LLMs Explained From First Principles

Subject: Vectors, Attention, Backpropagation, and Scaling Limits

Executive Summary: The Reality and Limits of Large Language Models

The Bottom Line Large Language Models (like the one powering ChatGPT or Gemini) are not "thinking" machines, nor do they possess human-like reasoning. They are highly advanced statistical engines. They generate text by predicting the most probable next word based on massive datasets and sheer computing power. Any appearance of intelligence is the result of extreme scale, not genuine comprehension.

Key Takeaways for Business Strategy

- **"Understanding" is an Illusion of Scale:** Under the hood, LLMs use complex math to find patterns in data. When a model answers a question correctly, it isn't reasoning through the problem; it is calculating the statistically most appropriate response based on billions of previous examples. They are excellent at pattern recognition but lack actual logic, causality, or the ability to know when they are wrong.
- **Learning is Static, Not Dynamic:** Once an LLM is trained, its core knowledge is frozen. Interacting with a chatbot does not teach it new facts. Any "learning" you see during a chat is temporary and vanishes when the session ends. Permanently updating a model with proprietary company data or new facts requires fine-tuning or entirely retraining the model, both of which are computationally expensive processes.
- **The Soaring Cost of AI:** AI capability has improved rapidly because tech companies have continually scaled up the hardware and data used to train them. However, this is a brute-force approach. The cost to train frontier models is skyrocketing—from around $50 million a few years ago to projected costs of over $1 billion for the next generation.
- **Diminishing Returns and Physical Ceilings:** The industry is facing a reality check regarding "Scaling Laws." While pouring more money and computing power into models does make them better, the gains are starting to flatten. Furthermore, the future of AI scaling is increasingly constrained by physical and economic limits: extreme energy consumption, finite high-quality training data, and the physical limits of microchip manufacturing.

Strategic Implication

Businesses should treat LLMs as powerful, albeit flawed, statistical tools rather than infallible synthetic employees. Strategic ROI will come from identifying highly specific, pattern-based tasks (like summarizing documents, generating boilerplate code, or structuring unstructured data) where the model's current capabilities excel, rather than banking on the imminent arrival of "Artificial General Intelligence" to solve complex, novel business problems.

1. The Core Architecture: Text to Math

The foundation of the Transformer architecture relies on converting human language into high-dimensional mathematics. Models do not "read" words; they process numerical representations called **vectors**.

- **Tokenization & Embeddings:** Each word (or token) is mapped to a vector (a long list of real numbers) in a high-dimensional space.
- **Contextual Shaping:** During training, the model slowly shapes where these words sit relative to one another to capture semantic relationships.

To process these vectors, the model creates three distinct representations for each token using basic matrix multiplication:

Vector Type	**Role in the Network**
Query (Q)	The question a token asks about other tokens in the sequence.
Key (K)	The label or "identity" a token broadcasts to other tokens.
Value (V)	The actual underlying information the token carries forward.

2. The Heart of the Transformer: Attention

The Attention mechanism dictates how tokens share information and build context. It measures how aligned the Query of one token is with the Keys of all other tokens using a dot product.

Mathematically, this core operation is expressed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The Attention Process:

1. **Dot Product:** Calculates raw similarity scores between Queries and Keys.
2. **Scaling:** Divides by the square root of the vector dimension ($\sqrt{d_k}$) for numerical stability.
3. **Softmax:** Converts raw scores into a probability distribution (all positive, summing to one).
4. **Weighted Sum:** Multiplies the probabilities by the Value vectors to create a new, context-aware token blend.

Multi-Head Attention & Position

Instead of calculating attention once, the Transformer runs multiple "heads" in parallel. Each head learns different language patterns (syntax, long-range dependencies). Because Transformers process everything simultaneously, **sinusoidal positional encodings** are injected into the vectors to give the model a mathematical sense of word order.

3. How Neural Networks Learn: Backpropagation

A neural network is a layered system of artificial neurons. Learning is not comprehension; it is the reduction of statistical error.

Component	**Function**
Inputs	Raw numerical values passed into the neuron.
Weights	Multipliers that determine the importance of an input.
Biases	Threshold offsets that help trigger neuron activation.
Activation Function	Introduces non-linearity (e.g., ReLU, GELU) to model complex data.

The Learning Cycle:

1. **Forward Pass:** The model processes inputs and predicts the next token.
2. **Loss Calculation:** The prediction is compared to the correct answer using a cross-entropy loss function.
3. **Backward Pass (Backpropagation):** The error is sent backward through the network.
4. **Gradient Descent:** Using calculus (the chain rule), the model calculates the gradient (sensitivity) of every weight and adjusts them microscopically to reduce future errors.

4. The Illusion of Memory in Chat

Once a model finishes training, its weights are frozen. It cannot learn new facts dynamically.

Mechanism	Description	Result
In-Context "Learning"	Tracking patterns within a single chat session.	Evaporates completely when the chat ends.
Targeted Updates	Trying to isolate and change a single fact in the weights.	Causes "catastrophic forgetting" due to distributed knowledge.
Fine-Tuning	Retraining a subset of parameters on new data.	Resource-intensive but creates permanent changes.

5. The Economics, Physics, and Limits of Scaling

Training an LLM is a brute-force statistical compression task that is entirely bound by physical and economic reality. The backward pass of backpropagation touches billions of parameters and requires massive matrix multiplications, making GPUs and TPUs mandatory.

Estimated Training Costs by Generation:

Model Generation	Estimated FLOPs	Hardware Setup	Estimated Cost
Early Generation	10^{24}	Thousands of GPUs (Weeks)	$10M - $50M
Current Frontier	10^{25}	10,000+ Accelerators (Months)	$100M - $300M
Next Generation	10^{26}	Data-center scale operations	$500M - $1B+

Scaling Laws and Diminishing Returns

The argument for making models larger is based on **Scaling Laws**. As compute, data, and parameters increase, the loss (error rate) decreases predictably according to a power law:

$$L \propto C^{-\alpha}$$
Where $L$ is Loss, $C$ is Compute, and $\alpha$ is a small exponent (e.g., 0.05 to 0.1).

Conceptual Graph: Diminishing Returns in LLM Scaling

Plaintext

```
Loss (Error Rate)
|
| *
|   *
|     *
|       * *
|           * * * |                * * * * * * |                          * * * * * * * * * * * (Flattening Curve)
+------------------------------------------------------- Compute ($/Power)
```

The Reality Check:

- **The Curve Flattens:** Because the exponent $\alpha$ is small, a 10x increase in compute yields increasingly smaller real-world gains.
- **The Data Wall:** High-quality, human-generated text is finite.
- **The Physics Wall:** Transistors cannot shrink forever, and power/cooling constraints dictate that moving data around the hardware is as expensive as doing the math itself.

Scaling has worked historically, but infinite intelligence is not guaranteed. Future breakthroughs will require architectural shifts, not just more data centers.