Computing the Feedback Capacity of Finite State Channels using Reinforcement Learning

Ziv Aharoni
Ben-Gurion University of the Negev
zivah@post.bgu.ac.il

Oron Sabag
Ben-Gurion University of the Negev
oronsa@post.bgu.ac.il

Haim H. Permuter
Ben-Gurion University of the Negev
haimp@bgu.ac.il

***Abstract*—In this paper, we propose a novel method to compute the feedback capacity of channels with memory using reinforcement learning (RL). In RL, one seeks to maximize cumulative rewards collected in a sequential decision-making environment. This is done by collecting samples of the underlying environment and using them to learn the optimal decision rule. The main advantage of this approach is its computational efficiency, even in high dimensional problems. Hence, RL can be used to estimate numerically the feedback capacity of unifilar finite state channels (FSCs) with large alphabet size. The outcome of the RL algorithm sheds light on the properties of the optimal decision rule, which in our case, is the optimal input distribution of the channel. These insights can be converted into analytic, single-letter capacity expressions by solving corresponding lower and upper bounds. We demonstrate the efficiency of this method by analytically solving the feedback capacity of the well-known Ising channel with a ternary alphabet. We also provide a simple coding scheme that achieves the feedback capacity.**

I. Introduction

Computing the capacity of a finite state channel (FSC) is a difficult task that has been vigorously researched in recent decades [1]. With the presence of feedback, the feedback capacity of a FSC can be expressed using the directed information [2], [3]. Despite the fact that the directed information is a multi-letter expression, it was shown that it can be formulated as a Markov decision process (MDP), which enables its computability using known MDP algorithms [4].

When formulated as a MDP, the feedback capacity of a FSC can be computed using a variety of methods, such as value and policy iteration. These algorithms have been proven very effective for channels with relatively small alphabets of the channel input, output and state [4]–[10]. However, a principal drawback is that their computational complexity grows with the cardinality of the channel alphabet. Indeed, even for channel parameters from the ternary alphabet, these algorithms might be intractable.

We propose a machine learning (ML) approach to compute the capacity of such channels. ML has been proven to be a useful tool with a great impact in many research fields. One example in communications is [11], wherein a learning-based algorithm was applied to design a reliable code for the additive white Gaussian noise channel with feedback. The present work introduces a new role of ML in communications, an efficient computation of multi-letter capacity expressions using RL algorithms.

We propose a methodology that uses RL to compute the feedback capacity of unifilar FSCs. Initially, a RL algorithm, namely the deep deterministic policy gradient (DDPG), is used to numerically estimate the feedback capacity. Then, the outcome of the RL algorithm is used to conjecture the structure of the analytic solution, which is expressed by a directed graph. The conjectured graph, that is called a *Q-graph*, can be used to compute analytic lower and upper bounds of the feedback capacity [12]. The bounds are guaranteed to coincide to the feedback capacity, in the case that the Q-graph of the analytic solution is extracted. Furthermore, the Q-graph can be used to derive a simple, capacity-achieving coding scheme of the channel. In our work, the proposed methodology enabled us to compute the feedback capacity of the Ising channel with a ternary alphabet (Ising3), and derive a capacity achieving coding scheme.

The remainder of the paper is organized as follows. Section II includes the notation and preliminaries. In Section III, we present our main results. Section IV provides background on RL and on the DDPG algorithm. In Section V, we estimate the feedback-capacity of the Ising3 using the DDPG algorithm. In Section VI, we prove the feedback-capacity of the Ising3 channel and present a simple capacity-achieving coding scheme. Section VII contains conclusions and a discussion of the future work.

