CptS 570 Machine Learning, Fall 2018
Homework #4
Due Date: Thu, Dec 6 (9:10am)

NOTE 1: Please use a word processing software (e.g., Microsoft word or Latex) to write your answers and submit a printed copy to me at the beginning of class on Oct 23. The rationale is that it is sometimes hard to read and understand the hand-written answers.

NOTE 2: Please ensure that all the graphs are appropriately labeled (x-axis, y-axis, and each curve). The caption or heading of each graph should be informative and self-contained.

1. **(10 points)** Suppose you are given 7 data points as follows: A = (1, 1); B = (1.5, 2.0); C = (3.0, 4.0); D = (5.0, 7.0); E = (3.5, 5.0); F = (4.5, 5.0); and G = (3.5, 4.5). Manually perform 2 iterations of K-Means clustering algorithm on this data. You need to show all the steps. Use Euclidean distance (L2 distance) as the distance/similarity metric. Assume number of clusters k=2 and the initial two cluster centers C_1 and C_2 are B and C respectively.

2. **(10 points)** Please read the following paper and write a brief summary of the main points. Michael Jordan and Tom Mitchell. Machine learning: Trends, perspectives, and prospects. Science 17 Jul 2015: Vol. 349, Issue 6245, pp. 255-260. `http://science.sciencemag.org/content/349/6245/255`

3. **(20 points)** Please go through the excellent talk given by Kate Crawford at NIPS-2017 Conference on the topic of "Bias in Data Analysis" and write a brief summary of the main points.

   Kate Crawford: The Trouble with Bias. Invited Talk at the NIPS Conference, 2017. Video: `https://www.youtube.com/watch?v=fMym_BKWQzk`

4. **(20 points)** Please read the following paper and write a brief summary of the main points.

   Matthew Zook, Solon Barocas, danah boyd, Kate Crawford, Emily Keller, Seeta Pea Gangadharan, Alyssa Goodman, Rachelle Hollander, Barbara Knig, Jacob Metcalf, Arvind Narayanan, Alondra Nelson, Frank Pasquale: Ten simple rules for responsible big data research. PLoS Computational Biology 13(3) (2017)
   `https://www.microsoft.com/en-us/research/wp-content/uploads/2017/10/journal.pcbi_.1005399.pdf`

5. **(Finite-Horizon MDPs.)** Our basic definition of an MDP in class defined the reward function $R(s)$ to be a function of just the state, which we will call a *state reward function*. It is also common to define a reward function to be a function of the state and action, written as $R(s, a)$, which we will call a *state-action reward function*. The meaning is that the agent gets a reward of $R(s, a)$ when they take action a in state s. While this may seem to be a significant difference, it does not fundamentally extend our modeling power, nor does it fundamentally change the algorithms that we have developed.

   a) **(5 points)** Describe a real world problem where the corresponding MDP is more naturally modeled using a state-action reward function compared to using a state reward function.

   b) **(10 points)** Modify the Finite-horizon value iteration algorithm so that it works for state-action reward functions. Do this by writing out the new update equation that is used in each iteration and explaining the modification from the equation given in class for state rewards.

c) **(10 points)** Any MDP with a state-action reward function can be transformed into an "equivalent" MDP with just a state reward function. Show how any MDP with a state-action reward function $R(s, a)$ can be transformed into a different MDP with state reward function $R(s)$, such that the optimal policies in the new MDP correspond exactly to the optimal policies in the original MDP. That is an optimal policy in the new MDP can be mapped to an optimal policy in the original MDP. *Hint: It will be necessary for the new MDP to introduce new "book keeping" states that are not in the original MDP.*

6. **(k-th Order MDPs.) (15 points)** A standard MDP is described by a set of states S, a set of actions A, a transition function T, and a reward function R. Where $T(s, a, s')$ gives the probability of transitioning to s' after taking action a in state s, and $R(s)$ gives the immediate reward of being in state s.

   A k-order MDP is described in the same way with one exception. The transition function T depends on the current state s and also the previous $k-1$ states. That is, $T(s_{k-1}, \cdots, s_1, s, a, s')$ $= Pr(s'|a, s, s_1, \cdots, s_{k-1})$ gives the probability of transitioning to state s' given that action a was taken in state s and the previous $k-1$ states were $(s_{k-1}, \cdots, s_1)$.

   Given a k-order MDP $M = (S, A, T, R)$ describe how to construct a standard (First-order) MDP $M' = (S', A', T', R')$ that is equivalent to M. Here equivalent means that a solution to M' can be easily converted into a solution to M. Be sure to describe S', A', T', and R'. Give a brief justification for your construction.

