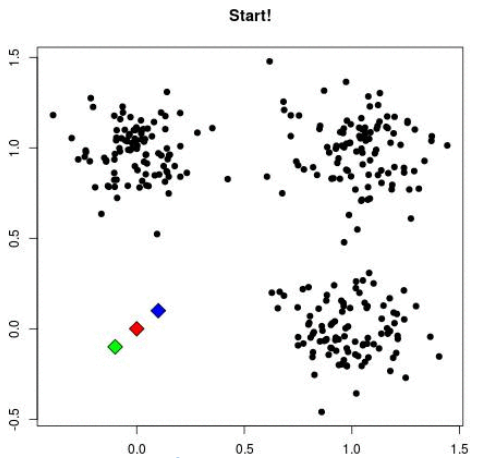
Лекция 3. Обучение без учителя и обучение с подкреплением / Unsupervised and Reinforcement Learning

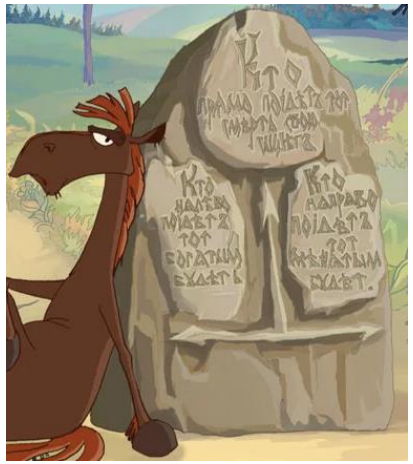
"If you drink too much from a bottle marked "poison," it's almost certain to disagree with you sooner or later".

Alice

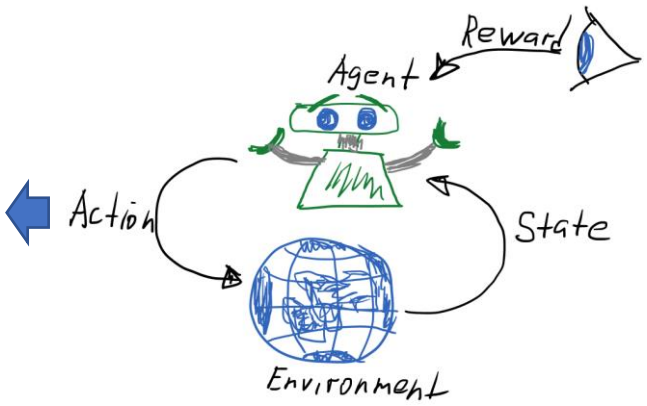
1. **Напоминание / Contents of the previous lecture**
2. **Обучение без учителя: кластеризация / Unsupervised learning: clustering**
3. **Обучение с подкреплением: оптимизация поведения / Reinforcement learning: policy optimization**
4. **Обучение с подкреплением: оптимизация награды / Reinforcement learning: value optimization**

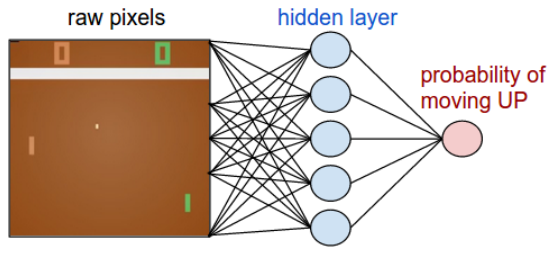


K-means clustering intuition



Q-table analogue





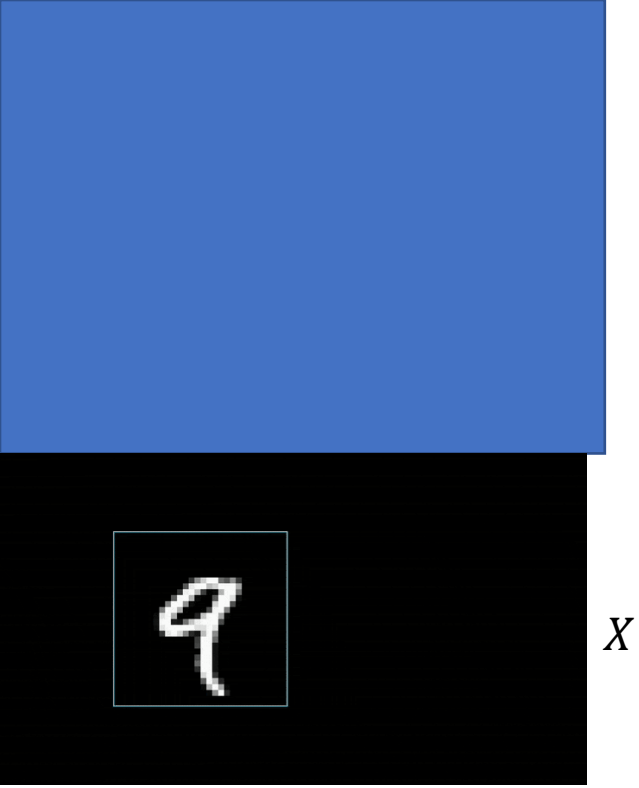
The Pong game

Напоминание / Contents of the previous lecture

Обучение с учителем подразумевает наличие правильных ответов (labeled data), которые можно сравнить с результатами вычислений ИНС.

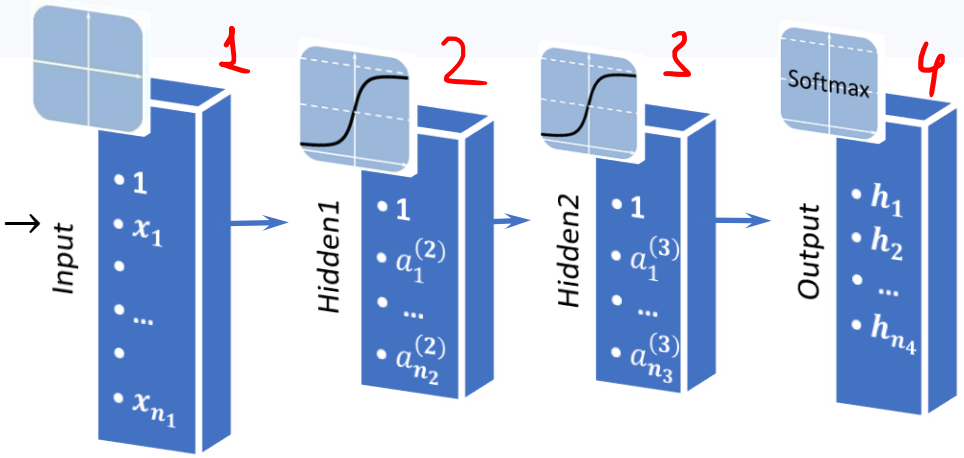
Задача распознавания (классификации) рукописных чисел

Архитектура ИНС: количество слоев - l; количество нейронов в k-ом слое - n_k (n_l - количество классов); логистическая функция активации в скрытых слоях и функция активации «софтмакс» в выходном слое.



Gray scale picture of "Nine"

$$X = \begin{pmatrix} x_1 & x_2 & \dots & x_{n_1} \end{pmatrix} \rightarrow$$



$$\rightarrow H = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ \dots \\ h_{10} \end{pmatrix} \leftrightarrow \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ \dots \\ 9 \end{matrix}$$

Вычисления в прямом направлении ИНС, расчет матриц $A^{(k)}$ результатов в каждом слое / Forward propagation

Сл.1 (входной). На вход слоя подается дополненная единицей матрица X. На выходе то же: $A^{(1)} = \begin{pmatrix} 1 & X \end{pmatrix}$.

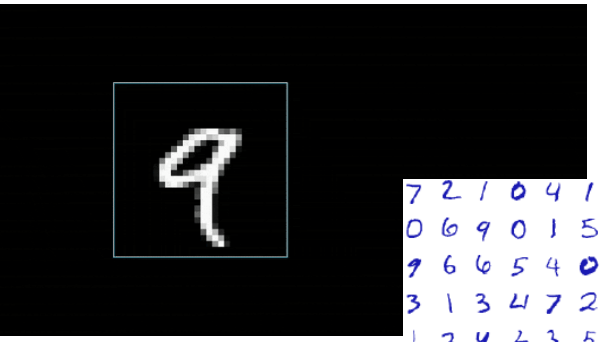
Сл.2 (скрытый). Данные с 1го слоя умнож. на веса $\Theta^{(1)}$ и сумм.: $Z^{(2)} = A^{(1)}\Theta^{(1)}$. Затем прим. ф-я актив.: $A^{(2)} = \begin{pmatrix} 1 & sigmoid(Z^{(2)}) \end{pmatrix}$.

Сл.3 (скрытый). $Z^{(3)} = A^{(2)}\Theta^{(2)}, \quad A^{(3)} = \begin{pmatrix} 1 & sigmoid(Z^{(3)}) \end{pmatrix}$.

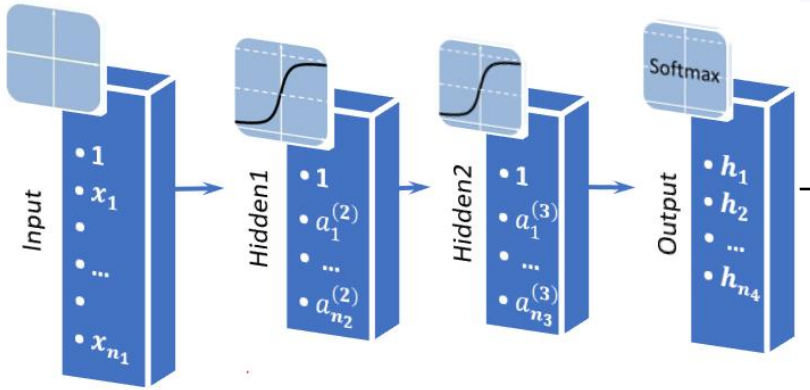
Сл.4 (выходной). $Z^{(4)} = A^{(3)}\Theta^{(3)}, \quad A^{(4)} = softmax(Z^{(4)}) = H$.

Напоминание / Contents of the previous lecture

Функция качества в ИНС. Количество слоев - l; количество нейронов в k-ом слое - n_k (n_l - количество классов); логистическая функция активации в скрытых слоях и функция активации «софтмакс» в выходном слое.

