Inductive Biases and Variable Creation in Self-Attention Mechanisms

Benjamin L. Edelman

Surbhi Goel

Sham Kakade

Cyril Zhang

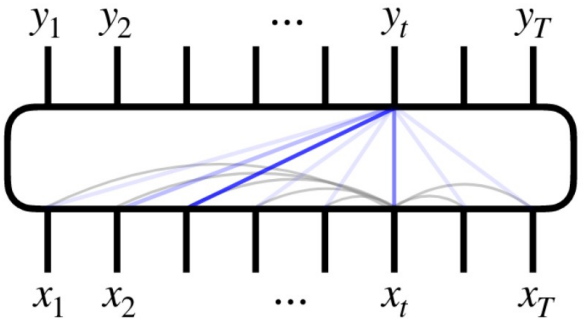
Harvard SEAS

Microsoft Research NYC

ICML | 2022

The Self-Attention Revolution

Language

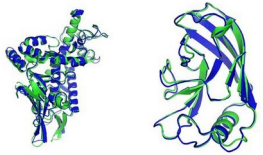


The Self-Attention Revolution

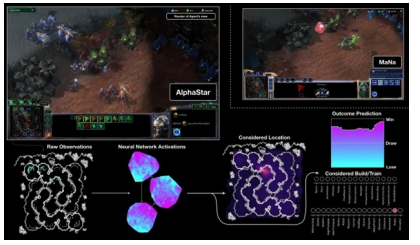
Language

OpenAI GPT-3

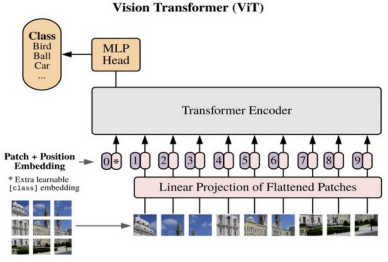
Computational biology



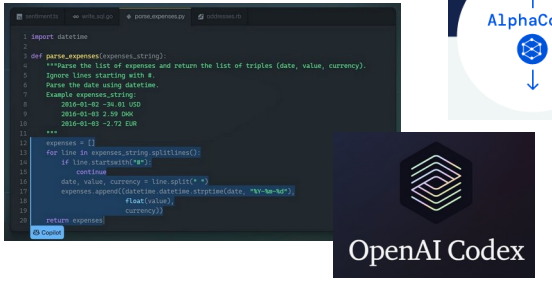
Reinforcement learning



Computer vision



Automated programming



Symbolic Mathematics

$$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}.$$

etc.

The Self-Attention Revolution

Language

Computational biology

Reinforcement learning

Why?

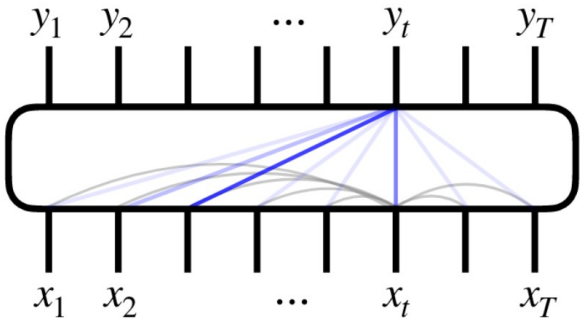
Computer vision

Automated programming

Symbolic Mathematics

$$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}.$$

etc.



What inductive biases does self-attention have?

Inductive biases of attention

Generalization: Can we prove Transformers won't overfit?

