Exploring the performance of DNN and RL models in 4-armed bandit problem: a comparative analysis

Dian Jin dj928

Xinxiaomeng Liu xl4701

Introduction

Inspired by our homework in Reinforcement Learning, in which we explored some simple reinforcement learning algorithms for the n-armed bandit problem, we conducted further research on the study of this human decision-making problem using two different approaches: a deep neural network model and a reward-oriented RL model.

With the tremendous success of deep neural networks in a wide range of fields including computer vision, natural language processing and robotics (Schmidhuber, 2015)(Collobert, 2011), they have become a focal point of attention. Due to their high capacity and data-driven design, deep neural networks hold the potential to offer novel insights in the study of cognitive processes (Ma & Peter, 2020), including human decision making. The n-armed bandit problem is a classic human decision-making problem in which an agent learns to select the most rewarding option from a set of n choices over a series of trials. Deep neural networks have the ability to learn complex representations of the input data, which can be super useful in the n-armed bandit problem. However, because of the opaque nature of deep neural networks, their ability to explain how an operation is carried out is limited (Rudin, 2019). We also created a reward-oriented reinforcement learning model. In the n-armed bandit problem, the goal is to maximize the total reward obtained over a number of trials. As such, a reward-oriented model is a natural fit for this problem. By optimizing for rewards, such a model can quickly learn which actions lead to the highest rewards and choose those actions more frequently.

In our study, we first trained an LSTM model. Then we created an RL agent which updates the expected reward of each bandit and policy in each trial and did hyperparameter tuning using human subjects' data. We did a comparative analysis about the prediction accuracy of these two models and the similarity between these two models for different payoff structures.

Literature Review

Deep Neural Networks and Reinforcement Learning models have emerged as powerful tools for studying human decision-making problems. One of the earliest studies in this area was conducted by (Daw et al., 2011), who built a RL model to explain the behavior of human subjects in a 2-armed bandit task. They found that the RL model provided a better account of the behavior of human subjects than traditional models based on expected utility theory. Since then, many studies have used RL models to study various kinds of human decision-making problems, such as financial decision making (Huys et al., 2012) and social decision making (Behrens et al., 2009).

Deep neural networks have also shown great promise in modeling human decision making. In the domain of perceptual decision-making, Yamins et al. (Yamins et al., 2014) trained a deep neural network on a large dataset of natural images and found that the deep neural network was able to predict neural responses in the primate visual cortex with high accuracy. In the domain of risky decision-making, Hu et al. (Hu et al., 2015) used deep neural networks to model neural activity in the human brain during a gambling task. They found that the deep neural network models were able to predict individual trial outcomes with high accuracy and that the representations learned by the models were consistent with those observed in the human brain. Deep neural networks have also been applied in the social domain. Zaki et al. (Zaki et al., 2017) used a deep neural network to model the neural mechanisms underlying empathy, and found that deep neural networks were able to predict individual differences in empathic accuracy. Overall, deep neural networks can be used as a great tool for modeling and understanding human decision-making. These studies have also provided insights into the underlying neural mechanisms and have demonstrated the utility of deep neural networks in predicting behavior.

Methods & Models

Objective

We experimented with an online dataset[1] of human subjects who participated in a four-armed bandit task. 965 participants were recruited and asked to complete 150 rounds where they chose among four options and received the reward associated with that option. The participants faced three payoff structures that were set before the experiment (Fig 1). Each option was initialized with a reward value between 0 and 98, which shifted over time. The mean reward and normalized standard deviations for the options respectively were (43.20, 0.39), (56.42, 0.28), (49.05, 0.41), and (31.04, 0.39). We notice that the average reward of option 2 was slightly higher than the rest. Each round, the participants had four seconds to provide a response, beyond which a null value would be recorded and they

[1] Bahrami, Bahador, and Joaquin Navajas. "4 Arm Bandit Task Dataset." OSF, 8 Feb. 2022. Web.

