Kryptonite-2.0: A Simple End to Machine Learning Hype?

Harley Quinn [1] **Lex Luther** [2]

Abstract

Machine learning has emerged as the *de facto* gold standard approach to many computational problems across financial services, medical diagnosis, and autonomous navigation. To solve these complex data-driven problems, the last decade of research has resulted in the use of larger and larger machine learning models which has lead to significant hype and even some scientists claiming that we are nearing artificial general intelligence (AGI). In this work, we develop a simple, procedurally generated dataset that we call Kryptonite-2.0 a binary classification task which defeats the best theoretical arguments and empirical models these arguments and thus call into question the hype surrounding modern machine learning models. Our challenge task, Kryptonite-2.0, is a dataset with an n dimensional feature space and binary label space. We find that with only $n = 10$ our dataset bests the universal function approximation argument and fools GPT3 which fails to score over 60% accuracy.

1. Introduction

In recent years, machine learning (ML) has become the dominant paradigm for solving complex, data-driven problems across a wide array of industries. From financial services (Heaton et al., 2017) to medical diagnosis (Esteva et al., 2019) and autonomous navigation (Bojarski et al., 2016), the versatility and success of machine learning models are undeniable. Driven by advances in hardware, data availability, and algorithmic breakthroughs, ML models have grown increasingly complex, with larger architectures tackling ever more challenging tasks (Bommasani et al., 2021).

Amidst these developments, the hype surrounding machine learning has grown exponentially. Some researchers and technologists claim that we are approaching a point where models may achieve artificial general intelligence (AGI) (Bubeck et al., 2023), a form of intelligence that can perform any intellectual task that a human can. Supporting these claims are both theoretical results, such as the Universal Approximation Theorem (Hornik, 1991), which suggests that neural networks can approximate any continuous function, and experimental findings in large language models (LLMs) which show ML models performing tasks with human-level proficiency. For example, the chain-of-thought reasoning capabilities exhibited by models like GPT-4 have sparked excitement about the potential of such models (Wei et al., 2022).

However, the remarkable growth and capabilities of ML models should be approached with caution. Recently, pertinent stream of research seeks to simultaneously challenge and spur important growth in machine learning models: challenge datasets (Chollet, 2019; Küttler et al., 2020; Sawada et al., 2023). Such datasets are carefully crafted such that modern machine learning algorithms struggle to achieve strong performance. These tasks not only pose critical questions about the capabilities of current machine learning approaches, but serve as a target metric to spur advances in the correct direction.

In this work, we introduce Kryptonite-2.0, a simple, procedurally generated dataset designed to challenge both the theoretical and experimental claims that drive the current ML hype. Comprising an n-dimensional feature space mapped onto a binary label space, Kryptonite-2.0 is designed to be the simplest challenge dataset that addresses both theoretical and empirical claims underpinning machine learning hype.

We begin by showing that Kryptonite-2.0 is potentially a counter-example to the universal function approximation theorem. This is demonstrated with an experiment that shows Kryptonite-2.0, even with $n = 10$, is impossible to solve with polynomial basis expansion even when many features are used, model's fail to accurately classify the data. Furthermore, even the most modern LLMs such as GPT-3, when tasked with our binary classification problem also fail, achieving no more than 60% accuracy even with advanced prompt-based embeddings. These results cast doubt on the sweeping claims of universality and general intelligence in machine learning models and highlight critical limita-

[1]Wayne University of Metropolis, Gotham, United States [2]Xavier Academy for Advanced Sciences, Westchester, United States. Correspondence to: Harley Quinn, PhD <No email given>.

Coursework. For review by instructors of Mathematics for Machine Learning (70015) . Do not distribute.

tions that must be addressed if we are to move forward in a meaningful way.

We begin by discussing the most pertinent related works to Kryptonite-2.0, then we cover our experimental design and results for invalidating the universal function approximation theorem. We then describe experiments on extension of Kryptonite-2.0 called Kryptonite-GPT-2.0 which is an encoding of our dataset by a GPT model. We open source all of our experiments, the datasets, and an unlabeled set of data to ultimately benchmark potential challenge solutions. We conclude the paper with prospective approaches for solving the Kryptonite-2.0 tasks. We feel that it is ultimately a useless exercise for future researchers and that we should all abandon machine learning research in favor of... something else?