7. (**10 points**) Some MDP formulations use a reward function $R(s, a)$ that depends on the action taken in a state or a reward function $R(s, a, s')$ that also depends on the result state s' (we get reward $R(s, a, s')$ when we take action a in state s and then transition to s'). Write the Bellman optimality equation with discount factor β for each of these two formulations.

8. (**10 points**) Consider a trivially simple MDP with two states $S = \{s_0, s_1\}$ and a single action $A = \{a\}$. The reward function is $R(s_0) = 0$ and $R(s_1) = 1$. The transition function is $T(s_0, a, s_1) = 1$ and $T(s_1, a, s_1) = 1$. Note that there is only a single policy π for this MDP that takes action a in both states.

   a) Using a discount factor $\beta = 1$ (i.e. no discounting), write out the linear equations for evaluating the policy and attempt to solve the linear system. What happens and why?

   b) Repeat the previous question using a discount factor of $\beta = 0.9$.

9. (**30 points**) Implementation of Q-Learning algorithm and experimentation.

   You are given a Gridworld environment that is defined as follows:

   State space: GridWorld has 10x10 = 100 distinct states. The start state is the top left cell. The gray cells are walls and cannot be moved to.

   Actions: The agent can choose from up to 4 actions (left, right, up, down) to move around.

   Environment Dynamics: GridWorld is deterministic, leading to the same new state given each state and action

   Rewards: The agent receives +1 reward when it is in the center square (the one that shows R 1.0), and -1 reward in a few states (R -1.0 is shown for these). The state with +1.0 reward is the goal state and resets the agent back to start.

   In other words, this is a deterministic, finite Markov Decision Process (MDP). Assume the discount factor β=0.9.

   Implement the Q-learning algorithm (slide 46) to learn the Q values for each state-action pair. Assume a small fixed learning rate α=0.01.