would move directly to the next trial with no reward received at the current. Here are some behavioral observations of the data. We denote the action at time *t* as a_t, and the reward at time *t* as r_t. Out of the 965 people, only 127 finished the 150 trials and an average subject completed 145 trials. Across all participants in all rounds, the average choice (1, 2, 3, 4) was 2.37 and the normalized standard deviation of choices was 0.36. The average reward received in each round was 58.53. We can make a rough observation that the participants were sensitive to the rewards and preferred options that they perceived to result in higher rewards, while their choices were also influenced by the moderately high variation in the reward outcomes and rounds.

Data Preprocessing

For the neural networks model, we performed the same procedure for each payoff structure. We first dropped records where the subjects failed to provide a response. 80% of all users were randomly drawn to form the training set. The rest of 20% of participants were assigned to the testing set. We used a sliding window function to transform user responses into sequences of length 5, where each sequence contains 5 steps [(a_{t-4}, r_{t-4}), (a_{t-3}, r_{t-3}), (a_{t-2}, r_{t-2}), (a_{t-1}, r_{t-1}), (a_t, r_t)]. After an evaluation of the tradeoff between model accuracy and computation efficiency, we decided that a 4-step sequence would be promising for this particular task. For each payoff structure, most participants missed one step and others missed more. We removed these missed trials, causing gaps in the data, e.g. steps at times $t-4$, $t-3$, $t-2$, $t-1$ might be used to predict the choice at time $t+1$ if the user did not respond at time $t+0$. Since the shifts in option rewards associated with these gaps were minimal, we did not exclude them from our experiment.

For the reward-oriented model, inspired by our homework2 - Reinforcement Learning, we used a reinforcement learning model which updates the expected reward of each bandit in each trial. We first calculated the mean of the rewards of all four bandits in each payoff structure. These values were then used as the initial expected reward of every bandit. For each payoff structure, the model was optimized for all participants' entire sequence of actions and rewards. In terms of handling missing values, since a null value in a participant's choice does not affect the operation of the model, we only need to be careful to skip these null values when calculating the accuracy rate.

Neural Network model

We chose long short term memory(LSTM) to construct the neural network model for this task. LSTM is a type of recurrent neural network that is designed so that it can selectively store, modify and retrieve information over some period of time. In our task to predict a series of human choices, LSTM is able to model the temporal dynamic behavior through incorporating feedback connections in its architecture. More specifically, the key advantage of LSTM in this context is the ability to capture long-term dependencies in the subject's decision making process. If the subject has a preference for certain arms over others, this preference may be influenced by their past experiences and feedback from the environment, and LSTM can capture these dependencies with its gate mechanism to selectively store or discard information over time.

We noted that it was possible to feed in the entire history of each user into the LSTM model but chose to train with a fixed length of sequences instead. The reason for this was two-fold. The first one is the restless nature of the underlying reward structure. As the reward of the four options volatilizes, choice-reward pairs from earlier in the timeline provide weaker values for training our model than more recent data that more strongly influences the human participant's next move. The second reason is that if we predict a new step with the entire history before it, it would cause different lengths of input data for each point in user history. Earlier predictions will be based on shorter sequences while later predictions may be based on more than one hundred steps, which leads to significant bias when evaluating model accuracy across participants and steps.

The emphasis on more recent experiences is analogous to how humans might make decisions, where newer experiences carry more weight than older ones, and using inconsistent or biased information can lead to suboptimal choices. For these reasons, we decided to train our neural network model and prediction a participant's choice at time t with their S previous steps and outcomes [(a_{t-S}, r_{t-S}), …., (a_{t-1}, r_{t-1})]. We set S to 4 after evaluating the tradeoff between prediction accuracy and computation efficiency when adjusting S.

