Language Models and Cross-Entropy: Arrows of Time and the Life of Games

Clément Hongler

works with

Andrew Emil

Vassilis Papadopoulos

Jérémie Wenger

Predictions, Scoring Rules, and Information

Setting: a random observation $i \in \{1, \dots, n\}$

Prediction: $\vec{\pi} \in \Delta_n$ ← Simplex $\{\vec{\pi} \in [0,1]^n : \sum \pi_i = 1\}$

Scoring rule: $r : \{1, \dots, n\} \times \Delta_n \to \mathbb{R}$

Upon i being observed, reward $\vec{\pi}$ with $r(i, \vec{\pi})$

Example (bad!): $r(i, \vec{\pi}) = \pi_i$ "True prob."

Expected predictor's reward: $E(\vec{\pi}) = \sum_{i=1}^{n} p_i \, r(i, \vec{\pi})$

Proper scoring rule: $E(\vec{\pi})$ is maximized when $\vec{\pi} = \vec{p}$

Examples: quadratic: $r(i, \vec{\pi}) = -(1-\pi_i)^2 - \sum_k \pi_k^2 = -\|\vec{\pi} - \delta_i\|^2$

quartic: $4\pi_i^3 - 3\sum_k \pi_k^4$ cross-entropy (xent): $r(i, \vec{\pi}) = \log \pi_i$

Proper scoring rule classification (Savage, ...)

$$r(i, \vec{\pi}) = G(\vec{\pi}) + \langle \delta_i - \vec{\pi}, \nabla G(\vec{\pi}) \rangle \quad (G \text{ convex})$$

Xent is special: it is the only local proper scoring rule

$r(i, \vec{\pi})$ only depends on π_i

Xent leads us to information theory...

Language Models, Xent, and Compression

Given n successive tokens ($\approx$ words) $x_1, \dots, x_n \in \mathcal{V}$ in a text, an autoregressive LLM outputs a prediction $(\pi_x^{n+1})_{x \in \mathcal{V}}$ for the next token x_{n+1}

Cross-Entropy (Xent) loss: (negative reward)

$$\underbrace{-\sum_{n=0}^{N} \log \pi_{x_{n+1}}^{n+1}}_{\text{Estimate of } \log \mathbb{P}\{X_1 = x_1, \dots, X_N = x_N\}}$$

$\pi_{x_{n+1}}^{n+1}$: Estimate of $\mathbb{P}\{X_{n+1} = x_{n+1} \mid X_1 = x_1, \dots, X_n = x_n\}$

The xent loss corresponds to the compressed size ($\approx$ shortest description) of a long text, if we use the LLM measure as an a priori: knowing the first n tokens, we can use e.g. arithmetic encoding to encode the $n+1$-st, using (in expectation)

$$\underbrace{-\sum_{x \in \mathcal{V}} p_x^{n+1} \log_2 \pi_x^{n+1}}_{\text{Minimized when } \pi^{n+1} = p^{n+1}} \text{ bits}$$

The pre-training process for LLMs consists in minimizing the xent over large sets of texts.

The idea that the most concise description leads to intelligence is suggested by Occam's Razor, and the works of Shannon, Kolmogorov, Solomonoff, Chaitin.

In practice, when the optimization is performed on transformers, the results are incredibly good!

Next-Token Prediction:

Once upon a time, there was a $\boxed{?}$ ...

$\rightarrow$ Estimate $\mathbb{P}\{X_k = x_k \mid X_1 = x_1, \dots, X_{k-1} = x_{k-1}\}$ $\forall k = 1, \dots, n$

$\wr$

Forward Model $\mathcal{M}^{\rightarrow}$:

$$\mathbb{P}^{\rightarrow}\{X_1 = x_1, \dots, X_k = x_k\} = \prod_{k=1}^{n} \mathbb{P}^{\rightarrow}\{X_k = x_k \mid X_1 = x_1, \dots, X_{k-1} = x_{k-1}\}$$

Previous-Token Prediction:

... $\boxed{?}$ and they lived happily ever after.

$\rightarrow$ Estimate $\mathbb{P}\{X_k = x_k \mid X_{k+1} = x_{k+1}, \dots, X_n = x_n\}$

$\wr$

Backward Model $\mathcal{M}^{\leftarrow}$:

$$\mathbb{P}^{\leftarrow}\{X_1 = x_1, \dots, X_n = x_n\} = \prod_{k=1}^{n} \mathbb{P}^{\rightarrow}\{X_k = x_k \mid X_{k+1} = x_{k+1}, \dots, X_n = x_n\}$$

Information and Time Reversibility

Can an LLM learn to speak backwards?

Train on data with a reversed time direction!

