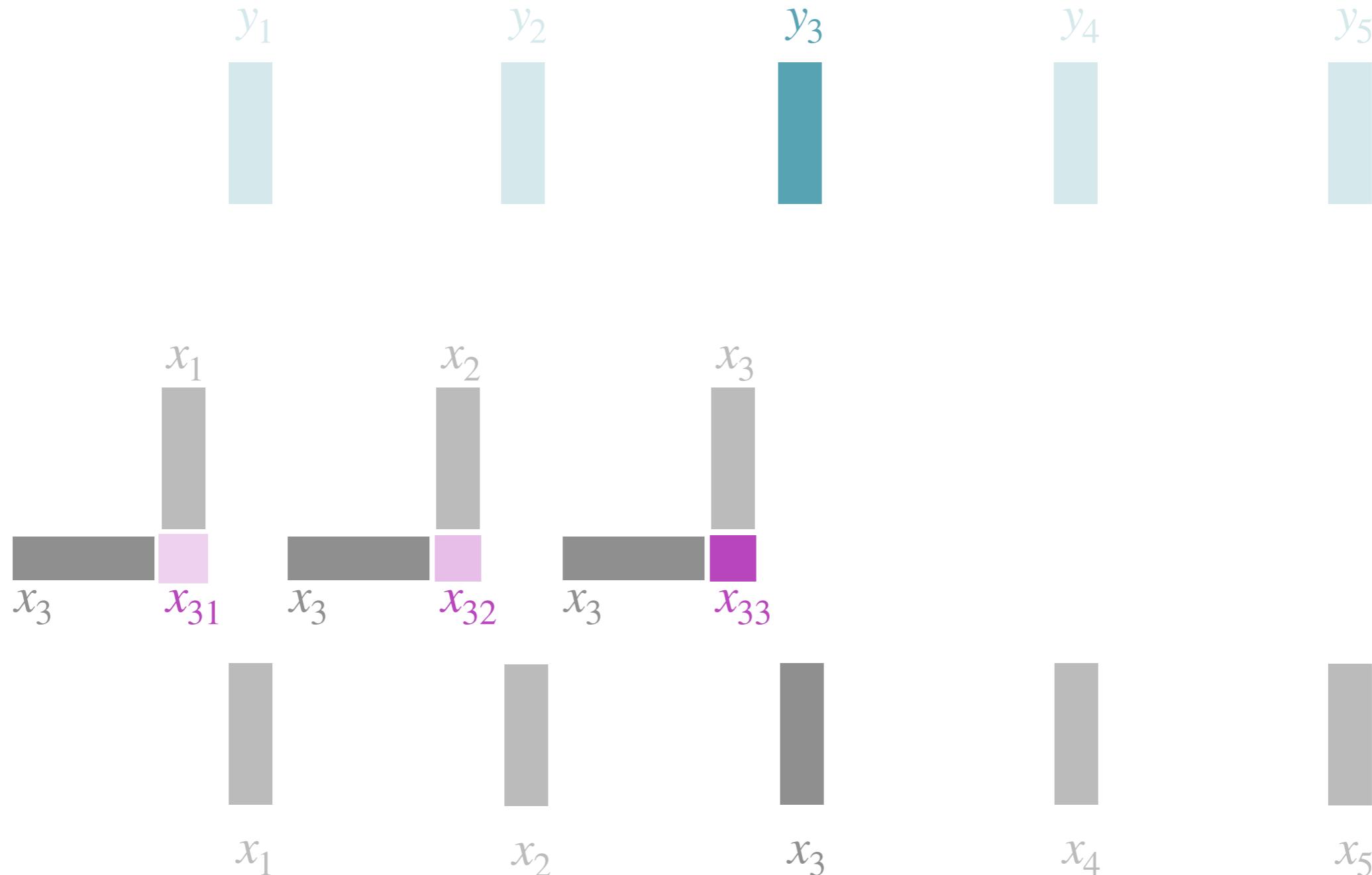
Self-Attention Basics



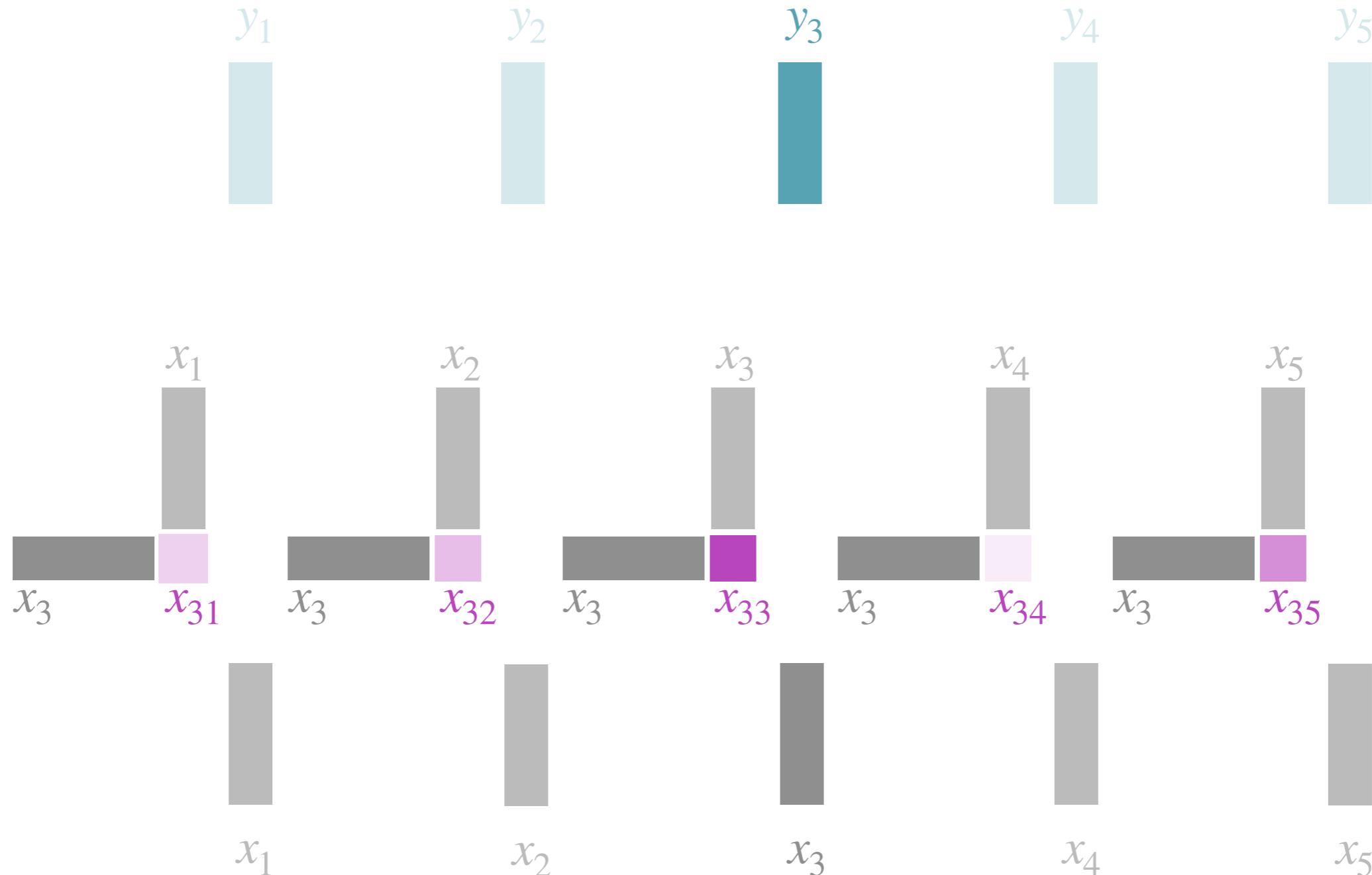
Self-Attention Basics



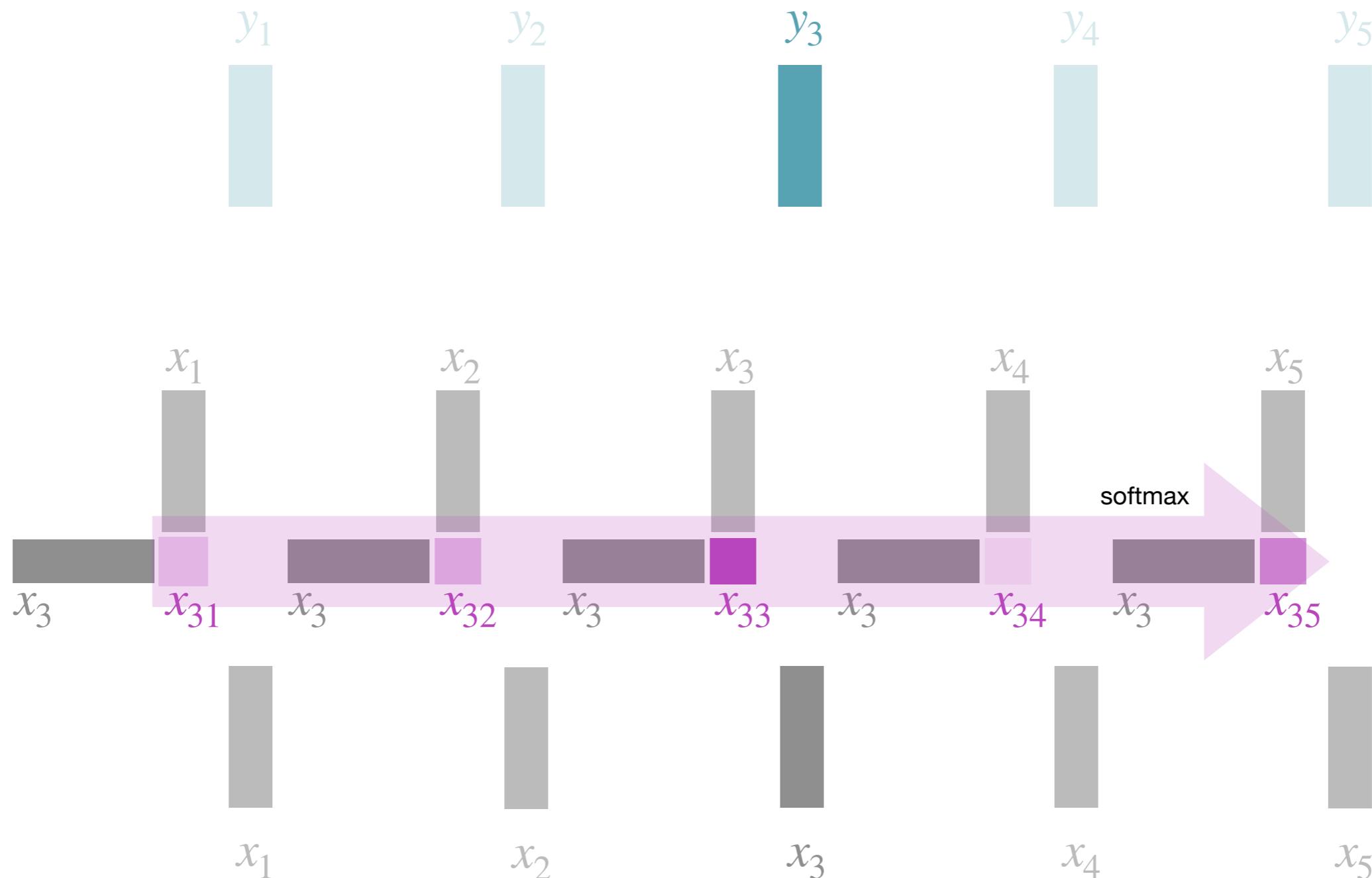
Self-Attention Basics



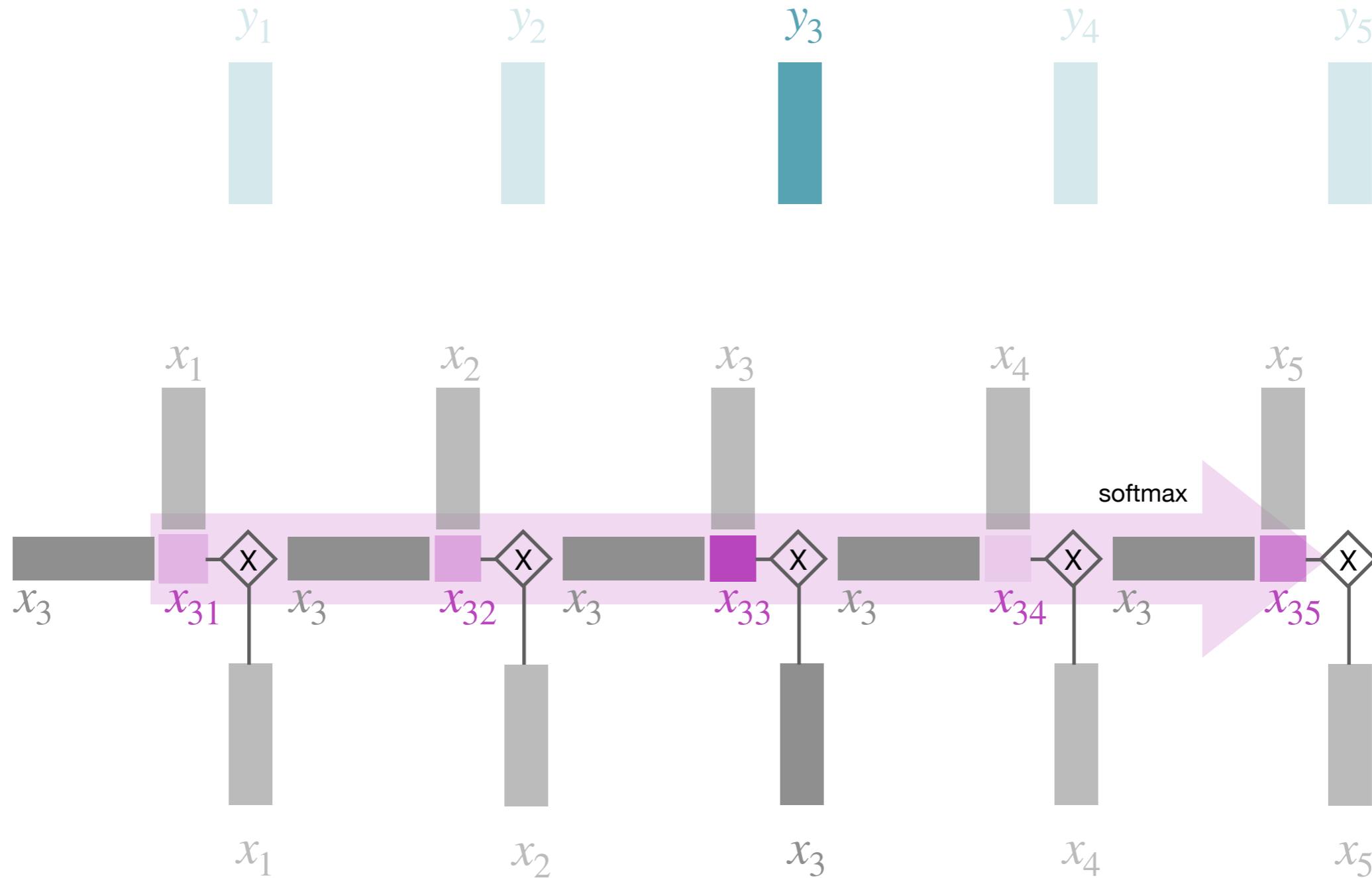
Self-Attention Basics



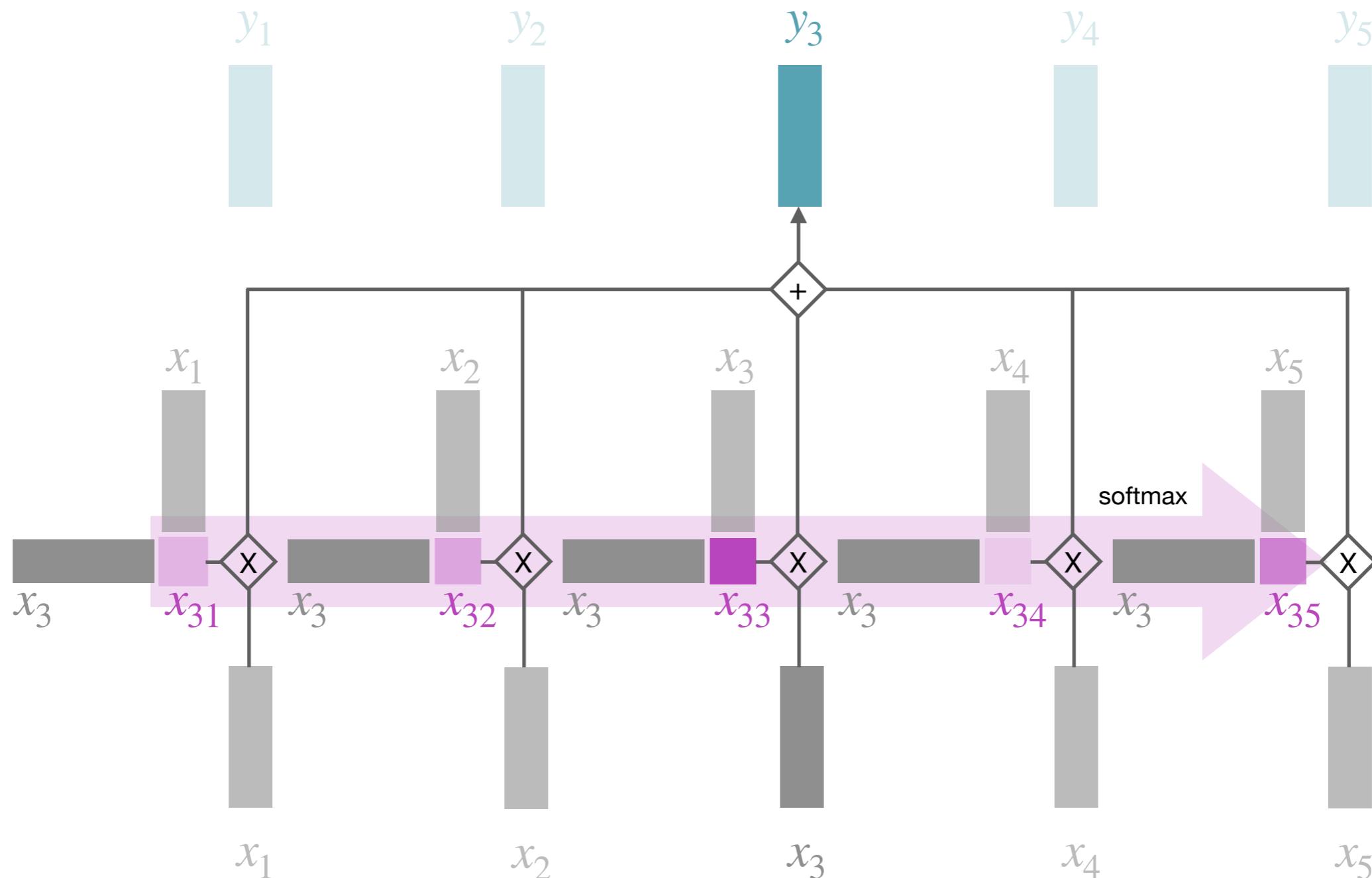
