Information based Language Modeling

Reading

read the paper where DINE is presented: https://arxiv.org/abs/2003.04179

> The main focus of the paper you should focus on is the directed information estimation.
>
> The part of using it to find the capacity is not relevant for now.

Theoretical questions

1. Exercise 1 (AWGN with optimal input distribution):
   - $\{X_n\}_{n\in\mathbb{N}}$ is i.i.d process with $X_i \sim \mathcal{N}(0,P)$
   - $\{Z_n\}_{n\in\mathbb{N}}$ is i.i.d process with $Z_i \sim \mathcal{N}(0,N)$
   - $P, N > 0$
   - $\{Y_n\}_{n\in\mathbb{N}}$ is defined by $Y_i = X_i + Z_i$
   - **Calculate**:

     $\lim_{n\to\infty} \frac{1}{n} I(X^n \to Y^n)$
2. Exercise 2 (parallel channels):
   - Consider k indepedent parallel channels. Each is defined as the channel in Ex. (1).
   - Let $\mathbf{X}_n = [X_n^1, \ldots, X_n^k]^T, \mathbf{Y}_n = [Y_n^1, \ldots, Y_n^k]^T$, where (X_n^i, Y_n^i) are the n^th input and output of the i^th channel.

- **Calculate**:
  $\lim_{n\to\infty} \frac{1}{n} I(\mathbf{X}^n \to \mathbf{Y}^n)$

Implementation

1. Implement DINE
2. Estimate the rate of the process in Ex. (1).
    1. show results for the same SNRs as presented in the paper.
    2. Show the convergence of the algorithm for every SNR.
3. Generalize your model to estimate the rate of a parallel channel.

   repeat (2) for parallel channel with $k = 2, 3, 5, 10$

Theoretical questions

1. Exercise 1 (AWGN with optimal input distribution):
   - $\{X_n\}_{n\in\mathbb{N}}$ is i.i.d process with $X_i \sim \mathcal{N}(0,P)$
   - $\{Z_n\}_{n\in\mathbb{N}}$ is i.i.d process with $Z_i \sim \mathcal{N}(0,N)$
   - $P,N>0$
   - $\{Y_n\}_{n\in\mathbb{N}}$ is defined by $Y_i = X_i + Z_i$
   - **Calculate**:

     $\lim_{n\to\infty} \frac{1}{n} I(X^n \to Y^n)$

$\mathbb{E}[X^2]=\delta^2$: $h(X) \le \frac{1}{2}\log_2(2\pi e\,\delta^2)$ (#)

נקבל שוויון עבור $X\sim N(0,\delta^2)$.

$$I(X\to Y) = \lim_{n\to\infty}\frac{1}{n} I(X^n\to Y^n) = \dots$$

$$\dots = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} I(X^i; Y_i \mid \underbrace{Y^{i-1}}_{\text{no feedback}}) = \dots$$

$$\dots = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} I(X^i; Y_i) = \dots$$

$$\dots = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\left[h(Y_i) - h(Y_i\mid X^i)\right] = \dots$$

$$\dots = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\left[h(X_i+Z_i) - h(X_i+Z_i\mid X^i)\right] = \dots$$

$$\dots = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\left[h(X_i+Z_i) - h(\underbrace{X_i+Z_i}_{Z_i\perp\!\!\!\perp X_i} - X_i \mid X^i)\right] = \dots \quad (X_i \text{ deterministic})$$

$$\dots = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\left[h(\underbrace{X_i+Z_i}_{Y_i}) - h(Z_i)\right] = \dots$$

$Y\sim N(0,N+P)$: $h(Y) = \frac{1}{2}\ln[2\pi e(N+P)]$ nats

(רק במקרה בשאנו מקבלים שוויון)

$Z\sim N(0,N)$: $h(Z) = \frac{1}{2}\ln[2\pi eN]$ nats

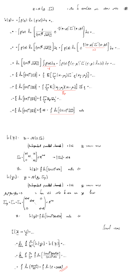
$$\dots = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2}\ln\left[\frac{N+P}{N}\right]\right] = \frac{1}{2}\ln(1+SNR)$$

