**1) Intro:**

научим компьютер играть в игру

для этого нам понадобится

написать игру и научить компьютер в нее играть

Пример: Google Deep Mind playing Atari Breakout

<https://www.youtube.com/watch?v=KkIf0Ok5GCE>

Лучше чем живой игрок - это наша цель не меньше)

Q: Кто уже видел этот ролик ?

Q: Кто пытался добится похожих результатов ?

Q: У кого получилось ? (не спойлирите))

Q: Кто занимался / пробовал ML ?

Q: На каких задачах ? Какой framework ?

**2) Всего грубо - три типа ML**

<https://www.analyticsvidhya.com/blog/2015/08/common-machine-learning-algorithms/>

**Supervised Learning** - [обучение с учителем](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81_%D1%83%D1%87%D0%B8%D1%82%D0%B5%D0%BB%D0%B5%D0%BC) (база с известными ответами)  например есть база с фото и текстом описание что на них - входы и выходы известны

Обучаем компьютер системе классификации, которую мы придумали

**Unsupervised Learning** - [обучение без учителя](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B1%D0%B5%D0%B7_%D1%83%D1%87%D0%B8%D1%82%D0%B5%D0%BB%D1%8F) (самообучение) есть база фото с разными объектами, мы хотим их категоризовать - например разложить по стопкам фото людей (как это делает iphone в альбомах)  кластеризация, ассоциация, дифференциация

<https://stackoverflow.com/questions/26182980/can-anyone-give-a-real-life-example-of-supervised-learning-and-unsupervised-lear>

**Reinforcement learning**  -  [Обучение с подкреплением](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81_%D0%BF%D0%BE%D0%B4%D0%BA%D1%80%D0%B5%D0%BF%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%D0%BC)

испытуемая система (агент) обучается, взаимодействуя с некоторой средой

to maximize the long-term benefits that the agent receives. good behaviour

чтобы поощряли и не наказывали

Q: Какая система обучения нам больше подходит?

**3) Open AI Gym**

Есть готовый framework "reinforcement learning",

где есть готовые игры из Atari и вся обвязка

<https://openai.com/>

<https://en.wikipedia.org/wiki/OpenAI>

Но мы не будем ее использовать именно по этой причине

нам интересно все сделать самим из базовых блоков -

наша игра + нейронная сеть + логика и т.п.

Q: Кто использовал Open AI Gym ?

**4) Игра**

Итак - нам надо написать игру и научить компьютер

в нее играть

критерии к игре - плоская, простая,

так как не хотим тратить много времени

будем писать на Python - это можно сказать язык ML

без рюшечек т.к. пишем для IA, а не для пользователей

Выбираем Snake, картинка игры snake

Иллюстрация игры - классика на Nokia

https://youtu.be/4kIUtQ1vsLY?t=1m0s

Кратко про правила игры - стенка, еда,

нельзя врезаться в себя, и т.п.

Q: Кто играл в такую игру на Нокия ? )

**5) Окружение**

HW: ПК + Nvidia GPU Card

OS: Desktop Ubuntu 16.06

ПО: TensorFlow x.x + Keras x.x + разные Python библиотеки

Сейчас вдаваться в подробности не будем, чтобы не тратить время

Настройка окружения - вот ссылки на репозиторий с Ansible скриптами

Все для того, чтобы воссоздать окружение самостоятельно

**6) Snake**

Игра для нашего друга AI

Поверьте мне это намного проще чем кажется

Должен признаться, что я получил большое

удовольствия от написания этой простой игры)

- когда-то давно я был разработчиком на ASM

и на ASM разработал игру в издательстве Питер:

[http://tv-games.ru/game/spectrum/Columns.html](http://tv-games.ru/game/spectrum/Columns.html" \t "_blank)

Игра писалась с рекламными целями Издательства

и свою задачу выполнила на 1000 %

Пришел мешок писем, всего из 11 городов.

Инфо об игре даже нашлось в [Википедии](http://www.worldofspectrum.org/infoseekid.cgi?id=0012787)

Не забываем что пишем для AI - как это

отражается на дизайне игры

Комментарии по коду игры

иллюстрация работы игры

Ограничения игры и параметры

Код игры можно найти в репозитории

**7) Q-Learning**

Что такое Q

Q[state,action] = immediate reward + discounted reward

Policy:

P(state) = argmax(i)(Q[state,action(i)])

Пример с шашками  и табличкой принятия решений.

место NN здесь - будет аппроксимировать нашу табличку

// тут много надо расписать

<https://keon.io/deep-q-learning/>

https://medium.freecodecamp.org/explained-simply-how-deepmind-taught-ai-to-play-video-games-9eb5f38c89ee

**8) Дисконтирование**

тут можно два слова про NPV и кредиты и банки

И то, что мы тут точно так же дисконтируем будущие

вознаграждения - чем дальше (по кол-ву ходов/времени) тем

больше скидка (и меньше влияние v.s. ближайшие шаги)

**9) Deep Q Network (DQN)**

Объединяем все вместе

Пояснения по коду

Простая Neural Network используя Keras

с двумя внутренними слоями

**10)Тестируем**

Время жизни, но счет никакой

Полный провал. Почему?

**11) Улучшение #1 "Окно"**

Мы смотрим на весь экран в целом

но в жизни не так - мы смотри

то, что ближе - определяет поведение.

Нам нужно уменьшить state до небольшой

области ("окна") вокруг головы snake.

Например 7x7, т.е. по 3 клетки в каждую сторону.

Время жизни увеличилось,

уже что-то осмысленное

**12) Улучшение #2 "Ротация"**

Когда играем - мы незаметно для себя

поворачиваем экран по направлению

движения - как это делает для нас

навигация в машине (если не делает,

например указан всегда север наверх,

мы это делаем автоматически за нее)

- картинка навигатора

Предположение - сети будет проще

учиться отворачивать от стены, и от себя,

если в состоянии всегда по ходу движения,

будет проще научится не сдавать назад,

так как "назад" всегда будет сзади

картинки различных "state" и с поворотом

для иллюстрации + action у нас преважается

вместо стрелок лево/право/вверх/вниз

в команды отоносительно направления движения

- вперед, поворот влево, поворот вправо, назад

**13) Улучшение #3 "Больше"**

Увеличиваем размер сети /кол-во нейронов

Помогает, но не очень

**14) Улучшение #4 "Convolution"**

Меняем сеть на конволюционную  / Convolution

???

**15) Улучшение #4 "Convolution"**

??