Estimating Information Flow in Deep Neural Networks

Ziv Goldfeld

MIT

56th Allerton Conference on Communication, Control, and Computing
Monticello, Illinois, US

October 4th, 2018

Collaborators: E. van den Berg, K. Greenewald, I. Melnyk, N. Nguyen, B. Kingsbury and Y. Polyanskiy

How do Deep Neural Networks Learn?

- Unprecedented practical success in hosts of tasks

How do Deep Neural Networks Learn?

- Unprecedented practical success in hosts of tasks
- Long way to go theory-wise:

How do Deep Neural Networks Learn?

- Unprecedented practical success in hosts of tasks
- Long way to go theory-wise:
  - What drives the evolution of hidden representations?

How do Deep Neural Networks Learn?

- Unprecedented practical success in hosts of tasks

- Long way to go theory-wise:
  - What drives the evolution of hidden representations?
  - What are properties of learned representations?

How do Deep Neural Networks Learn?

- Unprecedented practical success in hosts of tasks
- Long way to go theory-wise:
  - What drives the evolution of hidden representations?
  - What are properties of learned representations?
  - How fully trained networks process information?

How do Deep Neural Networks Learn?

- Unprecedented practical success in hosts of tasks
- Long way to go theory-wise:
  - What drives the evolution of hidden representations?
  - What are properties of learned representations?
  - How fully trained networks process information?

$\vdots$

How do Deep Neural Networks Learn?

- Unprecedented practical success in hosts of tasks
- Long way to go theory-wise:
  - What drives the evolution of hidden representations?
  - What are properties of learned representations?
  - How fully trained networks process information?

  $\vdots$

- Past attempts to understand effectiveness of deep learning

How do Deep Neural Networks Learn?

- Unprecedented practical success in hosts of tasks
- Long way to go theory-wise:
  - What drives the evolution of hidden representations?
  - What are properties of learned representations?
  - How fully trained networks process information?

⋮

- Past attempts to understand effectiveness of deep learning
  - Optimization in parameter space [Saxe'14, Choromanska'15, Advani'17]

How do Deep Neural Networks Learn?

- Unprecedented practical success in hosts of tasks
- Long way to go theory-wise:
  - What drives the evolution of hidden representations?
  - What are properties of learned representations?
  - How fully trained networks process information?

⋮

- Past attempts to understand effectiveness of deep learning
  - Optimization in parameter space [Saxe'14, Choromanska'15, Advani'17]
  - Classes of efficiently representable functions [Montufar'14, Poggio'17]

How do Deep Neural Networks Learn?

- Unprecedented practical success in hosts of tasks

- Long way to go theory-wise:
  - What drives the evolution of hidden representations?
  - What are properties of learned representations?
  - How fully trained networks process information?

$\vdots$

- Past attempts to understand effectiveness of deep learning
  - Optimization in parameter space [Saxe'14, Choromanska'15, Advani'17]
  - Classes of efficiently representable functions [Montufar'14, Poggio'17]
  - Information theory [Tishby'17, Saxe'18, Gabrié'18]

How do Deep Neural Networks Learn?

- Unprecedented practical success in hosts of tasks

- Long way to go theory-wise:
  - What drives the evolution of hidden representations?
  - What are properties of learned representations?
  - How fully trained networks process information?

⋮

- Past attempts to understand effectiveness of deep learning
  - Optimization in parameter space [Saxe'14, Choromanska'15, Advani'17]
  - Classes of efficiently representable functions [Montufar'14, Poggio'17]
  - **Information theory** [Tishby'17, Saxe'18, Gabrié'18]

How do Deep Neural Networks Learn?

- Unprecedented practical success in hosts of tasks
- Long way to go theory-wise:
  - What drives the evolution of hidden representations?
  - What are properties of learned representations?
  - How fully trained networks process information?

