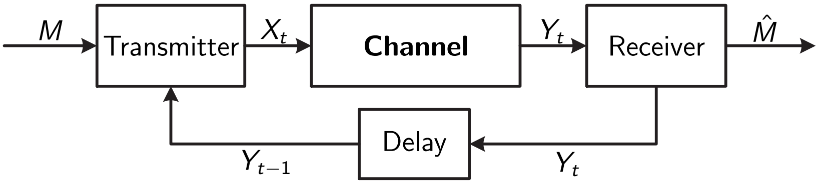
Computing the Feedback Capacity of Finite State Channels using Reinforcement Learning

Ziv Aharoni, Oron Sabag, Haim H. Permuter

Ben-Gurion University of the Negev

International Symposium on Information Theory
July 9th, 2019

Communication with Feedback



- Unifilar finite state channel (FSC):

$$p\left(y_t \middle| x_t, s_{t-1}\right)$$
$$s_t = f(x_t, y_t, s_{t-1})$$

- **The goal:** compute the capacity and coding scheme

The Capacity

Theorem (Permuter-Cuff-Van Roy-Weissman'08)

The **feedback capacity** *of unifilar FSC*

$$C_{fb} = \lim_{n\to\infty} \max_{\{p(x_i|s_{i-1},y^{i-1})\}_{i=1}^n} \frac{1}{n} I(X^n \to Y^n)$$

- The directed information (Massey 1990)

$$I(X^n \to Y^n) = \sum_{i=1}^{n} I(X^i; Y_i | Y^{i-1})$$

- This is a multi-letter expression

Markov Decision Process (MDP) Formulation

Theorem (Permuter-Cuff-Van Roy-Weissman'08)

The feedback capacity of a unifilar FSC can be formulated as a Markov decision process

Markov Decision Process (MDP) Formulation

Theorem (Permuter-Cuff-Van Roy-Weissman'08)

The feedback capacity of a unifilar FSC can be formulated as a Markov decision process

- MDP's state: $p(s_t|y^t)$

Markov Decision Process (MDP) Formulation

Theorem (Permuter-Cuff-Van Roy-Weissman'08)

The feedback capacity of a unifilar FSC can be formulated as a Markov decision process

- MDP's state: $p(s_t|y^t)$

Solution methods:

- Dynamic programming - value iteration algorithm
  Effective only for binary alphabet

Markov Decision Process (MDP) Formulation

Theorem (Permuter-Cuff-Van Roy-Weissman'08)

The feedback capacity of a unifilar FSC can be formulated as a Markov decision process

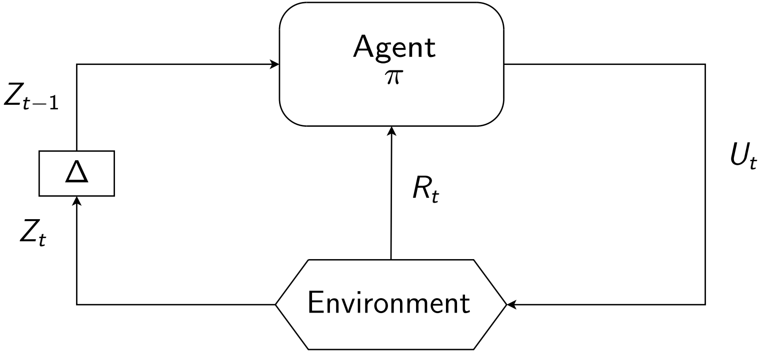
- MDP's state: $p(s_t|y^t)$

Solution methods:

- Dynamic programming - value iteration algorithm
  Effective only for binary alphabet

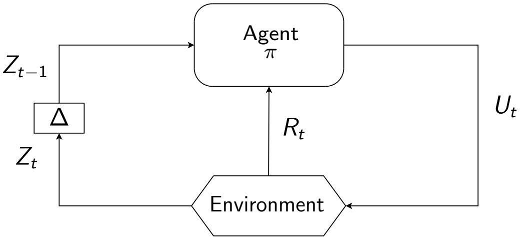
- Reinforcement learning (RL)
  Effective for large alphabets

