

## Problem Set #3: EM, Deep Learning, & Reinforcement Learning

### Problem 1 Neural Networks: MNIST image classification

Implement a simple convolutional neural network to classify grayscale images of handwritten digits from the MNIST dataset. The starter code splits the set of 60000 training images and labels into a sets of 59600 examples as the training set and 400 examples for the dev set. To start, implement convolutional neural network and cross entropy loss, and train it with the provided dataset. The architecture is as follows:

- (a) The first layer is a convolutional layer with 2 output channels with a convolution size of 4 by 4.
- (b) The second layer is a max pooling layer of stride and width 5 by 5.
- (c) The third layer is a ReLU activation layer.
- (d) After the four layer, the data is flattened into a single dimension.
- (e) The fifth layer is a single linear layer with output size 10 (the number of classes).
- (f) The sixth layer is a softmax layer that computes the probabilities for each classes.
- (g) Finally, we use a cross entropy loss as our loss function.

The cross entropy loss is

$$CE(y, \hat{y}) = - \sum_{k=1}^K y_k \log \hat{y}_k,$$

where  $\hat{y} \in \mathbb{R}^K$  is the vector of softmax outputs from the model for the training example  $x$ , and  $y \in \mathbb{R}^K$  is the ground-truth vector for the training example  $X$  such that  $y = [0, \dots, 0, 1, 0, \dots, 0]^T$  contains a single 1 at the position of the correct classes.

We also use mini-batch gradient descent with a batch size of 16. Normally we would iterate over the data multiple times with multiple epochs, but for this assignment we only do 400 batches to save time.

- (a) Implement functions within `p01_nn.py`.
- (b) Implement a function that computes the full backward pass.

■

**Problem 2 Off Policy Evaluatino And Causal Inference**

Need methods for evaluating policies without actually implementing them. This task is off-policy evaluation or causal inference.

For this problem, consider MDPs with a single timestep. Consider universe of states  $S$ , actions  $A$ , a reward function  $R(s, a)$  where  $s$  is a state and  $a$  is an action. We often only have a subset of  $a$  in our dataset. For eaxmple, each state  $s$  could represent a patient, each action  $a$  could represent which drug we prescribe to that patient and  $R(s, a)$  be their lifespan after prescribing that drug.

A policy is defined as  $\pi_i(s, a) = p(a | s, \pi_i)$  We are given observational dataset of  $(s, a, R(s, a))$  tuples. Let  $p(s)$  be the density for the distribution of the state  $s$  values within the dataset. Let  $\pi_0(s, a) = p(a | s)$  within our observational data.  $\pi_0$  corresponds to the baseline policy in observational data. Also given target policy  $\pi_1(s, a)$  which gives the conditional probability  $p(a | s)$  in our optimal policy. Our goal is to computer expected value of  $R(s, a)$  in the same population as our observatinoal data, but with a policy of  $\pi_1$  instead of  $\pi_0$ . In other words, we are trying to compute

$$\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s, a)}} [R(s, a)] = \sum_{(s, a)} R(s, a) p(s, a) = \sum_{(s, a)} R(s, a) p(s) p(a | s) = \sum_{(s, a)} R(s, a) p(s) \pi_1(s, a).$$

**Simplifying Assumptions:** We will assume that each action has a non-zero probability in the observed policy  $\pi_0(s, a)$ . In other words, for all actions  $a$  and states  $s$ ,  $\pi_0(s, a) > 0$ .

**Regression:** The simplest possible estimator is to directly use our learned MDP parameters to estimate our goal. This is called regression estimator. While training our MDP, we learn an estimator  $\hat{R}(s, a)$  that estimates  $R(s, a)$ . We can now directly estimate

$$\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s, a)}} [R(s, a)]$$

with

$$\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s, a)}} [\hat{R}(s, a)].$$

If  $\hat{R}(s, a) = R(s, a)$ , then this estimator is trivially correct. We will now consider alternative approaches and explore why you might use one estimator over another.

- (a) **Importance Sampling:** Let  $\hat{\pi}_0$  be an estimate of the true  $\pi_0$ . The *importance sampling estimator* uses that  $\hat{\pi}_0$  and has the form

$$\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s, a)}} \left[ \frac{\pi_1(s, a)}{\hat{\pi}_0(s, a)} R(s, a) \right].$$

We now show that if  $\hat{\pi}_0 = \pi_0$ , then the importance sampling estimator is equal to

$$\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s,a)}} [R(s, a)].$$

*Proof.* If  $\hat{\pi}_0 = \pi_0$ , then

$$\begin{aligned} \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \frac{\pi_1(s, a)}{\hat{\pi}_0(s, a)} R(s, a) \right] &= \sum_{(s,a)} \frac{\pi_1(s, a)}{\hat{\pi}_0(s, a)} R(s, a) p(s) \pi_0(s, a) \\ &= \sum_{(s,a)} \pi_1(s, a) R(s, a) p(s) \\ &= \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s,a)}} [R(s, a)], \end{aligned}$$

as desired.  $\square$

Note that this estimator only requires us to model  $\pi_0$  as we have the  $R(s, a)$  values in the observational data.

