Homework 4

Reet Barik, WSU ID: 11630142
CptS 570 Machine Learning

December 10, 2018

Exercise 1. 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 C1 and C2 are B and C respectively.

Answer:

The data points in the problem are: 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).

Iteration 1:
Taking C1 = (1.5,2) and C2 = (3,4),
Distance(A,C1) = $\sqrt{(1.5-1)^2+(2-1)^2} = 1.118$
Distance(B,C1) = 0
Distance(C,C1) = $\sqrt{(1.5-3)^2+(2-4)^2} = 2.5$
Distance(D,C1) = $\sqrt{(1.5-5)^2+(2-7)^2} = 6.103$
Distance(E,C1) = $\sqrt{(1.5-3.5)^2+(2-5)^2} = 3.605$
Distance(F,C1) = $\sqrt{(1.5-4.5)^2+(2-5)^2} = 4.24$
Distance(G,C1) = $\sqrt{(1.5-3.5)^2+(2-4.5)^2} = 3.201$
Distance(A,C2) = $\sqrt{(3-1)^2+(4-1)^2} = 3.65$
Distance(B,C2) = $\sqrt{(3-1.5)^2+(4-2)^2} = 2.5$
Distance(C,C2) = 0
Distance(D,C2) = $\sqrt{(3-5)^2+(4-7)^2} = 3.605$
Distance(E,C2) = $\sqrt{(3-3.5)^2+(4-5)^2} = 1.118$
Distance(F,C2) = $\sqrt{(3-4.5)^2+(4-5)^2} = 1.802$
Distance(G,C2) = $\sqrt{(3-3.5)^2+(4-4.5)^2} = 0.7071$

The clusters formed are:
Cluster 1 : A = (1,1); B = (1.5,2)
Cluster 2 : C = (3.0, 4.0); D = (5.0, 7.0); E = (3.5, 5.0); F = (4.5, 5.0); G = (3.5, 4.5)

Iteration 2:

The new Centroids are:
C1 = (.25,1.5) and C2 = (3.9,5.1)
Distance(A,C1) = $\sqrt{(1.25-1)^2+(1.5-1)^2} = 0.559$
Distance(B,C1) = $\sqrt{(1.25-1.5)^2+(1.5-2)^2} = 0.559$
Distance(C,C1) = $\sqrt{(1.25-3)^2+(1.5-4)^2} = 3.051$
Distance(D,C1) = $\sqrt{(1.25-5)^2+(1.5-7)^2} = 6.6567$
Distance(E,C1) = $\sqrt{(1.25-3.5)^2+(1.5-5)^2} = 4.1608$
Distance(F,C1) = $\sqrt{(1.25-4.5)^2+(1.5-5)^2} = 4.776$
Distance(G,C1) = $\sqrt{(1.25-3.5)^2+(1.5-4.5)^2} = 3.75$
Distance(A,C2) = $\sqrt{(3.9-1)^2+(5.1-1)^2} = 5.0219$
Distance(B,C2) = $\sqrt{(3.9-1.5)^2+(5.1-2)^2} = 3.920$
Distance(C,C2) = $\sqrt{(3.9-3)^2+(5.1-4)^2} = 1.421$ Distance(D,C2) = $\sqrt{(3.9-5)^2+(5.1-7)^2} =$ 2.1954
Distance(E,C2) = $\sqrt{(3.9-3.5)^2+(5.1-5)^2} = 0.412$ Distance(F,C2) = $\sqrt{(3.9-4.5)^2+(5.1-5)^2} =$ 0.6082 Distance(G,C2) = $\sqrt{(3.9-3.5)^2+(5.1-4.5)^2} = 0.7211$

The clusters formed are:
Cluster 1 : A = (1,1); B = (1.5,2)
Cluster 2 : C = (3.0, 4.0); D = (5.0, 7.0); E = (3.5, 5.0); F = (4.5, 5.0); G = (3.5, 4.5)

Exercise 2. 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

Answer:

Machine learning focuses on answering two questions: 1. How to construct computer systems that automatically improve through experience? 2. What are the fundamental statistical- computational-information-theoretic laws that govern all learning systems. Over the past decade, it has found use in many fields as it is all about automating the process of problem solving. It deals with data that we generate, then relating it to statistics and mathematics domain. Examples of recent successful application of Machine Learning include robotics and autonomous vehicle control, speech processing and natural language processing, neuroscience research, and applications in computer vision.