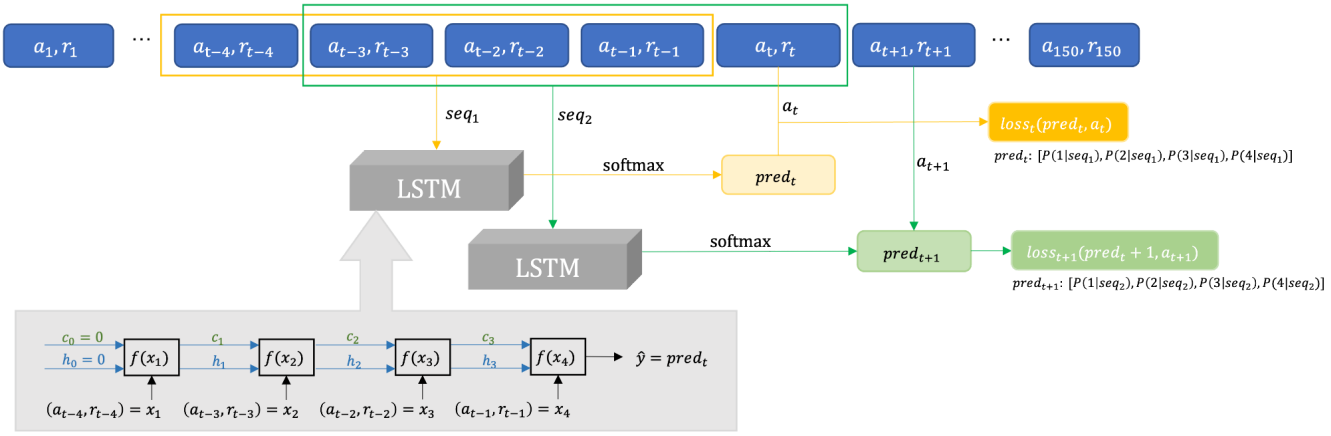

Figure 1 Neural network model architecture

Given the simplicity underlying this task, we used a single layer LSTM (Fig 1). As shown in the supplemental materials, each LSTM cell has 64 units, trained for 300 epochs and with 2048 batch size. We chose Adam optimizer with a learning rate of 0.001, beta_1=0.9 and beta_2=0.99. This configuration was chosen after performing a grid search of hyper parameter tuning, by fitting the model with different combinations of parameters in a five-fold cross validation. The last unit is a 4-way softmax Dense layer and outputs a list of probabilities of choosing each of the four options.

Since the true labels for predictions are one-hot encoded, the model was trained using categorical cross-entropy loss using training data defined above:

$$loss = -\frac{1}{N}\sum_{i=1}^{N} y_i log\hat{y}_i$$

The output from LSTM will later be compared with the predictions of our reward-oriented model. We aim to identify the similarity and disparity between these two models' performance and analyze if they are more similar or distinct in phases where the four bandits' rewards show different patterns.

Reward-oriented Model

The reward-oriented RL model assumes that decisions are driven by the expected reward of each option and these expected rewards are learned by updating the agent's expectations. We created an agent class which keeps track of the average reward earned from each draw of the bandit. In each trial, the agent makes a decision which of the four bandits to choose, noted as action a_t, and receives a reward r_t. In each round, the agent uses the "constant step size" update rule to update the value of each action.

$$V(a_t) = V(a_t) + \alpha(r_t - V(a_t))$$

where α is a constant step size parameter, $0 \leq \alpha \leq 1$.

This agent also includes a parameter ε which will determine the probability of choosing a random action, otherwise it makes its choice according to a softmax distribution based on the q-values:

$$p_t(a) = \frac{e^{\beta V_t(a)}}{\sum_i e^{\beta V_t(a_i)}}$$

where β is a positive scalar that controls the "temperature" of the distribution. A higher beta results in a "softer" distribution, where probabilities are more evenly distributed across the possible outcomes.

