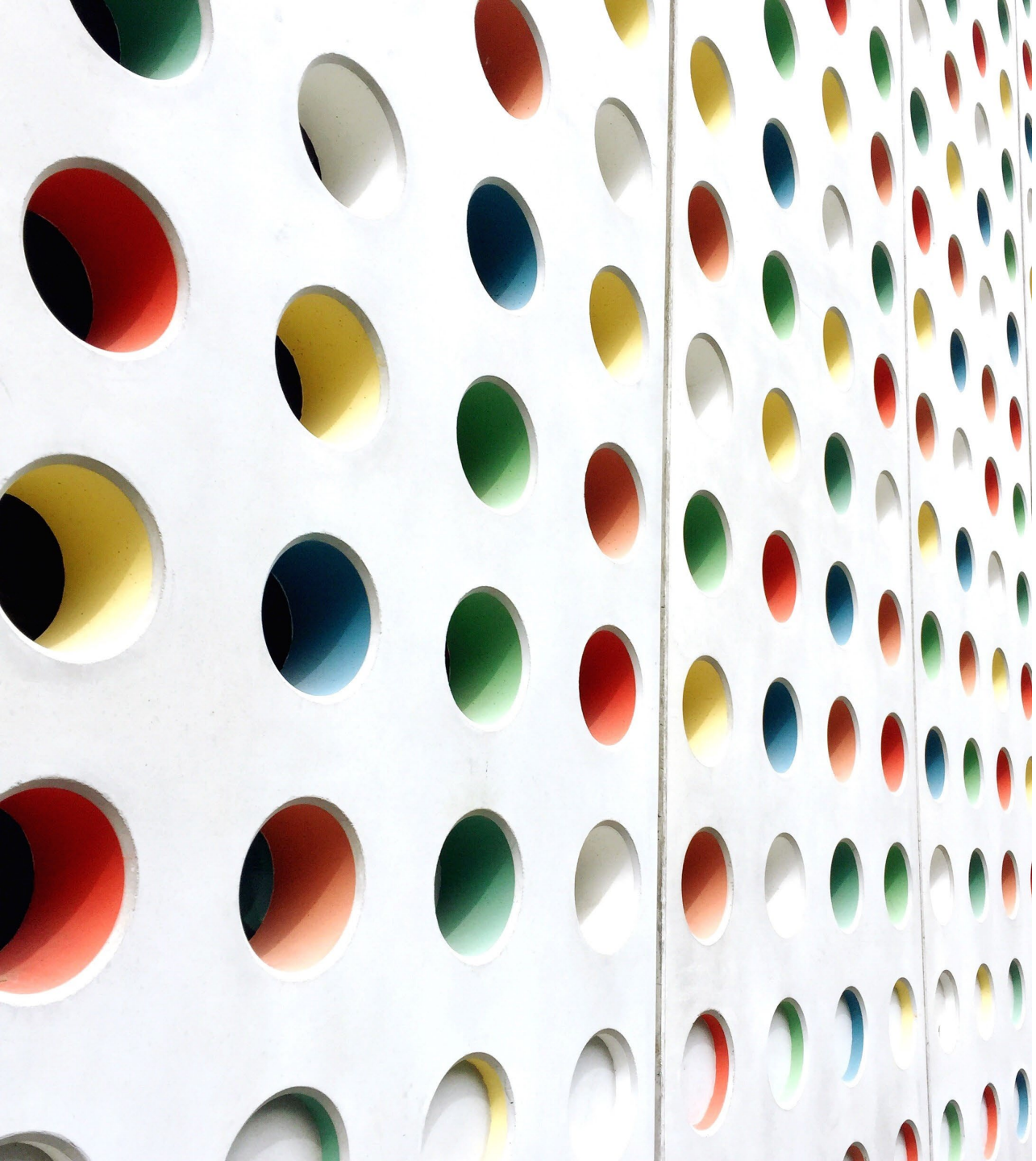
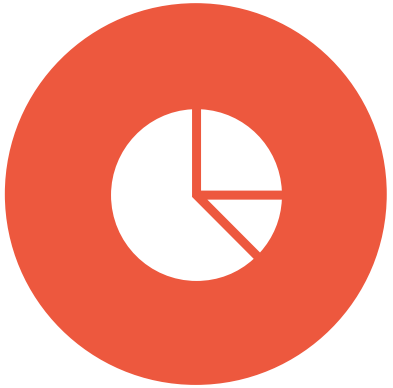