II. Notation and Problem Definition

A. Notation

Calligraphic letters, $\mathcal{X}$, denote alphabet sets, upper-case letters, X, denote random variables, and lower-case letters, x, denote sample values. A superscript, x^t, denotes the vector $(x_1, \ldots, x_t)$. The probability distribution of a random variable, X, is denoted by p_X. We omit the subscript of the random variable when it and the argument have the same letter, e.g. $p(x|y) = p_{X|Y}(x|y)$. The binary entropy is denoted by $H_2(\cdot)$

B. Unifilar state channels

A FSC is defined by the triplet $(\mathcal{X} \times \mathcal{S}, p(y, s'|x, s), \mathcal{Y} \times \mathcal{S})$, where X is the channel input, Y is the channel output, S is the channel state at the beginning of the transmission, and S' is the channel state at the end of the transmission, where the

cardinalities $\mathcal{X}, \mathcal{Y}, \mathcal{S}$, are assumed to be finite. At each time t, the channel has the memory-less property

$$p(s_t, y_t|x^t, s^{t-1}, y^{t-1}) = p(y_t|x_t, s_{t-1})p(s_t|x_t, s_{t-1}, y_t). \tag{1}$$

A FSC is called *unifilar* if the new channel state, s_t, is a time-invariant function of the triplet $s_t = f(x_t, y_t, s_{t-1})$. For a FSC with feedback, the input x_t is determined by the message and the feedback tuple y^{t-1}.

The feedback capacity of a unifilar FSC is given by a multi-letter expression that is presented in the following theorem.

Theorem 1. *[4, Thm 1] The feedback capacity of a strongly connected unifilar state channel, where the initial state s_0 is available to both to the encoder and the decoder, can be expressed by*

$$C_{fb} = \lim_{N\to\infty} \sup_{\{p(x_t|s_{t-1},y^{t-1})\}_{t=1}^{N}} \frac{1}{N}\sum_{i=1}^{N} I(X_i, S_{i-1}; Y_i|Y^{i-1}).$$

C. Ising3 channel

The Ising channel model was introduced as an information theory problem by Berger and Bonomi in 1990 [13], 70 years after it was introduced as a problem in statistical mechanics by Lenz and his student, Ernst Ising [14]. Berger and Bonomi studied the channel with a binary alphabet size. We investigate a generalized version of the Ising channel, where the alphabets are not necessarily binary. The Ising channel is defined by

$$Y = \begin{cases} X & \text{, w.p } 0.5 \\ S & \text{, w.p } 0.5 \end{cases}, \tag{2}$$

$$S' = X. \tag{3}$$

Hence, if $X = S$ then $Y = X = S$ w.p 1. Otherwise, Y is assigned by one of the last two symbols with equal probability.

III. Main Results

The following theorems constitute our main results.

Theorem 2. *The feedback capacity of a unifilar FSC can be estimated by a RL algorithm.*

Remark 1. Theorem 2 is a computational result. Specifically, while previous estimations of the capacity were constrained by the cardinality of the channel parameters, we show that the RL algorithm is dimensional free.

Using the numerical results from the RL algorithm, one can deduce the analytic solution structure by a Q-graph [12], which is used to compute the feedback capacity.

The following theorem is an instance of a known channel that we were able to solve using the numerical results from the RL algorithm.

Theorem 3. *The feedback-capacity of the Ising3 channel is given by*

$$C_{fb} = \max_{p\in[0,1]} 2\frac{H_2(p) + 1 - p}{p+3}, \tag{4}$$

where $C_{fb} \approx 0.961227$ for $p \approx 0.263805$.

Furthermore, we derive a simple coding scheme that achieves the feedback capacity in Theorem 3.

Theorem 4. *There exists a simple coding scheme for the Ising channel with general alphabet $\mathcal{X}$, with the following achievable rate:*

$$R(\mathcal{X}) = \max_{p\in[0,1]} 2\frac{H_2(p) + (1-p)\log(|\mathcal{X}|-1)}{p+3}. \tag{5}$$

Note that for $|\mathcal{X}| = 3$, the coding scheme achieves the capacity in Theorem 3.

