Лекция 5

Обучение с подкреплением

Никита Юдин, iudin.ne@phystech.edu

Московский физико-технический институт
Физтех-школа прикладной математики и информатики

6 марта 2024

ФФПМИ

Что делали?

Новая постановка задачи

Теперь в задаче удалим ограничение на множество состояний среды $\cancel{|S| \ll \infty}$.

Заменяем таблицу на сеть

Если s, a — входные данные, $y \in \mathbb{R}$ — наблюдаемые значения, которые будут получаться как несмещ. оценка решения уравнения Беллмана. Будем решать следующую задачу регрессии:

$$y(s,a) := r(s,a) + \gamma \max_{a'} Q^*(s', a', \theta_k)$$

Где $s' \sim p(s'|s,a)$.

Что делали?

Deep Q-learning

Инициализируем $Q^*(s, a, \theta)$ произвольно, $\theta^- := \theta, \mathcal{D} = \varnothing$; Получаем s_0 : **for** $k = 0, 1, 2, \ldots$

1. Выбрать действие $a_k \sim \varepsilon - \text{greedy}(Q^*(s_k, a, \theta))$;
2. Пронаблюдать $r_k, s_{k+1}, \text{done}_{k+1}$ и сохранить $(s_k, a_k, r_k, s_{k+1}, \text{done}_{k+1})$ в $\mathcal{D}$;
3. Засемплировать батч переходов $\mathbb{T} := (s, a, r, s', \text{done})$ из $\mathcal{D}$;
4. $y(\mathbb{T}) := r + \gamma(1 - \text{done}) \max\limits_{a'} Q^*(s', a', \theta^-)$;
5. Совершить шаг градиентного спуска:

$$\theta \leftarrow \theta - \frac{\alpha}{B} \sum_{\mathbb{T}} \nabla_\theta (Q^*(s, a, \theta) - y(\mathbb{T}))^2$$

6. Обновляем *Target Net*, если $k \mod K = 0 : \theta^- \leftarrow \theta$.

Что делали?

Проблема *vanilla DQN*

Выученная функция склонна к переоцениванию будущей награды, то есть Q-функция начинает неограниченно расти. Эта проблема называется *overestimation bias*.

Что делали?

Name	Networks	Targets
Double* Q-learning	Q_1 Q_2	$y_1 = r + \gamma Q_2(s', \arg\max_{a'} Q_1(s', a'))$ $y_2 = r + \gamma Q_1(s', \arg\max_{a'} Q_2(s', a'))$
Double Q-learning	Q Q_- — TN	$y = r + \gamma Q_-(s', \arg\max_{a'} Q(s', a'))$
Twin Q-learning	Q_1 Q_2	$y_1 = r + \gamma \min_{i=1,2} Q_i(s', \arg\max_{a'} Q_1(s', a'))$ $y_2 = r + \gamma \min_{i=1,2} Q_i(s', \arg\max_{a'} Q_2(s', a'))$

Мотивация

Q-функция по политике π

$$Q^{\pi}(s,a) = \mathbb{E}_{\tau}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a] = \mathbb{E}_{\tau} Z(s,a),$$

где $\tau \sim p(s_0) \cdot \prod_{t=0}^{\infty} (\pi(a_t|s_t) p(s_{t+1}|s_t, a_t))$. До этого момента вопрос о распределении Z нас несильно интересовал — мы искали сразу её матожидание.