- (b) **Weighted Importance Sampling.** A variant of the importance sampling estimator is the *weighted importance sampling estimator*. The weighted importance sampling estimator has the form

$$\frac{\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} R(s, a) \right]}{\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} \right]}.$$

We now show that if  $\hat{\pi}_0 = \pi_0$ , then the weighted importance sampling estimator is equal to

$$\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s,a)}} [R(s, a)].$$

*Proof.* We already showed that if  $\hat{\pi}_0 = \pi_0$ , then

$$\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \frac{\pi_1(s, a)}{\hat{\pi}_0(s, a)} R(s, a) \right] = \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s,a)}} [R(s, a)].$$

For the denominator, if  $\hat{\pi}_0 = \pi_0$ , we have

$$\begin{aligned} \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \frac{\pi_1(s, a)}{\hat{\pi}_0(s, a)} \right] &= \sum_{(s,a)} \frac{\pi_1(s, a)}{\hat{\pi}_0(s, a)} p(s) \pi_0(s, a) \\ &= \sum_{(s,a)} \pi_1(s, a) p(s) \\ &= \sum_{(s,a)} p(s, a \mid \pi_1) \\ &= 1. \end{aligned}$$

Hence,

$$\frac{\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} R(s,a) \right]}{\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} \right]} = \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s,a)}} [R(s,a)]$$

when  $\hat{\pi}_0 = \pi_0$ , as desired.  $\square$

- (c) One issue with the weighted importance sampling estimator is that it can be biased in many finite sample situations. In finite samples, we replace the expected value with a sum over the seen values in our observational dataset. Please show that the weighted importance sampling estimator is biased in these situations.

**Hint:** Consider the case where there is only a single data element in the observational dataset.

*Proof.* When there is only a single data element  $(s^*, a^*, R(s^*, a^*))$  in the observational dataset, we have

$$\begin{aligned} \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} R(s,a) \right] &= \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} R(s^*, a^*), \\ \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} \right] &= \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)}. \end{aligned}$$

The weighted importance estimator then gives

$$\frac{\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} R(s,a) \right]}{\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} \right]} = R(s^*, a^*) = \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} [R(s,a)].$$

If  $\pi_1 \neq \pi_0$ , that is, if  $\pi_1$  has nonzero probability taking any action  $a \neq a^*$  at state  $s^*$ , then the weighted importance sampling estimator

$$\frac{\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} R(s,a) \right]}{\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} \right]} = \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} [R(s,a)] \neq \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s,a)}} [R(s,a)],$$

causing a bias.  $\square$

- (d) **Doubly Robust.** One final commonly used estimator is the doubly robust estimator. The doubly robust estimator has the form

$$\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s,a)}} \left[ \mathbb{E}_{a \sim \pi_1(s,a)} [\hat{R}(s,a)] + \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} (R(s,a) - \hat{R}(s,a)) \right].$$

One advantage of the doubly robust estimator is that it works if either  $\hat{\pi}_0 = \pi_0$  or  $\hat{R}(s, a) = R(s, a)$ .

- i. First we show that the doubly robust estimator is equal to  $\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s, a)}} [R(s, a)]$  when  $\hat{\pi}_0 = \pi_0$ .

*Proof.* First of all, we have

$$\mathbb{E}_{a \sim \pi_1(s, a)} [\hat{R}(s, a)] = \sum_a \pi_1(s, a) \hat{R}(s, a).$$

Note that this expression is independent of  $a$ . Hence,

$$\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s, a)}} \left[ \mathbb{E}_{a \sim \pi_1(s, a)} [\hat{R}(s, a)] \right] = \sum_s \sum_a \pi_1(s, a) \hat{R}(s, a) p(s) = \sum_{(s, a)} \pi_1(s, a) \hat{R}(s, a) p(s).$$

For the second term, when  $\hat{\pi}_0 = \pi_0$ , we have

$$\begin{aligned} \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s, a)}} \frac{\pi_1(s, a)}{\hat{\pi}_0(s, a)} (R(s, a) - \hat{R}(s, a)) &= \sum_{(s, a)} \frac{\pi_1(s, a)}{\hat{\pi}_0(s, a)} (R(s, a) - \hat{R}(s, a)) p(s) \pi_0(s, a) \\ &= \sum_{(s, a)} \pi_1(s, a) R(s, a) p(s) - \pi_1(s, a) \hat{R}(s, a) p(s) \\ &= \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s, a)}} [R(s, a)] - \sum_{(s, a)} \pi_1(s, a) \hat{R}(s, a) p(s). \end{aligned}$$

