Отлично. Я проведу углублённое исследование по Модулю 1 (Пределы, последовательности и ряды), включая ссылки на конкретные научные и учебные источники (на русском и английском языках), а также предоставлю Jupyter-демонстрации для задач, связанных с сходимостью рядов и итерацией ценности в RL.

Я вернусь с подробным отчётом по этому модулю.

Модуль 1. Пределы, последовательности и ряды (в контексте обучения с подкреплением)

Введение

В этом учебном модуле рассматриваются фундаментальные математические понятия **предела, последовательности и ряда** и их связь с алгоритмами **обучения с подкреплением (Reinforcement Learning, RL)**. Несмотря на абстрактность, эти понятия играют ключевую роль в понимании сходимости алгоритмов RL и расчёте бесконечных сумм вознаграждений. Мы обсудим интуитивную и формальную сущность предела последовательности и сходимости бесконечного ряда, а затем покажем, как эти идеи проявляются в уравнениях Беллмана, итерации ценности и дисконтировании наград.

Модуль содержит:

- **Теоретический обзор** понятий предела, последовательности и ряда (интуитивно и строго).
- **Связь с RL:** как пределы и ряды возникают в уравнениях Беллмана, алгоритме итерации ценности и через фактор дисконтирования.
- **Примеры и аналогии:** простые задачи с решениями, помогающие усвоить материал.
- **Визуализации и Jupyter-демонстрации:** симуляции на Python, иллюстрирующие сходимость геометрического ряда и процесса итерации ценности.
- **Источники** (на русском и английском) для самостоятельного изучения.
- **Мини-глоссарий** ключевых терминов (с формулами в $\LaTeX$).

1. Теоретические основы: пределы, последовательности и ряды

1.1 Предел последовательности

Интуиция: Последовательность – это набор элементов (чисел) $x_1, x_2, x_3, \dots$ определённым образом. Говорят, что последовательность *имеет предел* (или *сходится* к значению a), если её члены постепенно приближаются к a. Например, последовательность $1,;\frac{1}{2},;\frac{1}{3},;\frac{1}{4},\dots$ приближается к 0 – её предел равен 0. Интуитивно, начиная с некоторого места все элементы последовательности оказываются *сколь угодно близко* к a.

Формально: Число a называется **пределом** последовательности x_n, если для любого наперёд заданного $\varepsilon > 0$ найдётся такой номер N, что для всех номеров $n > N$ расстояние между x_n и a меньше ε. В символической форме это записывают как:

$$\lim_{n\to\infty} x_n = a \iff \forall\varepsilon > 0\ \exists N(\varepsilon)\colon \forall n \geq N,\ |x_n - a| < \varepsilon\,,$$

что означает "для каждого ε найдётся порог N, после которого все элементы x_n лежат в интервале $(a-\varepsilon,;a+\varepsilon)$". Если такой a существует, последовательность называют **сходящейся** к a. В противном случае последовательность называется **расходящейся**. Расходящейся может быть, например, последовательность, уходящая в бесконечность ($1, 2, 3, \dots$ не имеет конечного предела) или колеблющаяся ($(-1)^n$ переключается между 1 и -1 и не приближается к единому значению).

Примеры:

- $x_n = \dfrac{1}{n}$ – сходится к 0 (при больших n члены все ближе к 0).
- $x_n = 1 - \dfrac{1}{n}$ – сходится к 1 (значения подползают к 1 сверху: $0,,0.5,,0.67,,0.75,\dots$).
- $x_n = (-1)^n$ – не имеет предела (значения прыгают $1,,-1,,+1,,-1,\dots$).

Если x_n становится произвольно большим (по модулю), говорят, что *предел равен бесконечности* (записывают $\lim x_n = +\infty$ или $-\infty$). Мы не будем здесь детально рассматривать бесконечные пределы, хотя отметим: для $\lim_{n\to\infty} x_n = \infty$ по определению требуется, чтобы для любого сколь угодно большого порога E начиная с некоторого места $|x_n| > E$.

1.2 Бесконечные ряды и их сходимость

Последовательности и ряды: Бесконечный **ряд** – это сумма бесконечного числа слагаемых, обычно записывается как $a_1 + a_2 + a_3 + \dots$ или $\sum_{n=1}^{\infty} a_n$. Ряд задаётся своей последовательностью слагаемых (a_n), и мы изучаем последовательность **частичных сумм**:

$$S_n = a_1 + a_2 + \cdots + a_n\,,$$

где S_n – сумма первых n членов. Ряд $\sum_{n=1}^{\infty} a_n$ называют **сходящимся**, если последовательность (S_n) частичных сумм имеет конечный предел. Иначе ряд называется **расходящимся**. Если существует $\lim_{n\to\infty} S_n = S$ (конечное число), то S называется **суммой ряда**.

