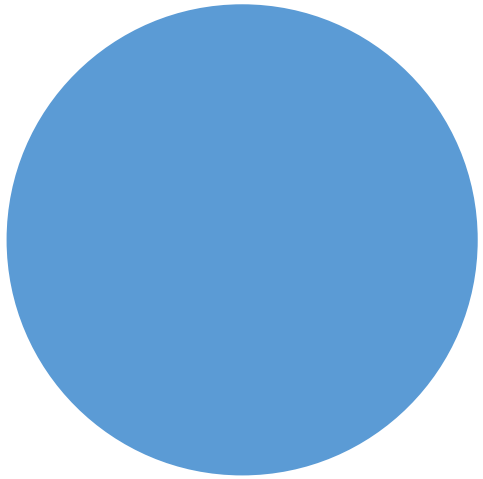
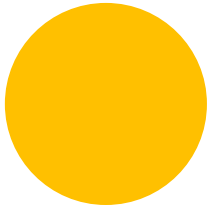



Seq2Seq Learning Advancements

Hardeep Arora

About Me

18+ years experience in Banking and Finance (MBA Finance, B.E. Computers)

Have worked as Developer, DBA, Data Scientist, and Leading Analytics functions at Bank.

Have delivered Data Warehouses and BI platforms for banks.

Have worked in Marketing Analytics

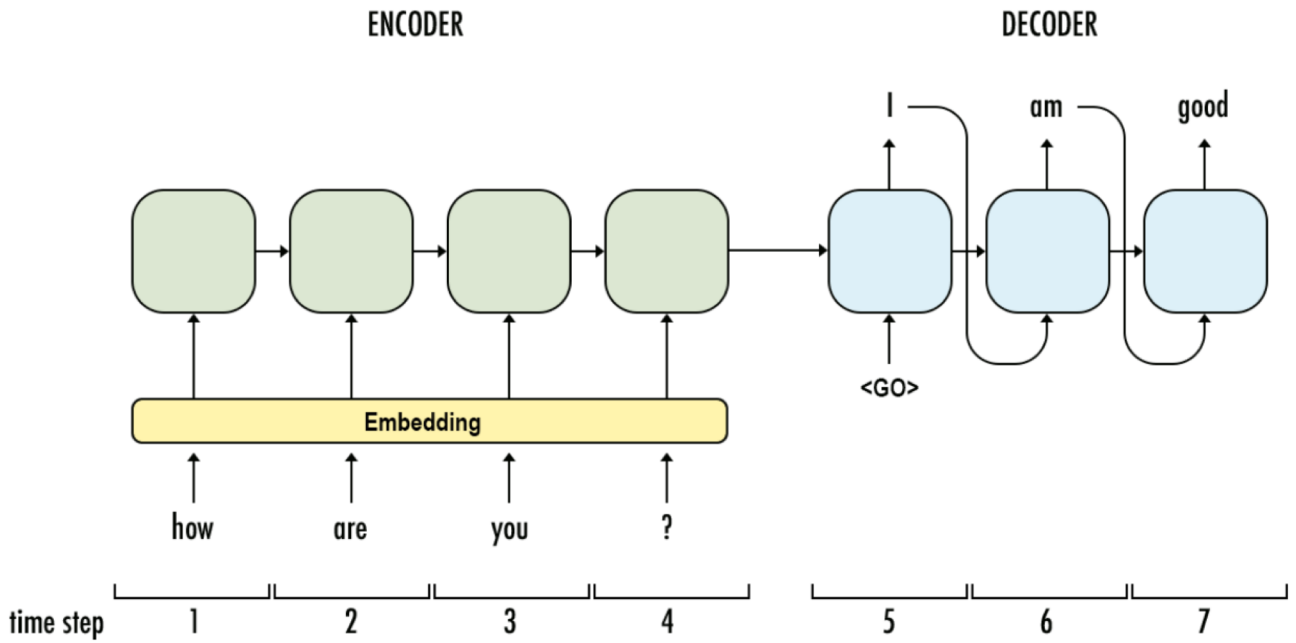
Currently Heading COE AI/Analytics @ Accenture focusing on Finance and Risk domain in ASEAN region.

Advisor for some startup's on AI Strategy

Work Life : Credit Risk and Financial Crime

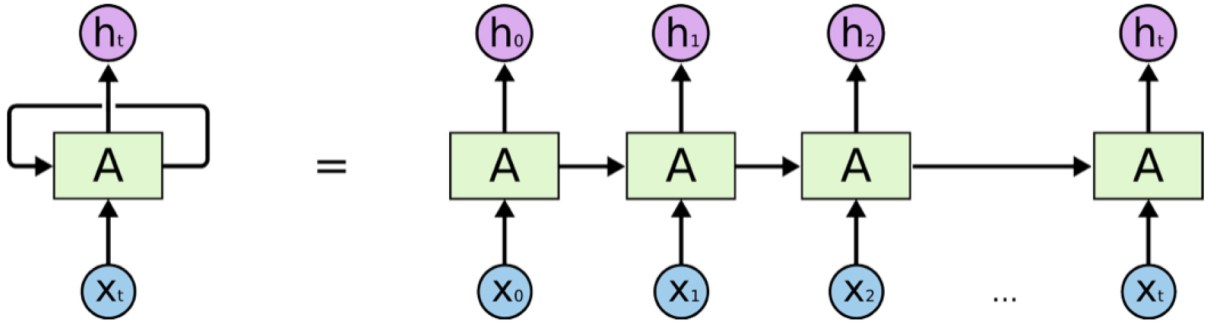
Night Life : Deep Learning, Blockchain, AGI, Kaggle

