Capacity of Continuous Channels with Memory via Directed Information Neural Estimator

Ziv Aharoni
Ben Gurion University
zivah@post.bgu.ac.il

Dor Tsur
Ben Gurion University
dortz@post.bgu.ac.il

Ziv Goldfeld
Cornell University
goldfeld@cornell.edu

Haim H. Permuter
Ben Gurion University
haimp@bgu.ac.il

***Abstract*—Calculating the capacity (with or without feedback) of channels with memory and continuous alphabets is a challenging task. It requires optimizing the directed information (DI) rate over all channel input distributions. The objective is a multi-letter expression, whose analytic solution is only known for a few specific cases. When no analytic solution is present or the channel model is unknown, there is no unified framework for calculating or even approximating capacity. This work proposes a novel capacity estimation algorithm that treats the channel as a 'black-box', both when feedback is or is not present. The algorithm has two main ingredients: (i) a neural distribution transformer (NDT) model that shapes a noise variable into the channel input distribution, which we are able to sample, and (ii) the DI neural estimator (DINE) that estimates the communication rate of the current NDT model. These models are trained by an alternating maximization procedure to both estimate the channel capacity and obtain an NDT for the optimal input distribution. The method is demonstrated on the moving average additive Gaussian noise channel, where it is shown that both the capacity and feedback capacity are estimated without knowledge of the channel transition kernel. The proposed estimation framework opens the door to a myriad of capacity approximation results for continuous alphabet channels that were inaccessible until now.**

I. Introduction

Many discrete-time continuous-alphabet communication channels involve correlated noise or inter-symbol interference (ISI). Two predominant communication scenarios over such channels are when feedback from the receiver back to the transmitter is or is not present. The fundamental rates of reliable communication over such channels are, respectively, the feedback (FB) and feedforward (FF) capacity. Starting from the latter, the FF capacity of an n-fold point-to-point channel $P_{Y^n|X^n}$, denoted C_{FF}, is given by [1]

$$C_{\mathsf{FF}} = \lim_{n\to\infty} \sup_{P_{X^n}} \frac{1}{n} I(X^n;Y^n). \tag{1}$$

In the presence of feedback, the FB capacity C_{FB} is [17]

$$C_{\mathsf{FB}} = \lim_{n\to\infty} \sup_{P_{X^n\|Y^{n-1}}} \frac{1}{n} I(X^n \to Y^n) \tag{2}$$

where,

$$I(X^n \to Y^n) := \sum_{i=1}^{n} I(X^i;Y_i|Y^{i-1}) \tag{3}$$

is the directed information (DI) from the input sequence X^n to the output Y^n [8], and $P_{X^n\|Y^{n-1}} := \prod_{i=1}^{n} P_{X_i|X^{i-1}Y^{i-1}}$ is the distribution of X^n causally-conditioned on Y^{n-1} (see [21], [24] for further details). Built on (3), for stationary processes, the DI rate is defined as

$$I(\mathcal{X} \to \mathcal{Y}) := \lim_{n\to\infty} \frac{1}{n} I(X^n \to Y^n). \tag{4}$$

As shown in [8], when feedback is not present, the optimization problem (2) (which amounts to optimizing over P_{X^n} rather than $P_{X^n\|Y^n}$) coincides with (1). Thus, DI provides a unified framework for representing both FF and FB capacities.

Computing C_{FF} and C_{FB} requires solving a multi-letter optimization problem. Closed form solutions to this challenging task are known only in several special cases. A common example for C_{FF} is the Gaussian channel with memory [14] and the ISI Gaussian channel [15]. There are no known extensions of these solutions to the non-Gaussian case. For C_{FB}, a solution for the 1st order moving average additive Gaussian noise (MA(1)-AGN) channel was found [12]. Another closed form characterization is available for auto-regressive moving-average (ARMA) AGN channels [11]. To the best of our knowledge, these are the only two non-trivial examples of continuous channels with memory whose FB capacity is known in closed form. Furthermore, when the channel model is unknown, there is no numerically tractable method for approximating capacity based on samples.

Recent progress related to capacity computation via deep learning (DL) was made in [9], where the mutual information neural estimator (MINE) [2] was used to learn modulations for memoryless channels. Later, [19] proposed an estimator based on a reinforcement learning algorithm that iteratively estimates and maximizes the DI rate was proposed, but only for discrete alphabet channels with a known channel model.