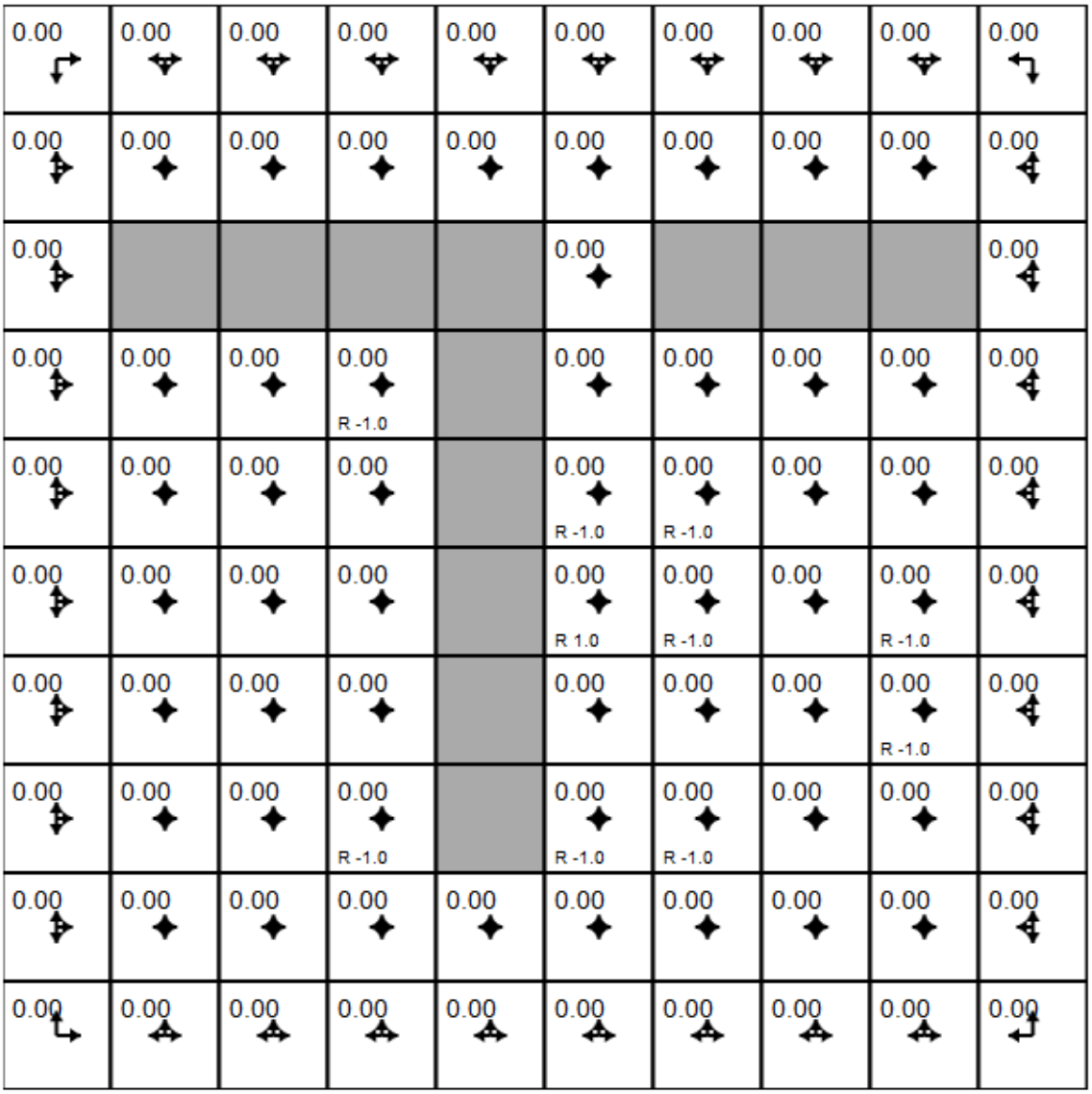


Figure 1: Grid world domain with states and rewards.

Experiment with different explore/exploit policies:

1) ϵ-greedy. Try ϵ values 0.1, 0.2, and 0.3.

2) Boltzman exploration. Start with a large temperature value T and follow a fixed scheduling rate. Give these details in your report.

How many iterations did it take to reach convergence with different exploration policies?

Please show the converged Q values for each state-action pair.

10. (Bonus Question for Extra Credit **25 points**) Implement a simple Convolutional Neural Network (CNN) based classifier of your choice for classifying images with digits.

    You can use any deep learning package of your choice. Keras API (`https://keras.io/`) is very easy to use.

    Please use MNIST database of handwritten digits
    (`https://keras.io/datasets/#mnist-database-of-handwritten-digits`).
    Dataset of 60,000 28x28 grayscale images of the 10 digits, along with a test set of 10,000 images (`http://yann.lecun.com/exdb/mnist/`).

    Please show the learning curve (error as a function of training epochs).

Instructions for Code Submission and Output Format.

Please follow the below instructions. It will help us in grading your programming part of the homework. We will provide a dropbox folder link for code submission.

- Supported programming languages: Python, Java, C++
- Store all the relevant files in a folder and submit the corresponding zipfile named after your student-id, e.g., 114513209.zip
- This folder should have a script file named

  `run_code.sh`

  Executing this script should do all the necessary steps required for executing the code including compiling, linking, and execution
- Assume relative file paths in your code. Some examples:

  `‘‘./filename.txt’’ or ‘‘../hw2/filename.txt’’`
- The output of your program should be dumped in a file named “output.txt”
- Make sure the output.txt file is dumped when you execute the script

  `run_code.sh`
- Zip the entire folder and submit it as

  `<student_id>.zip`

Grading Rubric

Each question in the students work will be assigned a letter grade of either A,B,C,D, or F by the Instructor and TAs. This five-point (discrete) scale is described as follows:

- **A) Exemplary (=100%)**.
  Solution presented solves the problem stated correctly and meets all requirements of the problem.
  Solution is clearly presented.
  Assumptions made are reasonable and are explicitly stated in the solution.
  Solution represents an elegant and effective way to solve the problem and is not overly complicated than is necessary.

- **B) Capable (=75%)**.
  Solution is mostly correct, satisfying most of the above criteria under the exemplary category, but contains some minor pitfalls, errors/flaws or limitations.

- **C) Needs Improvement (=50%)**.
  Solution demonstrates a viable approach toward solving the problem but contains some major pitfalls, errors/flaws or limitations.

- **D) Unsatisfactory (=25%)**
  Critical elements of the solution are missing or significantly flawed.
  Solution does not demonstrate sufficient understanding of the problem and/or any reasonable directions to solve the problem.

- **F) Not attempted (=0%)**
  No solution provided.

The points on a given homework question will be equal to the percentage assigned (given by the letter grades shown above) multiplied by the maximum number of possible points worth for that question. For example, if a question is worth 6 points and the answer is awarded a B grade, then that implies 4.5 points out of 6.