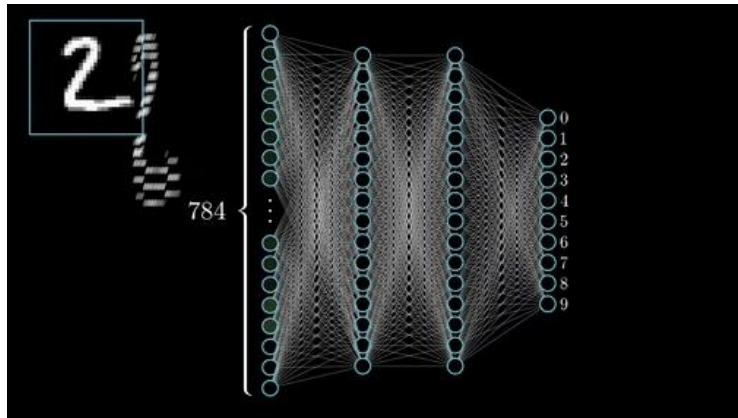


$$X = \left(\left(x_1 \dots x_{n_1} \right)\right); \rightarrow$$



$$\rightarrow H = \begin{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ \dots \\ h_{10} \end{pmatrix} \end{pmatrix} \leftrightarrow Y = \begin{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 1 \end{pmatrix} \end{pmatrix}.$$

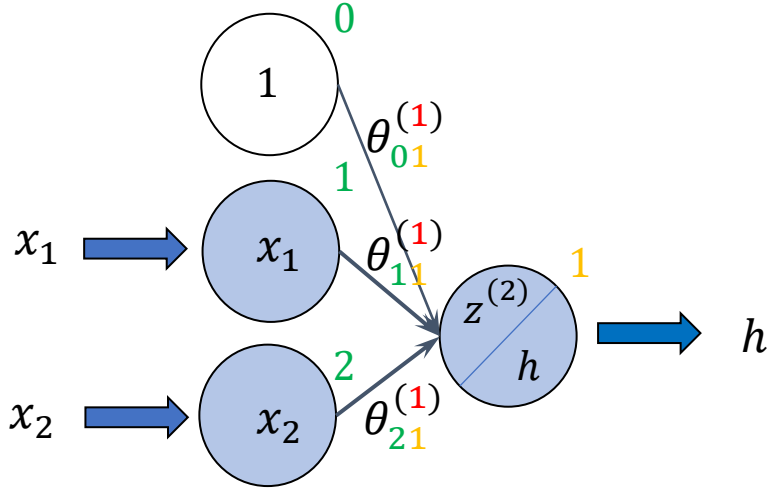
[Animations from "3Blue1Brown"](#)

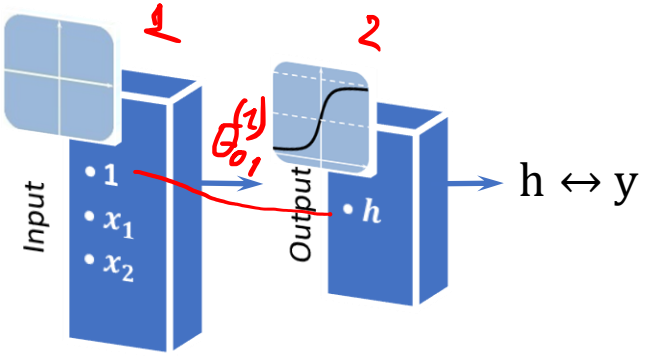


$$J\left(\Theta^{(k)}\right) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n_l}\left(y_j^{(i)}\ln\left(h_j^{(i)}\right) + \left(1 - y_j^{(i)}\right)\left(\ln\left(1 - h_j^{(i)}\right)\right)\right) + \frac{\lambda}{2m}\sum_{k=1}^{l-1}\sum_{i=1}^{n_k}\sum_{j=1}^{n_{k+1}}\left(\theta_{ij}^{(k)}\right)^2 \Rightarrow \min.$$

Напоминание / Contents of the previous lecture

Пример прямого расчета. ИНС содержит 2 входных нейрона и 1 выходной нейрон с логистической функцией активации. На вход подается матрица $X = ((x_1\ x_2))$.





Прямые вычисления в ИНС с количеством нейронов N = [2 1], характеризуемой синаптическими весами $\theta_{ij}^{(1)}$.

$$X = ((x_1\ x_2)); \rightarrow A^{(1)} = ((1\ x_1\ x_2)), \Theta^{(1)} = \begin{pmatrix} \theta_{01}^{(1)} \\ \theta_{11}^{(1)} \\ \theta_{21}^{(1)} \end{pmatrix};\ Z^{(2)} = A^{(1)}\Theta^{(1)} \text{ или } z^{(2)} = a_i^{(1)}\theta_{i1}^{(1)} = a_0^{(1)}\theta_{0j}^{(1)} + a_1^{(1)}\theta_{1j}^{(1)} + a_2^{(1)}\theta_{2j}^{(1)}.$$

$$A^{(2)} = h = sigmoid\left(Z^{(2)}\right) \text{ или } a^{(2)} = h = \frac{1}{1 + e^{-z^{(2)}}}.$$

$\Theta_{ij}^{(k)}$ — # слоя ИНС; # нейр. "k+1" слоя; # нейрона k-го слоя

Напоминание / Contents of the previous lecture

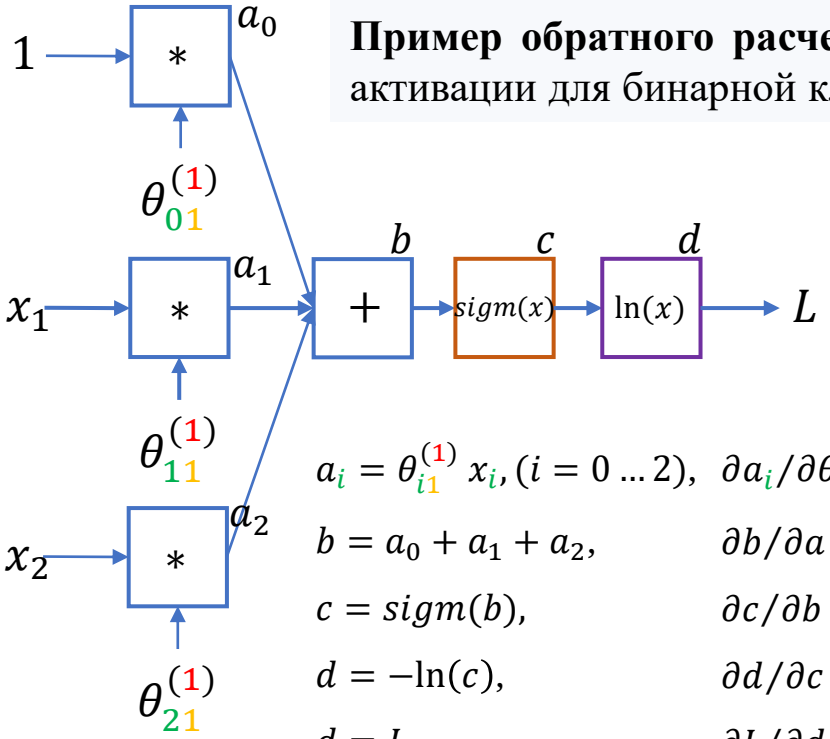
Обратный расчет выполняется с целью определения компонент градиента функции качества и является этапом процесса обучения:

$$L\left(\Theta^{(k)}\right), \rightarrow \left[\left[\partial L/\partial\theta_{ij}^{(k)}\right]\right], \rightarrow \theta_{ij}^{(k)} \rightarrow L\left(\Theta^{(k)}\right), \rightarrow \ldots$$

Функция качества для датасета из m образцов имеет вид:

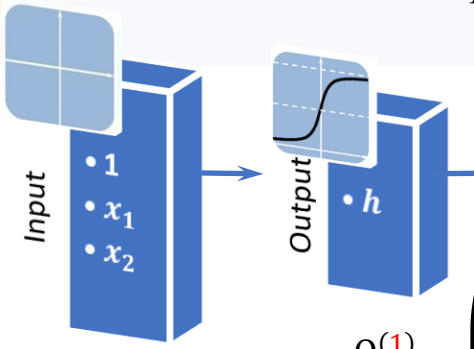
$$L\left(\Theta^{(1)}\right) = -\sum_{i=1}^{m} \ln(h^{(i)}) \Rightarrow \min.$$

Пример обратного расчета. ИНС с 2 входными нейронами и 1 выходным нейроном с логистической функцией активации для бинарной классификации.



$$a_i = \theta_{i1}^{(1)} x_i, (i = 0 \ldots 2), \quad \partial a_i/\partial\theta_{i1}^{(1)} = x_i,$$
$$b = a_0 + a_1 + a_2, \quad \partial b/\partial a = 1,$$
$$c = sigm(b), \quad \partial c/\partial b = c(1-c),$$
$$d = -\ln(c), \quad \partial d/\partial c = -1/c,$$
$$d = L. \quad \partial L/\partial d = 1.$$

$$X = \begin{pmatrix} x_1 & x_2 \end{pmatrix}$$



$$\Theta^{(1)} = \begin{pmatrix} \theta_{01}^{(1)} \\ \theta_{11}^{(1)} \\ \theta_{21}^{(1)} \end{pmatrix}$$

Пусть $m = 1$. Тогда:

$$\frac{\partial L}{\partial\theta_{i1}^{(1)}} = \frac{\partial L}{\partial d}\frac{\partial d}{\partial c}\frac{\partial c}{\partial b}\frac{\partial b}{\partial a}\frac{\partial a_i}{\partial\theta_{i1}^{(1)}} = (c-1)\, x_i = (h-1)\, x_i.$$

Пусть $m > 1$. Тогда:

$$\frac{\partial L}{\partial\theta_{i1}^{(1)}} = \sum_{k=1}^{m} \left(h^{(k)} - 1\right) x_i^{(k)}.$$

Напоминание / Contents of the previous lecture

Обратный расчет выполняется с целью определения компонент градиента функции качества и является этапом процесса обучения:

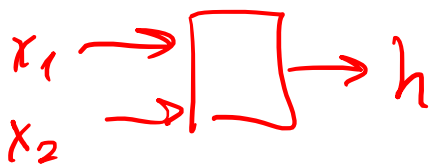
$$L\left(\Theta^{(k)}\right), \rightarrow \left[\left[\partial L/\partial\theta_{ij}^{(k)}\right]\right], \rightarrow \theta_{ij}^{(k)} \rightarrow L\left(\Theta^{(k)}\right), \rightarrow \ldots$$

Функция качества для датасета из m образцов имеет вид:

$$L\left(\Theta^{(1)}\right) = -\sum_{i=1}^{m} \ln(h^{(i)}) \Rightarrow \min.$$

Алгоритм обучения (обобщенный).