Reinforcement Learning



- Z_{t-1} - current state
- U_t - action
- R_t - reward
- Z_t - next state

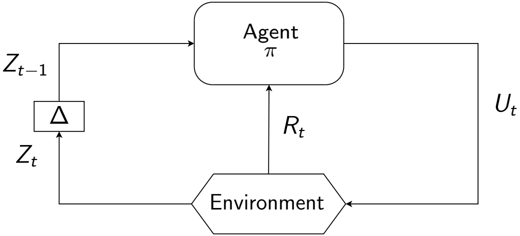
Reinforcement Learning



- The goal: maximize the expected average reward

$$\mathbb{E}_\pi[G] = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_\pi[R_t]$$

Reinforcement Learning



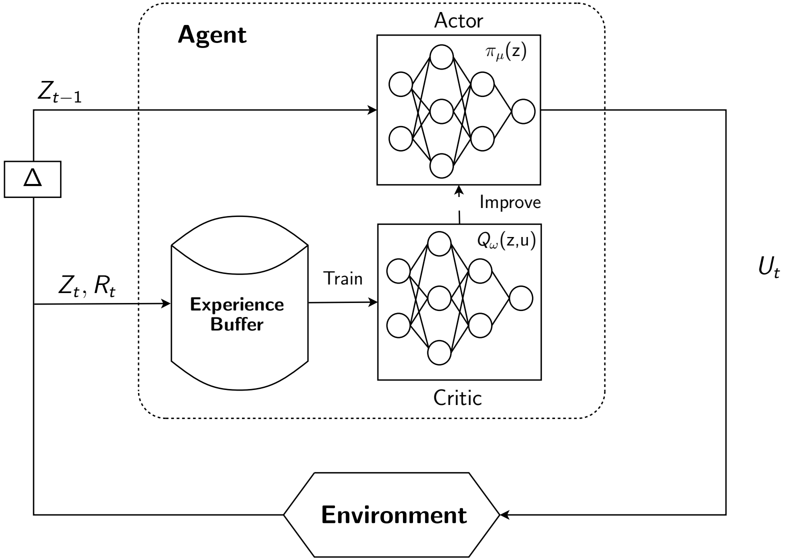
- The goal: maximize the expected average reward

$$\mathbb{E}_\pi [G] = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_\pi [R_t]$$

- The state-action value function

$$Q_\pi(z,u) = \mathbb{E}_\pi [G | Z_1 = z, U_1 = u]$$

Q-Learning Approach



The DDPG Algorithm

Deep Deterministic Policy Gradient, (Lillicrap et al.'16)

- Draw N interactions from experience (z_i, u_i, r_i, z_i')
- Train critic: minimize by ω

$$\frac{1}{N}\sum_{i=1}^{N}\left[Q_\omega\left(z_i, u_i\right) - \left[r_i - \rho_\mu + Q_\omega\left(z_i', \pi_\mu(z_i')\right)\right]\right]^2$$

- Improve actor: maximize by μ

$$\frac{1}{N}\sum_{i=1}^{N}\nabla_u Q_\omega\left(z_i, u\right)|_{u=\pi_\mu(z_i)}\nabla_\mu \pi_\mu\left(z_i\right)$$

The Ising Channel

- Defined by Berger and Bonomi (1990):

$$Y_i = \begin{cases} S_{i-1} & \text{, w.p. } 0.5 \\ X_i & \text{, w.p. } 0.5 \end{cases}, \quad S_{i-1} = X_{i-1}$$

- Models channel with ISI, magnetic recording

The Ising Channel

- Defined by Berger and Bonomi (1990):

$$Y_i = \begin{cases} S_{i-1} & \text{, w.p. } 0.5 \\ X_i & \text{, w.p. } 0.5 \end{cases}, \quad S_{i-1} = X_{i-1}$$

- Models channel with ISI, magnetic recording

- Solved the binary case (Elischo-Permuter'14, Sharov-Roth'16)