Seq2Seq


ENCODER
DECODER
I
am
good
<GO>
Embedding
how
are
you
?
time step
1
2
3
4
5
6
7

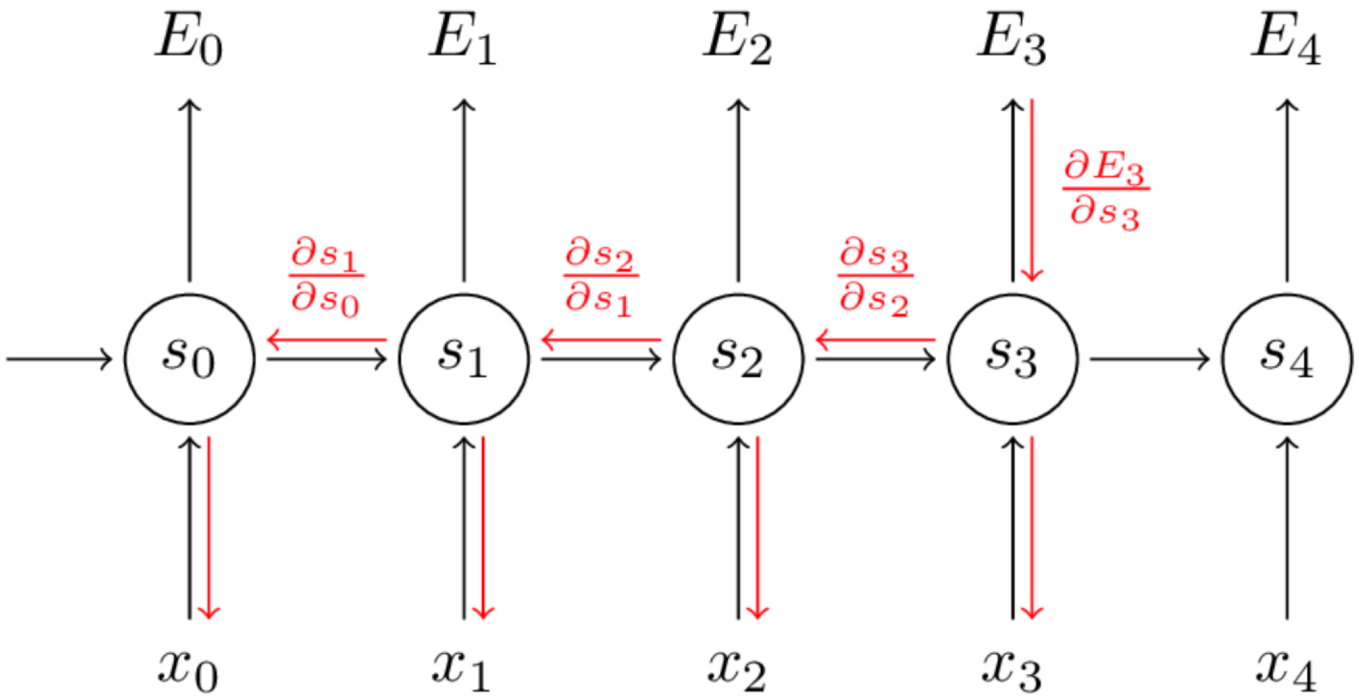
RNN

- RNN can be the encoder and decoder part of the Seq2Seq network seen in previous slide
- Can carry short-term context over. E.g. “the clouds are in the sky,”
- But struggles with long-term context. E.g. “I grew up in France… I speak fluent French.”

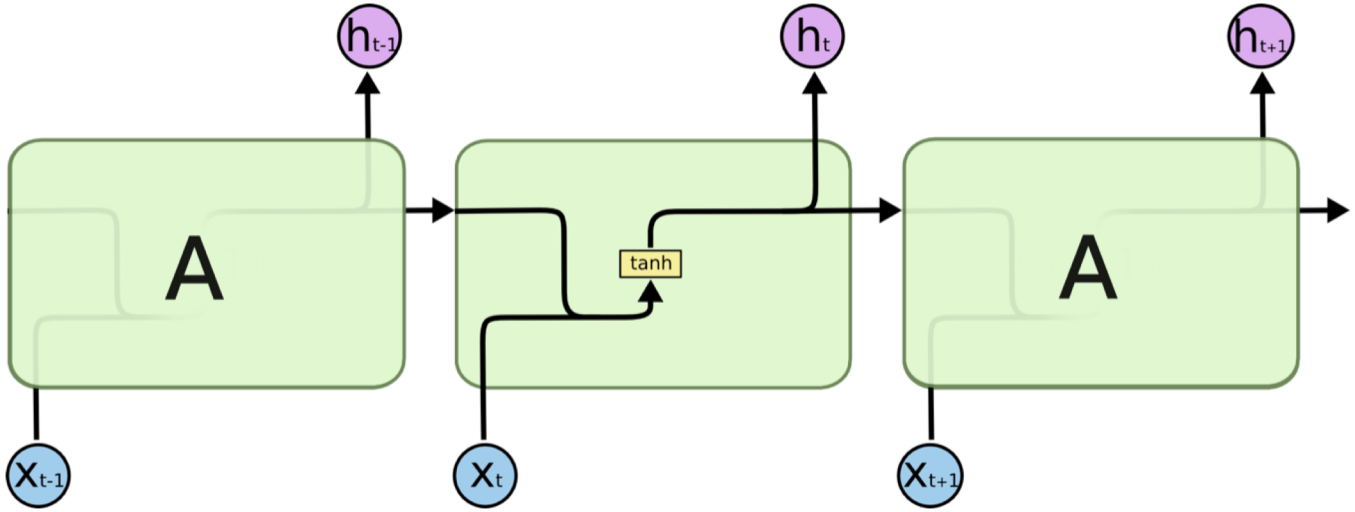


http://colah.github.io/posts/2015-08-Understanding-LSTMs/

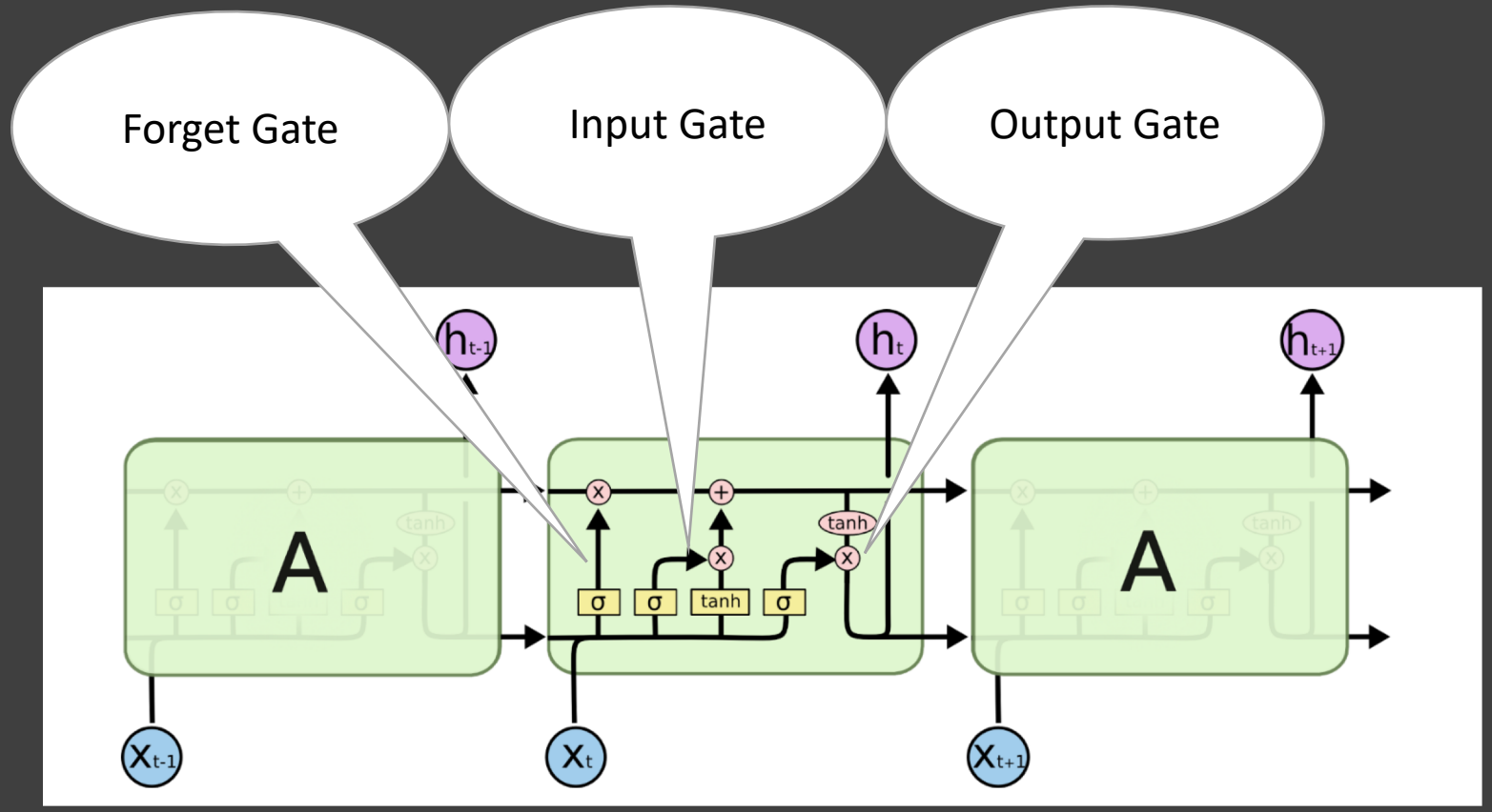
RNN Vanishing Gradients



RNN



LSTM



To the rescue, came the LSTM module, which today can be seen as multiple switch gates, and a bit like ResNet it can bypass units and thus remember for longer time steps. LSTM and GRU and derivatives are able to learn a lot of longer term information! But they can remember sequences of 100s, not 1000s or 10,000s or more.

