OTUS

ОНЛАЙН-ОБРАЗОВАНИЕ

Глубокое обучение с подкреплением.

– А как же завтрак?
– Вы уже завтракали.
– А как же второй завтрак?

Артур Кадурин
Преподаватель

План на сегодня

1. **Q-learning**
2. Deep Q-networks
3. AlphaGo
4. IRL

TD-learning

Простейший алгоритм TD-обучения можно записать так:

- Инициализировать s;
- Для каждого шага в эпизоде:
  - выбрать a по стратегии π или по формуле для Q^*
  - сделать a и получить награду r и новое состояние s'
  - обновить $Q(s,a) \coloneqq Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$
  - перейти к новому состоянию s'

Что из себя представляет $Q(s,a)$?

TD-learning

Простейший алгоритм TD-обучения можно записать так:

- Инициализировать s;
- Для каждого шага в эпизоде:
  - выбрать a по стратегии π или по формуле для Q^*
  - сделать a и получить награду r и новое состояние s'
  - обновить $Q(s,a) := Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$
  - перейти к новому состоянию s'

Для задач типа игры в Крестики-нолики все возможные комбинации состояний и действий представляют из себя очень небольшую табличку, значения Q в которой мы можем итеративно обновлять. Но что делать если мы хотим решить более сложную задачу?

Как задавать $Q(s,a)$?

Q-learning

- Пусть теперь состояния $s \in S$ и действия $a \in A$ описываются набором признаков
- А функция Q является параметрической моделью машинного обучения $Q(s, a|\theta)$
- Примерами на которых мы будем обучать модель будут четверки $(s_t, a_t, r_{t+1}, s_{t+1})$
- И тогда, каждый шаг обучения мы можем в качестве предсказаний модели использовать значение $Q(s, a|\theta)$, а в качестве «верного» ответа: $r + \gamma \max_{a'} Q(s', a')$

Q-learning

- Пусть теперь состояния $s \in S$ и действия $a \in A$ описываются набором признаков
- А функция Q является параметрической моделью машинного обучения $Q(s, a|\theta)$
- Примерами на которых мы будем обучать модель будут четверки $(s_t, a_t, r_{t+1}, s_{t+1})$
- И тогда, каждый шаг обучения мы можем в качестве предсказаний модели использовать значение $Q(s, a|\theta)$, а в качестве «верного» ответа: $r + \gamma \max_{a'} Q(s', a')$

Такой подход с использованием нейронной сети в качестве Q-функции применяется уже более 25 лет. В 1992 году программа TD-Gammon играла в нарды на уровне людей-чемпионов.
Успех TD-Gammon обусловлен в том числе природой игры, действия в которой совершаются с помощью случайных бросков кубиков, что позволяет нейросети исследовать пространство возможных состояний естественным образом.

Однако, в следующие 20 лет ни каких больших прорывов в области обучения с подкреплением не произошло.