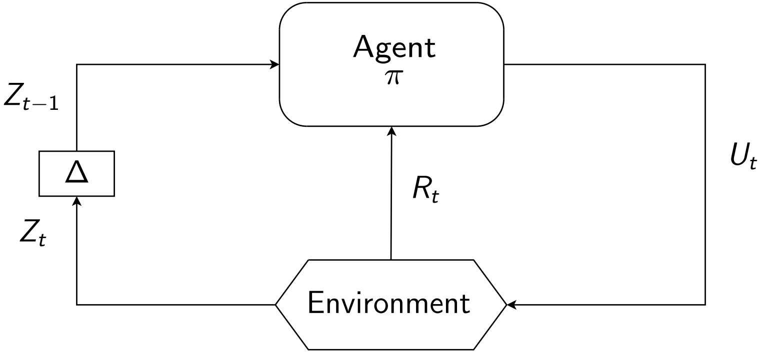
The Ising Channel

- Defined by Berger and Bonomi (1990):

$$Y_i = \begin{cases} S_{i-1} & \text{, w.p. } 0.5 \\ X_i & \text{, w.p. } 0.5 \end{cases}, \quad S_{i-1} = X_{i-1}$$

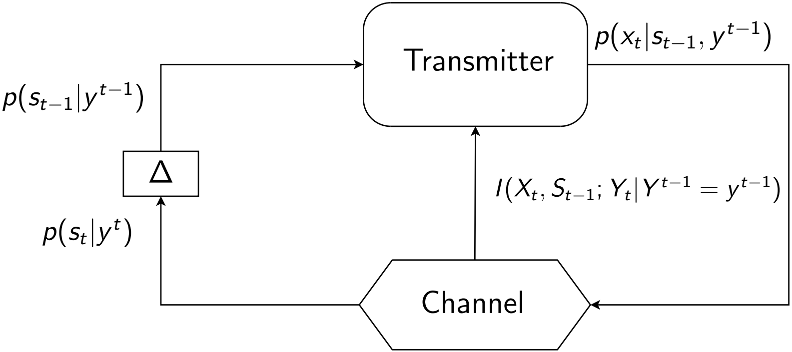
- Models channel with ISI, magnetic recording
- Solved the binary case (Elischo-Permuter'14, Sharov-Roth'16)
- **The goal:** apply RL methodology to larger alphabets