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Applications

Speech to Text – Siri, Cortana, Google Home, Alexa

Neural Machine Translation

Image to text and text to image

Video captioning

Challenges

Compute Challenges

RNNs operate in a strict left-to-right or right-to-left order, one word at a time.

This is a less natural fit to the highly parallel GPU hardware that powers modern machine learning.

The computation cannot be fully parallelized, because each word must wait until the network is done with the previous word

CNN's to Rescue

In comparison, CNNs can compute all elements simultaneously, taking full advantage of GPU parallelism.

They therefore are computationally more efficient.

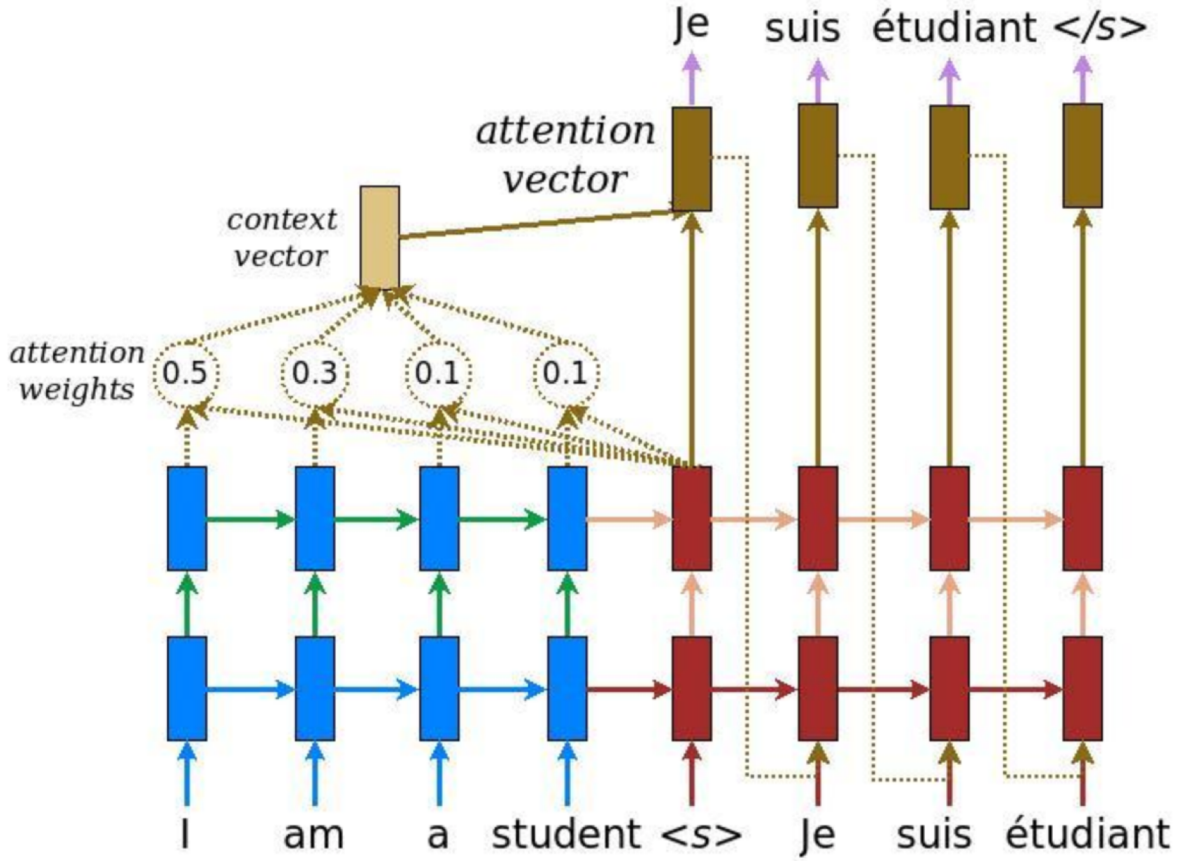
Another advantage of CNNs is that information is processed hierarchically, which makes it easier to capture complex relationships in the data.

Attention

$$\alpha_{ts} = \frac{\exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)} \quad \text{[Attention weights]}$$

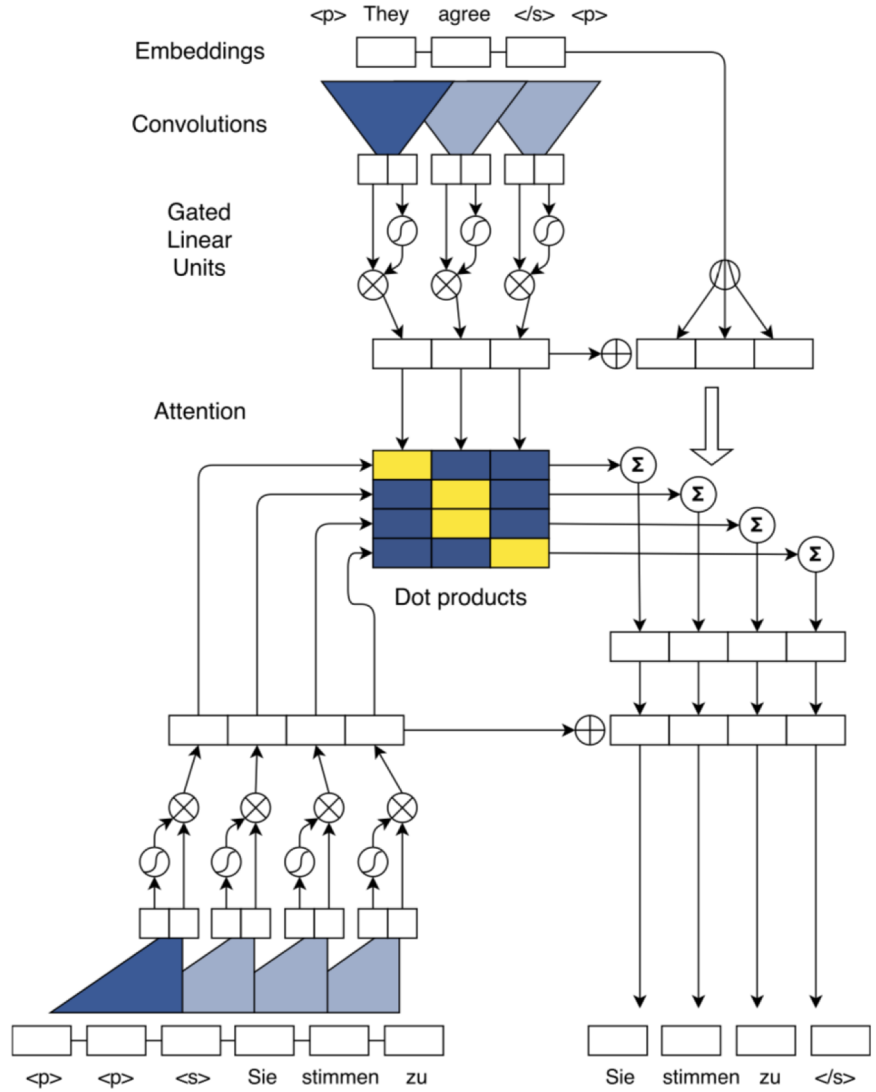
$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s \quad \text{[Context vector]}$$

$$\boldsymbol{a}_t = f(\boldsymbol{c}_t, \boldsymbol{h}_t) = \tanh(\boldsymbol{W}_c[\boldsymbol{c}_t; \boldsymbol{h}_t]) \quad \text{[Attention vector]}$$



