A completely uniform transformer for parity

Alexander Kozachinskiy[1]   Tomasz Steifer[2,3]

[1]Centro Nacional de Inteligencia Artificial, Chile
[2]Instituto de Ingeniería Matemática y Computacional, Universidad Católica de Chile
[3]Institute of Fundamental Technological Research, Polish Academy of Sciences

Abstract

We construct a 3-layer constant-dimension transformer, recognizing the parity language, where neither parameter matrices nor the positional encoding depend on the input length. This improves upon a construction of Chiang and Cholak who use a positional encoding, depending on the input length (but their construction has 2 layers).

1   Introduction

One of the ways to do mathematical analysis of the capabilities and limitations of the transformer architecture [8] is to study formal languages, recognizable by them. Namely, for a given formal language L, we study, if there exists a choice of parameters in the transformer architecture, for which words from L are accepted and words not from L are rejected by the resulting transformer. A seminal work of Hahn [5] performed such analysis for a number of formal languages, including the *parity language*, consisting of binary words with even number of 1s. Hahn have shown that transformers, recognizing this language, must have low confidence, partially explaining an empirically observed struggle of transformers in learning this language [1, 4].

However, this does not exclude the possibility of existence of a theoretical solution for this language. And indeed, as was shown by Chiang and Cholak [3], there exists a 2-layer transformer with constant embedding dimension, recognizing the parity language. Their construction is *uniform* in a sense that parameter matrices in it do no depend on the input length. However, there is one aspect of their construction which is not completely uniform – the *positional encoding*. At position i, they use i/n, where n is the input length. This means that their positional encoding has to be reset each time we want to use their transformer for a larger input length.

In this paper, we get rid of this disadvantage of their construction by giving a *completely uniform* 3-layer transformer, recognizing parity. That is, in our construction, neither parameter matrices nor positional encoding depend on n. We do not use neither positional masks nor layer norm.

It remains open if the parity language can be recognized by a 1-layer transformer with constant embedding dimension, even non-uniformly. Current lower bound methods against 1-layer transformers [2, 6, 7] do not seem to work against parity.

2   Preliminaries

An attention layer of dimension d is a length-preserving function $A\colon (\mathbb{R}^d)^* \to (\mathbb{R}^d)^*$, given by three matrices $K, Q, O \in \mathbb{R}^{d \times d}$ and a neural network $\mathcal{N}\colon \mathbb{R}^d \to \mathbb{R}^d$ with ReLU activation. On input $(f_1, \ldots, f_n) \in (\mathbb{R}^d)^n$,

1

the output $A(f_1, \ldots, f_n)$ is computed as follows. First, we define:

$$a_i = \frac{\sum_{j=1}^{n} e^{\langle Kf_j, Qf_i \rangle} f_j}{\sum_{j=1}^{n} e^{\langle Kf_j, Qf_i \rangle}}, \qquad i = 1, \ldots, n,$$

and set:

$$A(x_1 \ldots x_n)_i = \mathcal{N}(f_i + Oa_i), \qquad i = 1, \ldots, n.$$

Applying $\mathcal{N}$ allows us to do arbitrary piece-wise linear transformation of $\mathbb{R}^d$.

Definition 1. *We say that a language $L \subseteq \{0,1\}^*$ is recognizable by a completely uniform transformer with C layers if there exists $d \in \mathbb{N}$, C attention layers $A_1, \ldots, A_C$ of dimension d, and a letter embedding $\ell \colon \{0,1\} \to \mathbb{R}^d$, and a position encoding $p \colon \mathbb{N} \to \mathbb{R}^d$ such that for any $n \in \mathbb{N}$, and for any $x = x_1 x_2 \ldots x_n \in \{0,1\}^n$, the following holds. Define:*

$$f_1 = \ell(x_1) + p(1), \ldots, f_n = \ell(x_n) + p(n),$$

and set

$$(g_1, \ldots, g_n) = A_C \circ \ldots \circ A_1(f_1 \ldots f_n).$$

Then the following must hold. If $x \in L$, we have $g_1^1 > 0$, and if $x \notin L$, we have $g_1^1 < 0$.

