Transformers - Unbiased Neural Network Architectures

Siyu Liu

Motivation - Recent Breakthroughs

OpenAI - General Purpose Transformers (GPT) 3

- The most intelligent AI made up entirely of transformers
- Unmatched generalisability
- The model implicitly learned to perform many language related tasks (e.g. language translation, answering questions and even coding)

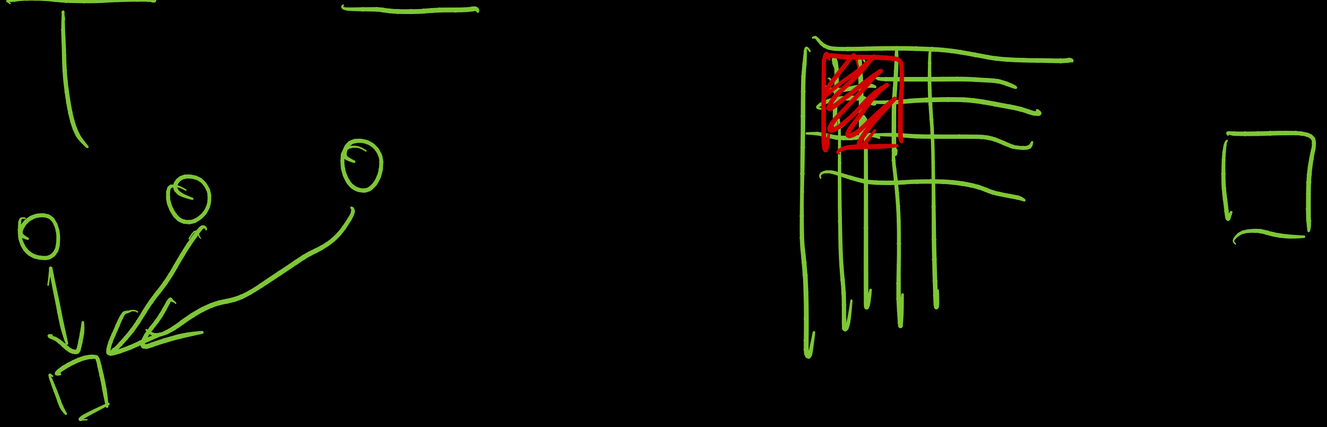
Alphafold 2

- Solve a 50-yr-old grand challenge in biology
- Can accurately predict protein structure from amino-acid sequence
- Impossible to brute force estimate for all possible protein configurations

Transformers

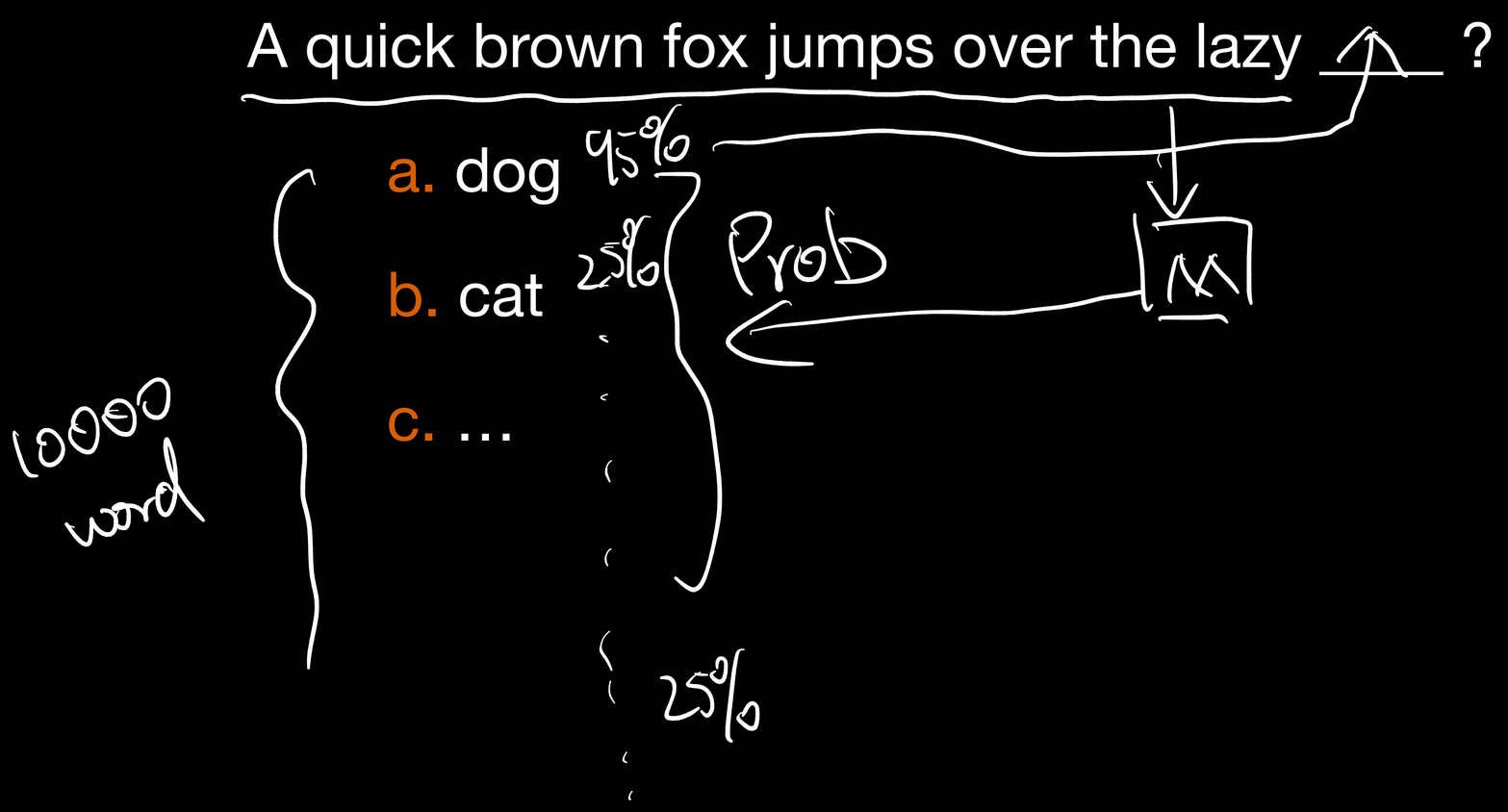
Transformers are

- A class of **general** neural network architectures
- Initially developed for natural language processing (NLP)
- Suitable for other data types due to the **lack of inductive biases** (Unlike RNNs and CNNs)



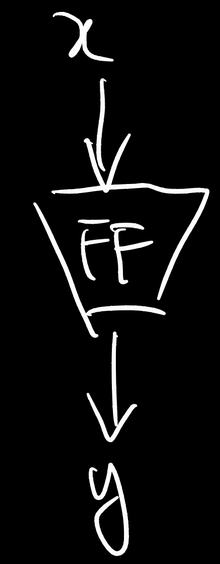
Modelling Languages - The Origin of Transformers