Convolutional Seq2Seq

- Top is the encoder
- Bottom left is decoder
- Attention is dot product of encoder and decoder

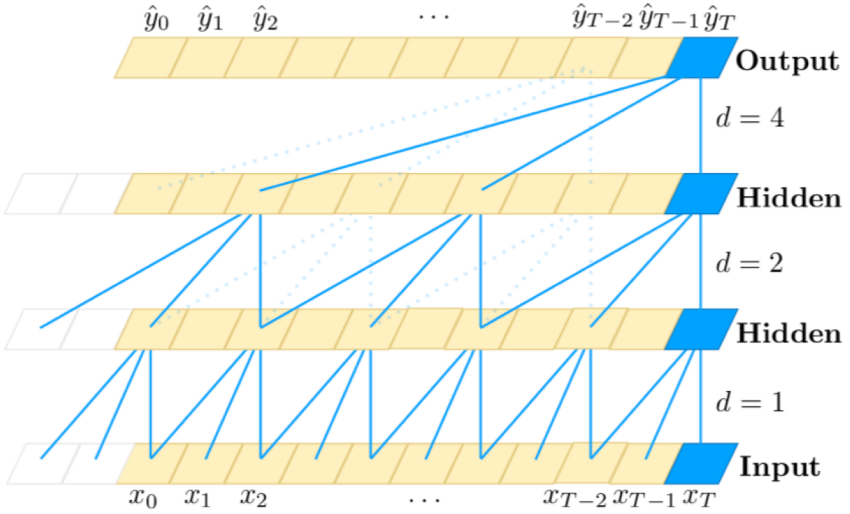


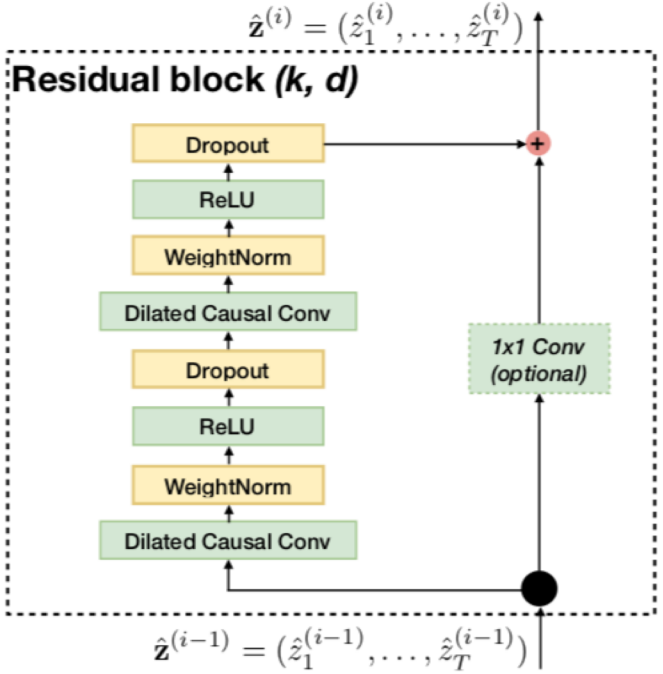
https://arxiv.org/pdf/1705.03122.pdf

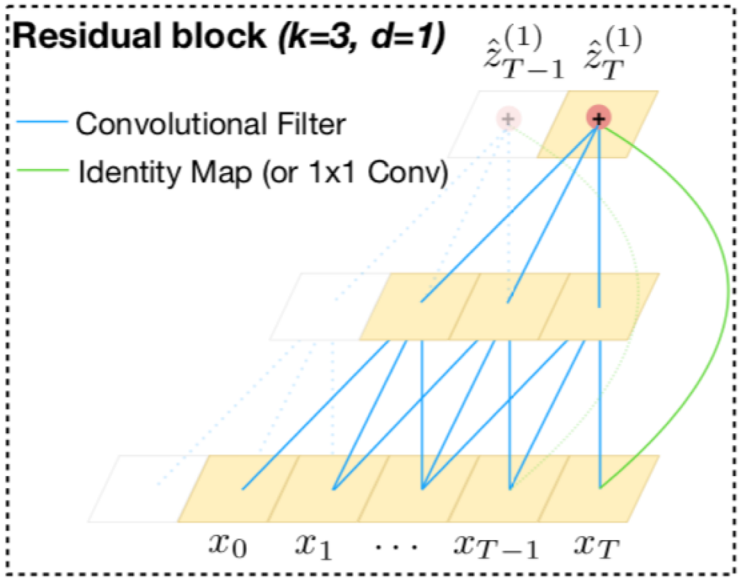
NMT – Application of CNN (Facebook)

Temporal convolutional network (TCN)

- TCNs convincingly outperform baseline recurrent architectures across a broad range of sequence modelling tasks.
- TCN's also exhibit substantially longer memory, and are thus more suitable for domains where a long history is required.