Inspired by the above, we develop the framework for estimating FF and FB capacity of arbitrary continuous-alphabet channels, possible with memory, without knowing the channel model. Our method does not need to know the channel transition kernel. We only assume a stationary channel model and that channel outputs can be sampled by feeding it with inputs. Central to our method are a new DI neural estimator (DINE), used to evaluate the communication rate, and a neural distribution transformer (NDT), used to simulate input distributions. Together, DINE and NDT lay the groundwork for our capacity estimation algorithm. In the remainder of this section, we describe DINE, NDT, and their integration into the capacity estimator.

978-1-7281-6432-8/20/$31.00 ©2020 IEEE

A. Directed Information Neural Estimation

The estimation of mutual information (MI) from samples using neural networks (NNs) is a recently proposed approach [2], [3]. It is especially effective when the involved random variables (RVs) are continuous. The concept originated from [2], where MINE was proposed. The core idea is to represent MI using the Donsker-Varadhan (DV) variational formula

$$I(X;Y) = \sup_{\mathsf{T}:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}} \mathbb{E}\left[\mathsf{T}(X,Y)\right] - \log \mathbb{E}\left[e^{\mathsf{T}(\widetilde{X},\widetilde{Y})}\right], \quad (5)$$

where $(X,Y) \sim P_{XY}$ and $(\widetilde{X},\widetilde{Y}) \sim P_X \otimes P_Y$. The supremum is over all measurable functions T for which both expectations are finite. Parameterizing T by an NN and replacing expectations with empirical averages, enables gradient ascent optimization to estimate $I(X;Y)$. A variant of MINE that goes through estimating the underlying entropy terms was proposed in [3]. The new estimators were shown empirically to perform extremely well, especially for continuous alphabets.

Herein, we propose a new estimator for the DI rate $I(\mathcal{X} \to \mathcal{Y})$. The DI is factorized as

$$I(X^n \to Y^n) = h(Y^n) - h(Y^n \| X^n), \quad (6)$$

where $h(Y^n)$ is the differential entropy of Y^n and $h(Y^n\|X^n) := \sum_{i=1}^n h(Y_i|Y^{i-1},X^i)$. Applying the approach of [3] to the entropy terms, we expand each as a Kullback-Leibler (KL) divergence plus a cross-entropy (CE) residual and invoke the DV representation. To account for memory, we derive a formula valid for causally dependent data, which involves RNNs as function approximators (rather than the FF network used in the independently and identically distributed (i.i.d.) case). Thus, DINE is an RNN-based estimator for the DI rate from X^n to Y^n based on their samples.

Estimation of DI between discrete-valued processes was studied in [25]–[27]. An estimator of the transfer entropy, which upper bounds DI for jointly Markov process with finite memory, was proposed [16]. DINE, on the other hand, does not assume Markovity nor discrete alphabets, and can be applied to continuous-valued stationary and ergodic processes. A detailed description of the DINE algorithm is given in subsection II-A.

B. Neural Distribution Transformer and Capacity Estimation

DINE accounts for one of the two tasks involved in estimating capacity, it estimates the objective of (2). It then remains to optimize this objective over input distributions. To that end, we design a deep generative model, termed the NDT, to approximate the channel input distributions. This is similar in flavor to generators used in generative adversarial networks [23].The designed NDT maps i.i.d. noise into samples of the channel input distribution. For estimating FB capacity, in addition to the i.i.d. noise, the NDT also receives channel FB as inputs. Together, NDT and DINE form the overall system that estimates the capacity as shown in Fig 1.

The capacity estimation algorithm trains DINE and NDT models together via an alternating optimization procedure (i.e., fixing the parameters of one model while training the other). DINE estimates the communication rate of a fixed NDT input distribution, and the NDT is trained to increase its rate with respect to fixed DINE model. Proceeding until convergence, this results in the capacity estimate, as well as an NDT generative model for the achieving input distribution. We demonstrate our method on the MA(1)-AGN channel. Both C_{FF} and C_{FB} are estimated using the same algorithm, using the channel as a black-box to solely generate samples. The estimation results are compared with the analytic solution to show the effectiveness of the proposed approach.

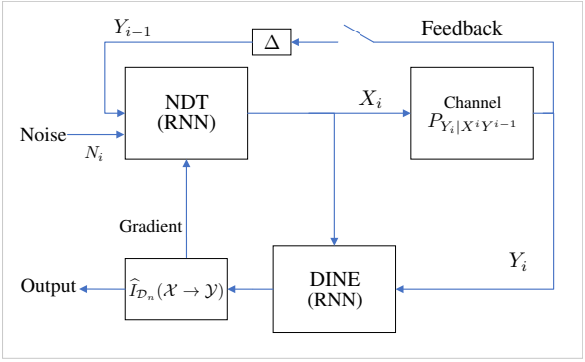

Fig. 1. The overall capacity estimator: NDT generates samples that are fed into the channel. DINE uses these samples to improve its estimation of the communication rate. DINE then supplies gradient for the optimization of NDT.

