Deep Recurrent Q-Network
Plan

• Motivation
• Architecture
• Experiments
• DARQN
• Playing Doom using DRQN
What is Deep Q-Network?

It works good if observation contains enough information about state
Atari Pong

1 frame
Po-MDP
DQN works bad

4 frames
“almost MDP”
DQN works good
LSTM

It has internal memory that we need to solve POMDP problems
XP-replay

DQN
1 million transitions
\((S, a, r, S')\)

Select random transitions and train NN using TD-error
\[
\left( R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta^-) - Q(S_t, a; \theta_t^-) \right)^2
\]

DRQN
100000 sequences of tuples
\(10 \times (S, a, r)\)

Select random sequences of transitions and train NN using TD-error
\[
\left( R_{t+1} + \gamma Q(S_{t+1}, \text{argmax}_a Q(S_{t+1}, a; \theta_t), \theta^-) - Q(S_t, a; \theta_t^-) \right)^2
\]
Experiment with Pong
Playing doom using DRQN

Guillaume Lample, Devendra Singh Chaplot from Carnegie Mellon University

2nd place at ViZDoom AI competition

“Playing FPS Games with Deep Reinforcement Learning”
Doom

POMDP
Rewards

• positive reward for kills and negative rewards for suicides
• positive reward for object pickup (health, weapons and ammo)
• negative reward for loosing health (attacked by enemies or walking on lava)
• negative reward for shooting, or loosing ammo
Internal game features

- There is an enemy in this frame
- There is a ammo pack in this frame
- There is a health pack in this frame
- etc

Try to predict it using neural network immediately with the q-values.
\[ L_{rai} = L + L \]
Neural network updates
Two agents

1) Main ("action") DRQN-agent for shooting
2) Additional ("navigation") DQN-agent for moving

Algorithm:

- Call for main agent

- If there is an enemy in this frame, make a decision using main agent

- Otherwise use additional agent to make a decision about moving
Table 1: Comparison of human players with agent. Single player scenario is both humans and the agent playing against bots in separate games. Multiplayer scenario is agent and human playing against each other in the same game.
DARQN

Ivan Sorokin, Alexey Seleznev, Mikhail Pavlov, Aleksandr Fedorov, Anastasiia Ignateva

“Deep Attention Recurrent Q-Network”
“Soft” attention mechanism

\[ v_t = \{v_t^1, \ldots, v_t^L\}, \quad v_t^i \in \mathbb{R}^D \]

\[ g(v_t^i, h_{t-1}) = \exp(\text{Linear}(\text{Tanh}(\text{Linear}(v_t^i) + Wh_{t-1}))) / Z \]

\[ z_t = \sum_{i=1}^{L} g(v_t^i, h_{t-1}) v_t^i \]
Attention in action
Articles

-"Playing Atari with Deep Reinforcement Learning”
  https://arxiv.org/abs/1312.5602

-"Как понять LSTM”
  http://alexsosn.github.io/ml/2015/11/16/LSTM.html

-"Deep Recurrent Q-Learning for Partially Observable MDPs”
  http://arxiv.org/abs/1507.06527

-"Playing FPS Games with Deep Reinforcement Learning”
  https://arxiv.org/abs/1609.05521

-"Deep Attention Recurrent Q-Network”
  https://arxiv.org/abs/1512.01693
Thanks for your attention!
Questions?