INTRODUCTION TO DEEP LEARNING

PART IV – TRANSFORMERS, LLMS, AND OTHER INTERESTING ARCHITECTURES

Auliya Fitri, Sai Vemuri, Sreerag Naveenachandran

Machine Learning Group
Institute of Data Science

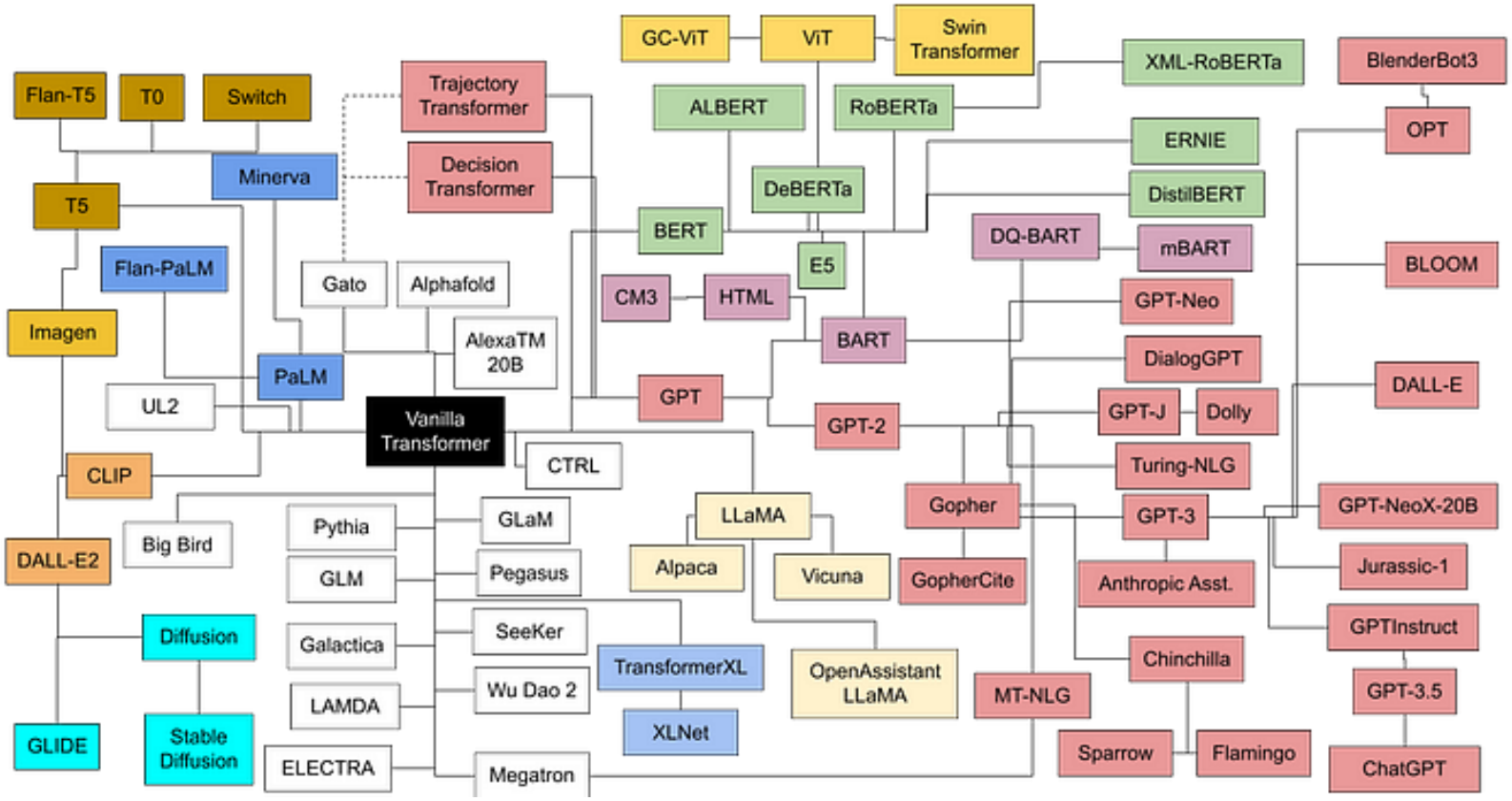
DLR

Schedule

Date	Time	Activity
13.11.2025 Day 1	09:00 - 10:00	Introduction and basics
	10:00 - 10:30	Hands-on I
	10:30 - 10:45	Coffee break
	10:45 - 11:45	Advanced concept and Convolutional Neural Network
	11:45 - 12:15	Hands-on II
	12:15 - 12:30	Recap Day 1
14.11.2025 Day 2	09:00 - 10:00	Deep Generative Models
	10:00 - 10:30	Hands-on III
	10:30 - 10:45	Coffee break
	10:45 - 11:45	Transformers, LLM, and other interesting architectures
	11:45 - 12:15	Hands-on IV
	12:15 - 12:30	Code and knowledge sources + closing

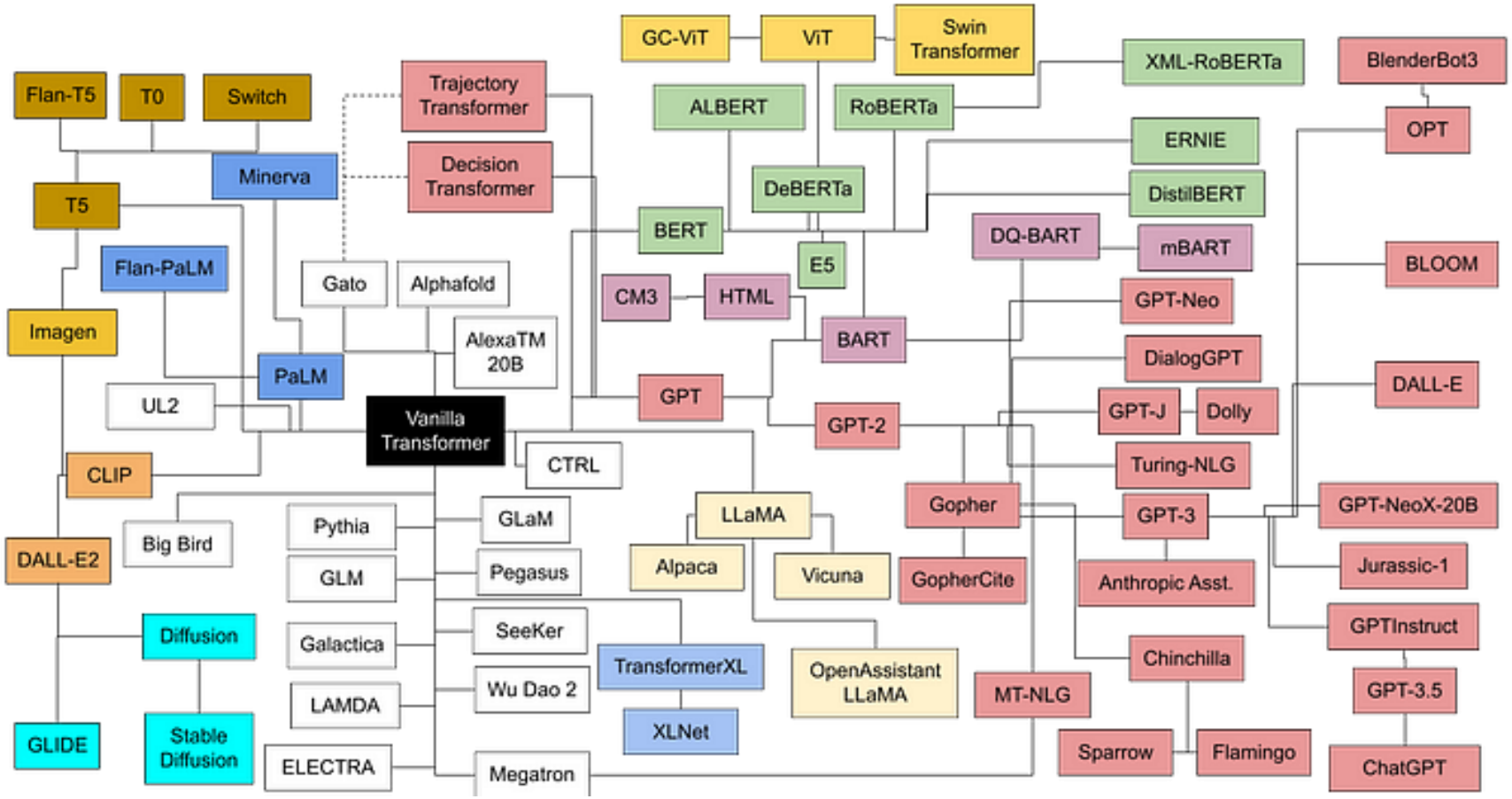
We're here!

The Evolution of Neural Network Architectures



https://medium.com/@jasminewu_yi/transformer-family-tree-69a7aa3c0d18

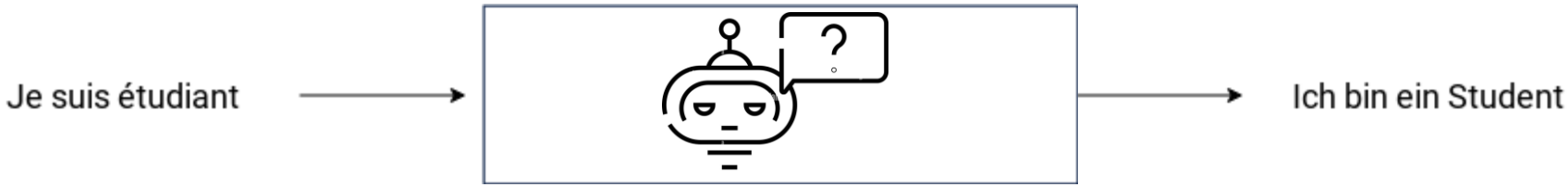
The Evolution of Neural Network Architectures



https://medium.com/@jasminewu_yi/transformer-family-tree-69a7aa3c0d18

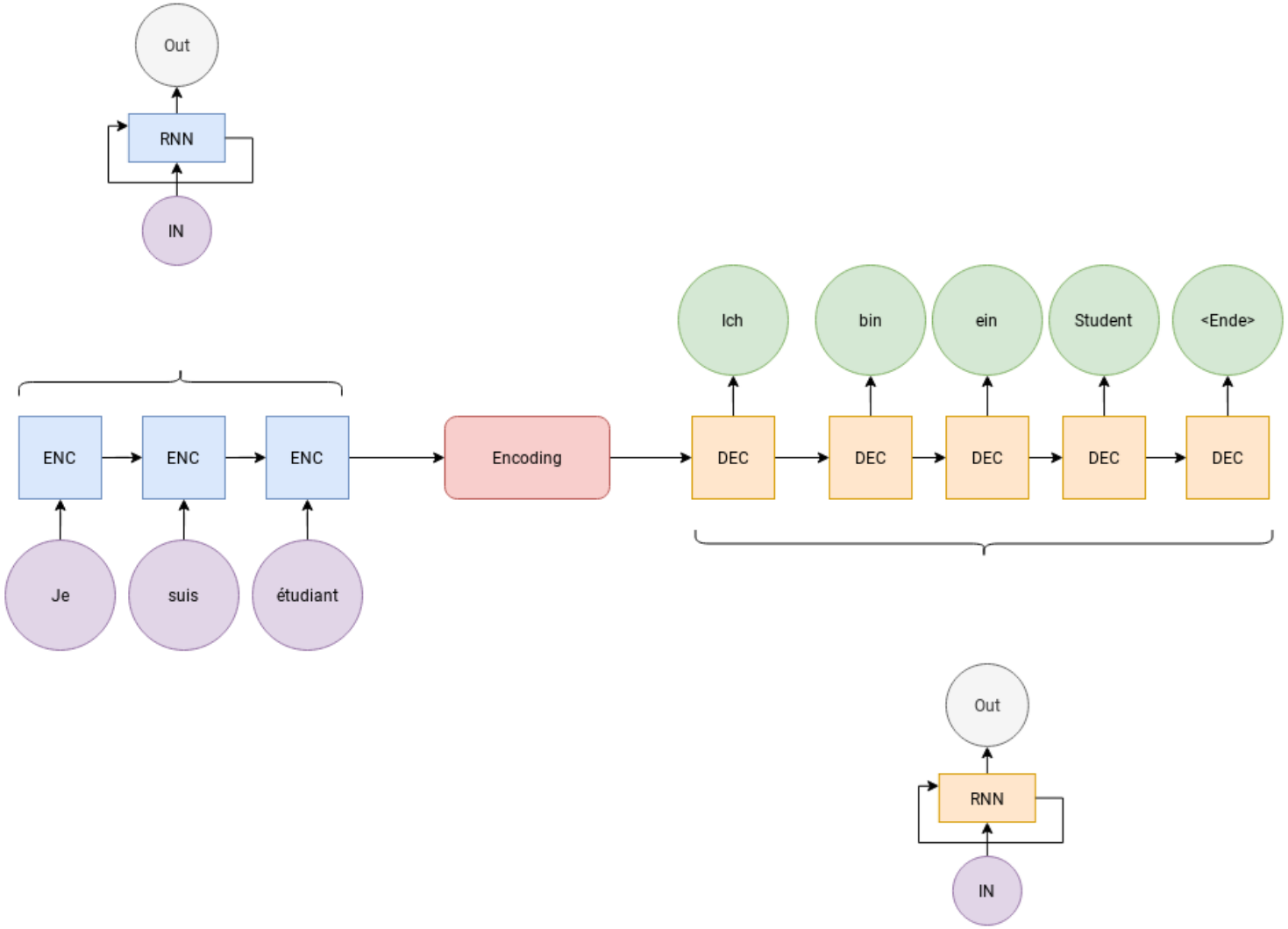
Back to Where it all started……

The Transformer was created in the context of language processing.

