Text Generation and Attention

(find and work with a partner)

1. *Text Generation.* Say we have a small vocabulary with just three letters `'a', 'b', 'c'`. At some point in text generation the probabilities for the next character are shown below, along with how the *index* of the next character is chosen.

```
char_arr = ['a', 'b', 'c']
proba = [[0.5, 0.4, 0.1]]
rescaled_logits = tf.math.log(y_proba) / temperature
char_idx = tf.random.categorical(rescaled_logits, num_samples=1)
```

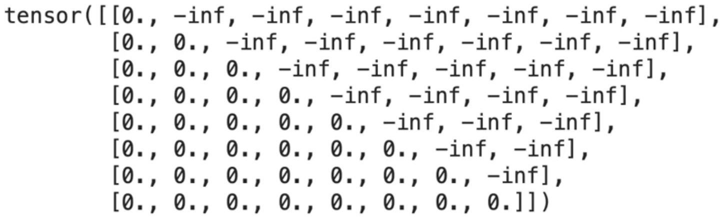
   (a) Explain the above code in words (note that `tf.random.categorical` chooses a random value proportional to the given logits – in other words it applies softmax first).

   (b) If we ran the above code many times and always chose character `'a'`, would this correspond to a high or low temperature?

   (c) If we ran the above code many times and saw roughly equal chances for each character, would this correspond to a high or low temperature?

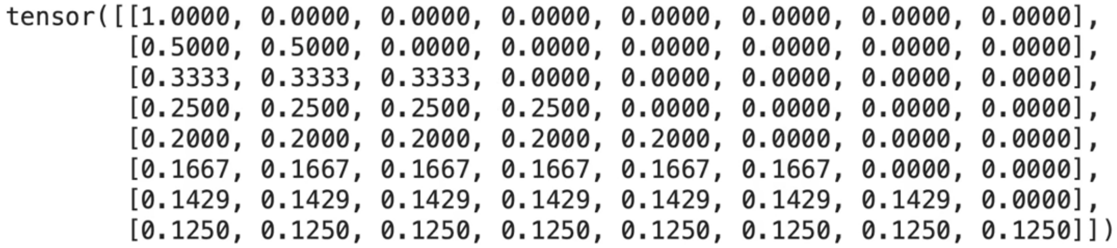
   (d) Using the equation for softmax, explain the above observations.

2. *Attention.* The mask below is for an 8-letter block, and row i shows the available letters target i can "pay attention" to. For example, the first letter in the target (which is the second letter in the block) can pay attention only to the previous letter (i.e. the first letter in the block).



```
tensor([[0., -inf, -inf, -inf, -inf, -inf, -inf, -inf],
        [0., 0., -inf, -inf, -inf, -inf, -inf, -inf],
        [0., 0., 0., -inf, -inf, -inf, -inf, -inf],
        [0., 0., 0., 0., -inf, -inf, -inf, -inf],
        [0., 0., 0., 0., 0., -inf, -inf, -inf],
        [0., 0., 0., 0., 0., 0., -inf, -inf],
        [0., 0., 0., 0., 0., 0., 0., -inf],
        [0., 0., 0., 0., 0., 0., 0., 0.]])
```

   (a) Apply the softmax function to each row of the above matrix. What do you get? Turn the page over to check your work!



```
tensor([[1.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000],
        [0.5000, 0.5000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000],
        [0.3333, 0.3333, 0.3333, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000],
        [0.2500, 0.2500, 0.2500, 0.2500, 0.0000, 0.0000, 0.0000, 0.0000],
        [0.2000, 0.2000, 0.2000, 0.2000, 0.2000, 0.0000, 0.0000, 0.0000],
        [0.1667, 0.1667, 0.1667, 0.1667, 0.1667, 0.1667, 0.0000, 0.0000],
        [0.1429, 0.1429, 0.1429, 0.1429, 0.1429, 0.1429, 0.1429, 0.0000],
        [0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250]])
```

(b) If we were analyzing 8-word blocks instead of 8-letter blocks and our input was:

`happy spring what are you planning for the`

with the following target (shifted one word to the right):

`spring what are you planning for the summer`

How would you predict the attention map would change? (i.e. when trying to predict `summer` from the input, what words would be the focus?)

Credit: adapted from "Hands-on Machine Learning" by Geron and "Let's build GPT" by Andrej Karpathy