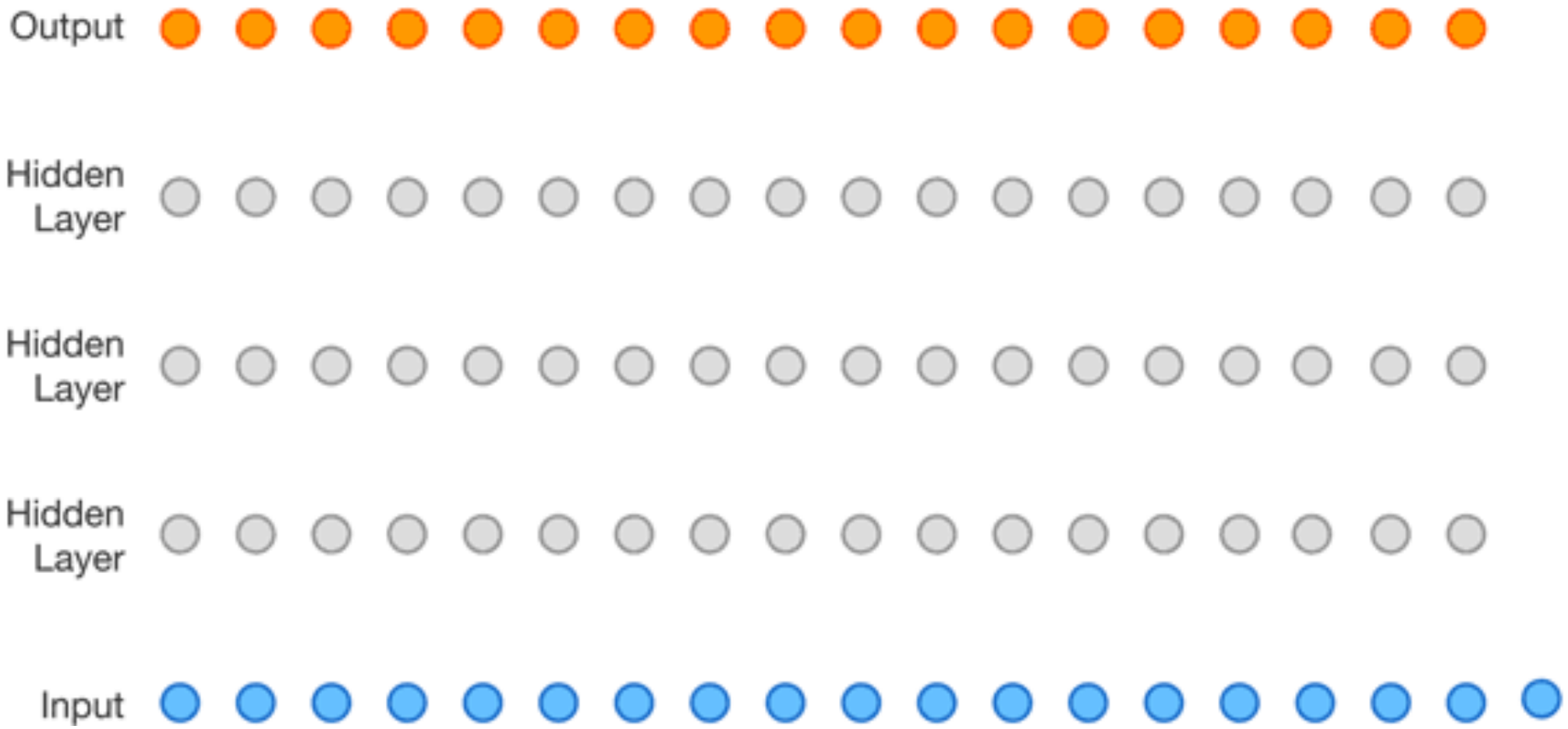




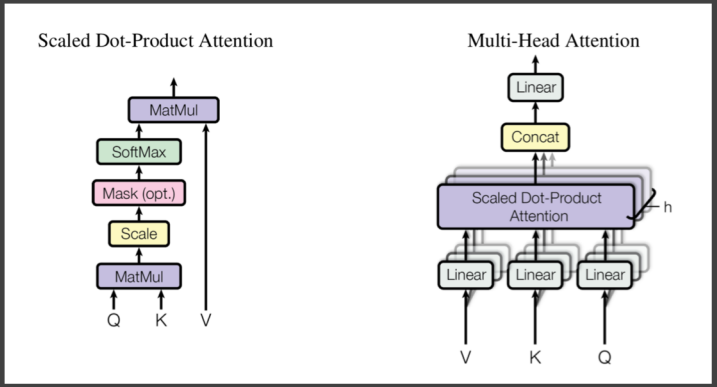


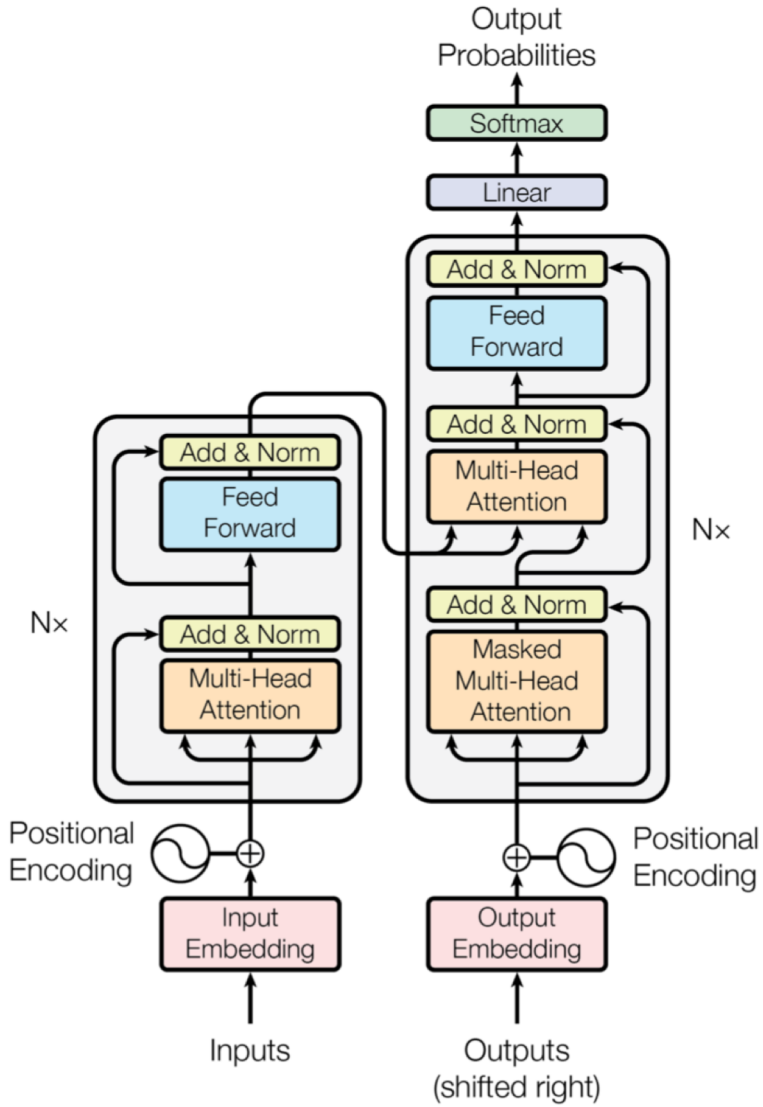
https://arxiv.org/pdf/1803.01271.pdf

WaveNet2 – Using Dilated CNN's



Attention is all you need

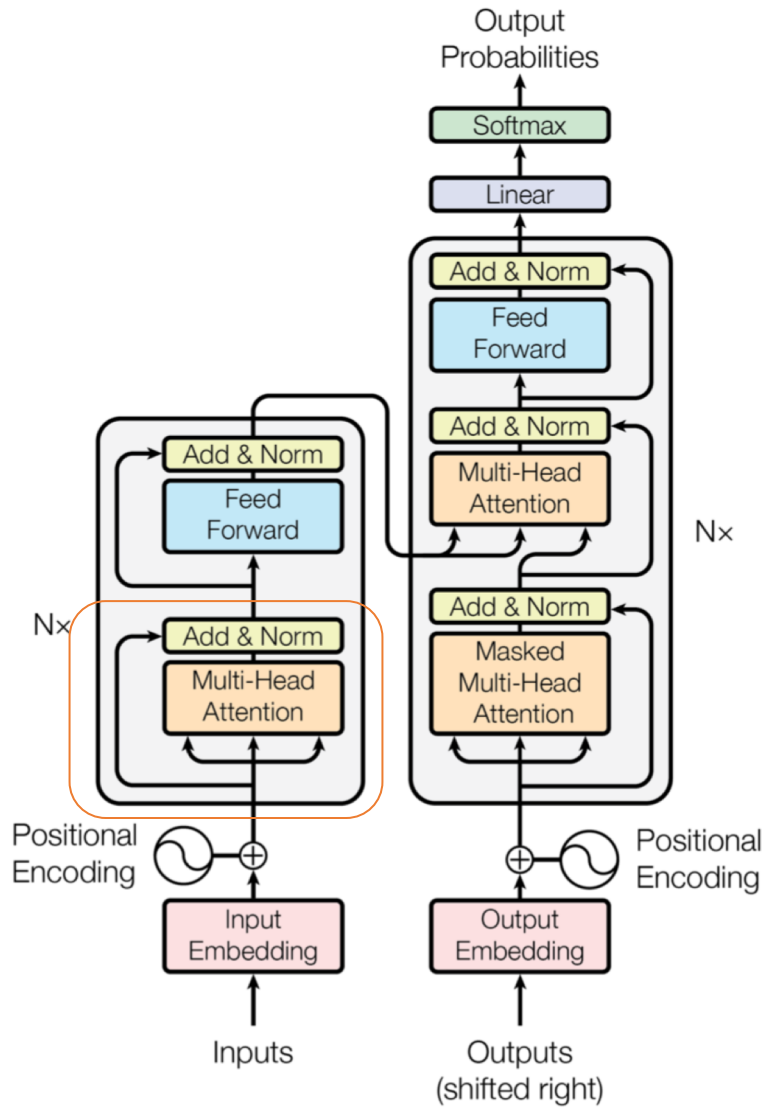




https://arxiv.org/pdf/1706.03762.pdf

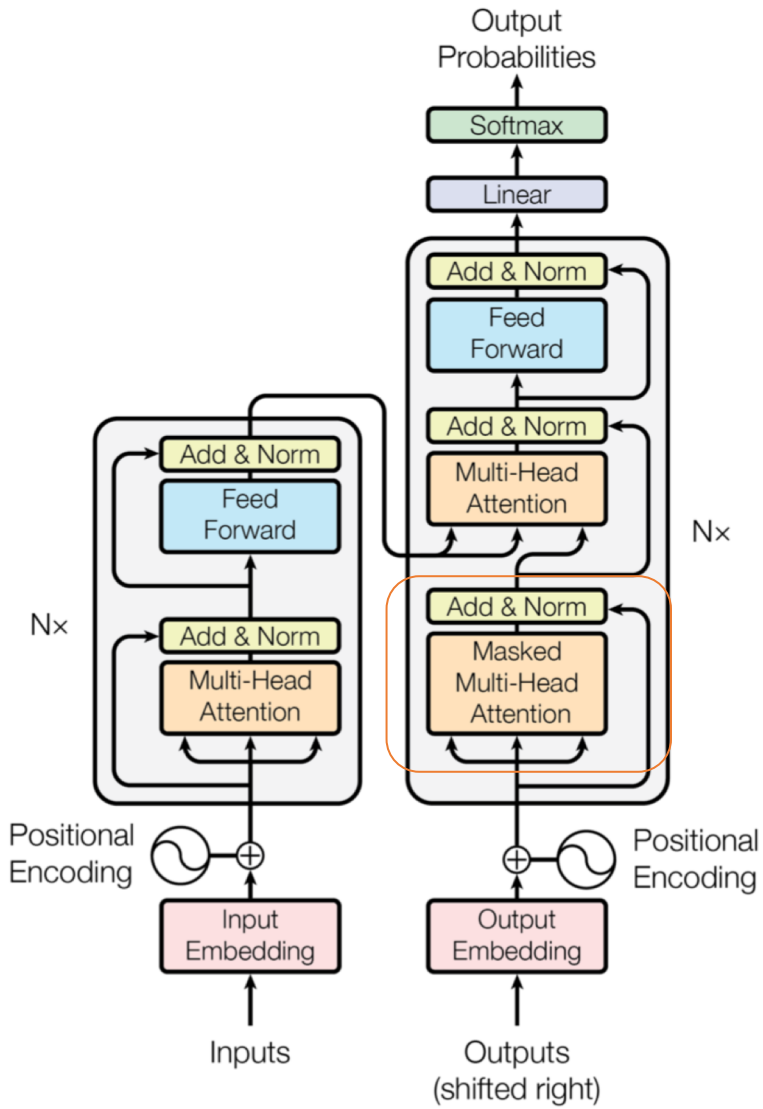
Attention is all you need

- Encoder Self-attention: All of the keys, values and queries come from the same place.