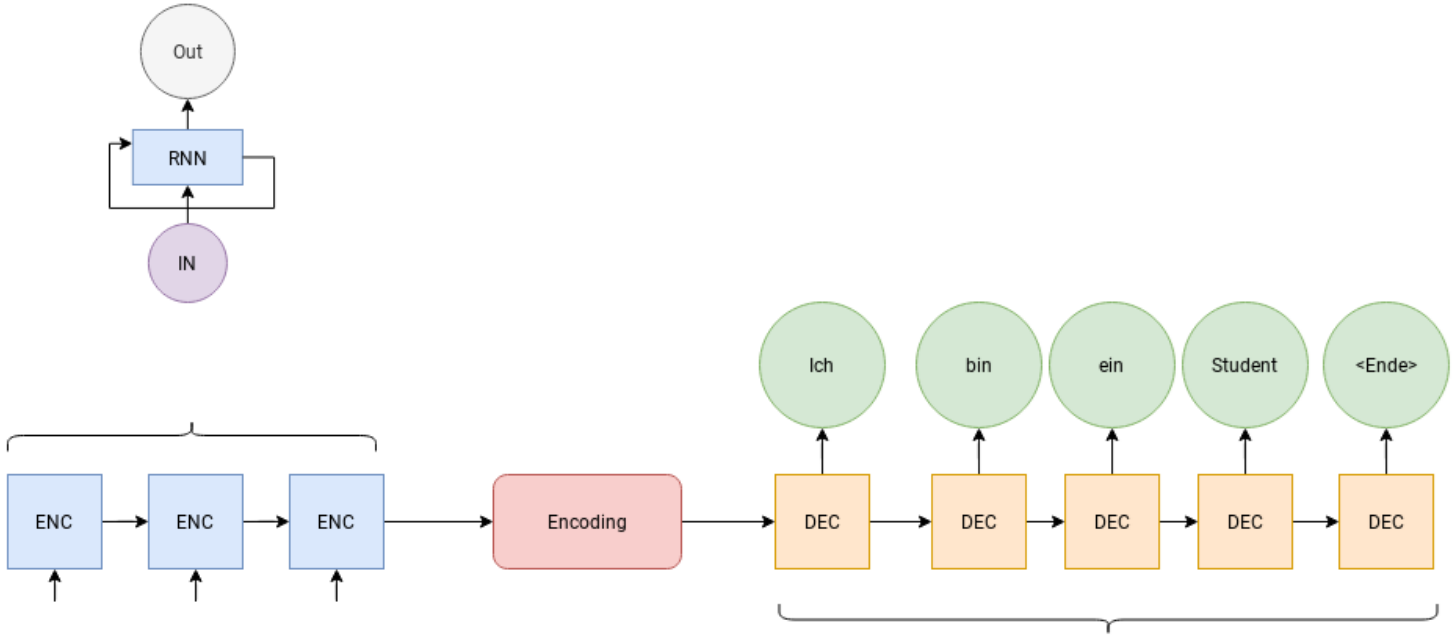

Je suis étudiant
?
Ich bin ein Student


Je suis étudiant
Encoder
Decoder
Ich bin ein Student

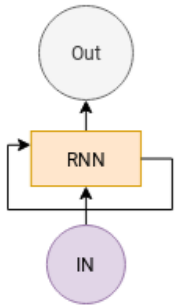
How was it done before: Recurrent Neural Networks



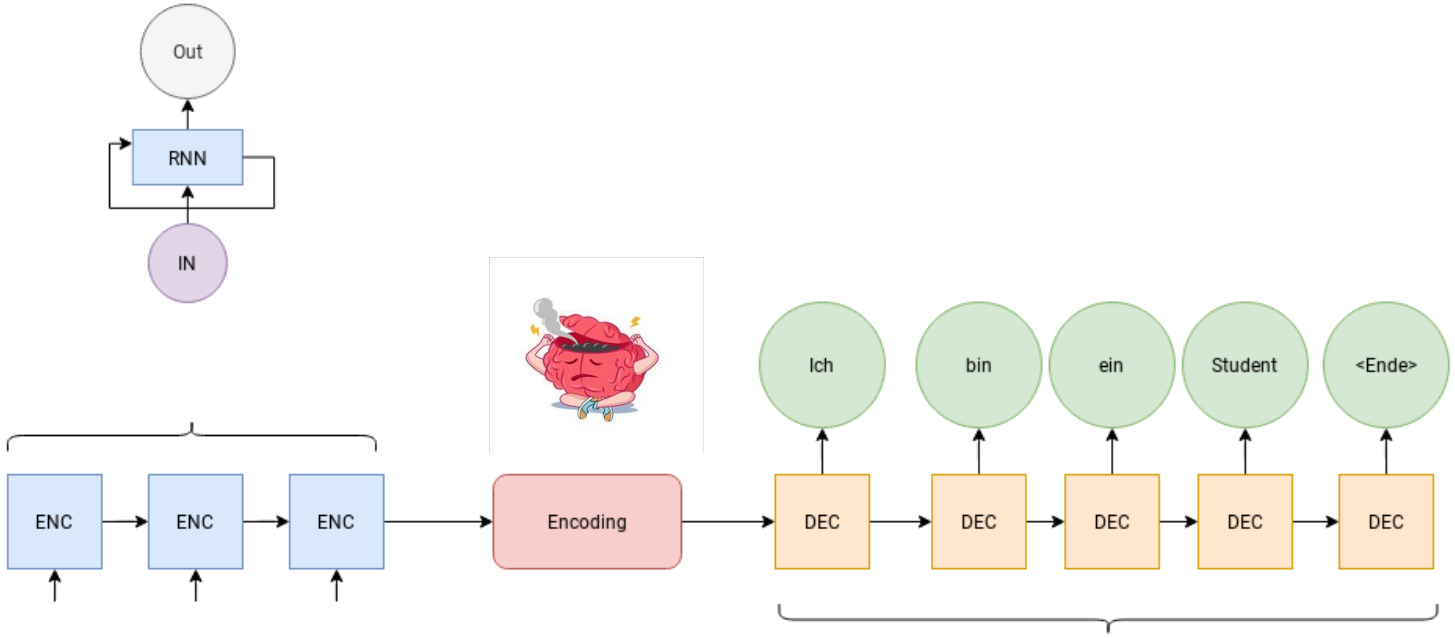
How was it done before: Recurrent Neural Networks



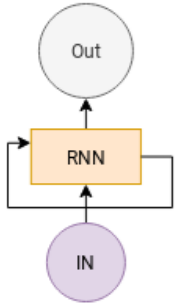
Au commencement, Dieu créa les cieux et la terre.
2 La terre était informe et vide: il y avait des ténèbres à la surface de l'abîme,
et l'esprit de Dieu se mouvait au-dessus des eaux.
3 Dieu dit: Que la lumière soit! Et la lumière fut.
4 Dieu vit que la lumière était bonne; et Dieu sépara la lumière d'avec les ténèbres.
5 Dieu appela la lumière jour, et il appela les ténèbres nuit. Ainsi, il y eut un soir,
et il y eut un matin: ce fut le premier jour.
6 Dieu dit: Qu'il y ait une étendue entre les eaux, et qu'elle sépare les eaux d'avec
les eaux.
7 Et Dieu fit l'étendue, et il sépara les eaux qui sont au-dessous de
l'étendue d'avec les eaux qui sont au-dessus de l'étendue. Et cela fut ainsi.
8 Dieu appela l'étendue ciel. Ainsi, il y eut un soir, et il y eut un matin: ce fut le
second jour.
9 Dieu dit: Que les eaux qui sont au-dessous du ciel se rassemblent en un seul
lieu,
et que le sec paraisse. Et cela fut ainsi.
10 Dieu appela le sec terre, et il appela l'amas des eaux mers. Dieu vit que cela
était bon.



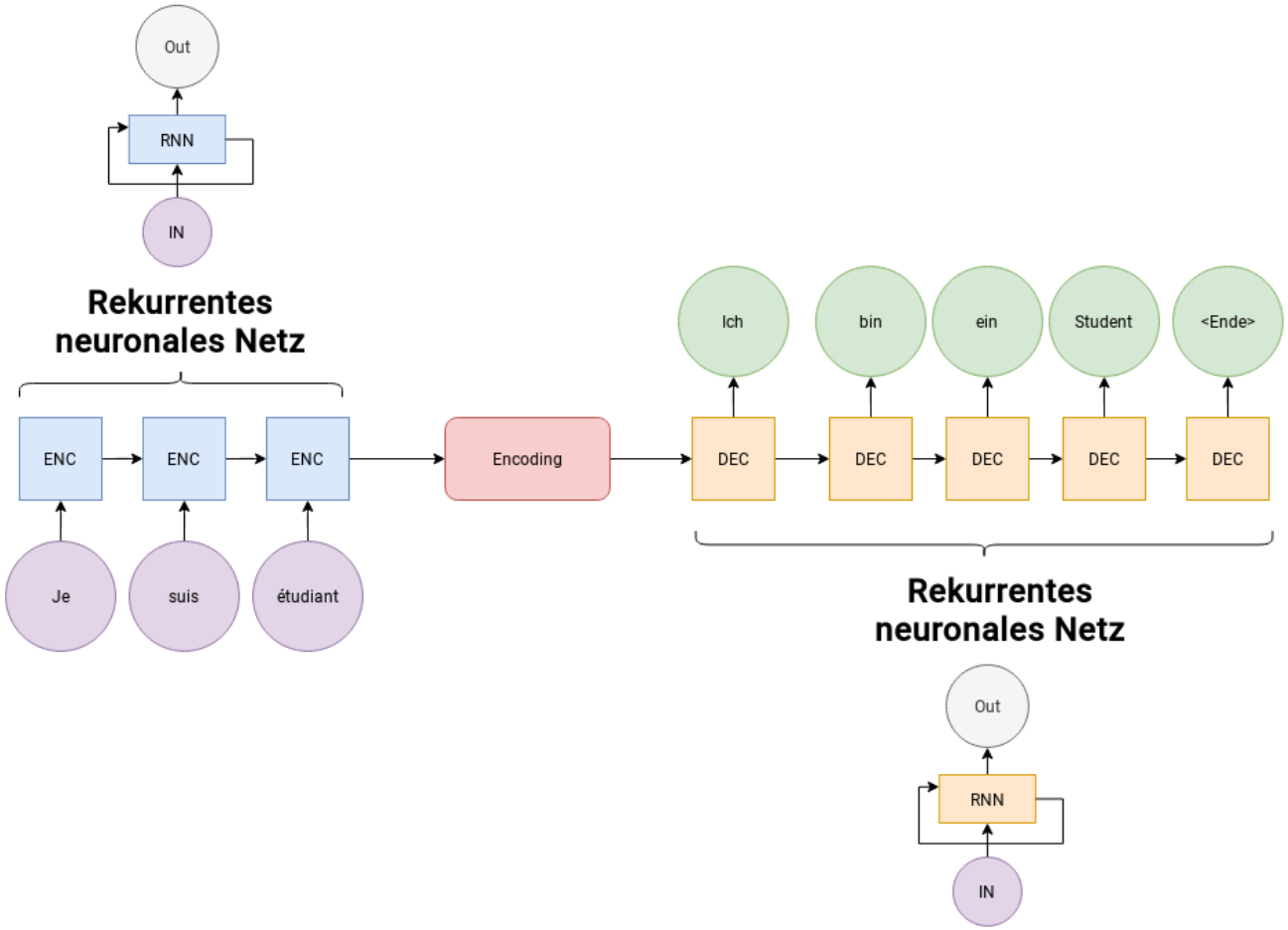
How was it done before: Recurrent Neural Networks



Au commencement, Dieu créa les cieux et la terre.
2 La terre était informe et vide: il y avait des ténèbres à la surface de l'abîme,
et l'esprit de Dieu se mouvait au-dessus des eaux.
3 Dieu dit: Que la lumière soit! Et la lumière fut.
4 Dieu vit que la lumière était bonne; et Dieu sépara la lumière d'avec les ténèbres.
5 Dieu appela la lumière jour, et il appela les ténèbres nuit. Ainsi, il y eut un soir,
et il y eut un matin: ce fut le premier jour.
6 Dieu dit: Qu'il y ait une étendue entre les eaux, et qu'elle sépare les eaux d'avec
les eaux.
7 Et Dieu fit l'étendue, et il sépara les eaux qui sont au-dessous de
l'étendue d'avec les eaux qui sont au-dessus de l'étendue. Et cela fut ainsi.
8 Dieu appela l'étendue ciel. Ainsi, il y eut un soir, et il y eut un matin: ce fut le
second jour.
9 Dieu dit: Que les eaux qui sont au-dessous du ciel se rassemblent en un seul
lieu,
et que le sec paraisse. Et cela fut ainsi.
10 Dieu appela le sec terre, et il appela l'amas des eaux mers. Dieu vit que cela
était bon.

