Transformer Architecture - LLM from Scratch Learning - Freecodecamp

2025-06-05 21:04

learn freecodecamp llms personal learning transformers

Title: LLMs from Scratch with Python

Forms of normalizing in ML

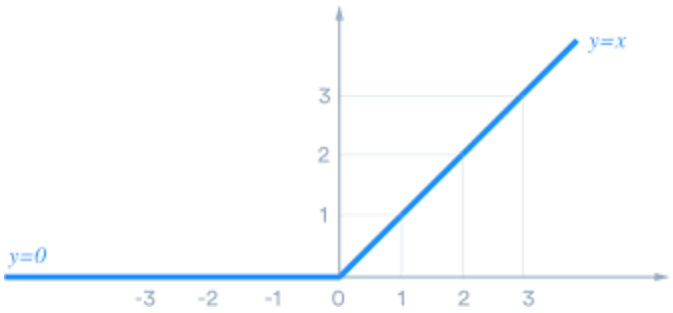
normalization techniques

softmax - not in input data, mostly used in output layer
min max scaling, z-score, decimal scaling, mean normalization, unit vector normalization (L2 normalization), robust scaling, power transformation

activation functions

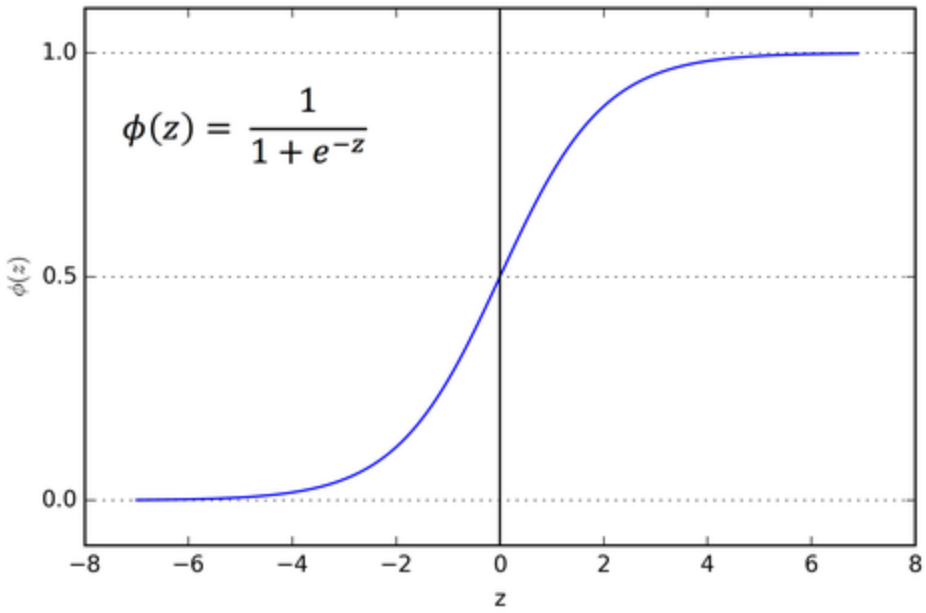
ReLU

0 or below 0 -> 0
above 0 -> remain the same