- Each position in the encoder can attend to all positions in the previous layer of the encoder.



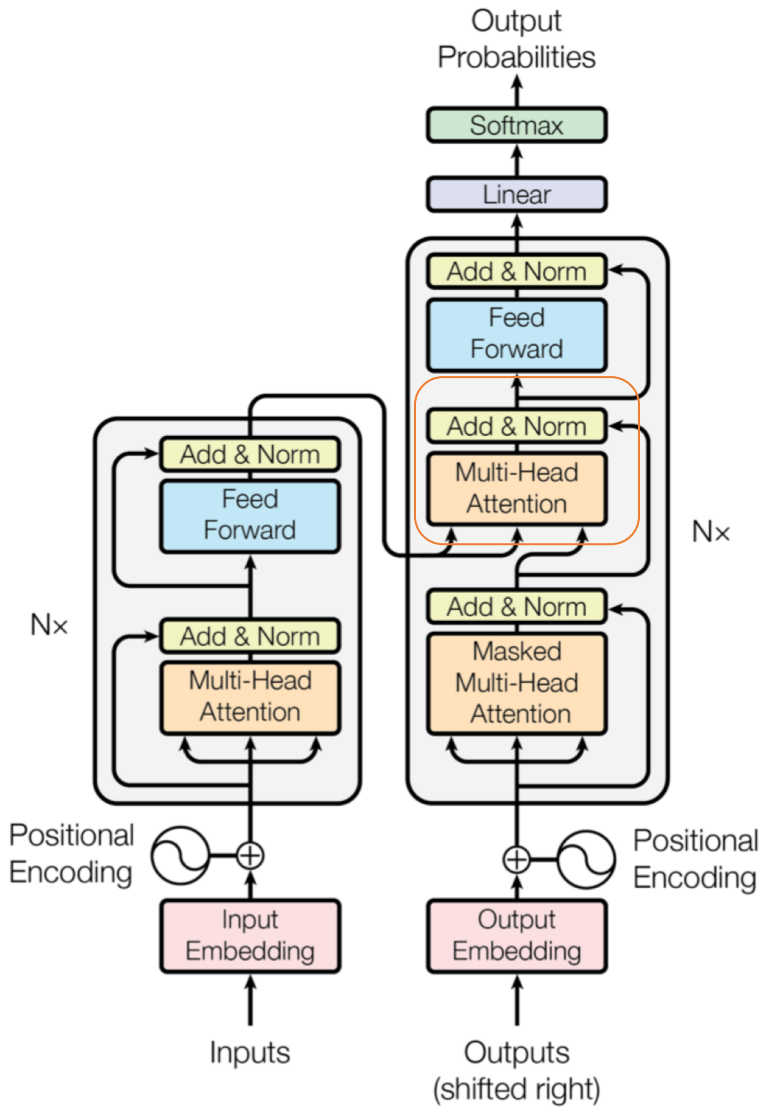
Attention is all you need

- Decoder – Self Attention: Each position in the decoder attend to all positions in the decoder up to and including that position.
- Masking ($-\infty$) - We need to prevent leftward information flow in the decoder to preserve the auto-regressive property.