2. Related Works

It is widely accepted that the creation and curation of challenging datasets is a valuable modality of contribution to AI research. One prominent example of this is the Abstraction and Reasoning Corpus for Artificial General Intelligence (ARC-AGI) benchmark and corresponding prize (Chollet, 2019). This benchmark procedurally generates complex logic puzzles which are currently outside of the scope for foundation models. This approach has been extended to groups of concepts (Moskvichev et al., 2023) as well as directly to text where the authors attempt to make them natural language centric (Sawada et al., 2023). Another example of a prominent challenge dataset is the NetHack challenge (Küttler et al., 2020) which requires models to reason about a text-based adventure game.

This work is different from those prior works as they establish complex abstract reasoning benchmarks whereas this work is a very simple, binary classification problem that also proves remarkably challenging for machine learning models.

3. Methodology

In order to preserve the challenge of the dataset we would like to avoid models that are *a priori* suited to the task i.e., models whose features space is tailored to the task at hand. Thus, we will not release the exact inner working of Kryptonite-2.0 dataset. We do specify that there *is* an underlying pattern to the feature-label pair. That is to say the dataset is not *impossible*, just designed to be difficult for machine learning models. We denote a machine learning model as a parametric function f with parameters $\theta \in \Theta$, which maps from features $x \in \mathbb{R}^n$ to labels $y \in \{0, 1\}$. We consider supervised learning in the classification setting with a labeled dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$. The model parameters are trained, starting from some initialization θ' denoted as M, as $\theta = M(f, \theta', \mathcal{D})$. In other words, given a model, initialization, and dataset, the training function M returns the "trained" parameters θ. The function M is typically taken to be stochastic with randomness stemming from the initialization, batch ordering, and any noise added.

3.1. Binary Logistic Regression and Gradient-Based Training

Binary logistic regression is one of the most well-investigated machine learning models for binary classification, aiming to estimate the probability that a given input $\mathbf{x} \in \mathbb{R}^n$ belongs to one of two classes, typically denoted $y = 0$ or $y = 1$. Logistic regression models the probability of class membership as a function of the input features:

$$P(y = 1|\mathbf{x}; \boldsymbol{\theta}) = \sigma(\mathbf{x}^\top \boldsymbol{\theta}) = \frac{1}{1 + \exp(-\mathbf{x}^\top \boldsymbol{\theta})},$$

where $\boldsymbol{\theta} \in \mathbb{R}^n$ is the parameter vector and $\sigma(\cdot)$ denotes the logistic sigmoid function. The objective typically used to learn this model is the categorical cross-entropy loss which we will denote generically with $\mathcal{L}$.

Formulating the problem of learning as an optimization problem for a dataset $\mathbf{X} \in \mathbb{R}^{N \times n}$ with corresponding labels $\mathbf{y} \in \{0, 1\}^N$, binary logistic regression does not admit a closed form solution. Instead, we use gradient-based methods to learn the model. This is particularly appropriate in our setting as gradient descent (and its variants) is the algorithm used to learn in some of the most prominent examples of machine learning successes. Gradient descent can be stated as an iterative process with the equations:

$$\begin{aligned} \boldsymbol{\theta}^{(1)} &\sim \mathcal{N}(0, \mathbf{I}\sigma^2) \\ \boldsymbol{\theta}^{(i+1)} &= \boldsymbol{\theta}^{(i)} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}\left(\boldsymbol{\theta}^{(i)}\right) \end{aligned} \tag{1}$$

Where α is called the learning rate and we run the algorithm for a fixed number of epochs.

The gradient of the objective function in the case of our loss function respect to $\boldsymbol{\theta}$ in its vectorized form is:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \mathbf{X}^\top (\sigma(\mathbf{X}\boldsymbol{\theta}) - \mathbf{y}),$$

where $\sigma(\mathbf{X}\boldsymbol{\theta})$ denotes the element-wise application of the sigmoid function across the predictions for each sample. By iteratively updating $\boldsymbol{\theta}$ in the direction of the negative gradient using methods such as stochastic gradient descent (SGD) or Adam, the model converges to parameter values that improve prediction accuracy over the training data.