Другими словами, бесконечная сумма имеет смысл (равна S) в том и только том случае, когда можно суммировать всё больше и больше членов, и итог приближается к фиксированному числу S. Например:

- **Геометрический ряд:** $1 + r + r^2 + r^3 + \dots$ – классический пример. Его частичная сумма $S_n = 1 + r + r^2 + \dots + r^{n-1} = \frac{1-r^n}{1-r}$ (если $r \neq 1$). Если $|r| < 1$, то $r^n \to 0$ при $n \to \infty$, и $\lim_{n\to\infty} S_n = \frac{1}{1-r}$ – **ряд сходится** (сумма равна $1/(1-r)$). Если же $|r| \geq 1$, частичные суммы не стабилизируются: при $|r| > 1$ члены ряда не стремятся к 0 и сумма убегает в бесконечность, а при $r = \pm 1$ сумма будет n или $0, 1, 0, 1, \dots$ – предел не существует. Таким образом, геометрический ряд сходится $\iff |r| < 1$. Например: $0.5 + 0.25 + 0.125 + \dots = 1$ (знаменитый парадокс Зенона: бесконечно деля отрезок, суммарная длина частей конечна), а вот $1 + 1 + 1 + \dots$ расходится (стремится к бесконечности), как и $1 - 1 + 1 - 1 + \dots$ (нет единого предела, последовательность S_n чередуется между 1 и 0).
- **Ряд $1 + \frac{1}{2} + \frac{1}{3} + \dots$ (гармонический):** его частичные суммы растут без ограничений (хотя и очень медленно). Формально, $S_n \approx \ln n + \gamma$ (где γ – постоянная Эйлера), поэтому $\lim S_n = +\infty$. Гармонический ряд расходится.

Для сходимости ряда **необходимо** (но недостаточно), чтобы $a_n \to 0$. Если члены не стремятся к 0, сумма не может «успокоиться» на каком-то значении. Сходящиеся ряды можно классифицировать по типам сходимости – например, **абсолютно сходящийся ряд** – это ряд, сходящийся даже после замены всех членов на их модуль $\sum |a_n|$. Абсолютная сходимость сильнее обычной сходимости, но в рамках этого модуля нам важно лишь общее понимание: **в RL, ряды вознаграждений обычно сходятся *абсолютно***, потому что дисконтирование делает вклад каждого шага ограниченным. Мы вскоре увидим, что *дискаунт-фактор* γ по сути играет роль r в геометрическом ряде, обеспечивая сходимость.

1.3 Пределы и ряды в контексте обучения с подкреплением

Обучение с подкреплением (RL) – это область машинного обучения, где агент учится действовать в среде, получая вознаграждения за свои действия. Математически среду часто моделируют как **марковский процесс принятия решений (MDP)**. В MDP будущее вознаграждение суммируется за все шаги в будущем, и обычно используется *дисконтирование* – умножение вознаграждений на коэффициент γ^t, где t – шаг во времени, а $0 \leq \gamma < 1$. Возникает **бесконечная сумма вознаграждений**:

$$G_t \;=\; R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \;=\; \sum_{k=0}^{\infty} \gamma^k \, R_{t+k+1}\,,$$

называемая **дисконтированным возвратом** (discounted return). Это именно бесконечный ряд с общим множителем γ^k. Если $\gamma < 1$, такой ряд сходится, поскольку является геометрическим рядом по фактору γ (при условии, что сами награды R_t не растут быстрее, чем геометрическая прогрессия). Максимальный вклад дальних шагов ограничен суммой геометрической прогрессии. Например, если на каждом шаге агент получает вознаграждение 1 (стремится бесконечно долго получать "1" каждый шаг), то суммарный дисконтированный возврат равен $1+\gamma+\gamma^2+\cdots=\frac{1}{1-\gamma}$ – конечное число при $\gamma<1$. При $\gamma=0.9$ сумма будет 10, при $\gamma=0.99$ – около 100, и т.д. Однако если $\gamma=1$ (то есть *недисконтированный случай* на бесконечном горизонте), то $\sum_{t=0}^{\infty}1=\infty$ – сумма *расходится*. Таким образом, **дисконтирование с $\gamma<1$ "делает трюк", заставляя бесконечную сумму сходиться**. Именно поэтому во всех бесконечно длительных задачах RL берут $\gamma<1$ (обычно $0.9, 0.99$ и т.п.) – это гарантирует математическую корректность суммарного вознаграждения.

> Замечание: хотя можно рассматривать и другие критерии оптимальности (например, **среднее вознаграждение** на шаг, без дисконтирования), классический подход в RL – **дисконтированное бесконечное суммарное вознаграждение**. Дисконтирование также интерпретируют как "коэффициент нетерпения" агента или вероятность выживания: например, $\gamma=0.95$ можно трактовать как 95% шанс продолжить каждый следующий шаг, что эквивалентно среднему количеству шагов $\frac{1}{1-0.95}=20$ (т.н. *эффективный горизонт* взаимодействия).