Assignment 4

Sheryl Mathew (11627236)

December 6, 2018

1. K-means Clustering

Input:

Data Points	x	y
A	1	1
B	1.5	2
C	3	4
D	5	7
E	3.5	5
F	4.5	5
G	3.5	4.5

Initial Cluster Centers:

Cluster 1: B

Cluster 2: C

Iteration 1:

Clusters	x	y
B	1.5	2
C	3	4

Data Points	Distance between B and other points	Distance between C and other points
A	$\sqrt{(1.5-1)^2+(2-1)^2}=1.12$	$\sqrt{(3-1)^2+(4-1)^2}=3.61$
D	$\sqrt{(1.5-5)^2+(2-7)^2}=6.10$	$\sqrt{(3-5)^2+(4-7)^2}=3.61$
E	$\sqrt{(1.5-3.5)^2+(2-5)^2}=3.61$	$\sqrt{(3-3.5)^2+(4-5)^2}=1.12$
F	$\sqrt{(1.5-4.5)^2+(2-5)^2}=4.24$	$\sqrt{(3-4.5)^2+(4-5)^2}=1.80$
G	$\sqrt{(1.5-3.5)^2+(2-4.5)^2}=3.20$	$\sqrt{(3-3.5)^2+(4-4.5)^2}=0.71$

Cluster 1: A,B

Cluster 2: C,D,E,F,G

Clusters	Mean value along x	Mean value along y
A,B	$\frac{1+1.5}{2} = 1.25$	$\frac{1+2}{2} = 1.5$
C,D,E,F,G	$\frac{3+5+3.5+4.5+3.5}{5} = 3.9$	$\frac{4+7+5+5+4.5}{5} = 5.1$

Iteration 2:

Clusters	x	y
C-1	1.25	1.5
C-2	3.9	5.1

Data Points	x	y
A	1	1
B	1.5	2
C	3	4
D	5	7
E	3.5	5
F	4.5	5
G	3.5	4.5

Data Points	Distance between C-1 and other points	Distance between C-2 and other points
A	$\sqrt{(1.25-1)^2+(1.5-1)^2} = 0.56$	$\sqrt{(3.9-1)^2+(5.1-1)^2} = 5.02$
B	$\sqrt{(1.25-1.5)^2+(1.5-2)^2} = 0.56$	$\sqrt{(3.9-1.5)^2+(5.1-2)^2} = 3.92$
C	$\sqrt{(1.25-3)^2+(1.5-4)^2} = 3.05$	$\sqrt{(3.9-3)^2+(5.1-4)^2} = 1.42$
D	$\sqrt{(1.25-5)^2+(1.5-7)^2} = 6.66$	$\sqrt{(3.9-5)^2+(5.1-7)^2} = 2.2$
E	$\sqrt{(1.25-3.5)^2+(1.5-5)^2} = 4.16$	$\sqrt{(3.9-3.5)^2+(5.1-5)^2} = 0.41$
F	$\sqrt{(1.25-4.5)^2+(1.5-5)^2} = 4.78$	$\sqrt{(3.9-4.5)^2+(5.1-5)^2} = 0.61$
G	$\sqrt{(1.25-3.5)^2+(1.5-4.5)^2} = 3.75$	$\sqrt{(3.9-3.5)^2+(5.1-4.5)^2} = 0.72$

Cluster 1: A,B,

Cluster 2: C,D,E,F,G

Clusters	Mean value along x	Mean value along y
A,B	$\frac{1+1.5}{2} = 1.25$	$\frac{1+2}{2} = 1.5$
C,D,E,F,G	$\frac{3+5+3.5+4.5+3.5}{5} = 3.9$	$\frac{4+7+5+5+4.5}{5} = 5.1$

2. Machine Learning: Trends, Perspectives and Prospects

Machine learning is used to construct computer systems that improve automatically through experience. It has applications in the field of biology to cosmology to social science. Machine Learning algorithms search through a large space of candidate programs to find a program that will optimize the chosen performance metric and this is guided by their training experience. Statistical decision theory and Computational complexity theory are used to characterize the sample complexity, computational complexity and their dependency on the features of the learning algorithm.

The need for Machine learning has arisen in the recent years because of the rapid increase in the wealth and diversity of data from individuals through mobile devices and embedded computing. Machine learning uses the data to obtain useful insights, make predictions and decisions and even customize the different needs of individuals based on the information they have generated. This leads to increasingly data-intensive, evidence-based decision making across different fields like science, commerce and government.

The different types of Machine learning include Supervised learning (Systems forming their predictions through a learned mapping f(x) which produces output y for each input x), Deep learning (Systems using gradient-based optimization algorithms to adjust parameters throughout a multi-layered network based on the error at its output), Unsupervised learning (Systems that analyze unlabeled data under assumptions about the structural properties of the data), Reinforcement learning (System that learns through reward signals gained based on whether the action taken is correct or not), Active learning (Systems that choose data points and question the trainer to request the label of an unlabeled data point) and Semi-supervised learning (Systems that use unlabeled data to supplement the labeled data in a supervised learning context). Model selection is used to select the best model from a family of models (Eg: Bayesian Optimization). Causal modeling is used to find which variables causally influence others.