Back to the Feedback Capacity



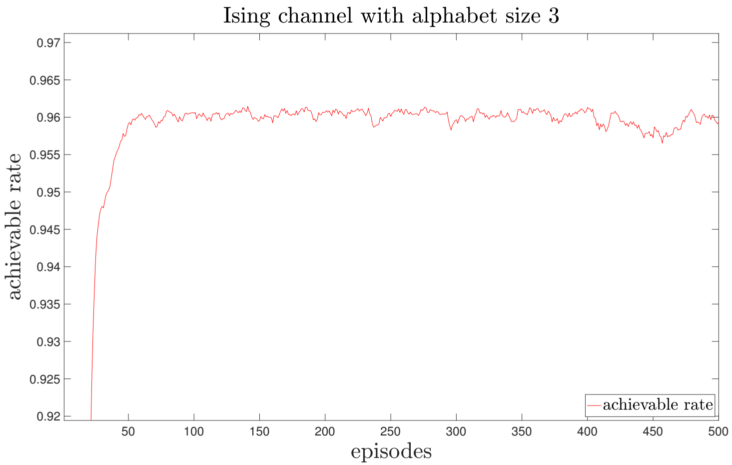
The goal: maximize **average reward**

Back to the Feedback Capacity



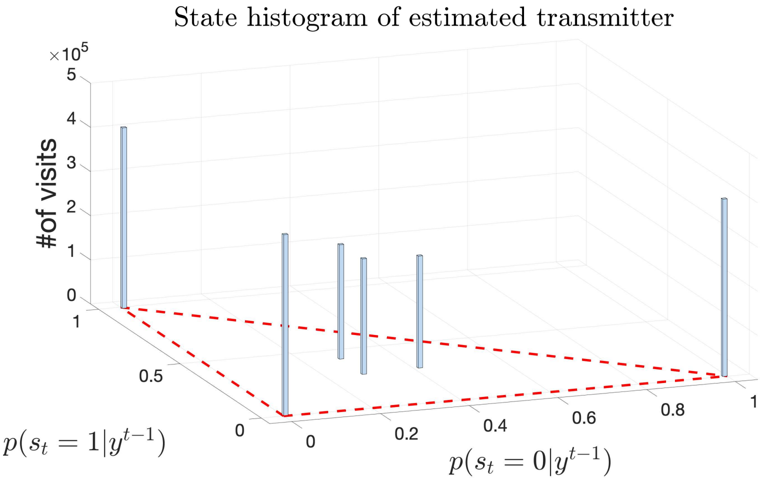
The goal: maximize **achievable rate**

Numerical results - Achievable Rate

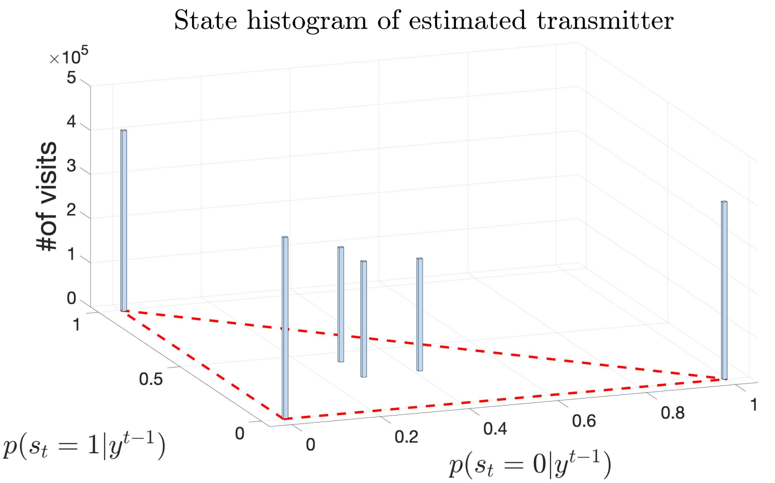


- Reveal the **structure** of the optimal solution

Properties of the Estimated Solution

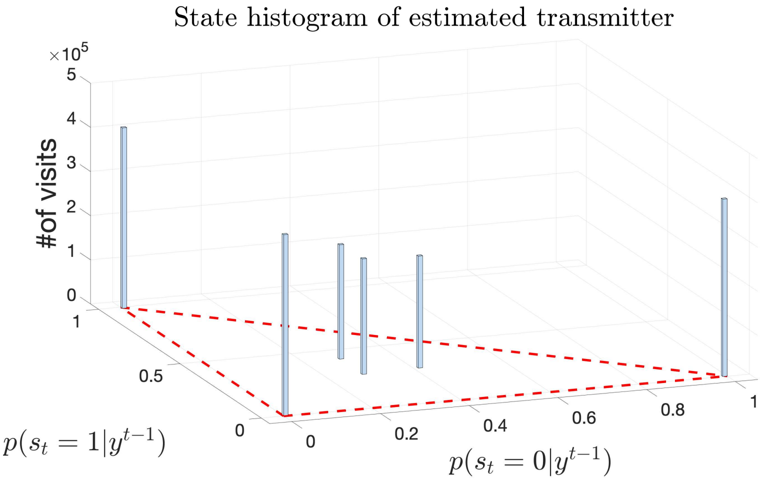


Properties of the Estimated Solution



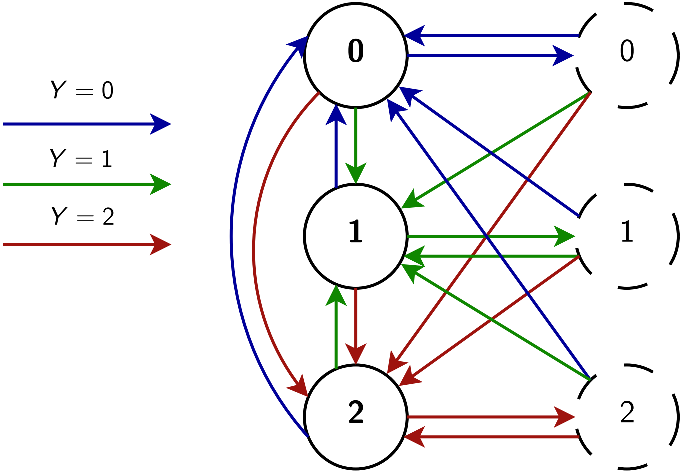
- Optimal input distribution structure

Properties of the Estimated Solution

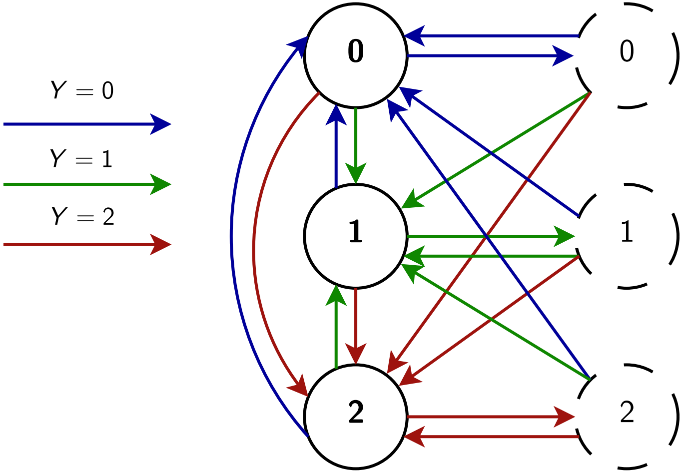


- Optimal input distribution structure
- Transitions between states as function of channel's output

Transitions of states by a Q-graph

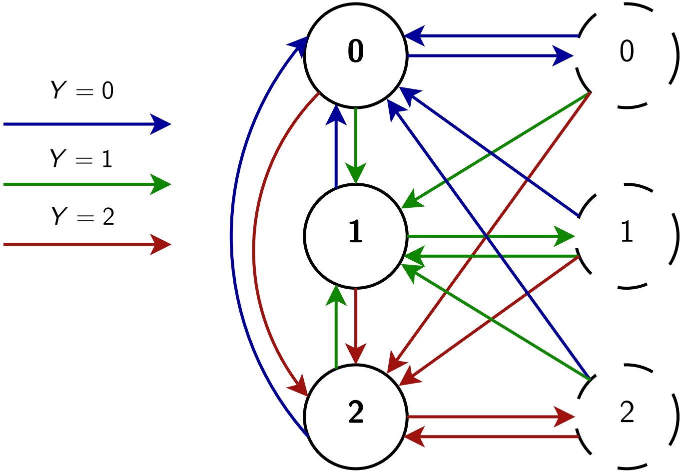


Transitions of states by a Q-graph



- Design coding scheme

