Yet Another Lottery Ticket Hypothesis

Aman Madaan*
ToTheMoon LLC
Pittsburgh, PA 15213
amn.madaan@gmail.com

Gary Yao *
Unicorn Crypto AI Cloud Inc.
Newark, CA 94560
gary.yao@gmail.com

Abstract

We fine-tune a pre-trained GPT-2 on a sequence historical powerball data. Despite limited data, the distribution of the generated numbers closely follows the training data distribution. Our work is the latest in the long line of works that apply deep neural networks to random problems in the hopes of hitting something big. We win a grand sum of $4 and open up new avenues of getting rich quick using deep neural networks.

1 Introduction

Language models trained on large body of text have repeatedly broken the records on multiple computational linguistic tasks in the recent years (Devlin et al., 2019; Radford et al., 2018, 2019; Brown et al., 2020). State-of-the-art language models like GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020) have billions of parameters (1.5 billion for GPT-2, 175 billion for GPT-3) and are trained over large corpora (the Internet), enabling them to capture subtle properties of the language allowing for slick demos[1] and hyped up Techcrunch articles about the AI singularity.

Applying such large networks to random problems in the hopes of beating the SOTA has received a lot of attention in the recent times. Further, numerous studies suggest that getting the most out of deep neural networks can sometimes depend on the random seed (Dodge et al., 2020; Mosbach et al., 2020) (so basically, luck). Motivated in equal parts by successes of deep neural networks and personal failures, we pose the following *research* questions: "can language models generate powerball numbers based on historical data?"

2 Methodology

Given a sequence of tokens $\{u_1, u_2, ..., u_{k-1}\}$, auto-regressive language models can be trained to efficiently estimate the next token distribution conditioned on the previous tokens: $p(u_k \mid \{u_1, u_2, ..., u_{k-1}\})$. This allows them to be essentially used as auto-completers. For example, trained on an English corpus, a language model will likely predict *milk* as the next token for the sentence *the cat drank the* ___.

We train a GPT-2 to autocomplete lottery numbers, given the information about the day and phases of moon. We obtain the past winning numbers between 1997-2020 from various sources, including the New York State Gaming Commission.[2] Table 2 shows the dataset statistics and Table 1 shows the two input-output formats that we experimented with.

2.1 Using lunar phases

Since ancient times, the phases of the moon is believed to have spiritual significance in one's life. For example, a new moon is believed to bring new beginnings and fresh starts. There are secrets to be unlocked here that can lead to potential of unlimited lotto winnings of a lifetime (or until the lottery associations decide to ban the approach). Also, we needed to fill some space.

3 Theoretical Analysis

Sir, this is a Wendy's.

4 Related work

None. This is a very original paper. Neither of the co-authors know of any paper with a similar name or idea.

* authors contributed sort of equally to this "work."
[1] https://app.inferkit.com/demo

[2] https://data.ny.gov/Government-Finance/

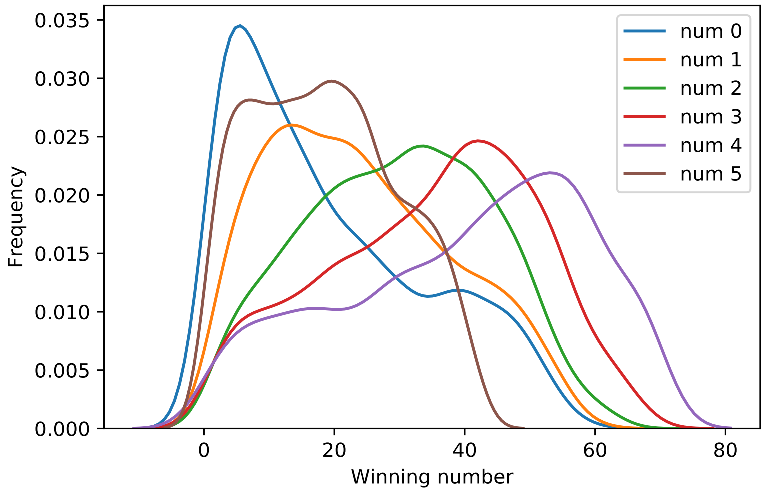

Figure 1: Distribution of winning number for each location as given by the historical data. Note that contrary to popular opinion, the numbers are not draw from a uniform distribution with numbers at earlier location showing a clear bias to take smaller values.

	Input	Output
simple	On Wed Jan 16 2019, the winning numbers were	14 29 31 56 61 01
moonphase	On Wed Jul 13 2005, the lunar phase was at 22.0%, and the winning numbers were	5 23 43 4 13 34

Table 1: Input-output formats used to train GPT-2

Split	Samples
Train	2350
Dev	82
Test	82
Total	2514

Table 2: Dataset size

5 Experiments

5.1 Baselines

We compared our approach with strong baselines shown in Table 3. We were not able to access any of these baselines but we assume that they are as random as the lottery itself, so we replicated the baselines by having these animals in our heart while purchasing the tickets generated by powerball's random picker.