$SNR = \frac{N}{P}$

2. Exercise 2 (parallel channels):

   - Consider k independent parallel channels. Each is defined as the channel in Ex. (1).
   - Let $\mathbf{X}_n = [X_n^1, \ldots, X_n^k]^T, \mathbf{Y}_n = [Y_n^1, \ldots, Y_n^k]^T$, where (X_n^i, Y_n^i) are the n^th input and output of the i^th channel.
   - **Calculate**:

     $\lim_{n\to\infty} \frac{1}{n} I(\mathbf{X}^n \to \mathbf{Y}^n)$



$$I(\underline{x} \to \underline{y}) = \lim_{n\to\infty} \frac{1}{n} I(\underline{x}^n \to \underline{y}^n) = \ldots$$

$$\ldots = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} I(\underline{x}^i ; \underline{y}_i) = \ldots$$

$$\ldots = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \left[h(\underline{y}_i) - h(\underline{z}_i) \right] = \ldots$$

מאחר ו- $\underline{y}_i$, $\underline{z}_i$ מורכבים מ-k רכיבים i.i.d. נוכל לרשום:

$$\ldots = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \left[h(y_{ij}) - h(z_{ij}) \right] = \ldots$$

וכו בעיה השקולה ל-k קיבולי ערוצים זהים כמקום, לכן: $(\forall i: \; P_i = P, N_i = N)$

$$\ldots = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} k \cdot \left[\frac{1}{2} \ln \left[\frac{N+P}{N} \right] \right] = k \cdot \frac{1}{2} \ln (1 + SNR)$$

DINE Implementation

Implement DINE – Directed Information Neural Estimation, by the paper Capacity of Continuous Channels with Memory via Directed Information Neural Estimator (H. Permuter, Z. Goldfeld, D. Tsur, Z. Aharoni).

DINE is constructed from the following formulas, which been proved in the paper:

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

$$D_Y^{(n)} := D_{\mathsf{KL}}(P_{Y^n} \| P_{Y^{n-1}} \otimes P_{\widetilde{Y}})$$

$$D_{Y\|X}^{(n)} := D_{\mathsf{KL}}(P_{Y^n\|X^n} \| P_{Y^{n-1}\|X^{n-1}} \otimes P_{\widetilde{Y}})$$

Proved in the paper the following identity:

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

Thus, to achieve the directed information of 2 R.V. we need to find the Kolbak-Liber Divergences. To do so, we use Donsker-Varadhan representation:

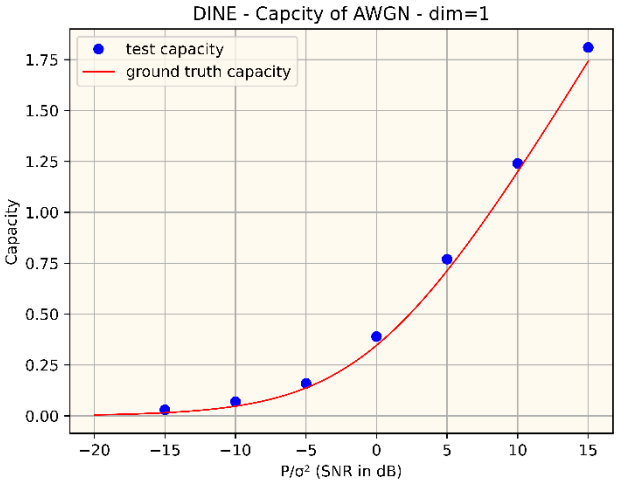
$$D_Y^{(n)} = \sup_{\mathsf{T}:\Omega\to\mathbb{R}} \mathbb{E}\left[\mathsf{T}(Y^n)\right] - \log \mathbb{E}\left[e^{\mathsf{T}(Y^{n-1},\widetilde{Y})}\right]$$