II. Methodology

We give a high-level description of the algorithm and its building blocks. Due to space limitations, full details are reserved to the extended version of this paper. The implementation is available on GitHub.[†]

A. Directed Information Estimation Method

We propose a new estimator of the DI rate between two correlated stationary processes, termed DINE. Building on [3], we factorize each term in (6) as:

$$\begin{aligned} h(Y^n) &= h_{\mathsf{CE}}(P_{Y^n}, P_{Y^{n-1}} \otimes P_{\widetilde{Y}}) \\ &\quad - D_{\mathsf{KL}}(P_{Y^n} \| P_{Y^{n-1}} \otimes P_{\widetilde{Y}}) \\ h(Y^n\|X^n) &= h_{\mathsf{CE}}\left(P_{Y^n\|X^n}, P_{Y^{n-1}\|X^{n-1}} \otimes P_{\widetilde{Y}} \middle| P_{X^n}\right) \\ &\quad - D_{\mathsf{KL}}\left(P_{Y^n\|X^n} \middle\| P_{Y^{n-1}\|X^{n-1}} \otimes P_{\widetilde{Y}} \middle| P_{X^n}\right) \end{aligned} \quad (7)$$

where $h_{\mathsf{CE}}(P_X, Q_X)$ and $D_{\mathsf{KL}}(P_X\|Q_X)$ are, respectively, the CE and KL divergence between P_X and Q_X, with

$$\begin{aligned} h_{\mathsf{CE}}(P_{Y|X}, Q_{Y|X}|P_X) &:= \int_{\mathcal{X}} h_{\mathsf{CE}}(P_{Y|X=x}, Q_{Y|X=x}) \mathrm{d}P_X(x) \\ D_{\mathsf{KL}}(P_{Y|X}\|Q_{Y|X}|P_X) &:= \int_{\mathcal{X}} D_{\mathsf{KL}}(P_{Y|X=x}\|Q_{Y|X=x}) \mathrm{d}P_X(x) \end{aligned} \quad (8)$$

[†]https://github.com/zivaharoni/capacity-estimator-via-dine

denoting their conditional versions; and $P_{\widetilde{Y}}$ is uniform reference measure over the support of the dataset. To simplify notation, we use the shorthands

$$\begin{aligned} D_Y^{(n)} &:= D_{\mathsf{KL}}(P_{Y^n} \| P_{Y^{n-1}} \otimes P_{\widetilde{Y}}) \\ D_{Y\|X}^{(n)} &:= D_{\mathsf{KL}}(P_{Y^n\|X^n} \| P_{Y^{n-1}\|X^{n-1}} \otimes P_{\widetilde{Y}}). \end{aligned} \tag{9}$$

Subtracting both elements in (7) and observing that the difference of CE terms equals the DI at the former time step, we have

$$I(X^n \to Y^n) = I(X^{n-1} \to Y^{n-1}) + D_{Y\|X}^{(n)} - D_Y^{(n)}. \tag{10}$$

Note that the difference of KL divergences equals $I(X^n; Y_n | Y^{n-1})$. For stationary data processes we take the limit and obtain

$$\lim_{n\to\infty} D_{Y\|X}^{(n)} - D_Y^{(n)} = \lim_{n\to\infty} I(X^n; Y_n | Y^{n-1}) = I(\mathcal{X} \to \mathcal{Y}). \tag{11}$$

Each D_{KL} is expanded by its DV representation [4] as:

$$\begin{aligned} D_Y^{(n)} &= \sup_{\mathsf{T}:\Omega\to\mathbb{R}} \mathbb{E}\left[\mathsf{T}(Y^n)\right] - \log \mathbb{E}\left[e^{\mathsf{T}(Y^{n-1},\widetilde{Y})}\right] \\ D_{Y\|X}^{(n)} &= \sup_{\mathsf{T}:\Omega\to\mathbb{R}} \mathbb{E}\left[\mathsf{T}(Y^n\|X^n)\right] - \log \mathbb{E}\left[e^{\mathsf{T}(Y^{n-1}\|X^{n-1},\widetilde{Y})}\right]. \end{aligned} \tag{12}$$