План на сегодня

1. Q-learning
2. **Deep Q-networks**
3. AlphaGo
4. IRL

Deep Q-Networks

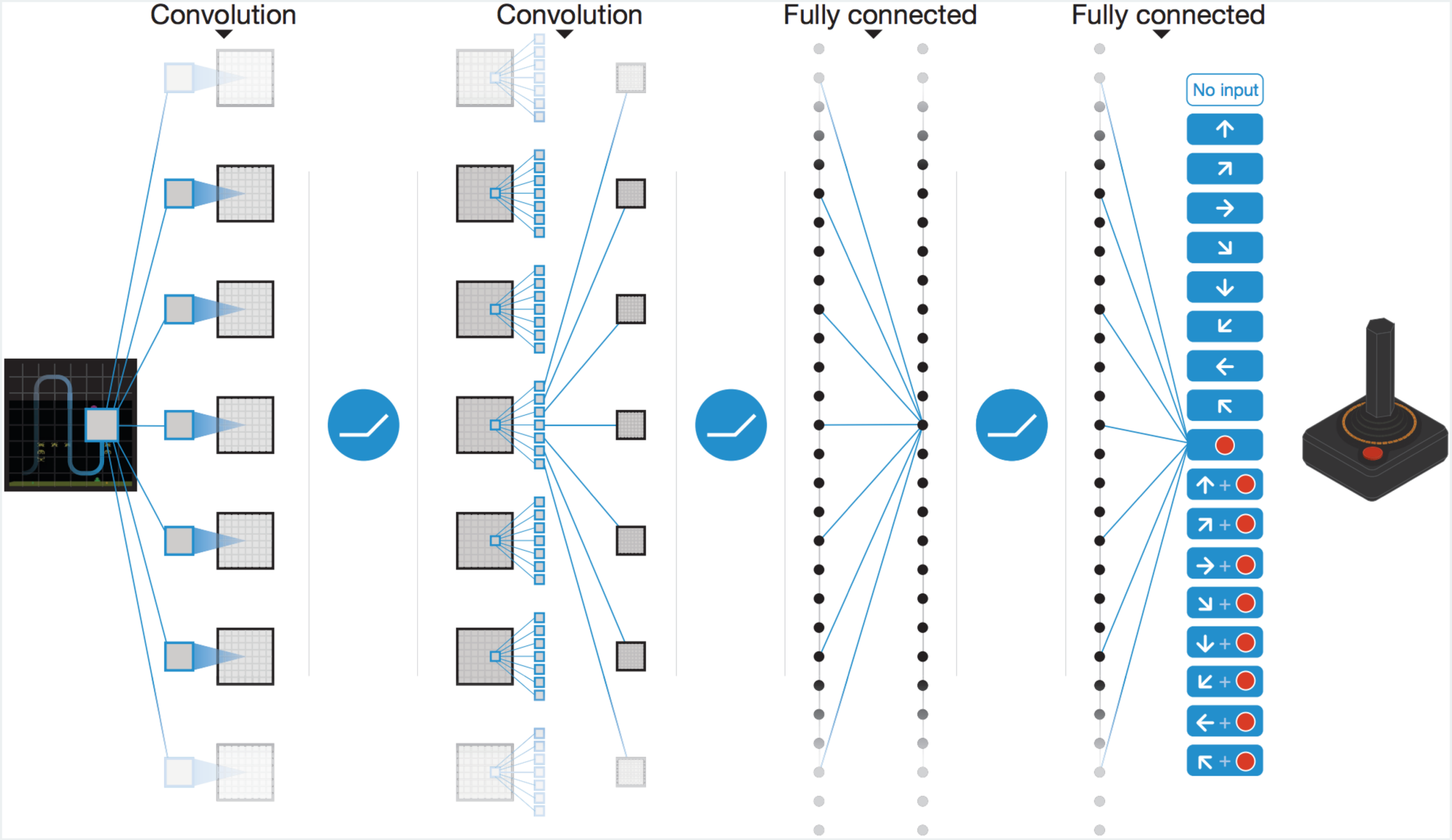
В 2013 году Мних с соавторами из DeepMind выложили на Arxiv препринт статьи «Playing Atari with Deep Reinforcement Learning» в которой показали как нейронная сеть способна научиться играть в компьютерные игры человеческим способом. В качестве описания состояния нейронная сеть получала текущее изображение, то же самое которое видел бы человек, а на выходе от нее ждали направление движения джойстика и нажатия кнопки.

Deep Q-Networks

В 2013 году Мних с соавторами из DeepMind выложили на Arxiv препринт статьи «Playing Atari with Deep Reinforcement Learning» в которой показали как нейронная сеть способна научиться играть в компьютерные игры человеческим способом. В качестве описания состояния нейронная сеть получала текущее изображение, то же самое которое видел бы человек, а на выходе от нее ждали направление движения джойстика и нажатия кнопки.

	B. Rider	Breakout	Enduro	Pong	Q*bert	Seaquest	S. Invaders
Random	354	1.2	0	−20.4	157	110	179
Sarsa [3]	996	5.2	129	−19	614	665	271
Contingency [4]	1743	6	159	−17	960	723	268
DQN	**4092**	**168**	**470**	**20**	**1952**	**1705**	**581**
Human	7456	31	368	−3	18900	28010	3690

Deep Q-Networks



Mnih, et al. Human-level control through deep reinforcement learning. Nature. 2015

Deep Q-Networks

Важные идеи из DQN:

1. Вместо того чтобы подавать действие на вход сети, мы просим сеть предсказать значение Q-функции сразу для всех возможных действий.