Self-Attention Basics



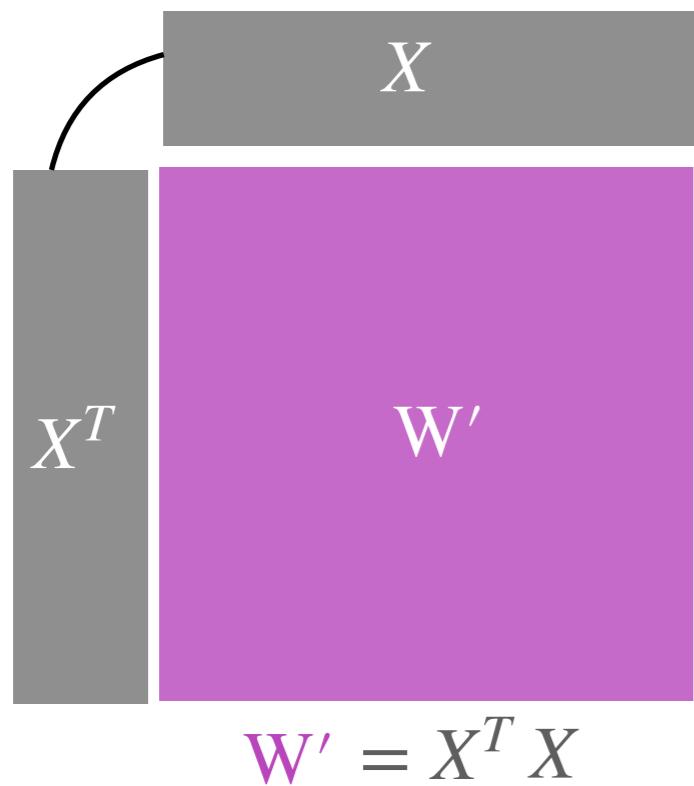
Self-Attention Basics



Self-Attention Basics

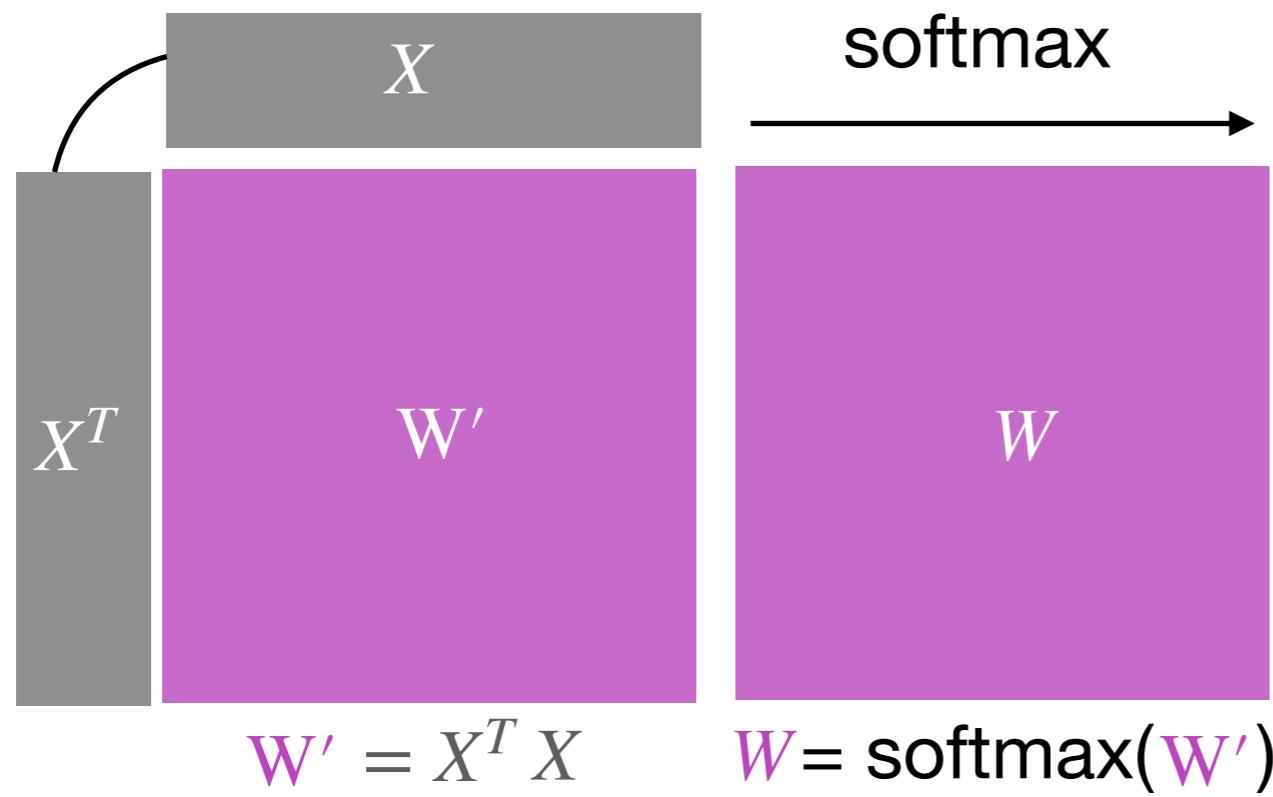


Self-Attention Basics



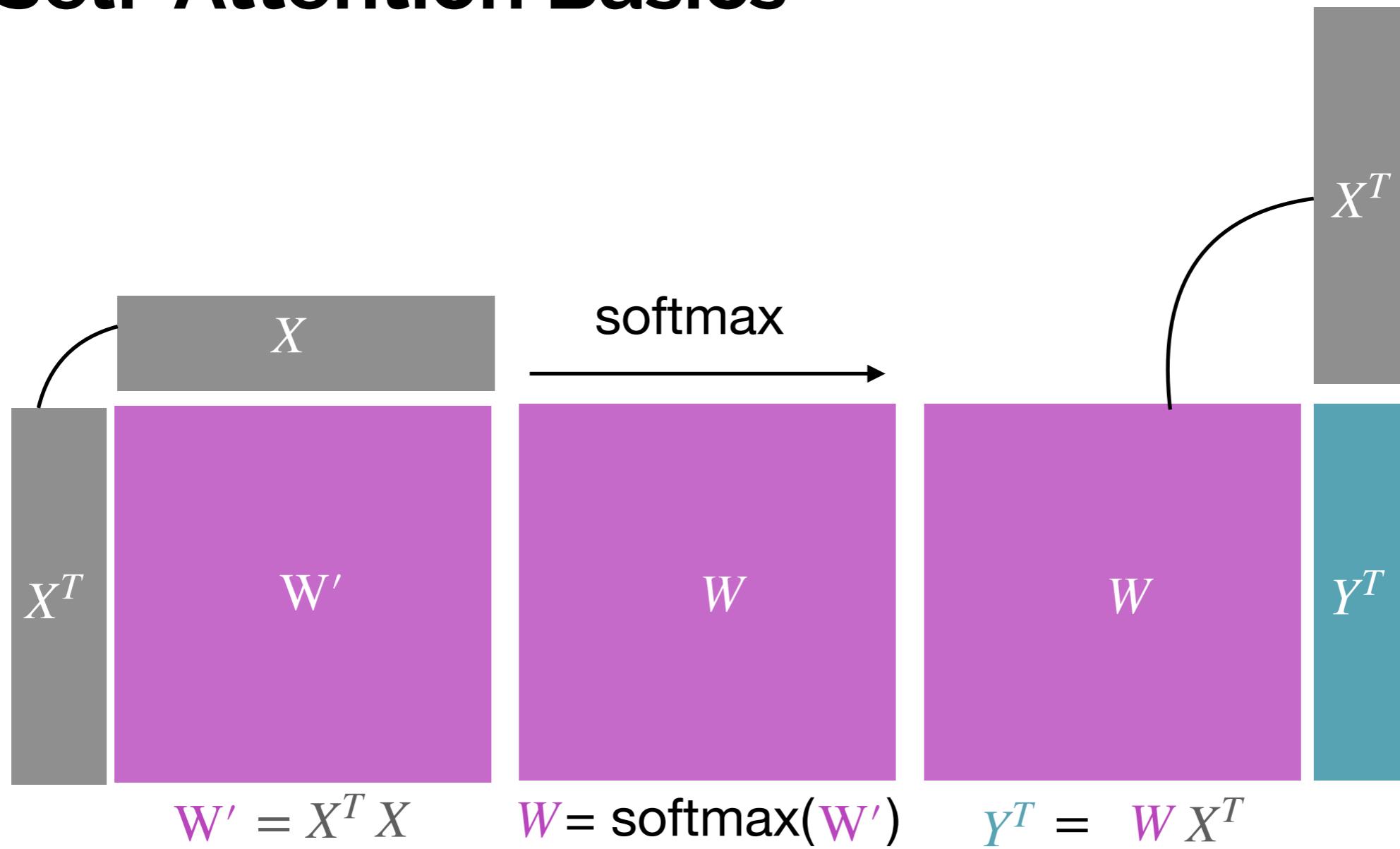
Vectorized version

Self-Attention Basics



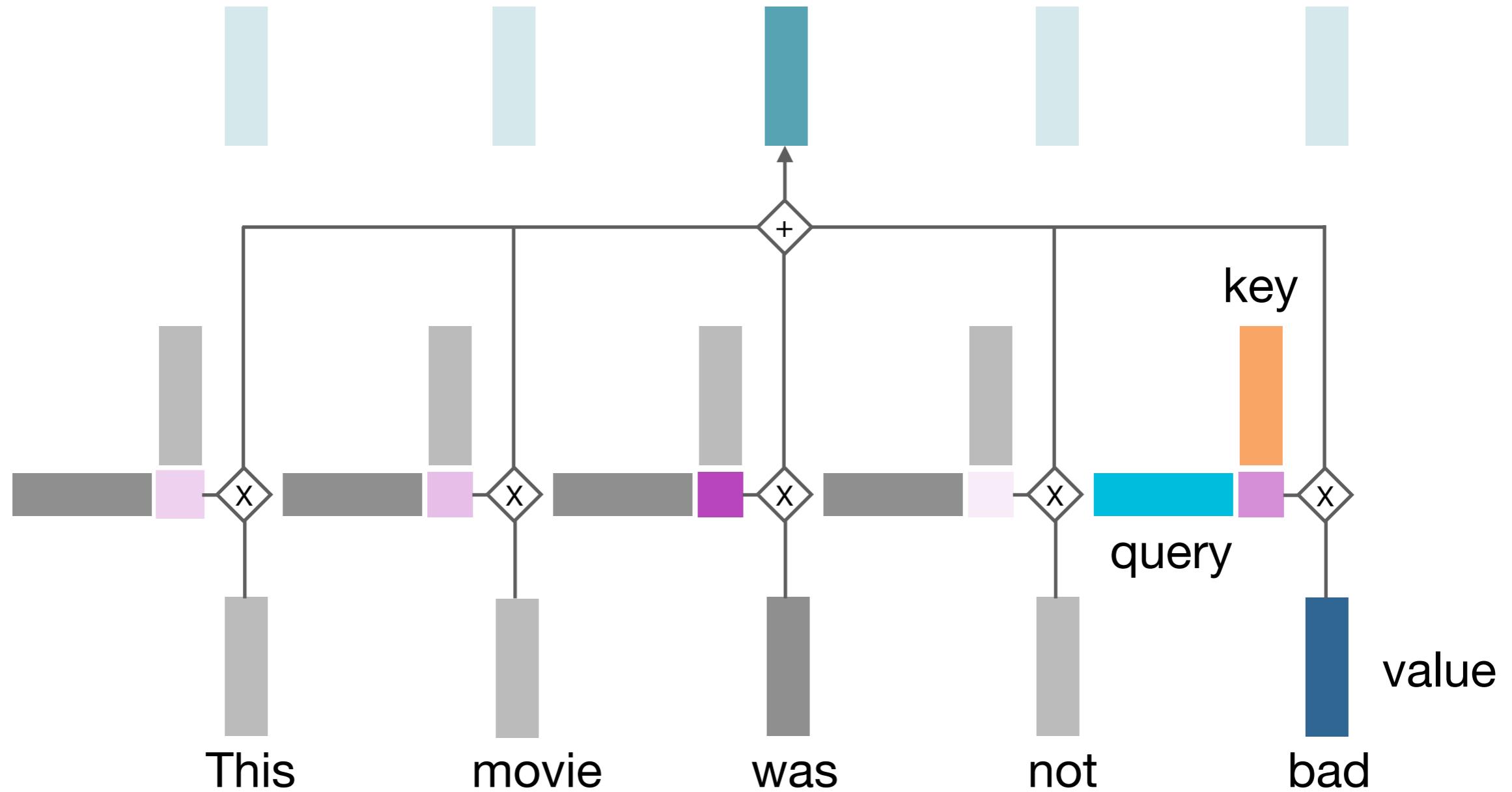