Transitions of states by a Q-graph

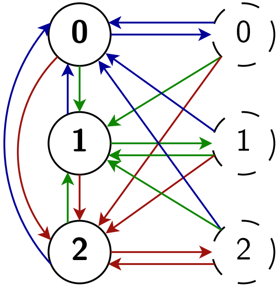


- Design coding scheme
- Prove upper-bound

Coding scheme

- **Pre-transmission:** generate an information sequence s.t.

$$x_i = \begin{cases} x_{i-1} & \text{, w.p } p \\ \text{Unif}[\mathcal{X} \backslash x_{i-1}] & \text{, w.p } 1-p \end{cases}$$

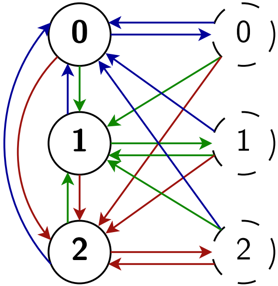


Coding scheme

- **Pre-transmission:** generate an information sequence s.t.

$$x_i = \begin{cases} x_{i-1} & \text{, w.p } p \\ \text{Unif}[\mathcal{X} \backslash x_{i-1}] & \text{, w.p } 1-p \end{cases}$$

  - At the beginning: assume decoder knows s

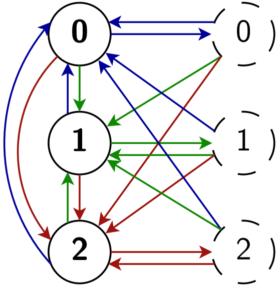


Coding scheme

- **Pre-transmission:** generate an information sequence s.t.

$$x_i = \begin{cases} x_{i-1} & \text{, w.p } p \\ \text{Unif}[\mathcal{X} \backslash x_{i-1}] & \text{, w.p } 1-p \end{cases}$$