Environment in Machine learning algorithm refers to the computing architecture (machine learning systems contain complex collections of software which are run on large-scale parallel and distributed computing platforms), source of data (ranges from people with privacy concerns, analysts that have requirements like having the output visualizable and to the social, legal, political framework surrounding the deployment of a system) and includes other machine learning systems, agents.

The recent line of research involves Privacy-enhancing machine learning system (decision theoretic framework providing users with the ability to choose the desired level of privacy based on the different kinds of questions that would be asked and the personal utility for the answers), Probably approximately correct learning framework (the study on observing the effect of adding polynomial-time computation constraint over error rates, training data size and other parameters), Subsampling, Random projections, Supply time and space budgets to machine learning systems in addition to accuracy requirements, Construct a computer lifelong that operate nonstop for years where it will improve its ability to learn a new skill having learned another and Machine learning methods that work in collaboration with humans to analyze complex datasets to generate and explain various hypothesis.

3. Bias in Data Analysis

Bias is generally defined as a skew that produces a type of harm. In Data Analysis, it generally comes from training data. The training data that is collected can be incomplete, biased or skewed. The model that is built might seem not biased but in reality it is biased as the data on which the model was trained was biased.

Bias in Machine Learning as a Service is difficult to identify and it is also correct as we do not build them from scratch and we do not know how it works under the hood. Bias should be viewed both as a technical as well as a social issue.

Datasets reflect not only the culture but also the hierarchy of the world that they were made in. Therefore the data is not neutral and cannot be neutralized. Classifications that are made on this data can be sticky and sometimes they stick around longer than they were intended to even when they are harmful.

Bias can be understood as harms of allocation and harms of representation. Harms of allocation takes an economically-oriented view while Harms of representation takes a culture-oriented view. Harms of allocation is when a system allocates or withholds certain groups, opportunities or resources. It is immediate, a time-bound moment of decision making and is quantifiable. Harms of representation is when a system that represent society but does not allocate resources and instead reinforces the subordination of certain groups based on the identity like color, race, gender, class etc. It is a long-time process that affects attitudes and beliefs, harder to formalize and track and is the root of all other forms of allocative harms.

The different types of allocative harms are discussed below:

- *Recognition*: This occurs when a group is erased or made invisible by a system. It includes the failure to recognize someone's humanity and denies the respect, dignity and personhood of an individual or community. E.g.: System that does not process darker skin tones
- *Denigration*: When people use culturally offensive or inappropriate labels. E.g.: Autosuggestions when people type "Jews should"
- *Under-representation*: The minority is not properly represented. E.g.: When doing an image search for "CEO" returns only one woman CEO at the bottom of the page while the remaining were all male.
- *Ex-nomination*: Majority hides its identity by not referring to themselves in such as a way to naturalize their ideology. E.g.: In western culture, when the word "male" is used it is assumed to be a white heterosexual male.

The technical responses to bias include improving accuracy, blacklisting, scrubbing to neutral, equal representation and creating awareness about the situation. We can do this by working on fairness forensics by testing the systems and tracking the life cycle of the training dataset to know who creates the dataset and analyzing what the demographics skews are in that dataset. This can be avoided by collaborating with people from different fields to avoid ethics of classification. The most important question that needs to be addressed in machine learning regarding fairness is "Who is going to benefit from the system we are building and who might be harmed?"

4. Ten Simple Rules for Responsible Big Data Research

The below 10 rules were developed by a group of 20 scholars funded by NSF to encourage ethical practices in scientific and engineering research while utilizing big data research methods and infrastructures.

Acknowledge that data are people and can do harm: Even public, benign or neutral data that seem to be nowhere related to people may cause unintentional breaches of privacy, shape the lives of individuals, stigmatize groups and cause social inequities. Therefore whenever doing research assume that data are people until proven otherwise and then proceed with the analysis.

Recognize that privacy is more than a binary value: Data which is shared publicly does not mean using that data would be unproblematic or that the data creators gave consent to use the data. Privacy norms differ from individual to individual and different groups. Data gathered must be situated and contextualized to anticipate privacy breaches and minimize the harm

Guard against the re-identification of your data: When published results of analysis of datasets which seem to have been anonymized are looked along with other parameters can lead to the identification of an individual and their personal history. It is very important to identify all possible vectors of re-identification in the data and minimize them before publishing the results.

Practice ethical data sharing: Data gathered for research can be collected without the informed consent of the participant through data clouds or through the broad consent from the participant. Researchers must inform funding agencies about all possible ethical concerns, likelihood of privacy breaches, and potential harm of the collected data. They should make sure to guard against re-identification keeping in mind their responsibilities to the humans behind the data they acquired before sharing the data collected as well as their work with others.