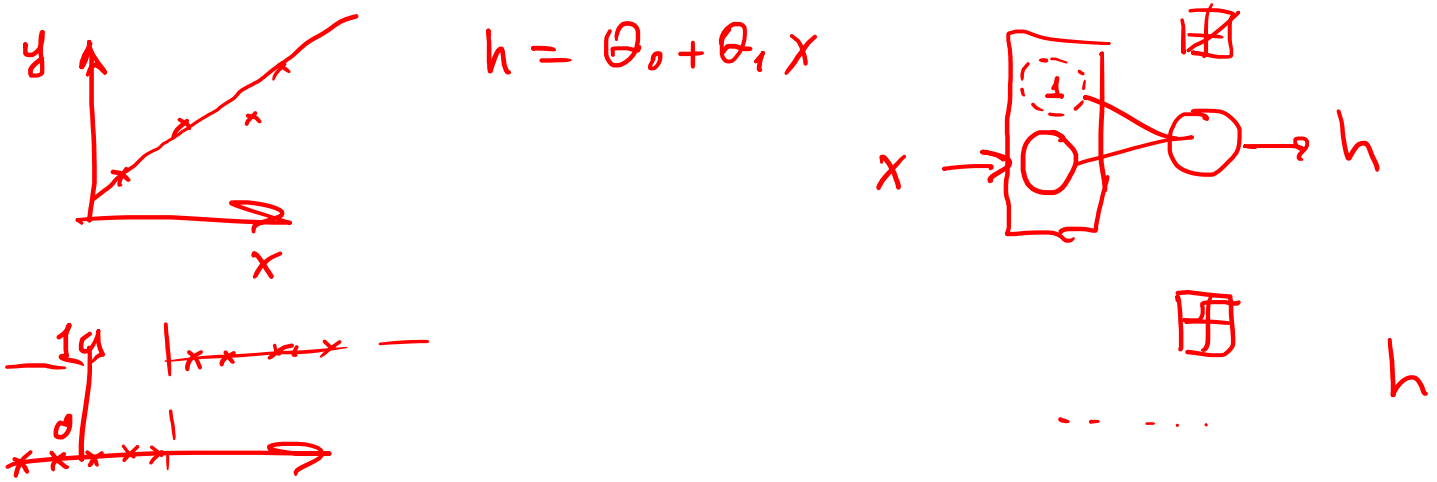
1. Задать начальные значения компонент матрицы $\Theta^{(k)}$ случайным образом.
2. Рассчитать вектор градиента $\nabla L = \left[\left[\partial L/\partial\theta_{ij}^{(k)}\right]\right]$ методом обратного распр. ошибки.
3. Найти новые значения компонент $\Theta^{(k)}$: ${\theta_{ij}^{(k)}}^{H} = {\theta_{ij}^{(k)}}^{C} - \alpha \frac{\partial L}{\partial \theta_{ij}^{(k)}}$.
4. Повторять пп. 2-3 до достижения минимума L: $L^{H} - L^{C} < \delta$ или #итерации > N_{max}.
5. Вывод результатов: $\Theta^{(k)}$.



Самостоятельная работа (лекция №2) / Homework

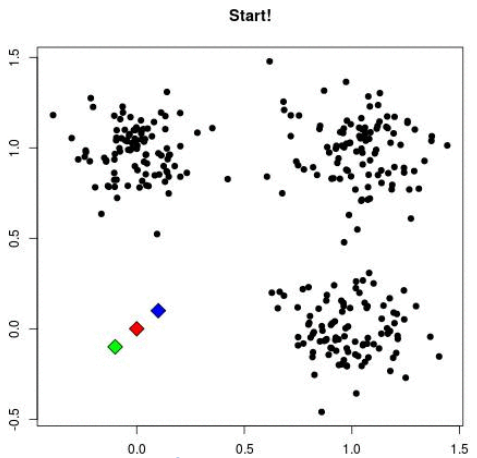
Вопросы и задания.

1. Изобразите архитектуры простейших нейронных сетей, вычисления в которых идентичны вычислениям при линейной и логистической регрессии.
2. Каким образом в ИНС хранятся знания и как они из ИНС извлекаются?
3. Позволяет ли применение метода Mini-Batch GD решить проблему поиска глобального экстремума при наличии локальных?

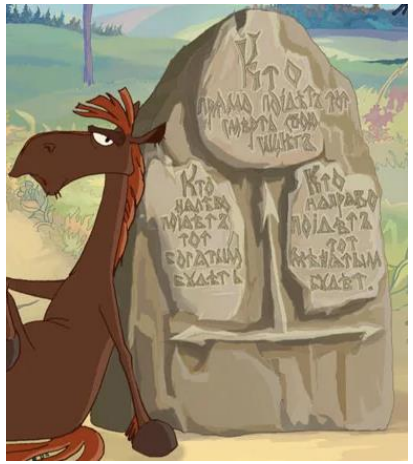


Лекция 3. Обучение без учителя и обучение с подкреплением / Unsupervised and Reinforcement Learning

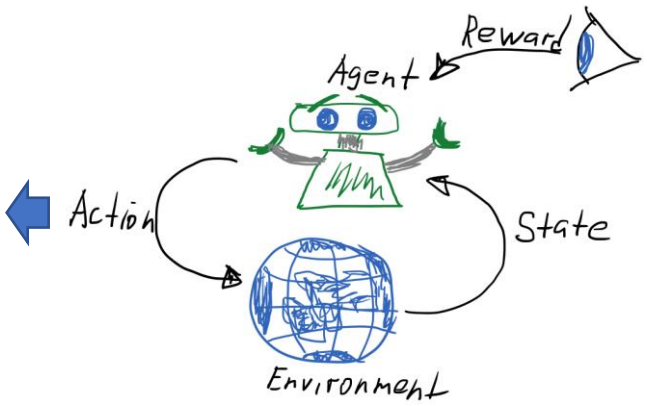
1. Напоминание / Contents of the previous lecture
2. Обучение без учителя: кластеризация / Unsupervised learning: clustering
3. Обучение с подкреплением: оптимизация поведения / Reinforcement learning: policy optimization
4. Обучение с подкреплением: оптимизация награды / Reinforcement learning: value optimization

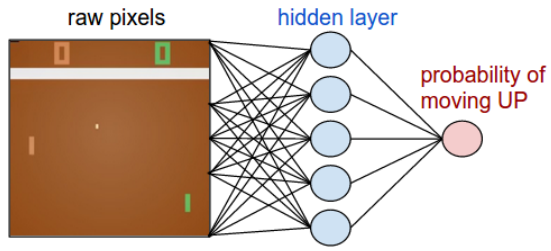


K-means clustering intuition



Q-table analogue



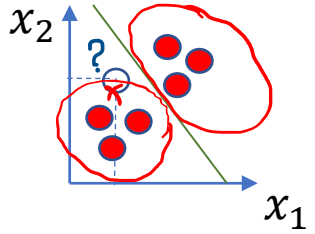


The Pong game

Кластеризация / Clustering

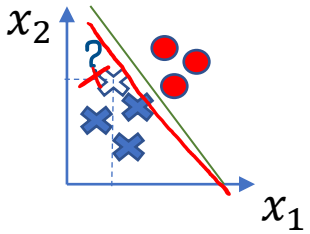
Машинное обучение / ML

Обучение без учителя /
Unsupervised learning



$$X = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} \\ \dots & \dots \\ x_1^{(m)} & x_2^{(m)} \end{pmatrix};$$

Обучение с учителем /
Supervised learning

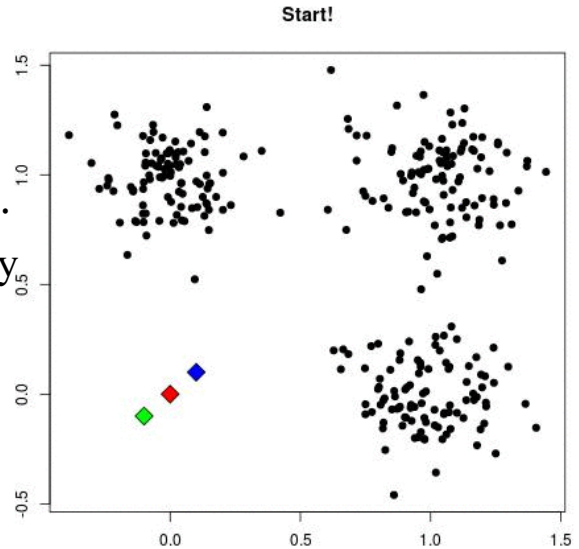


Кластерный анализ - статистическая процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в сравнительно однородные группы (кластеры) [Кластерный_анализ].

Метод k-средних. Основная **идея** метода в разделении данных на «*k*» кластеров по признаку наименьшего расстояния до одного из центров кластеров. При этом положение центров кластеров итерационно пересчитывается.

Алгоритм обучения.

1. Назначается количество кластеров «*k*».
2. Случайным образом назначаются центры кластеров $\mu_i^{(k)}(i = 1,2, \dots N)$, N – мерность пространства.
3. Определяется принадлежность каждой точки $x_i^{(j)}$ $(i = 1,2, \dots m)$ к одному из кластеров по признаку наименьшей из «k» величин: $\min\limits_k \sum_{i=1}^{m} \left(x_i^{(j)} - \mu_i^{(k)}\right)^2$, m – количество точек.
4. Вычисляются новые координаты каждого кластера $\mu_i^{(k)}$ как средние арифметические координат элементов данного кластера.
5. Пункты 3,4 повторяются, минимизируется функция: $J = \frac{1}{m}\sum_{i=1}^{m} \left(x_i^{(j)} - \mu_i^{(c_i)}\right)^2$.

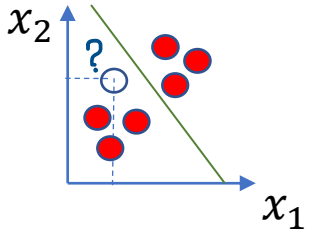


K-means clustering intuition

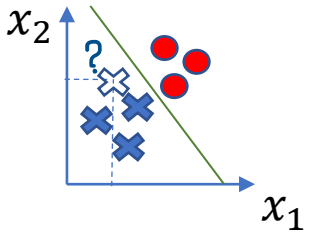
Кластеризация / Clustering

Машинное обучение / ML

Обучение без учителя /
Unsupervised learning



Обучение с учителем /
Supervised learning

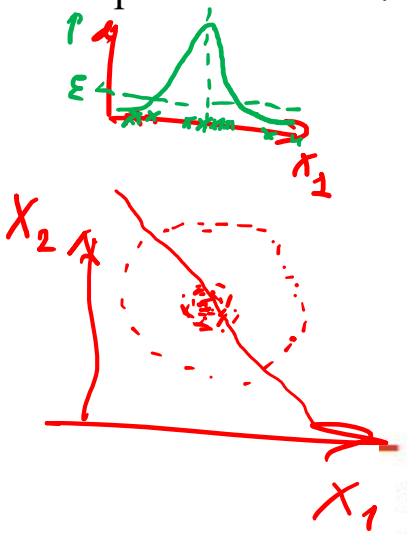


Кластерный анализ - статистическая процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в сравнительно однородные группы (кластеры) [Кластерный_анализ].