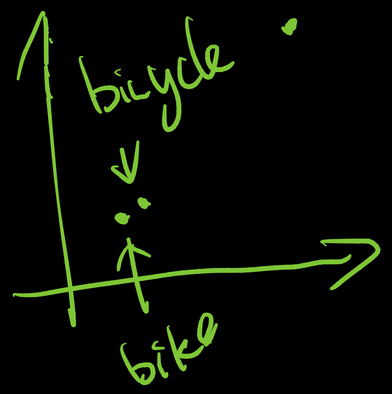
Language Model: A model that predicts the **next** word in a sentence …

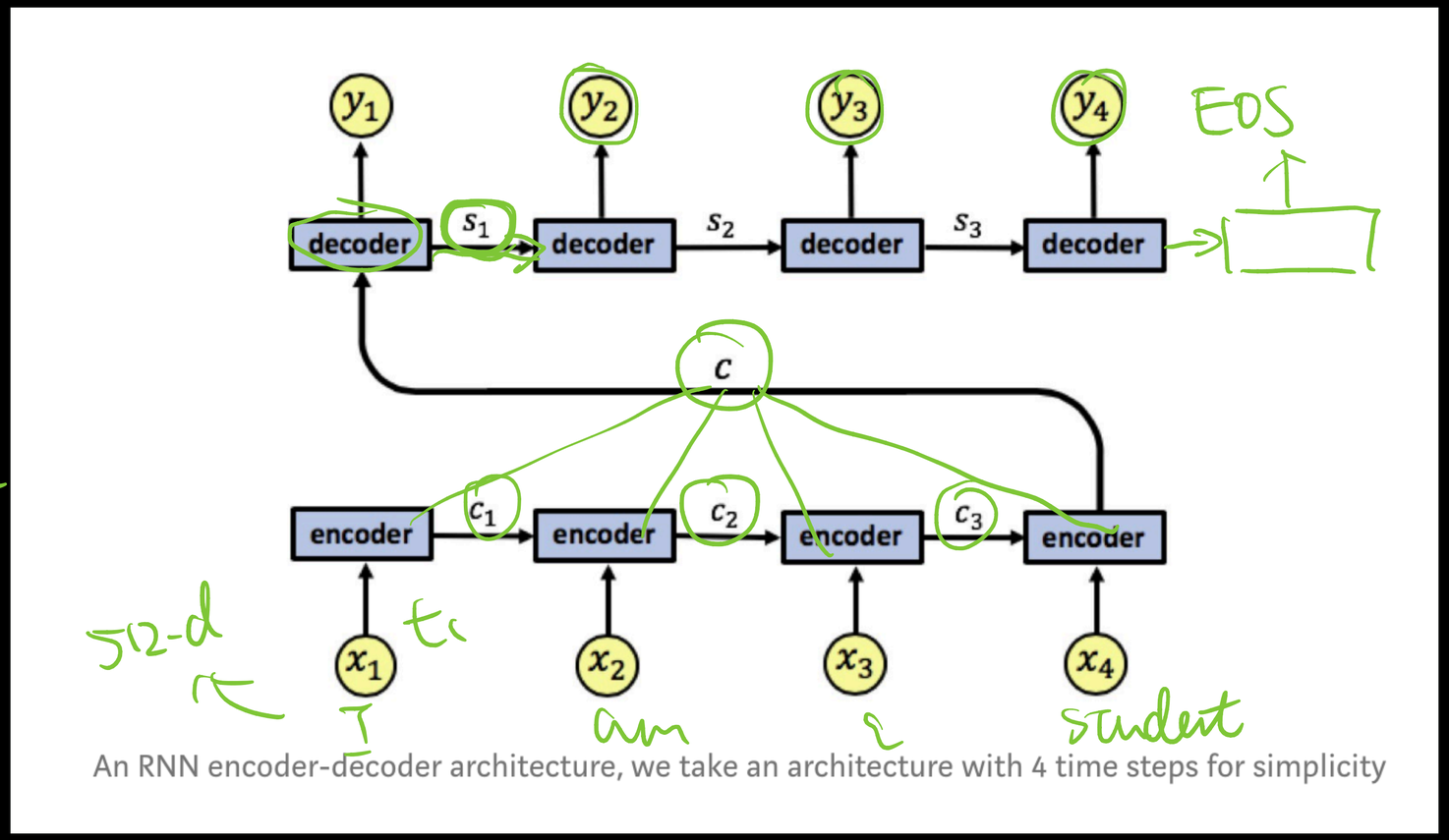


Recurrent Neural Nets - Before Transformers

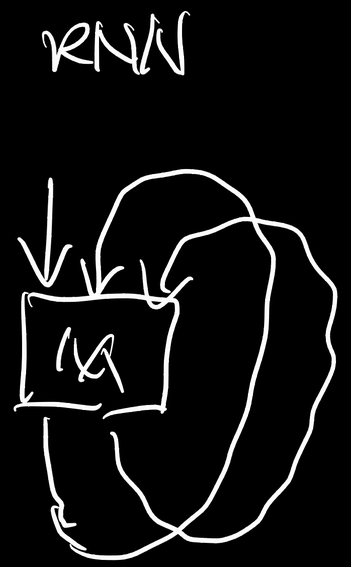
Languages can be modelled in a **recurrent** manner