Consider the strengths and limitations of your data; big does not automatically mean better: It is important to understand the source of data and the rules and regulations with which they were gathered. Evolution on the dataset must be tracked, bugs in the data must be found and articulating what the data or indicator represents and does not represent. The data's context must be clearly understood to check whether the data and analysis is working and not to overstate the clarity of the research findings. It is essential to acknowledge that the data acquired might have multiple meanings and might be messy.

Debate the tough, ethical choices: When encountered with new situations related to ethical issues for which there is no precedence and is outside the mandate of IRB's then researchers can debate about these issues within groups of their peers. It would be beneficial that the people participating in the debate belong to different discipline so that the issue discussed can be understood from a wide range of perspectives, giving voice to their insights ultimately leading to a beneficial policy developed to solve the issue.

Develop a code of conduct for your organization, research, community or industry: Codes of conduct should be developed within the organization or research community regarding the handling of data generated by people. Developing the code of conduct should involve feedback from researchers as well as from representatives of different affected communities. It acts as a guidance while researching as well as during publishing results to ensure that no issues arises or any ethical violation occurs.

Design your data and systems for auditability: Automated testing can be used for assessing those problematic outcomes which were documented while auditing. Auditing is used to keep track of what decision were made, when they were made, backtrack to an earlier dataset and address the issue at the root. It helps researchers to double check their work which leads to an increase in understandability and replicability.

Engage with the broader consequences of data and analysis practices: Researchers should not only try to understand the world but try to improve it. This means that instead of finding not only ways to improve the big data research they can also focus on issues like climate change, protecting the privacy of public resources and ensure that their research benefits even the underserved members of the society.

Know when to break these rules: The above rules can be broken during an emergency but what constitutes an emergency must be first analyzed by reviewing regulatory expectations and legal demands of the data that is being dealt with. To know when to break the rules, comes with experience by participating in debates, being involved in the formulation of code of conducts and by developing formal auditing systems for their research.

Responsible big data research does not mean restricting or preventing research but making absolutely sure that the work is sound, accurate, maximizes the good while minimizing the harm and making sure not to lose the public's trust.

5. Finite – Horizon MDP's

a. Real-World Examples

*State Reward Function***:** In a game of roulette, the goal is to predict in which number the ball will fall. Here, the state here is the position of the ball. If the ball falls on the selected number then the RL agent receives a reward and if it does not then the RL agent receives a penalty.

*State-Action Reward Function***:** In a game of chess, the goal is to check-mate the king. Here the RL agent will receive a reward based on the action taken at each state whether the action leads them to victory and if the action that they take at that state will lead to their defeat, then they will receive a penalty.

b. Modify finite horizon value iteration algorithm

Original Finite Horizon Value Iteration Algorithm:

$$V^0(s) = R(s), \forall s$$

$$V^k(s) = R(s) + \underbrace{max}_{a} \sum_{s'} T(s,a,s')\, V^{k-1}(s')$$

$$\pi^*(s,k) = arg\ \underbrace{max}_{a} \sum_{s'} T(s,a,s')\, V^{k-1}(s')$$

Update Equations:

$$V^0(s) = 0, \forall s$$

Since rewards are assigned only when an action is taken. Initially at the beginning no action would have been taken therefore we set it to zero

$$V^k(s) = \underbrace{max}_{a}\ R(s,a) + \sum_{s'} T(s,a,s')\, V^{k-1}(s')$$

$R(s,a)$ is inside the maximization function since the rewards are assigned only when an action is taken, therefore we need to maximize it.

$$\pi^*(s,k) = arg\ \underbrace{max}_{a} \sum_{s'} T(s,a,s')\, V^{k-1}(s')$$

c. MDP with a state-action reward function R(s, a) can be transformed into a different MDP with state reward function R(s), such that the optimal policies in the new MDP correspond exactly to the optimal policies in the original MDP

We are given an MDP M with a state-action reward function $R(s,a)$. A new MDP M^{new} with state reward function $R^{new}(s)$ is created. Let the state set be S, action set be A, transition function be T and reward function of M be $R(s,a)$.

For M^{new}, the state set S^{new} will contain all the states which are present in S along with a new set of states $q_{s,a}$ for each state action pair of M. The action set be A^{new}, transition function be T^{new} and reward function of M^{new} be $R^{new}(s)$.

$$T^{new}(s, a, q_{s,a}) = 1$$

$$T^{new}(q_{s,a}, a^{new}, s^{new}) = T(s, a, s^{new})$$

$$R^{new}(s) = 0$$

$$R^{new}(q_{s,a}) = R(s,a)$$

When a RL agent is in a state s of M^{new} and taken an action a, it receives a zero reward based on the transitions to state $q_{s,a}$. In $q_{s,a}$. We get reward $R(s,a)$ which is same as the reward we would have got in M had we taken a in s. When we are in $q_{s,a}$ for any action that is taken by the RL agent, then there will be a transition to a regular state s' with the transition probability $T(s,a,s')$. Therefore the transition probability from $q_{s,a}$ is the same as the probability of going from s to s' after taking a in M. Therefore when the RL agent is in the state s of M^{new} and performs the action a followed by any other action, it will get a reward $R(s,a)$ after 2 steps and then will transition to a state s' with the same probability if the RL agent had performed a in s. Therefore a single action in M is the same as 2 actions in M^{new}.

