Training Phase

Objective Function: The model minimizes a loss function, often the **cross-entropy loss**:

$$L(\theta) = -\sum_{i=1}^{N} y_i \log(p_\theta(y_i|x_i))$$

Where:

- θ represents the model parameters.
- x_i is the input sequence.
- y_i is the target (true next token).
- $p_\theta(y_i|x_i)$ is the probability the model assigns to y_i given x_i.

Gradient Descent: Parameters are updated using gradient descent:

$$\theta_{t+1} = \theta_t - \alpha \nabla L(\theta_t)$$

- α is the learning rate.
- $\nabla L(\theta_t)$ is the gradient of the loss with respect to the parameters.

Backpropagation: The computation of gradients involves backpropagation through the neural network, updating parameters layer by layer.

Inference Phase

Prediction: For a given input sequence x, the model predicts the next token:

$$\hat{y} = \text{argmax}_y p_\theta(y|x)$$

Where $\hat{y}$ is the predicted next token.
Sequence Generation: For generating longer sequences, this process is repeated:

$$y_1, y_2, ..., y_T = \text{argmax}_{y_1,...,y_T} \prod_{t=1}^{T} p_\theta(y_t|x, y_1, ..., y_{t-1})$$

Here, each token's prediction depends on the previous tokens and the input.

Few-Shot and Zero-Shot Learning

Few-Shot: The model leverages conditional probability for task-specific examples:

$$P(Y|X,E) = \prod_i P(y_i|x, e_1, ..., e_n, y_1, ..., y_{i-1})$$

Where $E = \{e_1, ..., e_n\}$ are example pairs.

Zero-Shot: The model infers from its training distribution without specific examples:

$$P(Y|X) \approx \sum_{E \in \text{training set}} P(Y|X,E)$$

Fine-Tuning

Parameter Adjustment: Fine-tuning involves adjusting parameters with a new loss function tailored to specific tasks:

$$L_{\text{fine-tune}}(\theta) = -\sum_{i=1}^{M} \tilde{y}_i \log(p_\theta(\tilde{y}_i|\tilde{x}_i))$$

Where:

- $\tilde{x}_i, \tilde{y}_i$ are from a task-specific dataset.
- M is the number of examples in the fine-tuning set.

This mathematical framework helps understand how LLMs are initially trained, how they generate responses, and how they can be adapted or fine-tuned for specific applications without real-time learning during user interactions.