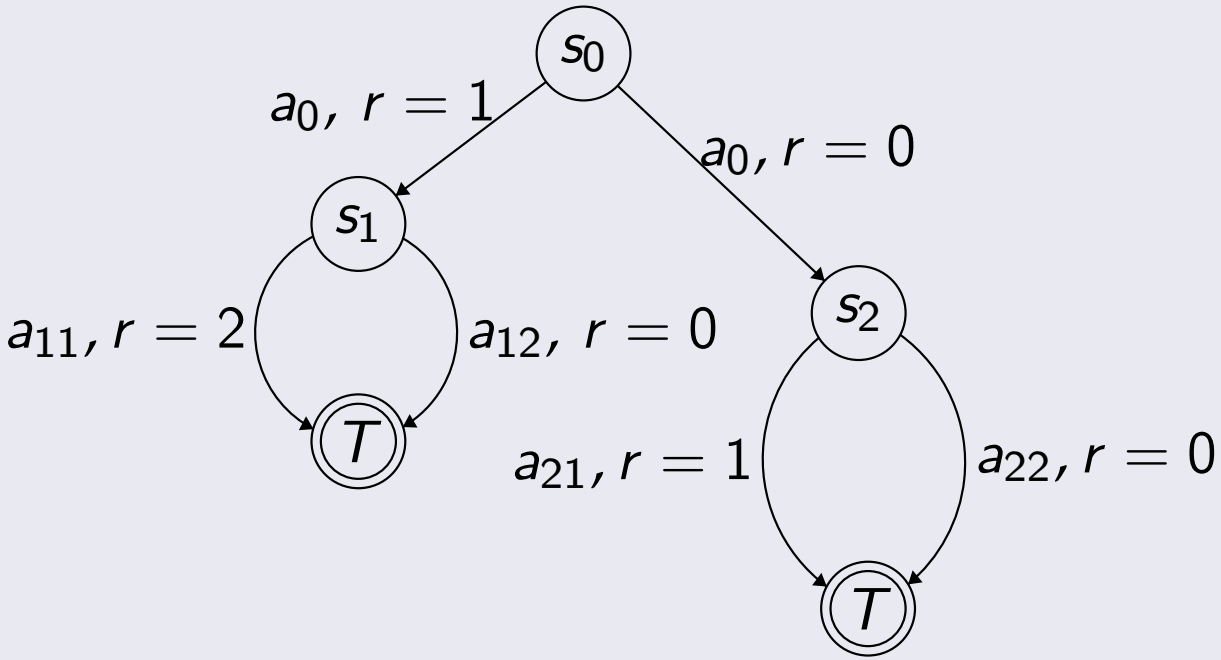
Мотивация

Пример

Посмотрим как выглядит Z на реальном примере. Положим $\gamma = 0.5$. Пусть мы находимся в состоянии s_0, из которого мы совершаем действие a_0, приводящее нас в s_1 с вероятностью $0,6$ и наградой 1 либо в состояние s_2 с вероятностью $0,4$ и наградой 0. Далее: из состояния s_1 есть два действия a_{11} с наградой 2 и a_{12} с наградой 0, приводящие нас в терминальное состояние; из состояния s_2 так же есть два действия a_{21} с наградой 1 и a_{22} с наградой 0. Текущая политика такова, что $\pi(a_{11}|s_1) = \pi(a_{12}|s_1) = 1/2$ и $\pi(a_{21}|s_2) = 1/4, \pi(a_{22}|s_2) = 3/4$.
Посчитаем $Z(s_0, a_0)$.

Мотивация

Пример



Мотивация

Пример

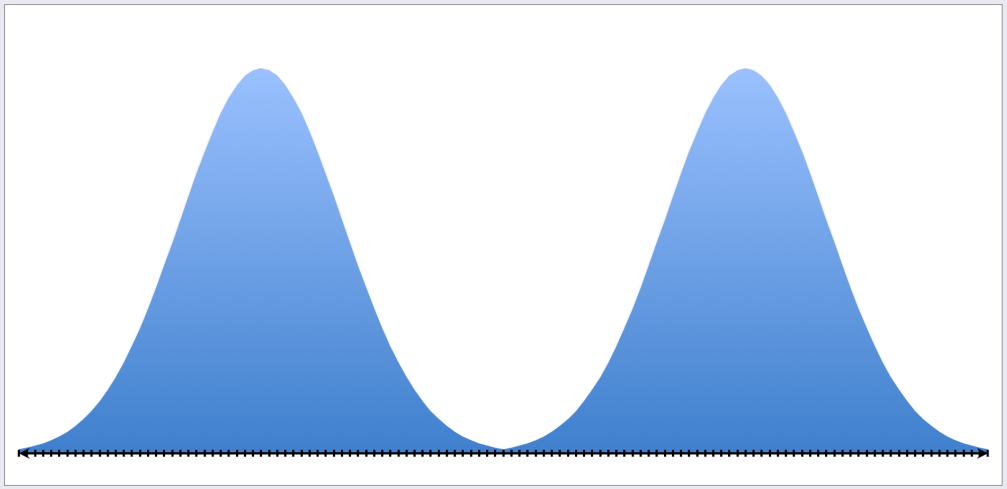
Итого $Z^{\pi}(s_0, a_0)$ — дискретная случайная величина с функцией распределения согласно таблице:

s'	a'	Вероятность	reward-to-go
s_1	a_{11}	0.3	$1 + 2\gamma$
s_1	a_{12}	0.3	1
s_2	a_{21}	0.1	γ
s_2	a_{22}	0.3	0

Мотивация

Почему иногда надо искать именно распределение?

Оказывается, поиск матожидания Z — не всегда оптимальное решение:



В данном случае среднее случайной величины будет в районе нуля!

QR-DQN

Идея

Почему бы нам не искать распределение Z (т.е. распределение наград) и понимать о задаче больше?

Иными словами заставим нейронную сеть аппроксимировать функцию Z и будем выбирать оптимальное действие согласно

$a^* = \arg\max_a \mathbb{E}Z(s, a)$

Вопрос к залу

Каким образом заставить нейросеть возвращать распределение?

QR-DQN

QR-DQN

Зафиксируем N и будем генерировать N дельта-функций. Их позиции будет выбирать нейронная сеть, каждая дельта-функция принимает максимальное значение $1/N$.

Следующий вопрос

Каким образом учить такую сеть?

Distributional Bellman Operator

Distributional Bellman Equation

$$Z^{\pi}(s,a) \overset{\text{c.d.f.}}{=} r(s,a) + \gamma Z^{\pi}(S',A'),$$

где $S' \sim p(\cdot|s,a); A' \sim \pi(\cdot|s)$ и равенство подразумевается по распределению.

Аналогично тому, как мы интерпретировали Q-функцию в виде таблицы, $Z(s,a)$ тоже может рассматриваться как таблица, в (s,a) ячейке которой находится случайная величина, а $Z(S',A')$ — это таблица случайных величин, к ячейке (S',A') которой мы обращаемся с помощью случайных величин (S',A').

Distributional Bellman Operator

Пояснение

Слева и справа записаны два процесса генерации одной и той же случайной величины. Мы можем бросить кость $Z^\pi(s,a)$ (случайная величина слева), а можем — сначала s', потом a', затем $Z^\pi(s',a')$ и выдать исход $r(s,a)+\gamma Z^\pi(s',a')$ (случайная величина справа), и эти две процедуры порождения эквивалентны.

Замечание

Подобные уравнения называются *recursive distributional equations* и рассматриваются математикой в одном из разделов теории вероятности.

Distributional Bellman Operator

Определение

Пусть $\mathcal{W}$ — метрика в пространстве $P(\mathbb{R})$ Тогда её *максимальной формой (maximal form)* будем называть следующую метрику в пространстве Z-функций:

$$\mathcal{W}^{\max}(Z_1, Z_2) := \sup_{s \in S, a \in A} \mathcal{W}(Z_1(s,a), Z_2(s,a))$$

Утверждение (без доказательства)

Distributional Bellman Operator — сжимающий оператор в пространстве с метрикой $d(Z_1, Z_2) := \max_{s,a} W_p(Z_1(s,a), Z_2(s,a))$, где W_p — расстояние Вассерштейна.

Distributional Bellman Operator

Замечание 1

Использовать KL-дивергенцию нельзя, потому что она не является метрикой, и с ней DBO не ялвяется сжимающим оператором.