minGPT -vs- Encoder/Decoder

HOW IT WORKS

DAVE BROWN
SIGNALPOP LLC
WWW.SIGNALPOP.COM

Encoder/Decoder Model Use Cases



LANGUAGE TRANSLATION

NATURAL LANGUAGE PROCESSING

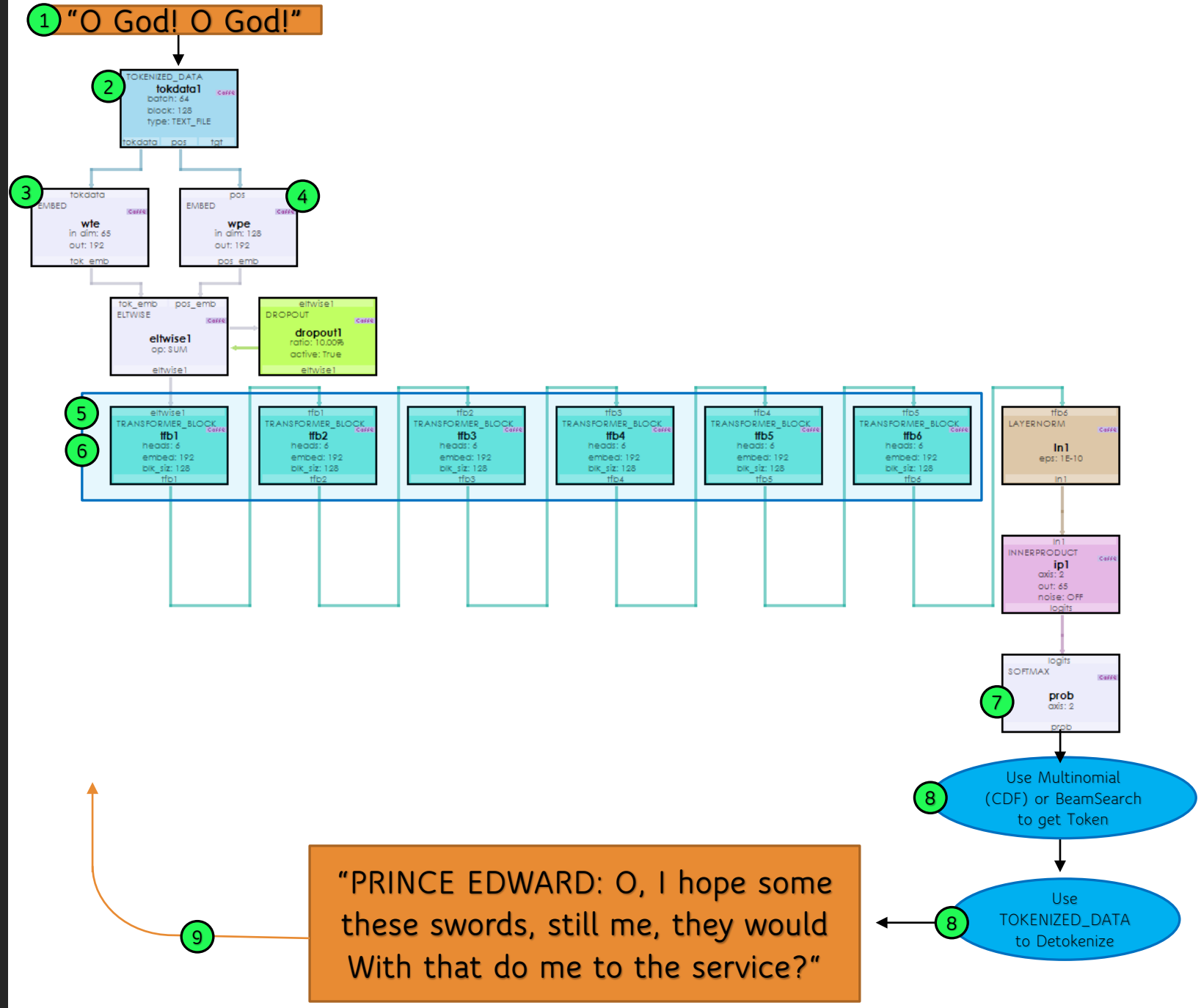
TIME SERIES

minGPT in action

1. Input sentence
2. Tokenize it (char, word, etc.)
3. Create token embedding and
4. Position encoding (e.g., sin(w))
5. Encoder creates encoding
6. Run self attention on encoding
7. Softmax creates probabilities
8. Detokenize
9. Produce predicted next char/word.

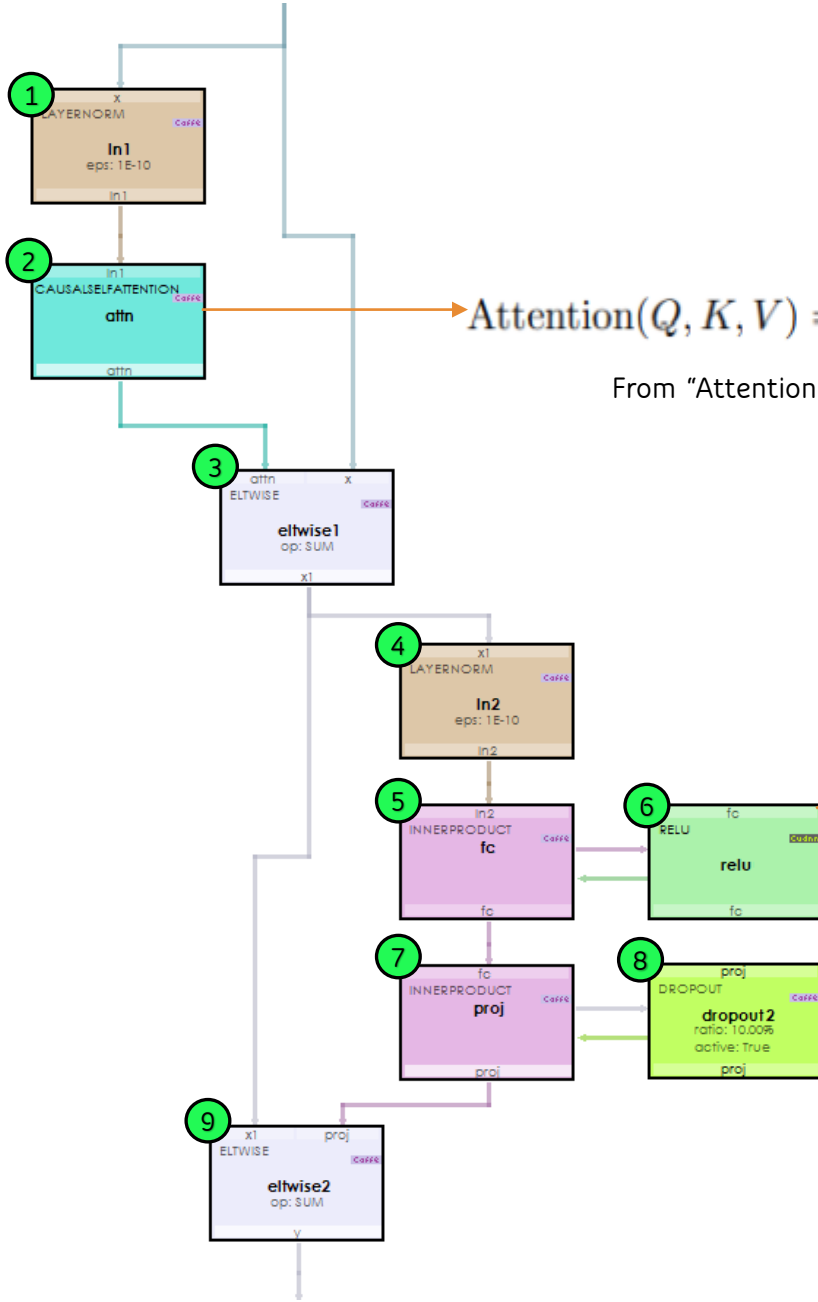
GPT-mini: 6 layers, 6 heads, 192 embedding size