Deep Q-Networks

Важные идеи из DQN:

1. Вместо того чтобы подавать действие на вход сети, мы просим сеть предсказать значение Q-функции сразу для всех возможных действий.
2. Experience replay. Вместо того чтобы обучаться после каждого совершенного действия, сеть некоторое время играла в игру, накапливая «опыт», после чего обновлялась с помощью мини-батчей из «опыта».

Почему это важно?

Deep Q-Networks

Важные идеи из DQN:

1. Вместо того чтобы подавать действие на вход сети, мы просим сеть предсказать значение Q-функции сразу для всех возможных действий.
2. Experience replay. Вместо того чтобы обучаться после каждого совершенного действия, сеть некоторое время играла в игру, накапливая «опыт», после чего обновлялась с помощью мини-батчей из «опыта».
3. Разделение сети на две — обучающуюся и оценивающую: предсказание сети мы делаем с помощью весов которые постоянно обновляем, а вот состояние после совершения действия мы оцениваем с помощью зафиксированных на долгое время весов $r + \gamma \max_{a'} Q(s', a'|\theta_0)$. Раз в какое-то время, конечно, весам θ_0 присваиваются текущие значения весов сети.
4. Разделение Q-функции на 2 части: $Q(s, a) = V(s) + A(s, a)$