Замечание 2

Если в алгоритме *vanilla DQN* с буфером заменить Q на Z, а *MSELoss* на $\max\limits_{s,a} W_p(Z_1(s,a), Z_2(s,a))$, то алгоритм потеряет свойство сходимости поскольку в таком случае не будет выполняться условие на сходимость алгоритма оптимизации *SGD*: $\mathbb{E}\nabla \text{loss} \neq \nabla\mathbb{E}\text{loss}$

Wasserstein metric

Расстояние Вассерштейна: общий случай

$$W_p(u, v) = (\inf_{\mu \in \Gamma(u,v)} \mathbb{E}_{(x,y)\sim\mu} \|x - y\|_p^p)^{1/p},$$

где $\Gamma(U, V)$ — множество всех распределений, маргиналы которых имеют распределения U и V.

Расстояние Вассерштейна: для одномерных случайных величин

$$W_p(U, V) = (\int_0^1 |F_U^{-1}(w) - F_V^{-1}(w)|^p dw)^{1/p}$$

Wasserstein metric

Свойства

- $W_p(aU, aV) \leqslant |a| W_p(U, V)$
- $W_p(A + U, A + V) \leqslant W_p(U, V)$
- $W_p(AU, AV) \leqslant \|A\|_p W_p(U, V)$

Wasserstein metric

Песок

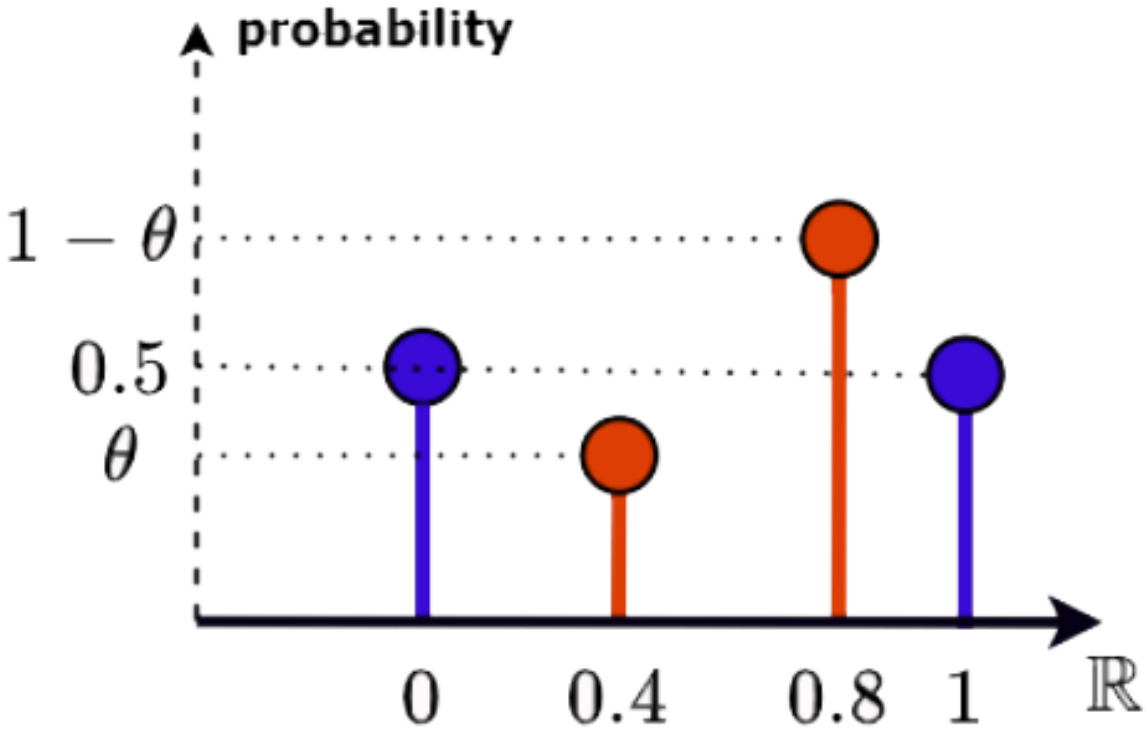
Расстояние Вассерштейна между двумя распределениями неспроста имеет второе название *Earth Moving Distance.* Аналогия такая: нам даны две кучи песка. Объём песка в кучах одинаков, но у них разные конфигурации, они «насыпаны» по-разному. Чтобы перенести каждую песчинку массы m на расстояние x, нам нужно затратить «работы» объёмом mx. Расстояние Вассерштейна замеряет, какое минимальное количество работы нужно совершить, чтобы перевести конфигурацию первой кучи песка во вторую кучу; объём песка в каждой кучи одинаков. Для дискретных распределений, когда функции распределения (и, соответственно, квантильные функции) — «ступеньки», минимальная работа полностью соответствует площади между функциями распределений.