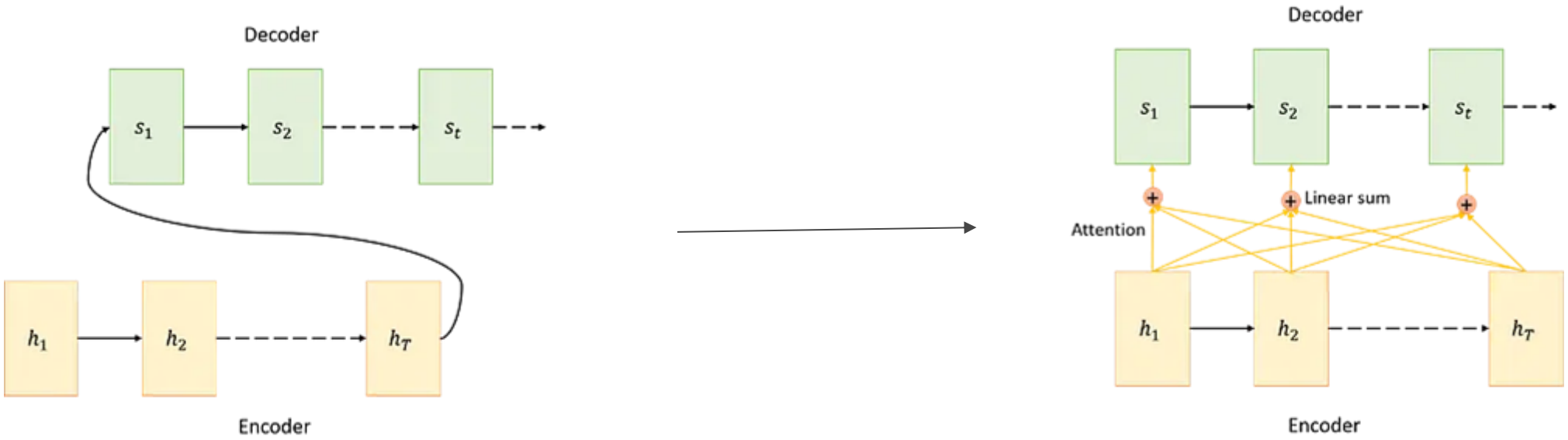


How was it done before: Recurrent Neural Networks



- The entire semantic context must be stored in a vector.
- Distant connections become difficult.
- The “recurrent” part seems to be the problem.

Just an extension of classic RNNs… Attention Mechanism



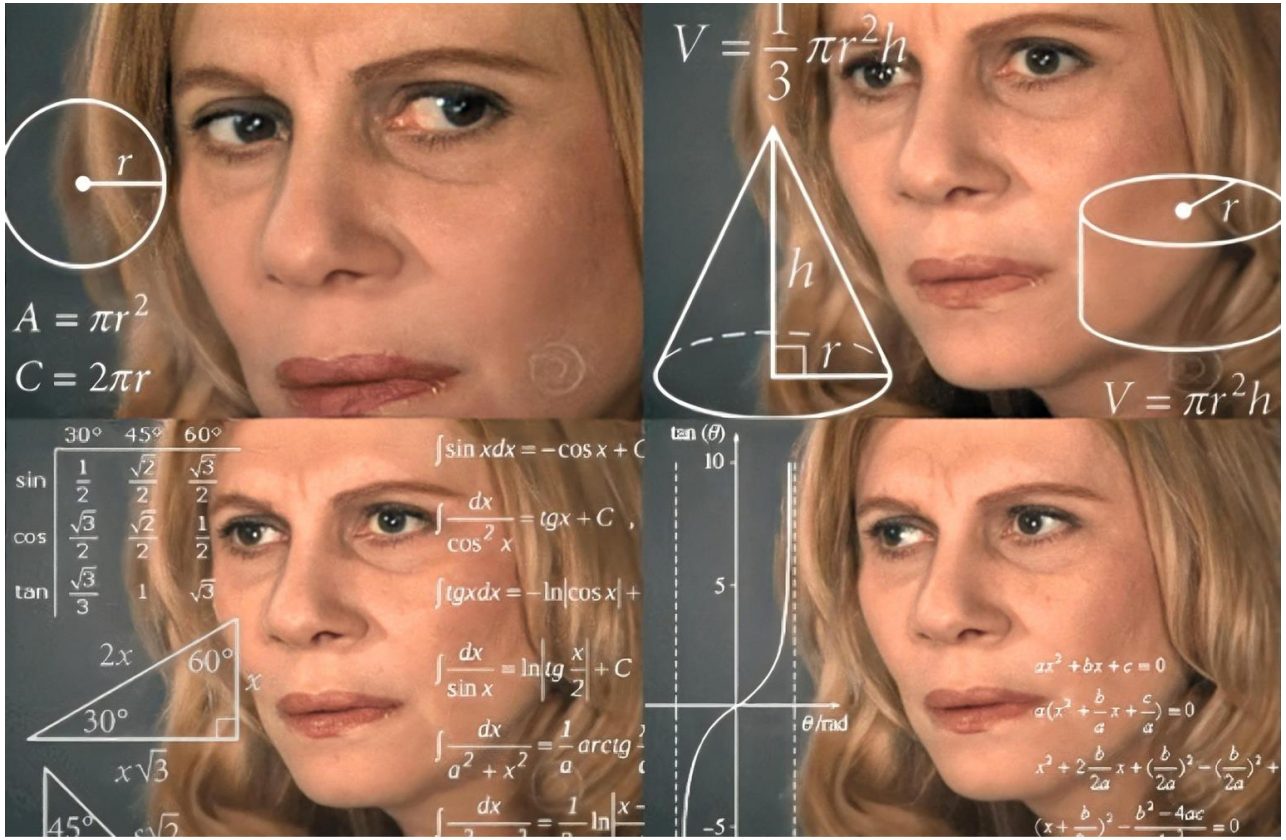
How do we consider the full context for each word individually?

- Intuition behind attention: focusing on relevant parts of the input data.
- Everyday analogy: reading and selectively focusing on key information.
- **Attention** mechanism helps to **look at all hidden states from encoder sequence for making predictions** unlike vanilla Encoder-Decoder approach.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

How does Attention work?

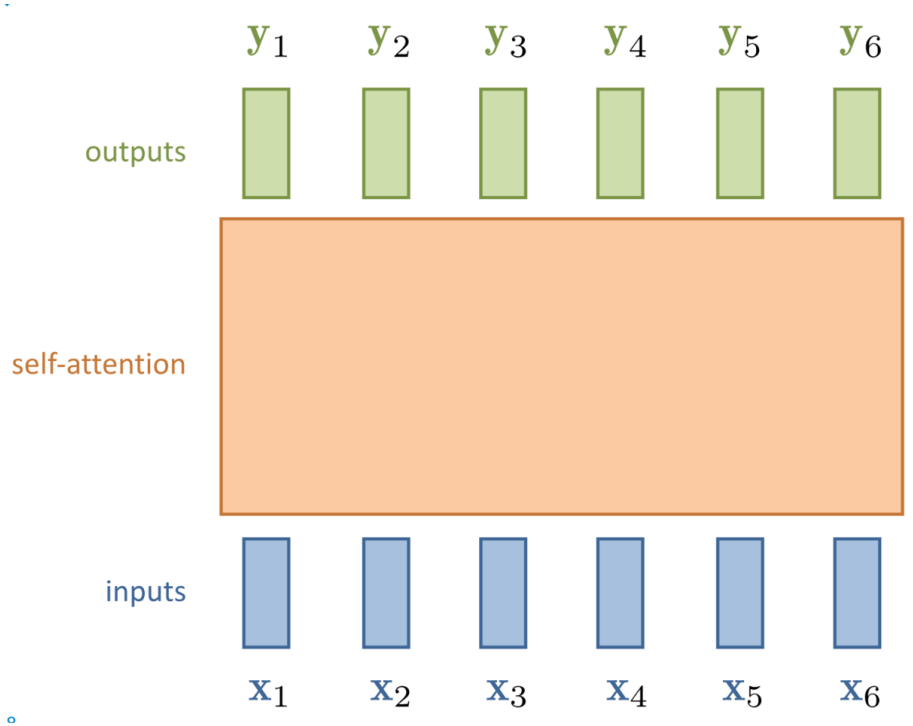
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



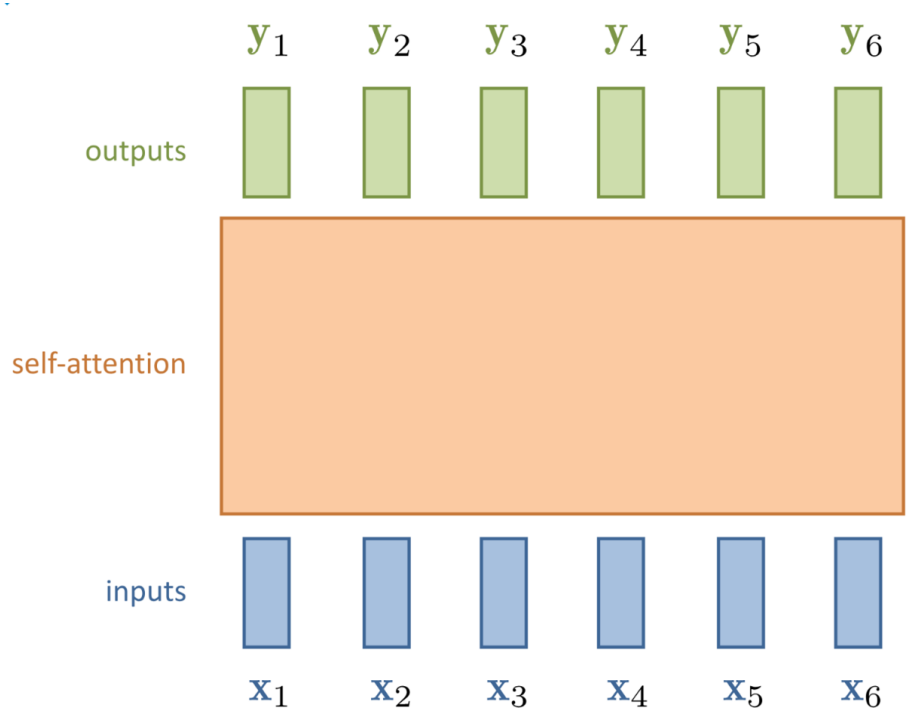
How does Attention work?

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- The fundamental operation in the transformers is the attention mechanism.
- At its core: It is a seq2seq operation.
- Seq2seq operation:



Intuition to Attention Mechanism

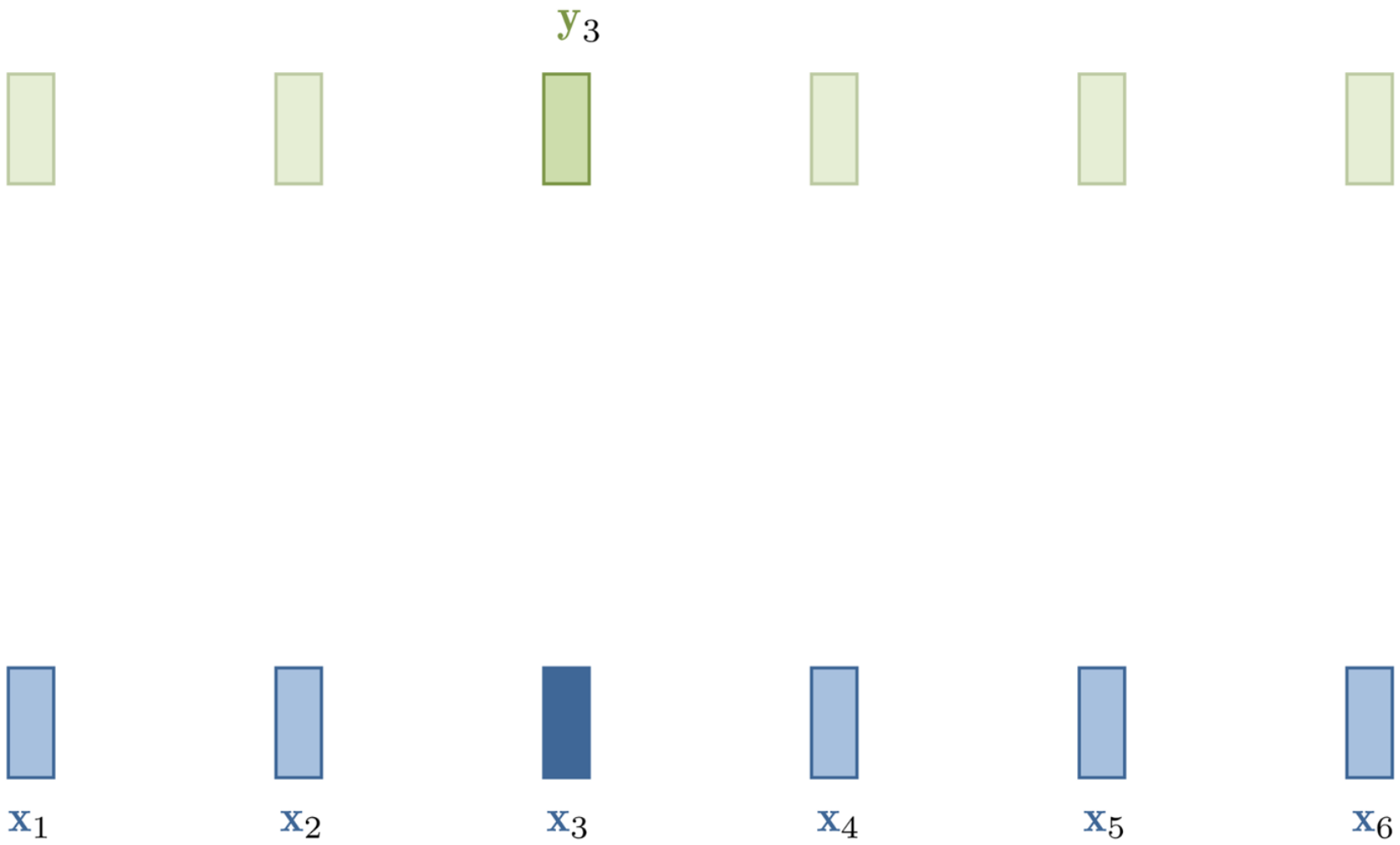


$$y_i = \sum_j w_{ij} x_j$$

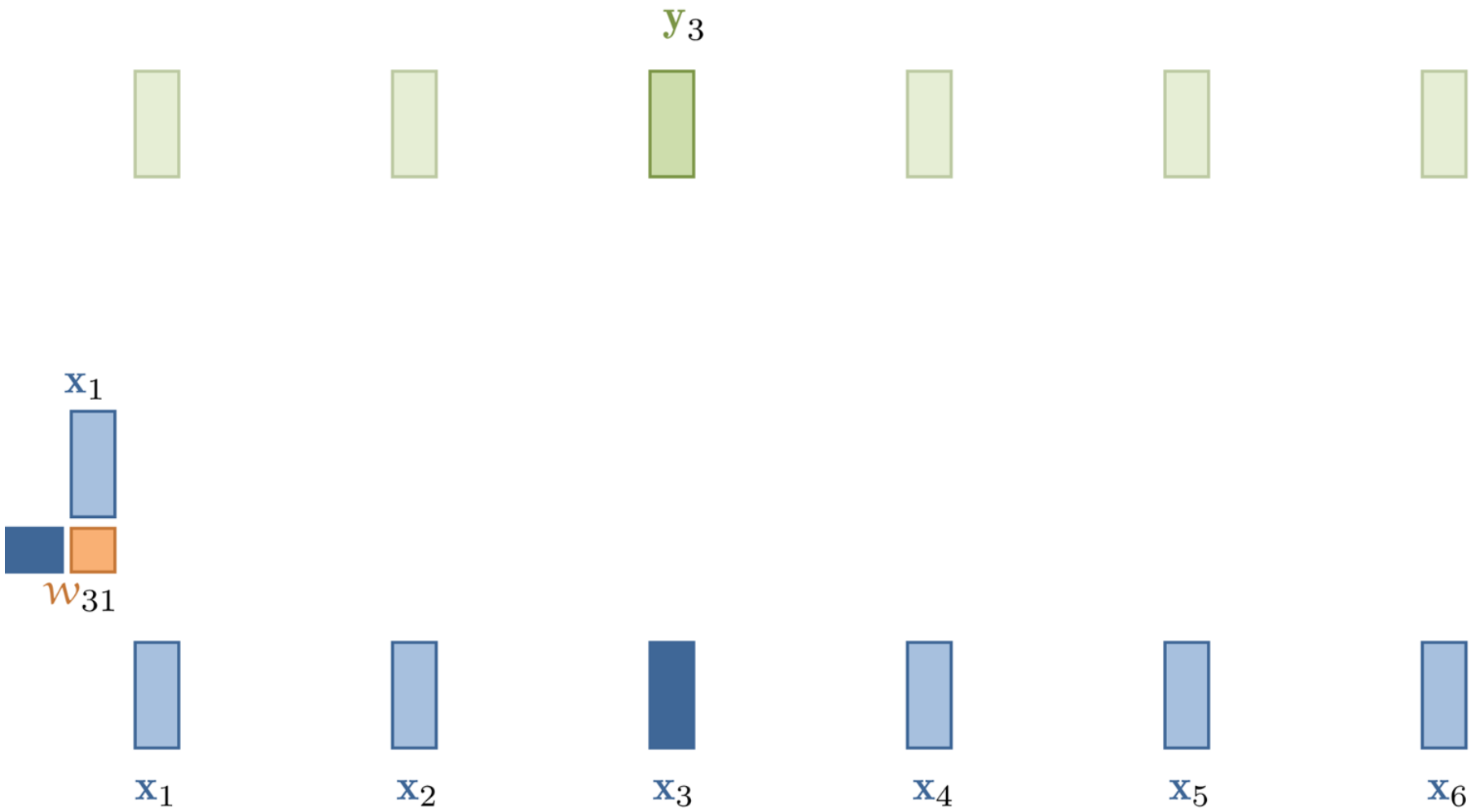
$$w'_{ij} = x_i^T x_j$$

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

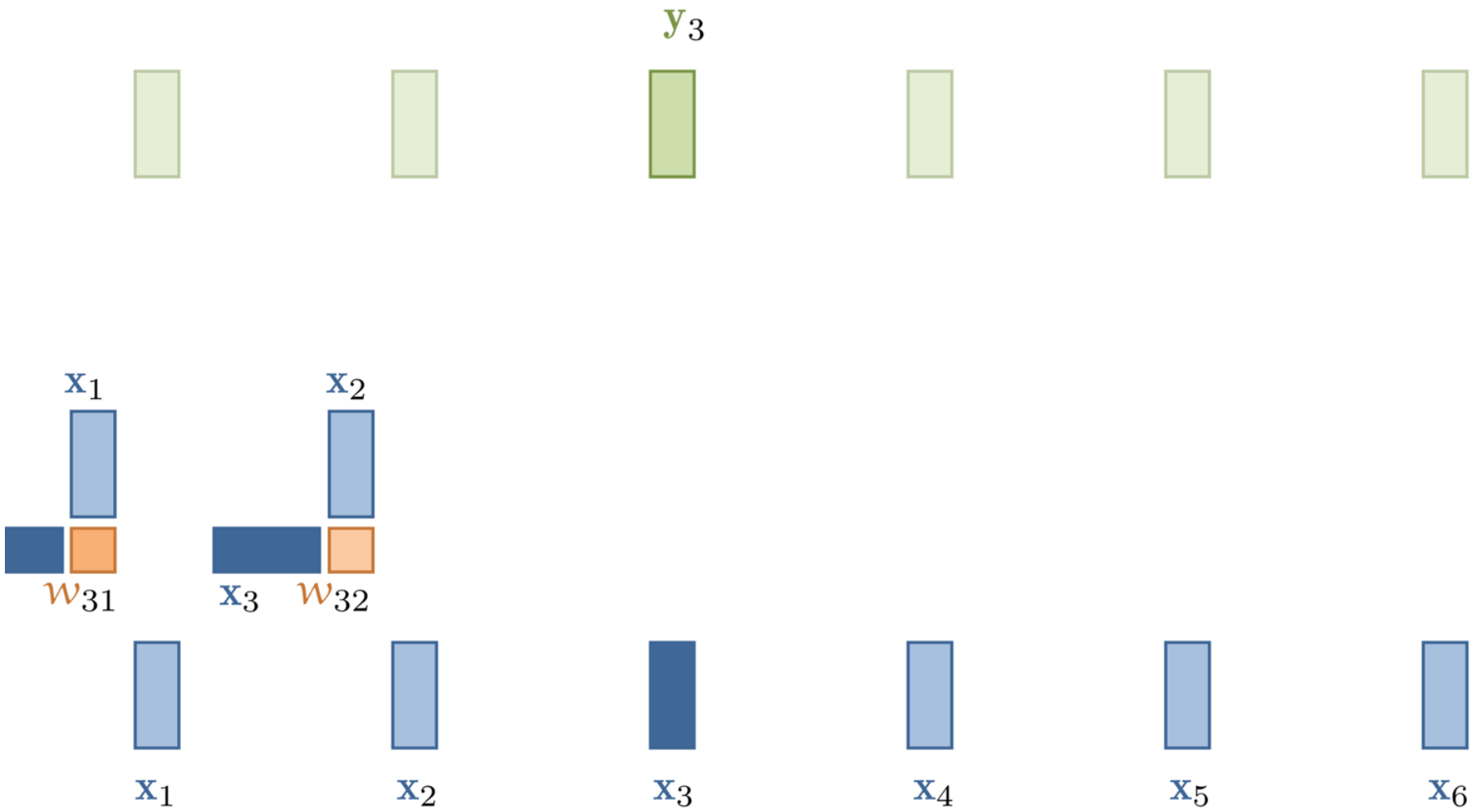
Intuition to Attention Mechanism



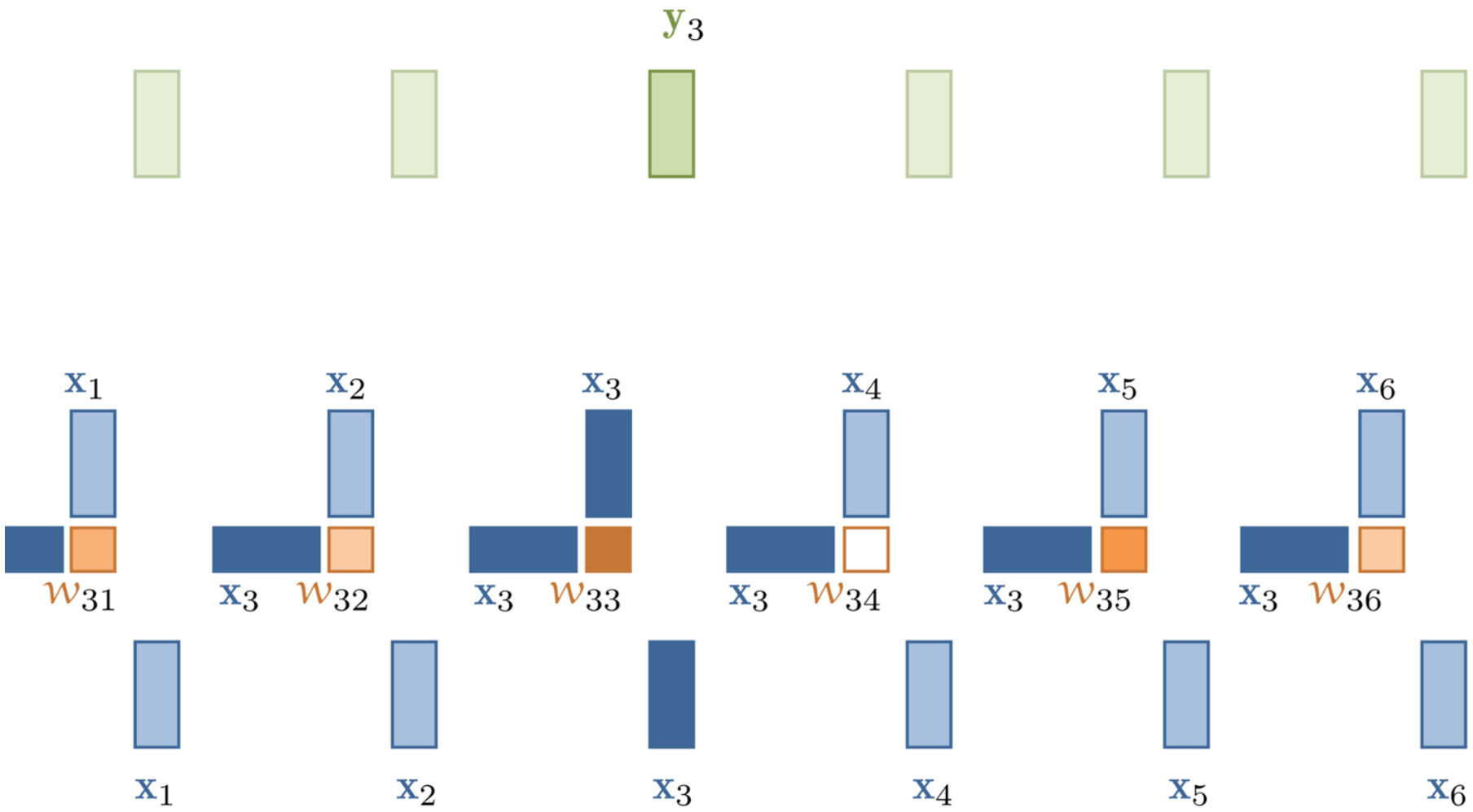
Intuition to Attention Mechanism



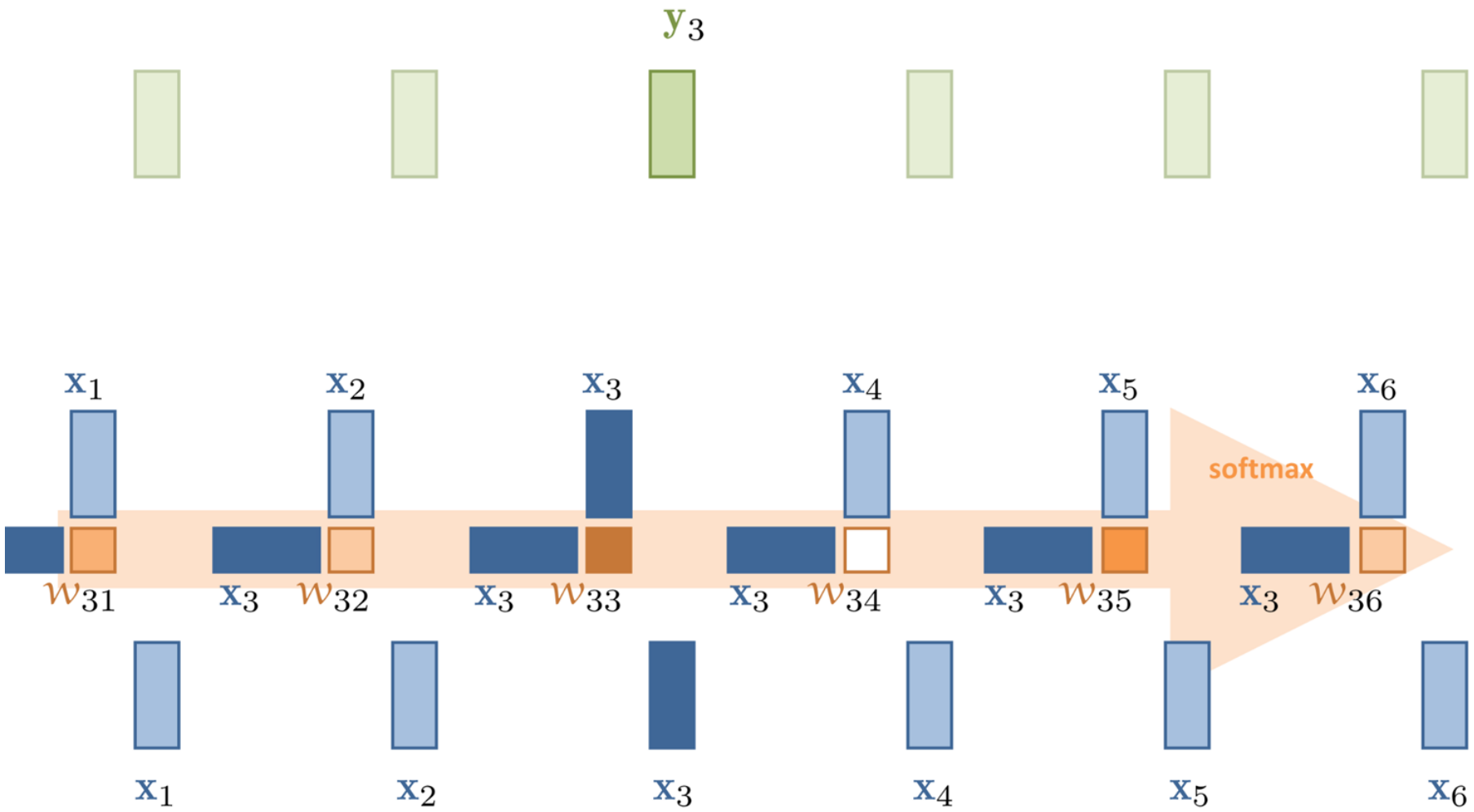