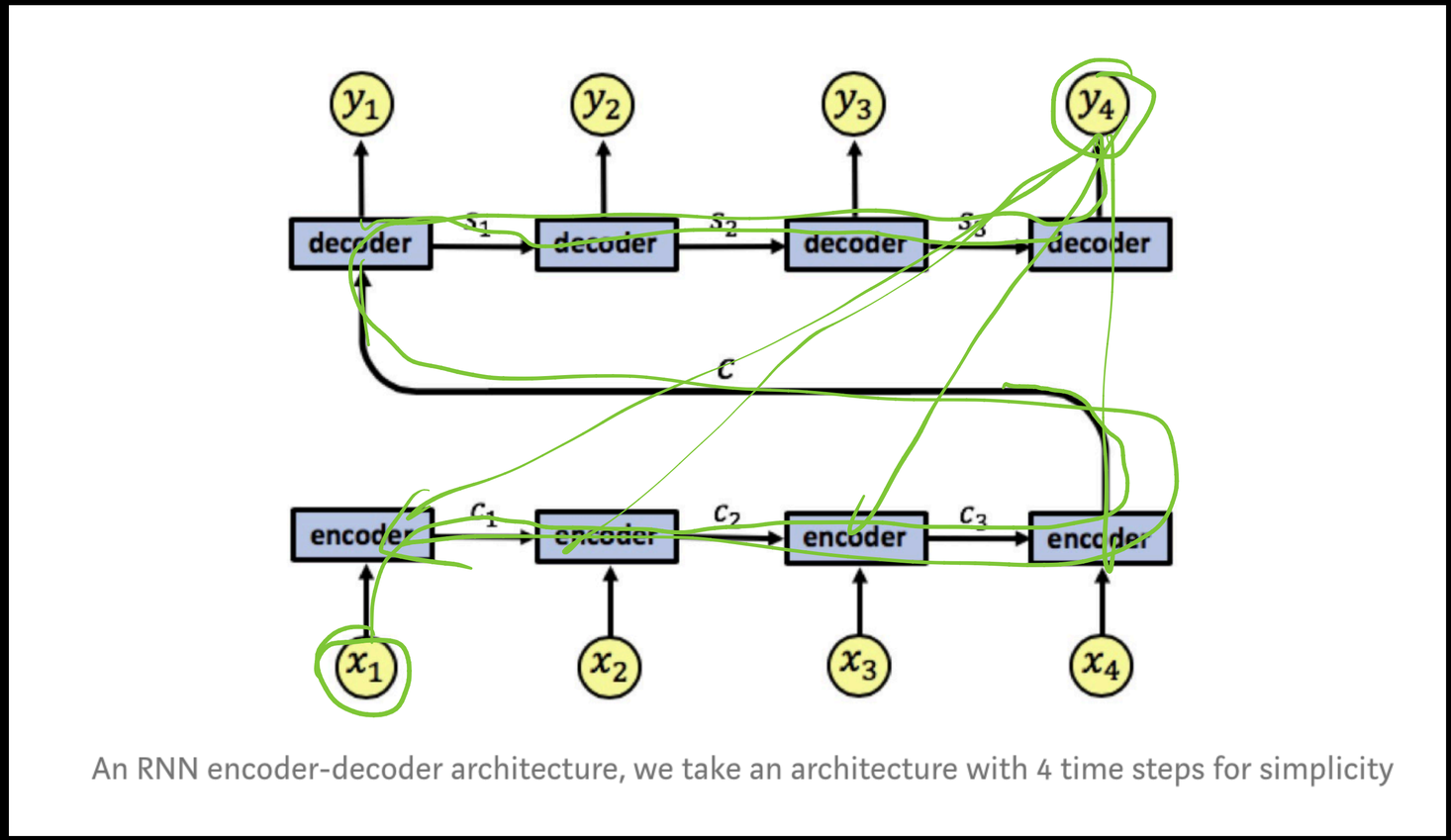


An RNN encoder-decoder architecture, we take an architecture with 4 time steps for simplicity



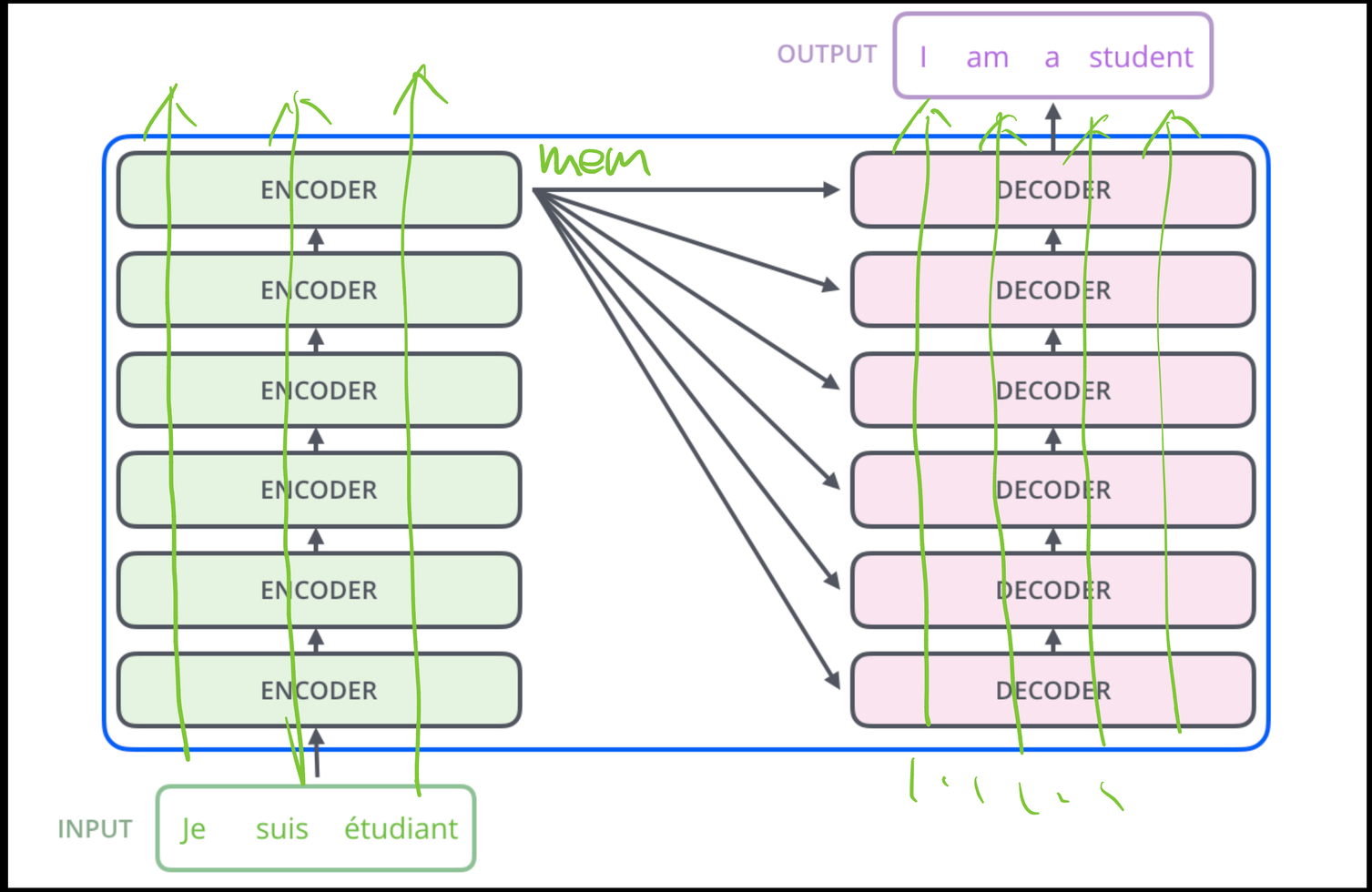
Limitations of Recurrent Neural Nets

- Not parallelisable (slow)
- Hard to maintain long-term memory (attention)