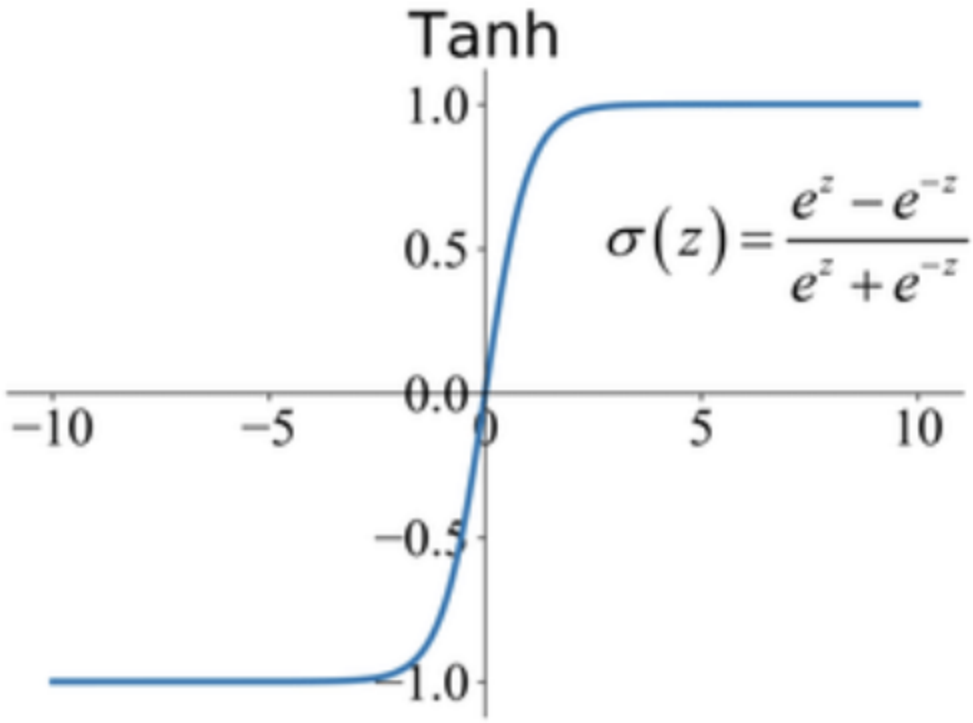
Sigmoid

normalize any inout into a range between== 0 and 1
big negatives -> closer to 0
big positives -> closer to 1
values near 0 -> closer to 0.5



tanh function

similar to sigmoid, but output values are between -1 and 1

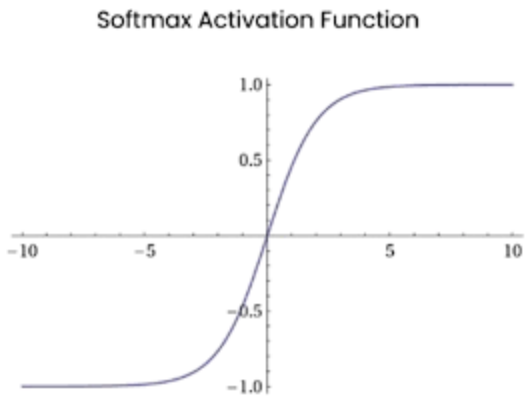


- **Hidden layers?** → Use **Tanh** (better zero-centered behavior).
- **Output layer for binary classification?** → Use **Sigmoid**.
- **Output layer for multi-class classification?** → Use **Softmax** (not Tanh or Sigmoid).

softmax function

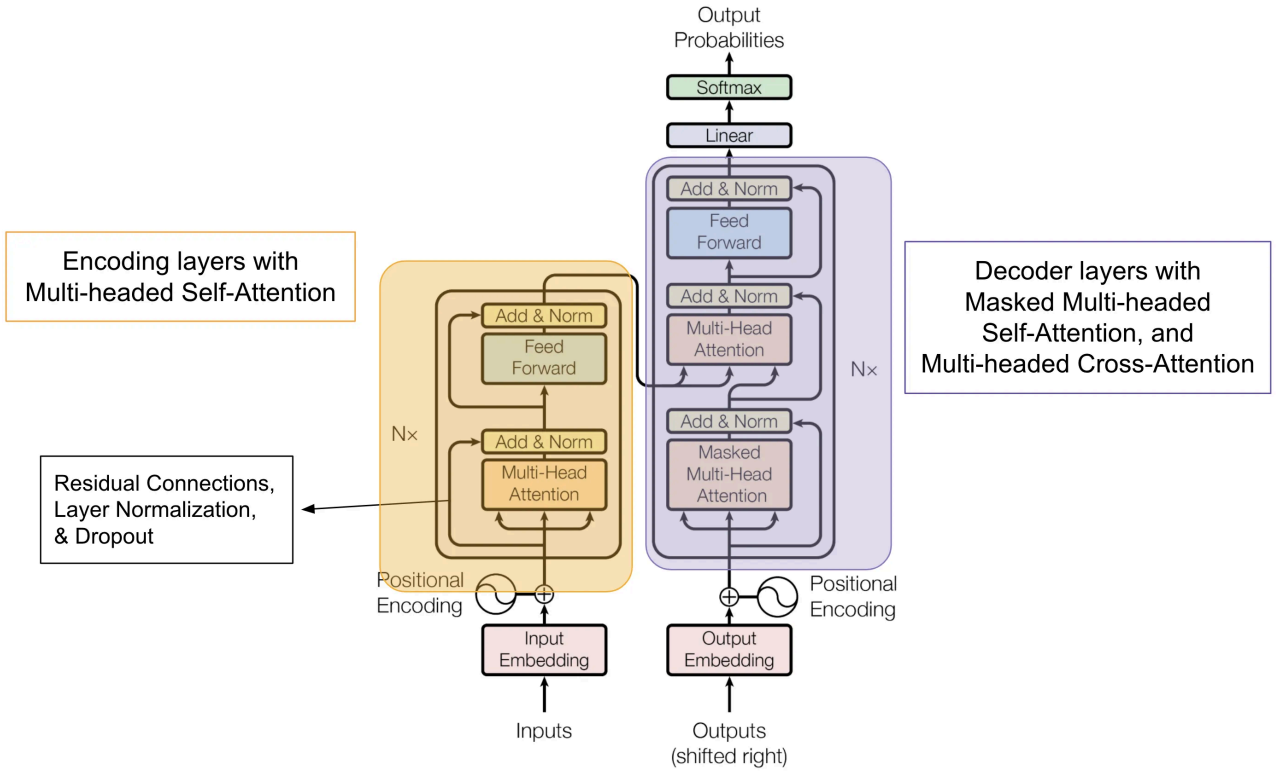
somewhat similar to sigmoid.