Vectorized version

Self-Attention Basics



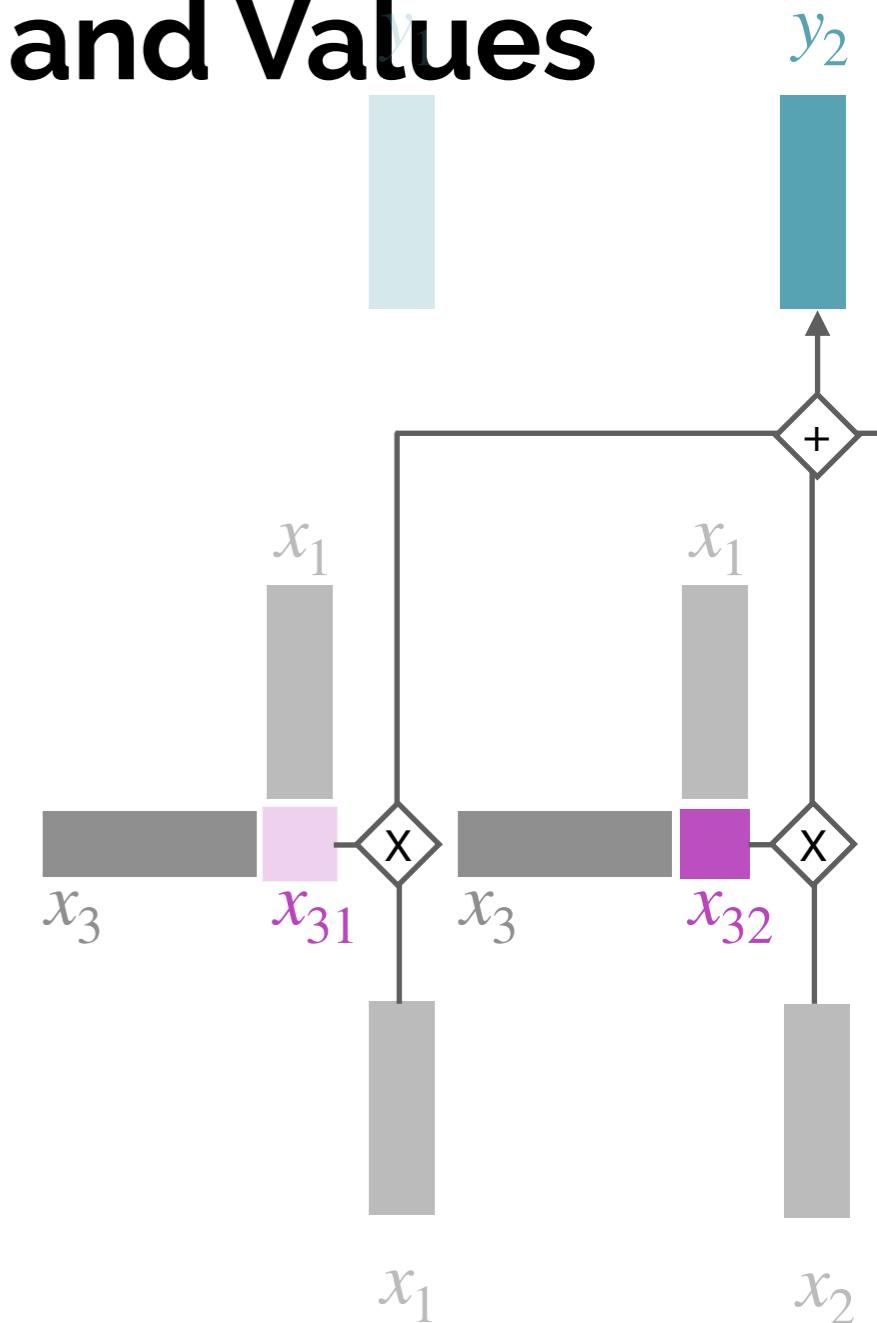
Vectorized version

Self-Attention: Keys, Queries and Values



Self-Attention: Keys, Queries and Values

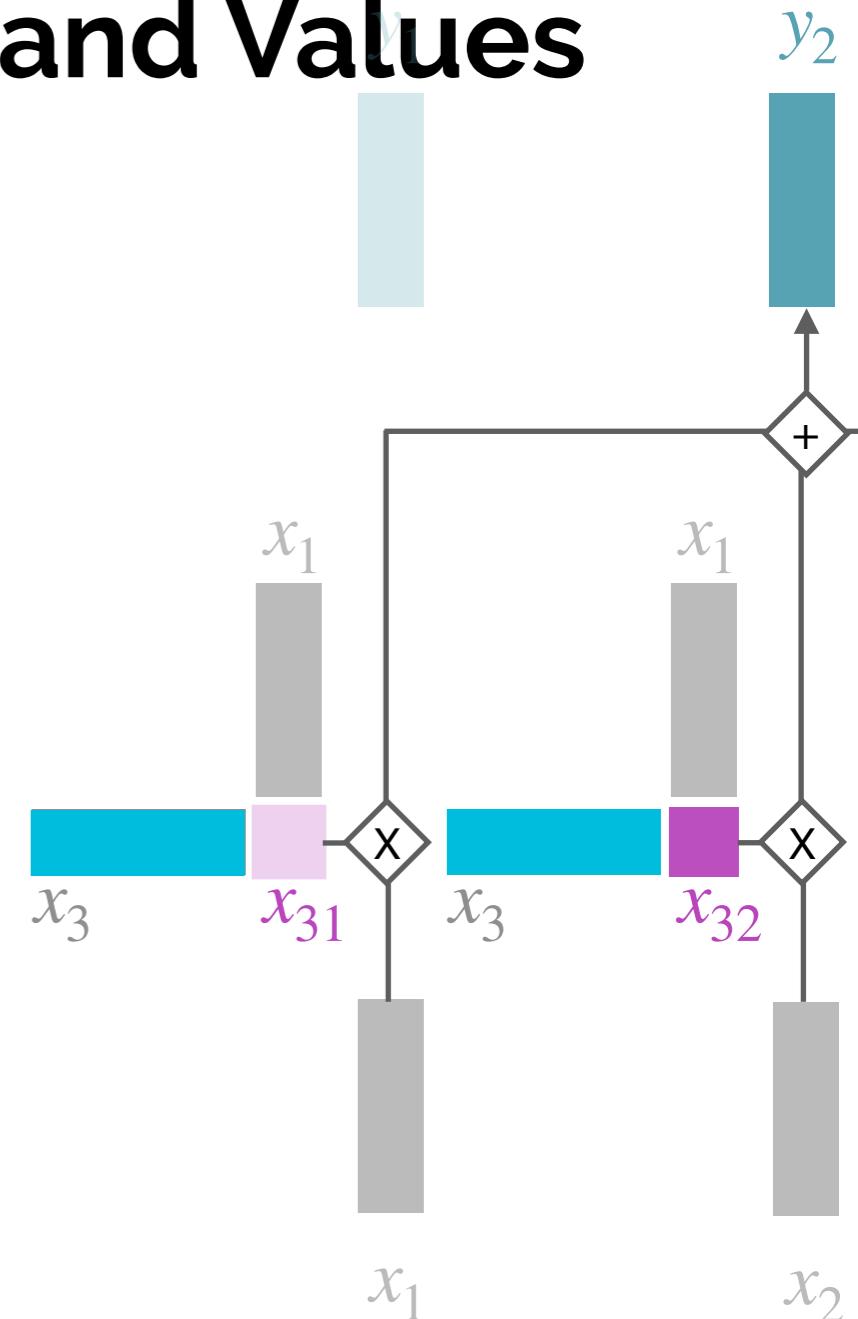
Every input vector x_i is used in 3 ways:



Self-Attention: Keys, Queries and Values

Every input vector x_i is used in 3 ways:

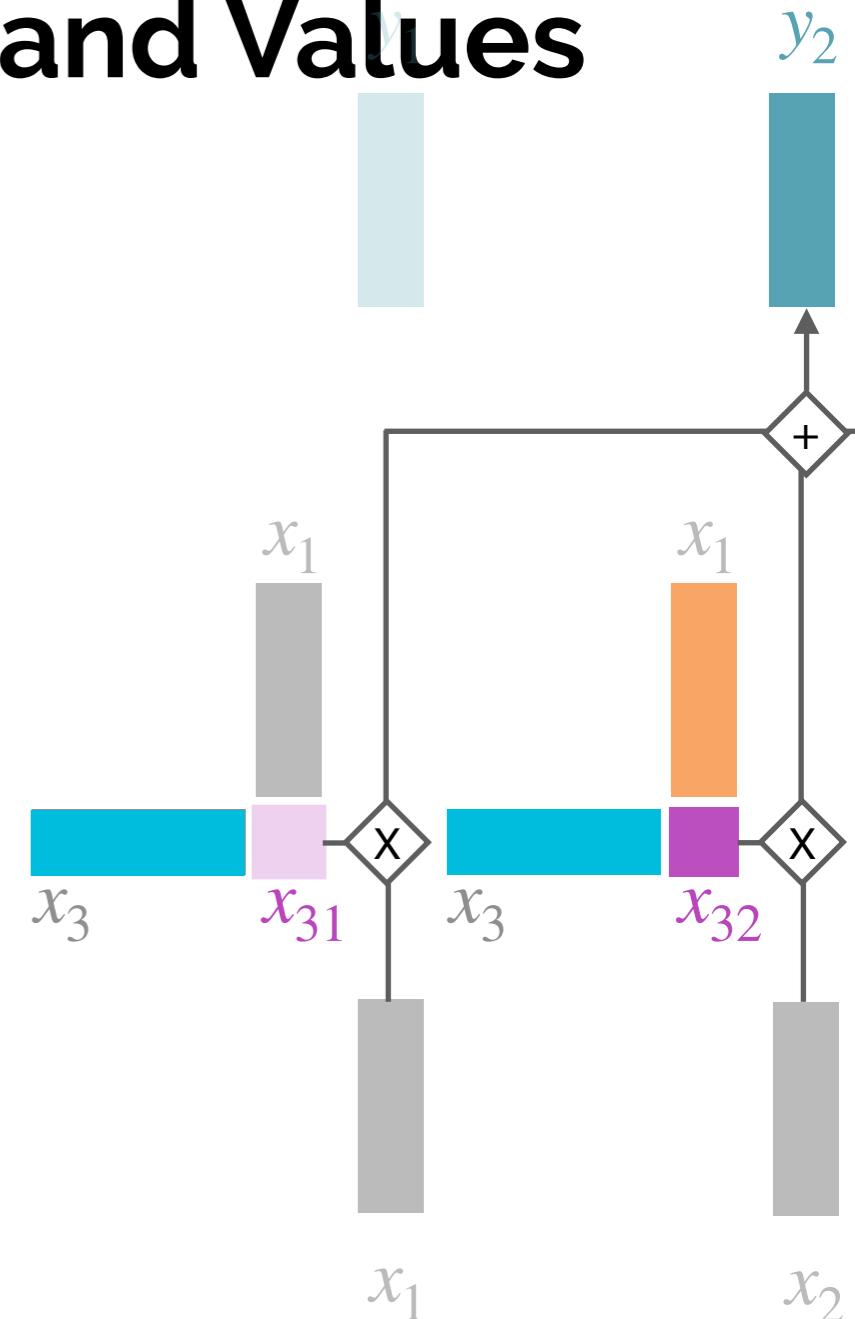
- Compared to every other vector to compute attention weights for its own output y_i (query)



Self-Attention: Keys, Queries and Values

Every input vector x_i is used in 3 ways:

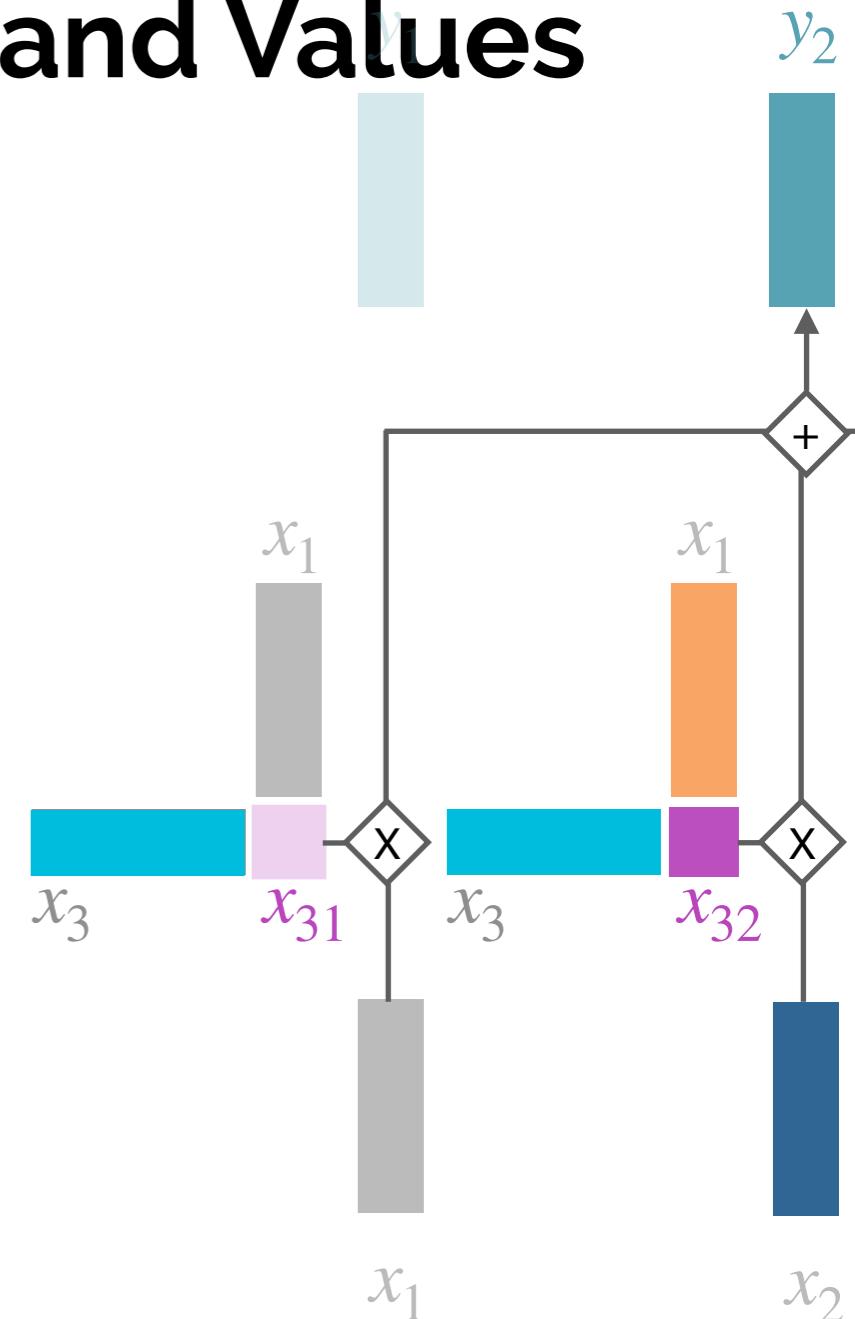
- Compared to every other vector to compute attention weights for its own output y_i (query)
- Compared to every other vector to compute attention weight w_{ij} for output y_j (key)



Self-Attention: Keys, Queries and Values

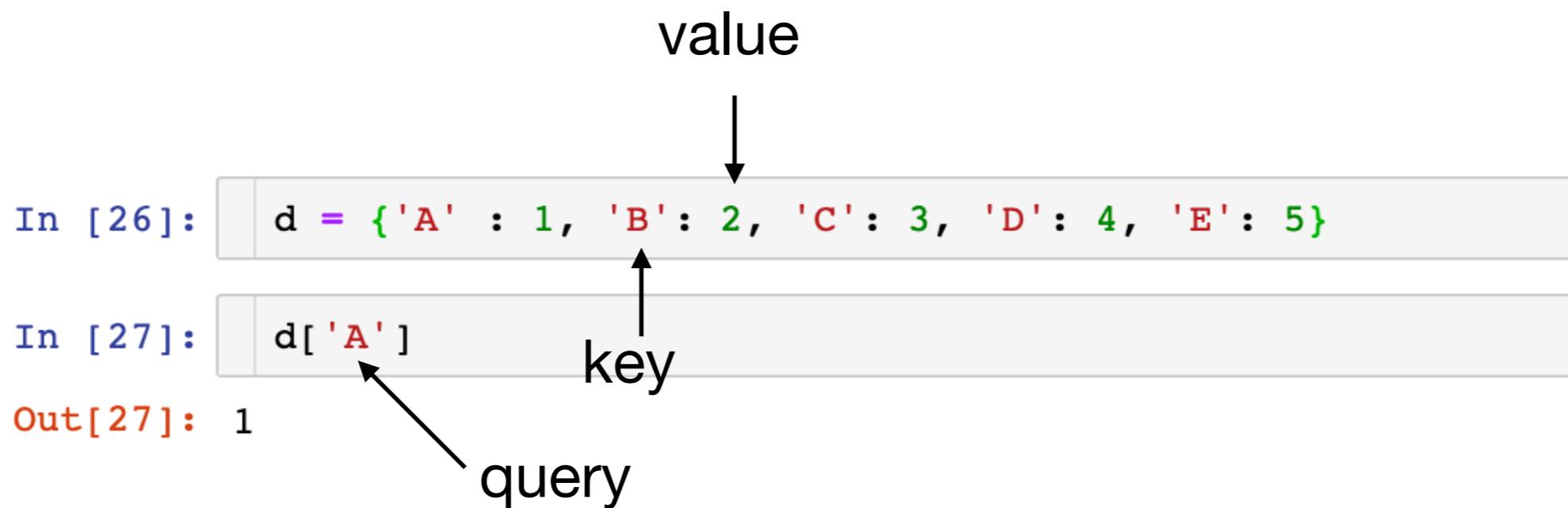
Every input vector x_i is used in 3 ways:

- Compared to every other vector to compute attention weights for its own output y_i (query)
- Compared to every other vector to compute attention weight w_{ij} for output y_j (key)
- Summed with other vectors to form the result of the attention weighted sum (value)



Self-Attention: Keys, Queries and Values

Intuition:



Self-Attention: Keys, Queries and Values

We can process each input vector to fulfill
the three roles with matrix multiplication:

$$\mathbf{q}_i = \mathbf{W}_q \mathbf{x}_i \quad \mathbf{k}_i = \mathbf{W}_k \mathbf{x}_i \quad \mathbf{v}_i = \mathbf{W}_v \mathbf{x}_i$$