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

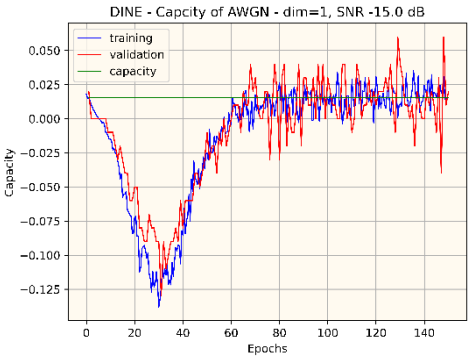
Each one of them is a neural network which it's loss function is the DV potential. We will learn the network and try to maximize that loss (Note that $\tilde{Y}$ is a uniform reference measure over the support of the dataset that has been taken).

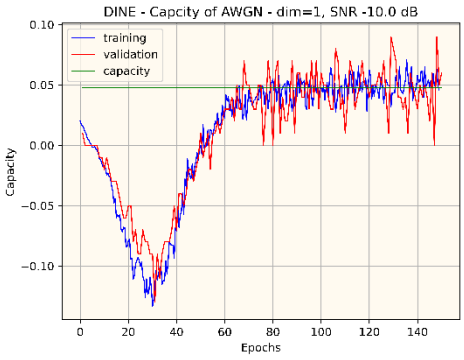
To implement it I used the structure of the PTB model I built. I changed the LSTM structure to be able to save hidden states between all timesteps and to insert modified hidden states to specific timesteps. Afterward, I created 2 models, F1 and F2, which are the DV potentials $D_Y^{(n)}, D_{Y||X}^{(n)}$ respectively. The loss function is computed by the DV potentials, over a batch of samples which are generated by the same distribution (i.e. Gaussian with mean 0 and variance P, $N = \sigma^2$ for input and noise respectively), each time a call was made to the model.

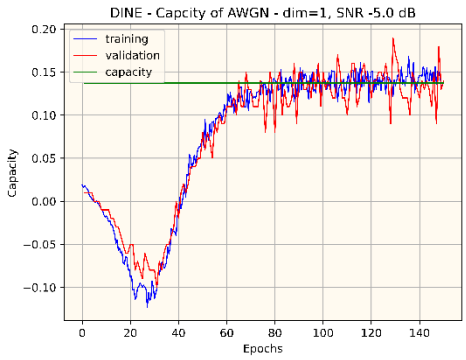
I tested the SNR ratios (P/σ^2 in dB) for the following cases: -15, -10, -5, 0, 5, 10, 15. The results are as follows:

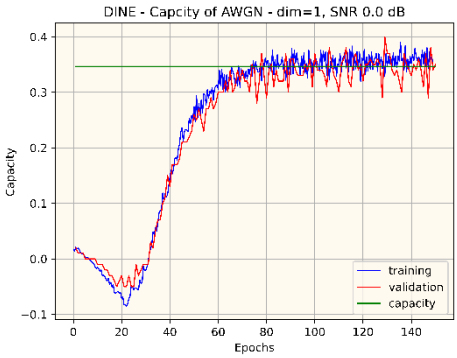


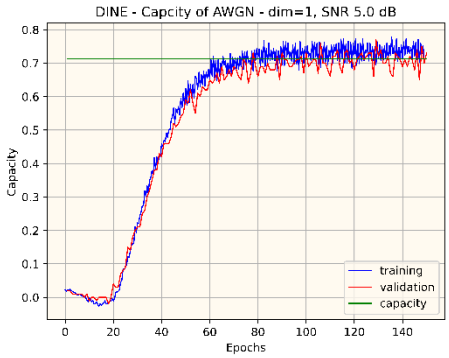
We can deduce that the model is predicting the capacity as for the analytical.