The coding scheme is described by a repeated procedure that is given by the following:

Code construction and initialization:

- The message is a stream of n uniform bits.
- Transform the message into a stream of symbols from $\mathcal{X}$, denoted by $\nu_1\nu_2\ldots$ with the following statistics:

$$\nu_i = \begin{cases} \nu_{i-1} & \text{, w.p } p \\ \text{Unif}[\mathcal{X}\backslash\nu_{i-1}] & \text{, w.p } 1-p \end{cases} \tag{6}$$

  In words, a new symbol equals the previous symbol with probability p and, otherwise, it is randomly chosen from the remaining symbols. This mapping can be done, for instance, by using enumerative coding, as shown in [15].
- At the first time, the encoder transmits ν_1 twice.
- The decoder, upon receiving y_1, y_2, decode $\hat{\nu}_1 = y_2$ and sets $c = 2$.

The transmission procedure is given by the following:

Encoder:

1) If $\nu_t = \nu_{t-1}$ transmit ν_t twice and move to the next symbol.
2) If $\nu_t \neq \nu_{t-1}$ transmit ν_t once and view the last feedback y.
   a) If $y = \nu_t$ move to the next symbol.
   b) If $y \neq \nu_t$ transmit ν_t again and move to the next symbol.

Decoder:

1) If $y_t \neq \hat{\nu}_{c-1}$ then $\hat{\nu}_c = y_t$, increment $c = c + 1$.
2) If $y_t = \hat{\nu}_{c-1}$ then wait for y_{t+1}, set $\hat{\nu}_c = y_{t+1}$, and increment $c = c + 1$.

In Section VI, we prove that the coding scheme yields a zero-error code and that its maximum rate equals the feedback capacity as given in Thm. 3.

IV. Reinforcement Learning

In this section, we provide the definition of the basic RL problem setting as presented in [16] and elaborate on the DDPG algorithm.

A. Background

The RL field in ML comprises an agent that interacts with an unknown environment by taking sequential actions. Formally, at time t, the agent observes the environment's state z_{t-1} and then takes an action $u_t = A(z_{t-1})$. This incurs an immediate reward r_t and the agent's next state z_t, as shown in Fig. 1.

The environment is assumed to satisfy the Markov property,

$$p\left(z_t, r_t | z^{t-1}, u^t, r^{t-1}\right) = p\left(z_t, r_t | z_{t-1}, u_t\right). \quad (7)$$

Hence, it can be defined by the conditional probabilities $p\left(z_t|z_{t-1}, u_t\right), p\left(r_t|z_{t-1}, u_t\right)$[1].

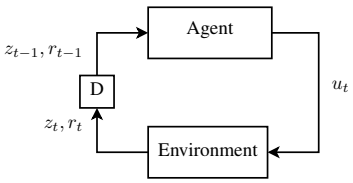

Fig. 1. Depiction of the agent-environment interface in RL. The agent observes the environment state and chooses an action. In return, the environment draws an immediate reward and a next state according to $p\left(r_t|z_{t-1}, u_t\right)$, $p\left(z_t|z_{t-1}, u_t\right)$.

The agent's policy is a sequence of actions $\pi = \{u_1, u_2, \dots\}$, and the cumulative rewards with respect to the policy from time t onward are defined by $G_t = \sum_{k=t}^{\infty} \gamma^{k-t} r_k$, where $\gamma \in [0, 1]$ is the discount factor. The agent's goal is to find an optimal policy π^* such that

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi}\left[G_t\right]. \quad (8)$$

The subscript π of the expectation represents its dependence on the policy.

In the next section, we present the state-action value function that is used as a tool to find π^*.