GPT2-Large: 36 layers, 20 heads, 1280 embedding size = 774M parameters



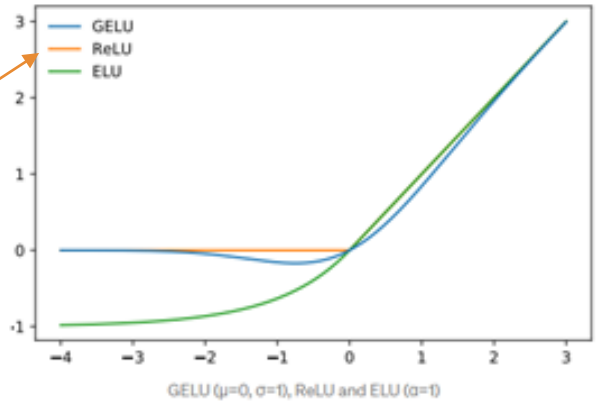
Transformer Block Internals

1. Layer Normalization 1
2. Causal Self Attention
3. Add X to Attn
4. Layer Normalization 2
5. Fc Inner Product
6. Activation (RELU, GELU, etc.)
7. Projection Inner Product
8. Dropout
9. Add proj to X1 (from #3 above)



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

From "Attention Is All You Need"

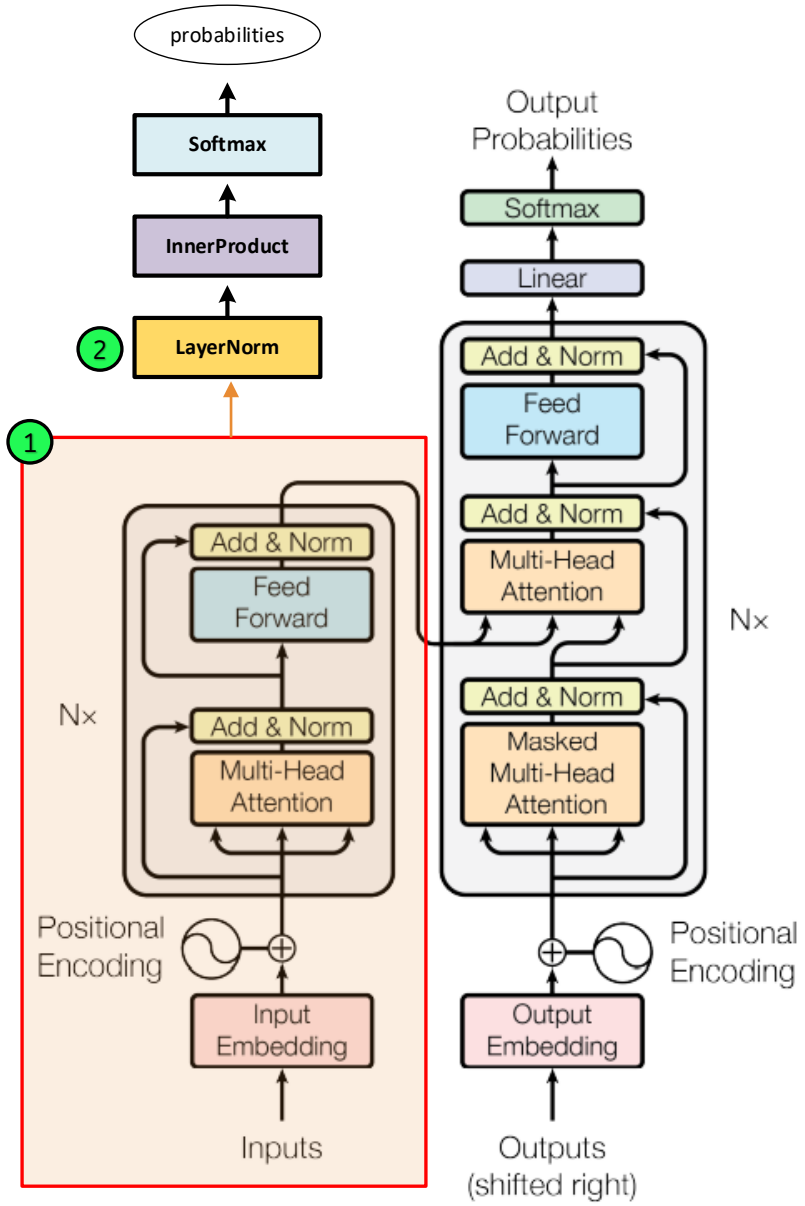


From "GELU activation"