Wasserstein metric

Пример

Посчитаем расстояние Вассерштейна между двумя следующими распределениями. Первое распределение — честная монетка с исходами 0 и 1 (красным), вторая случайная величина принимает значение 0.4 с вероятностью $\theta < 0.5$ и 0.8 с вероятностью $1 - \theta$ (синим). Можно нарисовать функции распределения и посчитать площадь между ними. А можно рассуждать так: давайте «превратим» вторую кучу песка в первую:

Wasserstein metric



Wasserstein metric

Пример

Посмотрим на песок объёма θ в точке 0.4. Куда его переносить? Наверное, в точку 0 куда его тащить ближе. Перенесли; совершили работы объёмом 0.4θ. Посмотрим на песок объёма $1-\theta$ в точке 0.8. Его удобно тащить в точку 1, но там для получения первой конфигурации нужно только 0.5 песка. Поэтому 0.5 песка из точки 0.8 мы можем перевести в точку 1, совершив работу $0.2 \cdot 0.5$, а оставшийся объём $1-\theta-0.5$ придётся переводить в точку 0, совершая работу $0.8(0.5-\theta)$. Итого расстояние Вассерштейна равно: $0.4\theta + 0.8(0.5-\theta) + 0.1$

Distributional Optimality Operator

Distributional Optimality Equation

$$Z^*(s,a) = r(s,a) + \gamma Z^*(S',A'),$$

где $S' \sim p(\cdot|s,a)$; $A' \sim \arg\max_a \mathbb{E}\, Z^\pi(S',a)$ ($\arg\max$ может достигаться в разных точках, поэтому используется знак $\sim$) и равенство подразумевается по распределению.

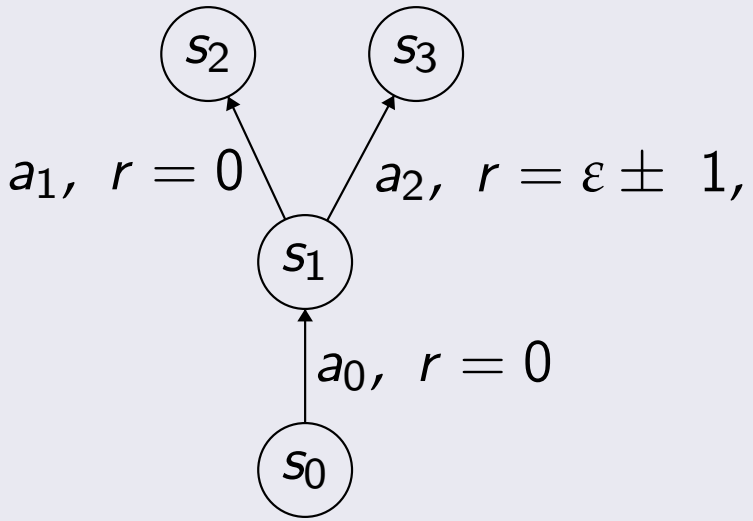
Аналогично тому, как мы интерпретировали Q-функцию в виде таблицы, $Z(s,a)$ тоже может рассматриваться как таблица, в (s,a) ячейке которой находится случайная величина, а $Z(A',S')$ — это таблица случайных величин, к ячейке (S',A') которой мы обращаемся с помощью случайных величин (S',A').

Distributional Optimality Operator

Замечание

DOO может не являться сжимающим оператором в пространстве с метрикой Вассерштейна.