Learning the matrices

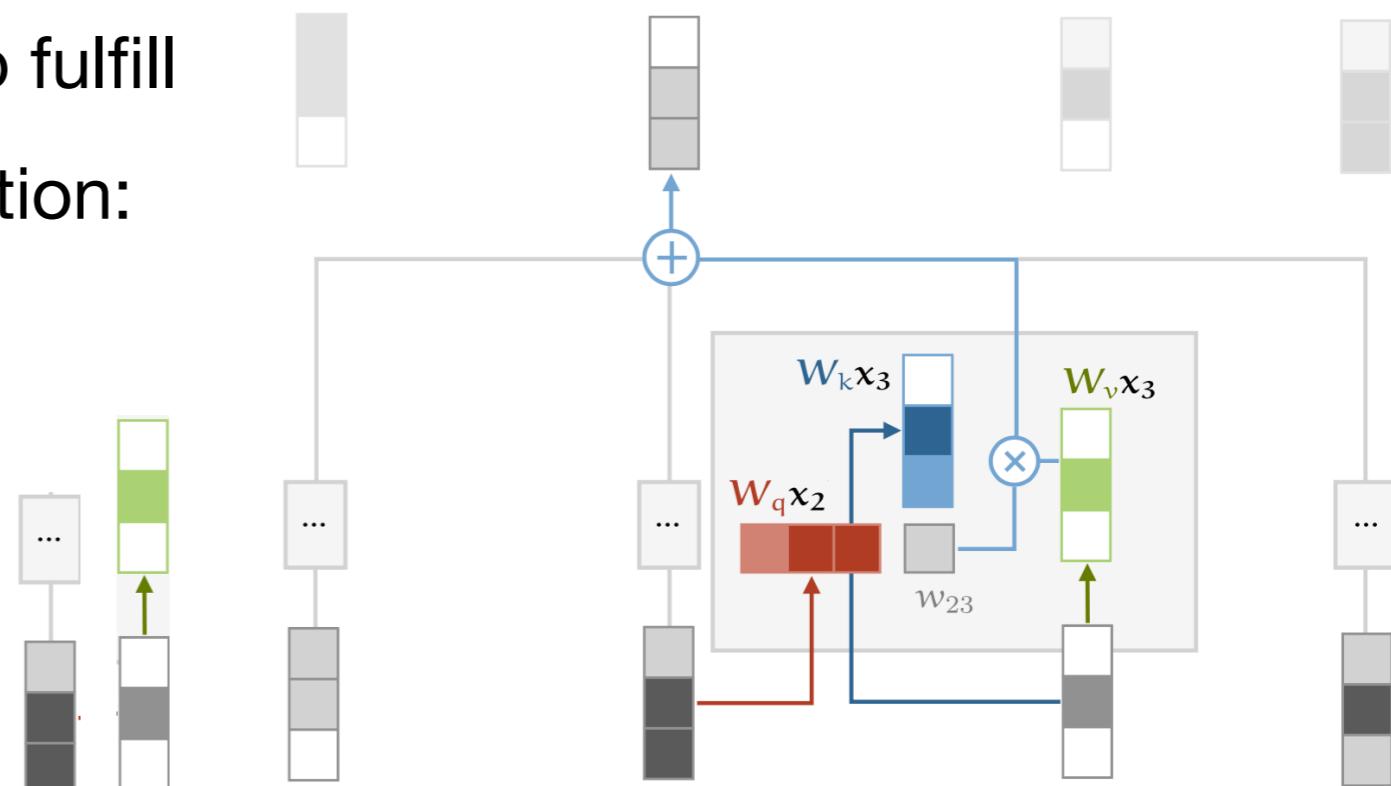


learning attention

$$w'_{ij} = \mathbf{q}_i^\top \mathbf{k}_j$$

$$w_{ij} = \text{softmax}(w'_{ij})$$

$$\mathbf{y}_i = \sum_j w_{ij} \mathbf{v}_j$$



Self-Attention: Scaled Dot-Product

$$w'_{ij} = \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\sqrt{d}}$$

↑
Input dimension

Self- and Multi-Head-Attention

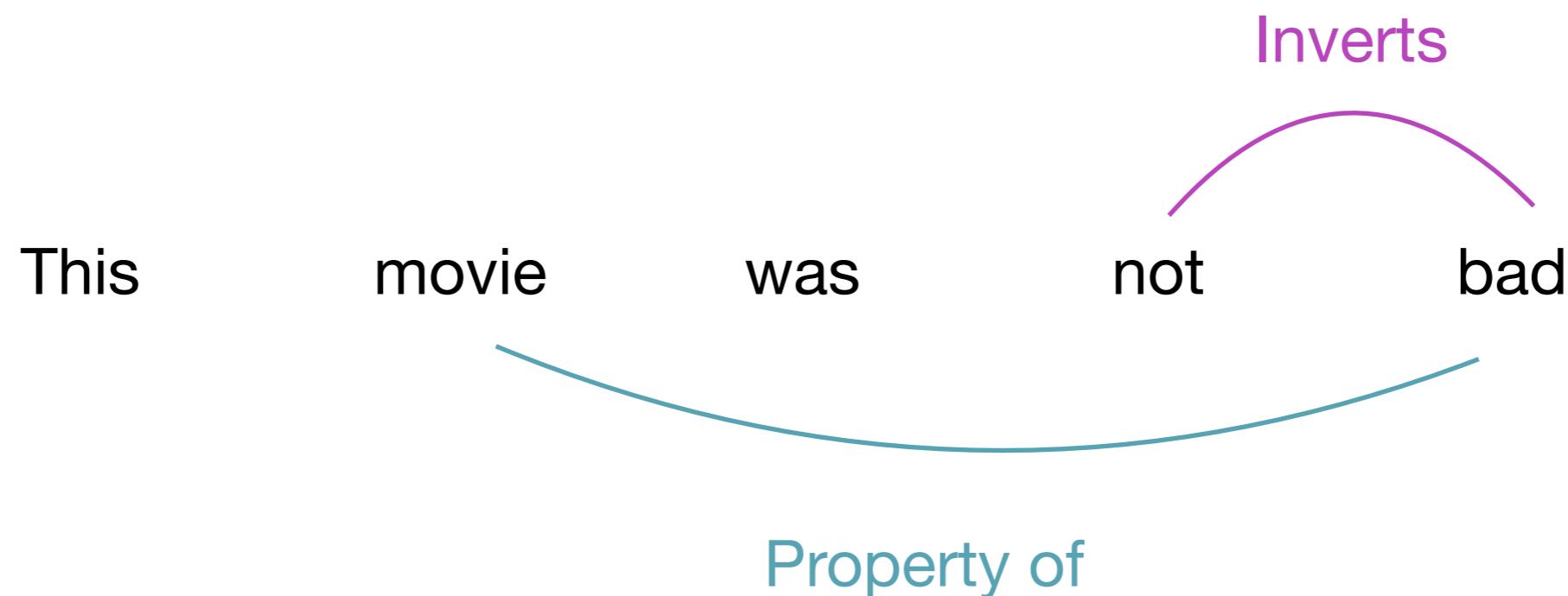
$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat} (\text{head}_1, \dots, \text{head}_h) \mathbf{W}^O$$

$$\text{where head } i = \text{Attention} \left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V \right)$$

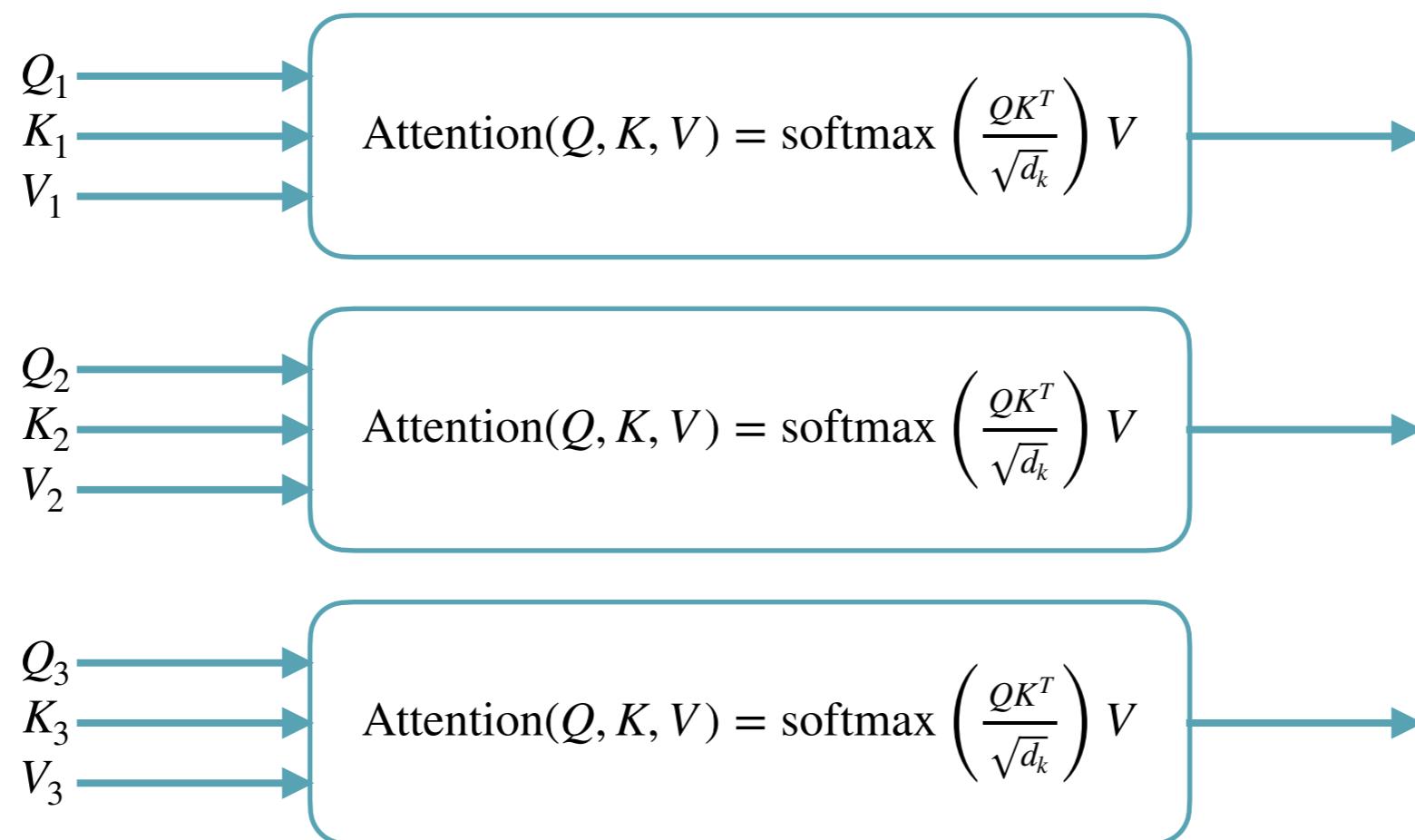
which results in:

$$\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in R^{d_{\text{model}} \times dk}$$