Most widely used machine learning methods are supervised learning methods which generally form their predictions via a learned mapping f(x), which produces an output y for each input x (or a probability distribution over y given x). Primary examples of supervised learning methods include decision trees, decision forests, logistic regression, support vector machines, neural networks, kernel machines, and Bayesian classifiers. The area involving

Deep Networks has seen a high impact in recent years. Exploiting modern parallel computing architectures, such as graphics processing units originally developed for video gaming, it has been possible to build deep learning systems that contain billions of parameters and that can be trained on the very large collections of images, videos, and speech samples available on the Internet.

The other widely used methodology is called unsupervised learning. This involves analysis of unlabeled data under assumptions about its structural properties. A criterion function is defined that embodies these assumptionsoften making use of general statistical principles such as maximum likelihood, the method of moments, or Bayesian integrationand optimization or sampling algorithms are developed to optimize the criterion.

The third learning paradigm is called Reinforcement Learning. Here, instead the training data, instead of indicating the correct output of the given input, merely provides an indication whether the action is correct or not. Reinforcement-learning algorithms generally make use of ideas that are familiar from the control-theory literature, such as policy iteration, value iteration, rollouts, and variance reduction.The scope of machine learning keeps increasing, advancements on processing information like humans are being made. One of the major concerns with these developments is privacy. As learning involves inputting large amounts of data, the data that is used may violate privacy of a person. This is a huge security risk which needs to be controlled. The trade off is between privacy and technological advancements for a better world. Another resource that needs to be managed within the overall context of the distributed learning system is communication. Research along this line will to bring the kinds of statistical resources studied in machine learning into contact with the classical computational resources of time and space. Such a bridge is present in the probably approximately correct (PAC) learning framework, which studies the effect of adding a polynomial-time computation constraint on this relationship among error rates, training data size, and other parameters of the learning algorithm.

Despite its practical and commercial successes, machine learning remains a young field with many underexplored research opportunities. Some researchers have begun to explore the question of how to construct computer lifelong or never-ending learners that operate nonstop for years, learning thousands of interrelated skills or functions within an overall architecture that allows the system to improve its ability to learn one skill based on having learned another. All in all, despite being a relatively young field society as a whole has started accepting machine learning to be a part and parcel of daily life.

Exercise 3. 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

Answer:

In an enlightening talk given by Kate Crawford at NIPS 2017 aptly titled 'Trouble with the bias', she talks about some of the problems faced by the Machine Learning community in terms of forms of bias stereotyping and unfair determinations everywhere from machine vision systems and object recognition to natural language processing and word embedding. These challenges have manifested themselves in real world application problems like gender biased applications, object classification datasets promoting racial disparities etc. This is because the long histories of discrimination live on in our digital systems often for very complex reasons and they become buried into the logics of our machine learning infrastructures. To meet this challenge the quantity of research in the field of machine learning fairness and bias has shot up in the recent years, especially in 2017. This has led to the misconception of data being inherently unbiased being broken.

Going forward, the talk broadly addressed the following topics:

1. What is bias?

In this, Kate talks about the history of the word 'bias', starting from the time it was used to refer to an oblique line in geometry to its technical meaning in statistics where it referred to to systematic differences between a sample and a population to its common definition in the machine learning community nowadays that correspond to underfitting. The popular and the legal definition means judgment based on preconceived notions or prejudices as opposed to impartial evaluation of facts. Bias has a way of creeping into the data because labeling is almost always done by a human. Hence, datasets are always influenced by the socio-cultural climate of the time which tend to make any application based on them skewed.

2. Harms of allocation.

Here, Kate brings to our attention that the literature available right now currently understands bias as producing harms of allocation which occurs when a system allocates or withhold certain groups an opportunity or resource. This is from an economic point of view and can be used to explain harms like who gets a mortgage, who gets a loan, who gets insurance etc.

3. Harms of representation.

This is different from harms of allocation as it refers to occurrences when when systems reinforce the subordination of some groups along the lines of identity, that is, race class gender etc. This has resulted in some infamous incidents like Google photos classifying the picture of an African American woman as that of a Gorilla.