3.2. Basis Expansion and Universal Function Approximation

While the linear form of logistic regression is effective for linearly separable data, it may struggle with more complex

decision boundaries. Basis expansion is a powerful extension that enhances the model's expressive capacity by applying a nonlinear transformation to the input features. This is done by mapping each input $\mathbf{x}$ to a higher-dimensional feature space $\phi(\mathbf{x})$, where $\phi(\cdot)$ represents a set of basis functions such as polynomial, radial basis, or kernel functions. Logistic regression with basis expansion thus models the probability of the positive class as:

$$P(y = 1|\mathbf{x}; \boldsymbol{\theta}) = \sigma(\phi(\mathbf{x})^\top \boldsymbol{\theta}).$$

By choosing a sufficiently rich set of basis functions, the model can approximate a wide range of complex decision boundaries. In fact, basis expansion enables logistic regression to achieve universal function approximation, meaning that with an adequate choice of basis functions, the model can approximate any continuous function on a compact domain arbitrarily well. This property broadens the applicability of logistic regression to a diverse set of classification problems, making it a versatile tool in both theoretical and practical machine learning applications. Moreover, basis function expansion has been shown theoretically to allow logistic regression models to fit arbitrarily complex functions (De Branges, 1959). Such powerful theoretical statements are sometimes used as the basis for making claims that machine learning models have limitless potential. In our experiments we will test this sort of reasoning.

3.3. Generative Pre-Trained Transformer Basis Expansion

Recent advances in large language models (LLMs) have demonstrated their ability to capture rich, high-dimensional representations of input data through natural language prompts. By leveraging these models, such as GPT, we can enhance the feature space of binary logistic regression with sophisticated, context-sensitive embeddings. This approach effectively transforms the raw input $\mathbf{x} \in \mathbb{R}^n$ into a higher-dimensional space $\phi(\mathbf{x}) \in \mathbb{R}^m$, where $m \gg n$, using the LLM as a basis expansion function.

To accomplish this transformation, each input $\mathbf{x}$ is first converted into a prompt that encapsulates relevant information about the data. For instance, in a classification task distinguishing between sentiment types, a prompt might rephrase the features into a sentence describing the tone or sentiment explicitly. This prompt is then fed to GPT, producing an embedding $\phi(\mathbf{x})$ that captures nuanced semantic and contextual features not accessible through traditional basis functions.

By leveraging GPT embeddings, logistic regression can operate in a feature space enriched with information drawn from vast language representations. To optimize this setup, we fine-tune a lightweight model head atop the GPT-generated embeddings, where the model head is a logistic regression layer. Fine-tuning this head on task-specific data allows for efficient adaptation to the unique characteristics of the dataset, while the pretrained GPT model remains fixed, providing a stable, high-quality feature representation. This separation enables the logistic regression model head to effectively learn from the expanded feature space, resulting in improved classification performance on tasks requiring nuanced understanding and contextual sensitivity.

4. Experiments

In this section, we experimentally evaluate the performance of logistic regression when varying the number of basis features, and we evaluate the GPT embedding using the publicly available GPT-2 large language model. All of the code for our experiments can be found on github.com.

4.1. Dataset Descriptions

As previously noted each version of Kryptonite-2.0 has an n dimensional feature space and represents a binary classification problem. One major difference in the datasets is that as we increase the dimensionality of each dataset we also increase the number of samples that comprise the dataset. This is because keeping the number of examples for the learning task fixed would make the problem impossible as n goes to infinity. Thus, having the number of samples increase with n keeps the dataset challenging, but not impossible.

4.2. Target Performance for Successful Models

For our experiments, we defined specific accuracy thresholds for each dimension to reflect the increasing challenge of classification in higher-dimensional spaces. For lower dimensions, where separability is typically easier, we set higher accuracy targets: 94% for $n = 10$ and 93% for $n = 12$. As dimensionality increased, we adjusted expectations accordingly, setting thresholds of 92% for $n = 14$, 91% for $n = 16$, 80% for $n = 18$, 75% for $n = 20$. These thresholds allow us to gauge performance realistically given the challenges associated with each dimensional setting. This is visualized in Figure 2.