Метод k-средних. Основная **идея** метода в разделении данных на «*k*» кластеров по признаку наименьшего расстояния до одного из центров кластеров. При этом положение центров кластеров итерационно пересчитывается.

Особенности обучения.

1. Исследователю необходимо выбирать количество кластеров.
2. Исходные данные должны иметь различимые центры кластеров.

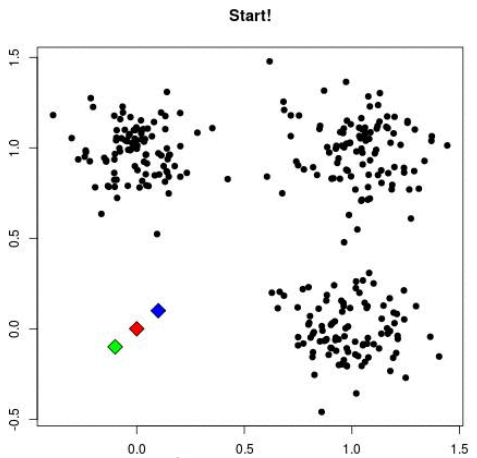


Lecture 15.1 — Anomaly Detection Problem | Motivatic
Anomaly Detection • 1 / 8

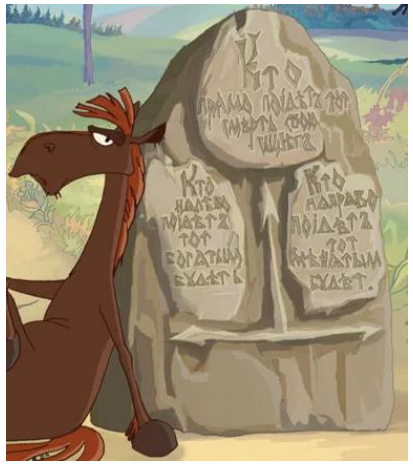
[Anomaly detection by A.Ng]

Лекция 3. Обучение без учителя и обучение с подкреплением / Unsupervised and Reinforcement Learning

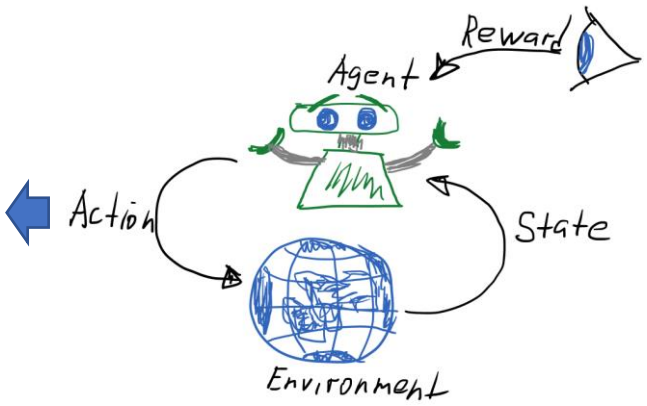
1. Напоминание / Contents of the previous lecture
2. Обучение без учителя: кластеризация / Unsupervised learning: clustering
3. Обучение с подкреплением: оптимизация поведения / Reinforcement learning: policy optimization
4. Обучение с подкреплением: оптимизация награды / Reinforcement learning: value optimization

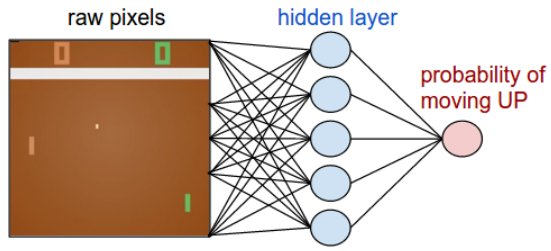


K-means clustering intuition



Q-table analogue

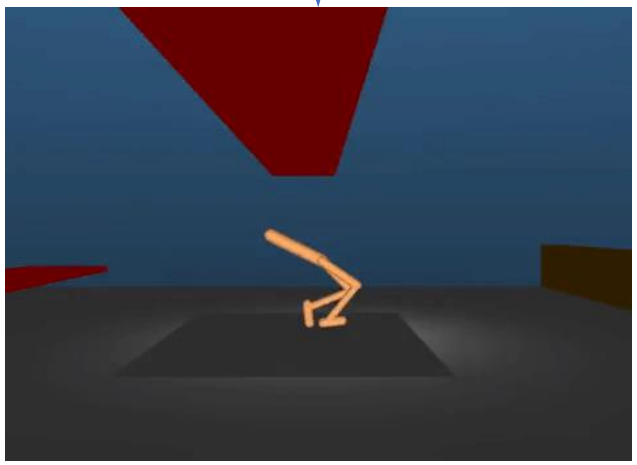




The Pong game

Зачем это нужно, если есть Федор? / FAQ

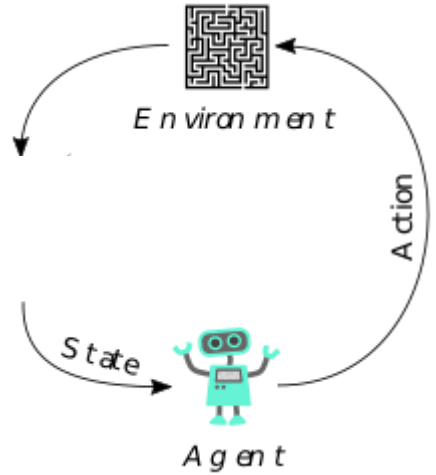
Обучение с подкреплением / Reinforcement learning



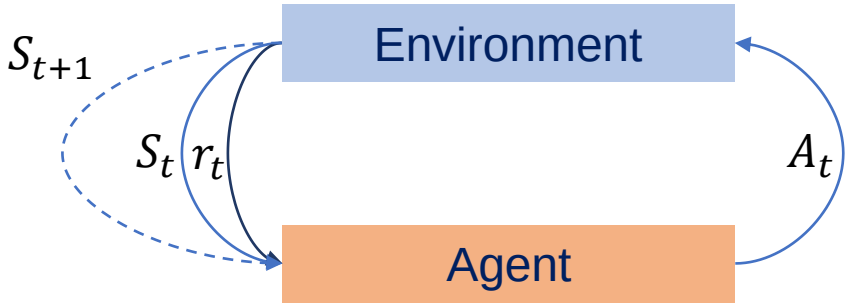
Основы обучения с подкреплением / RL basics

Control systems based on Reinforcement Learning (RL)

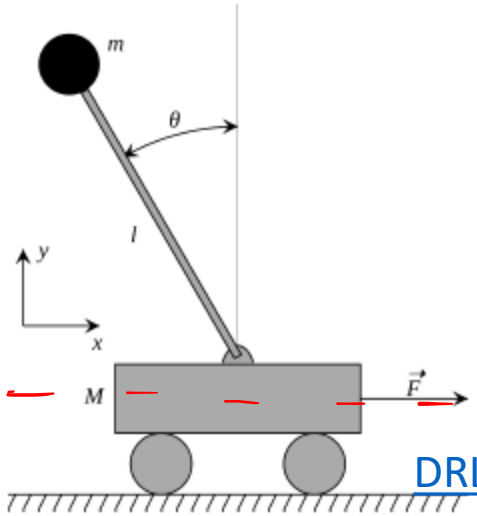
Control or decision process: at each time step, the controller (agent) receives feedback from the system (environment) in the form of a state signal, and takes an action in response. We supposed that current state completely characterizes the state of the system (Markov decision process).
The main problem is that the correct actions are unknown sometimes.
The main idea is to learn agent after the event, giving him higher reward for the better actions.



RL scenario



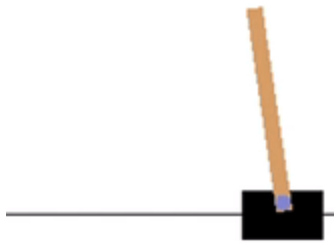
S_t (s_t) – состояние среды (state) в момент времени t;
A_t (a_t) – действие агента (action) в момент времени t;
r_t – награда агента (reward) в момент времени t.



Objective: Balance a pole on top of a movable cart

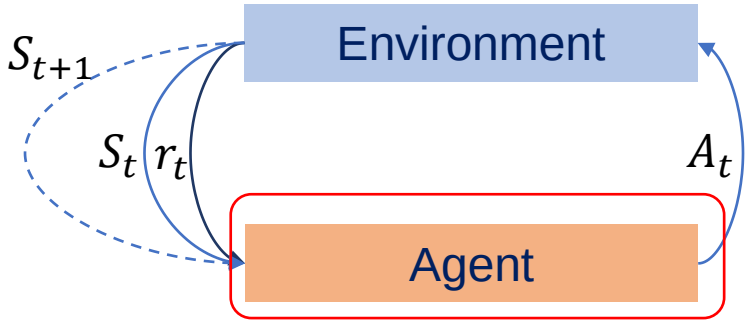
State: angle, angular speed, position, horizontal velocity
Action: horizontal force applied on the cart
Reward: 1 at each time step if the pole is upright

DRL course in Stanford University School of Engineering



Всякий задним умом крепок /
An after-wit is everybody's wit

Основы обучения с подкреплением / RL basics



Характеристики модели RL:

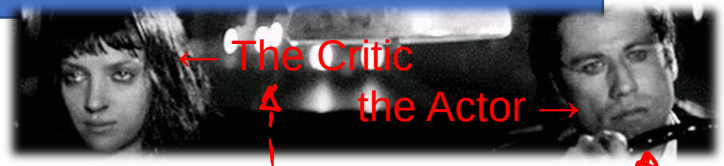
S_t (s_t) – состояние среды (state) в момент времени t;

A_t (a_t) – действие агента (action) в момент времени t;

r_t – награда агента (reward) в момент времени t;

$g_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$ - будущая награда (return);

γ - дисконт (discount).