Intuition to Attention Mechanism



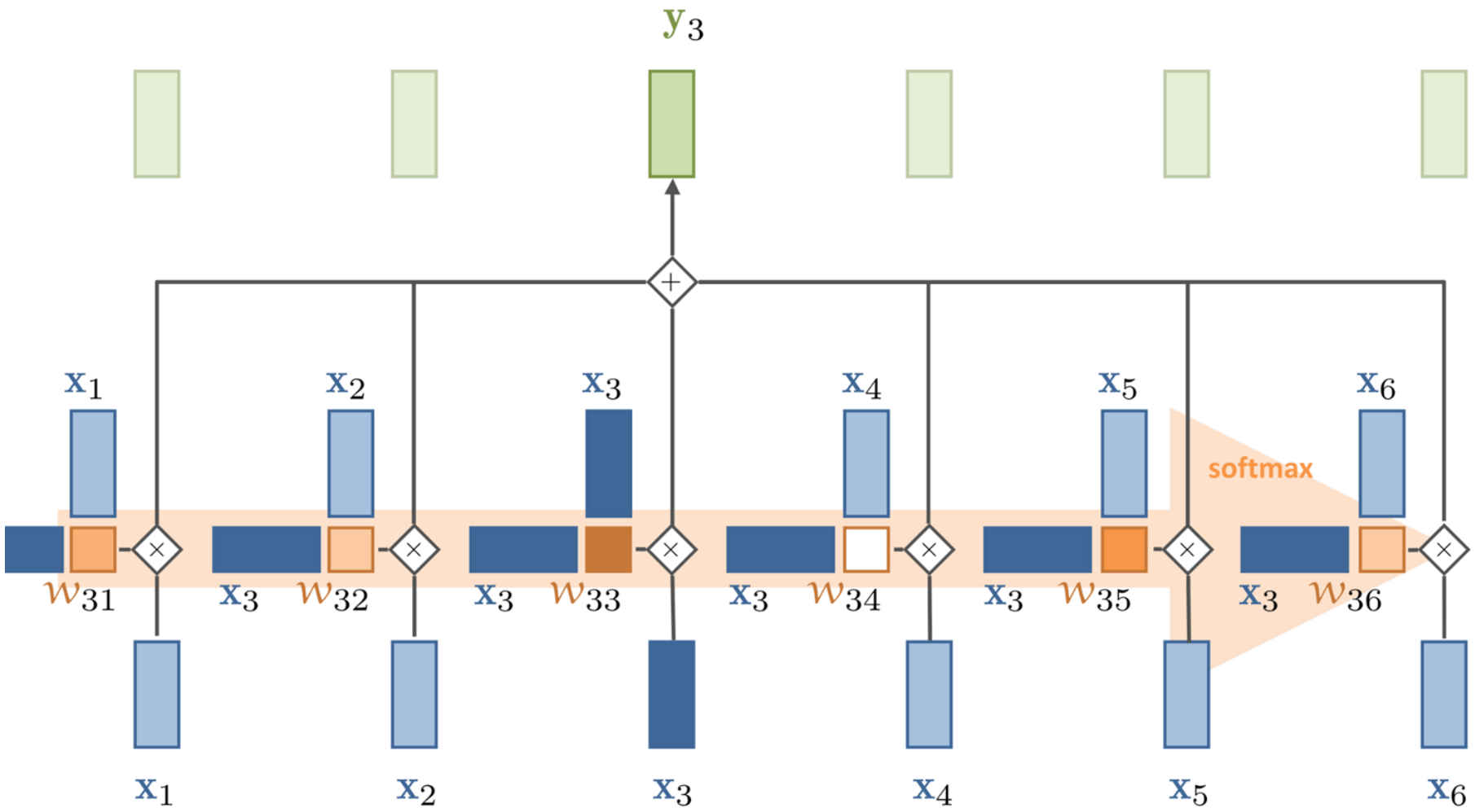
Intuition to Attention Mechanism



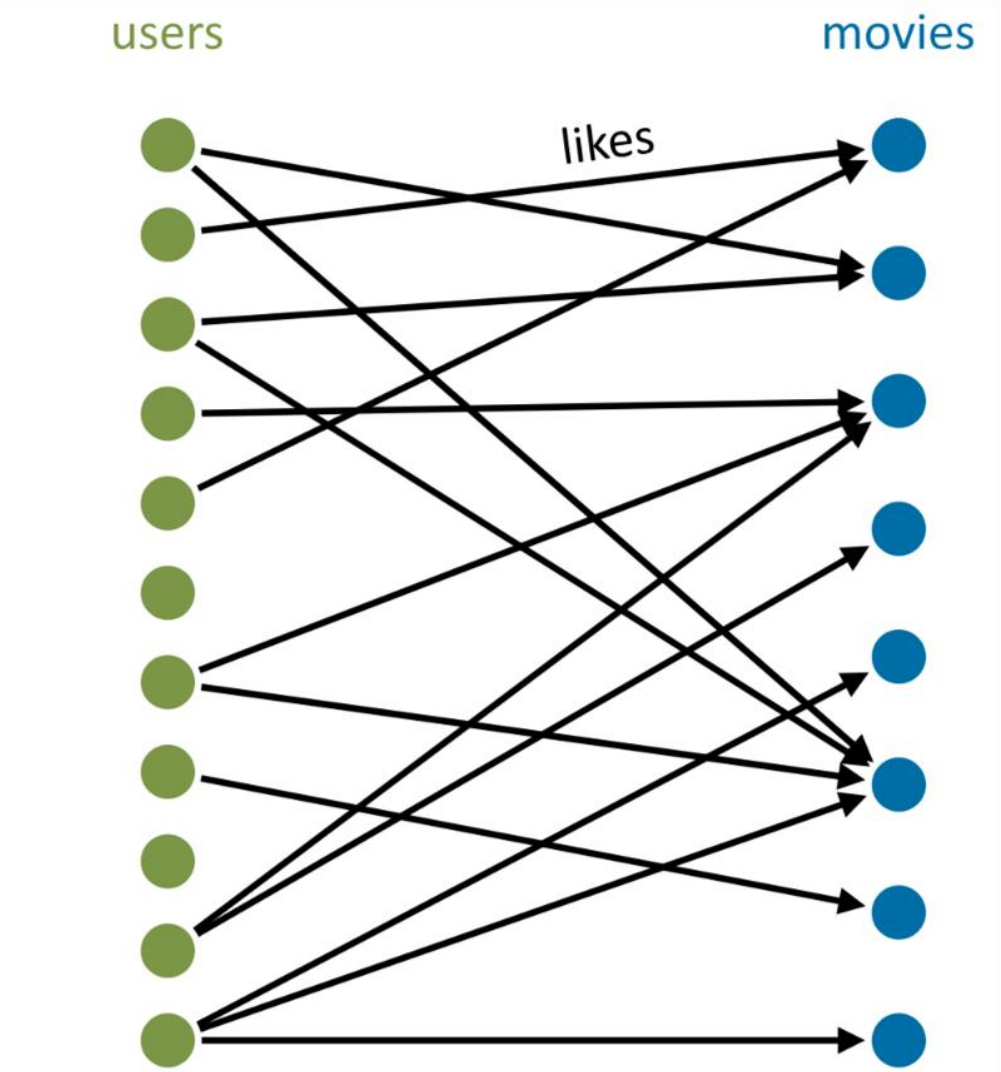
Intuition to Attention Mechanism

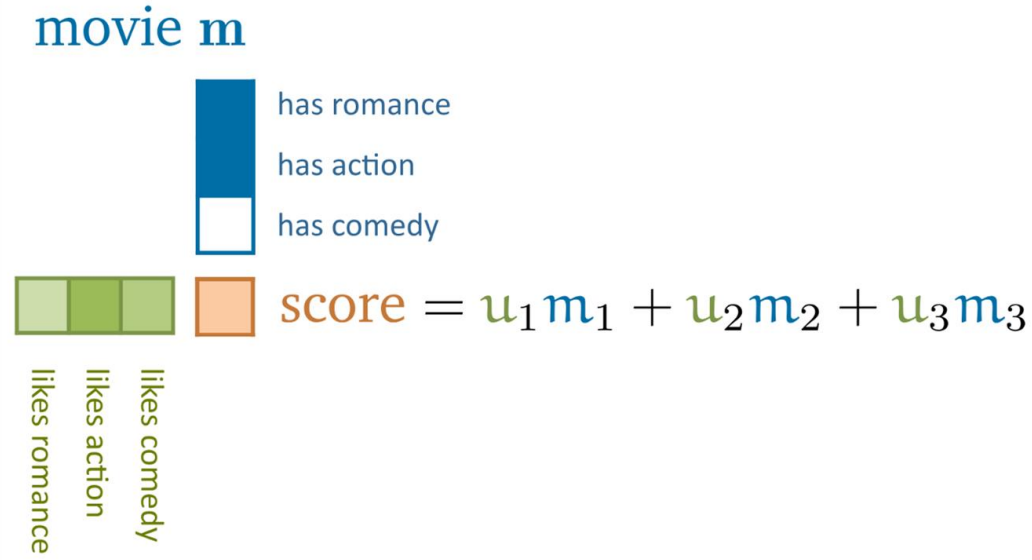


Intuition to Attention Mechanism



A little more intuition





Final additions

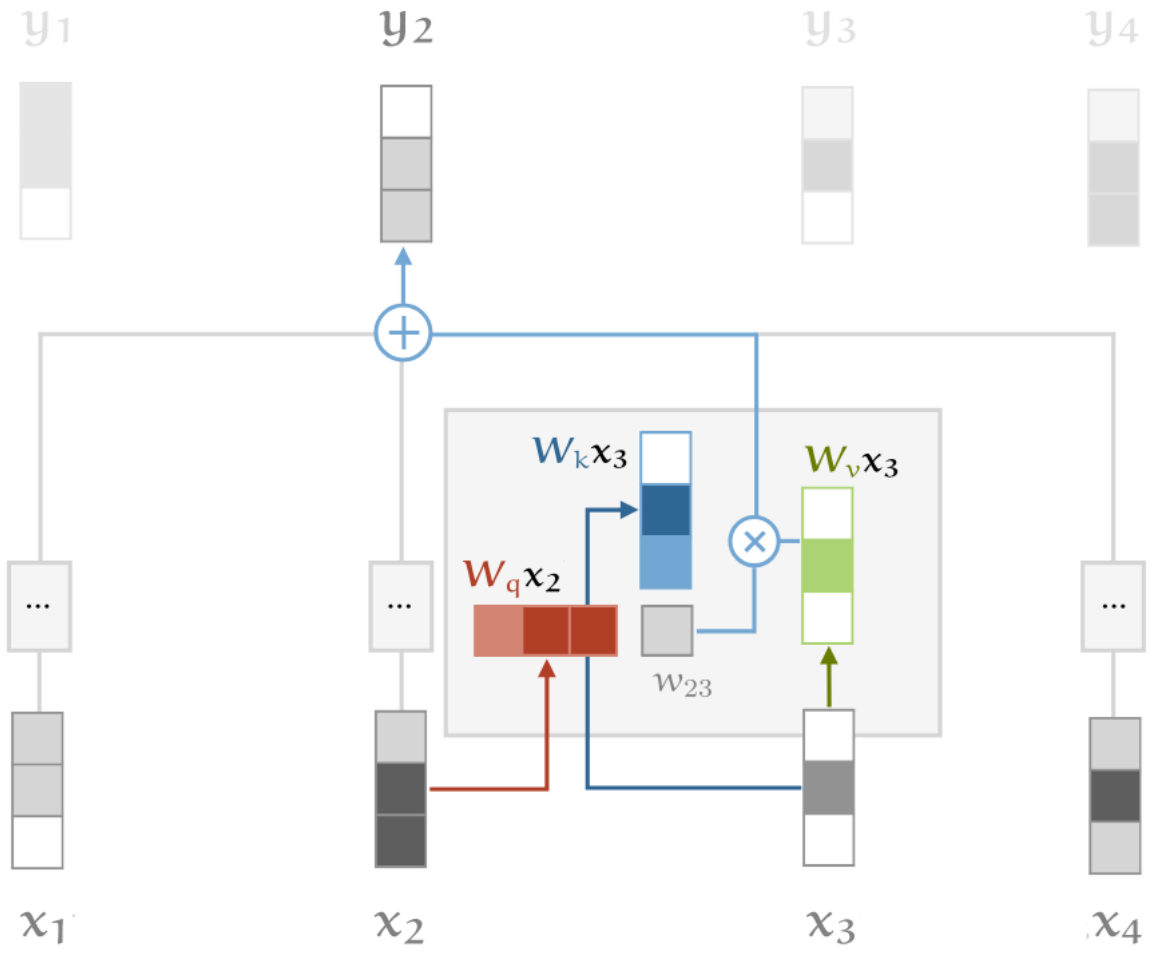
- The actual self-attention in transformers uses three more additional tricks.

