Key Points Explained

Vocabulary Size and Initial Weights

As we see in our Bigram that we have 80 characters in our vocabulary, which means my model can choose from 80 different tokens when predicting the next character. The model starts with random weights, indicating that it hasn't learned anything yet, so any prediction I make will be purely by chance.

Random Prediction and Likelihood

With this random initialization, I have a $\frac{1}{80}$ chance of correctly predicting the next token. This probability reflects the lack of learned information, which is typical when starting training from scratch. Likelihood, in this case, refers to how probable the observed outcome (the correct next token) is given my model's predictions. So, the likelihood of predicting the correct token is $\frac{1}{80}$ (or 0.0125).

Negative Log Likelihood

To measure how "bad" our predictions are, I use the negative log likelihood (NLL). This is a standard approach in machine learning, especially in classification tasks, to evaluate the model's performance. The formula for negative log likelihood is:

$$\text{NLL} = -\log(\text{likelihood})$$

If I plug in the likelihood of $\frac{1}{80}$:

$$\text{NLL} = -\log\left(\frac{1}{80}\right)$$

Calculating the Loss

When I calculate $-\log\left(\frac{1}{80}\right)$, I get a value of approximately 4.38. This loss value is high, indicating that my model's predictions are poor. In practical terms, a high loss means that the model is not performing well, reflecting a significant level of uncertainty in its predictions.

Interpreting the Loss

A loss value of 4.38 suggests that my model's chance of correctly predicting the next token is quite low (less than 2%). In training, my goal is to reduce this loss through gradient descent, which adjusts the model's weights based on the loss gradient. Over time, with enough training and data, I expect my model to improve its predictions, leading to a lower loss.

1 Probabilities and Logarithms

In classification problems, we deal with probabilities that represent how likely a prediction is correct. These probabilities are often small values between 0 and 1.

- For example, the likelihood of a prediction being correct might be as low as 0.001 (or $\frac{1}{1000}$).
- Small probabilities can lead to numerical instability in calculations.

2 Using Logarithms

The logarithm function transforms these small probabilities into a more manageable scale:

$$\begin{aligned}\log(0.5) &\approx -0.69\\ \log(0.1) &\approx -2.3\\ \log(0.001) &\approx -6.91\end{aligned}$$

This transformation makes it easier to work with the values mathematically.

3 Why Negative Log?

The negative sign is crucial because we want our loss function to be positive and easily minimized:

- **Loss Interpretation**: A lower loss indicates better model performance.
- Using the log likelihood directly would yield negative values for correct predictions, complicating the optimization.
- The negative sign ensures that lower probabilities (worse predictions) yield higher loss values.

4 Connection to Likelihood

The likelihood measures how well the model predicts the actual outcomes:

- High probabilities lead to lower (less negative) NLL values.
- Low probabilities result in higher (more negative) NLL values.

5 Example

If a model predicts the correct class with a probability of 0.01 (1

$$\text{NLL} = -\log(0.01) \approx 4.61$$

Conversely, if the model predicts the correct class with a probability of 0.5 (50

$$\text{NLL} = -\log(0.5) \approx 0.69$$

Here, the higher NLL value when predicting with 1% probability indicates a worse prediction compared to the 50% probability.

6 Summary

The negative log likelihood provides a quantifiable way to assess model performance based on predicted probabilities. By converting probabilities into a positive loss metric, it clarifies the learning process.

- Lower probabilities (indicating poor predictions) yield higher losses, guiding the model toward better predictions during training.

This understanding of NLL highlights its importance in machine learning, particularly in classification tasks.