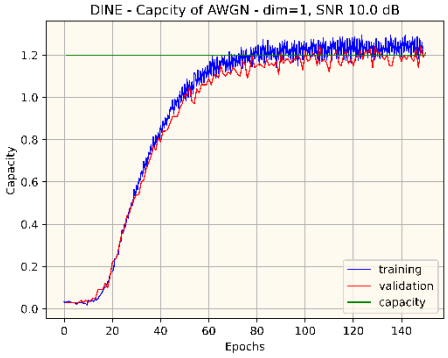


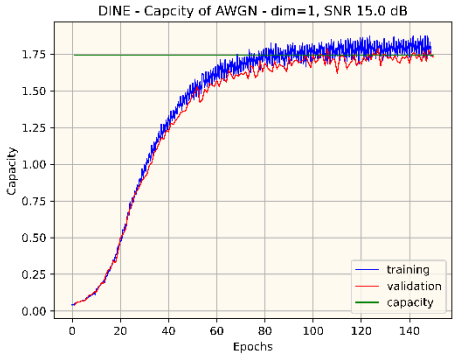




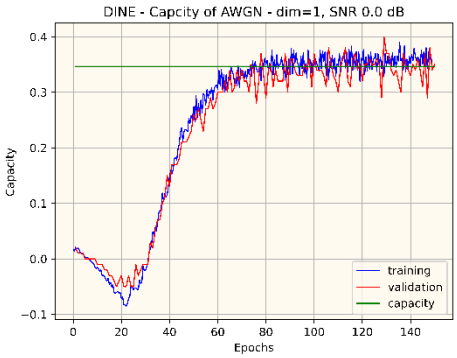


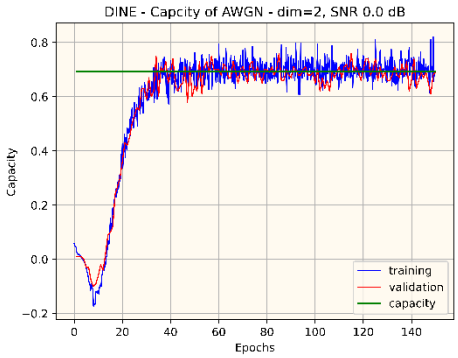


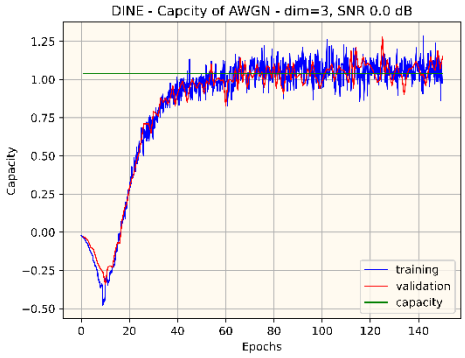


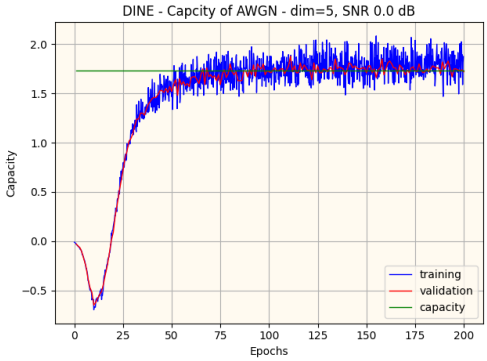


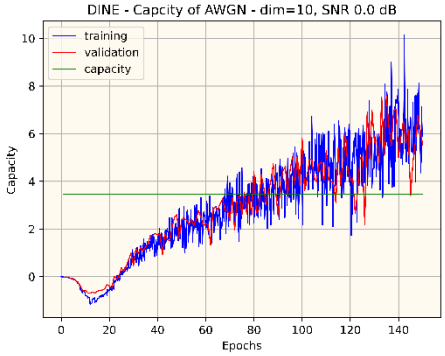
Next, I tested the capacity for higher dimensions of input and noise. As the analytical calculation is conducted as we took K i.i.d. channels and used them in parallel. The capacity equals the regular 1 dimension capacity, multiplied by K.











We can see that for lower dimensions the capacity is accurate, and for higher dimensions, the capacity is very noisy and even deviates from the real analytic capacity.