Useful for many things (reverse prompting, etc)

Train FW and BW models: same, except time

Xent answer: optimal compression is time flip-invariant

Probability answer:

$$\mathbb{P}\{X_1=x_1\}\cdot\mathbb{P}\{X_2=x_2|X_1=x_1\}\cdot\ \dots\ \cdot\mathbb{P}\{X_N=x_N|X_{<n}=x_{<n}\}$$
$$=\mathbb{P}\{X_N=x_N\}\mathbb{P}\{X_{N-1}=x_{N-1}|X_N=x_N\}\cdots\mathbb{P}\{X_1=x_1|X_{>1}=x_{>1}\}$$

Compare forward and backward LLMs with same data

- Information-theoretically: no difference
- If we memorize dataset: no difference
- Naively: we can't speak backwards, so that LLMs probably can't either
- Shannon: interesting question, unclear
- Google rumors (circa 2014): no difference
- Some (not well-cited) papers: backword easier

Shannon's Experiments: Next- and Previous-Letter Prediction

Prediction and Entropy of Printed English

By C. E. SHANNON

(*ManuscriptReceived Sept. 15, 1950*)

A new method of estimating the entropy and redundancy of a language is described. This method exploits the knowledge of the language statistics possessed by those who speak the language, and depends on experimental results in prediction of the next letter when the preceding text is known. Results of experiments in prediction are given, and some properties of an ideal predictor are developed.

→ The idea of measuring the sum of cross-entropies in natural languages was pioneered by Shannon. He noted that this could also be done backwards...

One experiment was carried out with "reverse" prediction, in which th[e] subject guessed the letter preceding those already known. Although th[e] task is subjectively much more difficult, the scores were only slightly poor[er]. Thus, with two 101 letter samples from the same source, the subject ob[-]tained the following results:

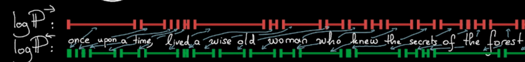
No. of guess		1	2	3	4	5	6	7	8	>8
Forward	....	70	10	7	2	2	3	3	0	4
Reverse	....	66	7	4	4	6	2	1	2	9

Incidentally, the N-gram entropy F_N for a reversed language is equal t[o] that for the forward language as may be seen from the second form in equa[-]tion (1). Both terms have the same value in the forward and reversed case[s].

Experiments were performed on human subjects; Shannon noted that to his surprise, they would perform worse predicting backwards, but only slightly so. ←

Training: Minimize Cross-Entropy Losses

$$\ell_{CE}^{\rightarrow}=\sum_{k=1}^{n}-\log\mathbb{P}^{\rightarrow}\{X_k=x_k|X_1=x_1,\dots,X_{k-1}=x_{k-1}\}$$
$$=-\log\mathbb{P}^{\rightarrow}\{X_1=x_1,\dots,X_n=x_n\}$$



$$\ell_{CE}^{\leftarrow}=\sum_{k=1}^{n}-\log\mathbb{P}^{\leftarrow}\{X_k=x_k|X_{k+1}=x_{k+1},\dots,X_n=x_n\}$$
$$=-\log\mathbb{P}^{\leftarrow}\{X_1=x_1,\dots,X_n=x_n\}$$

↳ if $\mathbb{P}^{\rightarrow}=\mathbb{P}^{\leftarrow}$ then we should have $\ell_{CE}^{\rightarrow}=\ell_{CE}^{\leftarrow}$

Universal Arrows of Time for LLMs
(w/ V. Papadopoulos & J. Wenger)

Compare FW and BW xent losses at end of training:

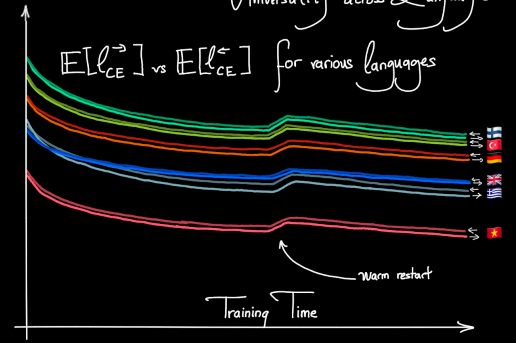
- For all human languages FW loss < BW loss
- The difference increases with the context length
- For small models pre-2017, the effect is tiny or reversed
- As the model sizes increase, the AoT becomes large

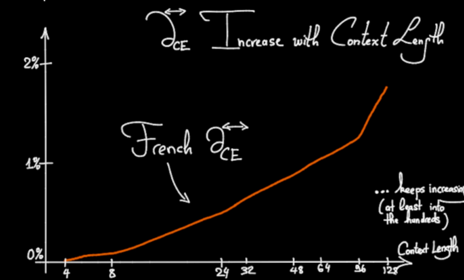
(Work by Zhang, Bai, Gu, Zhang, Gu, Abbé, Bengio, Jaitly)

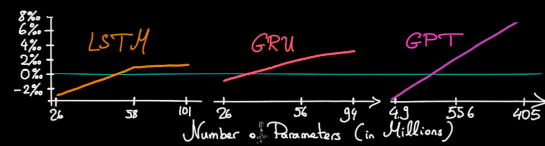
- The effect is related to the semantics, not the syntax
- The effect strength depends on the language (why?)
- The same effect can be observed on code (w/ Y. Romaniv)
- This effect cannot be observed on DNA
- The effect is stronger on LLM-generated data

Where do AoTs come from?