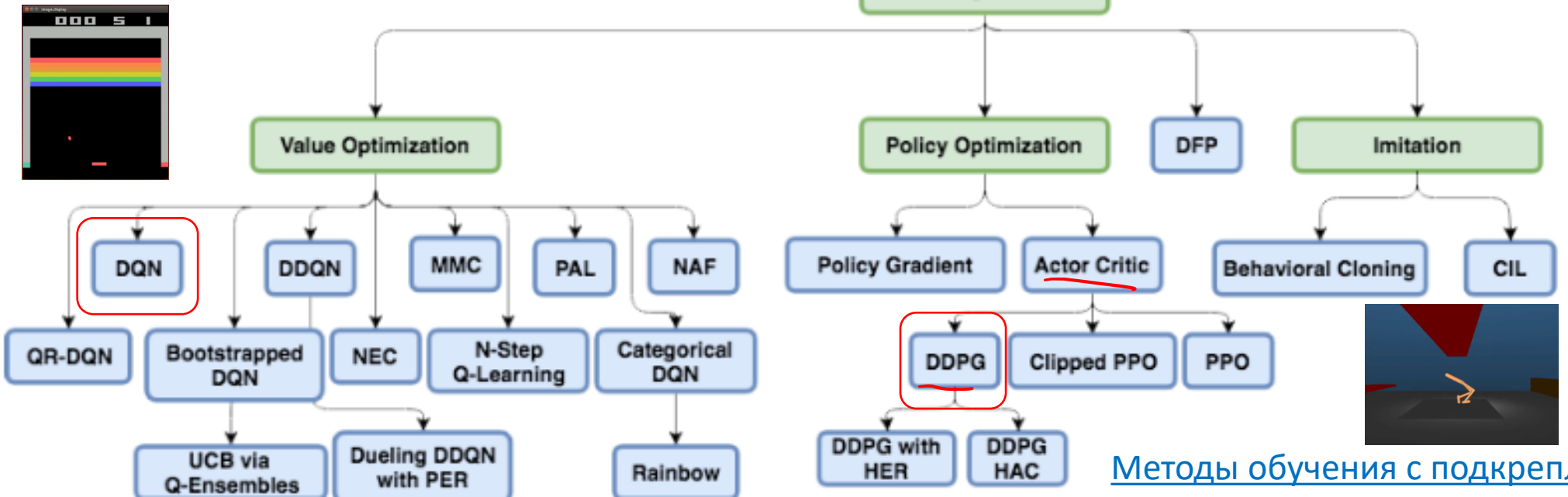


В зависимости от типа агента в процессе обучения аппроксимируются функции:

$v_t(S_t)$ - оценка будущей награды на основе наблюдения в мом. вр. t (value function);

$q_t(S_t, A_t)$ - оценка будущей награды на основе наблюдения и действия в мом. вр. t (q-function);

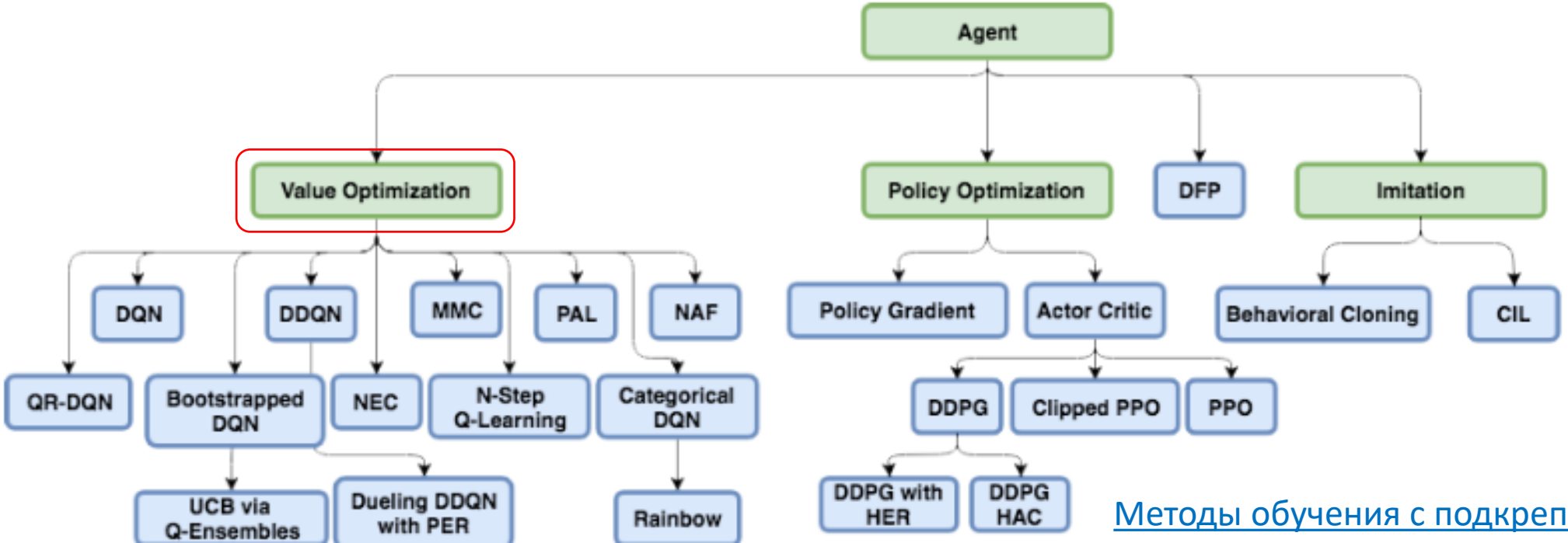
$p_t(S_t)$ - оценка действия на основе наблюдения (policy function).



Методы обучения с подкреплением

Оптимизация поведения / Policy optimization

Основная идея: на основании результатов исследования окружающей среды (environment) обучить модель (critic), которая при данных S_t, A_t предсказывает будущую награду g_t для любого возможного действия A_{t+1}. Тогда агенту (agent) при данных S_t, A_t из множества возможных дальнейших действий A_{t+1}, следует предпринимать то, которое приведет к наибольшей будущей награде g_t.



Методы обучения с подкреплением

Q-обучение / Q-learning

Основная идея: на основании результатов исследования окружающей среды (environment) обучить модель (critic), которая при данных S_t, A_t предсказывает будущую награду g_t для любого возможного действия A_{t+1}. Тогда агенту (agent) при данных S_t, A_t из множества возможных дальнейших действий A_{t+1}, следует предпринимать то, которое приведет к наибольшей будущей награде g_t.

Формализация. В каждый момент времени t функция будущей награды имеет вид:

$$q_t(S_t, A_t) = r_t + \gamma \quad r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{t^{\bowtie}-1} r_{t^{\bowtie}}, \tag{1}$$

где γ – дисконт, $0 < \gamma \leq 1$, $t^{\bowtie}$- шаг по времени при достижении конечного состояния (terminal state) .

Функцию (1) можно представить в виде:

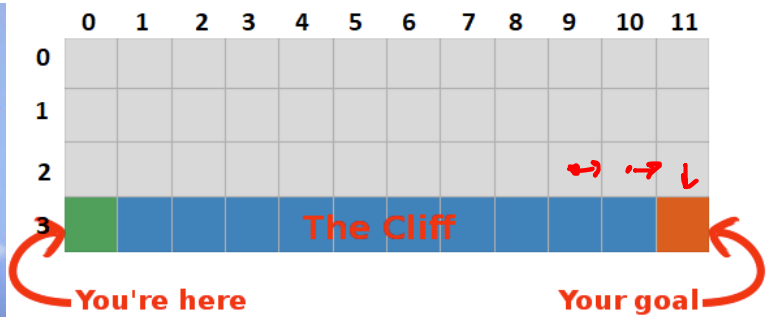
$$q_t(S_t, A_t) = r_t + \gamma\left(r_{t+1} + \gamma \quad r_{t+2} + \cdots + \gamma^{t^{\bowtie}-2} r_{t^{\bowtie}}\right) = r_t + \gamma\, q_{t+1}(S_{t+1}, A_{t+1}). \tag{2}$$

Тогда наилучшая действие q_t^* в момент времени t описывается уравнением Беллмана:

$$q_t^*(S_t, A_t) = r_t + \gamma \max_A [q_{t+1}(S_{t+1}, A_{t+1})]\,. \tag{3}$$

Пример. Задача о прогулке по скале: необходимо найти кратчайший путь до цели, не упав с обрыва [https://habr.com/ru/post/443240/].
Награда за обычный шаг -1, за достижение цели +0, за срыв с обрыва -100. Дисконт 0.9. $S = [\![s^{(i,j)}]\!]$, $A = [\![a^{(k)}]\!] \rightarrow Q = [\![q^{(i,j,k)}]\!]$. Q-?

Agent



4 × 12 = 48

Environment

	$a^{(1)}$ ↑	$a^{(2)}$ ↓	$a^{(3)}$ →	$a^{(4)}$ ←
$s^{(0,0)}$				
$s^{(0,1)}$				
…				
$s^{(2,10)}$	<-1	-1	<-1	<-1
$s^{(2,11)}$	<-1	0	<-1	<-1
…				
$s^{(3,11)}$	0	0	0	0

The Q table aims to optimal

Алгоритм заполнения всей таблицы Q пока не ясен, но для оценки наград в обратном направлении от конечного состояния можно воспользоваться уравнением Беллмана (3):

$$q_t^{*(2,11,2)} = 0 + 0.9 \max_{a_{t+1}}[0,0,0,0] = 0;$$

$$q_t^{*(2,10,3)} = -1 + 0.9 \max_{a_{t+1}}[<-1, \mathbf{0}, <-1, <-1] = -1;$$

$$q_t^{*(2,9,3)} = -1 + 0.9 \max_{a_{t+1}}[<-1, <-1, \mathbf{0}, <-1] = -1.9;$$

…

Q-обучение / Q-learning

Основная идея: на основании результатов исследования окружающей среды (environment) обучить модель (critic), которая при данных S_t, A_t предсказывает будущую награду g_t для любого возможного действия A_{t+1}. Тогда агенту (agent) при данных S_t, A_t из множества возможных дальнейших действий A_{t+1}, следует предпринимать то, которое приведет к наибольшей будущей награде g_t.

Формализация. Наилучшее действие q_t^* в момент времени t описывается уравнением Беллмана:

$$q_t^*(S_t, A_t) = r_t + \gamma \max_A [q_{t+1}(S_{t+1}, A_{t+1})]. \quad (3)$$

Процесс накопления опыта состоит в проигрывании эпизодов. В процессе обучения необходимо достичь минимизации ошибки между обучаемой функцией $Q(S, A)$ и оптимальной:

$$Q(S, A) - Q^*(S, A) \Rightarrow \min.$$

Тогда в результате обучения агент будет способен производить действия с максимальной наградой.