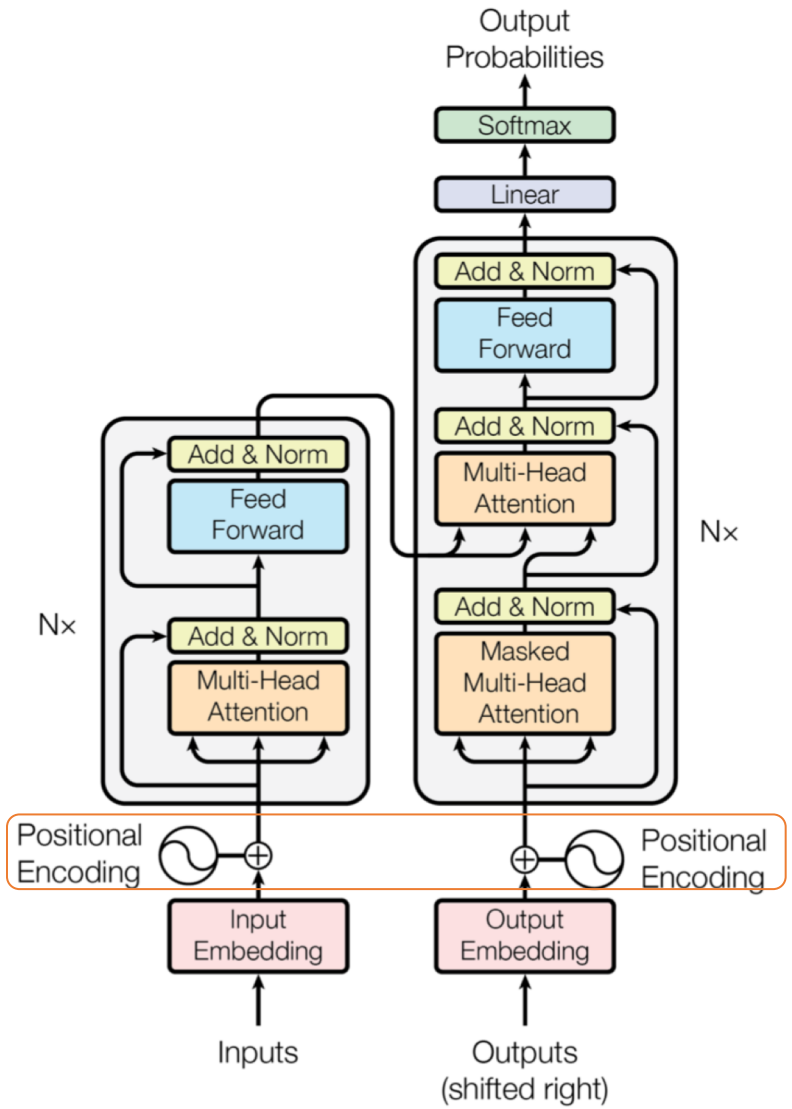
Attention is all you need

- Key & Value come from Encoder
- Query comes from Decoder
- This allows every position in the decoder to attend over all positions in the input sequence.



Attention is all you need

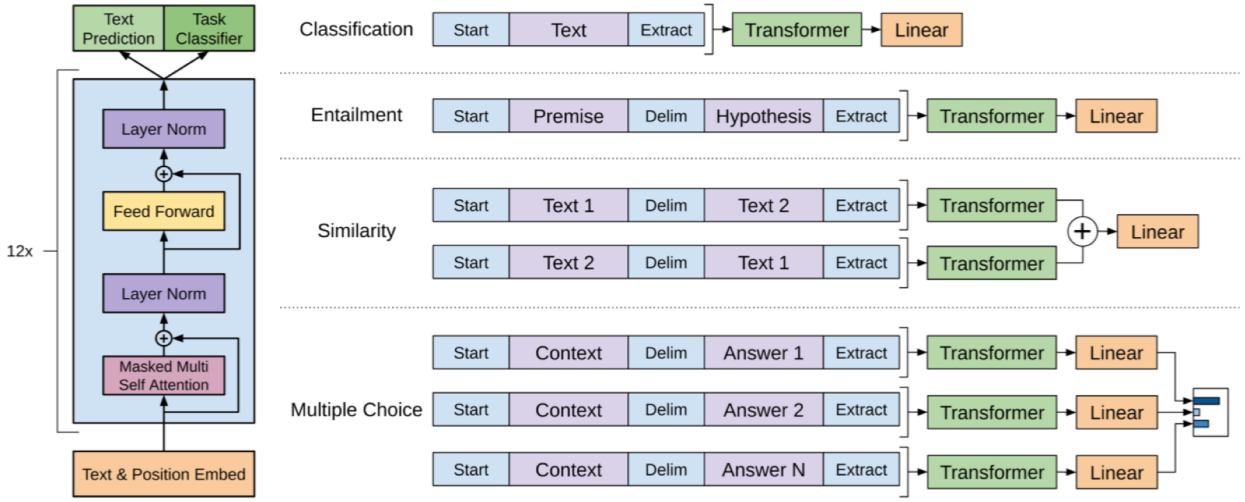
- Since we don’t have time steps how do we get positional information
- Positional Encodings – Learned or Fixed (Sin Function etc)



OpenAI
Improving Language Understanding by Generative Pre-Training

Stage 1 - Unsupervised

We use a multi-layer *Transformer decoder* for the language model, which is a variant of the transformer. This model applies a multi-headed self-attention operation over the input context tokens followed by position-wise feedforward layers to produce an output distribution over target tokens



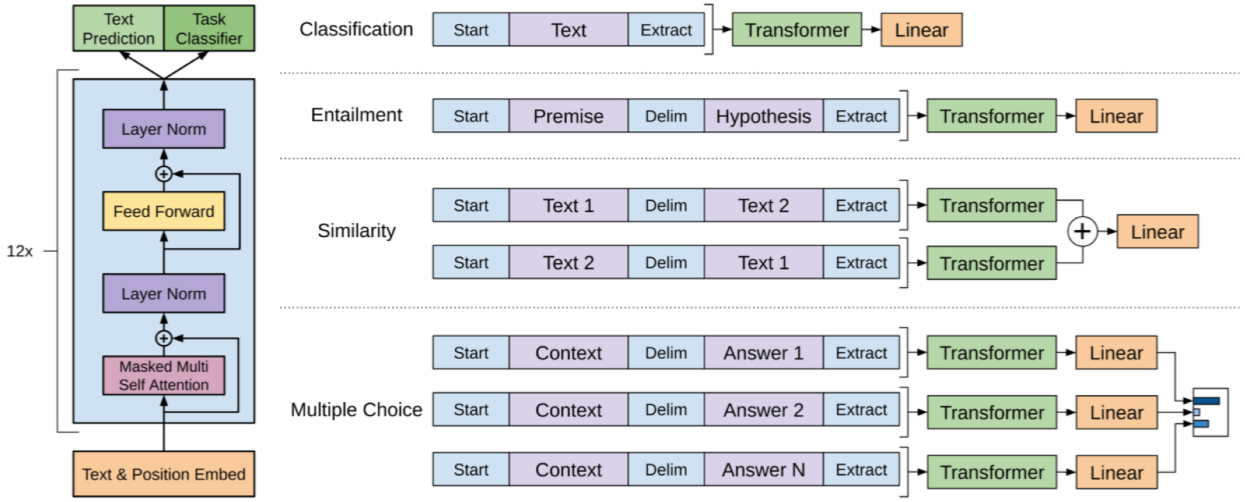
https://blog.openai.com/language-unsupervised/
https://github.com/openai/finetune-transformer-lm