To maximize (12), each DV potential is parametrized by a modified LSTM and expected values are estimated by empirical averages over the dataset $\mathcal{D}_n := \{(x_i, y_i)\}_{i=1}^n$. Thus, the optimization objectives are:

$$\begin{aligned} \widehat{D}_{Y\|X}(\theta_{Y\|X}, \mathcal{D}_n) &:= \frac{1}{n}\sum_{i=1}^n \mathsf{T}_{\theta_{Y\|X}}(y_i | x^i y^{i-1}) \\ &\quad - \log\left(\frac{1}{n}\sum_{i=1}^n e^{\mathsf{T}_{\theta_{Y\|X}}(\widetilde{y}_i | x^i y^{i-1})}\right) \\ \widehat{D}_Y(\theta_Y, \mathcal{D}_n) &:= \frac{1}{n}\sum_{i=1}^n \mathsf{T}_{\theta_Y}(y_i | y^{i-1}) \\ &\quad - \log\left(\frac{1}{n}\sum_{i=1}^n e^{\mathsf{T}_{\theta_Y}(\widetilde{y}_i | y^{i-1})}\right) \end{aligned} \tag{13}$$

where $\tilde{y}^n \overset{\text{i.i.d.}}{\sim} P_{\widetilde{Y}}$ and T_{θ_Y}, $\mathsf{T}_{\theta_{Y\|X}}$ are the parametrized potentials.

The estimator is given by:

$$\widehat{I}_{\mathcal{D}_n}(\mathcal{X} \to \mathcal{Y}) := \sup_{\theta_{Y\|X}\in\Theta_{Y\|X}} \widehat{D}_{Y\|X} - \sup_{\theta_Y\in\Theta_Y} \widehat{D}_Y \tag{14}$$

By universal approximation of RNNs [6] and Breiman's theorem [7], the maximizer of (14) approaches $I(\mathcal{X} \to \mathcal{Y})$ as the number of samples grows, provided the neural networks are sufficiently expressive.

To capture the time dependencies in $\mathcal{D}_n$ we introduce a modified LSTM network model for functional approximation. LSTM [5] is an RNN that receives a time series $\{y_i\}_{i=1}^T$ as input and for each i, performs a recursive non-linear transform to calculate its hidden state s_i. We denote the LSTM function by $F : (y_i, s_{i-1}) \longmapsto s_i$. The full characterization of F is provided in [5].

Algorithm 1 Directed Information Rate Estimation

input: Samples of the process $\mathcal{D}_n$.
output: $\widehat{I}_{\mathcal{D}_n}(\mathcal{X} \to \mathcal{Y})$, estimated directed information rate.

Initialize networks parameters $\theta_Y, \theta_{Y\|X}$.
Step 1, Optimization:
repeat
  Draw a batch $\mathcal{D}_B = \{(x_{(i-1)T}^{iT}, y_{(i-1)T}^{iT})\}_{i=1}^B$
  Feed the network with the examples and compute loss $\widehat{D}_{Y\|X}(\theta_{Y\|X}, \mathcal{D}_B)$, $\widehat{D}_Y(\theta_Y, \mathcal{D}_B)$.
  Update networks parameters:
  $\theta_{Y\|X} \leftarrow \theta_{Y\|X} + \nabla\widehat{D}_{Y\|X}(\theta_{Y\|X}, \mathcal{D}_B)$
  $\theta_Y \leftarrow \theta_Y + \nabla\widehat{D}_Y(\theta_Y, \mathcal{D}_B)$
until convergence
Step 2, Perfrom a Monte Carlo estimation over $\mathcal{D}_n$ and subtract loss evaluations to obtain estimation : $\widehat{I}_{\mathcal{D}_n}(\mathcal{X} \to \mathcal{Y}) = \widehat{D}_{Y\|X}(\theta_{Y\|X}, \mathcal{D}_n) - \widehat{D}_Y(\theta_Y, \mathcal{D}_n)$

We modify the structure of the LSTM to perform the calculations:

$$\begin{aligned} s_i &= F(y_i, s_{i-1}) = s(y_i | y^{i-1}) \\ \widetilde{s}_i &= F(\widetilde{y}_i, s_{i-1}) = s(\widetilde{y}_i | y^{i-1}) \end{aligned} \tag{15}$$

A similar modification is introduced for $\widehat{D}_{Y\|X}$ by substitution of y_i with (y_i, x_i) and $\widetilde{y}_i$ with $(\widetilde{y}_i, x_i)$, we have:

$$\begin{aligned} s_i &= F(y_i, x_i, s_{i-1}) = s(y_i | y^{i-1}, x^i) \\ \widetilde{s}_i &= F(\widetilde{y}_i, x_i s_{i-1}) = s(\widetilde{y}_i | y^{i-1}, x^i). \end{aligned} \tag{16}$$

A visualization of a modified LSTM cell (unrolled) is shown in Fig. 2. The LSTM cell's output is the sequence $\{(s_i, \widetilde{s}_i)\}_{i=1}^n$, which is fed into a fully-connected layer to obtain T_{θ_Y} and $\mathsf{T}_{\theta_{Y\|X}}$. As demonstrated by Algorithm 1 and Fig. 3, in each iteration we draw $\mathcal{D}_B$, a subset on $\mathcal{D}_n$, of size B. We feed the NN with $\mathcal{D}_B$ to acquire T_{θ_Y}, $\mathsf{T}_{\theta_{Y\|X}}$. Those enter the NN loss function (13), and gradients are calculated to update the NN parameters $\theta_Y, \theta_{Y\|X}$.