Note that in our definition, the positional encoding p does not take n, the input length, as an input.

3   Construction

Lemma 1. *For any function $f \colon \mathbb{N} \to \mathbb{R}$, there exists a completely uniform transformer that, for any input length n, computes $f(n)$ in every position in one layer.*

Proof. We need a positional encoding $p \colon \mathbb{N} \to \mathbb{R}$ such that:

$$\frac{p(1) + \ldots + p(n)}{n} = f(n),$$

for every n (this average is computable via softmax with uniform weights), which is achievable by setting $p(1) = f(1)$ and $p(i) = if(i) - (i-1)f(i-1)$ for $i \geq 2$, $\qquad \qquad \square$

Theorem 1. *There exists a 3-layer completely uniform transformer, recognizing the parity language.*

Proof. Assume that on input we get $x = x_1 \ldots x_n \in \{0,1\}^n$. Denote $\Sigma = x_1 + \ldots + x_n$. Let us first give a construction, assuming that $\Sigma \geq 1$, that is, that not all input bits are 0. We explain how to get rid of this assumption in the end of the proof.

The plan of the proof is as follows. At the first two layers, we need to compute a sequence of numbers

$$a_1, \ a_2, \ldots, \ a_n,$$

with the property that a_Σ is strictly larger than any other number in the sequence. Assuming we have done this, at the last layer we can compute the following:

$$\frac{\sum_{i=1}^{n} \exp\{a_i f(n)\} \cdot (-1)^i}{\sum_{i=1}^{n} \exp\{a_i f(n)\}}. \tag{1}$$

using a positional encoding $i \mapsto (-1)^i$ and Lemma 1 for a function $f\colon \mathbb{N} \to \{0,1\}$ of our choice. Our goal for (1) is to encode parity, i.e., (1) must be positive if Σ is even and negative otherwise. Indeed, the limit of (1) as $f(n) \to +\infty$ is $(-1)^\Sigma$, because $a_\Sigma > a_i$ for $i \neq \Sigma$. For any fixed n, there are finitely many inputs of length n. Hence, we can take $f(n)$ large enough so that, for any input x of length n, the difference between (1) for this x and $(-1)^\Sigma$ is at most $1/3$.

It remains to calculate a sequence $a_1, \ldots, a_n$ with this property at the first two layers. By Lemma 1, we compute $\ln n$ at every position in the first layer. Then, for an absolute constant $\delta \in \mathbb{R}$, to be specified later, we set $\alpha = e^\delta$ and use the second layer to compute the following expression at every position:

$$
\gamma = \frac{\sum\limits_{i=1}^{n} e^{(-\ln(n)+\delta)\cdot(1-x_i)} \cdot x_i}{\sum\limits_{i=1}^{n} e^{(-\ln(n)+\delta)\cdot(1-x_i)}} = \frac{\Sigma}{\Sigma + (n-\Sigma)\cdot\frac{\alpha}{n}} = \frac{1}{1 + \frac{\alpha}{\Sigma} - \frac{\alpha}{n}}.
\tag{2}
$$

It now suffices to prove the following lemma.

Lemma 2. *There exists $\alpha \in (0,1)$ such that for all n and $\Sigma \in \{1, \ldots, n\}$, the maximum of the expression:*

$$
a_i = -\left| \gamma - \left(1 - \left(\frac{\alpha}{i} - \frac{\alpha}{n}\right)\right) \right|
$$

over $i \in \{1, \ldots, n\}$ is attained uniquely at $i = \Sigma$, where γ is defined by (2).

Indeed, once we have computed γ, we can express a_i by computing $\frac{1}{n}$ in every position by Lemma 1, and using a positional encoding $i \mapsto \frac{1}{i}$. We can then calculate the absolute value using a constant-size ReLU network.

