Problem Set 4, CS229(Machine Learning)

Ma Yubo

August, 14th, 2021

1 Neural Networks:MNIST image classification

Code Implementation is shown in *p01_nn.py*.
And the results are shown below:

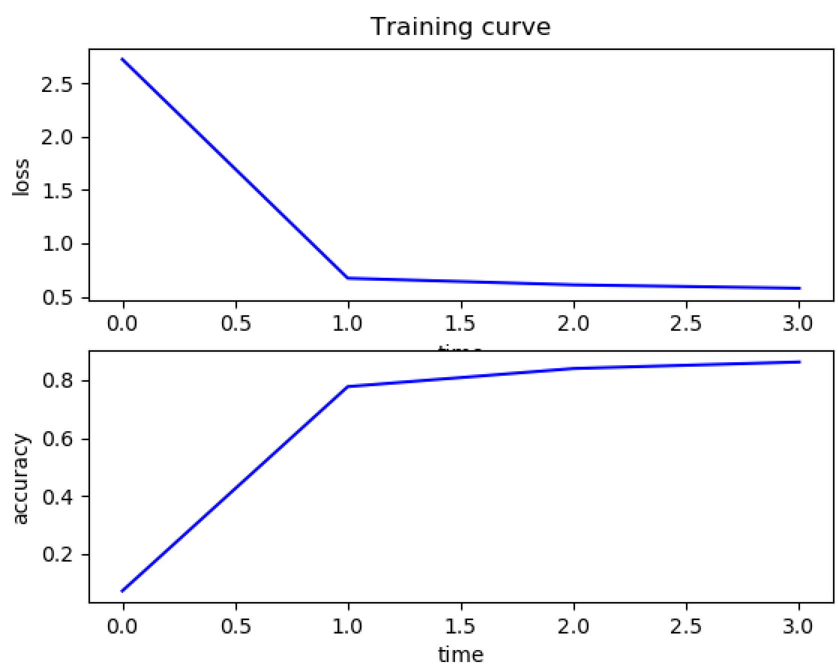


Figure 1.1: Training Loss(top) and Dev accuracy(bottom)

2 Off Policy Evaluation and Causal Inference

2.1 Importance Sampling

If $\hat{\pi}_0 = \pi_0$, then we have:

$$\begin{aligned}
&E_{s\sim p(s),a\sim\pi_0(s,a)} \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} R(s,a) \\
&= \sum_{(s,a)} p(s)\pi_0(s,a) \frac{\pi_1(s,a)}{\pi_0(s,a)} R(s,a) \\
&= \sum_{(s,a)} p(s)\pi_1(s,a) R(s,a) \\
&= E_{s\sim p(s),a\sim\pi_1(s,a)} R(s,a)
\end{aligned} \tag{2.1}$$

2.2 Weighted Importance Sampling

If $\hat{\pi}_0 = \pi_0$, then we have:

$$\begin{aligned}
&\frac{E_{s\sim p(s),a\sim\pi_0(s,a)} \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} R(s,a)}{E_{s\sim p(s),a\sim\pi_0(s,a)} \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)}} \\
&= \frac{E_{s\sim p(s),a\sim\pi_1(s,a)} R(s,a)}{\sum_{(s,a)} p(s)\pi_1(s,a)} \\
&= E_{s\sim p(s),a\sim\pi_1(s,a)} R(s,a)
\end{aligned} \tag{2.2}$$

2.3

Consider the case where there is only a single data element $(s^{(0)}, a^{(0)})$ in your observational dataset, then:

$$\begin{aligned}
&\frac{E_{s\sim p(s),a\sim\pi_0(s,a)} \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)} R(s,a)}{E_{s\sim p(s),a\sim\pi_0(s,a)} \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)}} \\
&= \frac{\frac{\pi_1(s^{(0)},a^{(0)})}{\hat{\pi}_0(s^{(0)},a^{(0)})} R(s^{(0)},a^{(0)})}{\frac{\pi_1(s^{(0)},a^{(0)})}{\hat{\pi}_0(s^{(0)},a^{(0)})}} \\
&= R(s^{(0)},a^{(0)})
\end{aligned} \tag{2.3}$$

Under this situation, the estimation of $R(s,a)$ is independent with the policy. Thus the weighted importance sampling estimator is biased.