Predicting $P_{Y|X}$:

In order to predict $P_{Y|X}$ we need to know what will trained $T_\Theta(X,Y)$ will give us.

According to the existence of supermum in Donsker-Varadhan variational represntation lemma, we can conclude that a maximized function to give:

$$D_{KL}(P||Q) = \mathbb{E}_P[T^*(X)] - \log(\mathbb{E}_Q[e^{T^*(X)}]).$$

Is the following function:

$$T^*(x) = \log\left(\frac{P(x)}{Q(x)}\right)$$

Thus, for $D_{Y||X}^{(n)}$ we will get:

$$T^* = \log\left(\frac{dP_{Y^n||X^n}}{dP_{Y^{n-1}||X^{n-1}}dP_{\tilde{Y}}}\right)$$

If we use AWGN process which is independent in time we'll get:

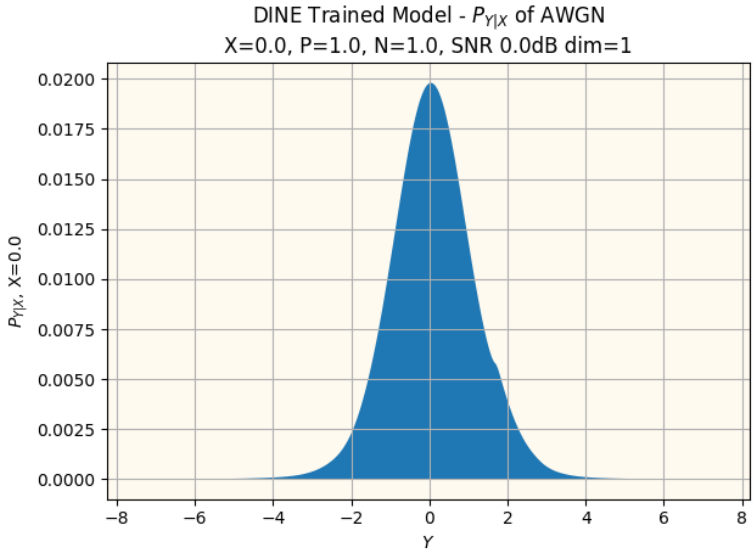
$$T^* = \log\left(\frac{dP_{Y_n|X_n}}{dP_{\tilde{Y}_n}}\right)$$

Because we took uniform distribution for $\tilde{Y}$, we can reconstruct $P_{Y_n|X_n}$ by exponenting the output of F_2 and dividing it by the support of the uniform distribution.

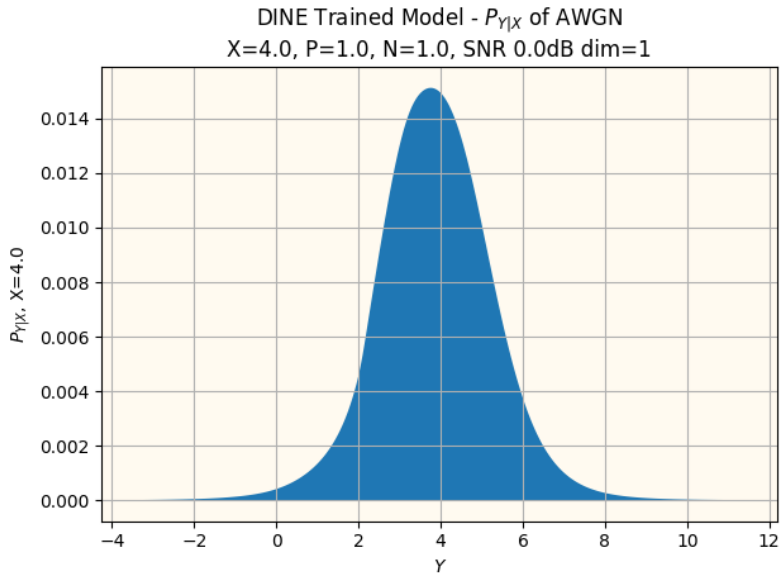
$$P(Y=y|X=x) = Support_{\tilde{Y}} \cdot e^{T^*(X^n,Y^n)}$$

To test it I used the model that trained on $X, V \sim N(0,1)$ for X input, and V noise. We expect to get $P_{Y|X} \sim N(x,1)$, when x is a draw from $x \in \mathcal{X}$.

