LLaMa2

→ we copy the kv-heads to match the numbers of query heads

SwiGLU Function

LLaMa 2 $FFN_{SwiGLU}(x, W, V, W_2) = (Swish_1(xW) \otimes xV)W_2$

$\beta = 1$, Swish = SiLU

INFERENCE :-

Logits : the output of the last linear layer in Transformer model. They are also called probabilities, but not clearly as they don't sum up to 1.

→ The softmax scales them to make them sum upto 1.

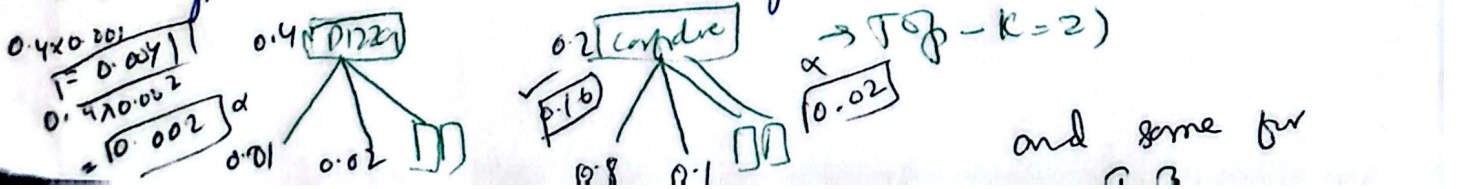
1) Greedy

→ select the highest probability

→ easy to implement
→ performs poor
→ initially wrong ones can lead to bad results.

2) Beam-Search (k=2)

→ select Top(2) or k with highest probability, we make 2 prompt in next step and ask the model to generate output next token and calculate cummulative reward for both path and select the highest cumulative reward.



increase inference time
perform better.

Temperature:

→ The idea is to scale the logits before applying softmax

→ low temperature → more confident model → high gap b/w 2 & 4 probabilities

→ high temperature → less confident model → low gap b/w 2 & 4 probabilities

probabilities/temperature → amplify & deamplify prob

Random Sampling

→ after softmax, as it is distribution we can take the number (high-probability sample) from it.

issues e.g [0.12, 0.07, 0.80]

first we chose 1, then 3, then 2, the higher the probability the more likely to be chosen.

Problem: the very little probability it may happen that we choose tokens that are bullshit. so we end up choosing low probabilities.

Top-K: to avoid bad tokens, we remove them

→ sort all the logits, and keep the largest K, and apply softmax to rest. But still low probabilities can make based on distribution. we want a best randomness at mid level.

Top-P: we keep only the tokens with highest probabilities such that their cumulative probability is greater than or equal to P (parameter). This let us deal with all types of distributions (flat or [illegible] ($P=0.5$), area under distribution.