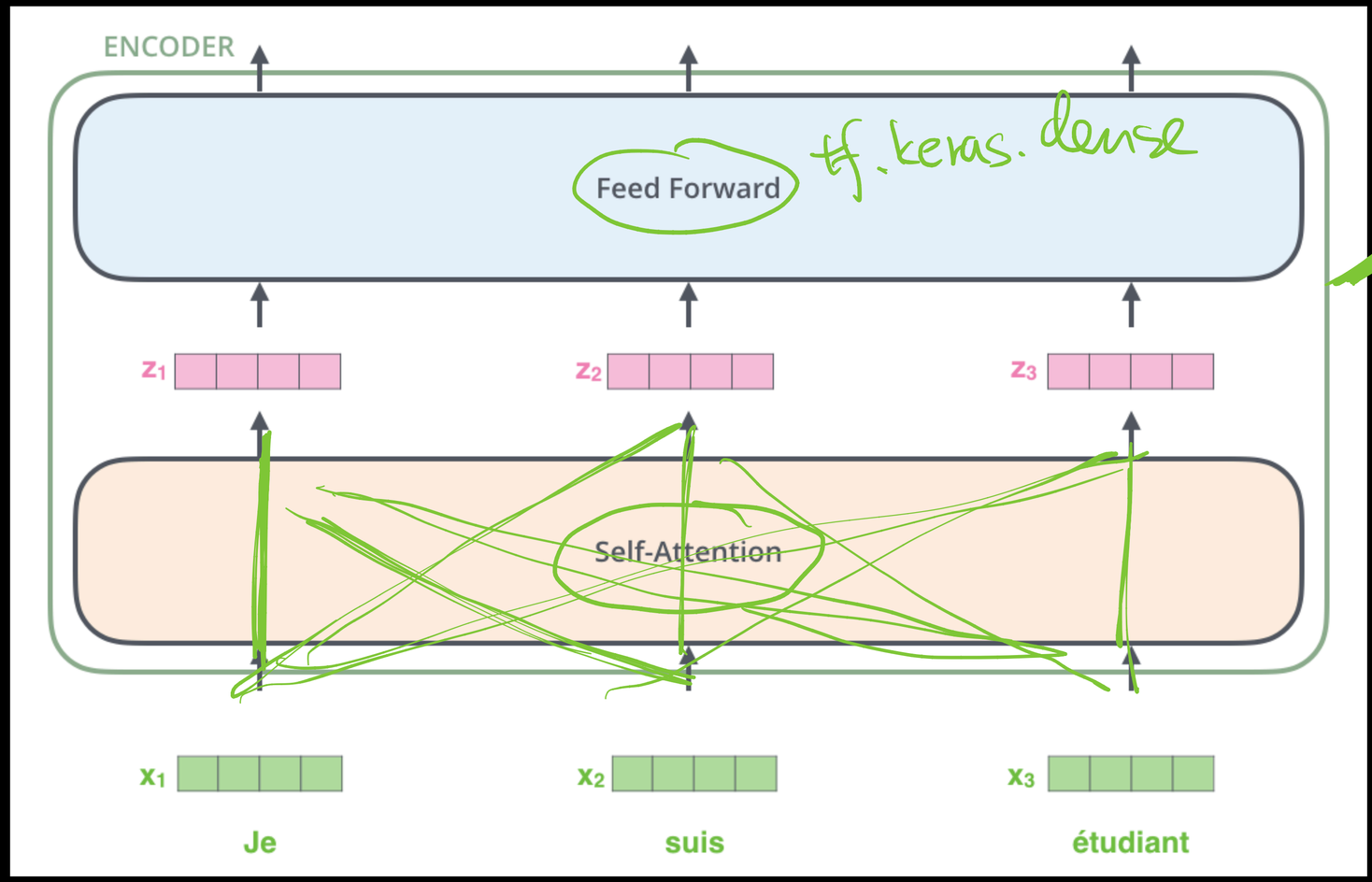


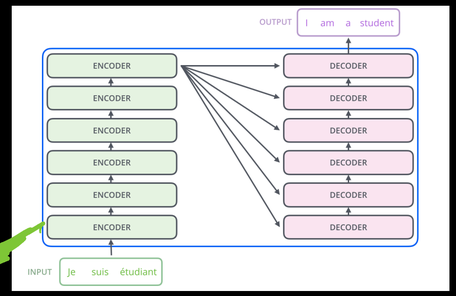
An RNN encoder-decoder architecture, we take an architecture with 4 time steps for simplicity