Self- and Multi-Head-Attention



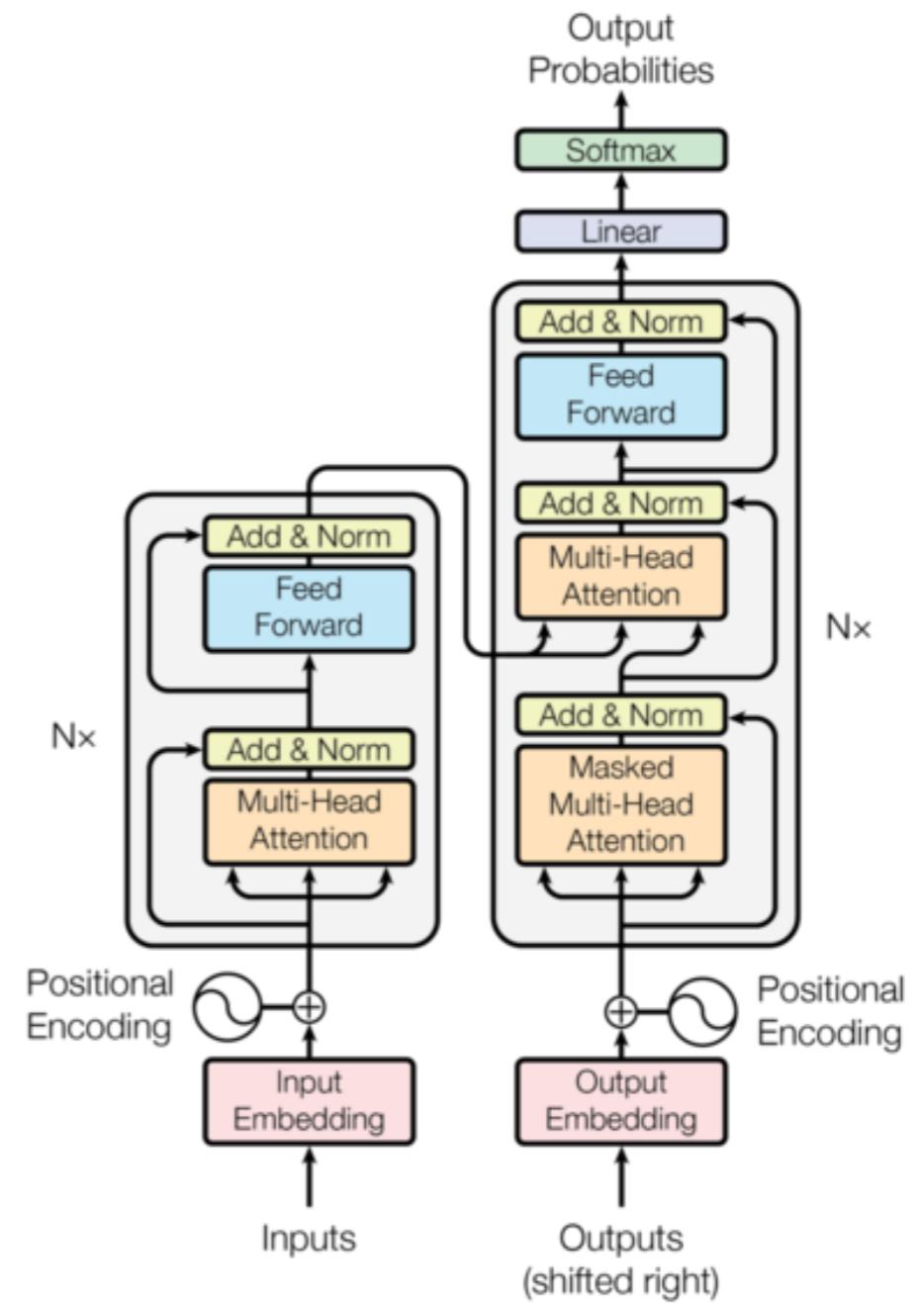
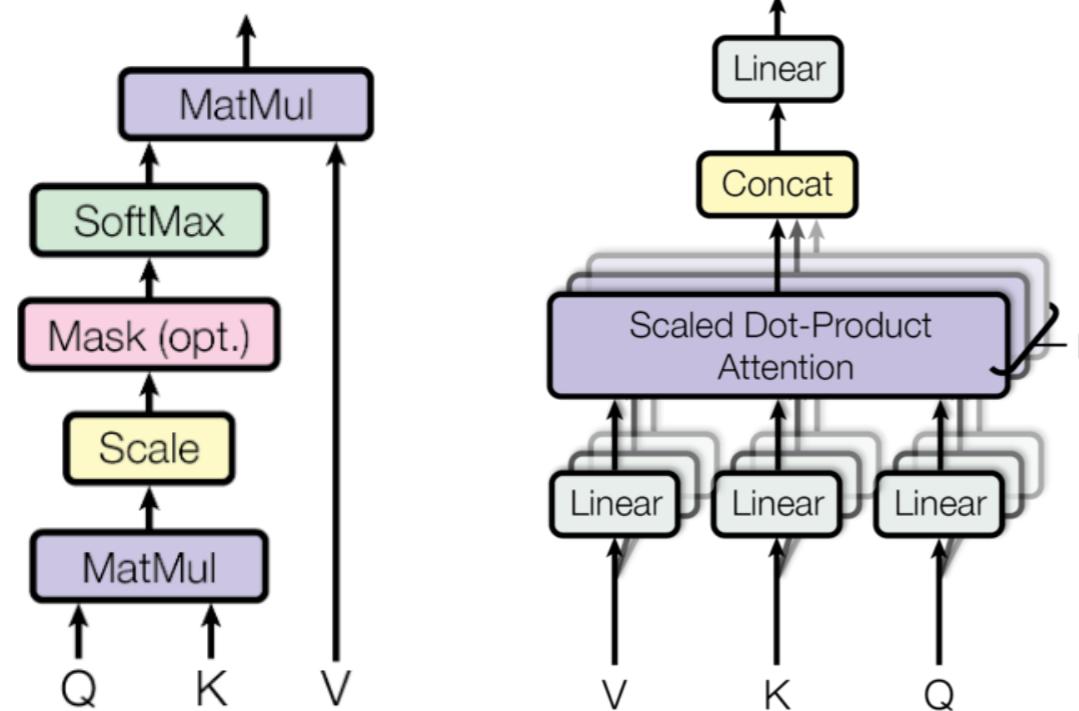
Self- and Multi-Head-Attention



Use Cases



Machine Translation



Vision Transformer (ViT)

Vision Transformer (ViT)

- An image is worth 16X16 words

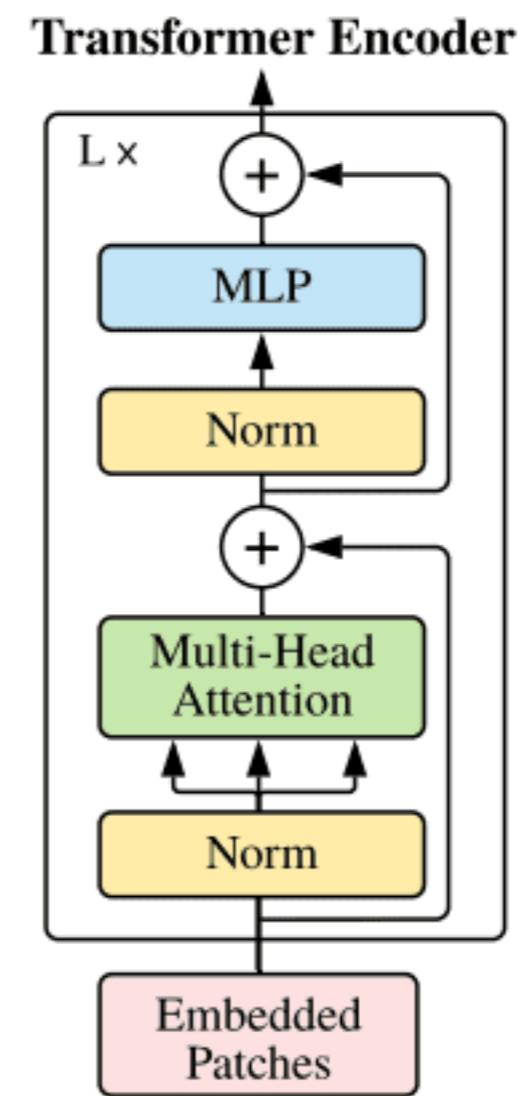
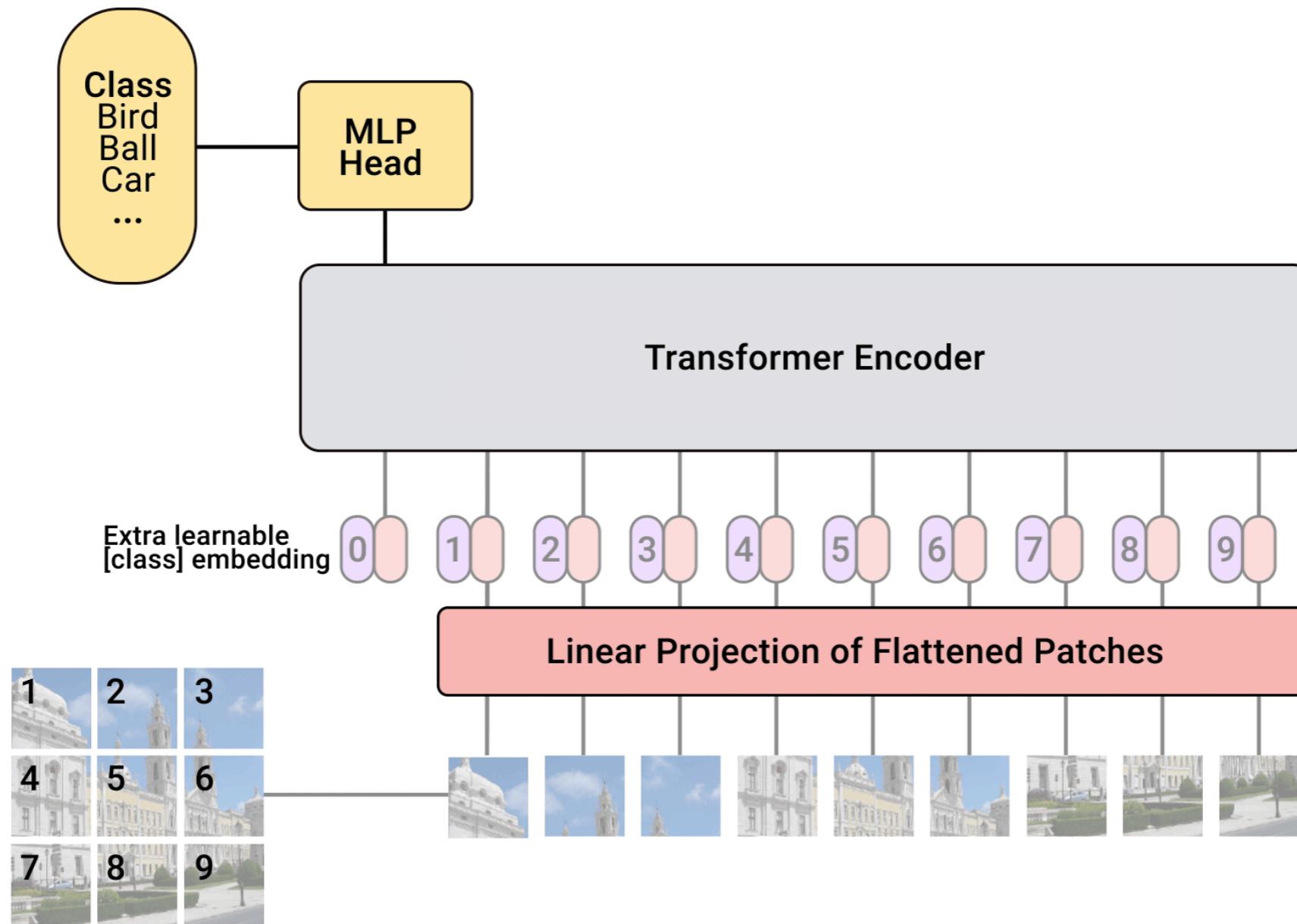
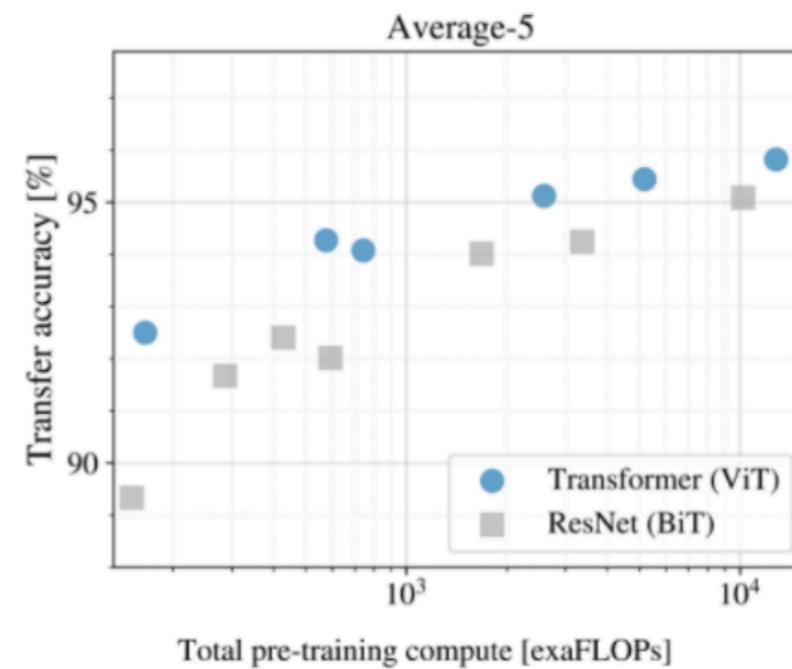
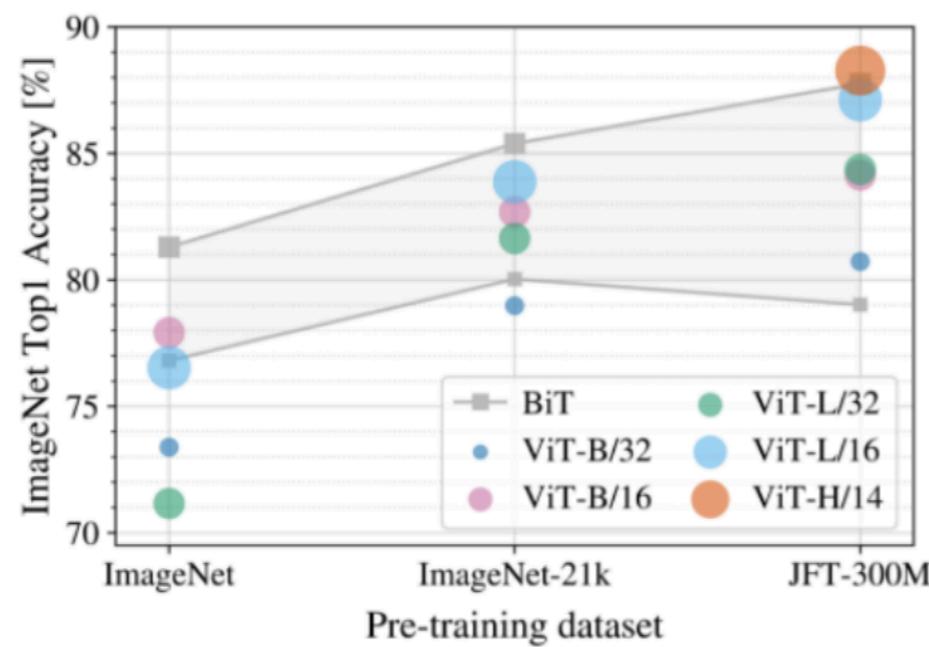