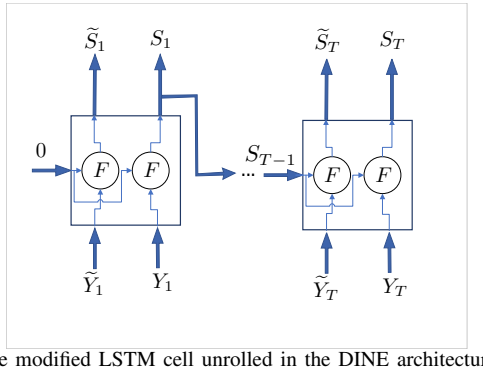

Fig. 2. The modified LSTM cell unrolled in the DINE architecture of $\widehat{D}_Y$. Recursively, at each time i, (y_i, s_{i-1}) and $(\widetilde{y}_i, s_{i-1})$ are mapped to s_i and $\widetilde{s}_i$, respectively.

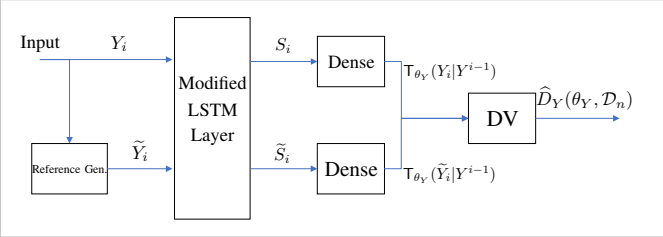

Fig. 3. End-to-end architecture for estimating $\widehat{D}_Y(\theta_Y, \mathcal{D}_n)$. For each batch of time sequences, a batch of the same size is sampled from the reference measure. Together, these samples are fed into the NN to compute T_{θ_Y} and $\mathsf{T}_{\theta_{Y\|X}}$, from which the estimate is assumbled.

B. Neural Distribution Transformer

The DINE model is an effective approach to estimate the argument of (2). However, finding the capacity comprises maximization of the DI with respect to the input distribution. For this purpose we present the NDT model that represents a general input distribution of the channel. At each iteration $i = 1, \ldots, n$ the NDT maps an i.i.d noise vector N^i to a channel input variable X_i. When feedback is present the NDT maps $(N^i, Y^{i-1}) \longmapsto X_i$. Thus, NDT is represented by an RNN with parameters μ as shown in Fig. 4. The NDT model is used to generate the channel input X^n, and the DINE estimates the DI between X^n and Y^n.

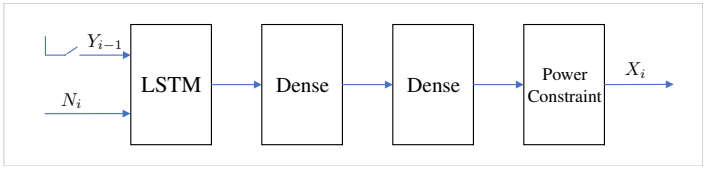

Fig. 4. The NDT. The noise and past channel output (if feedback is applied) are fed into an NN. The last layer performs normalization to obey the power constraint, if needed.

C. Complete Architecture Layout

Combining DINE and NDT models into a complete system enables capacity estimation. As shown in Fig. 1, the NDT model is fed with i.i.d. noise and its output is the samples X^n. These samples are fed into the channel to generate outputs. Then, DINE uses (X^n, Y^n) to produce the estimate $\widehat{I}_{\mathcal{D}_n}(\mathcal{X} \to \mathcal{Y})$. To estimate capacity, DINE and NDT models are trained together. The training scheme, as shown in Algorithm 2, is a variant of alternated maximization procedure. This procedure iterates between updating the DINE parameters θ and the NDT parameters μ, each time keeping one of the models fixed. At the end of training a long Monte-Carlo evaluation of $\sim 10^6$ samples is done in order to estimate the expectations in (13).

Applying this algorithm to channels with memory estimates their capacity without any specific knowledge of the channel underlying distribution. Next, we demonstrate the effectiveness of this algorithm on continuous alphabet channels.