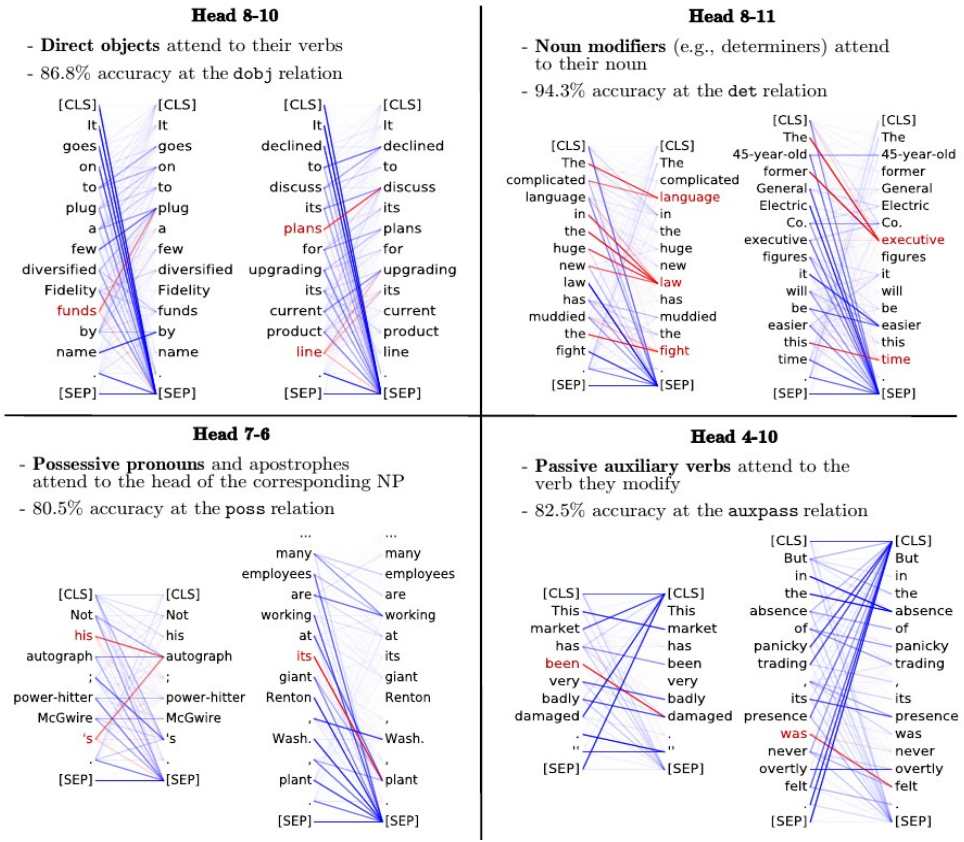
Representation: What functions can Transformers approximate?

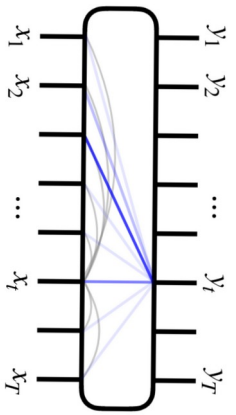
Optimization: Do Transformers learn these functions in practice?

Theory

Experiments

Inductive biases of attention



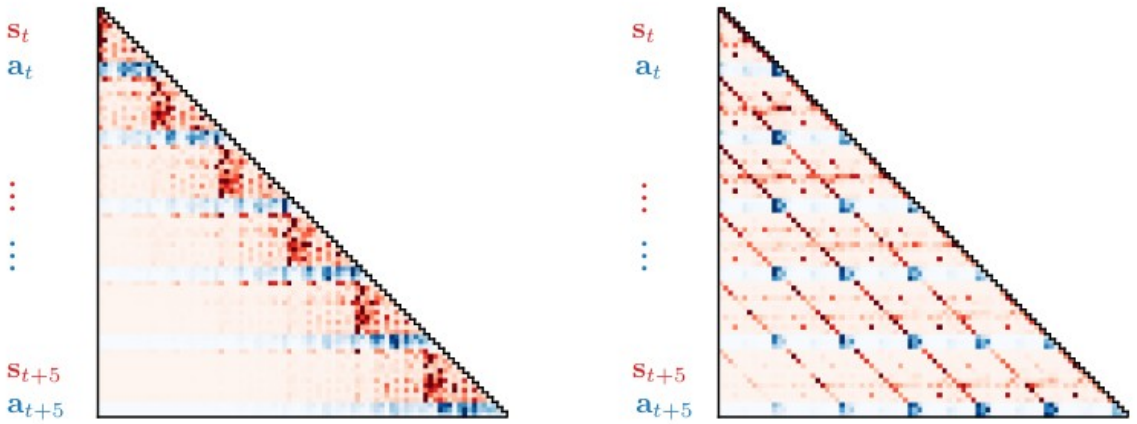


Attention weights are sparse (or close to uniform)

