CSCI 1010 Class 10

Profs. Michael Linderman and Phil Chodrow
Department of Computer Science
Middlebury College

The GPT2 model is trained for next-token prediction, i.e., to estimate $p(t_i|t_1, \ldots, t_{i-1})$. What do you predict would be highly likely next tokens given “See you”?

$$p(t_3|\text{“See”}, \text{“ you”})$$

later

tomorrow

soon

Slide 1 Notes

The vertical bar indicates a conditional probability, i.e., the probability of the token t_i given the preceding tokens, $t_1, \ldots, t_{i-1}$. Specifically here we are asking for the probability distribution over the next token given the preceding tokens “See you”.

We might predict ” soon”, ” later”, or ” tomorrow” to be highly probable next tokens.

True or false?

```
1 torch.argmax(logits, dim=-1) == torch.argmax(F.softmax(logits, dim=-1), dim=-1)
```

Recall that `logits` contains the raw, unnormalized scores for each token in the vocabulary and `softmax` is defined as $\mathrm{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$.

a. True

b. False

Slide 2 Notes

Answer: B

The argmax of the logits and the argmax of the softmax of the logits are the same, because softmax is a monotonic transformation. Thus, the token with the highest logit will also have the highest softmax probability. As a result, we typically don't compute the softmax if we only care about the most likely token.

```python
input_ids = tokenizer("See you", return_tensors="pt")["input_ids"]
for _ in range(20):  # Predict 20 tokens
    logits = model(input_ids).logits
    next_token_logits = logits[:, -1, :]
    next_token_id = torch.argmax(next_token_logits, dim=-1)
    input_ids = torch.cat((input_ids, next_token_id[:, None]), dim=-1)
```

Which of the expression best describes what this code does, where t_i is the next token and $\sim$ means "is sampled from"?

a. $t_i \sim p(t|t_{i-1})$

b. $t_i \sim p(t|t_1, \ldots, t_{i-1})$

c. $t_i = \arg\max p(t|t_{i-1})$

d. $t_i = \arg\max p(t|t_1, \ldots, t_{i-1})$

Slide 3 Notes

Answer: D

The code iteratively predicts the next token by taking the argmax of the predicted probability distribution given *all* preceding tokens. Note we are concatenating the newly predicted token onto the entire input sequence at each step, so the model has access to the full context of preceding tokens.

How would be implement C? We could only provide the last token as input to the model at each step, rather than the full sequence.

The sampling answers give us an idea for how we might generate more diverse text.

To the right is the probability of the 10 most likely tokens predicted by GPT2 after the prompt 'See you". Which of the following best describes what the "sampling decoding" strategy would pick as the next token?

Token	Probability
soon	0.084681
there	0.073554
all	0.072775
next	0.072505
guys	0.061731
in	0.055900
out	0.053875
on	0.040310
later	0.028322
,	0.027074
Total	0.5707

a. Always pick "soon" greedy

b. Pick "soon" approximately twice as often as "there"

c. Pick one of the 10 tokens above

d. Pick one of the 10 above more often than any other tokens

e. None of the above

Slide 4 Notes

Answer: D

The “sampling decoding” strategy samples from the full probability distribution over the vocabulary, so it could pick any token, but it is more likely to pick tokens with higher predicted probabilities. Since those 10 tokens account for about 57% of the total distribution, they will be picked more often than any other tokens.