Алгоритм обучения [MATLAB Documentation]/ Training algorithm.

Инициализировать $Q(S, A)$ случайными значениями или нулями. Задать гиперпараметры: вероятность ϵ, скорость обучения α.

Для каждого эпизода обучения:

1. Получить данные о начальном состоянии S_t $(t = 1)$
2. Повторять для каждого шага t до достижения terminal state:

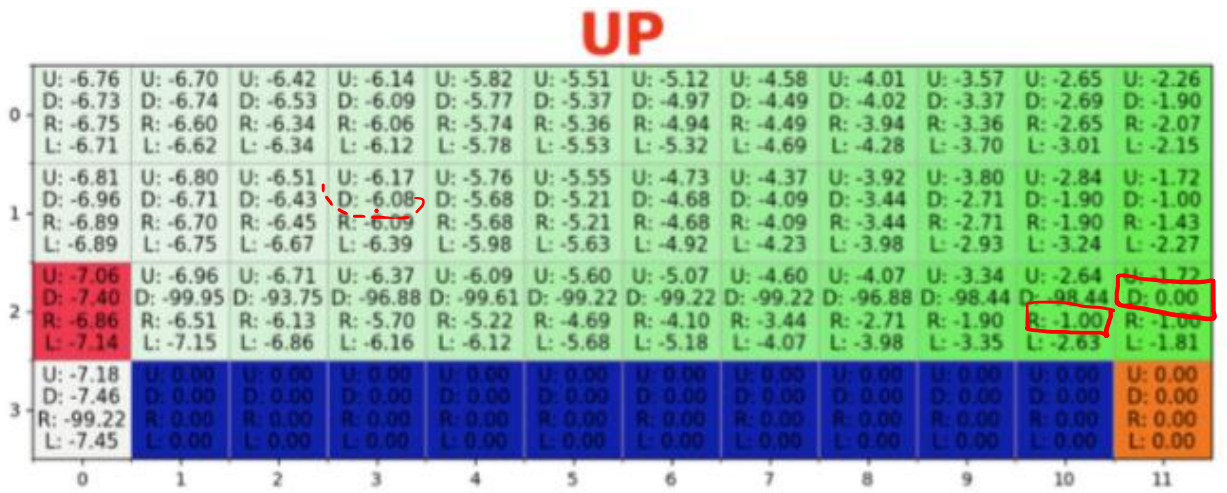
   2.1 Для текущего состояния S_t выбрать случайное действие A_t с вероятностью ϵ (может уменьшаться в процессе обучения), иначе действие, для которого значение наибольшее: $A_t = \max_A [Q_t(S_t, A_t)]$

   2.2 Выполнить действие A_t, получить награду r_t и данные о новом состоянии S_{t+1}.

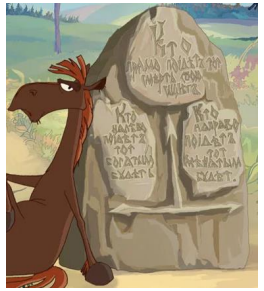
   2.3 Если новое состояние S_{t+1} терминальное, то установить значение целевой функции $y_t = r_t$. Иначе $y_t = r_t + \gamma \max_A [q_{t+1}(S_{t+1}, A_{t+1})]$

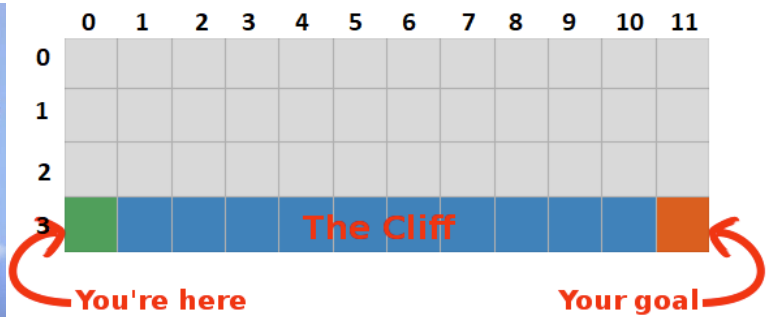
   2.4 Обновить компоненту матрицы $Q(S, A)$:

$$q(S_t, A_t) = q(S_t, A_t) + \propto [y_t - q(S_t, A_t)].$$



https://habr.com/ru/post/443240/





Q-обучение / Q-learning

UP

	0	1	2	3	4	5	6	7	8	9	10	11
0	U: -6.76 D: -6.73 R: -6.75 L: -6.71	U: -6.70 D: -6.74 R: -6.60 L: -6.62	U: -6.42 D: -6.53 R: -6.34 L: -6.34	U: -6.14 D: -6.09 R: -6.06 L: -6.12	U: -5.82 D: -5.77 R: -5.74 L: -5.78	U: -5.51 D: -5.37 R: -5.36 L: -5.53	U: -5.12 D: -4.97 R: -4.94 L: -5.32	U: -4.58 D: -4.49 R: -4.49 L: -4.69	U: -4.01 D: -4.02 R: -3.94 L: -4.28	U: -3.57 D: -3.37 R: -3.36 L: -3.70	U: -2.65 D: -2.69 R: -2.65 L: -3.01	U: -2.26 D: -1.90 R: -2.07 L: -2.15
1	U: -6.81 D: -6.96 R: -6.89 L: -6.89	U: -6.80 D: -6.71 R: -6.70 L: -6.75	U: -6.51 D: -6.43 R: -6.45 L: -6.67	U: -6.17 D: -6.08 R: -6.09 L: -6.39	U: -5.76 D: -5.68 R: -5.68 L: -5.98	U: -5.55 D: -5.21 R: -5.21 L: -5.63	U: -4.73 D: -4.68 R: -4.68 L: -4.92	U: -4.37 D: -4.09 R: -4.09 L: -4.23	U: -3.92 D: -3.44 R: -3.44 L: -3.98	U: -3.80 D: -2.71 R: -2.71 L: -2.93	U: -2.84 D: -1.90 R: -1.90 L: -3.24	U: -1.72 D: -1.00 R: -1.43 L: -2.27
2	U: -7.06 D: -7.40 R: -6.86 L: -7.14	U: -6.96 D: -99.95 R: -6.51 L: -7.15	U: -6.71 D: -93.75 R: -6.13 L: -6.86	U: -6.37 D: -96.88 R: -5.70 L: -6.16	U: -6.09 D: -99.61 R: -5.22 L: -6.12	U: -5.60 D: -99.22 R: -4.69 L: -5.68	U: -5.07 D: -99.22 R: -4.10 L: -5.18	U: -4.60 D: -99.22 R: -3.44 L: -4.07	U: -4.07 D: -96.88 R: -2.71 L: -3.98	U: -3.34 D: -98.44 R: -1.90 L: -3.35	U: -2.64 D: -98.44 R: -1.00 L: -2.63	U: -1.72 D: 0.00 R: -1.00 L: -1.81
3	U: -7.18 D: -7.46 R: -99.22 L: -7.45	U: 0.00 D: 0.00 R: 0.00 L: 0.00	U: 0.00 D: 0.00 R: 0.00 L: 0.00	U: 0.00 D: 0.00 R: 0.00 L: 0.00	U: 0.00 D: 0.00 R: 0.00 L: 0.00	U: 0.00 D: 0.00 R: 0.00 L: 0.00	U: 0.00 D: 0.00 R: 0.00 L: 0.00	U: 0.00 D: 0.00 R: 0.00 L: 0.00	U: 0.00 D: 0.00 R: 0.00 L: 0.00	U: 0.00 D: 0.00 R: 0.00 L: 0.00	U: 0.00 D: 0.00 R: 0.00 L: 0.00	U: 0.00 D: 0.00 R: 0.00 L: 0.00

The Q table

Пример. Задача о прогулке по скале: необходимо найти кратчайший путь до цели, не упав с обрыва [https://habr.com/ru/post/443240/]. Награда за обычный шаг -1, за достижение цели +0, за срыв с обрыва -100. Дисконт 0.9. $S = \llbracket s^{(i,j)} \rrbracket$, $A = \llbracket a^{(k)} \rrbracket \rightarrow Q = \llbracket q^{(i,j,k)} \rrbracket$. Q-?

Agent

Environment

	$a^{(1)}$ ↑	$a^{(2)}$ ↓	$a^{(3)}$ →	$a^{(4)}$ ←
$s^{(0,0)}$				
$s^{(0,1)}$				
…				
$s^{(2,10)}$	<-1	-1	<-1	<-1
$s^{(2,11)}$	<-1	0	<-1	<-1
…				
$s^{(3,11)}$	0	0	0	0

The Q table aims to optimal

Алгоритм заполнения всей таблицы Q пока не ясен, но для оценки наград в обратном направлении от конечного состояния можно воспользоваться уравнением Беллмана (3):

$$q_t^{*(2,11,2)} = 0 + 0.9 \max_{a_{t+1}}[0,0,0,0] = \underline{0};$$

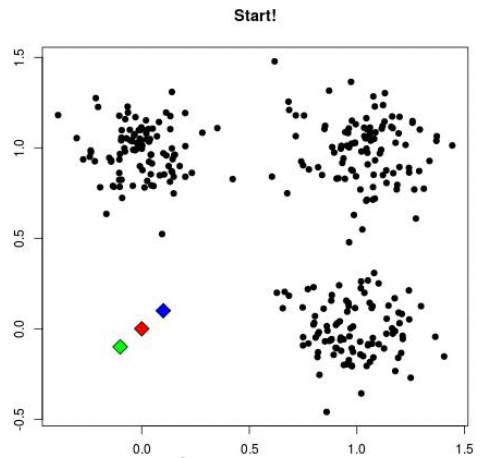
$$q_t^{*(2,10,3)} = -1 + 0.9 \max_{a_{t+1}}[<-1, \mathbf{0}, <-1, <-1] = \underline{-1};$$

$$q_t^{*(2,9,3)} = -1 + 0.9 \max_{a_{t+1}}[<-1, <-1, \mathbf{0}, <-1] = \underline{-1.9};$$

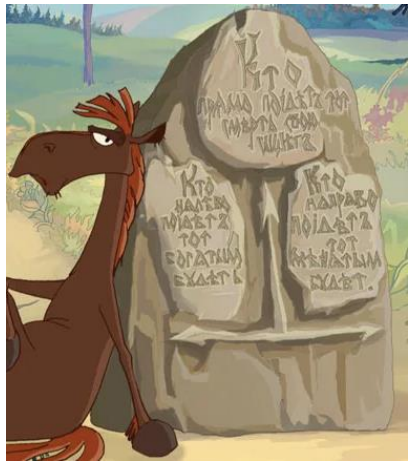
…

Лекция 3. Обучение без учителя и обучение с подкреплением / Unsupervised and Reinforcement Learning