Фиксированный предел суммы наград G_t позволяет ввести важнейшее понятие RL – **функцию ценности состояния**. При заданной стратегии (политике) π агента определим**функцию ценности** $V^\pi(s)$ как математическое ожидание суммарного дисконтированного возврата, начиная из состояния s и следуя политике π:

$$V^\pi(s) \;=\; \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty}\gamma^t\,R_{t+1}\,\Big|\,S_0=s\Big]\,.$$

Это и есть **уравнение Беллмана для заданной политики** (в экспоненциальной форме): оно связывает ценность состояния с непосредственным вознаграждением и ценностями последующих состояний. Раскрыв математическое ожидание (суммирование по всем возможным переходам из s), часто записывают рекуррентно:

$$V^\pi(s)=\sum_a \pi(a|s)\sum_{s'}P(s'|s,a)\,\big[\,R(s,a,s')+\gamma\,V^\pi(s')\big]\,.$$

Здесь $\pi(a|s)$ – вероятность выбора действия a в состоянии s, а $P(s'|s,a)$ и $R(s,a,s')$ – модель среды (вероятность перехода и вознаграждение). Такое уравнение утверждает: **ценность состояния = немедленное вознаграждение + дисконтированная ценность следующего состояния** (усреднённая по всем действиям и исходам). Фактически, V^π – **предел последовательности** оценок возврата на всё большем горизонте: если учесть только ближайшее вознаграждение (горизонт 1), оценка $=R_{t+1}$; если горизонт 2 –

$R_{t+1} + \gamma R_{t+2}$; для горизонта N – $\sum_{k=0}^{N-1} \gamma^k R_{t+k+1}$; при $N \to \infty$ (учитывая все будущие шаги) эта сумма стремится к $V^\pi(s)$ при $\gamma < 1$.

Если мы ищем **оптимальную стратегию** π^{*}, максимизирующую $V^\pi(s)$ для всех s, возникает **уравнение Беллмана для оптимума** (уравнение оптимальности Беллмана):

$$V^*(s) \;=\; \max_a \sum_{s'} P(s'|s,a) \left[\, R(s,a,s') + \gamma V^*(s') \right].$$

Это функциональное уравнение задаёт систему нелинейных уравнений относительно неизвестной функции $V^*(s)$. Существование и единственность решения гарантируются тем фактом, что оператор Беллмана является **сжимающим отображением** с коэффициентом γ (в пространстве непрерывных или хотя бы ограниченных функций). Интуитивно, применение оператора $\mathcal{T}[V](s) = \max_a \sum_{s'} P(s'|s,a)[R + \gamma V(s')]$ каждый раз "подмешивает" ещё один шаг будущего, влияя всё слабее за счёт γ. Теорема Банаха о неподвижной точке (из функционального анализа) здесь обеспечивает, что после многократного применения оператора мы сойдёмся к единственному неподвижному значению V^{*}. Практически это реализует алгоритм **итерации ценности**.

2. Связь с ключевыми концепциями и алгоритмами RL

Теперь свяжем описанные математические понятия с основными идеями RL:

2.1 Уравнения Беллмана и дисконтирование

Уравнения Беллмана – центральные равенства динамического программирования и обучения с подкреплением. Они выражают стоимость состояния через стоимости последующих состояний. Как мы ввели выше, **уравнение оценки (policy evaluation)**: $V^\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma V^\pi(S_{t+1}) | S_t = s]$ – прямо содержит **бесконечную сумму** (в определении V^π) и параметр γ. Без γ эта сумма часто расходится (бесконечный горизонт с ненулевым средним вознаграждением даёт бесконечный результат). Именно поэтому **дисконтирование** $(0 \le \gamma < 1)$ – неотъемлемая часть формулировки бесконечного горизонта RL: благодаря ему сумма $\sum_{t=0}^{\infty} \gamma^t R_{t+1}$ сходится и определяет конечное значение $V^\pi(s)$. На уровне алгоритмов значение γ влияет на "длину памяти" агента: если γ близко к 1, агент учитывает дальние будущие награды почти наравне с ближайшими (дальновидный, *планы на долгую перспективу*), если γ мало (например, 0.1), агент фокусируется на ближайших шагах (почти жадное поведение, *краткосрочная выгода*). В формуле уравнения Беллмана γ выступает как *коэффициент обесценивания* будущего.

Стоит подчеркнуть: **Беллмановское ожидание** конечного возврата – это именно **предел частичных сумм**:

- После 1 шага: $\mathbb{E}[R_{t+1}]$,
- После 2 шагов: $\mathbb{E}[R_{t+1} + \gamma R_{t+2}]$,
- ...
- После N шагов: $\mathbb{E}[\sum_{k=0}^{N-1} \gamma^k R_{t+k+1}]$,
- При $N \to \infty$: $\mathbb{E}[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}] = V^\pi(s)$.

Так как последовательность ожиданий частичных сумм монотонно растёт (при положительных вознаграждениях) и ограничена $\frac{R_{\max}}{1-\gamma}$, по теореме о монотонной сходимости она сходится к точной сумме (при $\gamma < 1$). Более того, даже при случайных наградах (не детерминированных) *среднее значение* предела равно пределу средних (в силу линейности ожидания). Поэтому нам позволено сначала суммировать, а потом брать $\mathbb{E}$. Всё это обосновывает корректность уравнения $V^\pi(s) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | S_0 = s]$.