B. State-action Value function

The state-action value function $Q_{\pi}(z, u)$ is defined as

$$Q_{\pi}(z, u) = \mathbb{E}_{\pi}\left[G_t | Z_t = z, U_t = u\right]. \quad (9)$$

That is, the expected cumulative rewards for taking action u at state z and thereafter following policy π. Using the Markov property (Eq. (7)) of the environment, one can decompose Eq. (9) to

$$\begin{aligned} Q_{\pi}(z, u) =& \mathbb{E}\left[r_t | Z_t = z, U_t = u\right] + \\ & \gamma \mathbb{E}_{\pi}\left[Q_{\pi}(Z_{t+1}, U_{t+1}) | Z_t = z, U_t = u\right]. \end{aligned} \quad (10)$$

The decomposition in (10) is essential when estimating the function $Q_{\pi}(\cdot, \cdot)$ when π is fixed. Once the state-action value function is estimated, it forms the basis for the improvement of a given policy. That is, for each state z, the current action $u(z)$ can be improved to the action $u'(z)$ by choosing

$$u'(z) = \arg\max_{u} Q_{\pi}(z, u). \quad (11)$$

[1]One can show that the marginal probabilities are sufficient since the objective is to maximize additive rewards

C. Function approximation

The *function approximators* in RL are parameterized models for $Q_{\pi}(z, u), A(z)$. The *actor* is defined by $A_{\mu}(z)$, a parametric model of $A(z)$, whose parameters are μ. The *critic* is defined by $Q_{\omega}(z, u)$, a parametric model of the state-action value function the corresponds to $A_{\mu}(z)$, whose parameters are ω. Generally, the actor and critic are modeled by neural networks (NNs), as shown in Fig 2.

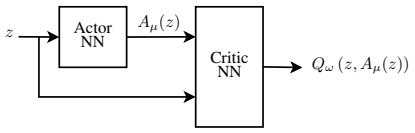

Fig. 2. Depiction of actor and critic networks. The actor network comprises a NN that maps the state z to an action $u = A_{\mu}(z)$. The critic NN maps z, u pair to the estimated future cumulative rewards.

The function approximations are modeled by a differentiable parametric model. Hence, learning $Q_{\pi}(z, u), A(z)$ can be done without visiting the entire state space. Specifically, the approximation interpolates its estimate for observed states to unobserved states, which enables the algorithm to converge without visiting the entire state space. This constitutes the main difference between the RL approach and previous methods, such as DP, which turn it into a tractable solution for channels with high cardinality.

D. DDPG algorithm

The DDPG algorithm [17] is a deep RL algorithm for deterministic action and continuous state and action spaces. The training procedure comprises N_{ep} episodes, where each episode contains T sequential steps. A single step of the algorithm comprises two parallel operations: (1) collecting experience from the environment, and (2) training the actor and critic networks to obtain the optimal policy.

In the first operation, the agent collects experience from the environment. Given the current state z_{t-1}, the agent chooses an action u_t according to a ϵ-greedy policy, where $\epsilon \in [0, 1]$. That is, with probability $1 - \epsilon$ the agent acts according to $A_{\mu}(z_{t-1})$, and with probability ϵ the agent takes a random action uniformly over the action space. The term ϵ denotes the exploration parameter, and it is crucial to encourage the agent to search the entire state and action spaces. Then the agent samples from the environment the incurred reward r_t and the next state z_t. The transition tuple (z_{t-1}, u_t, r_t, z_t) is then stored in a *replay buffer*, a bank of experience, that is used to improve the actor and critic networks. Finally, the agent updates its current state to be z_t and moves to the next step.

The second operation entails training the actor and critic networks. First, N_{mb} transitions are drawn randomly from the replay buffer. Second, for each transition, we compute its target based on the right-hand side of Eq. 10.

$$y_i = r_i + \gamma Q_{\omega}\left(z_i', A_{\mu}(z_i')\right), \quad \forall i = 1, \dots, N_{mb}. \quad (12)$$

Then we minimize the following objective with respect to the parameters of the critic network ω as given by

$$L(\omega) = \frac{1}{N_{mb}} \sum_{i=1}^{N_{mb}} \left[Q_\omega \left(z_i, A_\mu(z_i) \right) - y_i \right]^2 . \tag{13}$$

