Multi-armed Bandit Solutions to Neural Network Learning

Caputo, Tristan; Department of Mathematics, University of Arizona

Lenharth, Paul; Department of Chemistry and Biochemistry, University of Arizona

Ludington, River; Department of Mathematics, University of Arizona

Mentor: Joseph Gibney

Abstract

Various Multi-armed Bandit solutions will be investigated through use of probabilistic modeling. Solutions will be evaluated by comparison to the expectation of playing only the best arm. Multiple arm numbers and different payout distributions will be tested.

Introduction

Neural Networks are an increasingly important tool to modern society, used for everything from ad-targeting to self-driving cars to stock trading (3). One of the main aspects of a neural network design is how it learns, whether through external tests or through characteristics of the network itself. For supervised learning, the application of external tests, one key design consideration is determining how to backpropagate the network weights to minimize learning time while achieving high levels of performance. This creates a trade-off between exploration, determining the overall weight structure of the network, and exploitation, making smaller adjustments to fine-tune the current weights. While exploration is critical to achieving the best results, it is time-intensive. Thus there is a desire to minimize exploration in favor of fine-tuning weights, while still having a strong performance, even if it is not necessarily the best the network could have achieved.

The Multi-Armed Bandit is a formulation used to model the trade-off between exploitation and exploration. The idea is the following: a player faces a number of slot machines, each with a different payout distribution, with the intention of losing the least money and gaining as much as possible. There are multiple algorithms that we can use to optimize this problem. The most common algorithms in use today are the Epsilon-Greedy and its variations, such as Softmax. In the Epsilon-Greedy algorithm, the best experimental arm is played with probability 1-ε . The other arms are played with probability ε /(n-1) where n is the total number of arms (4). Thus if the experimental best arm is not played, the other arms have equal probability to be played that turn. Interestingly, “the most naive approach, the ε-greedy strategy, proves to be often hard to beat” (Vermorel, 2005). This finding is contrary to the adoption of more theoretically robust algorithms such as EXP3 or Interval Estimation. Furthermore, “although many algorithms for the multi-armed bandit problem are well-understood theoretically, empirical confirmation of their effectiveness is generally scarce” (Kuleshov, 2014). This reinforces the need for sound empirical evidence. The usage of Epsilon-Greedy does not imply the usage of a specific strategy however, as many variations of the Epsilon-Greedy exist, such as Softmax. The Softmax variation weights the nonoptimal arms using a softmax function, such that the second best arm experimentally is played much more often. Another variation involves decreasing ε over the course of the test, such that less exploration is done towards the end. Many variations are not mutually-exclusive, and so can be used congruently. This makes many specific algorithms possible for just one

archetype, with many archetypes existing.

Implementation/Testing

Our goal is to find out which algorithm works best with initial parameters (number of arms and reward variance) for a specific environment. We will write a code in Python (or C) to simulate an agent playing on K number of arms and each arm will have its own respective distribution (based on a set reward variance). Various chosen algorithms such as Softmax and Epsilon greedy will be implemented through a method function which will output which arm should be chosen for the next turn. After 1000 turns, we will measure the total expected regret.

The formula for total expected regret is explicitly defined by Kuleshov as:

$$R_T = T\mu^* - \sum_{t=1}^{T} \mu_{j(t)}$$

where R_T is the total expected regret at turn T, $(\mu^*) = max_{i=1,\dots k}\mu_i$ (which is the expected reward from the best arm), and $\mu_{j(t)}$ is the expected value of the slot machine arm with index j(t).

For each test, we will choose different values for the total number of arms and reward parameters and record each algorithm's response. After obtaining data from multiple tests, each algorithm will be refined through slight modification to minimize its total regret over all the tests. An example of such modification would be optimizing the value of epsilon for the Epsilon-greedy algorithm.

Applications

The Multi-Armed Bandit problem applies to a long list of applications ranging from clinical trials to recommendation algorithms (1)(5). We will focus on the latter as recommendation algorithms are used in almost every social media, entertainment and sometimes e-commerce websites. Clinical trials have also been studied extensively in relation to this exploitation/exploration trade-off problem. We hope to gain a greater understanding as to what influences these recommendation algorithms in the background.

Conclusion

The Multi-Armed Bandit problem is highly relevant to the training of computer neural networks, and as such, progress on it's solutions has directly relevant implications. With statistical measures such as total expected regret per turn, we have useful tools for measuring the usefulness of algorithms. To further advance algorithms, we intend to first understand why the basic epsilon-greedy and softmax algorithms outperform more advanced algorithms. From that basis we can formulate algorithms to attempt to reduce regret, and provide real testing rather than simply theoretical background. This serves the purpose of guiding algorithm development that is practical, useful, and recreatable.

References

1. Vermorel, Joannes, and Mehryar Mohri. "Multi-armed bandit algorithms and empirical evaluation." *European conference on machine learning*. Springer, Berlin, Heidelberg, 2005.
2. Kuleshov, Volodymyr, and Doina Precup. "Algorithms for multi-armed bandit problems." *arXiv preprint arXiv:1402.6028* (2014).
3. Raja, Sudeep. "Multi Armed Bandits and Exploration Strategies." *Multi Armed Bandits and Exploration Strategies – Sudeep Raja – MS/Phd Student at UMass Amherst*, 28 Aug. 2016, sudeepraja.github.io/Bandits/.
4. "Multi-Armed Bandits." *The Data Incubator MultiArmed Bandits Comments*, blog.thedataincubator.com/2016/07/multi-armed-bandits-2/.
5. Li, Lihong, et al. "A contextual-bandit approach to personalized news article recommendation." *Proceedings of the 19th international conference on World wide web*. ACM, 2010.