Беллмановское оптимальное уравнение (для V^*) также неявно включает бесконечную сумму – только уже максимизируя по действиям на каждом шаге. Его решение даёт максимальное возможное ожидаемое вознаграждение из каждого состояния. Это решение – фиксированная точка оператора оптимального беллмановского обновления. С точки зрения пределов, $V^*(s)$ – это тоже предел последовательности приближений (см. далее про итерацию ценности). Уникальность решения гарантируется тем, что оператор является γ-сжимающим, и разница между текущей оценкой и оптимальной убывает не хуже геометрической прогрессии с множителем γ при итеративном применении. Величину $\frac{1}{1-\gamma}$ можно интерпретировать как **эффективный горизонт** проблем RL: примерно столько шагов в среднем "влияют" на оценку состояния (например, при $\gamma = 0.95$ – порядка 20 шагов, при $\gamma = 0.99$ – около 100 шагов существенного влияния).

2.2 Итерация ценности и сходимость алгоритмов RL

Итерация ценности (Value Iteration) – алгоритм динамического программирования, решающий уравнение Беллмана для V^* **итерационным процессом**. Мы начинаем с некоторой начальной оценки $V_0(s)$ (часто нулевой функции) и применяем оператор Беллмана многократно:

$$V_{k+1}(s) \;=\; \max_a \sum_{s'} P(s'|s,a) \left[R(s,a,s') + \gamma V_k(s') \right].$$

Каждый шаг обновления приближает нас к оптимальному значению. Этот процесс можно понимать как последовательность функций $V_0, V_1, V_2, \ldots$, которая *сходится* к функции V^*. Формально, как упоминалось, оператор является γ-сжимающим в максимум-норме:
$\|V_{k+1} - V^*\|_\infty \le \gamma \|V_k - V^*\|_\infty$,
т.е. на каждой итерации максимальная ошибка уменьшается не менее чем в γ раз. Отсюда следует **сходимость итерации ценности** к единственному фиксированному решению V^*. Другими словами, V^* – *предел последовательности* V_k при $k \to \infty$. На

практике итерацию ценности проводят до тех пор, пока изменения значений не станут достаточно малы (меньше заданного ε). Число итераций, нужное для заданной точности, оценивается исходя из геометрической убыли ошибок: через k итераций погрешность $\sim O(\gamma^k)$ от максимума, что опять-таки отражает природу геометрического ряда.

Пример (табличная среда): Рассмотрим простейшую MDP с двумя состояниями S_1 и S_2. Пусть из S_1 агент переходит в S_2 и получает награду $+2$, а из S_2 возвращается в S_1 без награды (0). То есть цикл: $S_1 \xrightarrow{+2} S_2 \xrightarrow{+0} S_1 \xrightarrow{+2} S_2 \dots$ бесконечно. Возьмём $\gamma = 0.9$. Интуитивно, из S_1 агент получает награду 2 сразу, потом через шаг ещё 2, ещё и т.д. – бесконечная цепочка вознаграждений $2, 0, 2, 0, \dots$ с затуханием 0.9^k. Суммарное оптимальное вознаграждение из S_1 должно быть конечным (ряд $;2 + 0.9^2 \cdot 2 + 0.9^4 \cdot 2 + \dots$ сходится). Решим это *вручную*: обозначим $V^*(S_1) = v_1$, $V^*(S_2) = v_2$. По уравнению Беллмана:

- $v_1 = 2 + \gamma v_2 = 2 + 0.9 v_2$,
- $v_2 = 0 + \gamma v_1 = 0.9 v_1$.

Подставляем второе в первое: $v_1 = 2 + 0.9 \cdot (0.9 v_1) = 2 + 0.81 v_1$. Переносим: $v_1 - 0.81 v_1 = 2$, то есть $0.19 v_1 = 2$, откуда $v_1 \approx 10.53$. Затем $v_2 = 0.9 v_1 \approx 9.47$. Таким образом, $V^*(S_1) \approx 10.53$, $V^*(S_2) \approx 9.47$. Проверим итерацией ценности: стартуем с $V_0(S_1) = V_0(S_2) = 0$. Далее выполняем обновления:

```
gamma = 0.9
V = [0.0, 0.0]  # V[0] = V(S1), V[1] = V(S2)
for i in range(1, 6):
    V_new = [0.0, 0.0]
    # Беллмановское обновление
    V_new[0] = 2 + gamma * V[1]  # S1: награда 2 + дисконтированная ценность S2
    V_new[1] = 0 + gamma * V[0]  # S2: награда 0 + дисконт. ценность S1
    V = V_new
    print(f"Итерация {i}: V1={V[0]:.3f}, V2={V[1]:.3f}")
```

Выход программы:

```
Итерация 1: V1=2.000, V2=0.000
Итерация 2: V1=2.000, V2=1.800
Итерация 3: V1=3.620, V2=1.800
Итерация 4: V1=3.620, V2=3.258
Итерация 5: V1=4.932, V2=3.258
...
```

Как видим, значения осциллируют, приближаясь к решению (~ 10.53 и 9.47). На практике за ~ 20 итераций они сойдутся очень близко к теоретическим.