minGPT vs full encode/decode model

1. minGPT only implements the 'encoder' side of the full model.
2. Just add LayerNorm + InnerProduct + Softmax to get output probabilities

How do we convert GPT to an Encoder/Decoder model?



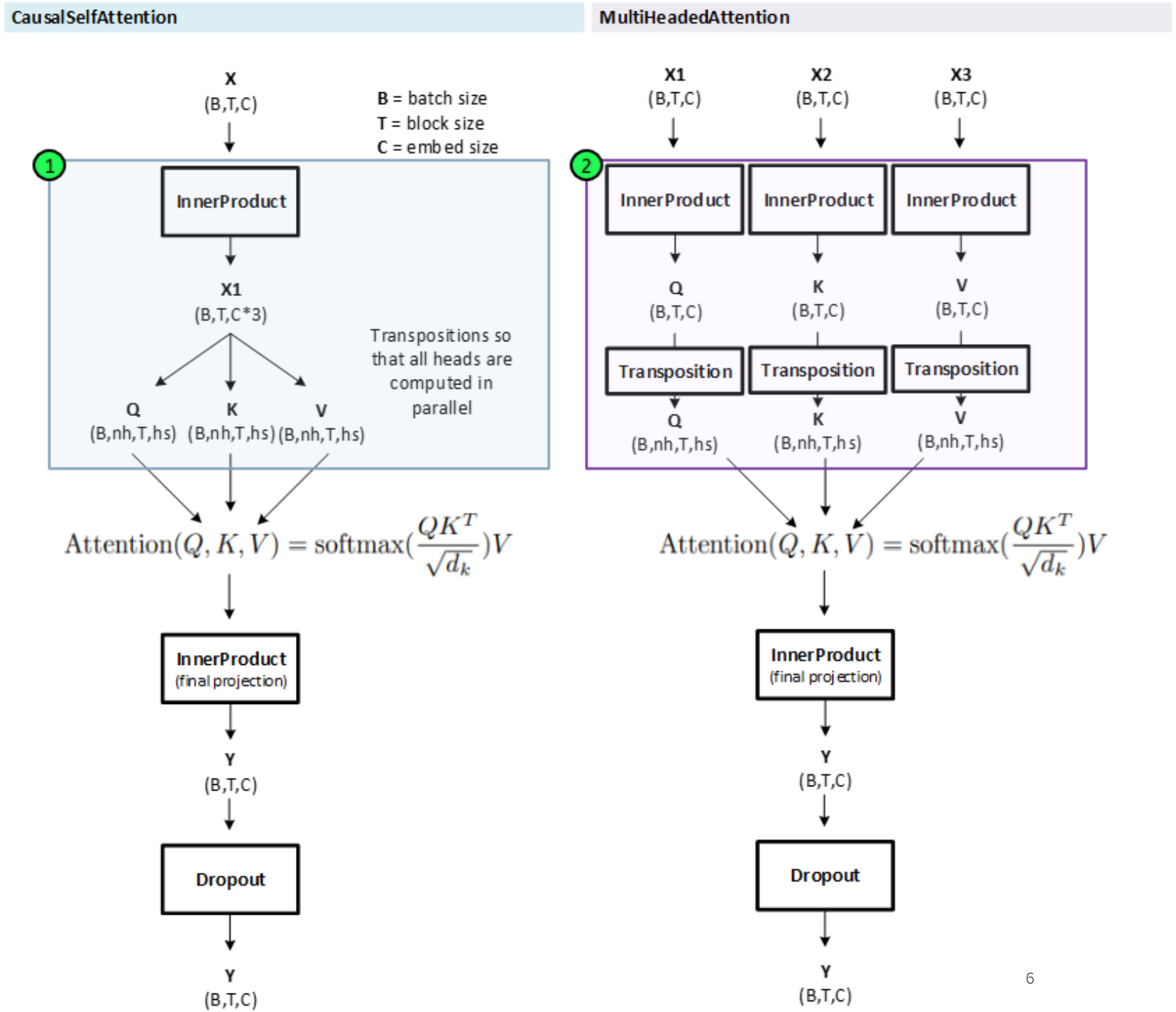
From "Attention Is All You Need", Fig1

First, convert …

CausalSelfAttention to MultiHeadedAttention