For each payoff structure, the mean rewards of all four bandits were initiated with the mean value of the entire sequence of rewards of all four bandits. Then for each payoff structure, we optimized the model for all participants' entire sequence of actions and rewards by maximizing the average accuracy of predictions using a Grid Search method. This involved searching over a predefined set of values for three parameters (ε, α, β) to find the combination of parameter values that maximize the log likelihood of the observed data. The range of values to search for (ε, α, β) is as follows:

params_grid = { 'epsilon': [0, 0.1, 0.2, 0.3, 0.4, 0.5],
'alpha': [0.1, 0.2, 0.3, 0.4, 0.5],
'beta': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]}

The best combinations are $(\varepsilon, \alpha, \beta) = (0, 0.1, 5)$ for payoff structure 2, $(\varepsilon, \alpha, \beta) = (0, 0.2, 4)$ for payoff structure 3 and $(\varepsilon, \alpha, \beta) = (0, 0.1, 3)$ for payoff structure 4. It turns out the ε is not a necessary parameter.

Model Accuracy Measure

For each payoff structure, we measured the proportion of correct predictions made by the Neural Network model and the Reward-Oriented model.

$$Accuracy_t = \frac{(number\ of\ correct\ predictions)_t}{(total\ number\ of\ predictions)_t}$$

Model Similarity Measure

For each payoff structure, we measured the similarity between the Neural Network model and the Reward-Oriented model.

$$Similarity_t = \frac{\sum_{i=1}^{N} \delta(a_t^{1,i}, a_t^{2,i})}{N}$$

$$\delta(a,b) = \begin{cases} 1 \; if \;\; a = b \\ 0 \; otherwise \end{cases}$$

Where $a_t^{k,i}$ is the action made by model k ($k = 1, 2$) at time t for participant i. $\delta(a, b)$ measures whether the Neural Network model and the Reward-Oriented model made the same action at time t for participant i or not.

Discussion & Conclusion

In our study, we first trained an LSTM model to mimic and predict human decision-making behavior for a classic computational cognitive modeling experiment. To understand its predictions, we also created an RL agent which updates the expected reward of each bandit and policy in each trial and uses human participants' data to choose the best hyperparameters. Using the accuracy and similarity measures that we defined, we made figure 2 to better understand the relationship between the LSTM model and reward-oriented model. We could make such observations as that the neural network model with its large set of available parameters performs generally better than the explicit reward-oriented reinforcement learning model while their performances varied over time.

Both models' accuracy scores grew higher when one option was distinctly better than rest. Naturally, since they both made better predictions, their results were more similar. The gap between their performance was the most obvious when the underlying reward patterns were ambiguous about which option tended to be significantly better than others in the next step (Fig 2). Dips appear in the trend of similarity scores between the two models when the underlying reward structure for the four bandits begin to oscillate horizontally. The training sequences from these periods are less indicative of potential human choices, whereas periods where one or more bandits are obviously trending up or down exhibit stronger signals for the models to base their predictions on. During periods of uncertainty, the neural network model's accuracy scores tended to trend higher above that of the reward-oriented model than periods when reward patterns were more pronounced. We can observe such phenomena in steps 40-60 for structure 2, steps 50-70 for structure 3, and steps 60-80 for structure 4.

The input to the LSTM model contains sequences of data where patterns were not obvious, as well as their associated 4 steps of human choice-reward pairs and the participant choice for step 5 and the subsequent reward. This could be the reason why the LSTM model performs better in these situations since they have "seen" similar data patterns in the training set. Another possible explanation is that the reward-oriented model is optimized to maximize the cumulative reward over time, which means it tends to favor the options that have a higher expected reward. When the underlying reward patterns are ambiguous, it becomes difficult for the reward-oriented RL model to distinguish between them and it may end up selecting a suboptimal option that has a slightly higher expected reward in the short term.