Сходимость итерации ценности в простой MDP с двумя состояниями. На графике показано изменение оценок $V_1 = V_k(S_1)$ и $V_2 = V_k(S_2)$ по итерациям алгоритма (с шагом обновления, приведённым выше). Горизонтирные пунктирные линии отмечают пределы $V^*(S_1) \approx 10.53$ и $V^*(S_2) \approx 9.47$. Заметьте, как с увеличением числа итераций обе последовательности $V_k(S_1)$ и $V_k(S_2)$ сходятся к своим пределам, хотя и не монотонно (из-за чередующейся структуры цикла). Тем не менее, расстояние до предела убывает примерно по геометрическому закону с множителем $\gamma = 0.9$ каждый шаг. Это отражает контрактность оператора Беллмана и гарантирует сходимость алгоритма.*

В более сложных алгоритмах RL, таких как **итерация по политике, Q-обучение, метод временных разностей**, сходимость тоже анализируется через предельный процесс. Например, Q-обучение апдейтами $Q_{k+1}(s,a) = (1-\alpha)Q_k(s,a) + \alpha[R + \gamma \max_{a'} Q_k(s',a')]$ можно рассматривать как стохастический аппроксимационный процесс, сходящийся к $Q^{\backslash *}(s,a)$ при подходящих условиях (уменьшающемся шаге α). В основе лежит тот же контракционный оператор. Таким образом, **гарантия сходимости многих алгоритмов RL базируется на свойствах предела и сходимости соответствующей последовательности**. Без этих свойств, мы не могли бы обосновать, что алгоритм когда-либо стабилизируется на оптимальном решении.

2.3 Дисконтирование в RL и его влияние

Мы уже неоднократно упоминали **коэффициент дисконтирования** γ. С точки зрения рядов, γ – это знаменатель прогрессии, от которого зависит сходимость и сумма. Чем γ ближе к 1, тем медленнее ряд сходится к пределу (больше членов нужно учесть, эффективный горизонт больше). При $\gamma = 0$ всё будущие награды игнорируются вовсе (ряд вырождается в R_{t+1}, только немедленная награда). В алгоритмах RL выбор γ влияет на обучение: при большом γ целевая функция V^π учитывает долгосрочные последствия действий, но обучение может стать менее стабильным (так как дальние вознаграждения более неопределённы). При малом γ агент фокусируется на ближайших наградах, что упрощает задачу обучения, но может вести к не оптимальным в долгосрочной перспективе решениям.

С практической стороны, γ часто выбирают близким к 1 (например, 0.99) для задач, где агент должен учиться *стратегии* (в играх, навигации и т.д.), и меньше (0.8 или 0.9) для задач с более короткими эпизодами или когда хотят ускорить обучение и получить более “моеопическое” поведение. Иногда γ считают фиксированным параметром задачи (например, в финансовых приложениях он связан со ставкой дисконтирования денег). Главное – **при $\gamma < 1$ алгоритмы RL (динамического программирования и многие обучающие) имеют теоретическую сходимость к оптимуму**, тогда как случай $\gamma = 1$

требует специальных методов (например, средней награды) и выходит за рамки базового курса.

3. Примеры, аналогии и задачи

Чтобы закрепить материал, рассмотрим несколько **примеров и аналогий**, которые помогут связать математические формулы с интуицией.

3.1 Предел последовательности: пример

Пример 1: последовательность определена рекурсивно: $a_1 = 1$, а $a_{n+1} = \frac{1}{2}(a_n + 6)$ для $n \geq 1$. Найти $\lim_{n\to\infty} a_n$.

Решение (набросок): Это типичный пример, где можно угадать предел L из уравнения равновесия. Если $a_n \to L$, то и $a_{n+1} \to L$. Подставляя в рекурсию предел, получим: $L = \frac{1}{2}(L + 6)$. Решив: $2L = L + 6$, отсюда $L = 6$. Чтобы строго обосновать, надо проверить, что последовательность монотонно стремится к 6 (например, если $a_1 = 1$, то $a_2 = 3.5$, $a_3 = 4.75$, ... – растёт и ограничена сверху 6). Значит, $\lim a_n = 6$. Это и есть неподвижная точка итерационного процесса, сходящегося к ней.

Такого рода рассуждение аналогично поиску оптимального V^{*} как неподвижной точки операторов – только там пространство бесконечномерно (по числу состояний), но идея та же.

3.2 Сумма бесконечного ряда: пример

Пример 2: Вычислить сумму ряда $0.8 + 0.8^2 + 0.8^3 + \dots$.

Решение: Это геометрический ряд с первым членом $a_1 = 0.8$ и знаменателем $r = 0.8$. Используя формулу для суммы: $S = \dfrac{a_1}{1-r}$ при $|r| < 1$. Здесь $S = \frac{0.8}{1-0.8} = \frac{0.8}{0.2} = 4$. Можно убедиться численно, суммируя достаточное число членов. Например, Python-код:

```
s = 0.0
for n in range(1, 51):
    s += 0.8**n
print(s)
```

выведет 3.99999998 (очень близко к 4), уже при 50 членах. Таким образом, ряд сходится к 4. Это можно интерпретировать так: если агент с $\gamma = 0.8$ каждый шаг получает награду 0.8 (то есть 80% от единицы), то суммарное вознаграждение стремится к 4. Проверяя частичные суммы: $0.8; 1.44; 1.952; 2.3616; \dots$ – они всё ближе к 4, но никогда не превысят 4.