Доказательство



Доказательство

Доказательство

Рассмотрим среду с диаграммы, где под $r = \varepsilon \pm 1$ подразумевается равновероятная награда $\varepsilon + 1$ и $\varepsilon - 1$, $\varepsilon > 0, \varepsilon \ll 1$. Оптимальная политика состоит в выборе a_2 из s_1, так как матожидание награды максимальна. Положим $\gamma = 1$.

Доказательство

Доказательство

Рассмотрим среду с диаграммы, где под $r = \varepsilon \pm 1$ подразумевается равновероятная награда $\varepsilon + 1$ и $\varepsilon - 1$, $\varepsilon > 0, \varepsilon \ll 1$. Оптимальная политика состоит в выборе a_2 из s_1, так как матожидание награды максимальна. Положим $\gamma = 1$.

	(s_0, a_0)	(s_1, a_1)	(s_1, a_2)
Z^*	$\varepsilon \pm 1$	0	$\varepsilon \pm 1$
Z_0^{our}	$\varepsilon \pm 1$	0	$-\varepsilon \pm 1$

Где Z^* — оптимальное, а Z_0^{our} — начальное приближение на которое мы будем действовать *DOO*. Тогда:

$$d(Z_0^{\text{our}}, Z^*) = \max_{(s,a)} W_1(Z_0^{\text{our}}(s,a), Z^*(s,a)) = 2\varepsilon$$

Доказательство

Доказательство

	(s_0, a_0)	(s_1, a_1)	(s_1, a_2)
Z^*	$\varepsilon \pm 1$	0	$\varepsilon \pm 1$
Z_0^{our}	$\varepsilon \pm 1$	0	$-\varepsilon \pm 1$
$W_1(Z_0^{\text{our}}(s,a), Z^*(s,a))$	0	0	2ε

Теперь применим к Z_0^{our} оператор Беллмана и получим $Z_1^{\text{our}} := T(Z_0^{\text{our}})$

	(s_0, a_0)	(s_1, a_1)	(s_1, a_2)
Z^*	$\varepsilon \pm 1$	0	$\varepsilon \pm 1$
Z_0^{our}	$\varepsilon \pm 1$	0	$-\varepsilon \pm 1$
$T(Z_0^{\text{our}})$	0	0	$\varepsilon \pm 1$

Доказательство

Доказательство

	(s_0, a_0)	(s_1, a_1)	(s_1, a_2)
Z^*	$\varepsilon \pm 1$	0	$\varepsilon \pm 1$
Z_0^{our}	$\varepsilon \pm 1$	0	$-\varepsilon \pm 1$
$T(Z_0^{\text{our}})$	0	0	$\varepsilon \pm 1$
$W_1(Z^*, T(Z_0^{\text{our}}))$	1	0	2ε

$$d(T(Z_0^{\text{our}}), Z^*) = \max(1, 0, 2\varepsilon) = 1$$

Получили, что за одну итерацию расстояние увеличилось! Значит сжатия нет.

Хорошие новости

Утверждение

$$\|\mathbb{E}\, Z_1 - \mathbb{E}\, Z_2\|_\infty > \|\mathbb{E}\, T(Z_1) - \mathbb{E}\, T(Z_2)\|_\infty$$

Использование *DOO* (хоть и несжимающего) приведет нас к оптимальной политике, но гарантированного приближения к настоящему Z не будет, найдется что-то другое $\hat{Z}$, такое что $\mathbb{E}\,\hat{Z} = \mathbb{E}\, Z$.

Это происходит из-за того, что *DOE* имеет не единственное решение.

QR DQN

Так же как и в *DQN*: θ — параметры сети. Для одного семпла (s, a, r, s'):

1. $a^* = \arg\max_a \mathbb{E}\, Z_\theta(s', a)$
2. Считаем целевое значение $y = r + \gamma Z_\theta(s', a^*)$
3. Считаем потери $\text{Loss}(y, Z_\theta(s, a)) \to \min_\theta$ и сделаем один шаг стох. оптимизации (например *SGD*).

