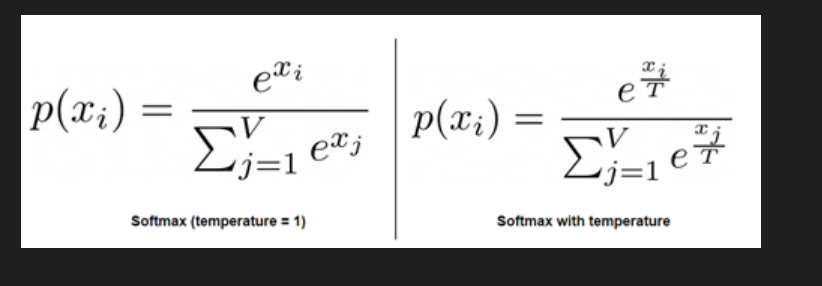
**### 1. Temperature**

*\*Higher the temperature, more the creativity. Lower the temperature, more deterministic the response.\**

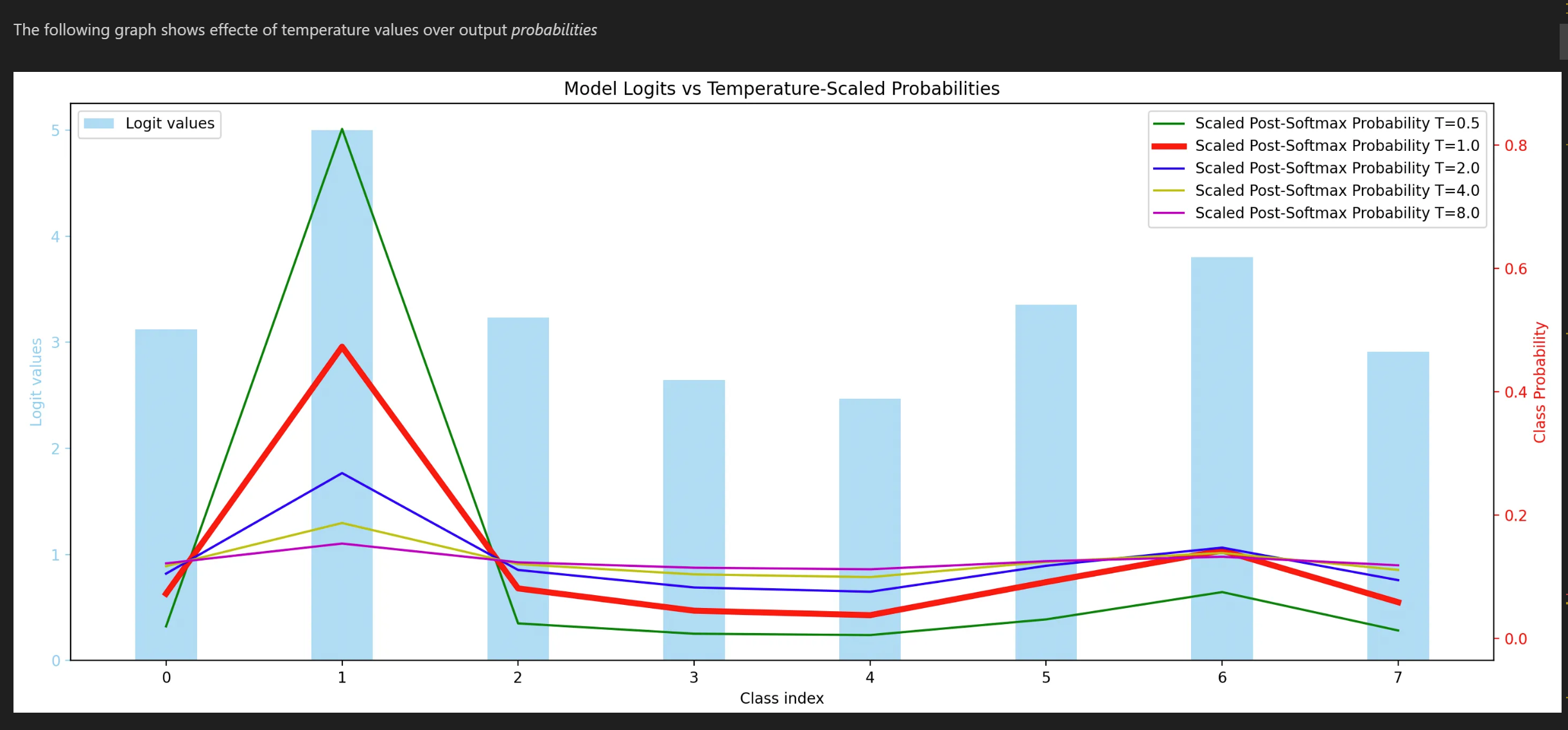
 >Temperature is a crucial hyperparameter in fine-tuning the output of large language models (LLMs) like GPT-3. It plays a vital role in controlling the randomness and creativity of generated text. The output of these large language models works as a function of the probability of word appearance. In other words, to generate a word, a probability is associated with each and every word in the dictionary and, based on this, it is determined how to proceed. The main idea of this hyperparameter is to adjust these probabilities to force randomness or determinism.