Therefore, the doubly robust estimator

$$\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s, a)}} \left[ \mathbb{E}_{a \sim \pi_1(s, a)} [\hat{R}(s, a)] + \frac{\pi_1(s, a)}{\hat{\pi}_0(s, a)} (R(s, a) - \hat{R}(s, a)) \right] = \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s, a)}} [R(s, a)].$$

when  $\hat{\pi}_0 = \pi_0$ . □

- ii. Now we show the the doubly robust estimator is equal to  $\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s, a)}} [R(s, a)]$  when  $\hat{R}(s, a) = R(s, a)$ .

*Proof.* As in the previous proof, we have

$$\mathbb{E}_{a \sim \pi_1(s, a)} [\hat{R}(s, a)] = \sum_a \pi_1(s, a) \hat{R}(s, a).$$

When  $\hat{R}(s, a) = R(s, a)$ ,

$$\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s, a)}} \left[ \mathbb{E}_{a \sim \pi_1(s, a)} [\hat{R}(s, a)] \right] = \sum_s \sum_a \pi_1(s, a) R(s, a) p(s) = \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s, a)}} [R(s, a)].$$

The second term clearly vanishes when  $\hat{R}(s, a) = R(s, a)$ . Hence, the doubly robust estimator

$$\mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_0(s, a)}} \left[ \mathbb{E}_{a \sim \pi_1(s, a)} \left[ \hat{R}(s, a) \right] + \frac{\pi_1(s, a)}{\hat{\pi}_0(s, a)} (R(s, a) - \hat{R}(s, a)) \right] = \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_1(s, a)}} [R(s, a)].$$

when  $\hat{R}(s, a) = R(s, a)$ . □

(e) In some situations, we can choose between the importance sampling estimator and the regression estimator. Which one is better in the following situations? In all of these situations, the states  $s$  consist of patients, actions  $a$  represent the drug to give to certain patients and  $R(s, a)$  represent the lifespan of the patient after receiving the drug.

i. Drugs are randomly assigned to patients, but the interaction between the drug, patient, and lifespan is very complicated.

In this case, importance sampling estimator is better, since it is easier to model an estimation  $\hat{\pi}_0$  of  $\pi_0$ .

ii. Drugs are assigned to patients in a very complicated manner, but the interaction between the drug, patient and lifespan is very simple.

In this case, the regression estimator is better, since it is easier to learn an estimator  $\hat{R}(s, a)$  that estimates  $R(s, a)$ . ■

### Problem 3 PCA

In class, we showed that PCA finds the "variance maximizing" directions onto which to project the data. In this problem, we find another interpretation of PCA.

Suppose set of points  $\{x^{(i)}\}_{i=1}^m$ . Let us assume that we have as usual preprocessed the data to have zero mean and unit variance in each coordinate. For a given unit-length vector  $u$ , let  $f_u(x)$  be the projection of point  $x$  onto the direction given by  $u$ . That is, if  $\mathcal{V} = \{\alpha u : \alpha \in \mathbb{R}\}$ , then

$$f_u(x) = \operatorname{argmin}_{v \in \mathcal{V}} \|x - v\|^2.$$

Show that the unit-length vector  $u$  that minimizes the mean squared error between the projected points and original points corresponds to the first principle component of the data. That is, show that

$$\operatorname{argmin}_{u: \|u\|=1} \sum_{i=1}^m \|x^{(i)} - f_u(x^{(i)})\|_2^2$$

gives the first principle component.

*Proof.* Let  $u$  be a vector with unit length. When  $\|x - v\|^2$  attains its minimum value, we have  $v = (x^T u)u$ . Therefore

$$f_u(x) = \operatorname{argmin}_{v \in \mathcal{V}} \|x - v\|^2 = (x^T u)u.$$

For each  $x$ , we have

$$\|x - f_u(x)\|_2^2 = \|x - (x^T u)u\|_2^2 = x^T x - u^T x x^T u$$

It follows that

$$\sum_{i=1}^m \|x^{(i)} - f_u(x^{(i)})\|_2^2 = \sum_{i=1}^m (x^{(i)})^T x^{(i)} - u^T x^{(i)} (x^{(i)})^T u.$$

Hence,

$$\operatorname{argmin}_{\|u\|=1} \sum_{i=1}^m \|x^{(i)} - f_u(x^{(i)})\|_2^2 = \operatorname{argmax}_{\|u\|=1} u^T \left( \sum_{i=1}^m x^{(i)} (x^{(i)})^T \right) u.$$

This is exactly the same equation as in "variance maximizing", so this gives the first principal component. □

■