4.3. Performance of Basis Expanded Logistic Regression

In order to test logistic regression models with polynomial basis expansion, we first split our data into 60% training data, 20% validation data, and 20% testing data. We then train our model using the stochastic average gradient (SAG) approach which is equivalent to what is described in Equation (1). We use basic weight-decay regularization with a constant of 1/0.85 and use sci-kit-learn implementations to ensure that our figures and analysis are not the result of

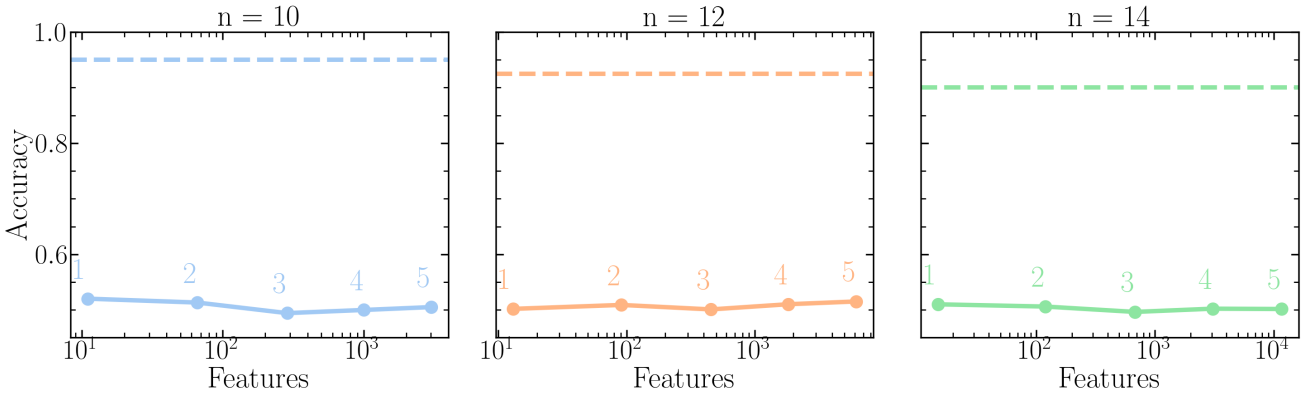

Figure 1. Classification accuracies on the Kryptonite-n dataset using polynomially-expanded logistic regression across various degrees of polynomial expansion. Notably, no configuration of n or polynomial degree achieves an accuracy above 55%, illustrating the model's limitations in capturing the dataset's complex structures despite significant increases in feature dimensionality. This suggests intrinsic challenges posed by the dataset.

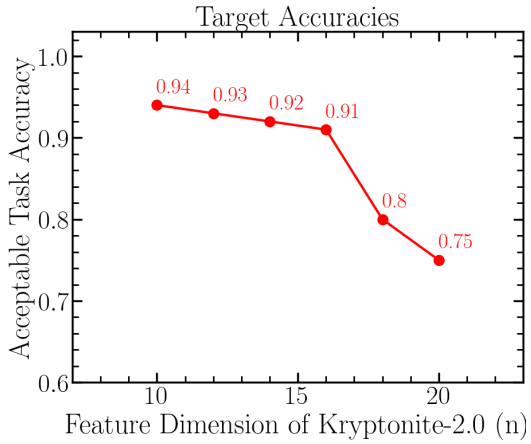

Figure 2. Acceptable accuracy targets for each value of n in our datasets. Higher dimensional problems are substantially harder to fit and thus have a lower threshold.

faulty implementation.

In Figure 1, we visualize the performance of logistic regression with polynomial basis expansion for various values of n. For $n = 10$, $n = 12$, and $n = 14$, the polynomial regression model does not come even close to the stated acceptable performance ranges which are discussed in the previous section and plotted on the y-axis as dotted lines.