The aim of this update is to train the Critic to comply with Eq. (10). Afterward, we train the actor to maximize the critic's estimation of future cumulative rewards. That is, we train the actor to choose actions that result in high cumulative rewards according to the critic's estimation. The actor update formula is given by

$$\nabla_\mu Q_\omega(z, A_\mu(z)) = \frac{1}{N_{mb}} \sum_{i=1}^{N_{mb}} \nabla_a Q_\omega \left(z_i, a \right) |_{a=A(z_i)} \nabla_\mu A \left(z_i \right) . \tag{14}$$

To conclude, the algorithm alternates between improving the critic's estimation of future cumulative rewards and training the actor to choose actions that maximize the critic's estimation. The algorithm work flow is depicted in Fig. 3.

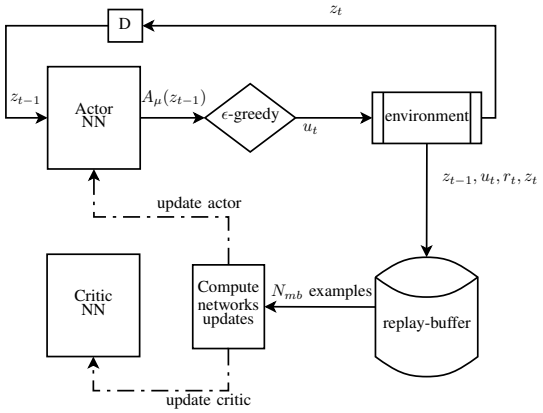

Fig. 3. Depiction of the work flow of the DDPG algorithm. At each time step t, the agent samples a transition from the environment using ϵ-greedy policy and stores the transition in the replay buffer. Simultaneously, N_{mb} past transition are drawn from the replay buffer and used to update the critic and actor NN according to Equations (13) and (14) respectively.

V. Estimating the capacity of the Ising3 channel using RL

In This section, we show the formulation of the feedback capacity as a RL problem, including details of the implementation of the RL algorithm and the experiments we conducted on various unifilar FSCs with feedback.

A. Formulation of the feedback capacity as a RL problem

We formulate the Ising3 feedback capacity as a RL problem that is based on the formulation as done in [4]. We define the state by a two-dimensional vector, $z_t = \left[p\left(s_t = 0 | y^t\right), p\left(s_t = 1 | y^t\right) \right]^T$. The action is defined by $u_t = p\left(x_t | z_{t-1}\right) \in \mathbb{R}^{3\times 3}$. The reward is defined by $r_t = I(X_t, S_{t-1}; Y_t | Y^{t-1})$, which is a deterministic function of $p\left(x_t, s_{t-1}, y_t | t^{t-1}\right) = z_{t-1} u_t p\left(y_t | x_t, s_{t-1}\right)$. Hence, the conditional distribution $p\left(r_t | z_{t-1}, u_t\right)$ is induced by the channel distribution Eq. (2). The next state distribution is given by the BCJR equation as given in Eq. (35) in [4]. Accordingly, the conditional distribution $p\left(z_t | z_{t-1}, u_t\right)$ is induced by the channel distribution, Eq. (2) and the state evolution, Eq. (3).

B. Implementation of the RL algorithm

We model $Q_\pi(z, u)$, $A_\mu(z)$ with two NNs, each of which is composed of three fully connected hidden layers of 300 units separated by a batch normalization layer. The actor network input is the state z and its output is a matrix $A_\mu(z) \in \mathbb{R}^{3\times 3}$ such that $A_\mu(z)^T \mathbf{1} = \mathbf{1}$. The critic network input is the tuple $\{z, A_\mu(z)\}$ and its output is a scalar, which is the estimate for the cumulative future rewards. In our experiments, we trained the networks for $N_{ep} = 10^4$ episodes. Each episode length is $T = 500$ steps. For the exploration, we chose $\epsilon = 0.1$ and decayed it by 0.999 each episode.