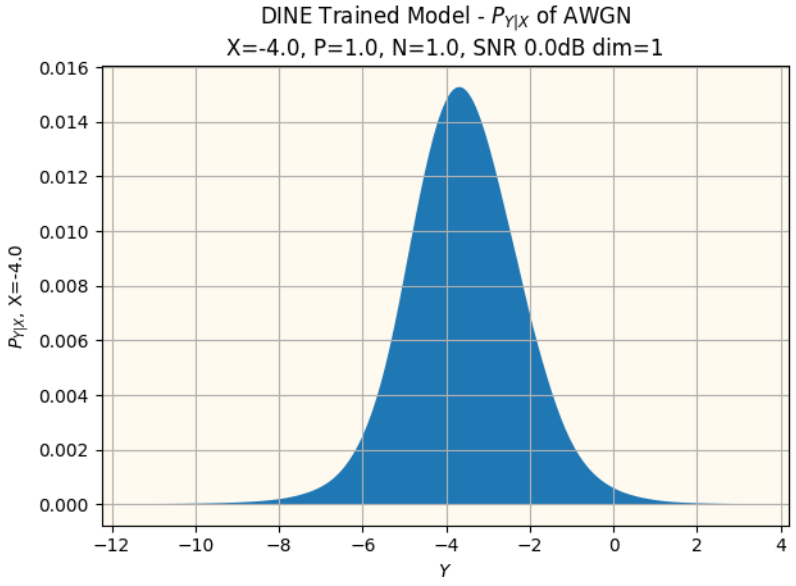
Results: for $x = 0$ we'll get –



For $x = 4$ we'll get –



For $x = -4$ we'll get –



Draft - Planning the Model

$$I(x^n \to y^n) = \lim_{n\to\infty} D_{y\|x}^{(n)} - D_y^{(n)}$$

(2) (1)

The model consists of 2 models which been ↑ maximized by Donsker-Varadhan formula.

(1) $$D_y^{(n)} = \sup_{T:\Omega\to\mathbb{R}} \mathbb{E}[T(y^n)] - \log \mathbb{E}[\exp\{T(y^{n-1}, \tilde{y}_n)\}]$$

(2) $$D_{y\|x}^{(n)} = \sup_{T:\Omega\to\mathbb{R}} \mathbb{E}[T(y^n\|x^n)] - \log \mathbb{E}[\exp\{T(y^{n-1}\|x^{n-1}, \tilde{y}_n)\}]$$

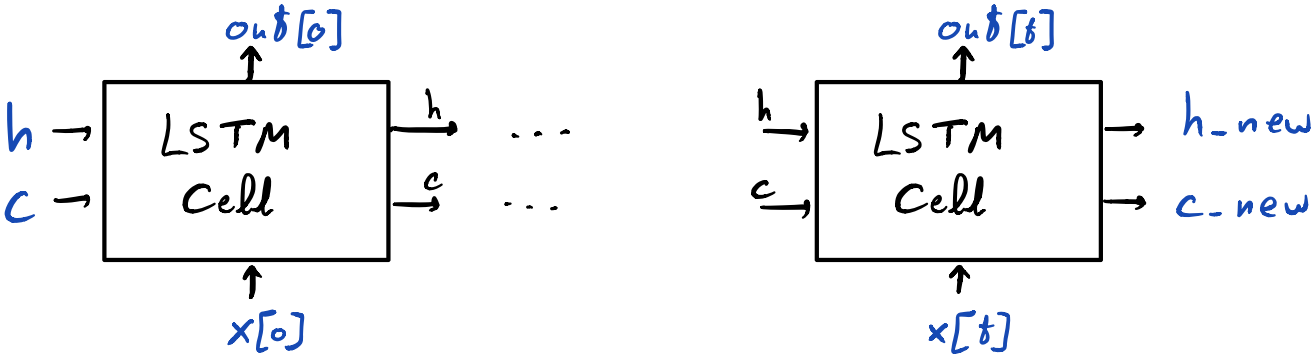
Modified LSTM - Unrolled

3 different modes:

Normal:

Input : input = x , hidden = (h, c)

Output : output = out , hidden = (h_new, c_new)

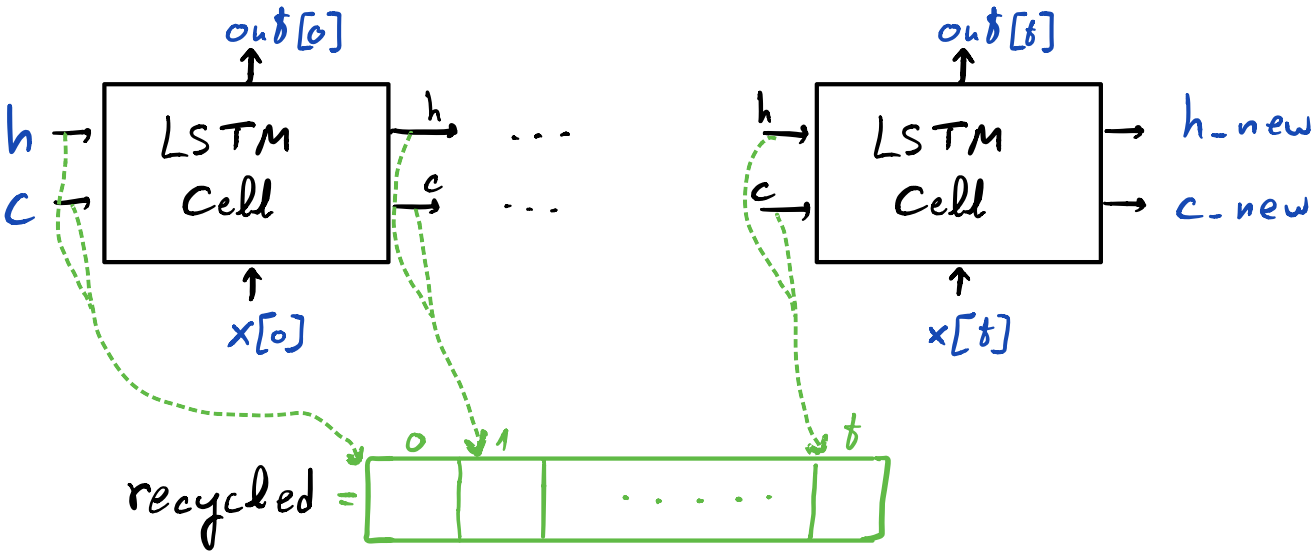


Normal - recycle ON

Each (h,c) will be returned as output.

Input : input = x , hidden = (h, c)

Output : output = out , hidden = (h_new, c_new) , recycled

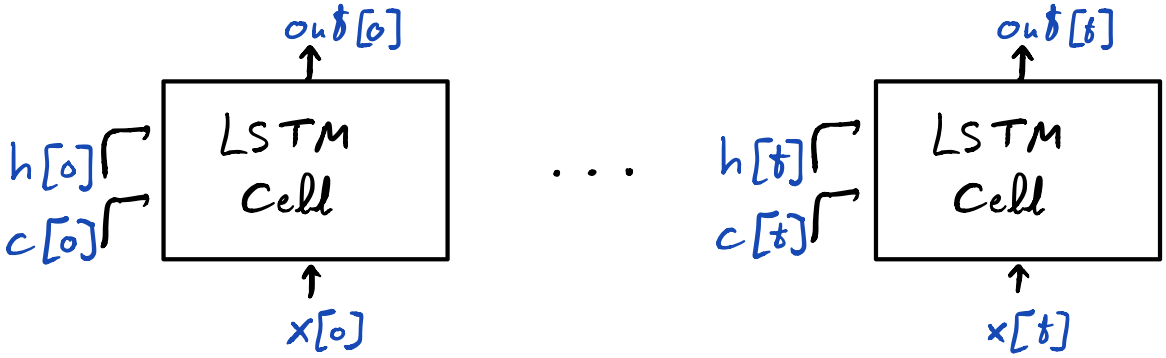


Reused

No connection between LSTM cells.