- Query, Key and Value:
    - **Query:** It is compared to every other vector to establish the weights for its own output.
    - **Key:** It is compared to every other vector to establish the weights for the output of the j-th vector.
    - **Value:** It is used as part of the weighted sum to compute each output vector once the weights have been established.

- Every input vector is converted to 3 vectors - Query, Key, Value using simple matrix multiplication.

- This makes the attention mechanism learnable.

Final Additions: Q, K, Vs and learnable attention



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

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

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

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

Final Additions: Scaling and Multi head

- **Scaling the dot product**
  - The attention is scaled by square root of it's dimension.
  - This is to make softmax more stable.

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

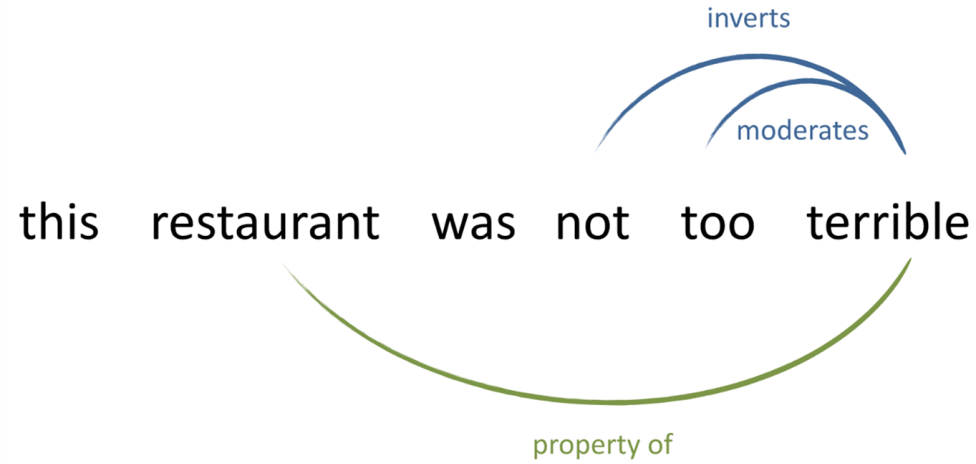
- **Multi Head Attention**
  - Intuition: Different *levels* of attention for fine grained connections.

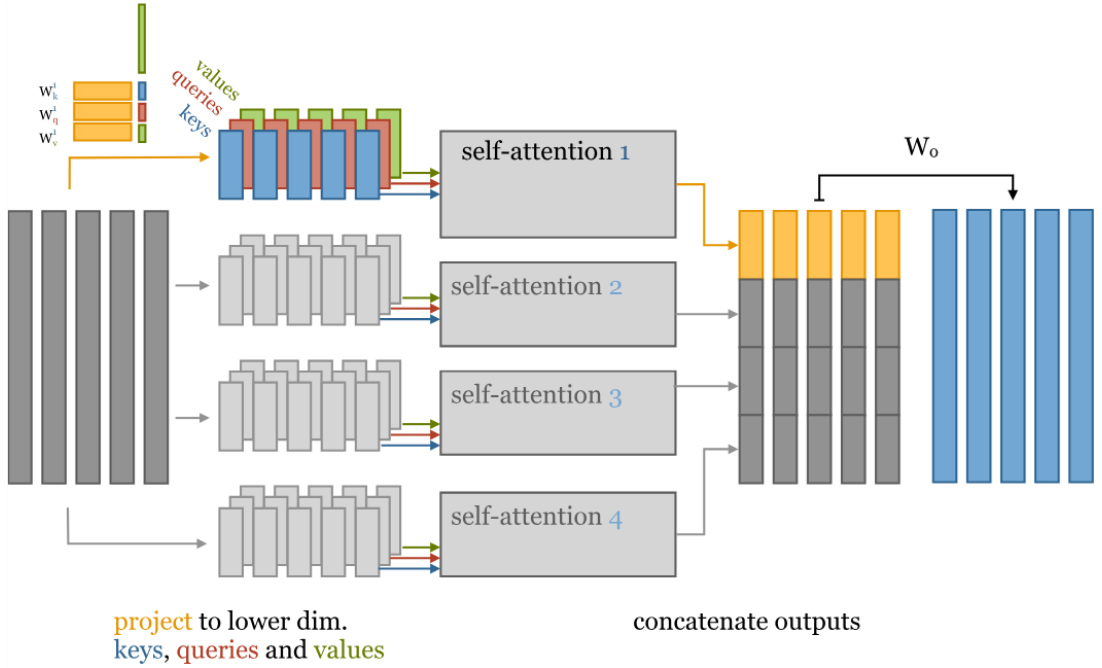
Final Additions: Scaling and Multi head

- **Scaling the dot product**
  - The attention is scaled by square root of it's dimension.
  - This is to make softmax more stable.

$$w'_{ij} = \frac{\mathbf{q}_i{}^T \mathbf{k}_j}{\sqrt{k}}$$

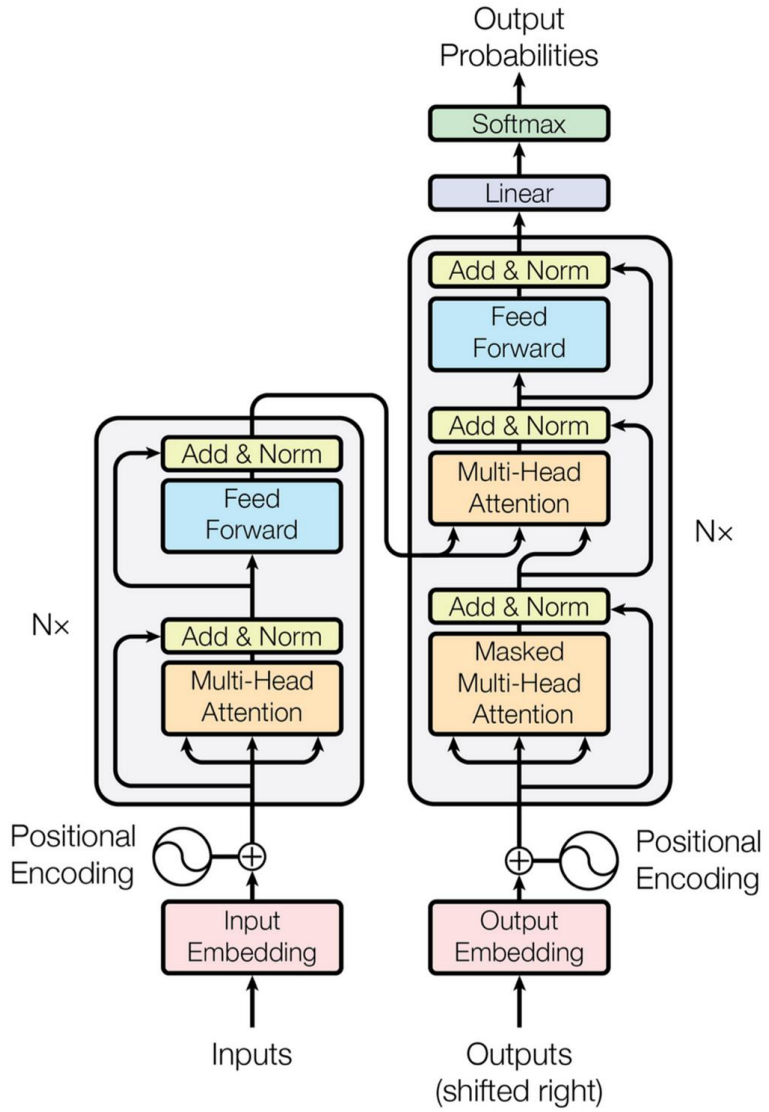
- **Multi Head Attention**
  - Intuition: Different *levels* of attention for fine grained connections.





The Transformer

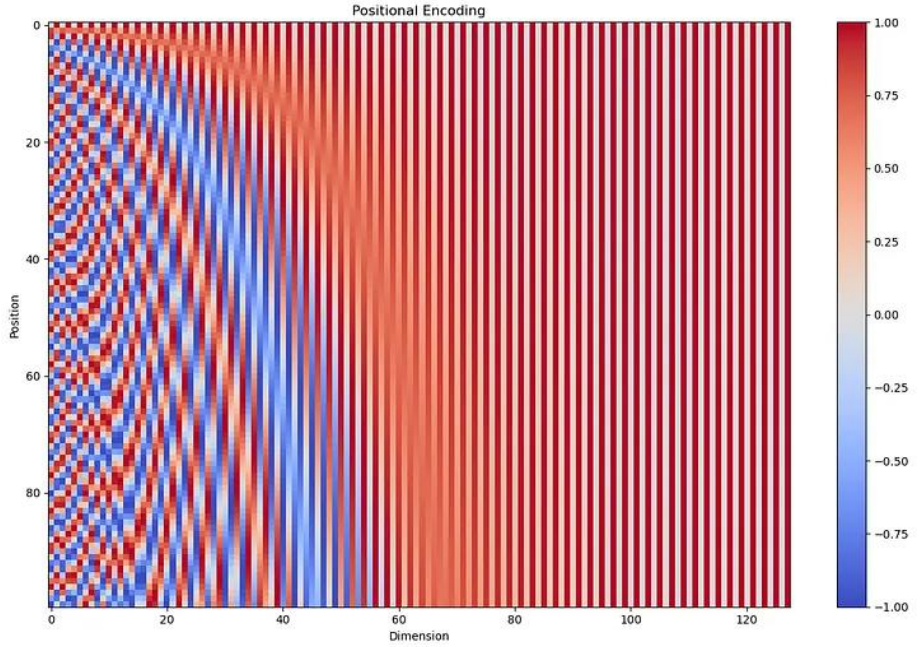
- A transformer is just a stack of attention blocks.
- Encoder-Decoder Architecture.
- The ingredients:
  - Input Embedding
  - Positional Encoding
  - Self Attention
  - Cross Attention
  - Feedforward and Residuals



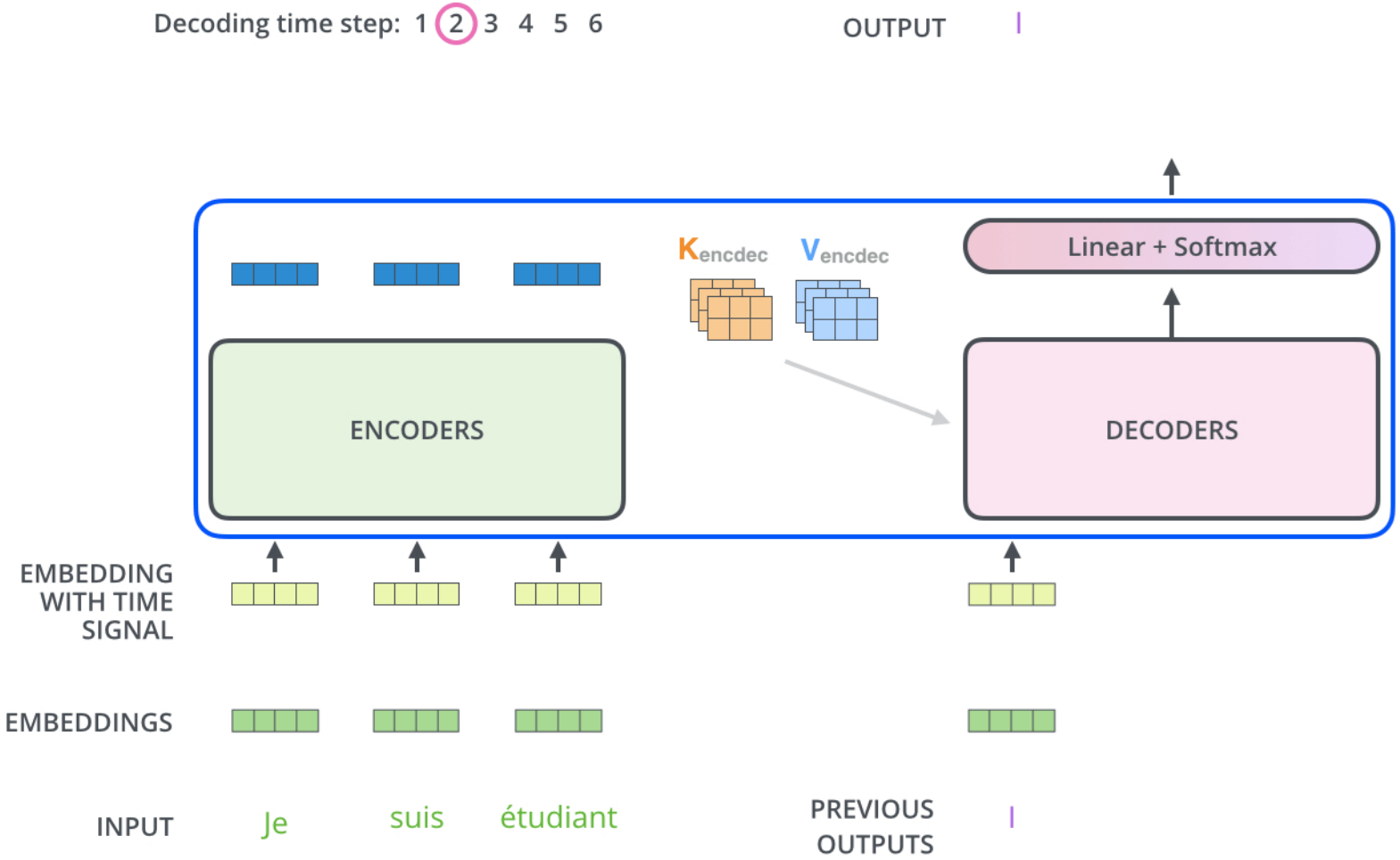
Embedding and Positional Encoding

- A computer doesn’t know any words- only numbers.
- Each word is converted into a number.
- **Embedding:** Each word has a corresponding vector.
- Each word also needs a time stamp- The order matters!
- **Positional Encoding:** We encode the position, using sinusoidal features.