Quantile Regression DQN (QR-DQN)

Гиперпараметры

B — размер мини-батчей, A — число атомов, K — периодичность обновления таргет-сети, $\varepsilon(t)$ — стратегия исследования, $z_i(s, a, \theta)$ — нейросетка с параметрами θ, SGD-оптимизатор

Предварительные вычисления

Предпосчитать середины отрезков квантильной сетки $\tau_i := \frac{\frac{i}{A} + \frac{i+1}{A}}{2}$
Инициализировать θ произвольно
Положить $\theta^- := \theta$
Пронаблюдать s_0

Quantile Regression DQN (QR-DQN)

На очередном шаге t:

1) выбрать a_t случайно с вероятностью $\varepsilon(t)$, иначе $a_t := \arg\max_{a_t} \sum_{i=0}^{A-1} z_i(s_t, a_t, \theta)$
2) пронаблюдать r_t, s_{t+1}, done_{t+1}
3) добавить пятёрку $(s_t, a_t, r_t, s_{t+1}, \text{done}_{t+1})$ в реплей буфер
4) засэмплировать мини-батч размера B из буфера
5) для каждого перехода $\mathbb{T} := (s, a, r, s', \text{done})$ посчитать таргет:

$$y(\mathbb{T})_j := r + (1 - \text{done})\gamma z_j \left(s', \arg\max_{a'} \sum_i z_i(s', a', \theta^-), \theta^- \right)$$

6) посчитать:

$$\text{Loss}(\theta) := \frac{1}{BA} \sum_{\mathbb{T}} \sum_{i=0}^{A-1} \sum_{j=0}^{A-1} (\tau_i - \mathbb{I}[z_i(s, a, \theta) < y(\mathbb{T})_j]) (z_i(s, a, \theta) - y(\mathbb{T})_j)$$

Quantile Regression DQN (QR-DQN)

7) сделать шаг градиентного спуска по θ, используя $\nabla_\theta \text{Loss}(\theta)$
8) если $t \bmod K = 0$: $\theta^- \leftarrow \theta$

Implicit Quantile Networks

Идея: давайте будем уметь в нашей нейросети выдавать произвольные квантили, каким-то образом задавая $\tau \in (0, 1)$ дополнительно на вход. Тогда наша модель $z(s, a, \tau, \theta)$ будет неявно (implicit) задавать, вообще говоря, произвольное распределение на $\mathbb{R}$. По сути, мы моделируем квантильную функцию «целиком»; очень удобно:

$$F^{-1}_{Z_\theta(s,a)}(\tau) := z(s, a, \tau, \theta)$$

Implicit Quantile Networks

Утвреждение

Пусть F — функция распределения случайной величины X. Тогда, если $\tau \sim U[0,1]$, случайная величина $F^{-1}(\tau)$ имеет то же распределение, что и X.

Доказательство

Заметим, что функция распределения равномерной случайной величины при $x \in [0,1]$ равна $\mathrm{P}(\tau < x) = x$. Теперь посмотрим на функцию распределения случайной величины $F^{-1}(\tau)$:

$$\mathrm{P}\left(F^{-1}(\tau) < x\right) = \mathrm{P}\left(\tau < F(x)\right) = F(x)$$

Implicit Quantile Networks

Итак, мы можем аппроксимировать жадную стратегию примерно так:

$$\pi^*(s) := \arg\max_a \sum_{i=0}^{N} z(s, a, \tau_i, \theta), \qquad \tau_i \sim U[0, 1].$$

В качестве функции потерь предлагается использовать тот же квантильный лосс, что и в QR-DQN, но в модифицированном виде:

$$\text{Loss}(\mathbb{T}, \theta) := \sum_{i=0}^{N'} \frac{1}{N''} \sum_{j=0}^{N''} \text{Loss}_{\tau_i}(z(s, a, \tau_i, \theta), r + \gamma z(s', \pi^*(s'), \tau_j, \theta^-)),$$

где $\tau_i, \tau_j \sim U[0, 1]$ и

$$\text{Loss}_\tau(c, X) = (\tau - \mathbb{I}[c < X])\,(c - X)\,.$$