If the policy function for M is $\pi(s,n)$ where n is the number of steps to go then, the policy function for M' is $\pi^{new}(s, 2n)$

6. k-th order MDP's

The state space for M' is

$$S' = S^k$$

So, each state in S' is of the form $s, s_1, s_2 \dots s_{k-1}$ where each value is a state in S.

The actions for M' is

$$A' = A.$$

The reward function for M' is

$$R'\big((s, s_1, s_2 \dots s_{k-1})\big) = R(s)$$

This implies that the reward in the new MDP depends only on the current state, s of M.

The transition function for M' is

$$T'\big((s, s_1, s_2 \dots s_{k-1}), a, \vec{s}\big) = \begin{cases} P_r(s'|a, s, s_1, s_2 \dots s_{k-1}) \text{ , } if\ \vec{s} = s, s_1, s_2 \dots s_{k-2} \\ 0 \qquad\qquad\qquad \text{, } otherwise \end{cases}$$

This means that the transition function makes sure that the history is maintained correctly after the state transitions and also makes sure that the new state s' has a probability given by the k-th order model. There is also a zero transition probability of moving to a state that does not update the history correctly and instead involves shifting the history in the current state by one step.

The policy for M' is

$$\pi'\big((s, s_1, s_2 \dots s_{k-1})\big) = \pi((s, s_1, s_2 \dots s_{k-1}))$$

This shows that for any k-order MDM M, there is an equivalent MDP M'.

7. Bellman Optimality Equation

Case 1: Reward function R(s)

$$V^*(s) = R(s) + \beta \underbrace{max}_{a} \sum_{s'} T(s, a, s')\, V^*(s')$$

Case 2: State-Action Reward function R(s,a)

$$V^*(s) = \underbrace{max}_{a}\ R(s, a) + \beta \sum_{s'} T(s, a, s')\, V^*(s')$$

Case 3: State-Action-State Reward function R(s,a,s')

$$V^*(s) = \underbrace{max}_{a} \sum_{s'} T(s, a, s')\, R(s, a, s') + \beta\, V^*(s')$$

8. Single Policy for MDP that takes action a in both states

a. Linear Equations

Let:

Policy = π

$V_0 = V^{\pi}(s_0)$

$V_1 = V^{\pi}(s_1)$

$R(s_0) = 0$

$R(s_1) = 1$

Linear Equations:

$V_0 = R(s_0) + \beta V_1 = \beta V_1$

$V_1 = R(s_1) + \beta V_1 = 1 + \beta V_1$

When $\beta = 1$

$V_0 = V_1$

$V_1 = 1 + V_1$

When we try to solve the above equations to get V_0 and V_1, no solution exists which means that the policy does not have a well-defined finite value function.

b. Linear Equations with discount factor

When $\beta = 0.9$

$V_0 = 0.9V_1$

$V_1 = 1 + 0.9V_1$

When we solve the above equations to get V_0 and V_1, we get $V_0 = 9$ and $V_1 = 10$

9. Gridworld Environment

Gridworld Q-Learning

Gridworld Environment

```
[[  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.]
 [  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.]
 [  0. 999. 999. 999. 999.   0. 999. 999. 999.   0.]
 [  0.   0.   0.  -1. 999.   0.   0.   0.   0.   0.]
 [  0.   0.   0.   0. 999.  -1.  -1.   0.   0.   0.]
 [  0.   0.   0.   0. 999.   1.  -1.   0.  -1.   0.]
 [  0.   0.   0.   0. 999.   0.   0.   0.  -1.   0.]
 [  0.   0.   0.  -1. 999.  -1.  -1.   0.   0.   0.]
 [  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.]
 [  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.]]
```

Size of the environment
(10, 10)