C. Experiments

We conducted several experiments to verify the effectiveness of our formulation. First, we focused on experimenting channels whose analytic solution was proven in the past, such as the Trapdoor channel [4], Ising channel with a binary alphabet [5], [8], Binary Erasure channel with input constraint [6], and the Dicode channel [12]. The results showed that the obtained achievable rates were within 99.99% of the feedback capacity for all channels.

Our aim is to solve a channel with large cardinality that previous methods have failed to solve due to computational complexity. We chose the Ising3 channel as a candidate and used our formulation to estimate its feedback capacity. We ran a simulation over the Ising3 channel with the same RL model as used in the previous experiments. By the end of training, we obtained a policy whose achievable rate is 0.96110. Another property of the obtained policy is that it visits only six discrete states as shown in Fig. 4. Furthermore, the transition between states is determined uniquely given the output of the channel. These transitions can be shown as a Q-graph, as depicted in Fig. 5.

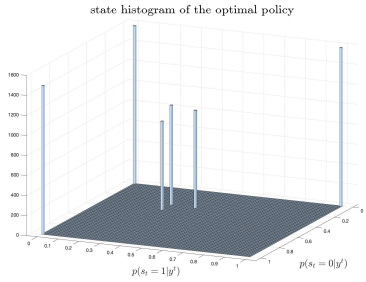

Fig. 4. State histogram of the optimal policy as obtained from RL. The histogram was generated by a Monte-Carlo evaluation of the estimated policy.

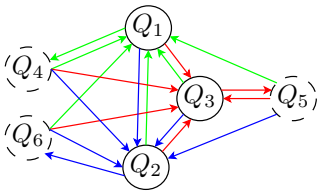

Fig. 5. Q-graph showing the transitions between states as a function of the channel's output. Blue, red and green lines correspond to $Y = 0, 1, 2$, respectively. States with dashed lines and states with solid line behave similarly.

In the next section we use the Q-graph we obtained from the estimated policy of the RL algorithm to solve the Ising3 channel.

VI. Analytic Solution for the Ising3 Channel

In this section, we prove Theorem 3 concerning the feedback capacity of the Ising3. Specifically, we use the graphical structure in Fig. 5 to compute a tight upper bound, and analyze the rate of the proposed coding scheme.

1) Bounds on the feedback capacity: The Q-graph method, introduced in [12], is a general technique that exploits the discrete histogram in Fig. 4 to provide upper and lower bounds on the capacity. The upper bound states that for any choice of a Q-graph,

$$C_{fb} \leq \sup_{p(x|s,q)} I\left(X, S; Y|Q\right), \tag{15}$$

where the joint distribution is $P_{S,Q}P_{X|S,Q}P_{Y|X,S}$ and $P_{S,Q}$ is a stationary distribution. The upper-bound is tight, that is, equals to the feedback capacity, when the maximizer of (15) satisfies the Markov chain $S' - Q' - (Q, Y)$.

We use convex optimization tool to compute the upper-bound in (15) with respect to the Q-graph in Fig. 5. The result is used to conjecture a parameterized input $P_{X|S,Q}$ as the optimal solution. Then, using the convexity of the upper bound (as a function of the entire joint distribution), one can show that the conjectured solution is optimal, and that the upper-bound can be simplified to the expression in Theorem 3. The tightness of the upper bound is shown via the Markov chain above.

2) Coding scheme - Sketch of proof for Theorem 4: The coding scheme in Section 4 is a generalization of the optimal coding scheme for $|\mathcal{X}| = 2$ that was presented in [5]. We analyze the achievable rate by computing the entropy rate of input symbols, divided by the expected time until decoding a single symbol.