2.4 Doubly Robust

We will deal with the term below first.

$$
\begin{aligned}
&E_{s\sim p(s),a\sim\pi_0(s,a)}(E_{a\sim\pi_1(s,a)}\hat{R}(s,a)) \\
&=\sum_s\sum_a p(s)\pi_0(s,a)\sum_{a'}\pi_1(s,a')\hat{R}(s,a') \\
&=\sum_s\sum_{a'} p(s)\pi_1(s,a')\hat{R}(s,a')\sum_{a}\pi_0(s,a) \\
&=\sum_s\sum_{a'} p(s)\pi_1(s,a')\hat{R}(s,a') \\
&=E_{s\sim p(s),a\sim\pi_1(s,a)}\hat{R}(s,a)
\end{aligned}
\tag{2.4}
$$

2.4.1

If $\hat{\pi}_0=\pi_0$, then we have:

$$
\begin{aligned}
&E_{s\sim p(s),a\sim\pi_0(s,a)}((E_{a\sim\pi_1(s,a)}\hat{R}(s,a))+\frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)}(R(s,a)-\hat{R}(s,a))) \\
&=E_{s\sim p(s),a\sim\pi_1(s,a)}\hat{R}(s,a)+E_{s\sim p(s),a\sim\pi_1(s,a)}(R(s,a)-\hat{R}(s,a)) \\
&=E_{s\sim p(s),a\sim\pi_1(s,a)}R(s,a)
\end{aligned}
\tag{2.5}
$$

2.4.2

If $\hat{R}(s,a)=R(s,a)$, then we have the answer directly:

$$
\begin{aligned}
&E_{s\sim p(s),a\sim\pi_0(s,a)}((E_{a\sim\pi_1(s,a)}\hat{R}(s,a))+\frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)}(R(s,a)-\hat{R}(s,a))) \\
&=E_{s\sim p(s),a\sim\pi_0(s,a)}((E_{a\sim\pi_1(s,a)}R(s,a)) \\
&=E_{s\sim p(s),a\sim\pi_1(s,a)}R(s,a)
\end{aligned}
\tag{2.6}
$$

2.5

2.5.1

Drugs are randomly assigned to patients, but the interaction between the drug, patient and lifespan is very complicated: $\pi(s,a)$ is easy to get while $R(s,a)$ is hard to get. So we use regression estimator to learn an estimate $\hat{R}(s,a)$ about $R(s,a)$.

2.5.2

Drugs are assigned to patients in a very complicated manner, but the interaction between the drug, patient and lifespan is very simple: $\pi(s,a)$ is hard to get while $R(s,a)$ is easy to get. So we use importance sampling estimator to learn an estimate $\hat{pi}(s,a)$ about $\pi(s,a)$.

3 Principle Components Analysis

From the definition of $f_\mu(x)$, we esaily derive that $f_\mu(x) = \mu^T x$, where μ is a unit length vector representing the projection (hyper)plane.
Then we have:

$$\begin{aligned}
& argmin_{\mu:\mu^T\mu=1} \sum_{i=1}^{n} ||x^{(i)} - f_\mu(x^{(i)})||_2^2 \\
&= argmin_{\mu:\mu^T\mu=1} \sum_{i=1}^{n} (x^{(i)} - \mu^T x^{(i)})^T (x^{(i)} - \mu^T x^{(i)}) \\
&= argmin_{\mu:\mu^T\mu=1} \sum_{i=1}^{n} ({x^{(i)}}^T x^{(i)} - \mu^T x^{(i)} {x^{(i)}}^T \mu) \\
&= argmax_{\mu:\mu^T\mu=1} \sum_{i=1}^{n} (\mu^T x^{(i)} {x^{(i)}}^T \mu)
\end{aligned} \tag{3.1}$$

which gives the first principal component.

4 Independent Components Analysis

4.1 Gaussian source

Assume sources are distributed according to a standard normal distribution, i.e $s \sim N(0,1)$, then we have

$$\begin{aligned} l(W) &= \sum_{i=1}^{n}(log|W| + \sum_{j=1}^{d} logg'(w_j^T x^{(i)})) \\ &= \sum_{i=1}^{n}(log|W| - \sum_{j=1}^{d} \frac{(w_j^T x^{(i)})^2}{2} + C \end{aligned} \tag{4.1}$$