Source: "What Does BERT Look At? An Analysis of BERT's Attention"

Clark, Khandelwal, Levy, Manning, 2019

Inductive biases of attention



Attention weights are sparse
(or close to uniform)

Source: “Offline Reinforcement Learning as One Big Sequence Modeling Problem”
Janner, Li, Levine

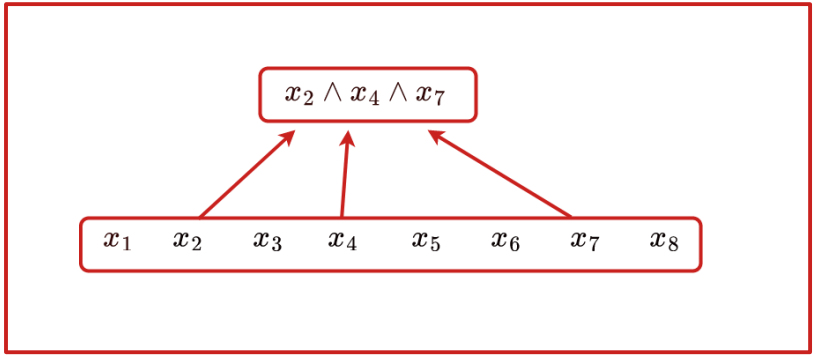
Main result: Sparse variable creation

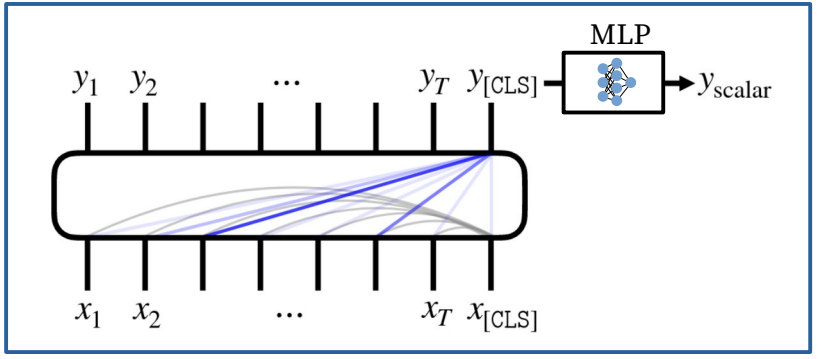
The class of ***s*-sparse functions of length-*T* inputs**

can be learned by

the class of **Transformers layers with weight norms $2^{O(s)}$**

with **sample complexity scaling as log(*T*)**





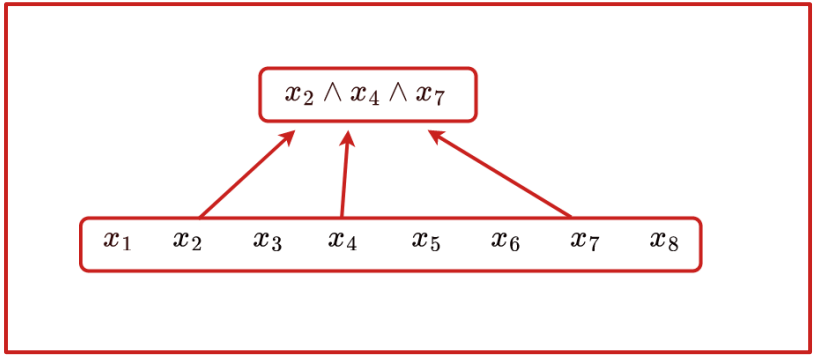
Main result: Sparse variable creation

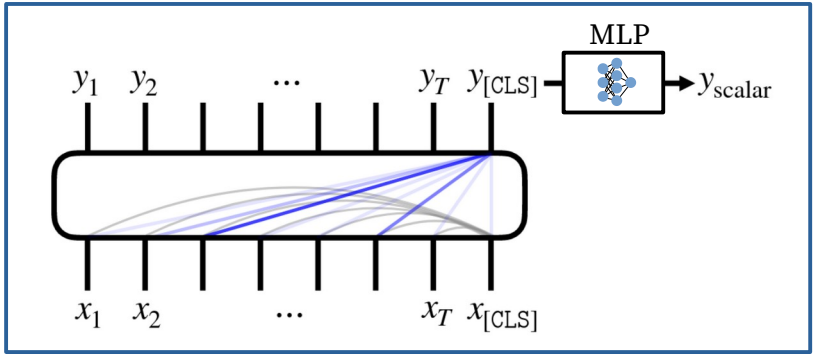
The class of ***s*-sparse functions of length-*T* inputs**