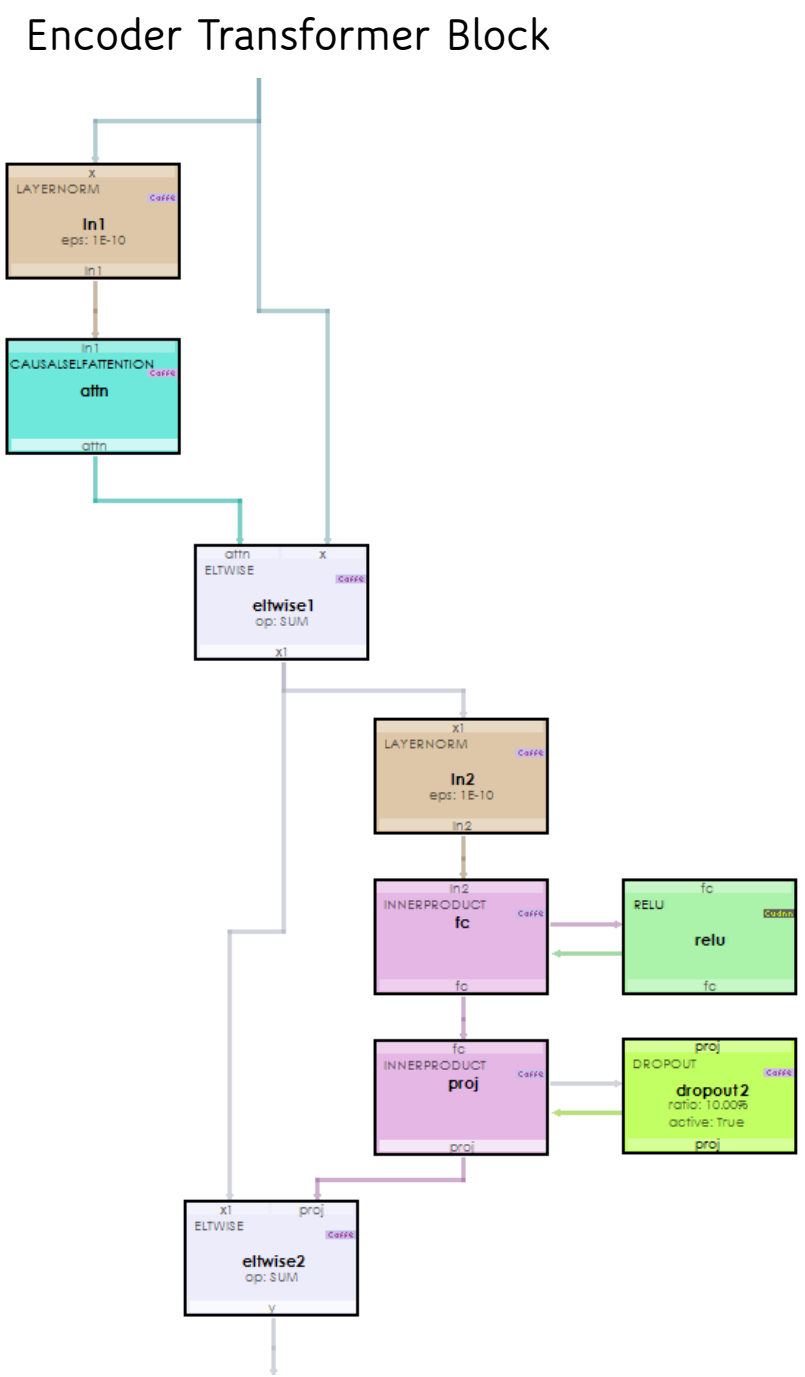
1. CausalSelfAttention takes one input ‘X’ and converts it to Q, K, V using a single InnerProduct

2. MultiHeadedAttention takes 3 inputs that are converted to Q, K, V using three InnerProducts.



Decoder Transformer Block Internals

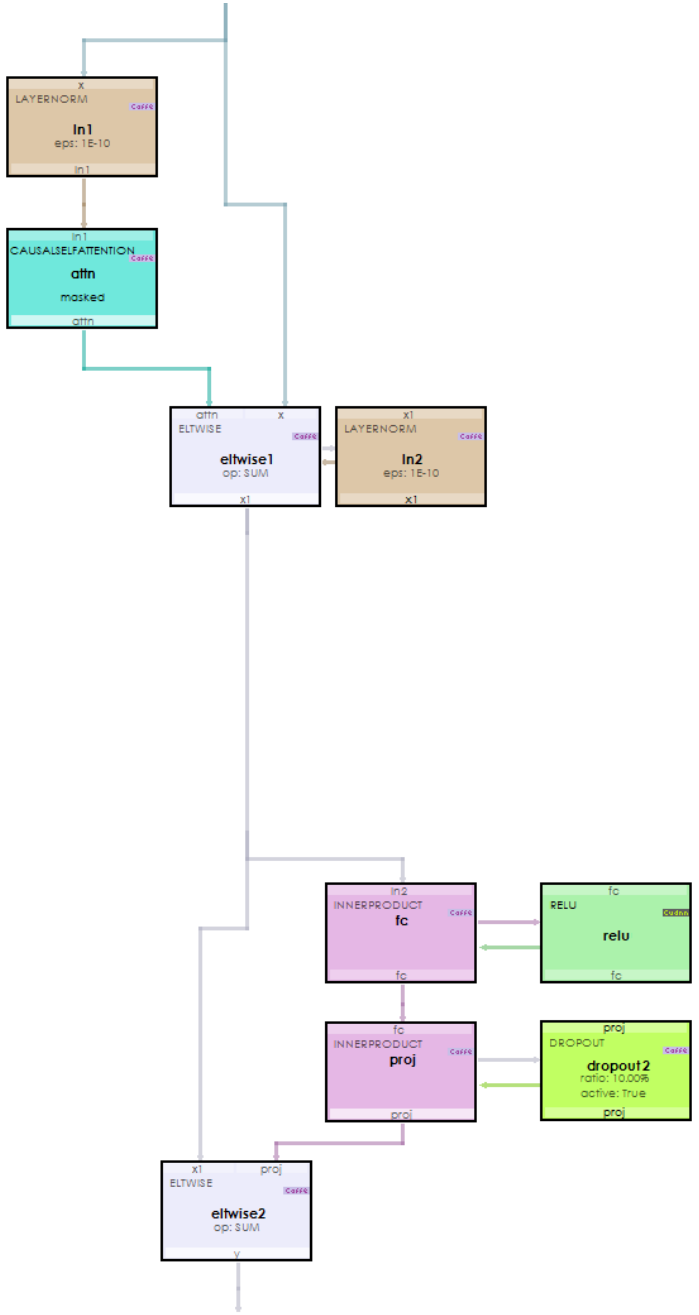
Starting with Encoder Transformer Block (Transformer Block in GPT)…