План на сегодня

1. Q-learning
2. Deep Q-networks
3. **AlphaGo**
4. IRL

AlphaGo

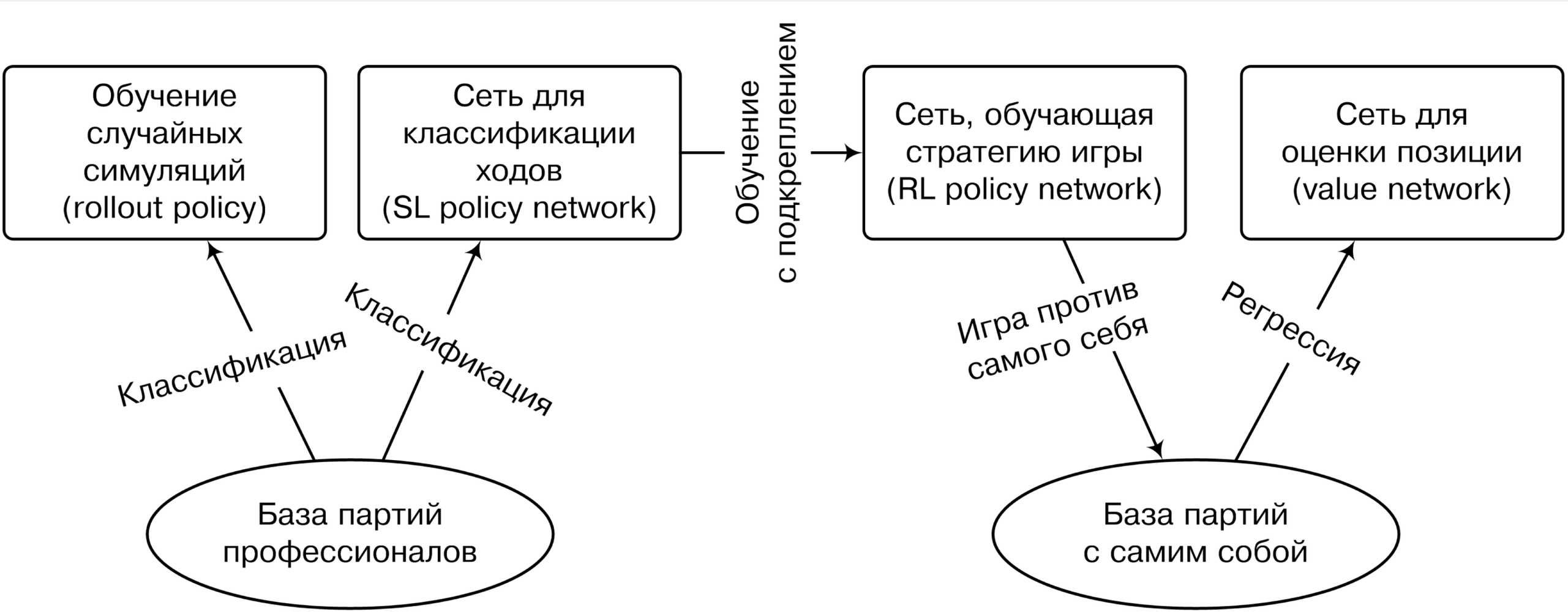
 Артур Кадурин, преподаватель

AlphaGo

OTUS

В AlphaGo используются несколько ключевых идей:

1. Policy Network: отдельная сеть для предсказания человеческого хода
2. Value Network: сеть для оценки позиции
3. Tree search: поиск по дереву возможных продолжений для оценки позиции



План на сегодня

1. Q-learning
2. Deep Q-networks
3. AlphaGo
4. **IRL**

Inversed RL

В задачах типа Atari или Go, награда задается естественным образом и нет ни какой магии в том чтобы использовать ее для обучения нейросети. Но представьте себе что вы хотите научиться управлять автомобилем, невозможно однозначно задать награду основанную на комбинации из безопасности, скорости, комфорта, расхода топлива, нарушения правил и т.д. За что вообще из всего этого давать награду и как балансировать между этими наградами?

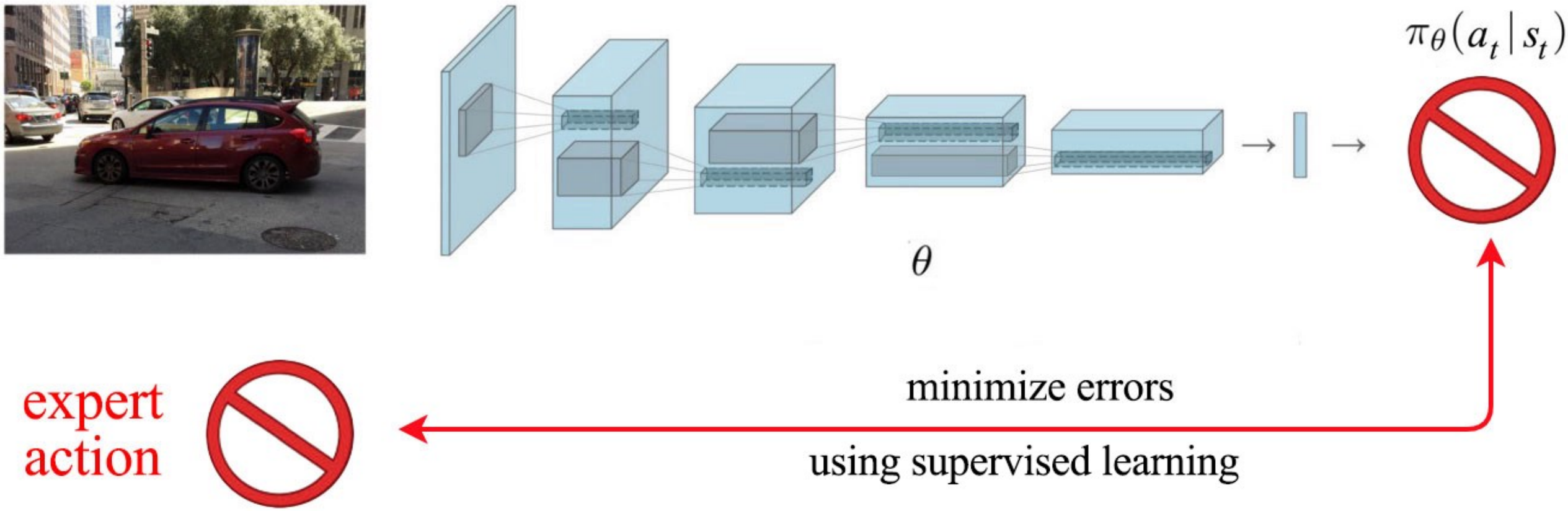
Суть обучения с подкреплением заключается в том, что если мы знаем награду, то мы можем выработать стратегию. Обратное обучение с подкреплением состоит в том, чтобы представляя себе оптимальную стратегию поведения определить функцию наград.