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$



How does it work ?



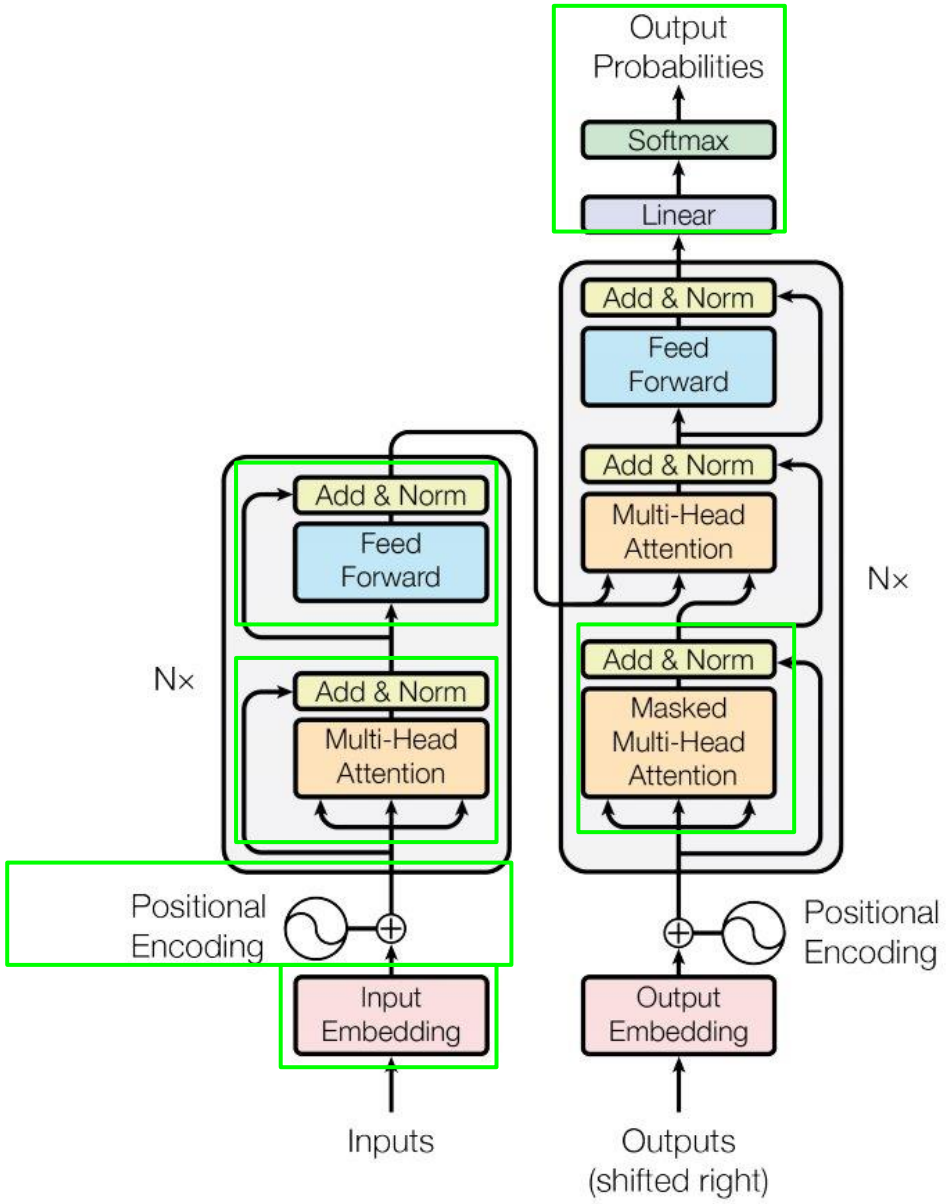


Figure 1: The Transformer - model architecture.

Simply, Do this for the whole text on internet, you have your ChatGPT, Gemini etc.

Vision Transformer

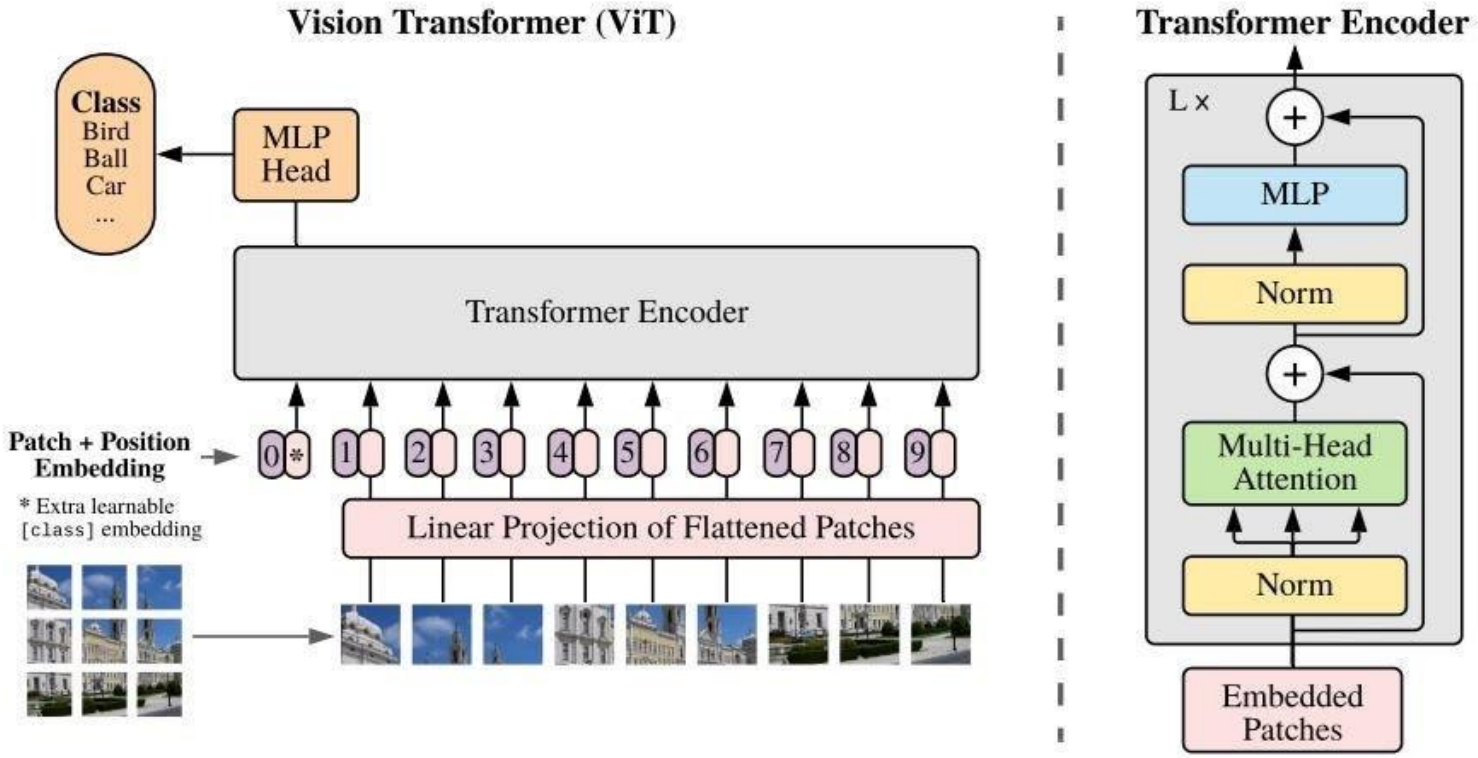


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

Object Detection

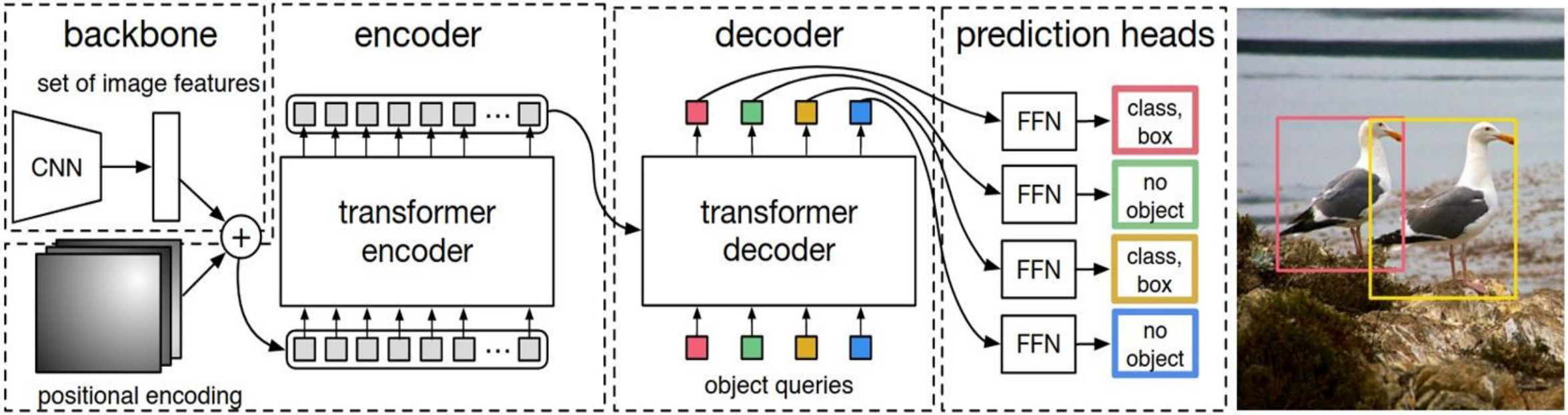
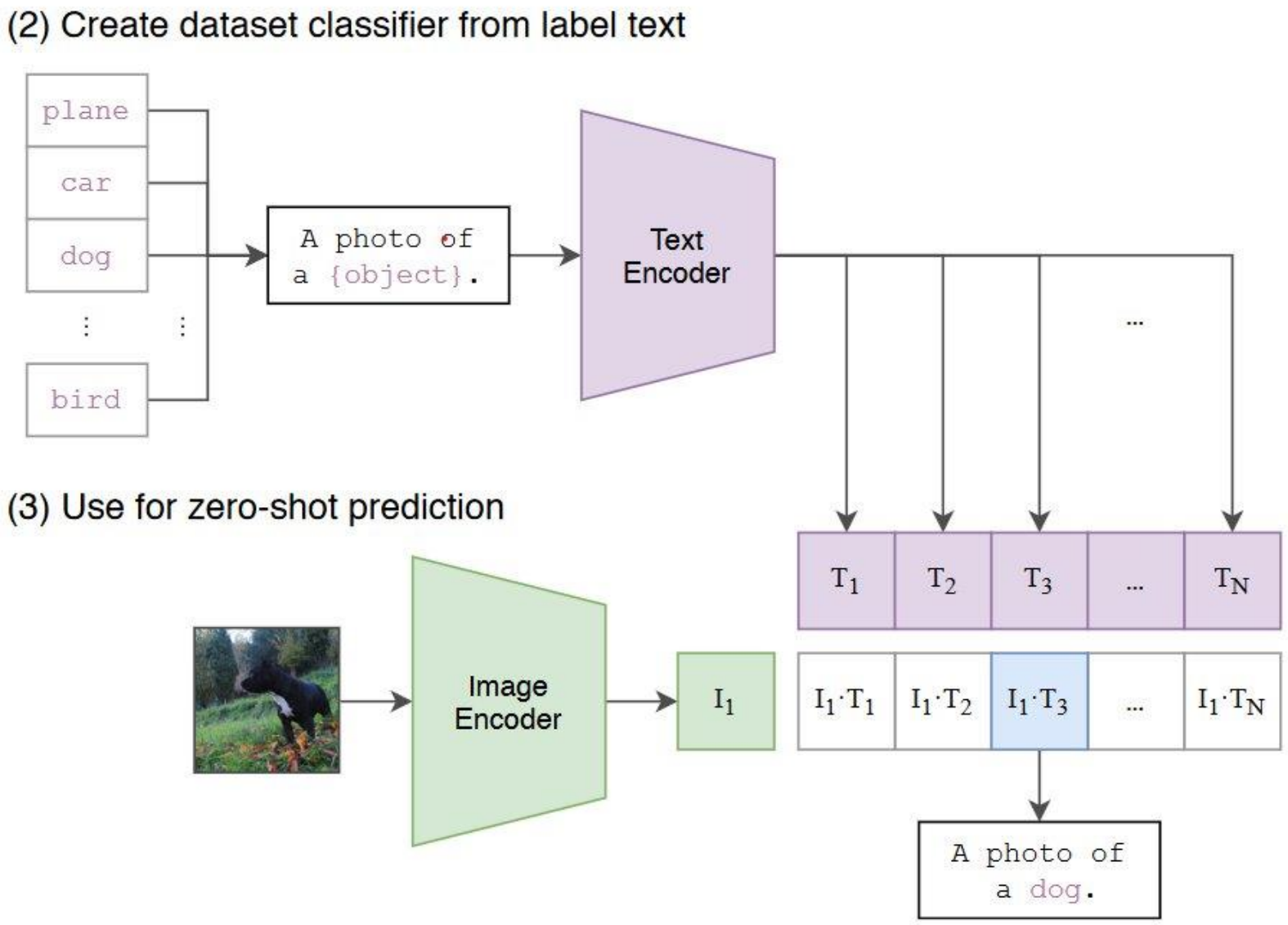
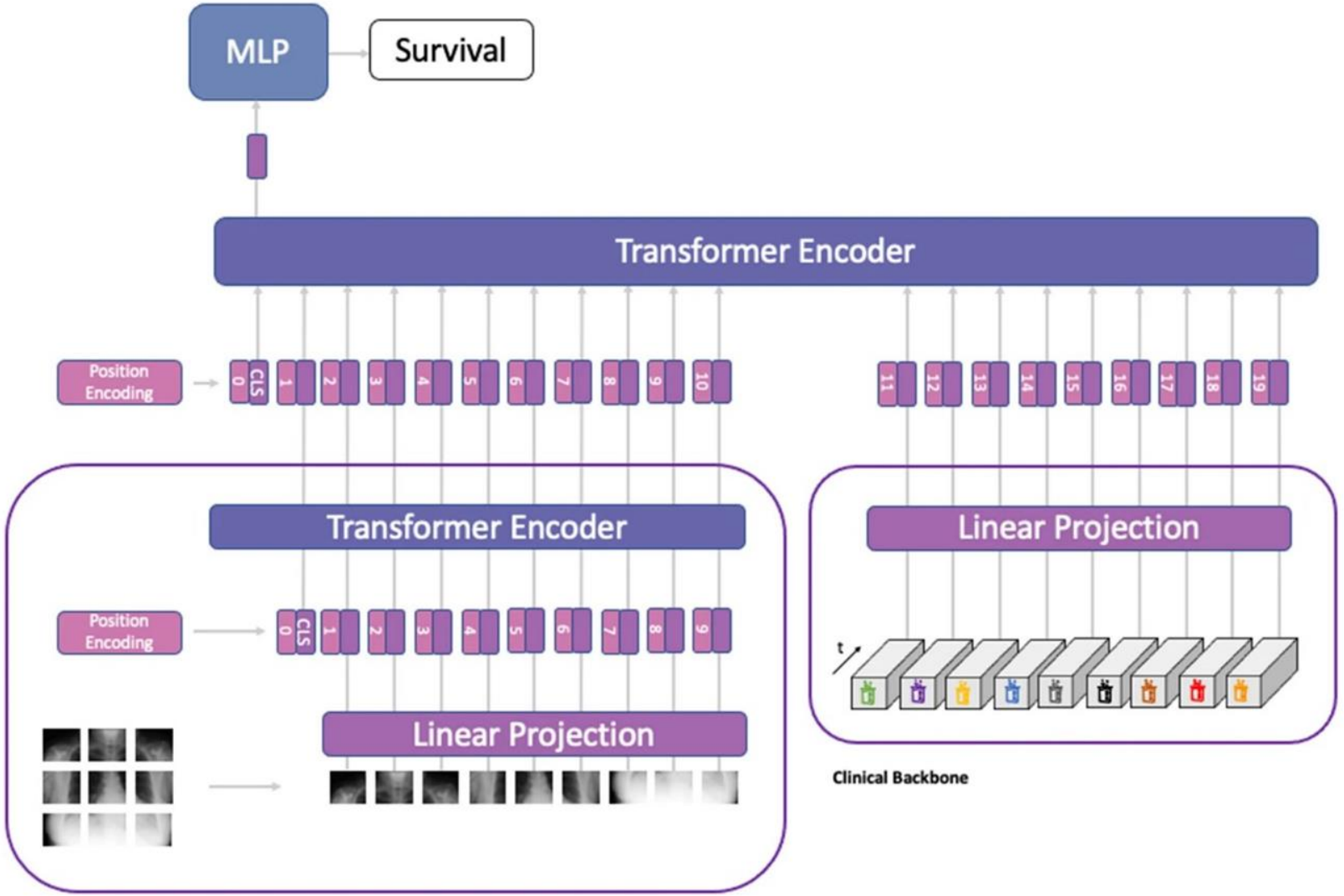