Transformer Architecture - Overall

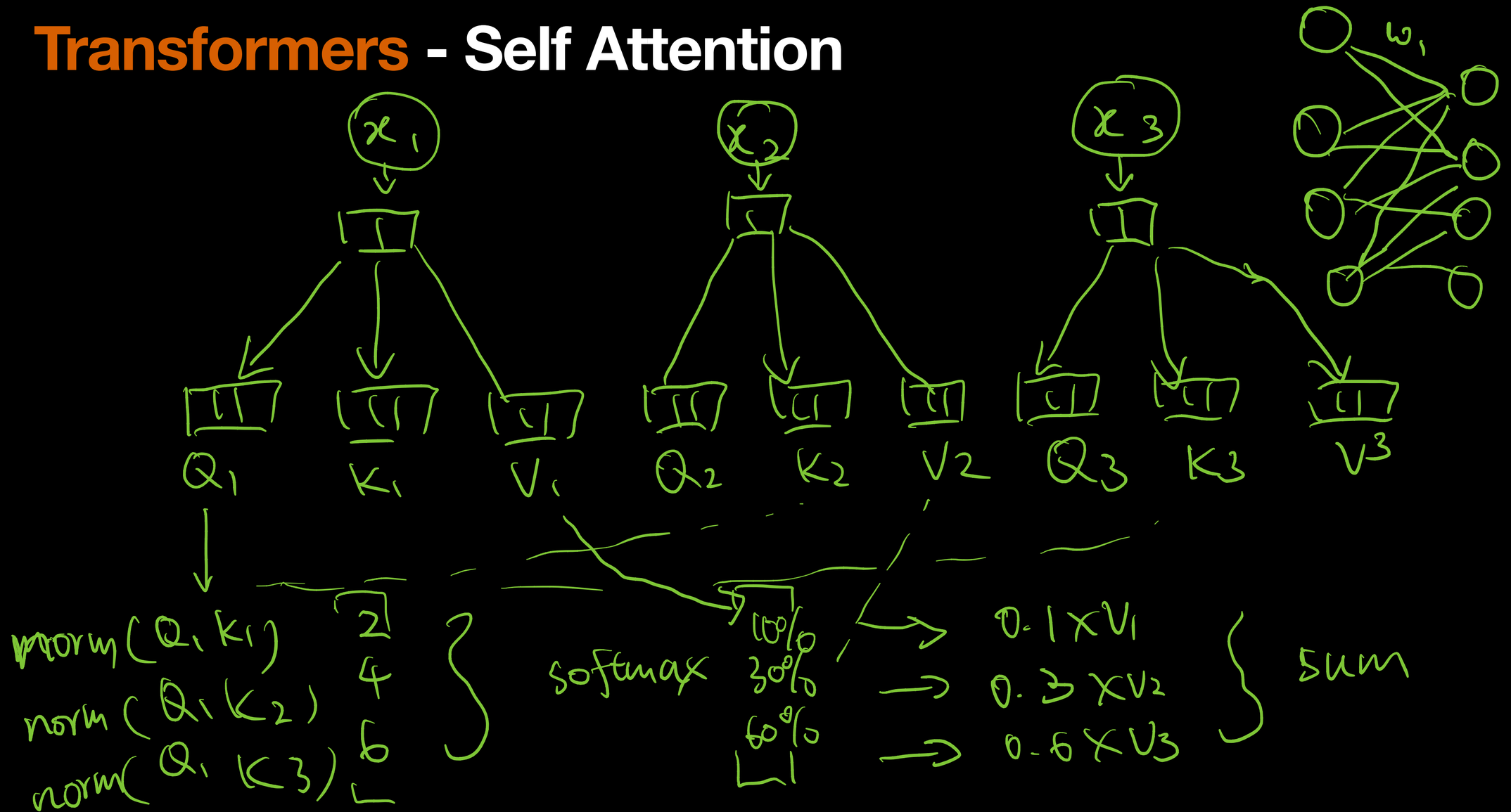


Transformers - Encoder Block

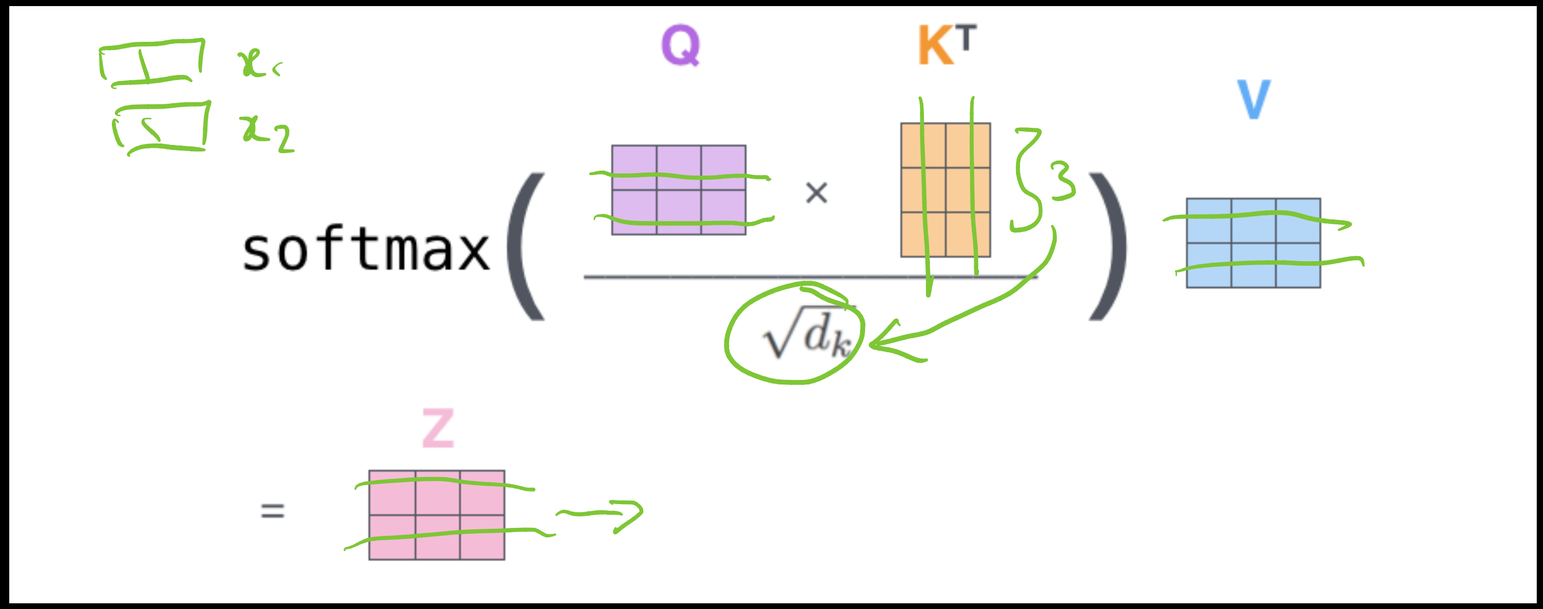




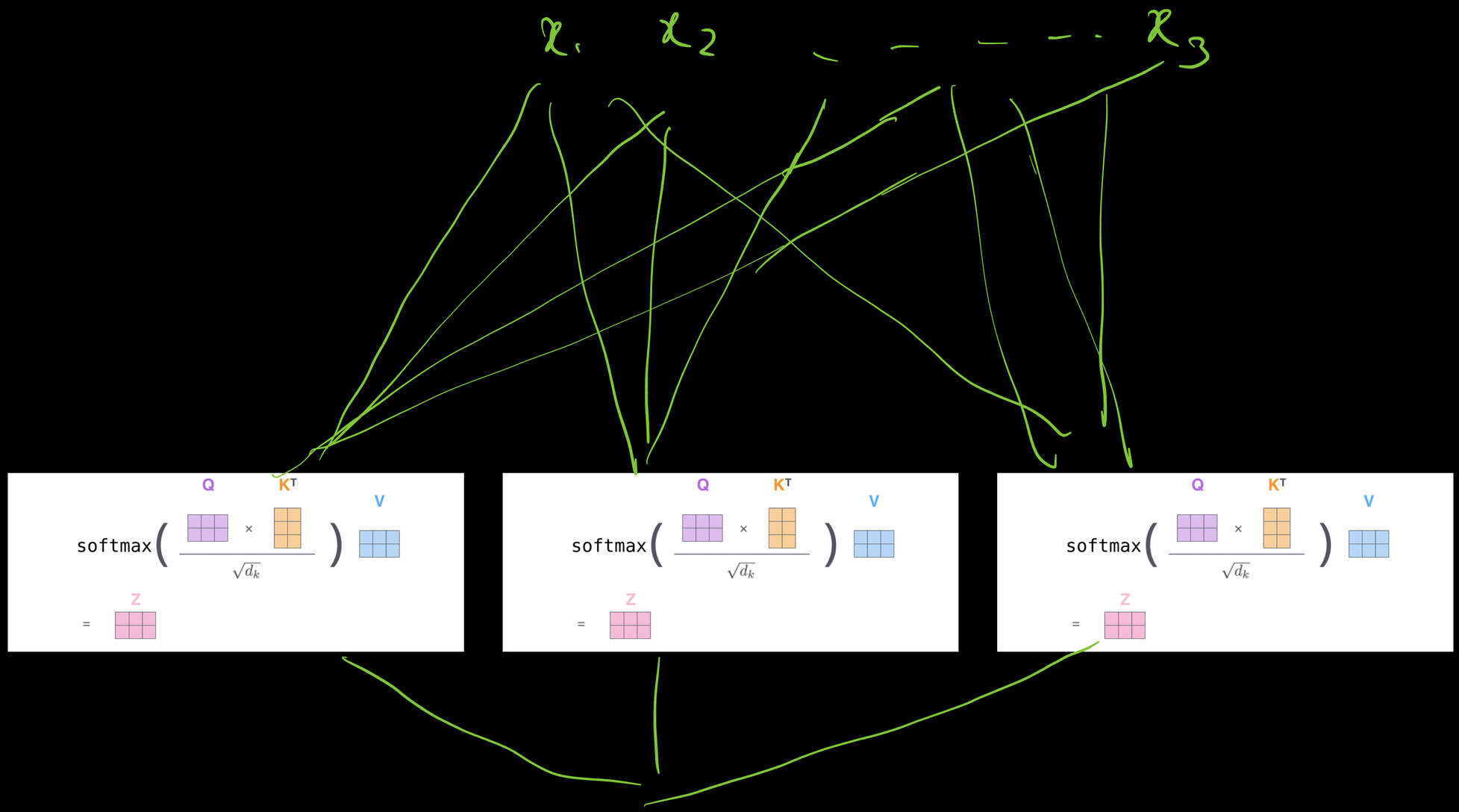
Transformers - Self Attention



Transformers - Self Attention (Vectroised)