  - At the beginning: assume decoder knows s
  - **Encoder:**
    Transmit x, if $y = s$, repeat x

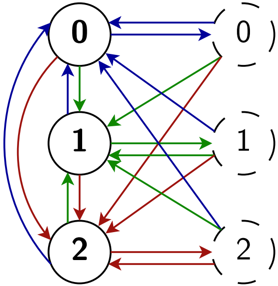


Coding scheme

- **Pre-transmission:** generate an information sequence s.t.

$$x_i = \begin{cases} x_{i-1} & \text{, w.p } p \\ \text{Unif}[\mathcal{X} \backslash x_{i-1}] & \text{, w.p } 1-p \end{cases}$$

  - At the beginning: assume decoder knows s
  - **Encoder:**
    Transmit x, if $y = s$, repeat x
  - **Decoder:**
    Receive y, if $y \neq s$ store y, else store next output

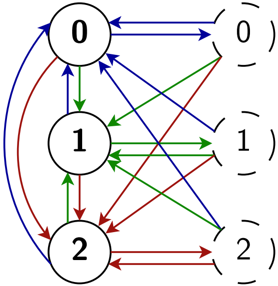


Coding scheme

- **Pre-transmission:** generate an information sequence s.t.

$$x_i = \begin{cases} x_{i-1} & \text{, w.p } p \\ \text{Unif}[\mathcal{X} \backslash x_{i-1}] & \text{, w.p } 1-p \end{cases}$$

  - At the beginning: assume decoder knows s
  - **Encoder:** Transmit x, if $y = s$, repeat x
  - **Decoder:** Receive y, if $y \neq s$ store y, else store next output



- **The rate of the scheme:**

$$R(\mathcal{X}) = \max_{p\in[0,1]} 2\frac{H_2(p) + (1-p)\log(|\mathcal{X}|-1)}{p+3}$$

Upper-bound

Theorem (Sabag-Permuter-Pfister'17)

For any choice of Q-graph

$$C_{fb} \leq \max_{p(x|s,q)\in\mathcal{P}_\pi} I(X,S;Y|Q)$$

Upper-bound

Theorem (Sabag-Permuter-Pfister'17)

For any choice of Q-graph

$$C_{fb} \leq \max_{p(x|s,q)\in\mathcal{P}_\pi} I(X,S;Y|Q)$$

Theorem (Duality bound)

For any FSC channel and $T_{Y|Q}$

$$C_{fb} \leq \lim_{n\to\infty} \max_{f(x^n\|y^{n-1})} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[D\left(P_{Y|X=X_i,X^-=X_{i-1}} \| T_{Y|Q=Q_{i-1}}\right)\right]$$

The feedback capacity

Theorem

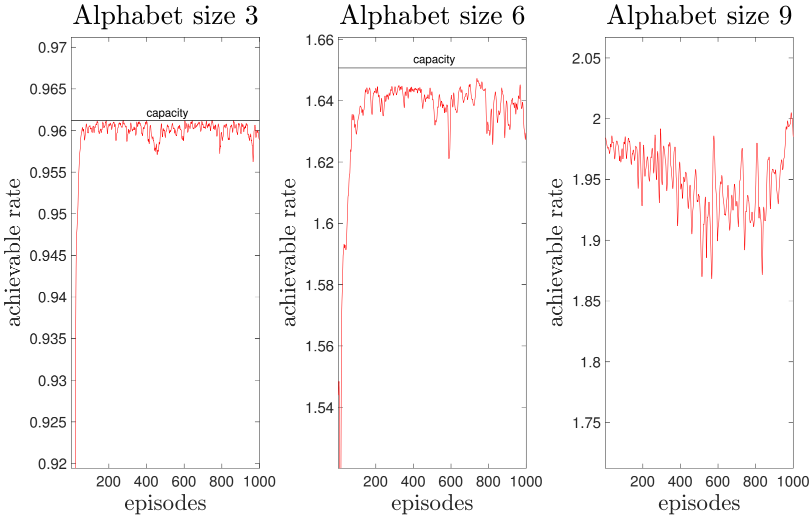
For all $|\mathcal{X}| \leq 8$, the feedback capacity of the Ising channel is given by

$$C_{fb}(\mathcal{X}) = max_{p\in[0,1]} 2^{\frac{H_2(p)+(1-p)\log(|\mathcal{X}|-1)}{p+3}}$$