4. Politics of classification.

In this section, Kate brings to our attention how classification systems are often sites of political and social struggle. This is so because political agendas are sometimes presented as purely technical and then hidden away from the public and they gradually become taken for granted. A prime example of this is how it was used during the height of Apartheid in South Africa in the 1970s it classified people into one of four categories: colored, Indian, white or black and it was then it was made technically possible by building it into a system by IBM. Depending on what category you were classified in it would determine where you could live, what job you could have, and who you could marry.

5. What can we do?

As a remedy that can be applied by the machine learning community as a whole, Kate suggest we carry out rigorous fairness forensics which includes pre-release of software to different parts of the world to judge its performance. Another helpful way would be to track the lifecycle of training data by looking at its author and trying to figure out the demographic skews attached to it. This can be done by giving the interdisciplinary aspect of the whole operation a lot more importance that it has been till now. The final recommendation comes in the form of an advice to think of the ethical implications of what type of systems should be or shouldn't be built.

Exercise 4. 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

Answer:

There has been a tremendous growth in the use of big data research methods which has confronted the world with various ethical questions as a result of the tools being increasingly woven into our day to day lives. As an answer to these ethical questions raised by irresponsible big data research methods, this article has provided ten simple rules for addressing the complex ethical issues that will inevitably arise. They are as follows:

1. Acknowledge that data are people and can do harm:

This talks about the recognition of the fact that most data represent or impact people. Even seemingly innocuous data some time end up revealing more than intended. At other times, some publicly available datasets are used for highly invasive research that compromises the security and privacy of the affected individuals. This rule aims to prevent such occurrences.

2. Recognize that privacy is more than a binary value:

This is the recognition of the fact that privacy has a situational and contextual element to it and does not have a black and white nature corresponding to somethings which are public and somethings which are private. Technologies which can perceived as 'creepy' even when they are not violating privacy laws, like social media apps which take into account your location to send you targeted ads, fall under this rule. This rule also addresses the practices against communities which have been on the receiving end of discriminatory data-driven policies historically.

3. Guard against the reidentification of your data:

According to this rule, any researcher should make sure that the use of data doesn't lead to re-identification. This means that the data has to be annonymized sufficiently so that identification of individuals to whom the data refer to is not possible. Technologies such as google reverse image or ones dealing with 'aggregate statistics' fall under this umbrella.

4. Practice ethical data sharing:

Sharing of data is an important factor in order to find a cure in the medical field. But sharing must only be done upon the patients consent. When sharing , the researchers must also factor in reidentification issues and privacy breaches.

5. Consider the strengths and limitations of your data; big does not automatically mean better:

This rule recommends that the context of the data be kept in mind since it provides the foundation for clarifying when your data and analysis are working and when they are not. Instead of blindly interpreting findings based on big data research, one must be sensitive to the data's potential to have multiple meanings.

6. Debate the tough, ethical choices:

More often than not, ethical issues arise from big data research that are outside the mandate of governance of established IRBs that are traditionally charged with preventing harm through well-established procedures. The article emphasizes the importance of debating those issues within the group of peers so as to prevent unethical decisions from being taken.

7. Develop a code of conduct for your organization, research community, or industry:

Ethical practices with respect to big data research needs to be internalized by the research community rather than considered as an afterthought and thus propagating the culture of "faking ethics". According to this article, the best way to incorporate this in a permanent manner is to develop codes of conduct for use in your organization or research community and for inclusion in formal education and ongoing training.

8. Design your data and systems for auditability:

According to this article another way of showing responsibility by big data researchers is to design their system in such a way which makes it easy to audit. When such practices are incorporated, the whole chain of collecting data, cleaning, labeling and the subsequent research becomes relatively free of associated unethical issues by providing a mechanism for double-checking work and forcing oneself to be explicit about decisions, increasing understandability and replicability.

9. Engage with the broader consequences of data and analysis practices:

This rule suggests that researchers understand the impact big data can have on the society and redefine what constitutes success as far as their research is concerned by expanding the boundaries of their research such that it can be used to combat some of the more prevalent issues faced by the contemporary society of their time.