OpenAI
Improving Language Understanding by Generative Pre-Training

Stage 2 - Supervised

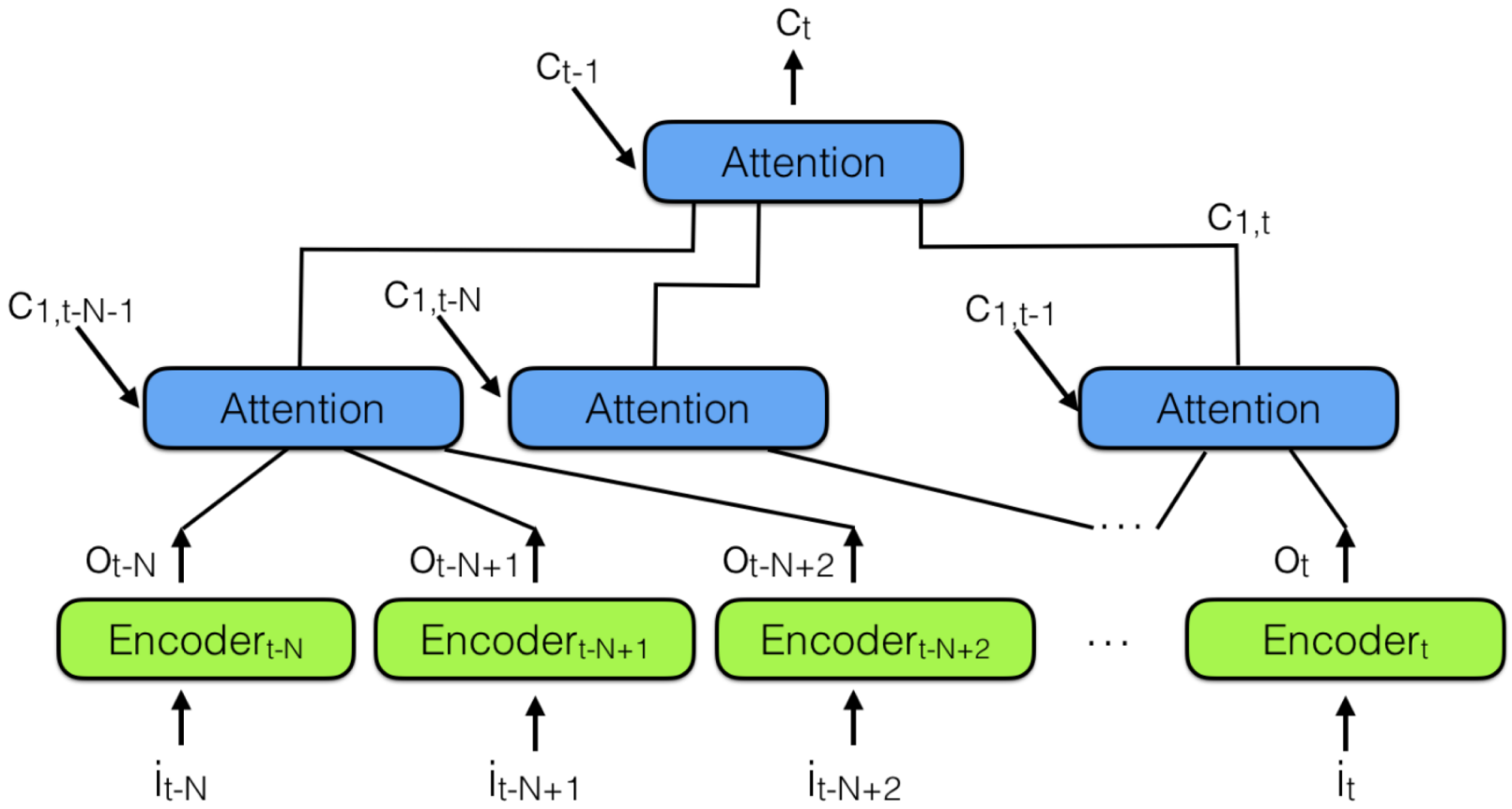
The fine tune the trained model on few label tasks to get task specific outputs.

They used the knowledge gained in the Transformer to transfer it on specific tasks, similar to transfer learning

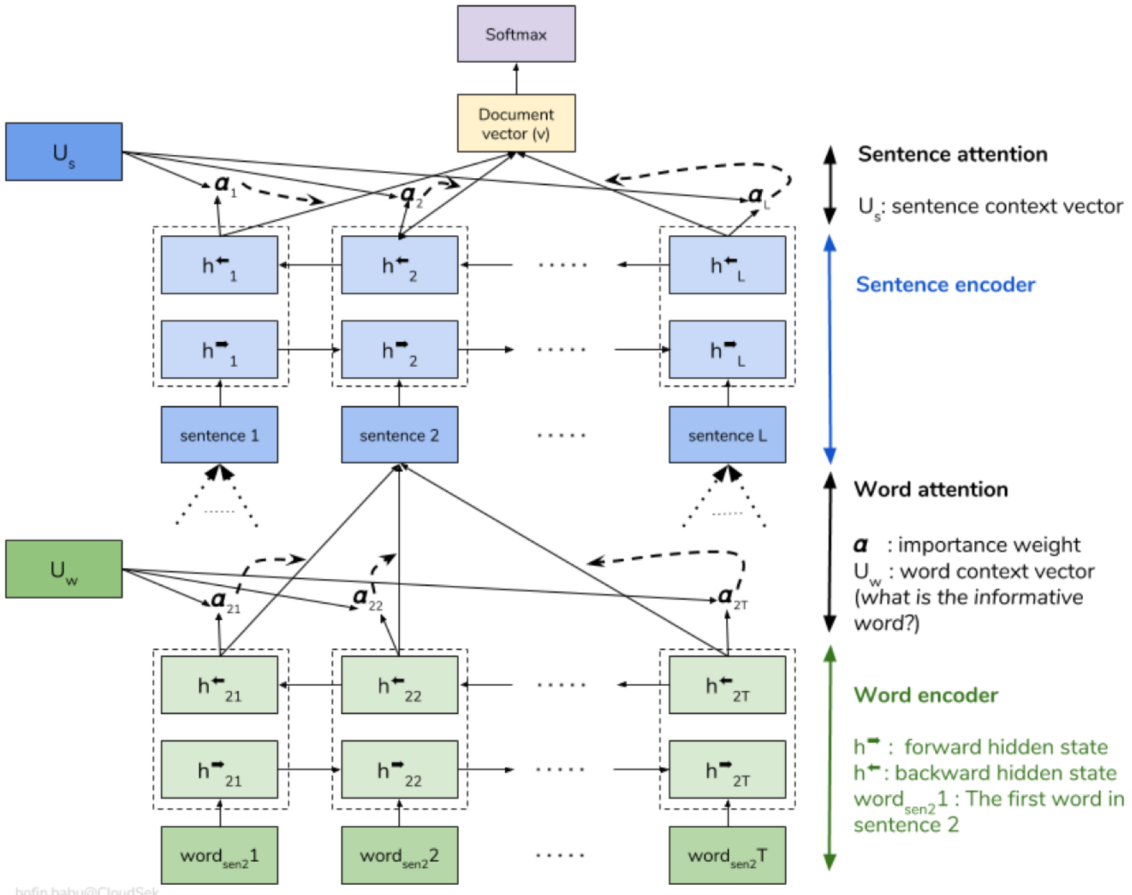


https://blog.openai.com/language-unsupervised/
https://github.com/openai/finetune-transformer-lm

HAN – Hierarchical Neural Attention Encoder



HAN Example – Document Classification



http://www.cs.cmu.edu/~./hovy/papers/16HLT-hierarchical-attention-networks.pdf

Thanks

- Connect with me on LinkedIn - www.linkedin.com/in/arorahardeep