Feature	Sigmoid	Softmax
Output range	0 to 1	0 to 1
Used for	Binary classification	Multi-class classification
Number of outputs	One (per class, separately)	One set of outputs (for all classes)
Probabilities add up	No (each is independent)	Yes (they always add up to 1)
Best for	One-vs-rest problems	One correct class only (mutually exclusive classes)

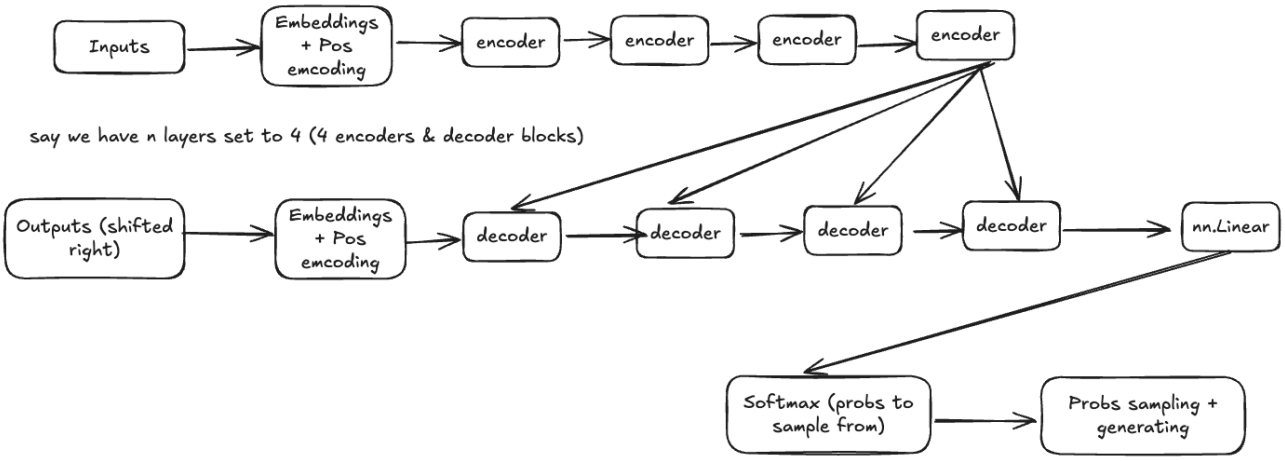


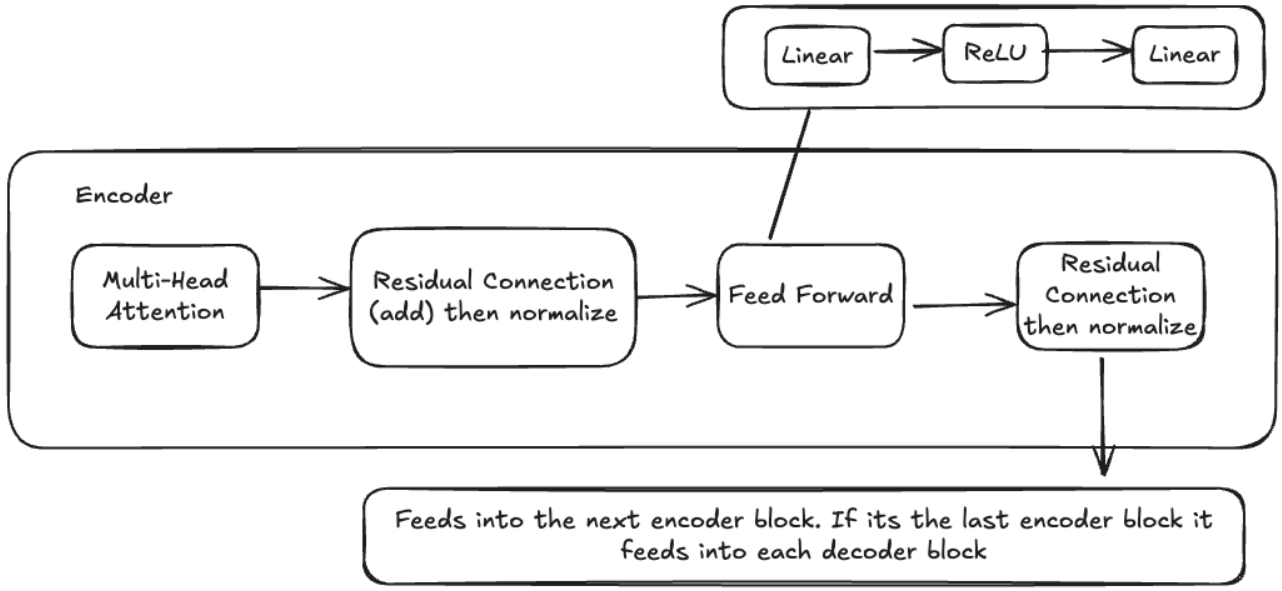
Transformer Architecture

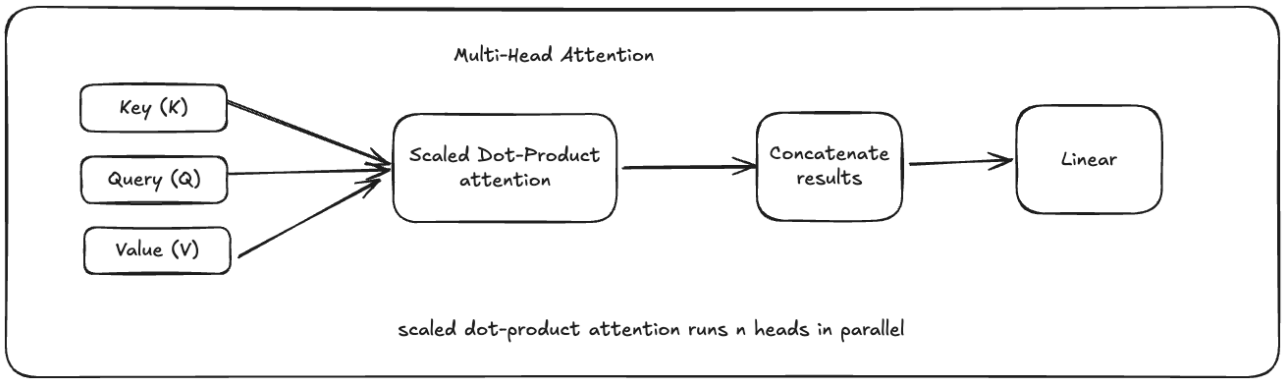