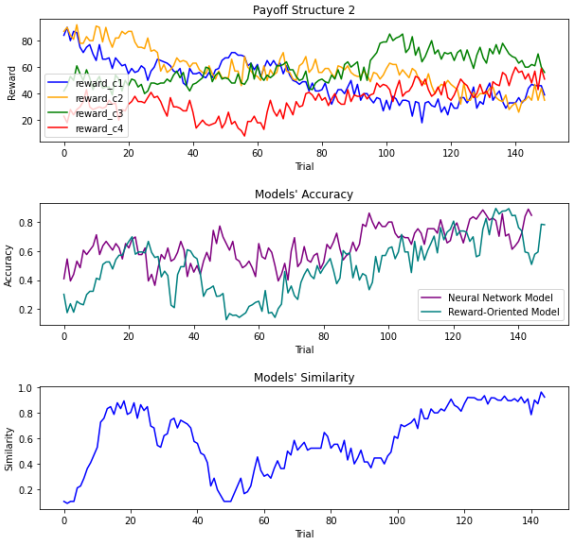

Figure 2 NN vs. reward-oriented (structure 2)

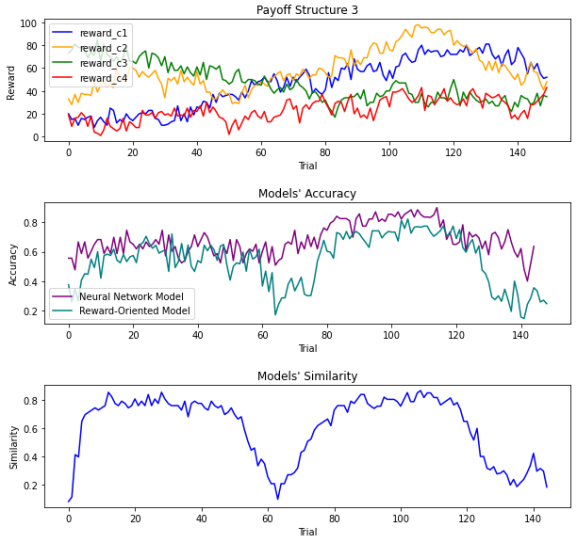

Figure 2 NN vs. reward-oriented agent (structure 3)

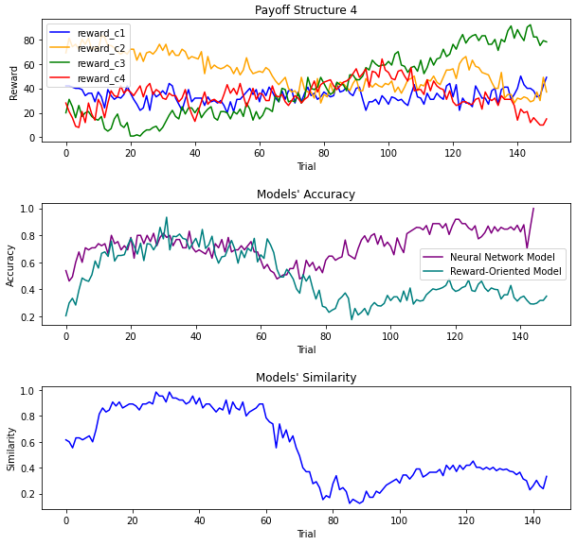

Figure 2 NN vs. reward-oriented agent (structure 4)

The difference and the temporal changes between their accuracy scores suggest that most participants' actions were better captured by the neural network model, which means their actions might be less affected by the outcomes of their precious choices but were still represented rather accurately by the LSTM model with its large set of parameters. A small number of participants during some of the time steps showed the reverse as there were moments where the RL model trended above the NN.

There are many advantages of using neural networks for scientific research in the related fields. In our case with the LSTM model, we focus on its ability to handle sequences in inputs and make decisions through tuning its parameters about the amount of memory to carry over long periods of time. Besides capturing temporal dependencies, LSTM handles complex input sequences with potential inter-item interactions, typical for visual, auditory and cognitive data, making it an effective tool for such studies. Though operations are done in a black box, there is still some interpretability to its structure. LSTM provides insights to human's decision making by highlighting which input and features are more important for predictions, which can help researchers better understand the underlying mechanisms of human decision making.