Decoder Transformer Block Internals

Make room for new layers used by Decoder Transformer Block…

Encoder Transformer Block



Decoder Transformer Block Internals

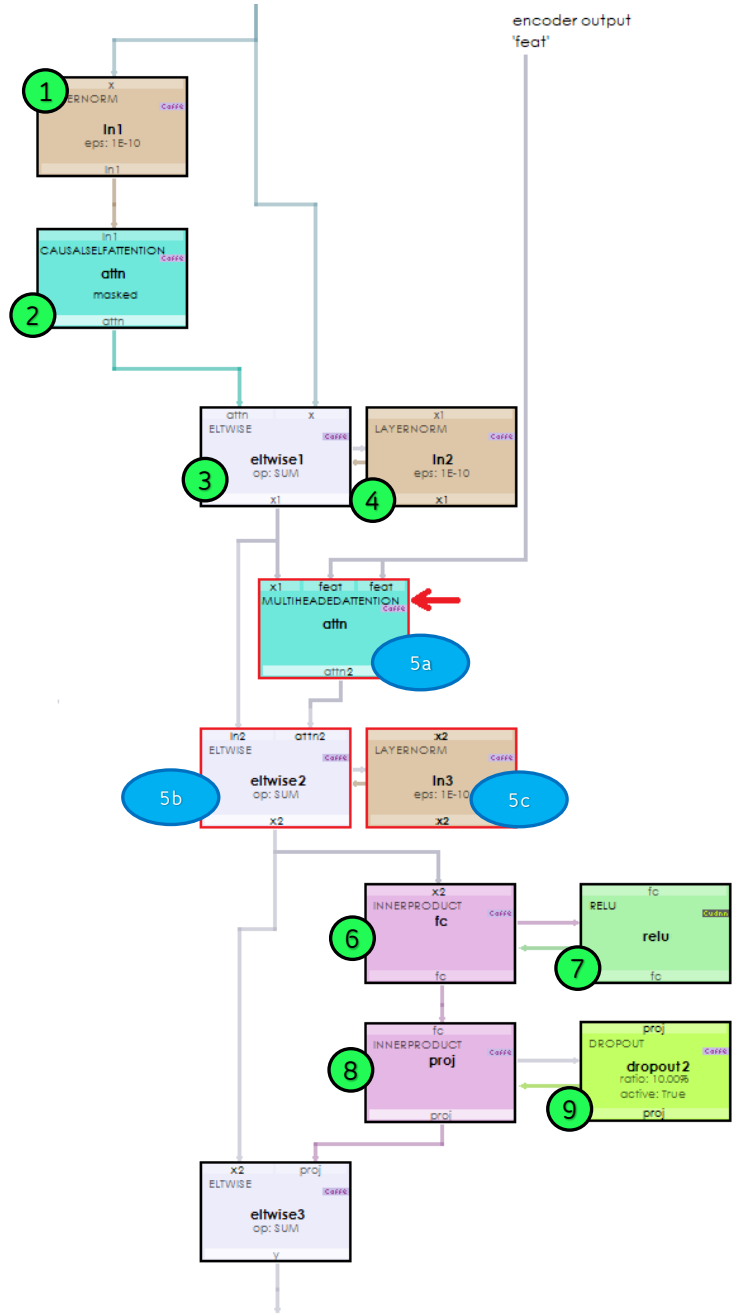
1. Layer Normalization 1
2. Causal Self Attention
3. Add X to Attn
4. Layer Normalization 2

5a. Multi-Head Attention

5b. Add X1 to Attn2

5c. Layer Normalization 3

6. Fc Inner Product
7. Activation (RELU, GELU, etc.)
8. Projection Inner Product
9. Dropout
10. Add proj to X2 (from #5c above)



