Theoretical Foundations of Large Language Models (LLMs)

1. Language Modeling Theory

A language model assigns a probability to a sequence of words w1, w2, ..., wn.

Formally:

$P(w1, w2, ..., wn) = P(wt \mid w1, ..., wt-1)$

This formulation is autoregressive each word is predicted based on previous context.

2. Statistical Foundations

Traditional language models include n-gram models and Markov assumptions:

$P(wt \mid wt-1, ..., wt-n+1)$

These models have limitations in capturing long-range dependencies and suffer from data sparsity.

3. Neural Language Modeling

Neural networks learn to represent sequences using continuous vectors.

The objective is to minimize the cross-entropy loss:

$L = - \log P(wt \mid w<t)$

4. Transformers: The Foundation of LLMs

Transformers use self-attention to process sequences in parallel.

a. Self-Attention Mechanism:

Given input X, compute queries Q, keys K, and values V:

$Q = XW^Q, K = XW^K, V = XW^V$

$Attention(Q, K, V) = softmax(QK^T / sqrt(dk)) * V$

b. Multi-Head Attention:

Multiple attention heads capture different relationships:

MHA(Q, K, V) = Concat(head1, ..., headh)W^O

5. Positional Encoding

To inject order into tokens, positional encodings are added:

PE(pos, 2i) = sin(pos / 10000^(2i/d))

PE(pos, 2i+1) = cos(pos / 10000^(2i/d))

6. Model Objective and Training

The model is trained to predict the next word from context:

L = - log P(wt | w1, ..., wt-1)

Optimized using backpropagation and gradient descent.

7. Representation Learning

Tokens are embedded into a vector space and transformed into contextual embeddings based on surrounding words.

8. Emergent Properties and Scaling Laws

LLMs improve as model size, data, and compute increase. Scaling laws suggest loss decreases as resources grow.

9. Limitations

- Non-deterministic outputs
- Training instability
- No grounding in real-world knowledge