Algorithm 2 Capacity Estimation

input: Continuous channel, feedback indicator
output: $\widehat{I}_{\mathcal{D}_n}(\mathcal{X} \to \mathcal{Y}, \mu)$, estimated capacity.

Initialize DINE parameters, $\theta_Y, \theta_{Y\|X}$
Initialize NDT parameters μ
if feedback indicator **then**
  Add feedback to NDT
repeat
  Step 1: Train DINE model
  Generate B sequences of length T of i.i.d random noise
  Compute $\mathcal{D}_B = \{(x_i^T, y_i^T)\}_{i=1}^B$ with NDT and channel
  Compute $\widehat{D}_{Y\|X}(\theta_{Y\|X}, \mathcal{D}_B)$, $\widehat{D}_Y(\theta_Y, \mathcal{D}_B)$
  Update DINE parameters:
    $\theta_{Y\|X} \leftarrow \theta_{Y\|X} + \nabla \widehat{D}_{Y\|X}(\theta_{Y\|X}, \mathcal{D}_B)$
    $\theta_Y \leftarrow \theta_Y + \nabla \widehat{D}_Y(\theta_Y, \mathcal{D}_B)$
  Step 2: Train NDT
  Generate B sequences of length T of i.i.d random noise
  Compute $\mathcal{D}_B = \{(x_i^T, y_i^T)\}_{i=1}^B$ with NDT and channel
  compute the objective:
  $\widehat{I}_{\mathcal{D}_B}(\mathcal{X} \to \mathcal{Y}, \mu) = \widehat{D}_{Y\|X}(\theta_{Y\|X}, \mathcal{D}_B) - \widehat{D}_Y(\theta_Y, \mathcal{D}_B)$
  Update NDT parameters:
    $\mu \leftarrow \mu + \nabla_\mu \widehat{I}_{\mathcal{D}_B}(\mathcal{X} \to \mathcal{Y}, \mu)$
until convergence
Monte Carlo evaluation of $\widehat{I}_{\mathcal{D}_n}(\mathcal{X} \to \mathcal{Y}, \mu)$
return $\widehat{I}_{\mathcal{D}_n}(\mathcal{X} \to \mathcal{Y}, \mu)$

III. Numerical Results

We demonstrate the performance of Algorithm 2 on the AWGN channel and the first order MA-AGN channel. The numerical results are then compared with the analytic solution to verify the effectiveness of the proposed method.

A. AWGN channel

The power constrained AWGN channel is considered. This is an instance of a memoryless, continuous-alphabet channel for which analytic solution is known. The channel model is

$$Y_i = X_i + Z_i, \quad i \in \mathbb{N}, \tag{17}$$

where $Z_i \sim \mathcal{N}(0, \sigma^2)$ are i.i.d RVs, and X_i is the channel input sequence bound to the power constraint $\mathbb{E}\left[X_i^2\right] \leq P$. The capacity of this channel is given by $\mathsf{C} = \frac{1}{2}\log\left(1 + \frac{P}{\sigma^2}\right)$. In our implementation we chose $\sigma^2 = 1$ and estimated capacity for a range of P values. The numerical results are compared to the analytic solution in Fig. 5, where a clear correspondence is seen.

B. Gaussian MA(1) channel

We consider both the FB (C_{FB}) and the FF (C_{FB}) capacity of the MA(1) Gaussian channel. The model here is:

$$\begin{aligned} Z_i &= \alpha U_{i-1} + U_i \\ Y_i &= X_i + Z_i \end{aligned} \tag{18}$$

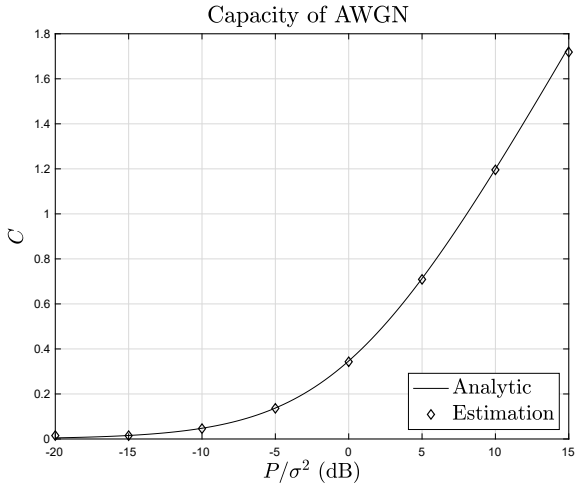

Fig. 5. Estimation of AWGN channel capacity for various SNR values

where, $U_i \sim \mathcal{N}(0,1)$ are i.i.d., X_i is the channel input sequence bound to the power constraint $\mathbb{E}\left[X_i^2\right] \leq P$, and Y_i is the channel output.

