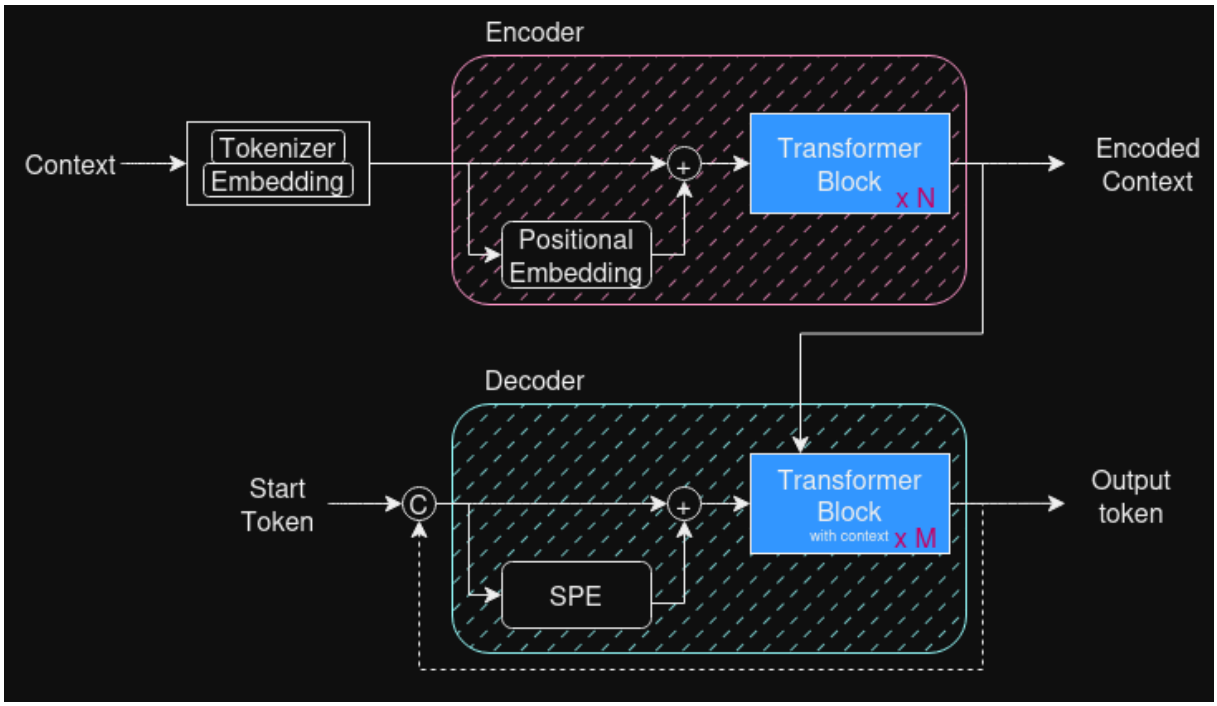
Deep Learning

Practical 6

1. A Look At GPT

GPT are transformer structures:



But only uses the decoder/generative part of the full network:

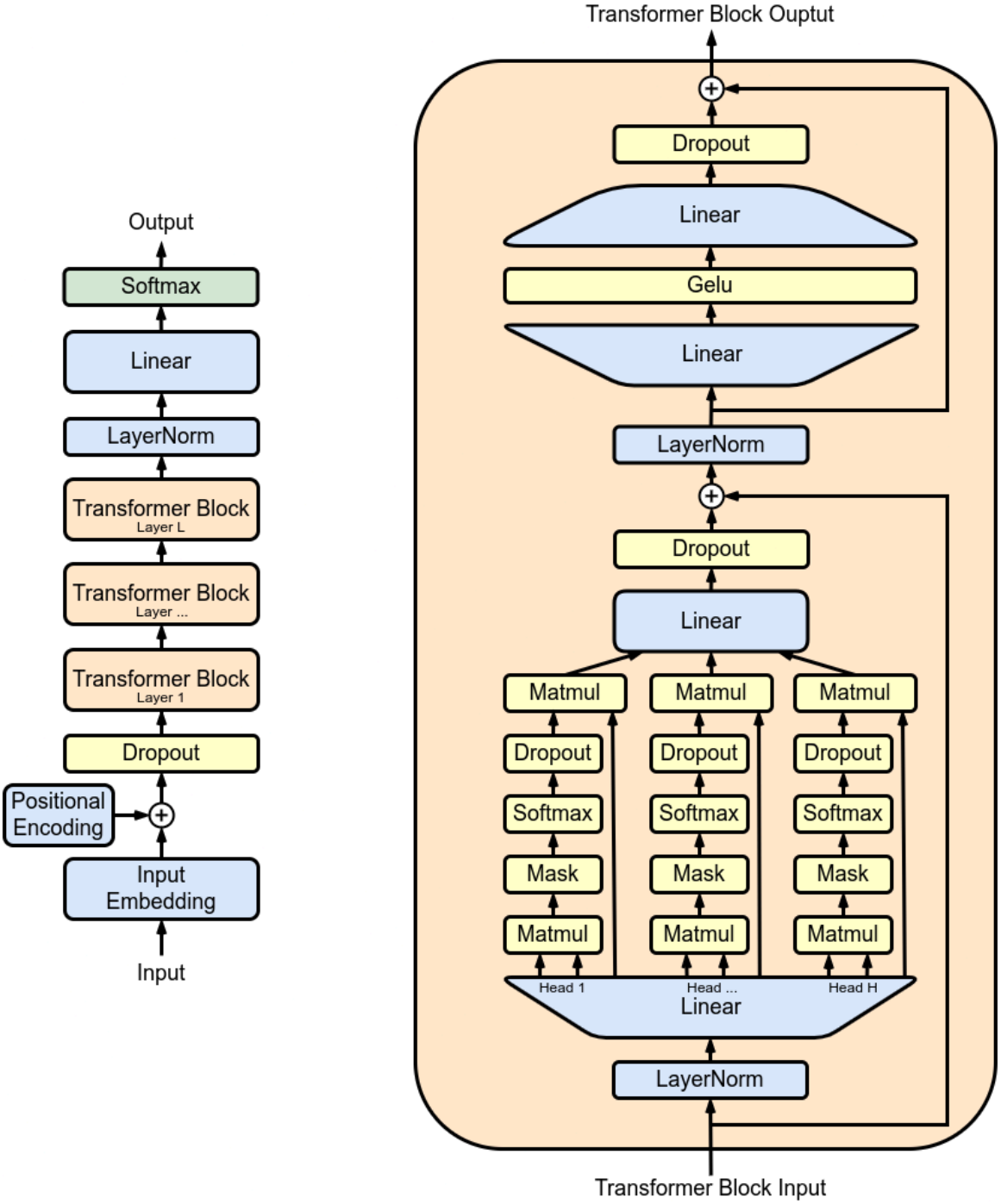


Figure 1: Full GPT architecture.

The objective of this practical is to build such a network !

2. Disclaimer

The GPT we will build in this practical is very hard to tune for it to generate proper sentences, event at this small scale, text generation is a very complicated and involved process. Nonetheless, the code provided here is basically what you could find looking at

bigger models (but with probably more parallelism on multiple GPUs).

3. Some Provided Files

You will start the practical with 3 files:

- `tp6.py` : the file to build, compile and run the model, you will need to fill in the model code.
- `tp6_utils.py`: the file with the intermediate layers (SPE, Transformer, Embeddings) plus a series of function to generate predictions. You will need to provided for some parts of the layers code.
- `tp6_data.py` : this file contains everything needed to load and transform the text dataset.

This practical should run "smoothly" in a google colab instance with T4 GPUs: https://colab.research.google.com/

4. First, the Layers:

4.1. SPE

SPE (Sinusoidal Positional Encoding) is already done, nothing to do here.

4.2. Embeddings

The embedding layer need its `call` method to be written ! According to the Figure 1, it should return the sum of a positional embedding and a token embedding.

4.3. TransformerBlock

This one is quite involved, there is a lot of layers to assemble for it to be complete.

The special `MultiHeadAttention` layers (https://www.tensorflow.org/api_docs/python/tf/keras/layers/MultiHeadAttention) is already coded. But you will need to provide the `LayerNormalization` that comes before, and also the `Dropout` (use `0.2`) that comes after (the `Dense` layers are already in the `MultiHeadAttention`).

For the next residual part, you will find some information about the `Gelu` here: https://www.tensorflow.org/api_docs/python/tf/keras/activations/gelu.

Write the missing parts for the TransformerBlock, provide for the `build` method and the `call` one.

5. Secondly, the Model:

Finally, write the code for the `build_GPT` function in `tp6.py` and run it!

6. What's More ?

Play with all the parameters, see what work or not, try to construct a GPT that generate sentences that ressemble real ones!