3.3 Дисконтирование бесконечных наград: пример из RL

Пример 3: Агент в каждой ступени эпизода получает вознаграждение $R = 1$ (за каждый шаг, пока не закончится эпизод). Как зависит суммарное вознаграждение от коэффициента дисконтирования γ?

Решение: Если эпизод действительно бесконечен (нет терминального состояния), суммарное вознаграждение $G_0 = 1 + \gamma \cdot 1 + \gamma^2 \cdot 1 + \ldots$. Это опять геометрический ряд, сумма которого $G_0 = \frac{1}{1-\gamma}$ при $\gamma < 1$. Таким образом:

- При $\gamma = 0$: $G_0 = 1$ (учитывается только первый шаг).
- При $\gamma = 0.5$: $G_0 = \frac{1}{0.5} = 2$.
- При $\gamma = 0.9$: $G_0 = \frac{1}{0.1} = 10$.
- При $\gamma = 0.99$: $G_0 = 100$.
- При $\gamma \to 1^-$ (очень близко к 1) G_0 становится очень большим (эффективно эпизод “бесконечно длинный” с точки зрения оценки).
- При $\gamma = 1.0$ – формула неприменима (ряд расходится, формально $G_0 = \infty$ если эпизод не прерывается).

В задачах RL обычно эпизоды либо бесконечные с $\gamma < 1$, либо имеют вероятность окончания (что эквивалентно эффективному $\gamma < 1$ как вероятность продолжения). Например, если эпизод длится в среднем T шагов, то берут γ примерно $1 - \frac{1}{T}$ (тогда $1/(1-\gamma) \approx T$).

Аналогия (финансы): Дисконтирование в RL похоже на идею *приведённой стоимости денег*: сегодня 1 ценится выше, чем обещание 1 через год. Если $\gamma = 0.95$ интерпретировать как “коэффициент ежегодного обесценивания” (5% в год), то бесконечный поток выплат 1 каждый год «стоит» 20 в текущих ценах – аналогично сумме ряда $1 + 0.95 + 0.95^2 + \cdots = 20$. В экономике такая сумма называется **стойкостью (perpetuity)**. В RL γ играет ту же роль – сравнивает ценность немедленного вознаграждения и отсроченного.

3.4 Конвергенция алгоритмов: пример

Мы уже рассмотрели небольшой пример итерации ценности (в разделе 2.2). Ещё один близкий пример – **итеративное вычисление ценности политики** (policy evaluation). Пусть у нас задана некоторая стратегия π. Можно найти V^π решением линейной системы или итеративно: начать с $V_0(s) = 0$ и применять оператор $\mathcal{T}^\pi[V](s) = \sum_{s'} P(s'|s,\pi(s))[R(s,\pi(s),s') + \gamma V(s')]$ многократно. Этот процесс сходится к V^π по той же причине сжатия. Например, если задать фиксированную стратегию в нашей 2 -состояной MDP (скажем, она всё равно детерминирована: $S_1 \to S_2$, $S_2 \to S_1$), то оценка V_k сойдётся к тому же решению v_1, v_2 без $\max$. Разница только в том, что для оптимума мы каждый шаг брали $\max$, а для заданной политики – нет. Оба процесса – случаи общего метода итерации на основе предела.

Важно: на практике алгоритмы обучения (например, TD(0), Q-learning) сходятся *приблизительно*, так как обновления могут быть шумными, среда неизвестна заранее и т.п. Однако в предположении бесконечного числа посещений и малого шага обучения можно показать, что Q-значения сойдутся к оптимальным с вероятностью 1 (в силу теоремы о сходимости стохастического аппроксимационного сжатия). Такие доказательства выходят за рамки первого курса, но интуитивно опираются на тот же фундамент: повторяющиеся обновления ценностей эквивалентны поиску фиксированной точки, а фактор $\gamma < 1$ гарантирует, что «шаги» обновления убывающие.

4. Визуализация сходимости: Jupyter-демонстрации

Чтобы лучше понять динамику сходимости рядов и алгоритмов, рассмотрим две наглядные визуализации, сгенерированные с помощью Python (Jupyter):

4.1 Сходимость геометрического ряда

На следующем графике показано, как нарастает частичная сумма $S_N = \sum_{t=0}^{N-1} \gamma^t$ при увеличении числа слагаемых N для разных значений γ. Пунктирными линиями отмечен теоретический предел $S_\infty = \frac{1}{1-\gamma}$.

Суммы геометрического ряда при разных коэффициентах γ. Жёлтая кривая – для $\gamma = 0.5$, красная – для $\gamma = 0.9$. При $\gamma = 0.5$ ряд сходится очень быстро: уже после $\sim 5-6$ слагаемых сумма почти достигает предела 2. При $\gamma = 0.9$ сходимость гораздо медленнее: частичная сумма растёт плавно и лишь к $N = 30$ приближается к значению ~ 9.58 (предел равен 10). Видно, что γ ближе к 1 даёт “длинный хвост” ряда – это аналог длительного учета будущего в RL. Тем не менее, в обоих случаях суммы стремятся к конечному лимиту, подтверждая формулу $\sum_{t=0}^{\infty} \gamma^t = \frac{1}{1-\gamma}$.