can be learned by

the class of **Transformers layers with weight norms $2^{O(s)}$**

with **sample complexity scaling as log(*T*)** ← **optimal**





Generalization result

When learning any bounded distribution using a Transformer layer, w.p. 99%,

$$\left|\text{error on training set} - \text{error on distribution}\right| = \widetilde{O}\left(\sqrt{\frac{\text{poly}\left(C\right)\cdot\log\left(T\right)}{m}}\right).$$

Norms of attention & MLP weight matrices

Number of training samples

Sequence length

Generalization result

When learning any bounded distribution using a Transformer layer,

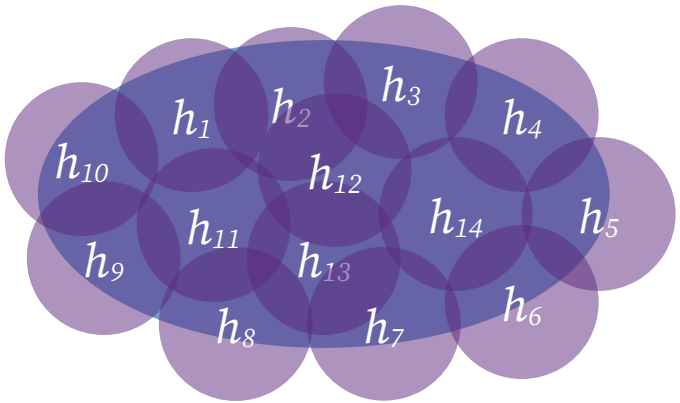
$$\left|\text{error on training set} - \text{error on distribution}\right| = \widetilde{O}\left(\sqrt{\frac{\text{poly}\left(C\right)\cdot\log\left(T\right)}{m}}\right).$$

Norms of attention & MLP weight matrices

Number of training samples

Sequence length

Technique: ∞-covering numbers



Generalization result

When learning any bounded distribution using a Transformer layer,

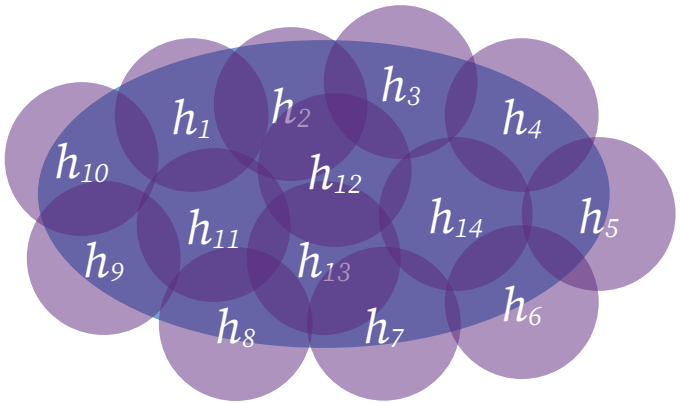
$$\left|\text{error on training set} - \text{error on distribution}\right| = \widetilde{O}\left(\sqrt{\frac{\text{poly}\left(C\right)\cdot\log\left(T\right)}{m}}\right).$$

Norms of attention & MLP weight matrices

Number of training samples

Sequence length

Technique: ∞-covering numbers

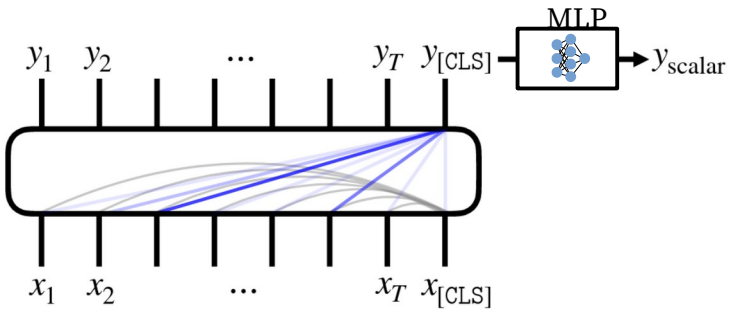


*original draft had dependence on embedding dimension d

Representation result

Any s-sparse Boolean function f can be exactly represented by a Transformer layer with weight norms $2^{O(s)}$.