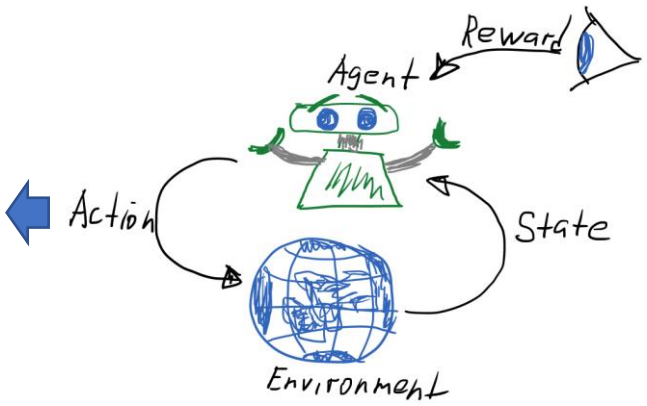
1. Напоминание / Contents of the previous lecture
2. Обучение без учителя: кластеризация / Unsupervised learning: clustering
3. Обучение с подкреплением: оптимизация поведения / Reinforcement learning: policy optimization
4. Обучение с подкреплением: оптимизация награды / Reinforcement learning: value optimization

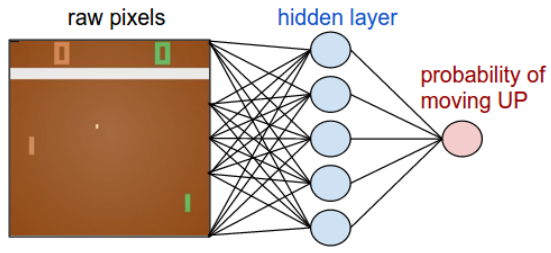


K-means clustering intuition



Q-table analogue





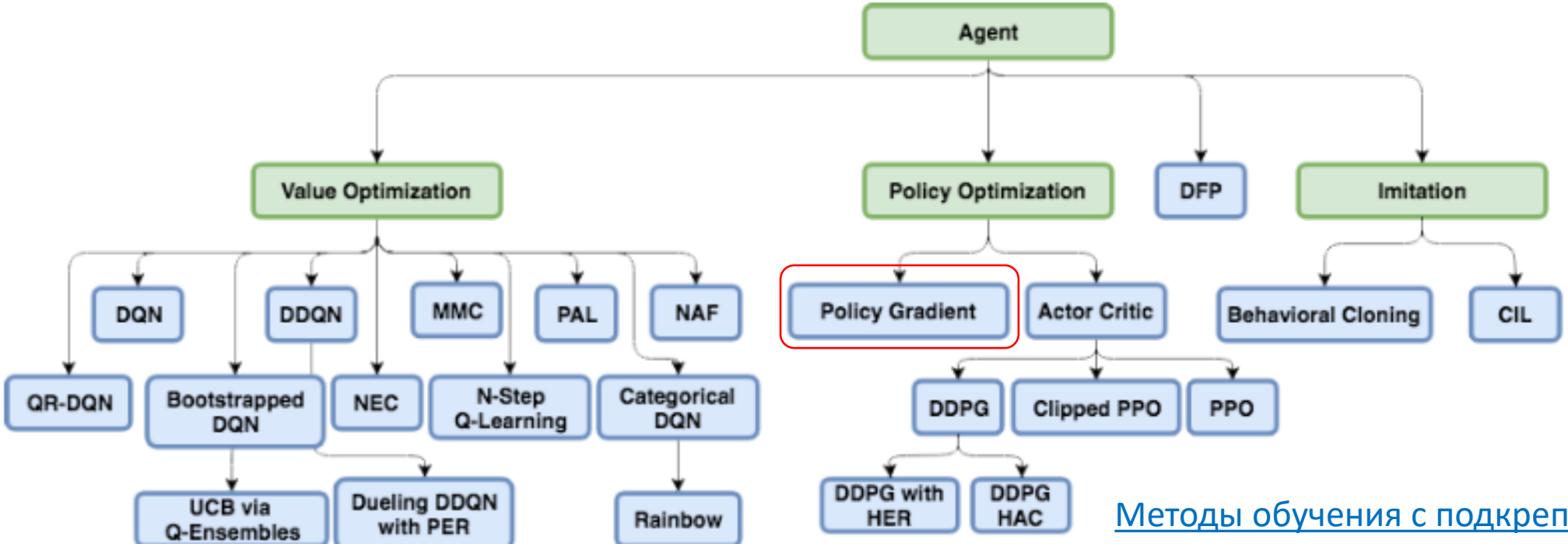
The Pong game

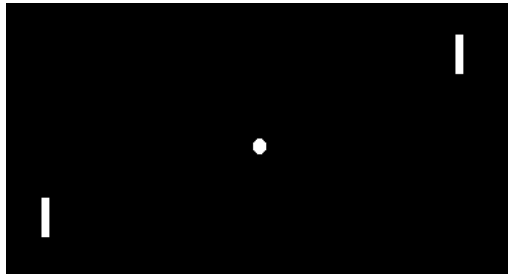
Оптимизация поведения / Policy optimization

Основная идея в поиске оптимальной функции вероятности действия (policy function) в каждом данном состоянии $p\left(A\middle|S,\Theta^{(k)}\right)$, которая максимизирует будущую награду (return):

$$g_t = \sum_{k=t}^{t^{\bowtie}} \gamma^{k-t} r_t ,$$

где γ – дисконт, $0 < \gamma \leq 1$, $t^{\bowtie}$- конечный шаг по времени (terminate state) или бесконечность.



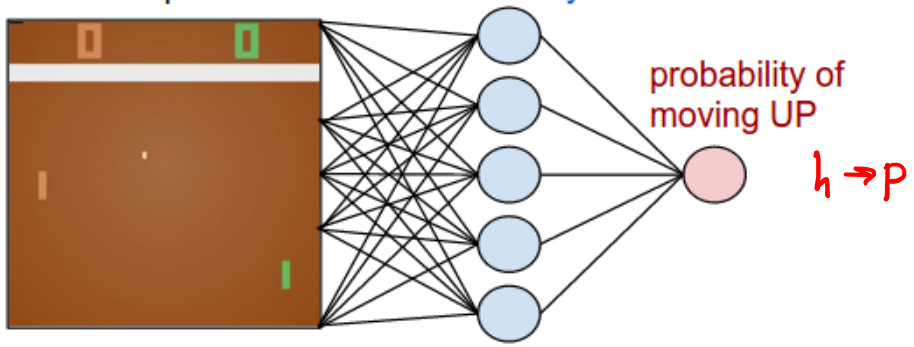


Методы обучения с подкреплением

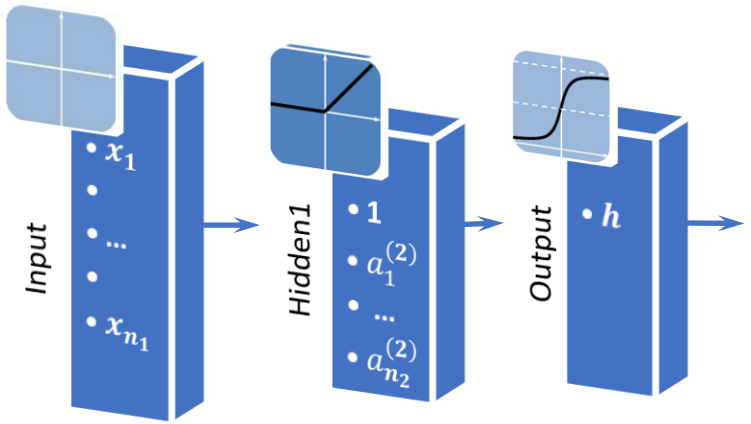
Оптимизация поведения / Policy optimization

Пример простейшей реализации алгоритма PG: задача игры в пинг-понг.
State: матрица S_t размером [210, 160, 3], составленная из значений цветов пикселей цветного изображения состояния игры (точнее разница матриц для двух соседних моментов времени / difference frames). **Action:** бинарный выбор движения ракетки $a_t = \{0,1\}$ вниз или вверх (0 – «DOWN», 1 – «UP»). **Reward:** положительная $r_t = +1$, если соперник пропустил мяч; отрицательная $r_t = -1$, если агент пропустил.

$S_t \rightarrow 0 < p_t < 1, a_t = \{0,1\}.$



Policy network is a 3-layer fully-connected net



Обучение с подкреплением / RL

$$L(\Theta^{(k)}) = \sum_{i=1}^{m} g_i \ln(p_i(a_i|S_i, \Theta^{(k)})) \Rightarrow max.$$

Напоминание. **Обучение с учителем**

$$X = ((x_1\ x_2 \dots x_{n_1})) \rightarrow 0 < h < 1,\ y = \{0,1\}.$$

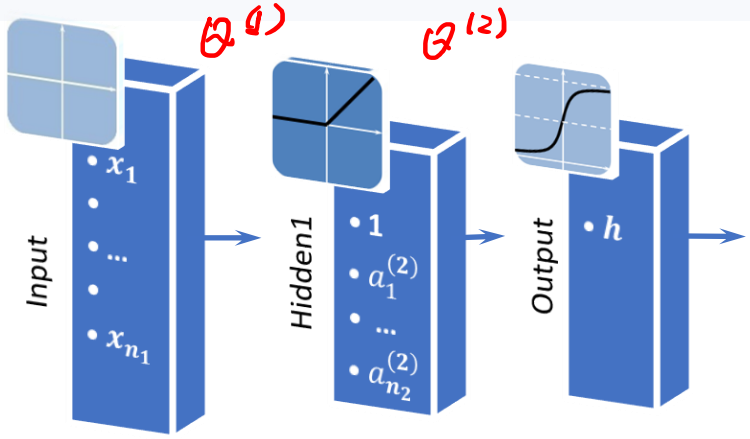
$$J(\Theta^{(k)}) = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} \ln(h^{(i)}) + (1-y^{(i)})(\ln(1-h^{(i)})\right) \Rightarrow \min.$$

$$L(\Theta^{(k)}) = -\sum_{i=1}^{m} y^{(i)} \ln(h^{(i)}) \Rightarrow \min, k = 1, \dots, l-1.$$