1) Feedforward capacity: The FF capacity of the MA(1) Gaussian channel with input power constraint can be obtained via the water-filing algorithm [14]. This is the benchmark against which we compare the quality of the C_{FF} estimate produced by Algorithm 2. Results are shown in Fig. 6.

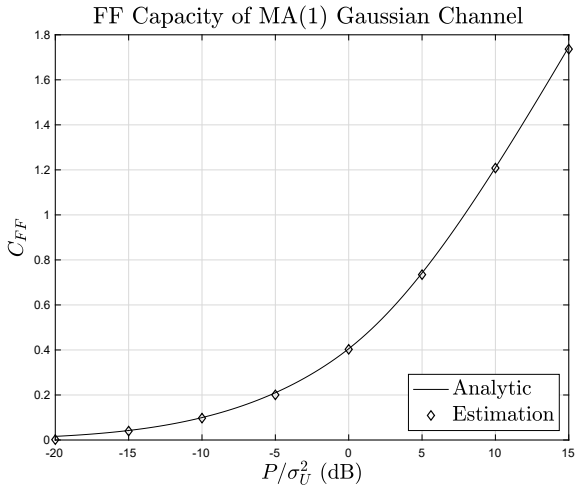

Fig. 6. Performance of C_{FF} estimation in the MA(1)-AGN channel.

2) Feedback capacity: Computing the FB capacity of the ARMA(k) Gaussian channel can be formulated as a dynamic programming, which is then solved via an iterative algorithm [11]. For the particular case of (18), C_{FB} is given by $-\log(x_0)$, where x_0 is a solution to a 4th order polynomial equation. The estimates for C_{FB} produced by Algorithm 2 are compared to the analytic solutions in Fig. 7. The optimization dynamics for our algorithm are shown in Fig. 8.

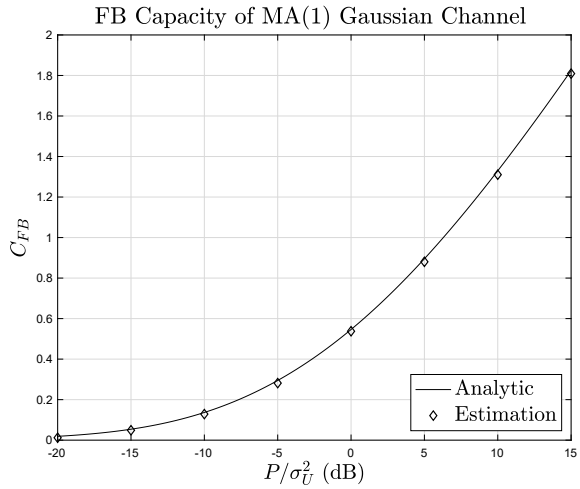

Fig. 7. Preformance of C_{FB} estimation in the MA(1)-AGN channel.

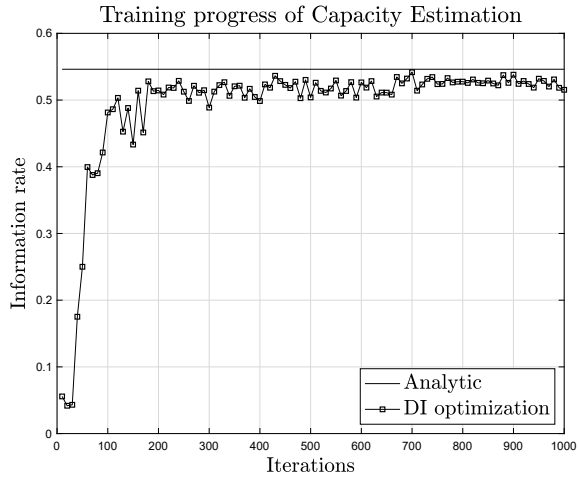

Fig. 8. Optimization progress of DI rate of Algorithm 2 for the FB setting with $P = 1$. The information rates were estimated by a Monte-Carlo evaluation of (14) with 10^5 samples.

IV. Conclusion and Future Work

We presented a methodology for estimating FF and FB capacities that uses the channel as a black-box, i.e., without assuming the channel model is known and only relying its output samples. The main building block were a novel DI estimator (DINE) and the NDT model, both implemented based on RNNs. The performance of the estimator was tested on AWGN and MA(1)-AGN channels, showing estimates that agree well with analytic solution.

Despite the empirical effectiveness of DINE, we stress that it is neither a lower nor a upper bound on the true DI (see (6)-(7)). A main goal going forward is to revise DINE so that is provably lower bounds the true value. This will imply that the induced capacity estimator lower bounds the theoretical fundamental limit. Extension of our method to multiuser channels is also of interest, as capacity results in multiuser information theory are quite scarce. Another objective is coupling DINE with theoretical performance guarantees.