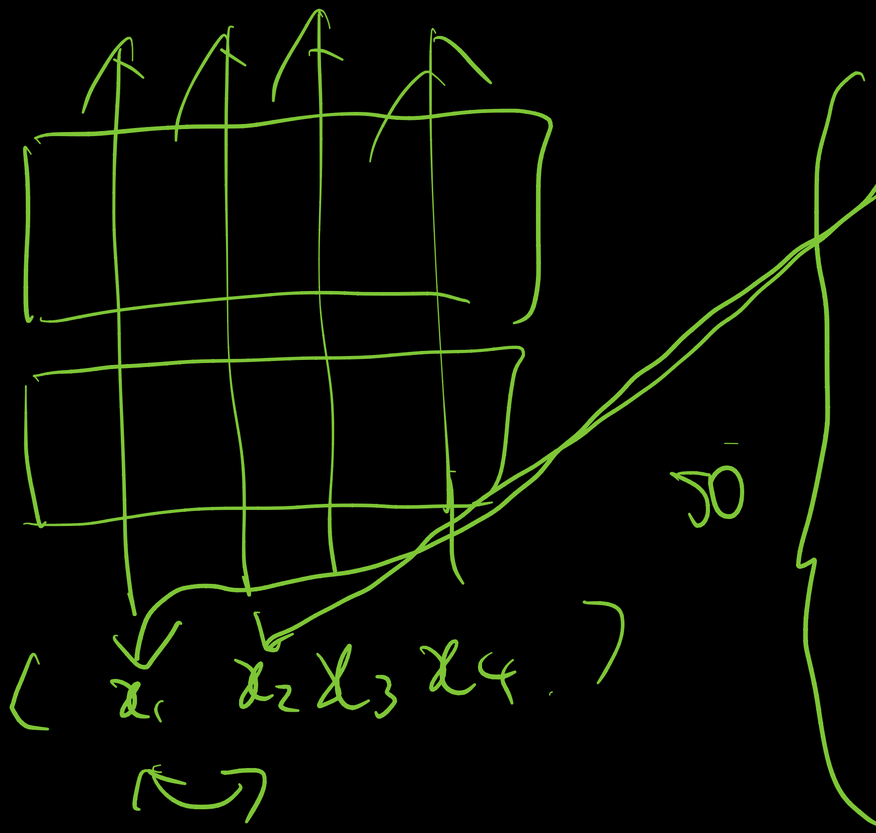


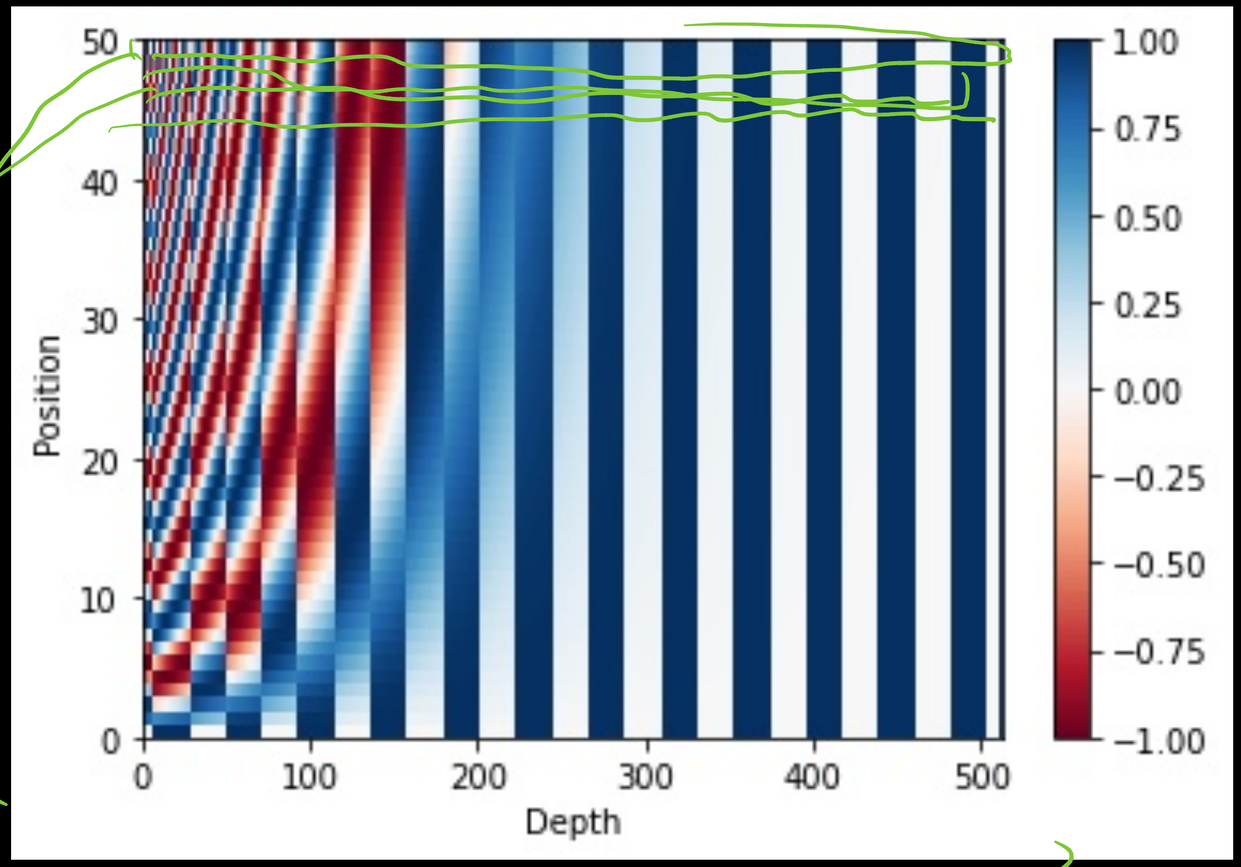
Transformers - Multi-head Self Attention