4.4. Performance of GPT-Finetuning

We use the open-source weights and model for GPT-2 using the huggingface library. We encode each input using the following prompt: "*Please encode the following vector for a binary classification task:*" followed by the feature values for each vector. This kind of basis expansion is equivalent to embedding each of the feature vectors in a 786-dimensional space. Unfortunately, training a logistic regression head yields a classifier that has worse than random-guessing accuracy (0.496) which is not only considerably beneath the target accuracy, but also seems to indicate that the GPT model we used destroys any even slightly useful information that existed in the original feature space.

5. Future Works

Different Machine Learning Models In this work, we primarily examined the performance of logistic regression for our binary classification task. While logistic regression is well-suited for linearly separable data, it may be suboptimal for complex patterns. Future studies could explore alternative models, such as neural networks, which are known for their capacity to capture intricate, non-linear relationships through deep representations. Additionally, tree-based models or clustering-based approaches could provide a richer structure for capturing decision boundaries, especially in scenarios where the dataset presents high variance or multiple classes. Comparative analysis across these models could yield valuable insights into the trade-offs in accuracy, interpretability, and computational efficiency.

Different Learning Algorithms Our implementation employed stochastic gradient descent (SGD) due to its simplicity and computational efficiency. However, the use of more sophisticated optimization techniques, such as adaptive gradient methods (e.g., Adam, RMSprop) or even second-order optimization methods, could improve convergence rates and potentially achieve higher accuracy. Future work could involve systematically evaluating the performance of these algorithms across different training

configurations and regularization schemes.

Convergence Analysis The convergence properties of gradient descent in logistic regression are fundamental for efficient training. In this study, we selected hyperparameters based on empirical tuning, yet the theoretical implications of these choices were not fully explored. Future work could undertake a rigorous convergence analysis, investigating the sensitivity of convergence rates to learning rates, batch sizes, and initialization schemes. Additionally, deriving theoretical bounds for our chosen parameters could provide insights into whether the observed behavior aligns with optimal convergence patterns, thereby facilitating the selection of hyperparameters.

Function Approximation Our findings challenge the universal function approximation theory, suggesting that the basis expansions chosen in our models may not fully capture the target function. This raises questions about the adequacy of the basis functions in our analysis. A detailed investigation into the theoretical properties of different basis expansions could provide clarity on whether the function space we explored is sufficiently expressive to approximate complex functions. Future studies might also consider kernel-based approaches or higher-dimensional basis expansions to ensure adequate function coverage.

Predictive Uncertainty Accounting for noise is critical in developing reliable machine learning models, as uncertainty quantification directly impacts the robustness of predictions. Incorporating probabilistic models that explicitly model noise could yield improved predictive performance, particularly in noisy environments. Evaluating these models with predictive uncertainty metrics, such as confidence intervals or entropy measures, would provide a benchmark for assessing model reliability. This direction could enhance the interpretability and robustness of our predictions, offering a clear understanding of model confidence in real-world applications.

References

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Bubeck, S., Chandak, V., Eldan, R., Gehrke, J., Hoi, S., Hou, L., Krawczuk, P., Laskin, M., Li, Y. T., Terzi, E., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Chollet, F. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

De Branges, L. The stone-weierstrass theorem. *Proceedings of the American Mathematical Society*, 10(5):822–824, 1959.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., and Corrado, G. A guide to deep learning in healthcare. *Nature medicine*, 25(1): 24–29, 2019.

Heaton, J., Polson, N., and Witte, J. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1):3–12, 2017.

Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

Küttler, H., Germain, T., Teh, Y. W., Rocktäschel, T., and Grefenstette, E. The nethack learning environment. In *NeurIPS*, 2020.

Moskvichev, A., Odouard, V. V., and Mitchell, M. The conceptarc benchmark: Evaluating understanding and generalization in the arc domain. *arXiv preprint arXiv:2305.07141*, 2023.

Sawada, T., Paleka, D., Havrilla, A., Tadepalli, P., Vidas, P., Kranias, A., Nay, J. J., Gupta, K., and Komatsuzaki, A. Arb: Advanced reasoning benchmark for large language models. *arXiv preprint arXiv:2307.13692*, 2023.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.