10. Know when to break these rules:

This rule asks responsible researchers to be prepared to come out of the 'meeting the checklist' mindset when it comes to their research in terms of ethics and to do what's necessary when the situation demands it. They have to recognize the fact the same set of rules that work well generally are not applicable in case of emergencies and have to be disregarded according to the best of their judgment.

In conclusion, these ten rules were made so as not to limit research but to make it more transparent, accurate, sound while at the same time maximizing good and minimizing harm.

Exercise 5. 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) 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) 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) 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.

Answer:

a) Let there be a set of 'n' games, G = $G_1, G_2, ..., G_n$ with $F_1, F_2, ..., F_n$ as the respective fees associated to them. This is a real world problem which can modeled as an MDP using a state-action reward function where reward equal to the negative cost of the machine.

b) The Finite-horizon value iteration algorithm is given by:

$$V^0(s) = R(s)$$
$$V^{k+1}(s) = R(s) + max_{a \in A} \sum_{s' \in S} T(s, a, s')V^k(s')$$

For state-action reward function, the updation is as follows:

$$V^0(s) = 0$$
$$V^{k+1}(s) = max_{a \in A} R(s, a) + \sum_{s' \in S} T(s, a, s')V^k(s')$$

The initialization is done to 0 instead of R(s) because the value is zero since no action has been taken yet. The maximization is put over R(s,a) because it is dependent on the action.

c) Given an MDP M with a state-action reward function R(s,a) we need to transform it into an MDP M' with a state reward function R'(s). New 'book keeping' states have to introduced in M' that will keep track of the state-action pairs. A new set of states in M' is given by $\{q_{s,a} | s \in S, a \in A\}$ where S, A, T is the set of states, set of actions, and the transition function in M. The transformation from M to M' is as follows:

$$T'(s, a, q_{s,a}) = 1, \forall s \in S, a \in A$$
$$T'(q_{s,a}, a', s') = T(s, a, s'), \forall s \in S, a \in A, a' \in A, s' \in S$$
$$R'(s) = 0, \forall s \in S$$
$$R'(q_{s,a}) = R(s, a), \forall s \in S, a \in A$$

The way it works is that if the agent is in state s, and takes an action a, it deterministically moves to the state $q_{s,a}$ with a reward of zero. Once in $q_{s,a}$, it will get the reward R(s,a) which is the same reward that the agent would have gotten for taking the action a while at state s in M. When in $q_{s,a}$, for any action, the agent will move to a regular state s' with the transition probability given by T(s,a,s') which is the same as the probability of going from s to s' after taking a in M.

Exercise 6. 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 k1 states. That is, $T(s_{k-1}, ..., s_1, s, a, s')$ $= Pr(s'|a, s, s_1, ..., s_{k-1})$ gives the probability of transitioning to state s' given that action a was taken in state s and the previous k - 1 states were $(s_{k-1}, ..., s_1)$.

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.

Answer:

The desired correspondence between M and M' is as follows:
The state-space S' of M' is given by S^k where S is the state-space in M and k is the order of M'. Each state in S' is of the desired form which is $(s_{k-1}, ..., s_1)$. The action set of M' same as that of M. Therefore, A' = A. The reward function of M' is defined as R'($(s_{k-1}, , s_1)$) = R(s). This implies that the reward in M' depends only on the current state s. The transition function

$$T'((s_{k-1}, , s_1), a, \vec{s}) = Pr(s'|a, s, s_1, ..., s_{k-1}),\ if\ \vec{s} = (s_{k-2}, ..., s_1)$$
$$= \ 0\ ,\ in\ all\ other\ cases$$

This shows that the probability of moving to a state that does not maintain the history of states correctly is zero.
Now that we have the formulation of M', we can solve it to get a policy π' over the states in it. The mapping is given as $\pi(s_{k-1}, ..., s_1) = \pi'((s_{k-1}, ..., s_1))$.
Also, it has to be noted that the transformation was made possible by significantly increasing the size of the state-space. The number of states in M' ends up being an exponential in k.

Exercise 7. 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.