Andrej Karpathy blog: Pong from Pixels

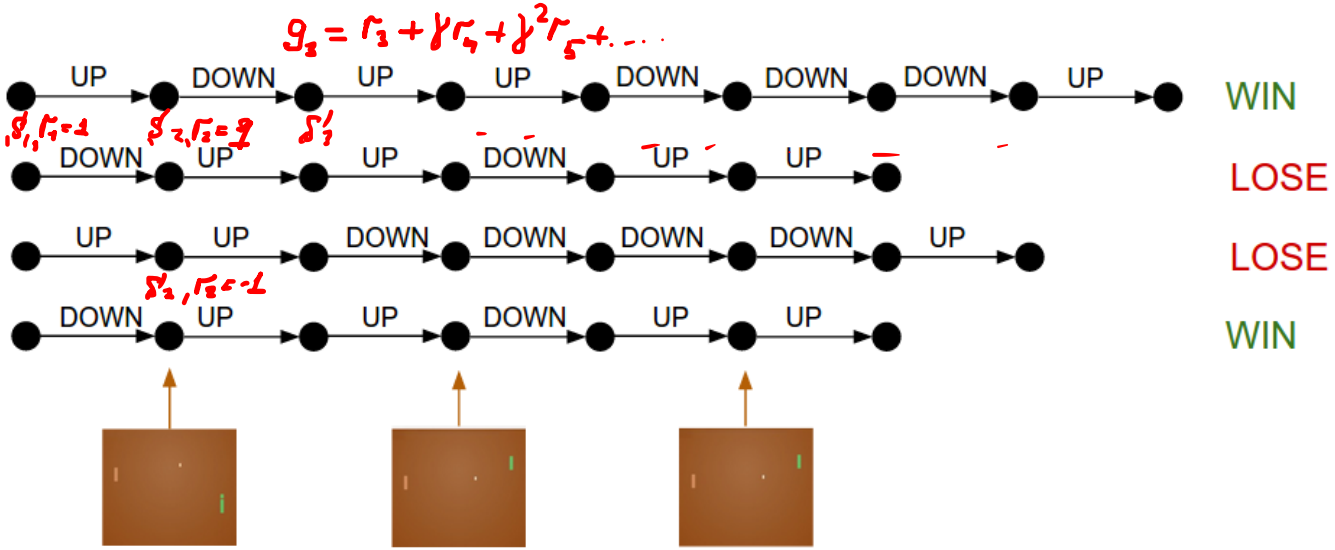
Оптимизация поведения / Policy optimization

Пример простейшей реализации алгоритма PG: задача игры в пинг-понг.
State: матрица S_t размером [210, 160, 3], составленная из значений цветов пикселей цветного изображения состояния игры (точнее разница матриц для двух соседних моментов времени / difference frames). **Action:** бинарный выбор движения ракетки $a_t = \{0,1\}$ вниз или вверх (0 – «DOWN», 1 – «UP»). **Reward:** положительная $r_t = +1$, если соперник пропустил мяч; отрицательная $r_t = -1$, если агент пропустил.



Алгоритм обучения / Training algorithm.

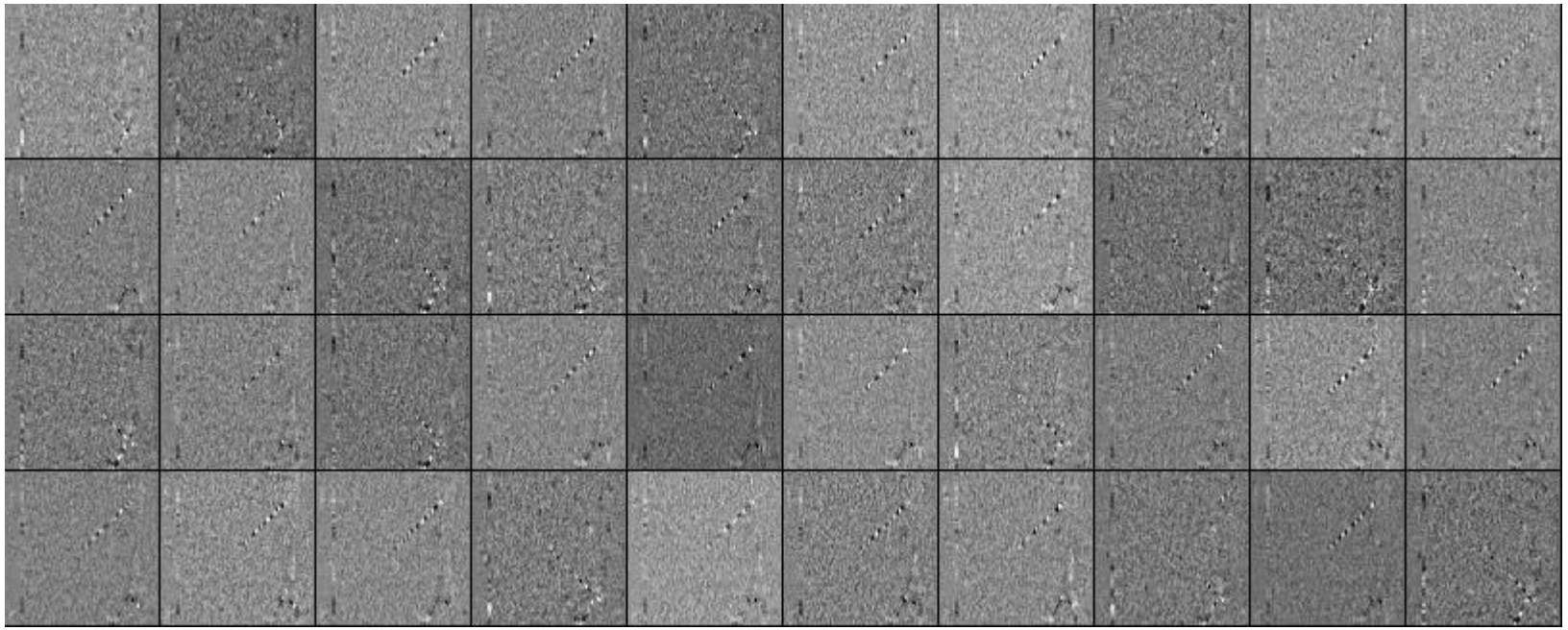
1. Случайным образом назначаются веса $\Theta^{(k)}$ ИНС.
2. Выполняется прогон (rollout) из 100 игровых эпизодов. Все действия выигранных эпизодов считаются правильными и поощряются наградой $r_t = +1$. И наоборот, для всех действий проигранных эпизодов награда $r_t = -1$.
3. Для каждого действия эпизода рассчитывается награда: $g_t = r_{t+j}\gamma^j$ $(j = 0,1,\dots)$.
4. Формируются данные (dataset) прогона: s_i, p_i, g_i.
5. Рассчитывается для прогона функция качества: $L\left(\Theta^{(k)}\right) = \sum_{i=1}^{m} g_i \ln(p_i)$.
6. Рассчитывается градиент $\nabla L\left(\Theta^{(k)}\right)$, затем новые значения весов: $\theta_{ij}^{(k)} = \theta_{ij}^{(k)} + \alpha \frac{\partial L}{\partial \theta_{ij}^{(k)}}$.
7. Повторяются пп. 2-6 до выполнения нек. условий.

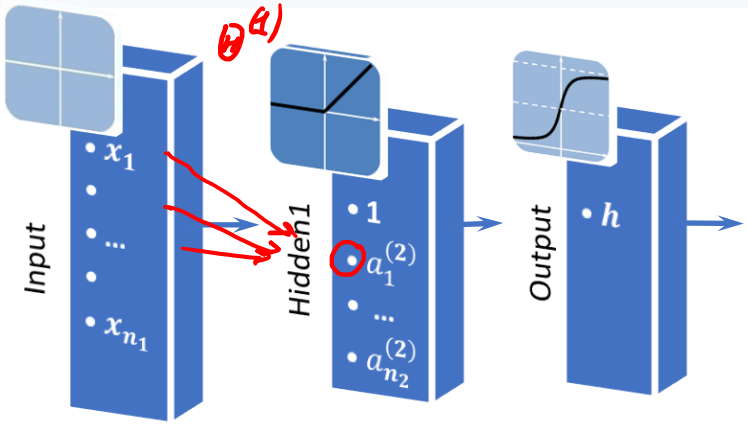


Andrej Karpathy blog: Pong from Pixels

Оптимизация поведения / Policy optimization

Пример простейшей реализации алгоритма PG: задача игры в пинг-понг.
State: матрица S_t размером [210, 160, 3], составленная из значений цветов пикселей цветного изображения состояния игры (точнее разница матриц для двух соседних моментов времени / difference frames). **Action:** бинарный выбор движения ракетки $a_t = \{0,1\}$ вниз или вверх (0 – «DOWN», 1 – «UP»). **Reward:** положительная $r_t = +1$, если соперник пропустил мяч; отрицательная $r_t = -1$, если агент пропустил.





Изображения весов 40 нейронов (из 200) скрытого слоя, которые визуализируют траекторию движения шарика. Белые пиксели означают положительные значения весов, черные – отрицательные.

Andrej Karpathy blog: Pong from Pixels

Полезные ссылки / Links

Онлайн курсы, обучающие ресурсы:

RL Course by David Silver: курс из 10 лекций Д. Силвера «Введение в обучение с подкреплением»

Stanford CS234: Reinforcement Learning: курс лекций Стэнфордского университета

Книги, статьи:

About RL please see Sutton and Barto (1998) or Bertsekas and Tsitsiklis (1996) for information about reinforcement learning, and Mnih et al. (2013) for the deep learning approach to reinforcement learning.

Elsevier , Springer : поисковые системы статей крупнейших издательств

SJR , WoS : поисковые системы журналов, рейтинг журналов

Самостоятельная работа / Homework

Вопросы.

1. Можно ли инициировать все веса ИНС нулями или единицами, а не случайными числами?
2. Имеет ли смысл выполнять процедуры валидации и тестирования при решении задач кластеризации методом к-ближайших соседей?
3. Можно ли в рассмотренной задаче обучения игры в пинг-понг вместо функции $L\left(\Theta^{(k)}\right) = \sum_{i=1}^{m} g_i \ln(p_i(a_i|s_i))$ использовать суммарную дисконтированную награду $L\left(\Theta^{(k)}\right) = \sum_{i=1}^{m} g_i$ в качестве функции качества обучения ИНС?
4. Почему трассировка шарика в картинках весов $\Theta^{(1)}$ (RL на примере игры в пинг-понг) отображаются пунктирными линиями, а не сплошными?
5. На чём, по вашему мнению, держится вера в приметы? Точно не на статистике!