REFERENCES

[1] R. G. Gallager. Information theory and reliable communication. Vol. 2. New York: Wiley, 1968.

[2] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm. Mine: mutual information neural estimation. arXiv preprint arXiv:1801.04062. June 2018.

[3] C. Chan, A. Al-Bashabshesh, H. P. Huang, M. Lim, D. S. H. Tam and C. Zhao. Neural Entropic Estimation: A faster path to mutual information estimation. arXiv preprint arXiv:1905.12957, May 2019.

[4] M. Donsker, and S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time, iv. Communications on Pure and Applied Mathematics, 36(2):183-212. March 1983.

[5] S. Hochreiter and J. Schumidhuber.Long short-term memory. Neural Computation 9(8): 1735-1780. November 1997.

[6] A. M. Schäfer and H. G. Zimmermann. Recurrent neural networks are universal approximators. International journal of neural systems 17.04: 253-263. 2007.

[7] L. Breiman. "The individual ergodic theorem of information theory" The Annals of Mathematical Statistics: 809-811. September 1957. Information Theory, IEEE Trans. Comm., vol. COM-21, pp. 1345-1351. December 1973.

[8] J. Massey, Causality, feedback, and directed information. Proc. Int. Symp. Inf. Theory Appl. , pp. 303–305. November 1990.

[9] R. Fritschek, R. F. Schaefer, and G. Wunder. Deep Learning for Channel Coding via Neural Mutual Information Estimation. arXiv preprint arXiv:1903.02865 March 2019.

[10] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. Neural networks 2.5 : 359-366. March 1989.

[11] S. Yang, A. Kavcic, and S. Tatikonda. On the feedback capacity of power-constrained Gaussian noise channels with memory. IEEE Trans. Inf. Theory 53.3 : 929-954. March 2007.

[12] Y. H. Kim. Feedback capacity of the first-order moving average Gaussian channel. IEEE Trans. Inf. Theory 52.7: 3063-3079. July 2006.

[13] Y. H. Kim. Feedback capacity of stationary Gaussian channels. IEEE Trans. Inf. Theory 56.1: 57-85. Januaray 2010.

[14] T. M. Cover, and J. A. Thomas. Elements of information theory. John Wiley and Sons, 2012.

[15] W. Hirt, and J. L. Massey. Capacity of the discrete-time Gaussian channel with intersymbol interference. IEEE Trans. Inf. Theory 34.3: 38-38. May 1988.

[16] J. Zhang, O. Simeone, Z. Cvetkovic, E. Abela, and M. Richardson. ITENE: Intrinsic Transfer Entropy Neural Estimator. arXiv preprint arXiv:1912.07277. January 2020.

[17] Y. H.Kim . A coding theorem for a class of stationary channels with feedback. IEEE Trans. Inf. Theory 54.4: 1488-1499. April 2008.

[18] S. Yang, A. Kavcic, and S. Tatikonda. Feedback Capacity of Stationary Sources over Gaussian Intersymbol Interference Channels. GLOBECOM, 2006.

[19] Z. Aharoni, O. Sabag, and H. H. Permuter. Computing the Feedback Capacity of Finite State Channels using Reinforcement Learning. 2019 IEEE International Symposium on Information Theory (ISIT). IEEE, 2019.

[20] S. Molavipour, G. Bassi, and M. S. Conditional Mutual Information Neural Estimator. arXiv preprint arXiv:1911.02277. November 2019

[21] H. H. Permuter, Y. H. Kim, and T. Weissman. Interpretations of directed information in portfolio theory, data compression, and hypothesis testing. IEEE Trans. Inf. Theory 57.6: 3248-3259. June 2011.

[22] D. P. Kingma, M. Welling. Auto-Encoding Variational Bayes. arXiv preprint arXiv:1312.6114. 2013.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville & Y. Bengio. Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680). 2014.

[24] G. Kramer. Directed information for channels with feedback. Hartung-Gorre, 1998.

[25] L. Zhao, H. Permuter, Y. Kim, and T. Weissman. Universal estimation of directed information. IEEE Trans. Inf. Theory 59.10: 6220-6242. October 2013.

[26] I. Kontoyiannis, and M. Skoularidou. Estimating the directed information and testing for causality. IEEE Transactions on Information Theory 62.11: 6053-6067. November 2016.

[27] C. J. Quinn, T.P. Coleman, N. Kiyavash et al. Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. J Comput Neurosci 30, 17–44. June 2010.