Universality across Languages







AoTs and Computational Hardness

A simple-to-understand example:

Dataset: a multiplication table $p \times q = pq$ for primes $p < q$

- The FW model predicts RHS from LHS by multiplying
- The BW model predicts LHS from RHS by factoring

Modern LLMs can learn to multiply well; factoring is hard

Numerical Example $\quad p < q, \; p \cdot q < 10^5$

	p	q	pq (reversed)
FW	8.98	8.67	4.55
BW	0.02	8.41	21.56

Similarly $pq = p \times q$ exhibits a reverse direction...

Why do FW AoTs arise spontaneously?

AoTs via Sparsity Symmetry Breaking

Idea: the reverse of a sparse circuit is (generically) sparse, but not as sparse (hard to prove, but intuitive)

Linear $\underbrace{\text{language}}$ dataset: "$x \longleftrightarrow y$"

where $x, y \in \mathbb{F}_2^m$ are made of m i.i.d. bits

$x = f^{\leftarrow} y$, $y = f^{\rightarrow} x$, for fixed bijective matrices $f^{\rightarrow}, f^{\leftarrow} : \mathbb{F}_2^m \to \mathbb{F}_2^m$

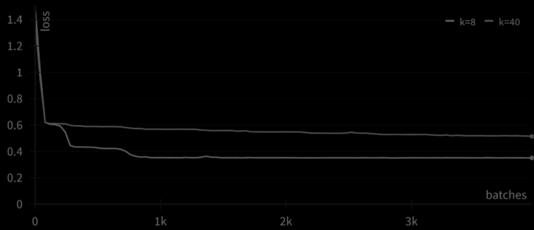
- The LHS of "$\longleftrightarrow$" determines the RHS and vice versa
- If a random $f^{\rightarrow}$ is sparse $f^{\leftarrow}$ is typically less sparse

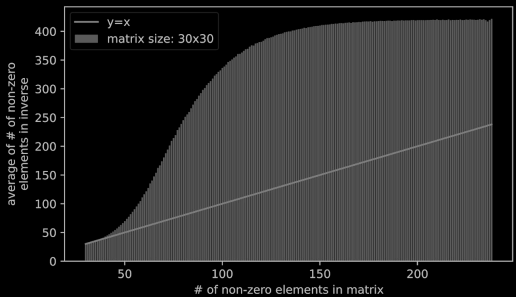
Now: sparsity $\leftrightsquigarrow$ ease of learning

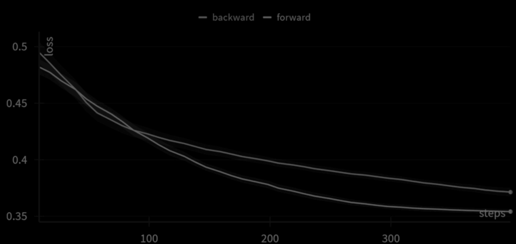
So: if we condition a linear language to be sparse FW, it is less sparse BW

Similarly, if we perform an update to a lin. lang. that is FW sparse, it is less BW sparse

Suppose Alice & Bob are FW agents with a common language $\mathcal{L}$, and Carol is a BW agent who knows $\mathcal{L}$. Now, if Alice updates $\mathcal{L}$ sparsely, the update for Carol will typically be less sparse $\rightsquigarrow$ AoT!







AoT Mysteries and Perspectives

- Can we make sense of the following idea:

Theoretically:

$$\log \mathbb{P}\{\text{initial configuration}\} + \sum_{\text{step}} \log \mathbb{P}\{\text{step} \mid \text{past}\}$$

$$= \underbrace{\log \mathbb{P}\{\text{final configuration}\}}_{\text{Covered by entropy creation ?}} + \underbrace{\sum_{\text{step}} \log \mathbb{P}\{\text{step} \mid \text{future}\}}_{\text{Covered by algorithmic AoT ?}}$$

- Can we find algorithmic AoTs with small datasets?
- Do we have AoTs in animal communication?

Maybe not?

- Can algorithmic AoTs arise in physical settings?
- Is the manifestation of AoTs a sign of "life"?
- Are there systems exhibiting reverse AoTs?
- Can we unify with the vision with diffusion models?