After fitting the data to a specific context, more explanation was called upon, particularly due to the fact that computation and optimization are done unseen in the model's hidden cells, so that we can promote more motivation and trustworthiness to transfer models developed in one field/topic to others. There are different ways of characterizing model behaviors. For example, adversarial examples can be generated deliberately in a way that neural networks would fail on these examples so that researchers could study the vulnerabilities in their processes (Dezfouli et al. 2020). In our study, we chose to model and compare a neural network with results from an explicit reward-oriented agent to characterize its mechanism.

The limitations of neural networks were also considered. One of the most prominent was the problem of overfitting. This is likely to be caused by a redundant amount of parameters. In our experiment, the average accuracy over test participants and their choices was around 2% higher than where the training set's accuracy oscillates, which meant overfitting was non-existent for the three payoff structures. Second of all, the LSTM models were trained over all human participants from the same payoff structure. In this way, it may fail to capture individual differences in behavior and decision-making. The model instead learns an average representation of the population, which may not accurately represent any individual person.

In all, we worked to show how neural network models can be trained to mimic human decision processes in the n-armed bandit task, and how comparing its performance with that of a reward-oriented agent under different payoff structures can shine light on their similarity and disparity from the perspective of their underlying mechanisms. Furthermore, we proposed that utilizing explicit theory-driven models in the experimental setting can help characterize black-box models. Such characterization is beneficial not only for advancing scientific understanding but also for communicating the performance of these models to practitioners and the general public.

Supplemental Material

https://github.com/DianJin11/1016-final-project

References

Behrens, T.E.J., Hunt, L.T., Woolrich, M.W., & Rushworth, M.F.S. (2009). Associative learning of social value. *Nature*, *456*(7219), 245-249.

Collobert, R. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*, 2493-2537.

Daw, N.D., Gershman, S.J., Seymour, B., Dayan, P., & Dolan, R.J. (2011). Model-based influences on humans' choices and striatal prediction errors. *Neuron*, *69*(6), 1204-1215.

Dezfouli A, Nock R, Dayan P. Adversarial vulnerabilities of human decision-making. Proc Natl Acad Sci U S A. 2020 Nov 17;117(46):29221-29228. doi: 10.1073/pnas.2016921117. Epub 2020 Nov 4. PMID: 33148802; PMCID: PMC7682379.

Hu, X., Hong, S., Ge, S., & Zhang, H. (2015). Modeling neural activity using fMRI-inspired convolutional neural network. *2015 IEEE International Conference on Bioinformatics and Biomedicine*, 830-835.

Huys, Q.J., Eshel, N.E., O'Nions, E.J., Sheridan, L., Dayan, P., & Roiser, J.P. (2012). Bonsai trees in your head: How the pavlovian system sculpts goal-directed choices by pruning decision trees. *PLoS Computational Biology*, *8*(3), e1002410. https://doi.org/10.1371/journal.pcbi.1002410

Ma, W.J., & Peter, B. (2020). A Neural Network Walks into a Lab: Towards Using Deep Nets as Models for Human Behaviour. 1-39. https://doi.org/10.48550/arXiv.2005.02181

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat Mach Intell*, *1*, 206-215. https://doi.org/10.1038/s42256-019-0048-x

Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Netw*, (61), 85-117.

Yamins, D.L., Hong, H., Cadieu, C.F., Solomon, E.A., Seibert, D., & DiCarlo, J.J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, *111*(23), 8619-8624.

Zaki, J., Weber, J., Bolger, N., & Ochsner, K. (2017). The neural bases of empathic accuracy. *Proceedings of the National Academy of Sciences*, *114*(16), 4004-4009.