On the Ability To Reason and Large Language Models

Bill Cochran

July 20, 2025

1 The Idealized Language Model

We model a Large Language Model (LLM) as a two stage operator:

$$\mathcal{M} = Q \circ N$$

where $N : \mathcal{P} \to \mathbb{R}^{n \times d}$ is a *neural network operator* that maps the *promp space* $\mathcal{P}$ to a set of activations in the space of real numbers. n is the number of levels of the neural net and d is the number of nodes per level. This is an idealization of current modern LLMs, but captures the spirit of what these networks are trying to accomplish.

The second operator, $Q : \mathbb{R}^{n \times d} \to \mathcal{R}$, is the quantization process, the mapping of these activations on to tokens that can be projected into natural languages. Generally, it is assumed that there exists some space of "language" such that both $\mathcal{P}$ and $\mathcal{R}$ are subsets of it.

We can observe that Q introduces noise into the project and we can model that noise as ζ. In other words $Q(x) = \hat{x} + \zeta$ for $\hat{x}$ some idealized set of tokens that map to a set of activations. We call this mapping *meaning* Q transforms text while preserving a 1-to-1 mapping with a set of activations. This noise is currently modeled in LLMs through various filters and, especially, the temperature mechanism. As in, the noise is explicitly add by LLMs to generate realistic behavior.

This idea of modeling ζ this way hinges on the ability of Q preserving a one-to-one mapping between an idealized set of tokens and the set of activations, a reasonable assumption given current LLM constructions.

The first operator, the neural net, is the place of interest here as it is possible to put a bound on how clever the neural net can be based on n. We model N as a linear approximation of n-th order logic and demonstrate correctness. From there, we can estimate error of N as a linear combination of $N+1$-th order logics and demonstrate correctness.

Then, we can recursively apply this logic to demonstrate that an LLM can only reason about n-Order logic with error associated with $n+1$-Order logic. The uncanniness experienced during interactions with LLM come down to the randomness in the error in the $n+1$-Order and the idealizations that could be made from them.

1.1 The Language Generation Process

A chat discourse is an ordered a set of prompts $p_i \in P \subset \mathcal{P}$ from prompt space. This is transformed into ordered results $r_i \in R \subset \mathcal{R}$.

$$r_i = Q(N(p_i))$$

Note that we assume $\mathcal{R} = \mathcal{P}$ throughout. This is an idealization as not all prompts are produdcible. Fixed point arithmetic is finite. For the sake of arguement we assume real numbers which would make this assumption less conspicuous.

We decompose each p_i into tokens $t_{i,j}$. According to the formalism, we can compute some idealized token $\hat{t}_{i,j} = Q(N(t_{i,j})) + \zeta$. It is clear that $N(t_{i,j})$ is representable in $\mathrm{Im}(Q)$. As a result, we say the $t_{i,j}$ is *expressable* by N.

We posit that there exists basis prompts $b_{i,j}$ such that for some token $t_{i,j} \in b_{i,j}$, exactly one node in the neural net is activated and that the norm of the activation is minimal. We can select this vector due to the finite nature of the net.

$$Q(N(b_{i,j})) = \hat{b}_{i,j} + \zeta.$$

This allows the network to construct an approximation of some arbitrary result $\tilde{t}$ as:

$$\tilde{t} = \sum_{i=1}^{n} \sum_{j=1}^{d} a_{i,j} b_{i,j}$$

where a is just the scaling of the basis vector to achieve the appropriate activation. In this case i is the prompt id and j is the token id.

And, it precisely $\hat{t} - \tilde{t}$ that we wish to quantify to understand how "intelligent" the model really is. In the next section, we derive $\hat{b}$.

1.2 The Discretization of Logic

Let $\mathcal{L}_n$ be the space of n-th order logical predicates. We define the $(n{+}1)$-st order logic space as:

$$\mathcal{L}_{n+1} = \text{Pred}(\mathcal{L}_n) = \{\varphi : \mathcal{L}_n \to \mathbb{B}\}$$

That is, each predicate $\varphi \in \mathcal{L}_{n+1}$ operates on a predicate $\psi \in \mathcal{L}_n$ and returns a Boolean value.

If we consider structured compositions of logic, we generalize this to:

$$\varphi \in \text{Pred}(\mathcal{L}_n^k) = \left\{\varphi : \mathcal{L}_n^k \to \mathbb{B}\right\}$$

This represents higher-order predicates over tuples of lower-order predicates, supporting structured reasoning over prompts.