Epsilon Greedy Exploration Policy

Epsilon = 0.100000
Number of iterations: 233
Q-table

```
[[ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
```

```
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.       -0.029701]
[ 0.        0.        0.       -0.01    ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.01    ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
```

```
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]]

Epsilon = 0.200000
Number of iterations: 149
Q-table
[[ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.        ]
```

```
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
[ 0.        0.        0.        0.        ]
```

```
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.01    ]
[ 0.        0.        0.       -0.01    ]
[ 0.        0.        0.        0.      ]
[ 0.        0.       -0.01      0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.       -0.029701]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
[ 0.        0.        0.        0.      ]
```

```
 [ 0.        0.        0.        0.      ]
 [ 0.        0.        0.        0.      ]
 [ 0.        0.        0.        0.      ]
 [ 0.        0.        0.        0.      ]
 [ 0.        0.        0.        0.      ]
 [ 0.        0.        0.        0.      ]
 [ 0.        0.        0.        0.      ]
 [ 0.        0.        0.        0.      ]]
```

Epsilon = 0.300000
Number of iterations: 31
Q-table

```
[[ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
 [ 0.   0.   0.   0.  ]
```

```
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.   -0.01]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.01]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
[ 0.    0.    0.    0.  ]
```

```
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.  ]]
```

Boltzman Exploration Policy

The temperature T value is started at T = 10. After every 10 steps the temperature is reduced by 0.05

Unique temperature values used

```
[10.   9.95  9.9   9.85  9.8   9.75  9.7   9.65  9.6   9.55  9.5   9.45
  9.4   9.35  9.3   9.25  9.2   9.15  9.1   9.05  9.    8.95  8.9   8.85
  8.8 ]
```

Total unique temperature values = 25

Number of iterations: 249

Q-table

```
[[ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
```

```
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.         -0.01       -0.01      ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.         -0.01      ]
[ 0.          0.          0.         -0.05851985]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
[ 0.          0.          0.          0.        ]
```

```
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.01  ]
[ 0.     0.     0.    -0.01  ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.    -0.01   0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
[ 0.     0.     0.     0.    ]
```

```
[ 0.    0.    0.    0.    ]
[ 0.    0.    0.    0.    ]
[ 0.    0.    0.    0.    ]
[ 0.    0.    0.    0.    ]
[ 0.    0.    0.    0.    ]
[ 0.    0.    0.    0.    ]]
```

10. Mnist

MNIST Digits Classification

Training Data

Loss: 0.012170

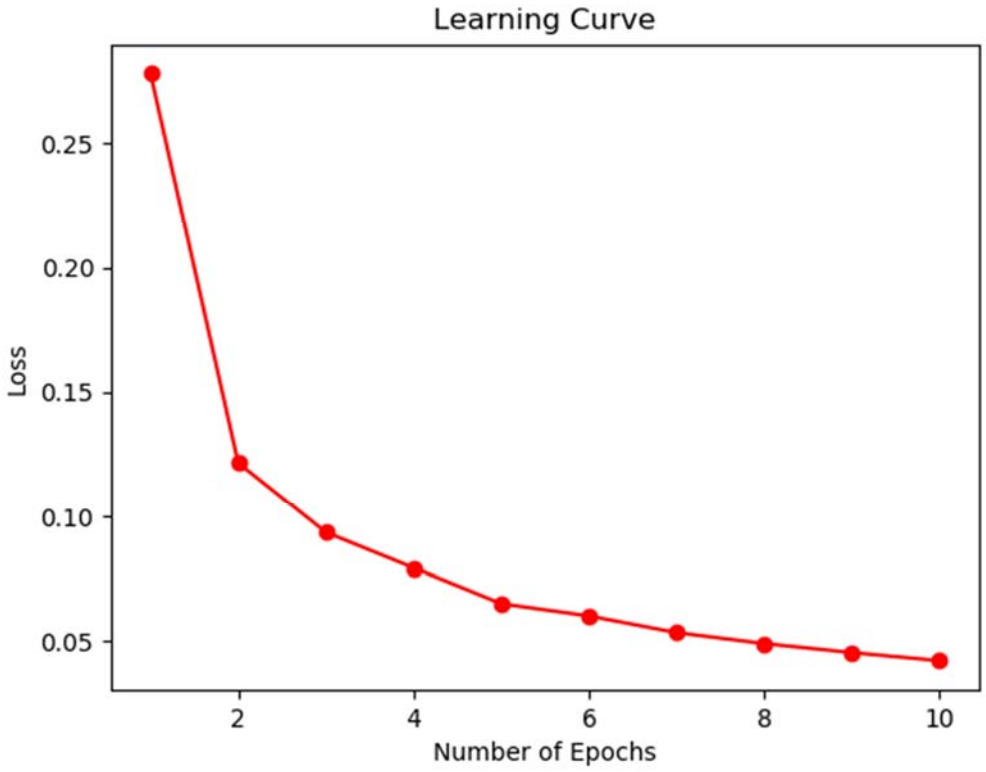
Accuracy: 99.673333

Testing Data

Loss: 0.037946

Accuracy: 98.900000

Learning Curve



As the number of epochs increases, loss reduces