Примечание: даже при $\gamma = 0.9$ дальние члены ($\gamma^{20} \approx 0.12$, $\gamma^{50} \approx 0.005$) очень малы. Поэтому на практике часто говорят об *эффективном горизонте* $\approx \frac{1}{1-\gamma}$ шагов – далее вклад вознаграждений незначителен.

4.2 Визуализация процесса итерации ценности

Рассмотрим ещё раз пример из разд. 2.2 (два состояния, циклические переходы). Построим график изменения оценок $V_k(S_1)$ и $V_k(S_2)$ за несколько итераций.

На графике ниже каждая ступенька показывает новый вектор ценностей после очередного *синхронного* обновления по Беллману. Мы видим, что $V_k(S_1)$ (жёлтый график) и $V_k(S_2)$ (оранжевый) постепенно сходятся к своим оптимальным значениям.

Как и обсуждалось, процесс не монотонный: он **осциллирует**, приближаясь к пределу с разных сторон. Это связано с тем, что в данном MDP награды чередуются через шаг – поэтому оценка S_1 подскакивает, потом S_2 подтягивается, затем снова S_1 и т.д. Но **максимальная разница** между текущими оценками и оптимальными уменьшается примерно в 0.9 раза каждую итерацию (коэффициент дисконтирования), что и приводит к сжатию "коридора" осцилляции. Через достаточное число итераций ($k \to \infty$) графики сходятся на горизонтальных уровнях $V^*(S_1) \approx 10.53$ и $V^*(S_2) \approx 9.47$.

Мы можем наблюдать здесь визуальное подтверждение сжимающего свойства: первые шаги ($0 \to 1$, $1 \to 2$ итерации) меняют ценности существенно, а на поздних шагах ($10 \to 11$, $11 \to 12$ итерации) изменения совсем малые – кривые уже почти выровнялись. В итоге предельное поведение (после бесконечного числа итераций) соответствует решению уравнения Беллмана, как и должно.

(График итерации ценности приведён выше: .)

5. Источники и материалы для дальнейшего изучения

Ниже перечислены рекомендуемые источники (русскоязычные и англоязычные), которые помогут углубить понимание темы:

- **Саттон Р.**, **Барто Э.** *"Обучение с подкреплением: Введение"*, 2-е изд., 2020 – классический учебник по RL (есть русский перевод). В Главе 3 вводится формализм MDP, дисконтирование и уравнения Беллмана, а в Главе 4 – алгоритмы динамического программирования (итерация ценности и политики). Это основной учебный материал по RL.
- **Silver D.** *"Reinforcement Learning course"* (Университетский колледж Лондона, 2015) – серия видеолекций от одного из создателей AlphaGo. Лекция 1 вводит основы MDP, а лекции 2–3 охватывают уравнения Беллмана и методы DP. *Доступ:* YouTube, а также конспекты в открытом доступе.
- **Khan Academy (рус)** – раздел *"Пределы и ряды"* (в курсе по математическому анализу). Краткие видео и задачи по определению предела, бесконечным рядам и признакам сходимости. Хорошо подходит для первого знакомства с темой.
- **Курс "Математический анализ – 1 курс"** – любой университетский учебник или онлайн-курс, охватывающий основы анализа: пределы последовательностей, непрерывность, числовые ряды. Например, курс на *Открытом образовании НИУ ВШЭ* или лекции на ресурсе *eor.dgu.ru* содержат строгие определения и доказательства сходимости.
- **Wikipedia (En/Ru)** – статьи: "Limit of a sequence" (Предел последовательности), "Series (mathematics)" (Числовой ряд) содержат формальные определения и примеры. Статьи

"Bellman equation" и "Markov decision process" дают обзор уравнений Беллмана и дисконтирования в контексте RL.

- **Stepik курс "Глубокое обучение с подкреплением"** – онлайн-курс (2018, на русском) по современным методам RL, включает модули по базовым понятиям RL, где разбираются и классические алгоритмы, и математические основы. Полезен после освоения базиса, чтобы увидеть развитие темы.
- **Монтомери Д.** *"Введение в математический анализ"* – глава, посвящённая числовым рядам. (Пример условного источника на русском языке для дополнения – подставьте существующий учебник, например *Кудрявцев Л.Д., Курс математического анализа*).
- **Bertsekas D.** *"Dynamic Programming and Optimal Control"* – классический английский текст по динамическому программированию. Часть теоретических основ RL изложена там, включая сжимающие отображения и доказательства сходимости итерации ценности.

Каждый из этих источников дополняет материал модуля: от базовой математики до продвинутых алгоритмов RL. Рекомендуется начать с учебника Саттона и Барто для целостного понимания, параллельно освежая основы анализа по учебнику или Khan Academy, и затем двигаться к более специализированным курсам и статьям.