If f is symmetric, only poly(s) weight norms are required.



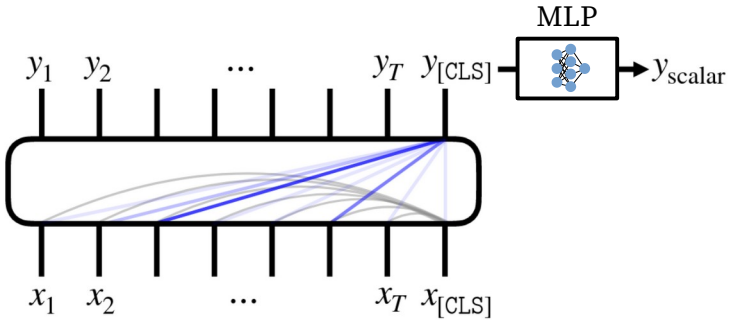
Representation result

Any s-sparse Boolean function f can be exactly represented by a Transformer layer with weight norms $2^{O(s)}$.

If f is symmetric, only poly(s) weight norms are required.

Intuition

- `Softmax` allows sparse variable selection
- MLP allows arbitrary function to be applied



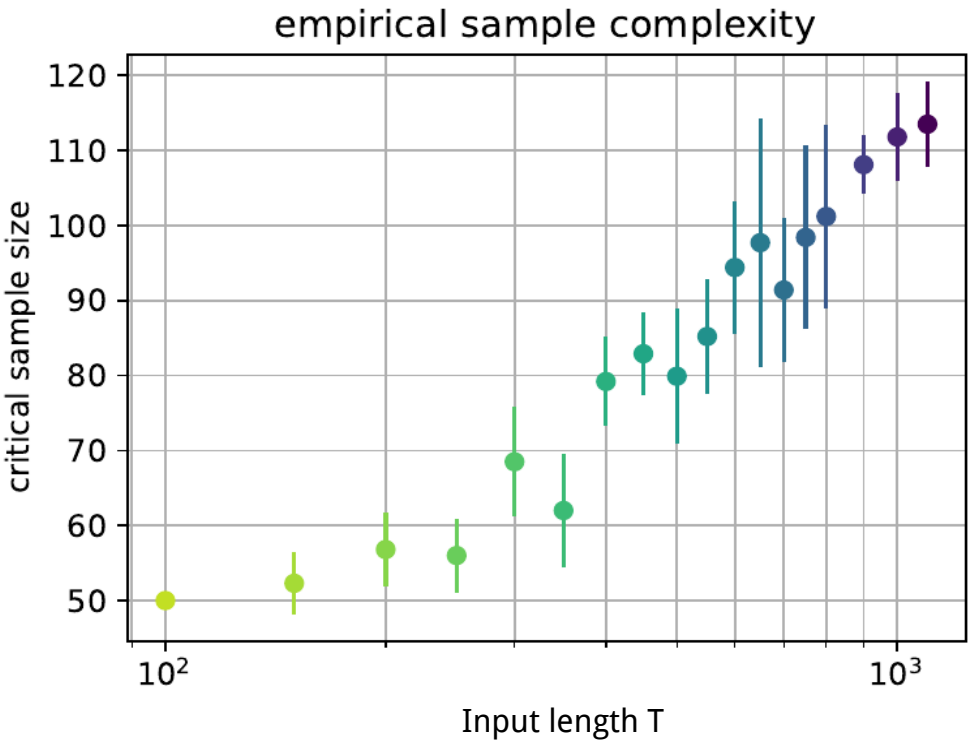
Optimization (sparse conjunctions)

$$\vec{x} \sim \text{unif}\left(\{0,1\}^T\right)$$
$$y = x_{i_1} x_{i_2} x_{i_3}$$

Train a Transformer layer

- As input length T grows, how large does the training set need to be to avoid overfitting?