Image by Alexey Dosovitskiy et al 2020. Source: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

Vision Transformer (ViT)



Vision Transformer (ViT)



Time Series Anomaly Detection

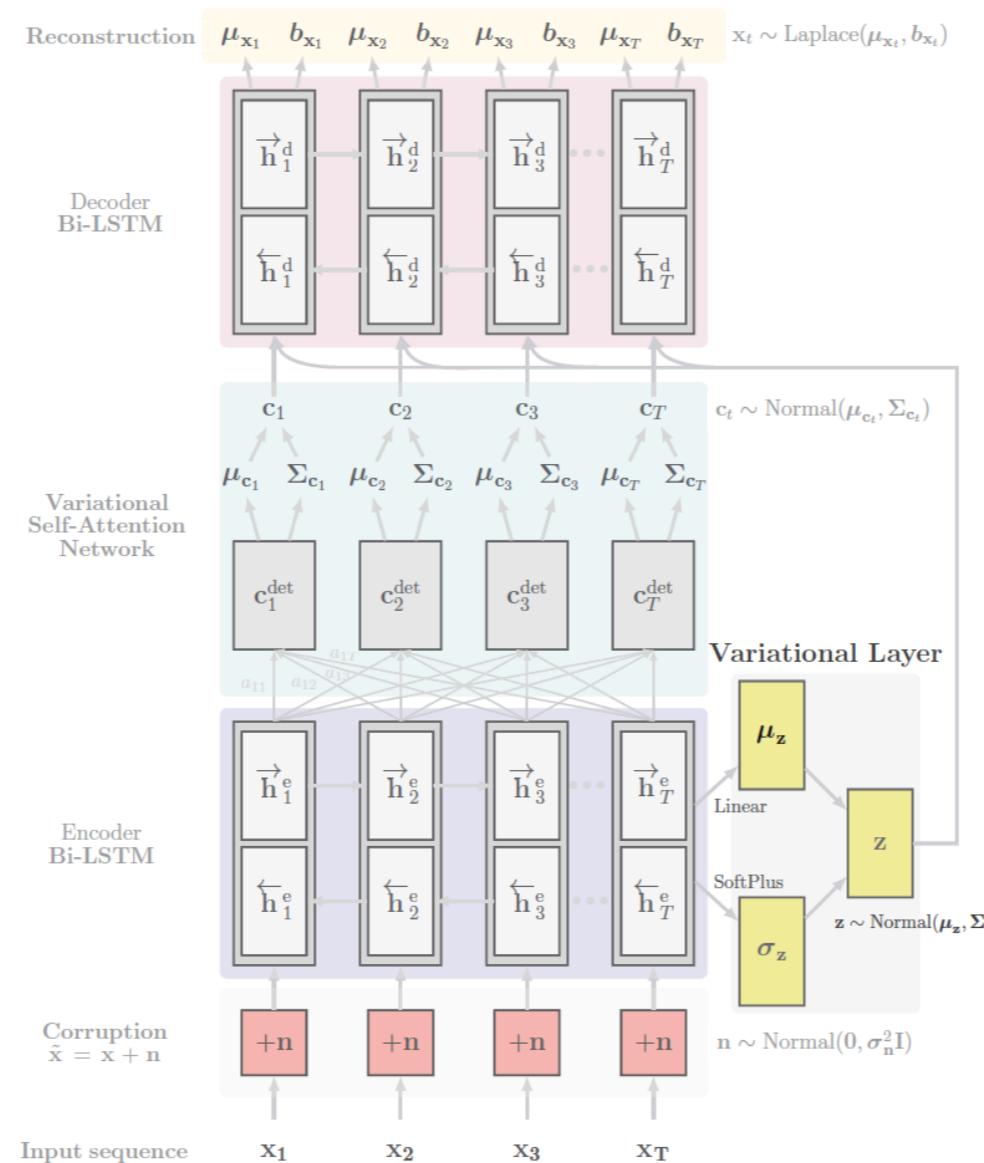


Image source: Unsupervised Anomaly Detection in Energy Time Series Data using Variational Recurrent Autoencoders with Attention

Time Series Anomalie Detection

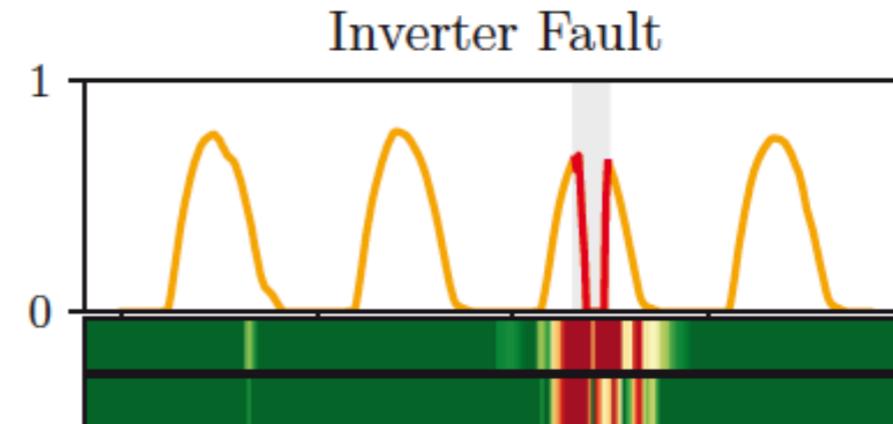
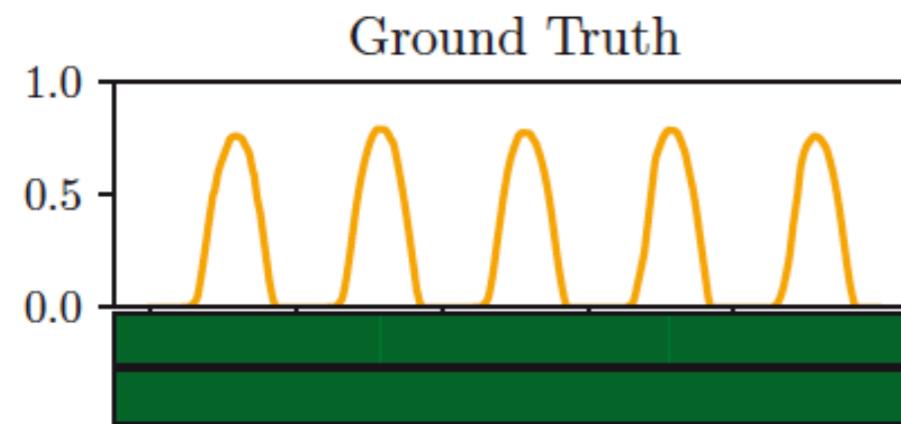


Image source: Unsupervised Anomaly Detection in Energy Time Series Data using Variational Recurrent Autoencoders with Attention