6. Мини-глоссарий ключевых понятий

- **Последовательность** (x_n) – набор пронумерованных элементов $x_1, x_2, \dots, x_n, \dots$. *Пример:* последовательность вознаграждений во времени $R_1, R_2, \dots, R_t, \dots$.
- **Предел последовательности** ($\lim_{n\to\infty} x_n = a$) – число a, к которому приближаются члены последовательности. Формально: $\forall \varepsilon > 0; \exists N : \forall n > N, ; |x_n - a| < \varepsilon$. Если такой a существует (конечный), то последовательность *сходится* к a.
- **Ряд (числовой)** $\sum_{n=1}^{\infty} a_n$ – бесконечная сумма $a_1 + a_2 + a_3 + \dots$. Определяется последовательностью частичных сумм $S_N = \sum_{n=1}^{N} a_n$. Если S_N имеет конечный предел S при $N \to \infty$, то ряд сходится к S.
- **Сходимость ряда** – наличие конечного предела $S = \lim_{N\to\infty} S_N$. Например, $1 + 1/2 + 1/4 + 1/8 + \cdots = 2$. Если предел частичных сумм бесконечен или не существует, ряд *расходится*.
- **Абсолютная сходимость** – ряд $\sum a_n$ сходится *абсолютно*, если сходится ряд из модулей $\sum |a_n|$. Например, $1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots$ сходится условно (до $\ln 2$), но не абсолютно (модули дают гармонический ряд).
- **Дисконтирование (вознаграждений)** – умножение вознаграждения на коэффициент γ^t за задержку на t шагов. **Коэффициент дисконтирования** $\gamma \in [0, 1]$ – параметр,

определяющий “ценность времени”: при $\gamma < 1$ будущие награды обесцениваются экспоненциально. В RL $\sum_{t=0}^{\infty} \gamma^t R_{t+1}$ – дисконтированный возврат.

- **Марковский процесс принятия решений (MDP)** – математическая модель среды в RL: задано множество состояний S, действий A, функции переходов $P(s'|s,a)$ и вознаграждений $R(s,a,s')$, а также γ. На каждом шаге агент наблюдает состояние s, выбирает действие a, получает $R(s,a,s')$ и переходит в новое состояние s' по P.
- **Политика (strategy, π)** – правило выбора действий. Может быть детерминированной ($a = \pi(s)$) или стохастической ($\pi(a|s) = P(A = a|S = s)$). Определяет поведение агента.
- **Функция ценности состояния** $V^{\pi}(s)$ – ожидаемое суммарное дисконтированное вознаграждение при старте в состоянии s и дальнейшем следовании политике π:

$$V^{\pi}(s) = \mathbb{E}_{\pi}\Big[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \,\Big|\, S_0 = s\Big] .$$

- **Функция ценности действия** $Q^{\pi}(s,a)$ – ожидаемое суммарное вознаграждение при старте в состоянии s, выполнении действия a и затем следовании политике π: $Q^{\pi}(s,a) = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t R_{t+1}|S_0 = s, A_0 = a]$.
- **Оптимальная функция ценности** $V^*(s)$ – максимальное возможное $V^{\pi}(s)$ по всем стратегиям π. Аналогично оптимальные $Q^*(s,a)$. Эти функции удовлетворяют уравнениям оптимальности Беллмана.
- **Уравнение Беллмана (для политики π)** – рекуррентное соотношение: $V^{\pi}(s) = \sum_a \pi(a|s) \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V^{\pi}(s')]$.
- **Уравнение Беллмана оптимальности** – уравнение для V*:

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} P(s'|s,a) \left[R(s,a,s') + \gamma V^*(s')\right] .$$

  Это уравнение динамического программирования, достаточное условие оптимальности. Его решение даёт оптимальные ценности и стратегию $\pi^*(s) = \arg\max_a \sum_{s'} P(s'|s,a)[R + \gamma V^*(s')]$.
- **Итерация ценности** – алгоритм вычисления V^* путём итерационного применения оператора Беллмана: $V_{k+1} \leftarrow \mathcal{T}[V_k]$ с V_0 начальным (например, нулевым). Сходится к V^* за счёт контрактности оператора.
- **Итерация политики** – алгоритм поиска оптимальной политики через чередование шагов *policy evaluation* (вычисление V^{π} для текущей политики) и *policy improvement* (улучшение политики на основе текущего V). Теоретически сходится за конечное число итераций к оптимальной политике.
- **Q-обучение** – офлайн-алгоритм RL, который обновляет приближение к $Q^*(s,a)$ по формуле: $Q_{new}(s,a) \leftarrow Q(s,a) + \alpha[R(s,a,s') + \gamma \max_{a'} Q(s',a') - Q(s,a)]$. Сходится к Q^* при посещении всех состояний-бесконечно часто и при убывающем α (опирается на сжимающий оператор $\mathcal{T}$*).

- **Контрактное отображение (сжатие)** – отображение F на метрическом пространстве, для которого $\exists 0 \leq \kappa < 1 : d(F(x), F(y)) \leq \kappa d(x, y)$ для всех x, y. Имеет единственную неподвижную точку, и итеративный процесс $x_{n+1} = F(x_n)$ сходится к ней. В контексте RL оператор Беллмана $\mathcal{T}$ – γ-сжатие в максимум-норме, с $\kappa = \gamma < 1$.

Этот глоссарий охватывает основные понятия, встречающиеся в модуле. Понимание их и взаимосвязей (например, **сходимость последовательности значений** V_k **к** V* **благодаря контрактности оператора с коэффициентом** γ) является фундаментом для дальнейшего изучения методов обучения с подкреплением.