Take derivation on W,

$$\begin{aligned} \frac{\partial l}{\partial W} &= \sum_{i=1}^{n}((W^{-1})^T + Wx^{(i)}{x^{(i)}}^T \\ &= n(W^{-1})^T - WX^TX \\ &= 0 \end{aligned} \tag{4.2}$$

$$\Rightarrow WW^T = \frac{1}{n}(X^TX)^{-1} \tag{4.3}$$

Now, let R be an arbitrary orthogonal matrix and $W' = WR$. Then we have:

$$W'{W'}^T = WRR^TW^T = WW^T \tag{4.4}$$

So there exists ambiguity in unmixing matrix caused by the **Rotation invariance** of Gaussian Distribution

4.2 Laplace Source

Assume sources are distributed according to a standard Laplace distribution, i.e $f(s) = \frac{1}{2}exp(|s|)$.

$$\begin{aligned} l(W) &= \sum_{i=1}^{n}(log|W| + \sum_{j=1}^{d} logf(w_j^T x^{(i)})) \\ &= \sum_{i=1}^{n}(log|W| - \sum_{j=1}^{d} |w_j^T x^{(i)}| + C \end{aligned} \tag{4.5}$$

Take derivation on W,

$$\frac{\partial l}{\partial W} = \sum_{i=1}^{n}((W^{-1})^T - sgn(Wx^{(i)}){x^{(i)}}^T) \tag{4.6}$$

For each sample $x^{(i)}$, therefore, the update rule is:

$$W := W + \alpha((W^{-1})^T - sgn(Wx^{(i)}){x^{(i)}}^T) \tag{4.7}$$

5 Markov Decision Process

5.1

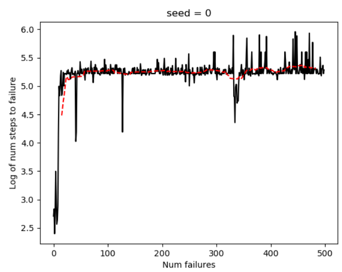
$$
\begin{aligned}
&||B(V_1) - B(V_2)||_\infty \\
&= \gamma \max_{s \in S} | \max_{a_1 \in A} \sum_{s' \in S} P_{s,a_1}(s') V_1(s') - \max_{a_2 \in A} \sum_{s' \in S} P_{s,a_2}(s') V_2(s')| \\
&\leq \gamma \max_{s \in S} \max_{a \in A} | \sum_{s' \in S} P_{s,a}(s')(V_1(s') - V_2(s'))| \\
&\leq \gamma \max_{s \in S} \max_{s' \in S} |(V_1(s') - V_2(s')| \\
&= \gamma \max_{s' \in S} |(V_1(s') - V_2(s')| \\
&= \gamma ||V_1 - V_2||_\infty
\end{aligned}
\tag{5.1}
$$

Note we use the property of p.d.f $\sum_{s' \in S} P_{s,a}(s') = 1$ at the second inequality.

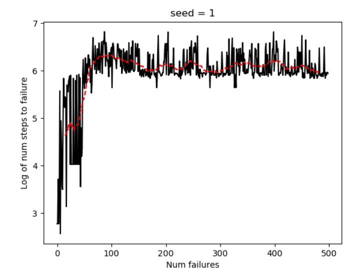
5.2

Suppose We have two fixed points V_1 and V_2 satisfying $B(V) = V$.
Then we have $||B(V_1) - B(V_2)||_\infty = ||V_1 - V_2||_\infty$, which contradicts the γ-contraction max-norm property of Bellman Operator.

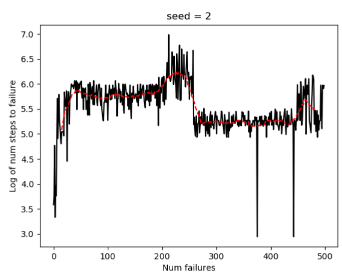
6 Reinforcement Learning: The inverted pendulum



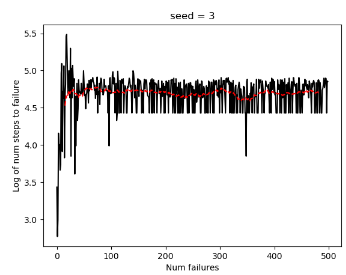
(a) Trial 0



(b) Trial 1



(c) Trial 2



(d) Trial 3

Figure 6.1: Scatter about failure nums and log-time before failure

We can observe from figures above that the algorithm tends to converge at about iteration 50-100.

Also, random experiments show diverse results on each trial, which shows the unrobustness of this algorithm.