Decoder Transformer Block Internals

1. Layer Normalization 1
2. Causal Self Attention
3. Add X to Attn
4. Layer Normalization 2

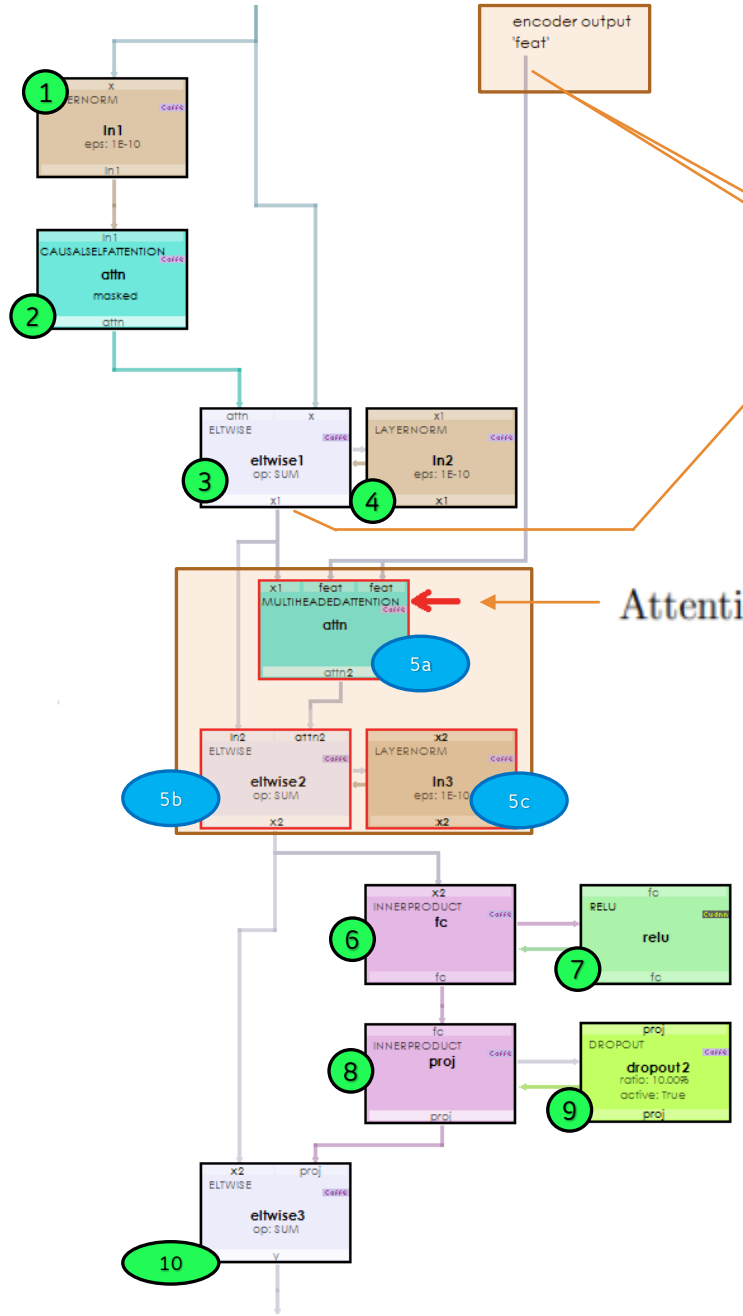
5a. Multi-Head Attention

5b. Add X1 to Attn2

5c. Layer Normalization 3

6. Fc Inner Product
7. Activation (RELU, GELU, etc.)
8. Projection Inner Product
9. Dropout
10. Add proj to X2 (from #5c above)

Decoder Transformer Block



Learning to match input and target probability distributions occurs here.

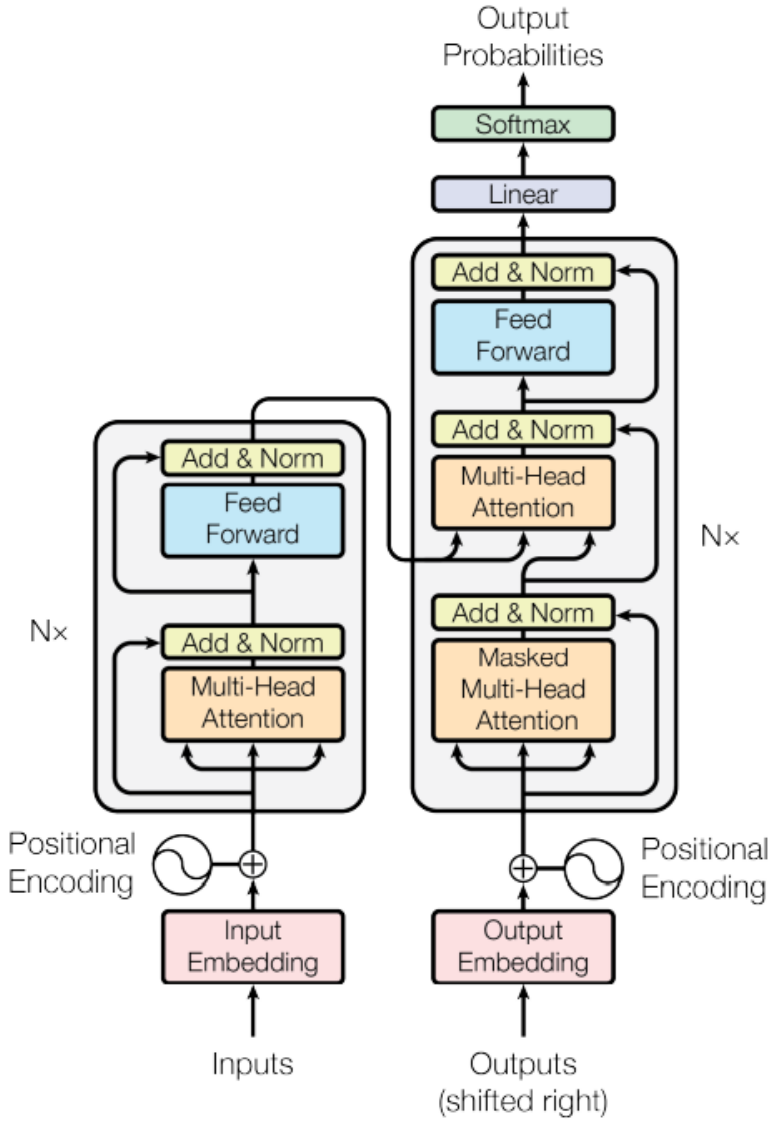
The same encoder output is sent to each decoder transformer block layer.