>The generation of probabilities for each word in the dictionary is done in the last layer by applying softmax as an activation function. Recall that softmax acts on the logits to transform them into probabilities. And this is precisely where the temperature comes into play.

**\*\*Softmax\*\***



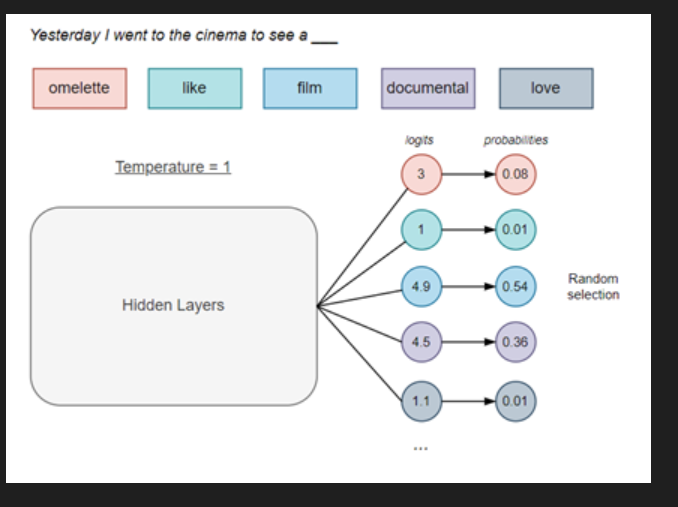
As can be seen from the above formula, softmax exponentiates each logit (x) and then, each exponentiated value is divided by the sum of all the exponentiated values. This step ensures that the output is a probability distribution, meaning that the values are between 0 and 1 and sum up to 1. The temperature hyperparameter is the value defined as “T” that is applied to each of the logits, making low temperatures skew the probabilities much more to the extremes.

The following graph shows effecte of temperature values over output \*probabilities\*

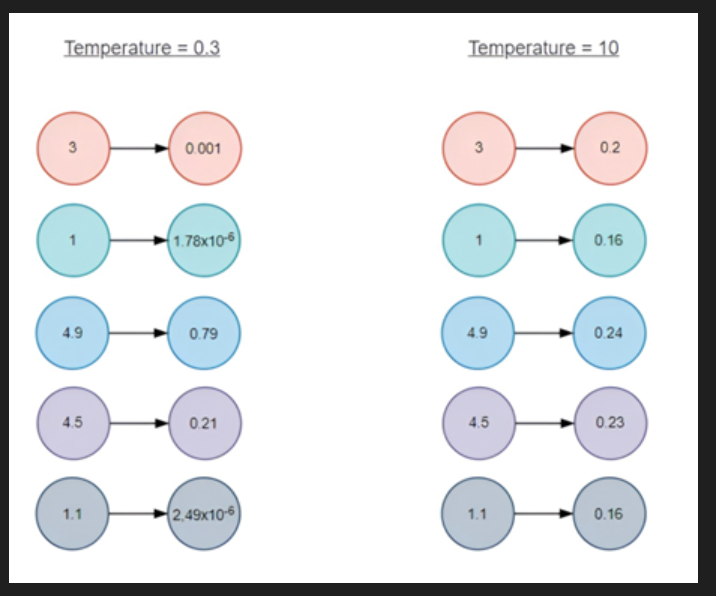
 Let’s imagine we have the following sentence:

- Yesterday I went to the cinema to see a \_\_\_

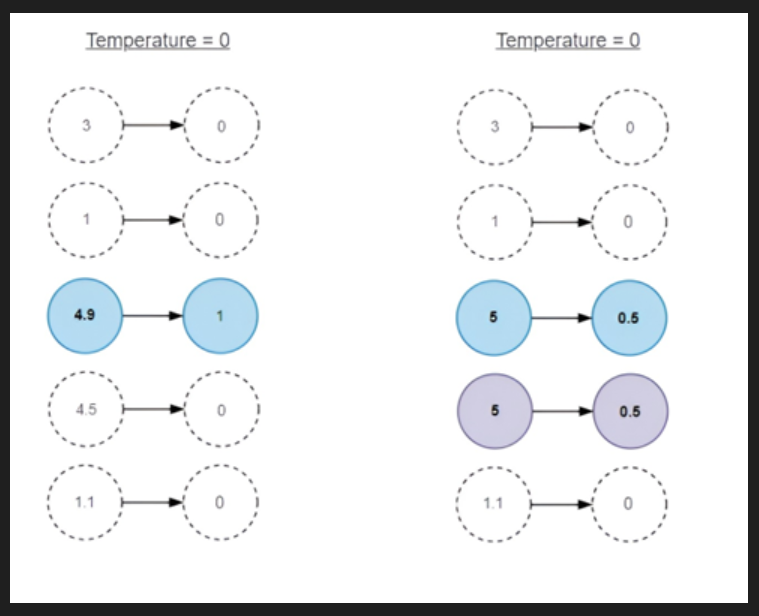
The idea is to predict the next word. The neural network will determine a probability for each of the words in the dictionary, in this case, we will use only 5 to simplify the process.



From the probabilities generated, randomness will be responsible for determining the word that follows. As can be seen, in the example above, normal softmax (temperature = 1) has been applied. In the following, we will see the result in the case of extreme values of these hyperparameters.



As temperature values approach 0 the higher probabilities increase further, making selection much more likely. Conversely, when the temperature gets much higher, the probabilities are softened, making more unexpected words more likely to be selected.



Therefore, when the temperature value is equal to 0, it becomes a deterministic solution. However, in case there are 2 words with the same logit value and hence the same probability, temperature 0 will make those words equally likely to be selected and add up to 1.

 Given a value of *\*temperature\**, and a probability distribution for next possible choices for tokens,  These 2 hyperparameters limit the choices further: *\*top\_p\** does it dynamically and *\*top\_k\** has a hard cut-off.

>**\*\*top\_p (***\*Nucleus\** **Sampling)\*\***: It selects the most likely tokens from a probability distribution, considering the cumulative probability until it reaches a predefined threshold “p”. This limits the number of choices and helps avoid overly diverse or nonsensical outputs.

>Effect of top\_p would **\*\*limit choices dynamically\*\*** from token to token.  Cumulative probability in one case may limit choice to three tokens and in the other case to four tokens.

>**\*\*top\_k (top-k Sampling)\*\***: It restricts the selection of tokens to the “k” most likely options, based on their probabilities. This prevents the model from considering tokens with very low probabilities, making the output more focused and coherent.