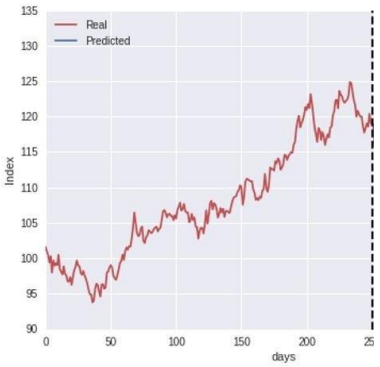
Image Captioning

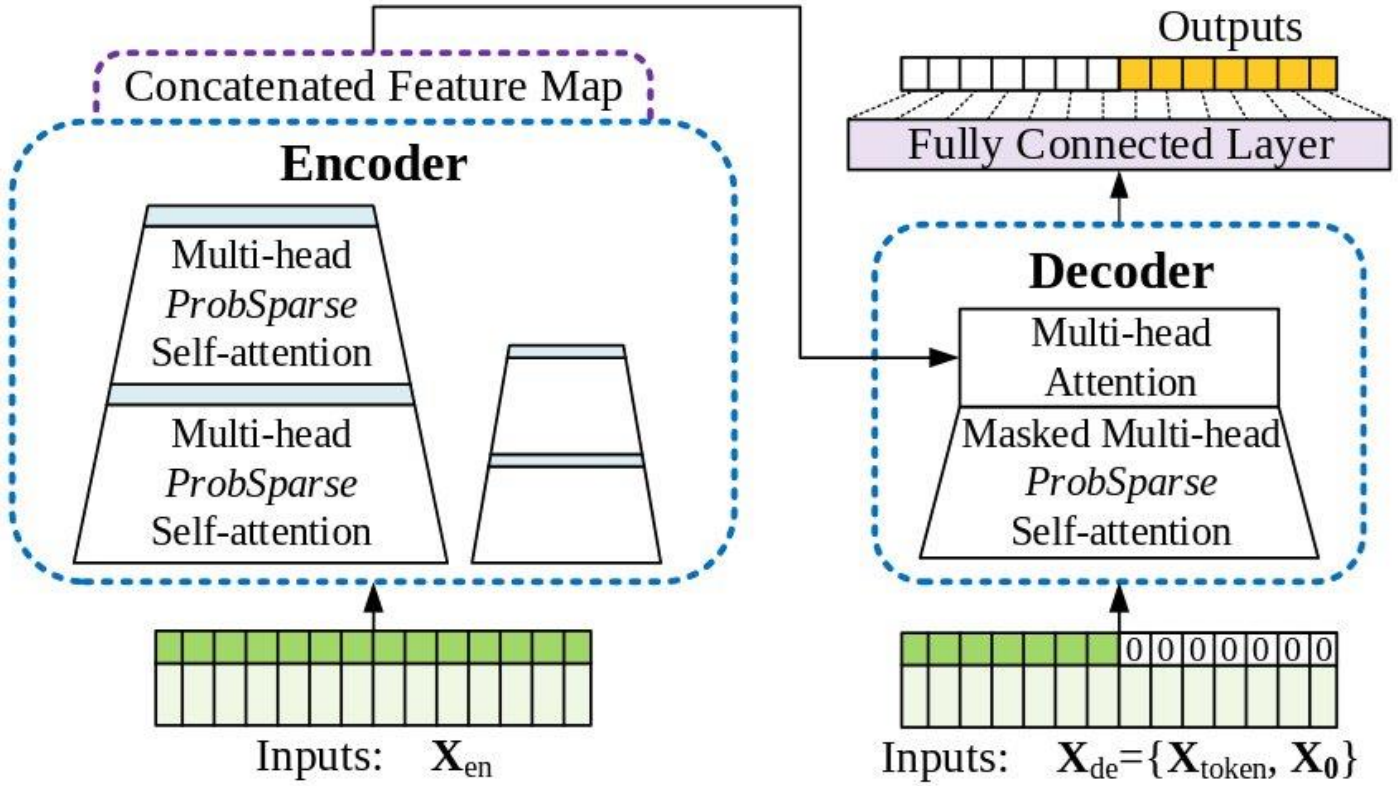


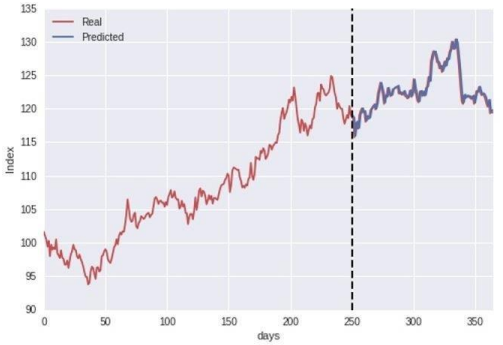
Multi Modal Regression



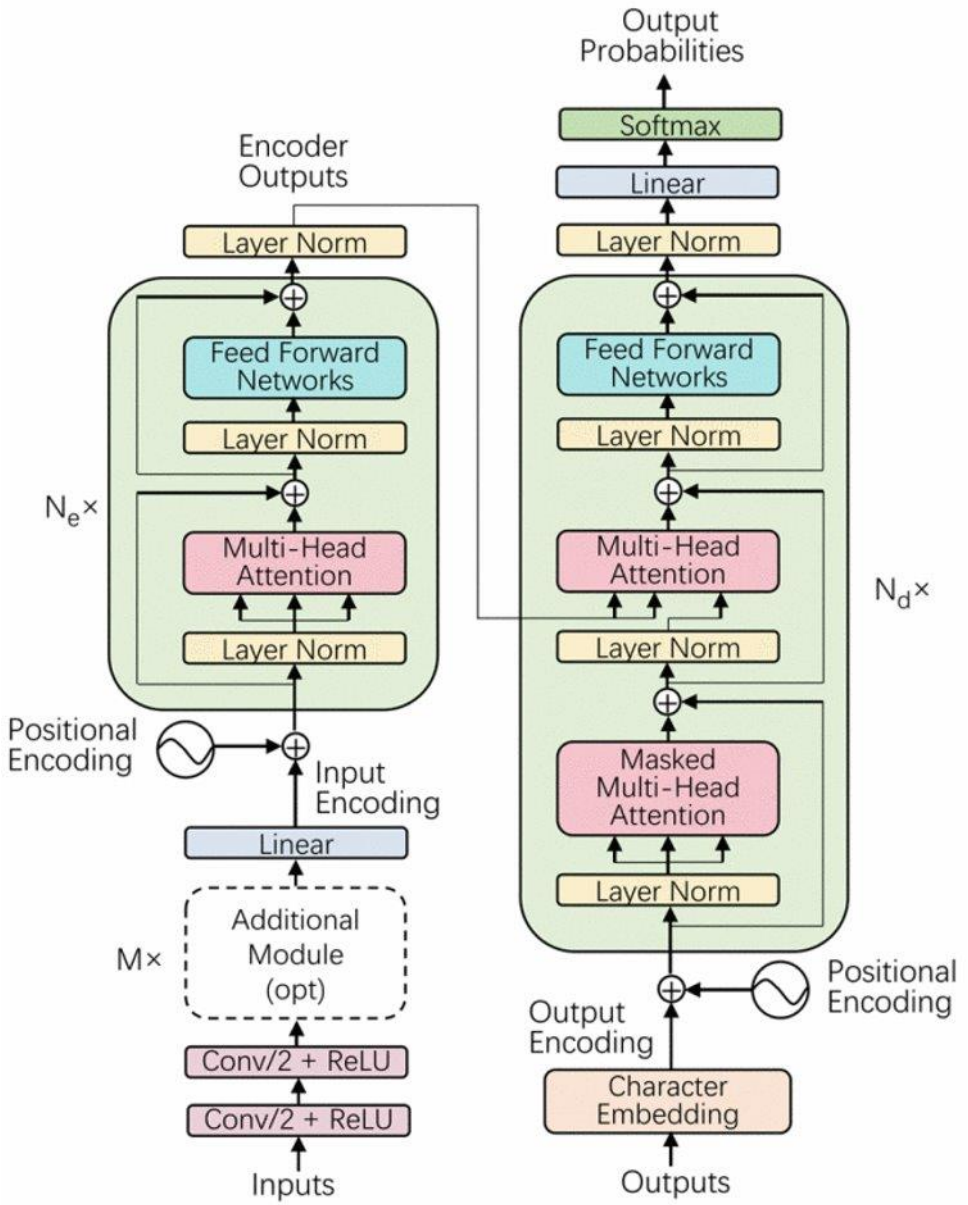
Time Series Forecasting







Speech - To - Text



Strengths

- Parallelism and efficiency.
- Scalability to large datasets.
- Universality across modalities (text, images, protein structures, etc.).

Challenges and Limitations

- Computational cost and memory usage.
- Need for large datasets and pretraining.
- Opportunities for improvement (e.g., lightweight transformers).

Curriculum

Next:

- Hands-on IV

Imprint

Topic: **Introduction to Deep Learning**
Part IV – Transformers, LLM, and other interesting architectures

Date: 2025-11-14

Author: Auliya Fitri, Sai Vemuri, Sreerag Naveenachandran

Institute: Data Science