5.2 Main results

Yes, we actually did train GPT-2 using the formats shown in Table 1 and actually also purchased the tickets 3.

We summarized our main observations and results next:

1. Our total winnings from our method and the baselines are $0. While the outcome is not really surprising, we take confidence in the fact that at least we were not beaten by the baselines.

2. The output was realistic: i) the model always produced valid number sequences (e.g., never produced numbers outside of the powerball range) and ii) The distribution of the generated numbers closely matches the original distribution.

3. The model was sensitive to additional information like the moon phases (i.e. changing the moon phase changed the prediction) but there was no correlation between the two (no we promise we were not expecting anything).

Table 3: **Our baselines.** Clockwise from the top-left: Mani the parrot astrologer, Goldy paws the lottery picking dog, Paul the octopus of the World cup fame, and Gray the juggler seal. Images updated using Dall-E mini[3] following a copyright notice by **PicRights International Inc.** on 7/2/2022. Someone actually read this "paper".

6 Conclusion

Training on the historical lottery data and incorporating the moon phases is successfully producing reasonably good looking numbers.

In a real world application: we went out and purchased a set 9 tickets for March 6, 2021 drawing. Comparison with the control group of 9 randomly selected numbers yielded promising results: the numbers from the trained model won a total of $4 vs. $0 from the random selection. So this totally works 100 percent of the times when it does!

As a bonus, here are some predictions on the upcoming winning lotto numbers on a few key dates:

1. **Cinco de Mayo**, time to keep the party going, why not - May 1, 2021 - **02 07 19 42 64 03**.

2. **New Moon**, it's a new beginning and a new you - October 6, 2021 - **01 25 45 60 68 06**.

3. **Christmas**, this year's presents could get a lot better - December 25, 2021 - **12 20 41 50 67 23**.

Note: Authors would like to claim 3.14 percent of the lottery winnings should you use these numbers - accepting all fiat and crypto payment methods. Good luck!

7 Acknowledgement

This work was partially supported by the stimulus checks. The authors would also like to thank the staff of the numerous retail stores and gas stations for not judging them for repeated visits.

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith.

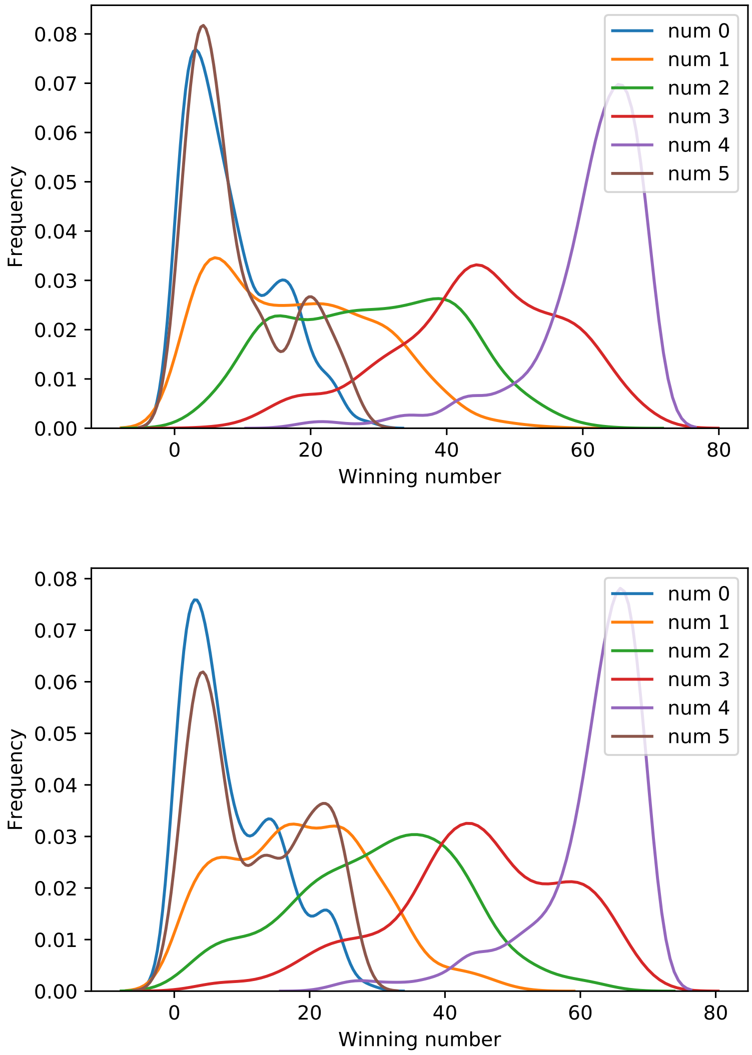

Figure 2: Distribution of winning number **generated** by our models. Comparing with Figure 1, we see that the generated numbers closely match the training data distribution.

2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. arXiv preprint arXiv:2002.06305.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. arXiv preprint arXiv:2006.04884.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI Blog, 1(8):9.

Figure 3: We actually got the tickets.