The entropy rate can be computed from the the symbols transition entropy:

$$H(\nu_i|\nu_{i-1}) = H_2(p) + (1-p)\log\left[|\mathcal{X}|-1\right]. \tag{16}$$

The expected time until decoding a single symbol ν_i is

$$\mathbb{E}\left[L\right] = p \cdot 2 + (1-p)\cdot 1.5. \tag{17}$$

That is since when $\nu_i = \nu_{i-1}$, the symbol is sent twice, and when $\nu_i \neq \nu_{i-1}$, the symbol is sent once or twice with equal probability. The proof is completed by dividing Eq. (16) by Eq. (17) and taking a maximum over p.

VII. Conclusions

We derived an estimation algorithm of the feedback capacity of a unifilar FSC using RL. The RL approach addresses the cardinality constraint and establishes RL as a useful tool for channels with high cardinality. We provided an example over the Ising3 channel, where we used the insights provided by the numerical results to analytically compute its feedback capacity. Furthermore, we showed a simple capacity-achieving coding scheme for the Ising3 channel with feedback.

Additionally, our preliminary results imply that we are able to solve the Ising channel for any alphabet size. Then, we plan to solve different channels numerically and, hopefully, establish methods to induce their analytic solution, and capacity-achieving coding schemes. Furthermore, we plan to use the feedback capacity problem as a framework to improve the RL algorithms.

References

[1] R. G. Gallager, *Information theory and reliable communication*. New York: Wiley, 1968.

[2] H. H. Permuter, T. Weissman, and A. J. Goldsmith, "Finite state channels with time-invariant deterministic feedback," *IEEE Trans. Inf. Theory*, vol. 55, no. 2, pp. 644–662, 2009.

[3] S. Tatikonda and S. Mitter, "The capacity of channels with feedback," *IEEE Trans. Inf. Theory*, vol. 55, no. 1, pp. 323–349, Jan. 2009.

[4] H. H. Permuter, P. Cuff, B. V. Roy, and T. Weissman, "Capacity of the trapdoor channel with feedback," *IEEE Trans. Inf. Theory*, vol. 54, no. 7, pp. 3150–3165, Jul. 2009.

[5] O. Elishco and H. Permuter, "Capacity and coding for the Ising channel with feedback," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5138–5149, Sep. 2014.

[6] O. Sabag, H. Permuter, and N. Kashyap, "The feedback capacity of the binary erasure channel with a no-consecutive-ones input constraint," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 8–22, 2016.

[7] H. Permuter, H. Asnani, and T. Weissman, "Capacity of a POST channel with and without feedback," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 6041–6057, Oct. 2014.

[8] A. Sharov and R. M. Roth, "On the capacity of generalized ising channels," *IEEE Trans. on Inf. Theory*, vol. PP, no. 99, pp. 1–1, Dec. 2016.

[9] O. Sabag, H. H. Permuter, and N. Kashyap, "Feedback capacity and coding for the BIBO channel with a no-repeated-ones input constraint," *IEEE Trans. Inf. Theory*, vol. 64, no. 7, pp. 4940–4961, Jul. 2018.

[10] J. Wu and A. Anastasopoulos, "On the capacity of the chemical channel with feedback," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 295–299.

[11] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback codes via deep learning," *CoRR*, vol. abs/1807.00801, 2018. [Online]. Available: http://arxiv.org/abs/1807.00801

[12] O. Sabag, H. H. Permuter, and H. D. Pfister, "A single-letter upper bound on the feedback capacity of unifilar finite-state channels," *IEEE Trans. Inf. Theory*, vol. 63, no. 3, pp. 1392–1409, March 2017.

[13] T. Berger and F. Bonomi, "Capacity and zero-error capacity of Ising channels," *IEEE Trans. Inf. Theory*, vol. 36, pp. 173–180, 1990.

[14] E. Ising, "Beitrag zur theorie des ferromagnetismus," *Zeitschrift für Physik*, vol. 31, no. 1, pp. 253–258, 1925.

[15] T. Cover, "Enumerative source encoding," *IEEE Trans. Inf. Theory*, vol. 19, no. 1, pp. 73–77, January 1973.

[16] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[17] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.