Input : input = x , all_hidden = (h, c) [t]

Output : output = out

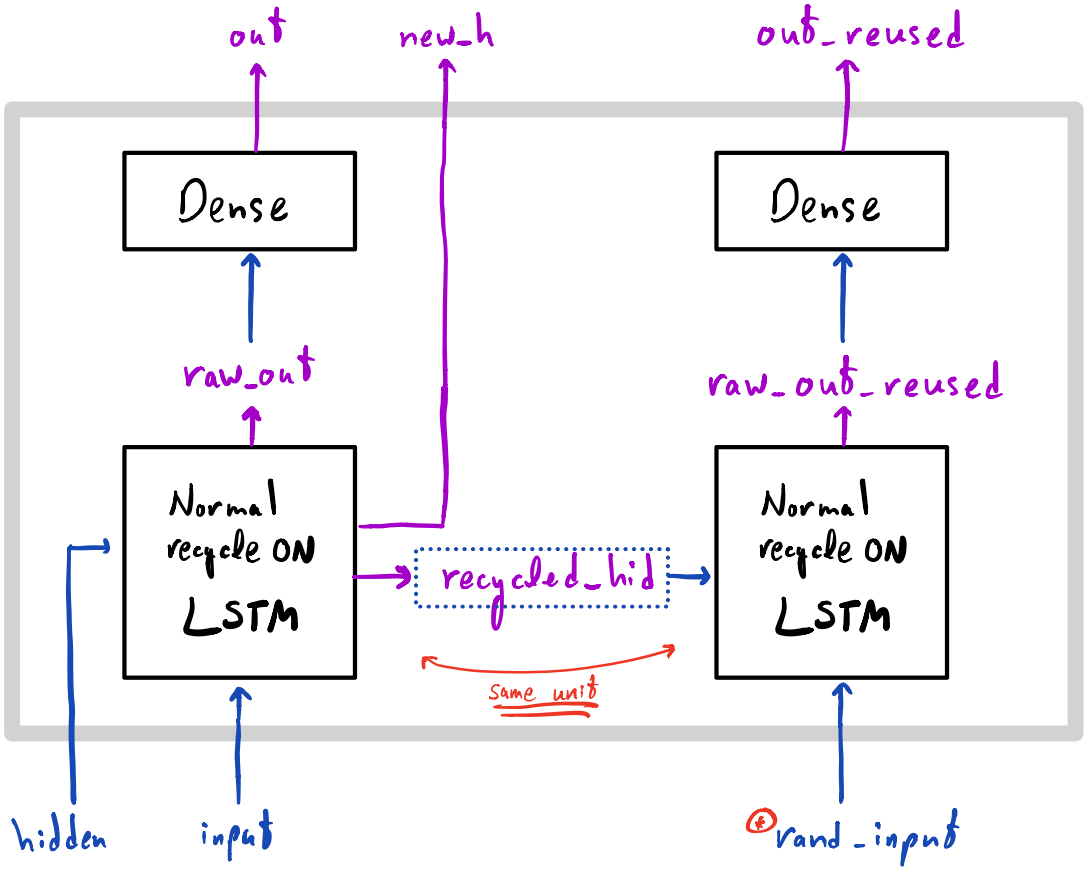


DFiModel - DINE Fi Model

$i \in \{1, 2\}$

Input: input, rand_input, hidden

$$loss = \mathbb{E}[out] - \log \mathbb{E}[\exp\{out\_reused\}]$$



* same x, but y is random-U(min, max)

F_1 - input is only y_i (and rand input...)

F_2 - input is x_i, y_i (and rand input...)