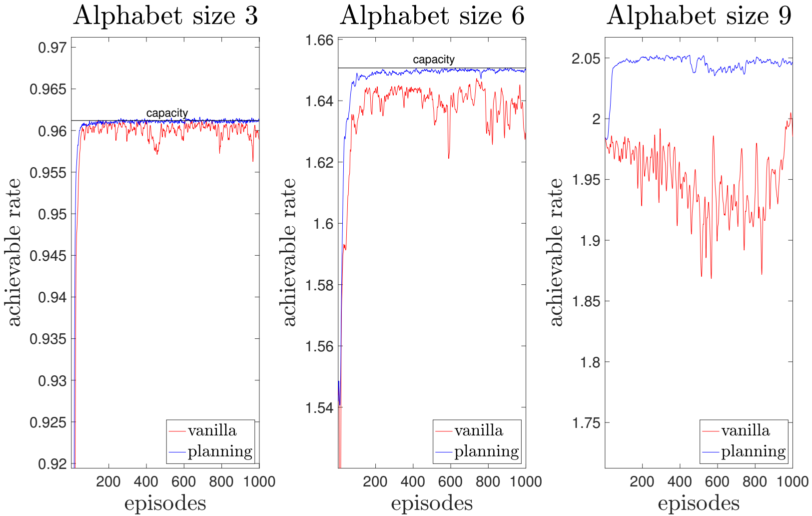
- What happens for $|\mathcal{X}| \geq 9$?
- Asymptotic better rate

$$R(\mathcal{X}) = \frac{3}{4}\log\frac{|\mathcal{X}|}{2}$$

Improving RL: DDPG with planning



Improving RL: DDPG with planning



Conclusions

- RL methodology for analytic feedback capacity of unifilar FSC
- Analytic solution to improve RL

Conclusions

- RL methodology for analytic feedback capacity of unifilar FSC
- Analytic solution to improve RL

Future Work:

- What is the solution for $|\mathcal{X}| \geq 9$?

Conclusions

- RL methodology for analytic feedback capacity of unifilar FSC
- Analytic solution to improve RL

Future Work:

- What is the solution for $|\mathcal{X}| \geq 9$?
- Solve more channels to constitute an IT benchmark for RL algorithms

Conclusions

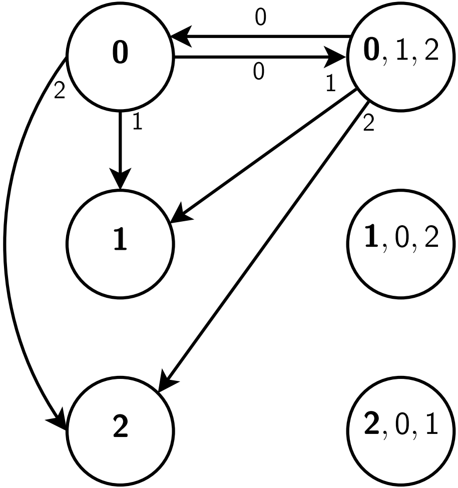
- RL methodology for analytic feedback capacity of unifilar FSC
- Analytic solution to improve RL

Future Work:

- What is the solution for $|\mathcal{X}| \geq 9$?
- Solve more channels to constitute an IT benchmark for RL algorithms

Thank You!

Transitions of states by a directed graph



The DDPG Algorithm

Deep Deterministic Policy Gradient, (Lillicrap et al.'16)

- Draw N interactions from experience (z_i, u_i, r_i, z_i')
- Train critic: minimize by ω

$$\frac{1}{N}\sum_{i=1}^{N}\left[Q_\omega(z_i,u_i) - \left[r_i - \rho_\mu + \sum_{z'} p(z'|z_i,u_i)Q_\omega(z',\pi_\mu(z'))\right]\right]^2$$

- Improve actor: maximize by μ

$$\frac{1}{N}\sum_{i=1}^{N}\nabla_u Q_\omega(z_i,u)\,|_{u=\pi_\mu(z_i)}\nabla_\mu \pi_\mu(z_i)$$