Proof of Lemma 2. Observe that:

$$
\left| \frac{1}{1+z} - (1-z)) \right| = \frac{z^2}{1+z} \leq z^2
$$

for $z \geq 0$. Applying this to $z = \frac{\alpha}{\Sigma} - \frac{\alpha}{n}$, we get:

$$
\gamma = 1 - \left(\frac{\alpha}{\Sigma} - \frac{\alpha}{n} \right) + \rho,
$$

for some ρ with $|\rho| \leq \frac{\alpha^2}{\Sigma^2}$. We now can write:

$$
a_i = -\left| \left(\frac{\alpha}{i} - \frac{\alpha}{\Sigma} \right) + \rho \right|
$$

Observe that $a_\Sigma = -|\rho|$. To finish the proof it suffices to establish that

$$
\left| \frac{\alpha}{i} - \frac{\alpha}{\Sigma} \right| > 2|\rho|, \qquad \text{for } i \neq \Sigma.
\tag{3}
$$

Indeed, from (3), we get

$$
a_i \leq -\left| \frac{\alpha}{i} - \frac{\alpha}{\Sigma} \right| + |\rho| < -2|\rho| + |\rho| = a_\Sigma,
$$

for $i \neq \Sigma$. We now establish Σ. If $i \neq \Sigma$ and $i < 2 \cdot \Sigma$, we get

$$
\left| \frac{\alpha}{i} - \frac{\alpha}{\Sigma} \right| = \left| \frac{\alpha(\Sigma - i)}{i\Sigma} \right| \geq \frac{\alpha}{2\Sigma^2} > 2\frac{\alpha^2}{\Sigma^2} = 2|\rho|,
$$

where the strict inequality is true as longs as $\alpha < 1/4$.

Now, if $i \neq \Sigma$ and $i \geq 2\Sigma$, then $i - \Sigma \geq i/2$, which gets us:

$$
\left| \frac{\alpha}{i} - \frac{\alpha}{\Sigma} \right| = \left| \frac{\alpha(i - \Sigma)}{i\Sigma} \right| \geq \frac{\alpha}{2\Sigma} > 2\frac{\alpha^2}{\Sigma^2} = 2|\rho|,
$$

where again it is enough to require that $\alpha < 1/4$. $\qquad\square$

Finally, we explain how to get rid of the assumption $\Sigma \geq 1$. Let us denote the value of the expression (1) by θ. We have that θ is positive for even $\Sigma \geq 1$, and θ is negative for odd $\Sigma \geq 1$. Now, at the first layer, we can compute the quantity $1/(2n) - (x_1 + \ldots + x_n)/n$ by taking the arithmetic mean of the input bits using sotftmax with uniform weights, and computing $1/(2n)$ by Lemma 1. Observe this quantity is positive for $\Sigma = 0$ and negative otherwise. It remains to output $\max\{\theta, 1/(2n) - (x_1 + \ldots + x_n)/n\}$ in the third layer.

$\square$

References

[1] Bhattamishra, S., Ahuja, K., and Goyal, N. On the ability and limitations of transformers to recognize formal languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2020), pp. 7096–7116.

[2] Bhattamishra, S., Hahn, M., Blunsom, P., and Kanade, V. Separations in the representational capabilities of transformers and recurrent architectures. *arXiv preprint arXiv:2406.09347* (2024).

[3] Chiang, D., and Cholak, P. A. Overcoming a theoretical limitation of self-attention. In *Annual Meeting of the Association for Computational Linguistics* (2022).

[4] Delétang, G., Ruoss, A., Grau-Moya, J., Genewein, T., Wenliang, L. K., Catt, E., Cundy, C., Hutter, M., Legg, S., Veness, J., and Ortega, P. A. Neural networks and the Chomsky hierarchy. *arXiv preprint arXiv:2207.02098* (2022).

[5] Hahn, M. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics 8* (2020), 156–171.

[6] Peng, B., Narayanan, S., and Papadimitriou, C. On limitations of the transformer architecture. *arXiv preprint arXiv:2402.08164* (2024).

[7] Sanford, C., Hsu, D. J., and Telgarsky, M. Representational strengths and limitations of transformers. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023* (2023), A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds.

[8] Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).