transformers attention



Attention sets specific scores to each token in a sentence as well as its position (positionally encoding)

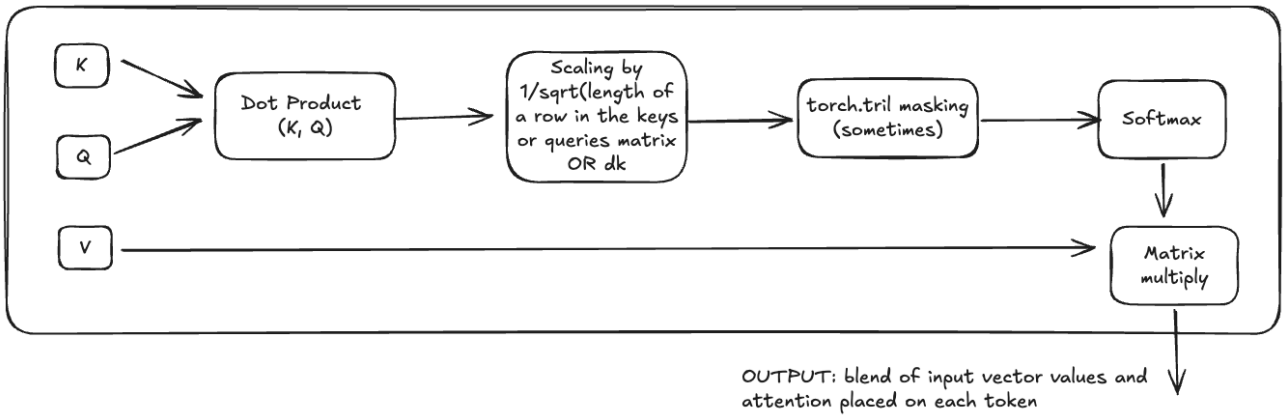






we call it multi-head attention because there are a bunch of heads learning different semantic info from a unique perspective

Scaled Dot product attention block



References

youtube: https://www.youtube.com/watch?v=UU1WVnMk4E8&t=132s
github: https://github.com/Infatoshi/fcc-intro-to-llms