Time Series Anomaly Detection

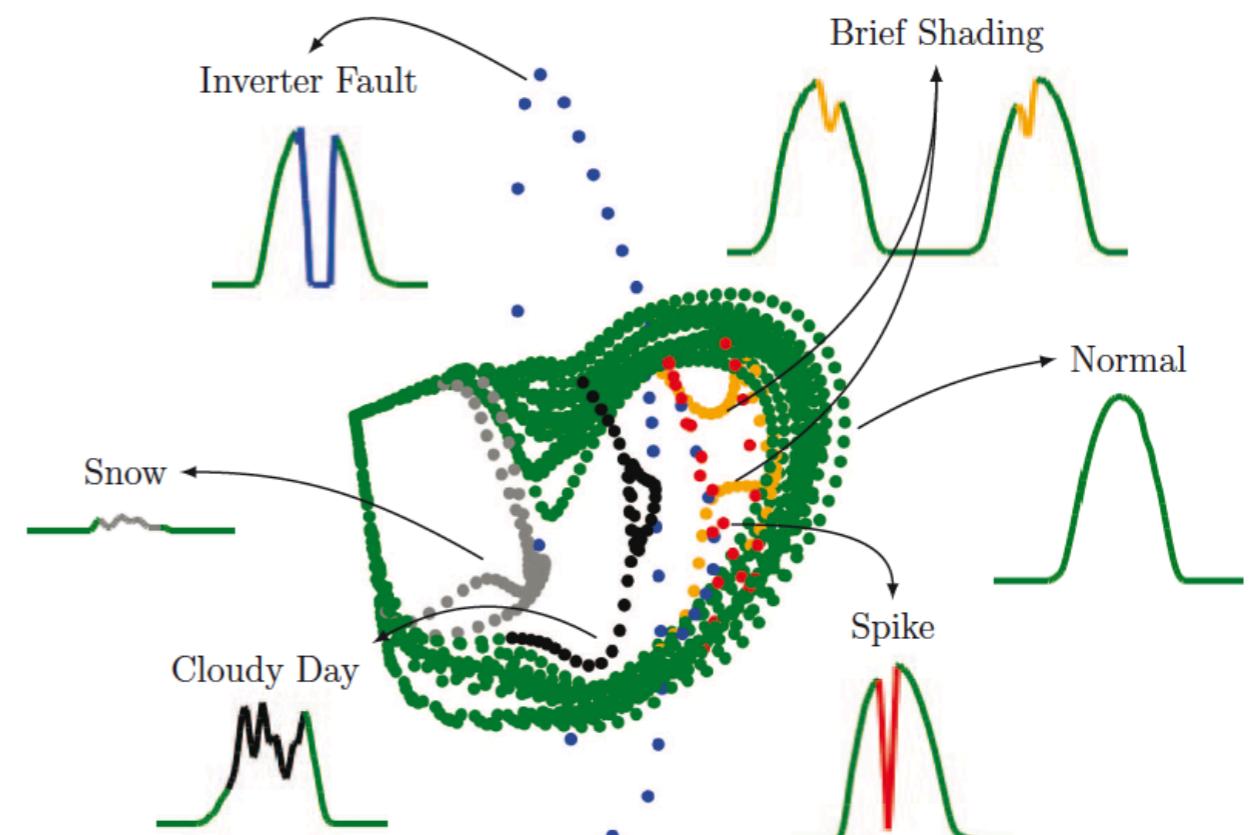
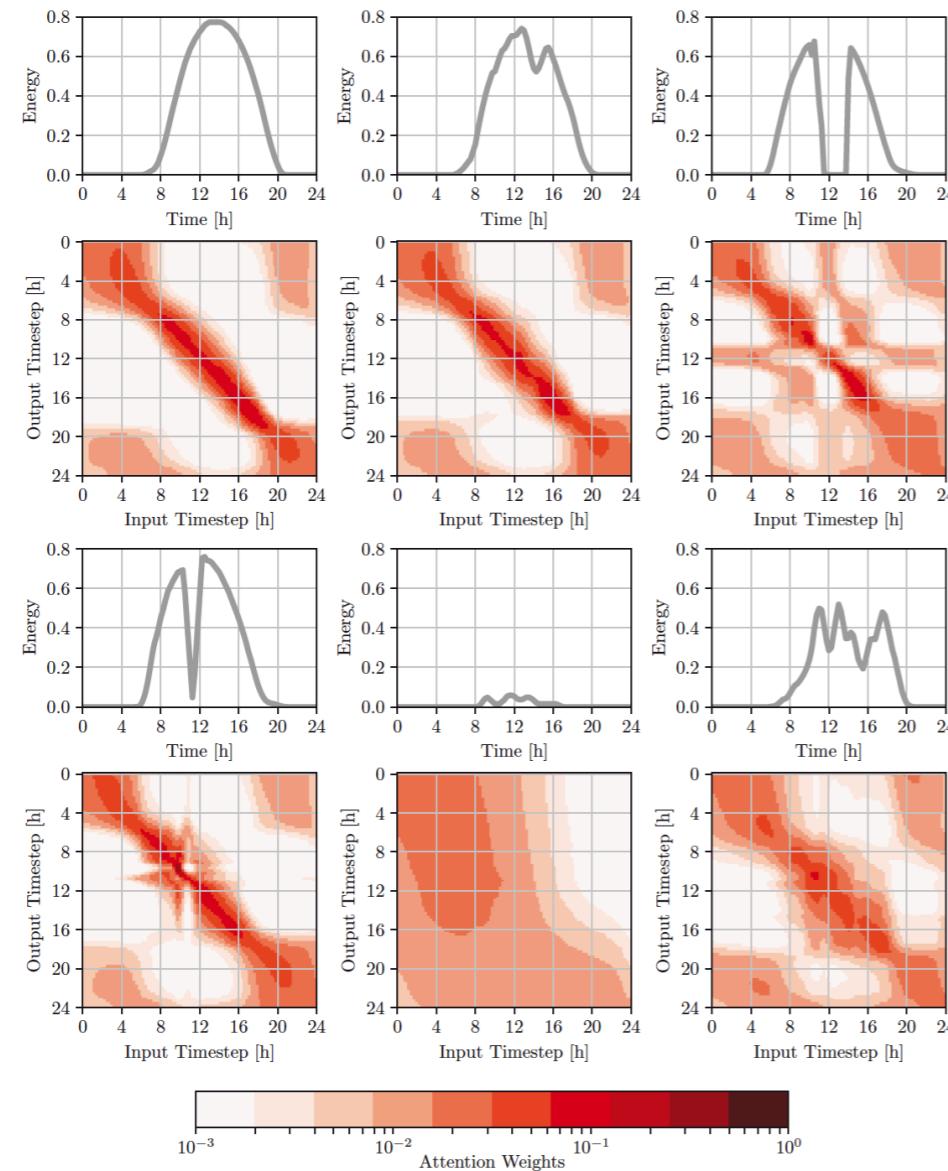
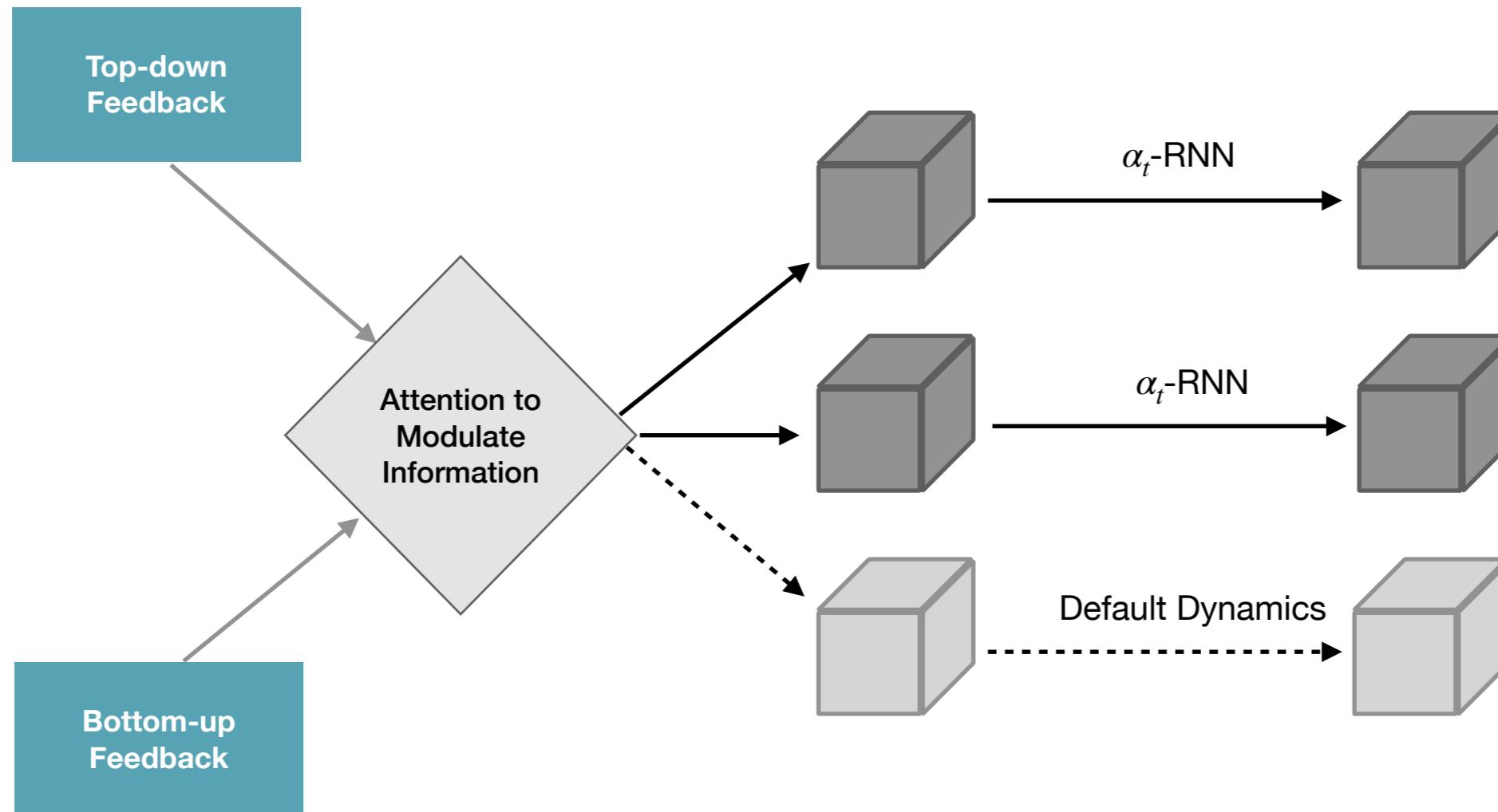
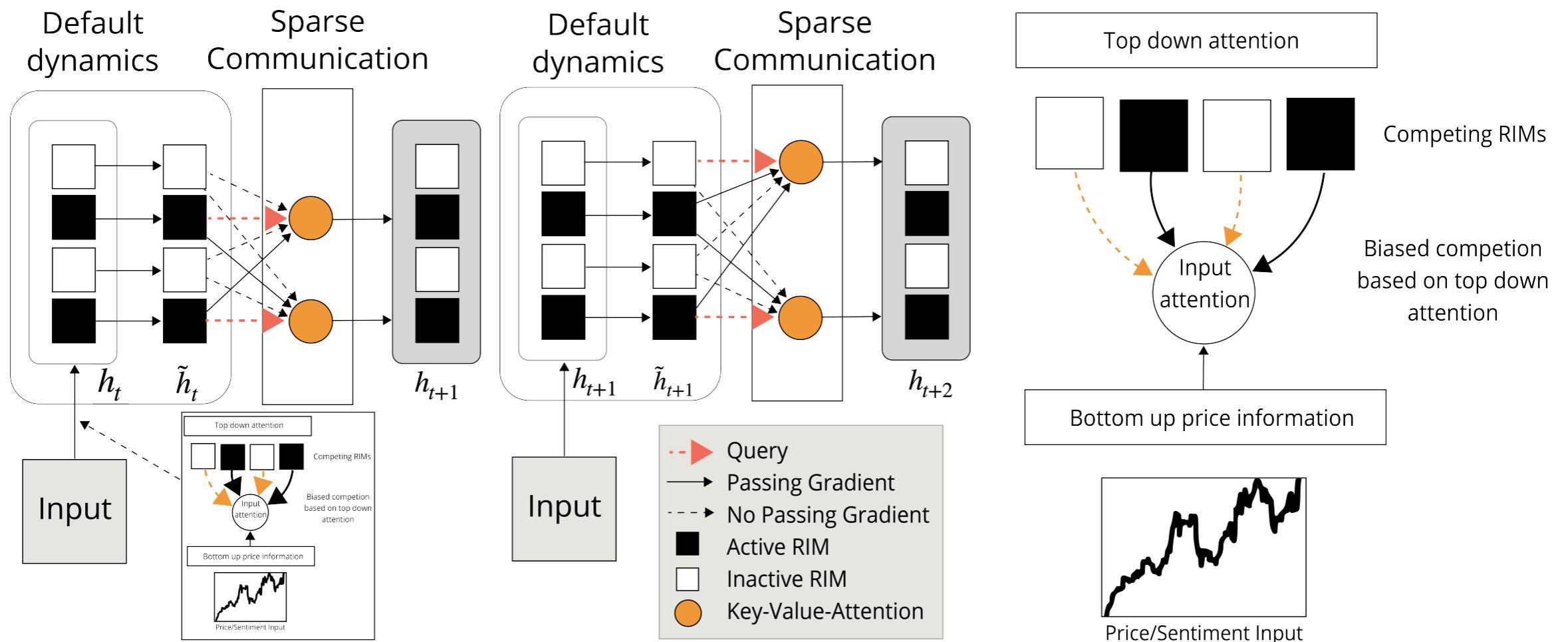


Image source: Unsupervised Anomaly Detection in Energy Time Series Data using Variational Recurrent Autoencoders with Attention

Stock Price Prediction



Stock Price Prediction



Stock Price Prediction

Thermo Fischer - ten lags input, five steps ahead prediction

Closing price

	RNN	LSTM	α_t -RIM
Lag	MAPE*	MAPE	MAPE
1	4.9015	11.0398	2.7522
2	3.4432	12.4441	2.9103
3	5.2922	10.1909	2.4824
4	3.5335	10.8379	2.5419
5	4.2818	8.9843	2.5887

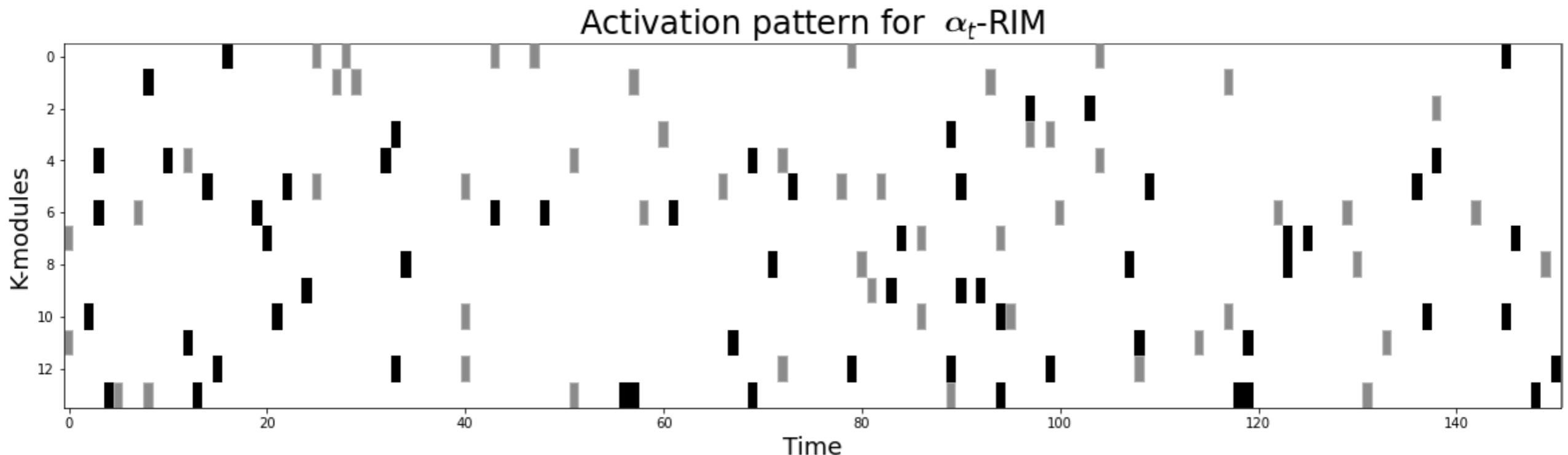
Closing price and Sentiment

	RNN	LSTM	α_t -RIM
Lag	MAPE	MAPE	MAPE
1	17.8271	18.6483	1.7189
2	13.5076	19.7892	1.9708
3	18.0445	20.2516	2.2388
4	11.2579	18.7996	2.4439
5	15.5029	17.4120	2.7314

* MAPE: Mean absolute percentage error

Stock Price Prediction

Attention can be used to enhance the explainability



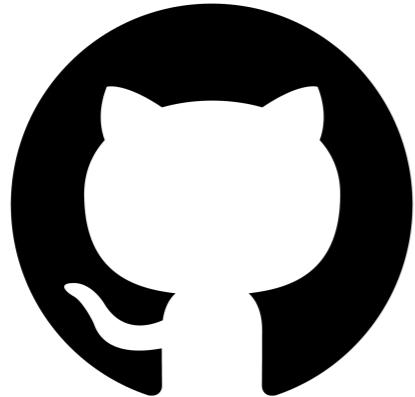
Activation pattern of the α_t -RIM, for the Brown Forman stock,
with only the closing price (black) and closing price and sentiment score (grey).



Sources Use Cases

- Attention Is All You Need
<https://arxiv.org/abs/1706.03762>
- An Image is Worth 16x16 Words:
Transformers for Image Recognition at Scale
<https://arxiv.org/abs/2010.11929>
- Unsupervised Anomaly Detection in Energy Time Series Data using Variational Recurrent Autoencoders with Attention
<https://ieeexplore.ieee.org/document/8614232>
- Dynamic and Context-Dependent Stock Price Prediction Using Attention Modules and News Sentiment
<https://arxiv.org/abs/2205.01639>

Sources



GitHub

https://github.com/Nicolepcx/women_in_tech_2022



<https://www.linkedin.com/in/nicole-koenigstein/>