The discretization error introduced at level $n{+}1$ stems from the inability to fully resolve the truth value of φ due to the limited expressivity of $\mathcal{L}_n$. We aim to bound this error in terms of projection residuals at each level.

Consider some token $t_{i,j} \in p_i$. That token is said to be of approximate order n for $n = \arg\max_i t_{i,j}$. Let $p \in \text{Pred}(\mathcal{L}_{\setminus+\infty})$.

Now consider some exhaustive space of experessable knowledge

$$\mathcal{K} = \bigcup_{i=0}^{\infty} Pred(\mathcal{L}_\rangle)$$

with some piece of knowledge $\hat{k} \in \mathcal{K}$ written as a linear combination over infinite-order logic:

$$\hat{k} = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \hat{k}_{i,j}, \quad \hat{k}_{i,j} \in \text{Pred}(\mathcal{L}_i)$$

i the fact number of that order, either semantically or syntactically, and j the order of logic of the predicate.

1.3 The Leap of Faith

At this point, the identical looking constructions of the piece of knowledge and its representation suggests a mapping between being able to construct

prompt ids and enumerating all predicates of order n. Let predicates in $\mathcal{L}_n$ be enumerated by a bijective mapping:

$$\mathrm{id}_{\mathcal{L}} : \mathrm{Pred}(\mathcal{L}_n) \to \mathbb{N}$$

Assuming a canonical encoding (e.g., Gödel numbering or any prefix-free code), each predicate $\varphi \in \mathcal{L}_n$ has a unique integer ID.

We now define the prompt construction function Π that maps a sequence of predicates to a sequence of tokens $t_{i,j}$, where:

$$p_i = \Pi(\varphi_{i,1}, \ldots, \varphi_{i,d}) = (t_{i,1}, \ldots, t_{i,d})$$

Then:

$$\mathrm{id}_{\mathrm{prompt}}(p_i) = \mathrm{concat}\left(\mathrm{id}_{\mathcal{L}}(\varphi_{i,1}), \ldots, \mathrm{id}_{\mathcal{L}}(\varphi_{i,d})\right)$$

where concat may be defined via a pairing function (e.g., Cantor pairing or Gödel encoding) to map the tuple to a unique $\mathbb{N}$.

This defines the prompt space $\mathcal{P}$ as a constructive image of $\bigcup_n \mathrm{Pred}(\mathcal{L}_n)$, with the *token position* representing *predicate order*, and *prompt ID* tied to *predicate enumeration*. Thus, entropy in token positions (within a prompt and within prompt space) can be ordered to reflect the Order-ness of the logic necessary to represent it in a neural net.

Therefore, we can talk about the error applied by the network being the difference of its internal representation ($N(\tilde{k})$) from the actual representation ($\hat{k}$) by first computing its image in N).

$$N(\hat{k}) = N\left(\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\hat{k}_{i,j}\right) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}N(\hat{k}_{i,j})$$

And we can discuss the error of $N(\hat{k}) - N(\tilde{k})$, which we call "conceptual error." It is a measure of the ability of the model to distinguish between concepts. Computing $N(\hat{k}_{i,j})$ gives:

$$N(\hat{k}_{i,j}) = \sum_{i=0}^{\infty}\sum_{i=0}^{\infty}a_{i,j}b_{i,j}.$$

This can be substituted back in to give the quadruple sum for $N(\hat{k})$:

$$N(\hat{k}) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\sum_{x=0}^{\infty}\sum_{y=0}^{\infty}a_{x,y,i,j}b_{i,j}.$$

This definse a unique tensor operator A_k we call the *embedding tensor of fact k*. We posit that $||A||$ is inversely proportional to "intelligence" and recommend scoring neural nets based on $||A^{-1}||$.o

And we can discuss the error of $N(\hat{k}) - N(\tilde{k})$, which we call the *conceptual error*. It measures the model's ability to distinguish between concepts. Computing $N(\hat{k}_{i,j})$ gives:

$$N(\hat{k}_{i,j}) = \sum_{x=0}^{\infty} \sum_{y=0}^{\infty} a_{x,y}^{(i,j)} b_{x,y}.$$

This can be substituted back in to give the quadruple sum for $N(\hat{k})$:

$$N(\hat{k}) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{x=0}^{\infty} \sum_{y=0}^{\infty} a_{x,y}^{(i,j)} b_{x,y}.$$

This defines a unique tensor operator A for each fact $\hat{k}$. The norm $\|A\|$ grows with the model's inability to resolve or compress the fact accurately.