- Consistent with $\log(T)$ dependence in generalization bound!



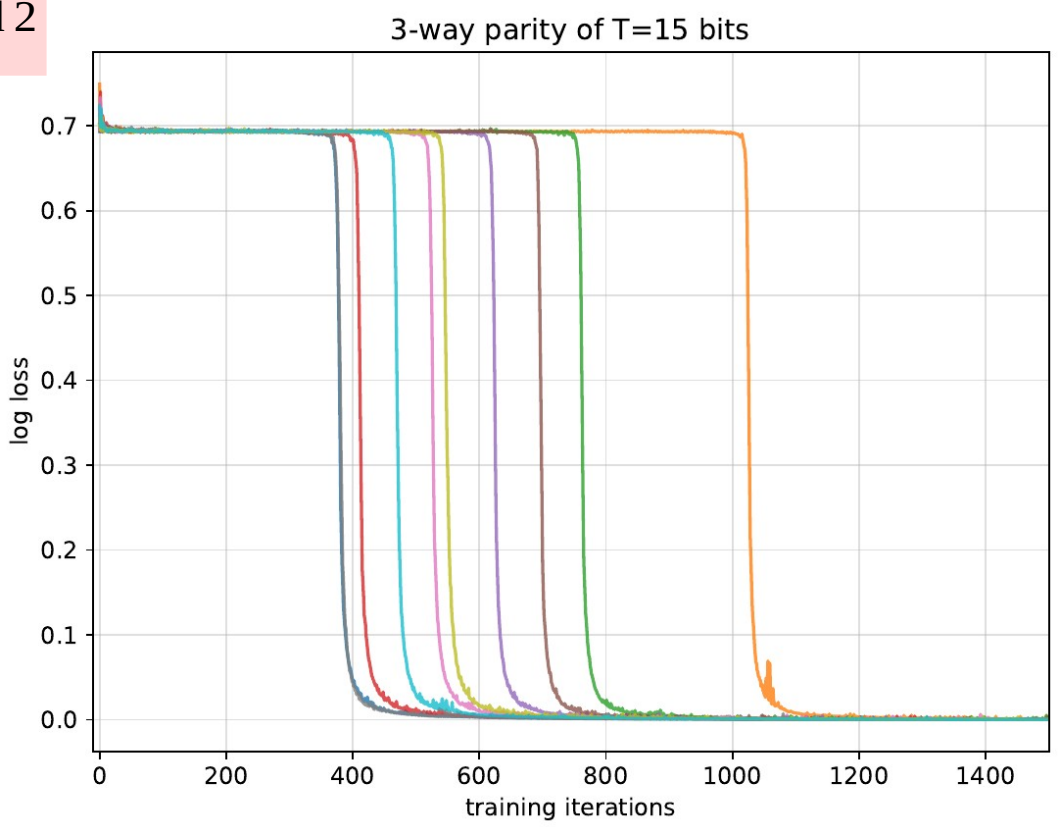
Further contributions

- We define a **general class of attention mechanisms** that includes standard Transformer attention as a special case, and prove a general-purpose generalization bound
  - $l_1 \to l_\infty$ **smoothness** of `softmax` is crucial for the generalization result,

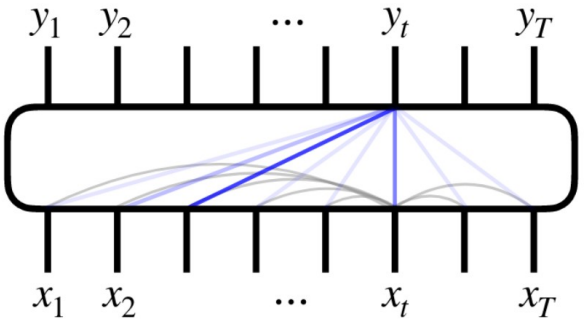
    and the fact that `softmax` approximates `max` is crucial for the representation result.
- We extend our generalization bound to deep Transformer networks

Transformer layers learn sparse parities

$$\vec{x} \sim \text{unif}\left(\{0,1\}^T\right)$$
$$y = x_{i_1} + x_{i_2} + x_{i_3} \mod 2$$



Related: “Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets” Power, Burda, Edwards, Babuschkin, Misra



Thank you!