Inversed RL

В реальных задачах мы, конечно, не знаем даже оптимальной стратегии.
Но мы можем ее подглядеть!

Imitation Learning

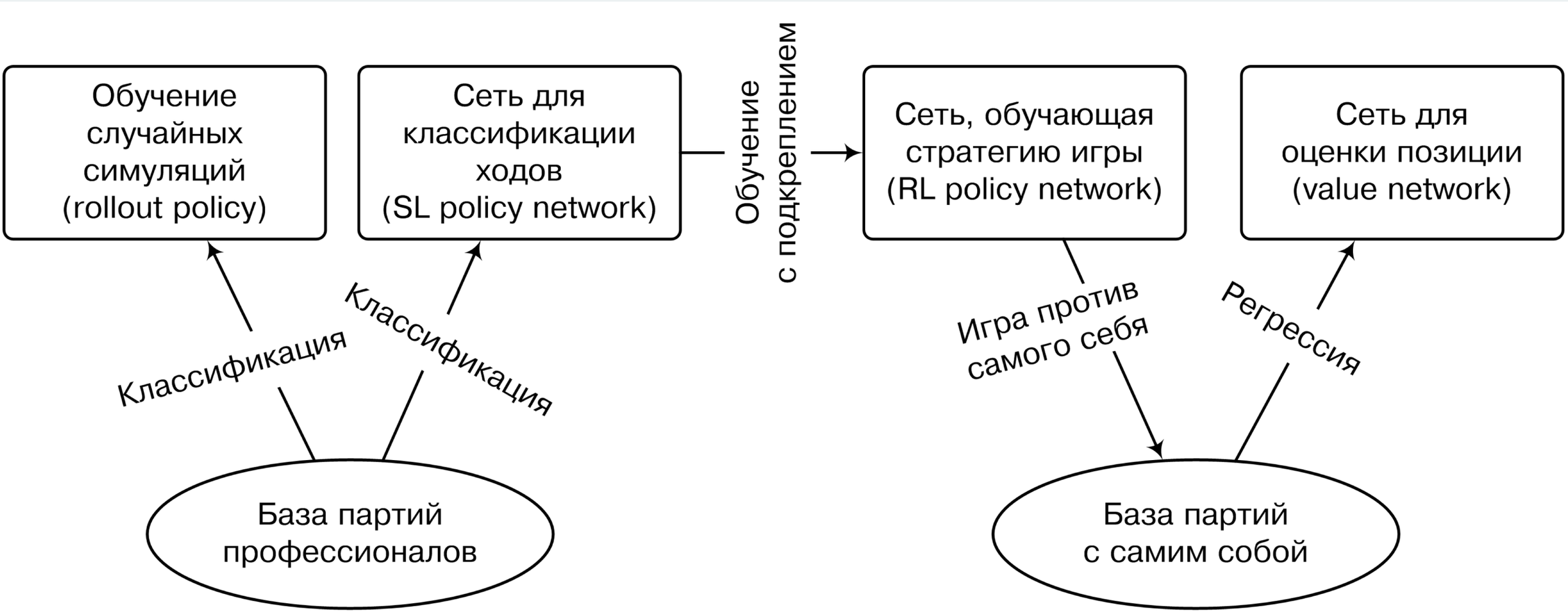
В реальных задачах мы, конечно, не знаем даже оптимальной стратегии.
Но мы можем ее подглядеть!



AlphaGo

OTUS

В AlphaGo объединились оба подхода. Одна сеть пытается имитировать поведение игроков-людей, а другая, основываясь на своей собственной игре учится оценивать позицию.



Заключение

1. Глубокое обучение с подкреплением основано на замене табличных функций их аппроксимациями с помощью нейросетей.
2. При обучении глубокой сети нужно пользоваться такими трюками как память и разделение во времени.
3. Иногда надо учиться не оценивать действие с точки зрения награды, а оценивать насколько вероятно такое действие совершит «мастер»

Спасибо
за внимание!