Transformers - Positional Encoding

512-d 8 heads
64-d / head

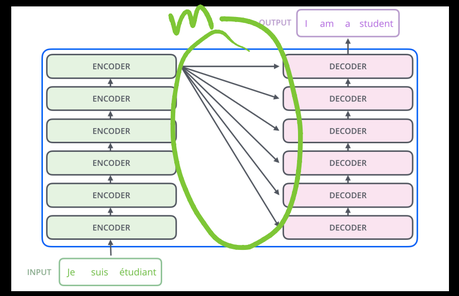


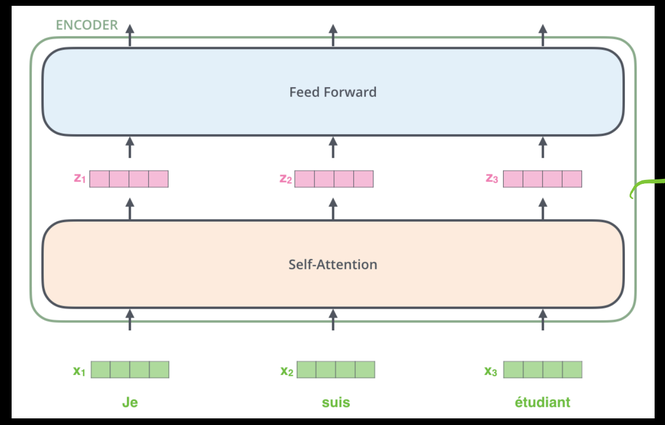


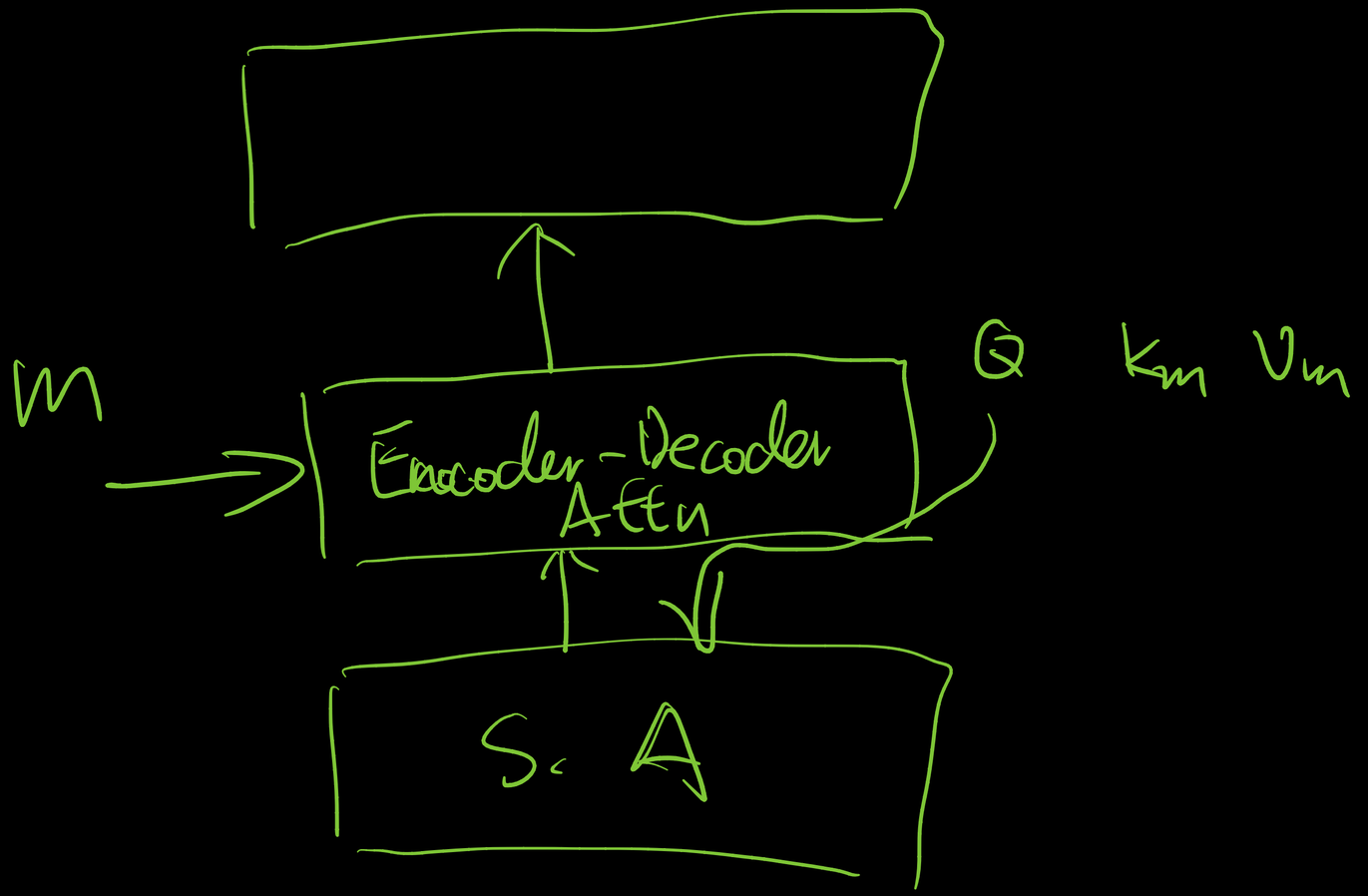
50

512

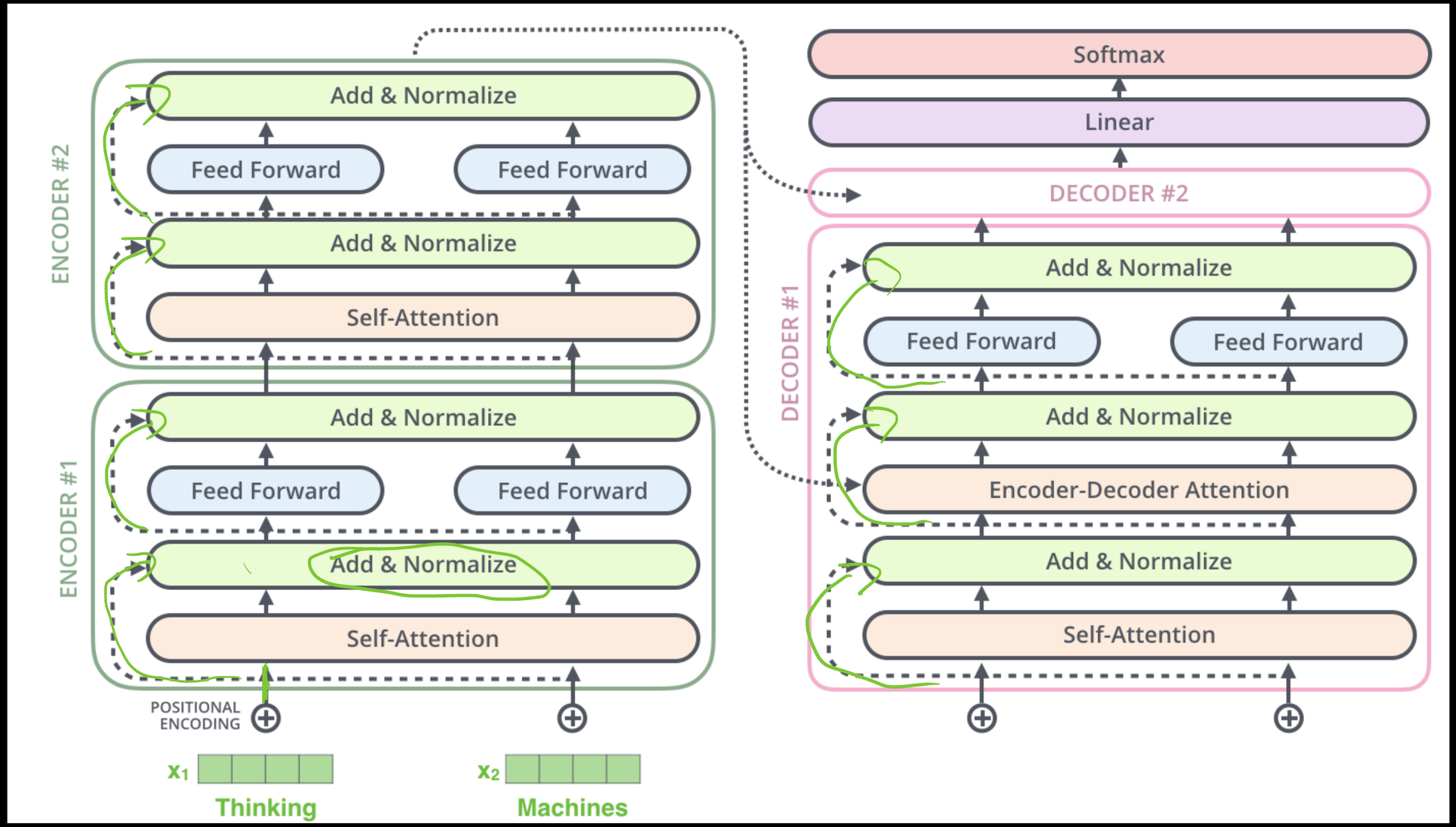
Transformers - Decoder Block





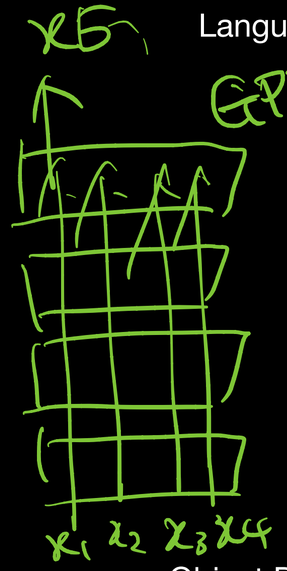


Transformers - Other Components

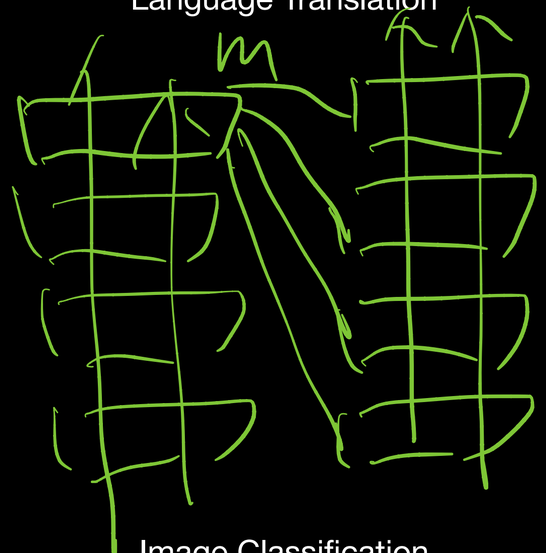


Transformers - Applications

Language Modelling



Language Translation



Object Detection (DETR)

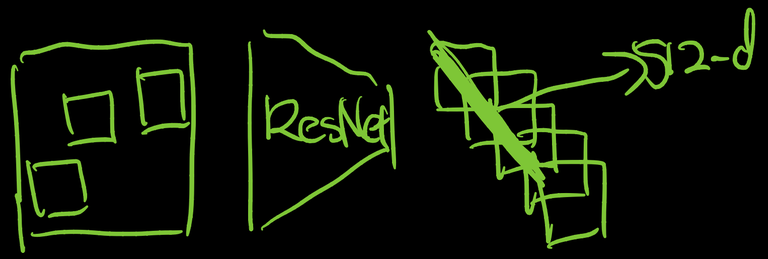
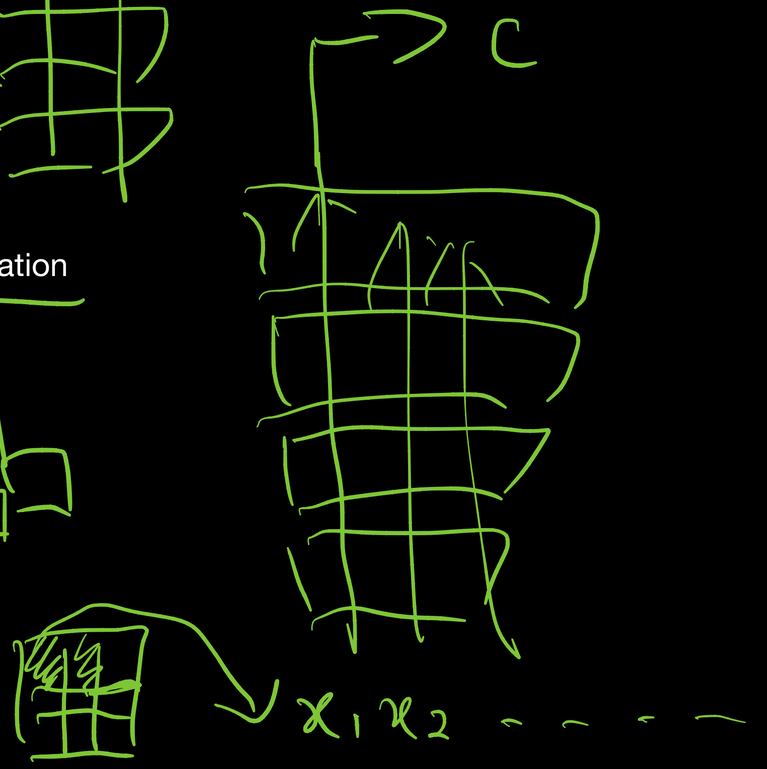


Image Classification



Transformers - Code

Self-attention is powerful but notoriously hard to implement correctly!

Luckily it is included in pytorch and is being added to keras

```
CLASS torch.nn.MultiheadAttention(embed_dim, num_heads, dropout=0.0, bias=True,
      add_bias_kv=False, add_zero_attn=False, kdim=None, vdim=None)        [SOURCE]
```

Pytorch

```
tf.keras.layers.MultiHeadAttention(
    num_heads, key_dim, value_dim=None, dropout=0.0, use_bias=True,
    output_shape=None, attention_axes=None,
    kernel_initializer='glorot_uniform',
    bias_initializer='zeros', kernel_regularizer=None,
    bias_regularizer=None, activity_regularizer=None, kernel_constraint=None,
    bias_constraint=None, **kwargs
)
```

Keras

References

- Original paper: Attention is all you need: https://arxiv.org/abs/1706.03762
- Illustrated Transformer: http://jalammar.github.io/illustrated-transformer/