Q = x1
K = feat
V = feat

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

minGPT vs full encode/decode model

Full model uses both encode (left) and decode (right).



From "Attention Is All You Need", Fig1

Full encode/decode model

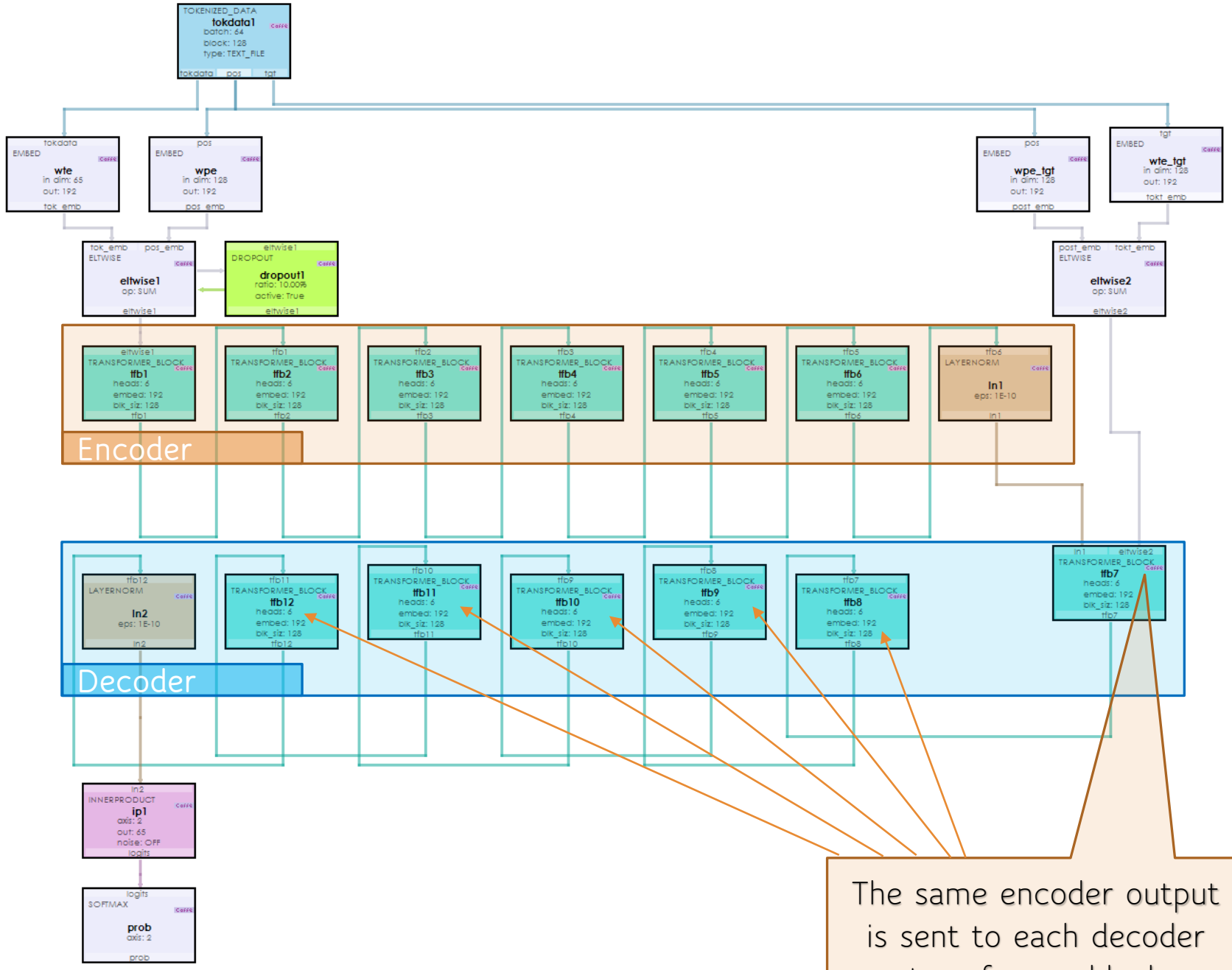
Full encoder/decoder model based off original minGPT model.

Note:

Target values are offset by one and may use a separate positional embedding.

Also Note:

Decoder TRANSFORMER_BLOCKS have ‘decode’ enabled which uses both the **CausalSelfAttention** and **MultiHeadAttention** layers internally.



Thank You!

References

Attention Is All You Need; Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, arXiv 2017 https://arxiv.org/abs/1706.03762

Transformer's Encoder-Decoder: Let's Understand The Model Architecture; KiKaBeN, 2021; https://kikaben.com/transformers-encoder-decoder/

The Illustrated Transformer; Jay Alammar, 2020, Jay Alammar Blog, https://jalammar.github.io/illustrated-transformer/

Transformer-based Encoder/Decoder Models; Patrick von Platen, Hugging Face, 2020; https://huggingface.co/blog/encoder-decoder

GELU activation; Shaurya Goel; Medium, 2019; https://medium.com/@shauryagoel/gelu-gaussian-error-linear-unit-4ec59fb2e47c