Answer:

Case 1: For the MDPs that use state only reward functions R(s), the Bellman equation is given by:

$$V^*(s) = R(s) + \beta max_{a \in A} \sum_{s' \in S} T(s, a, s')V^*(s')$$

Case 2: For the MDPs that use a state-action reward function R(s,a), the Bellman equation can by obtained by moving the reward function inside the maximization over actions:

$$V^*(s) = max_{a \in A} R(s) + \beta \sum_{s' \in S} T(s,a,s') V^*(s')$$

Case 3: For the MDPs using a state-action-state rewards function R(s,a,s'), the Bellman equation can be obtained by moving the reward inside the expectation since it depends on the next state as well as shown:

$$V^*(s) = max_{a \in A} \sum_{s' \in S} T(s,a,s')(R(s,a,s') + \beta V^*(s'))$$

Exercise 8. Consider a trivially simple MDP with two states S = s0, s1 and a single action A = a. The reward function is R(s0) = 0 and R(s1) = 1. The transition function is T(s0, a, s1) = 1 and T(s1, a, s1) = 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 $\beta = 0.9$.

Answer:

a) Let the policy be denoted by π. Let $V_0 = V^\pi(s_0)$ and $V_1 = V^\pi(s_1)$. Then the linear equations are:

$$V_0 = R(s_0) + \beta V_1 = \beta V_1$$
$$V_1 = R(s_1) + \beta V_1 = 1 + \beta V_1$$

Putting $\beta = 1$, we get,

$$V_0 = V_1$$
$$V_1 = 1 + V_1$$

Since this has no solution, it can be surmised that the policy does not have a well defined finite value function.

b) Similarly, if we put $\beta = 0.9$, we get the following:

$$V_0 = 0.9V_1$$
$$V_1 = 1 + 0.9V_1$$

Solving the above, we get $V_0 = 9$ and $V_1 = 10$. Hence, it can be said that including a discount factor $\beta \in [0, 1)$ results in a well conditioned system.

Exercise 9. 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 $\beta = 0.9$.
Implement the Q-learning algorithm (slide 46) to learn the Q values for each state-action pair. Assume a small fixed learning rate $\alpha = 0.01$.
Experiment with different explore/exploit policies:
1) ϵ-greedy. Try 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.

Answer:

Running two different explore/exploit policies:
A. Epsilon-greedy, epsilon = [0.1, 0.2, 0.3]
B. Boltzmann exploration, temperature = [1000, 800, 600, 500, 250, 100, 10, 1]

Doing Q-learning:
1. Running EpsilonGreedyPolicy, epsilon = 0.1
Number of iterations required to converge: 1136
2. Running EpsilonGreedyPolicy, epsilon = 0.2
Number of iterations required to converge: 1951
3. Running EpsilonGreedyPolicy, epsilon = 0.3
Number of iterations required to converge: 1447
4. Running BoltzmannExplorationPolicy, temperature = 1000
Number of iterations required to converge: 1000
5. Running BoltzmannExplorationPolicy, temperature = 800
Number of iterations required to converge: 799
6. Running BoltzmannExplorationPolicy, temperature = 600
Number of iterations required to converge: 600
7. Running BoltzmannExplorationPolicy, temperature = 500

```
Number of iterations required to converge: 495
8. Running BoltzmannExplorationPolicy, temperature = 250
Number of iterations required to converge: 250
9. Running BoltzmannExplorationPolicy, temperature = 100
Number of iterations required to converge: 100
10. Running BoltzmannExplorationPolicy, temperature = 10
Number of iterations required to converge: 39
11. Running BoltzmannExplorationPolicy, temperature = 1
Number of iterations required to converge: 25
```

For the converged Q-values for each state-action pairs of the above policies, please refer to 'output.txt'.

Exercise 10. 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).

Answer:

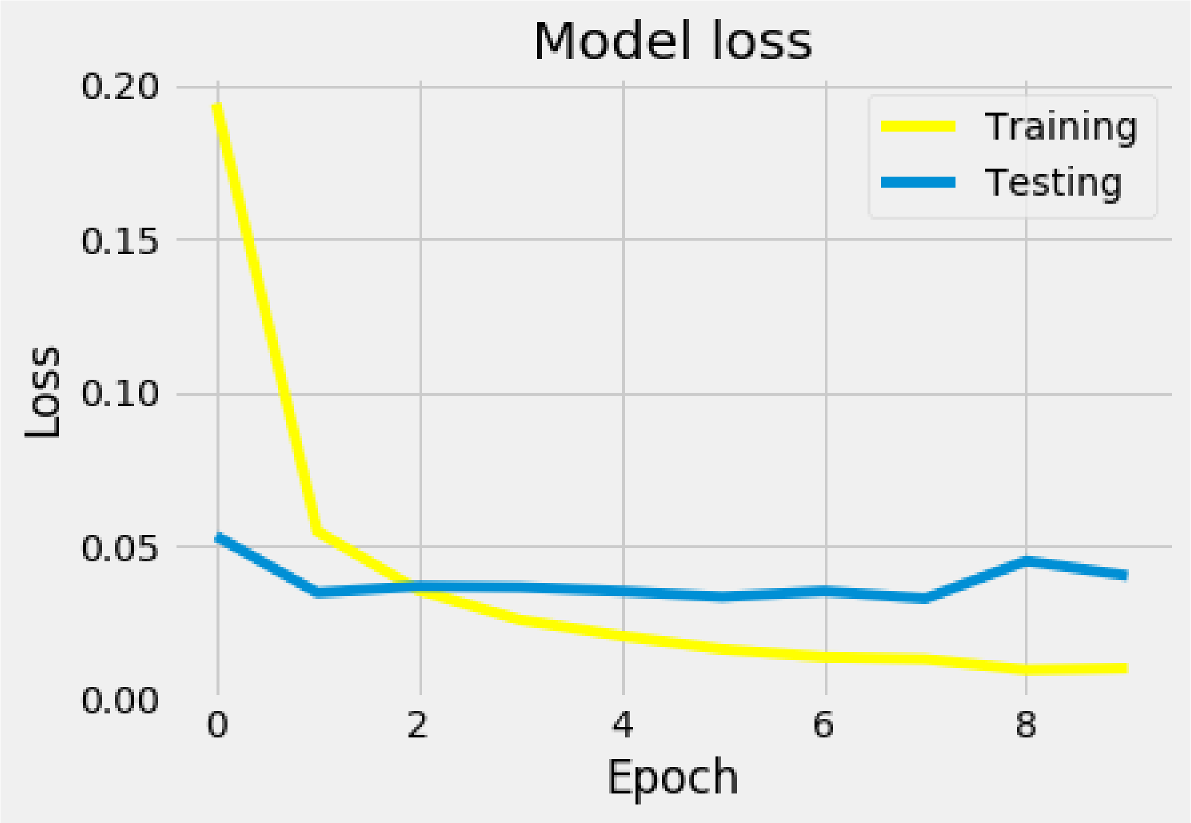
Train on 60000 samples, validate on 10000 samples

```
Epoch 1/10
- 20s - loss: 0.1942 - acc: 0.9425 - val_loss: 0.0523 - val_acc: 0.9831
Epoch 2/10
- 17s - loss: 0.0538 - acc: 0.9832 - val_loss: 0.0371 - val_acc: 0.9888
Epoch 3/10
- 17s - loss: 0.0350 - acc: 0.9894 - val_loss: 0.0380 - val_acc: 0.9874
Epoch 4/10
- 17s - loss: 0.0261 - acc: 0.9915 - val_loss: 0.0398 - val_acc: 0.9867
Epoch 5/10
- 17s - loss: 0.0207 - acc: 0.9935 - val_loss: 0.0395 - val_acc: 0.9871
Epoch 6/10
- 17s - loss: 0.0166 - acc: 0.9946 - val_loss: 0.0397 - val_acc: 0.9888
Epoch 7/10
- 17s - loss: 0.0135 - acc: 0.9956 - val_loss: 0.0334 - val_acc: 0.9907
Epoch 8/10
- 17s - loss: 0.0126 - acc: 0.9959 - val_loss: 0.0333 - val_acc: 0.9910
Epoch 9/10
```

```
- 18s - loss: 0.0093 - acc: 0.9968 - val_loss: 0.0407 - val_acc: 0.9898
Epoch 10/10
- 18s - loss: 0.0088 - acc: 0.9974 - val_loss: 0.0338 - val_acc: 0.9919

Test loss: 0.03377656660213915
Test accuracy: 0.9919
```

The learning curve (error as a function of training epochs) is as follows:


Model loss
Loss
Epoch
Training
Testing