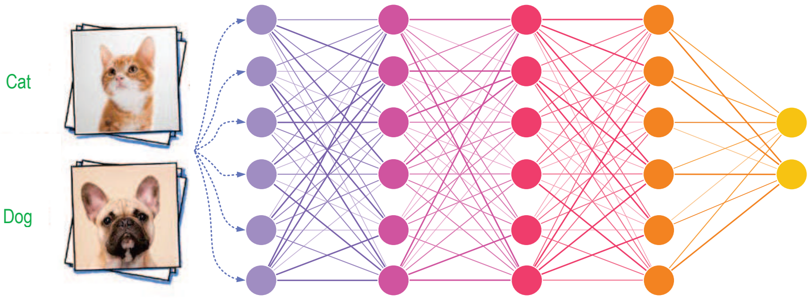
$$\vdots$$

- Past attempts to understand effectiveness of deep learning
  - Optimization in parameter space [Saxe'14, Choromanska'15, Advani'17]
  - Classes of efficiently representable functions [Montufar'14, Poggio'17]
  - **Information theory** [Tishby'17, Saxe'18, Gabrié'18]

★ **Goal:** Explain 'compression' in Information Bottleneck framework

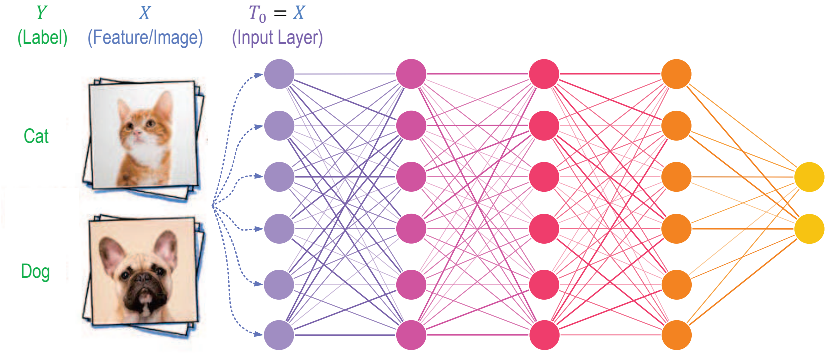
Setup and Preliminaries

Feedforward DNN for Classification:



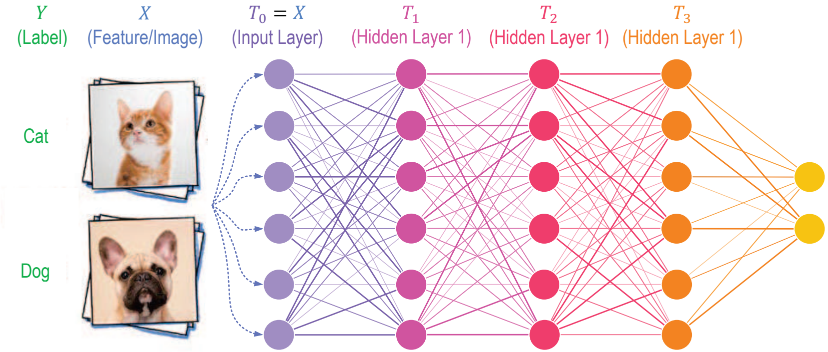
Setup and Preliminaries

Feedforward DNN for Classification:

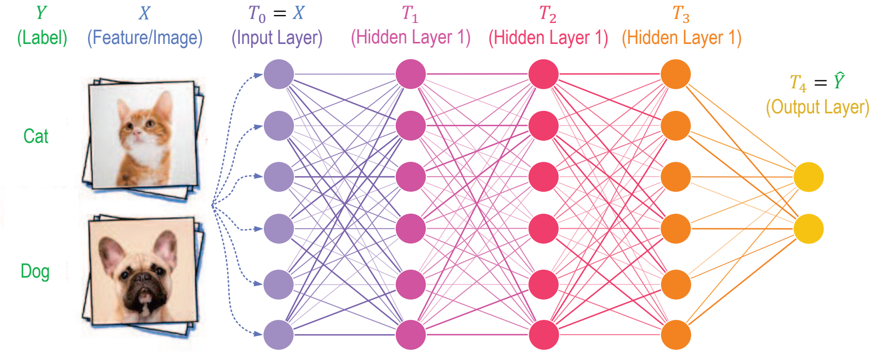


Setup and Preliminaries

Feedforward DNN for Classification:

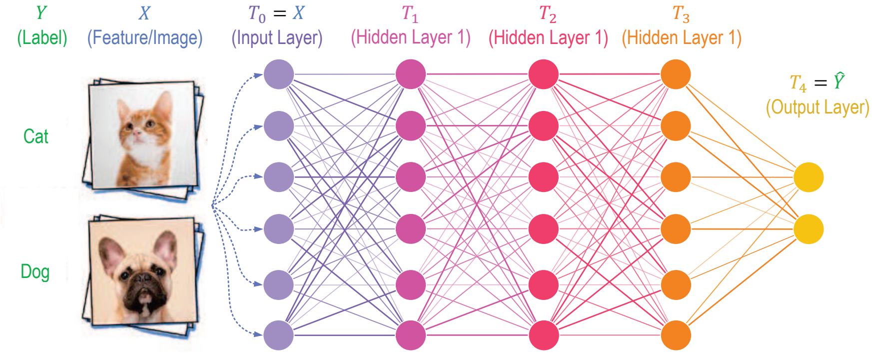


Feedforward DNN for Classification:



Setup and Preliminaries

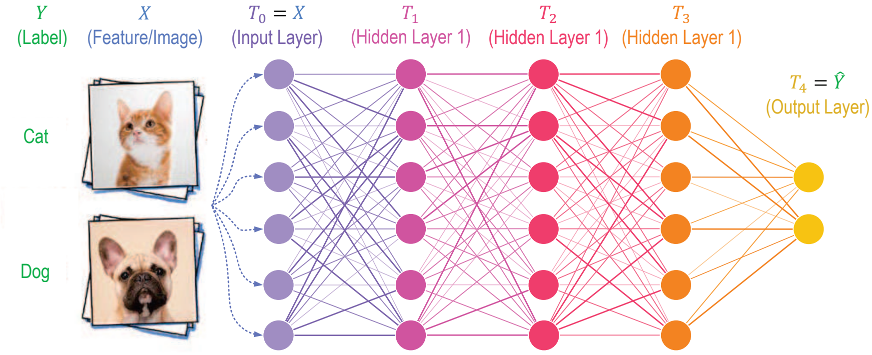
Feedforward DNN for Classification:



- **Deterministic DNN:** $T_\ell = f_\ell(T_{\ell-1})$ (**MLP:** $T_\ell = \sigma(\mathrm{W}_\ell T_{\ell-1} + b_\ell)$)

Setup and Preliminaries

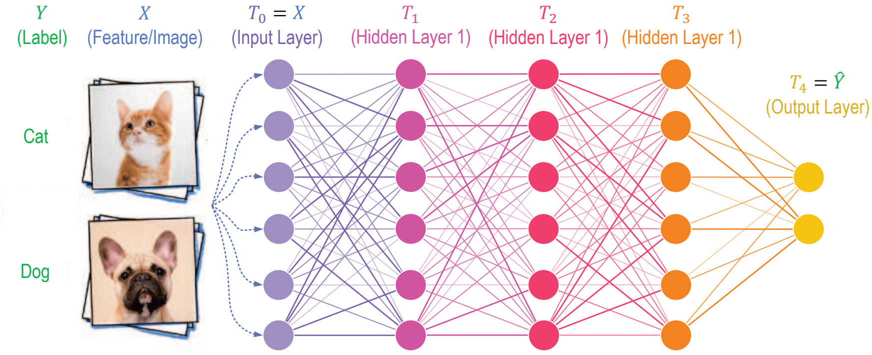
Feedforward DNN for Classification:



- **Deterministic DNN:** $T_\ell = f_\ell(T_{\ell-1})$ (**MLP:** $T_\ell = \sigma(\mathrm{W}_\ell T_{\ell-1} + b_\ell)$)
- **ℓth Hidden Layer Enc & Dec:** $P_{T_\ell|X}$ (enc) and $P_{\hat{Y}|T_\ell}$ (dec)

Setup and Preliminaries

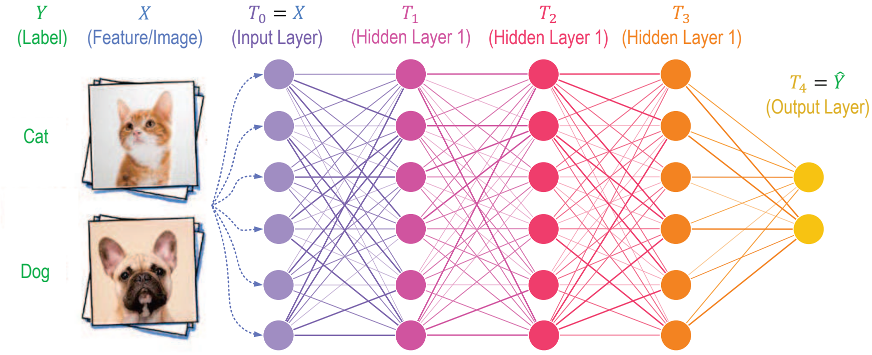
Feedforward DNN for Classification:



- **Deterministic DNN:** $T_\ell = f_\ell(T_{\ell-1})$ (**MLP:** $T_\ell = \sigma(\mathrm{W}_\ell T_{\ell-1} + b_\ell)$)
- **ℓth Hidden Layer Enc & Dec:** $P_{T_\ell|X}$ (enc) and $P_{\hat{Y}|T_\ell}$ (dec)
- **IB Theory:** Track MI pairs $\big(I(X;T_\ell), I(Y;T_\ell)\big)$ (information plane)

Setup and Preliminaries

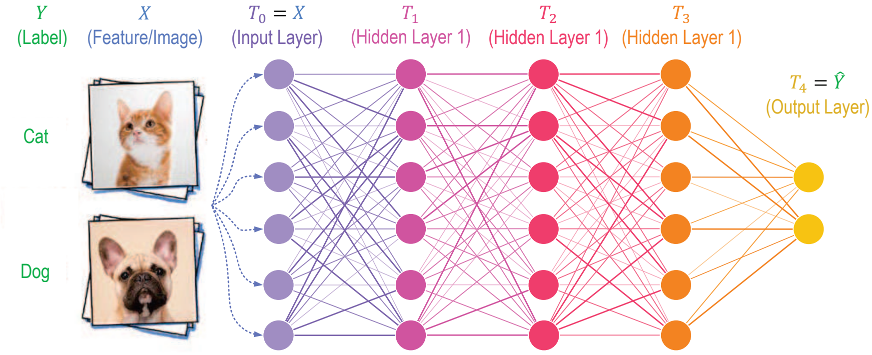
Feedforward DNN for Classification:



IB Theory Claim: Training comprises 2 phases

Setup and Preliminaries

Feedforward DNN for Classification:

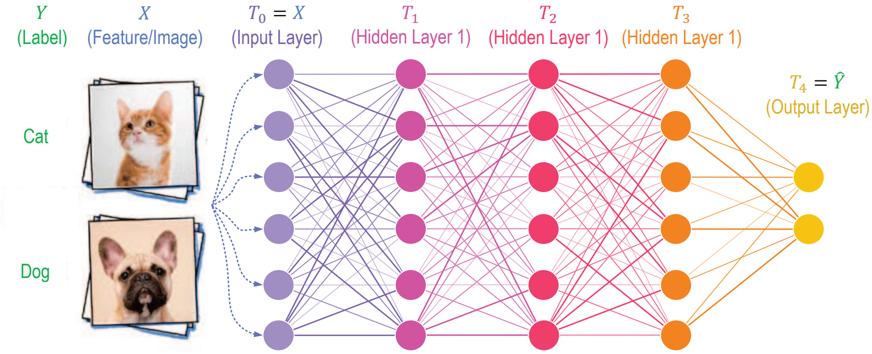


IB Theory Claim: Training comprises 2 phases

- **Fitting:** $I(Y;T_\ell)$ & $I(X;T_\ell)$ rise (short)

Setup and Preliminaries

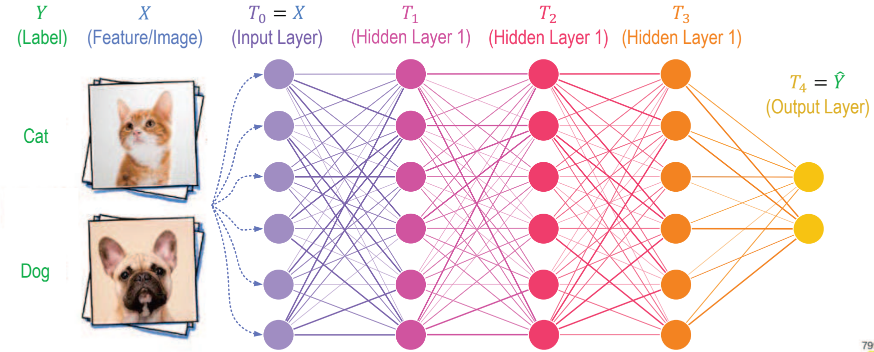
Feedforward DNN for Classification:

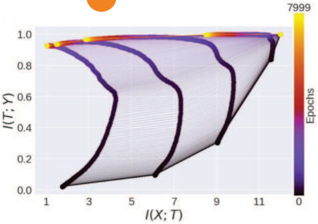


IB Theory Claim: Training comprises 2 phases

- **Fitting:** $I(Y;T_\ell)$ & $I(X;T_\ell)$ rise (short)
- **Compression:** $I(X;T_\ell)$ slowly drops (long)

Feedforward DNN for Classification:



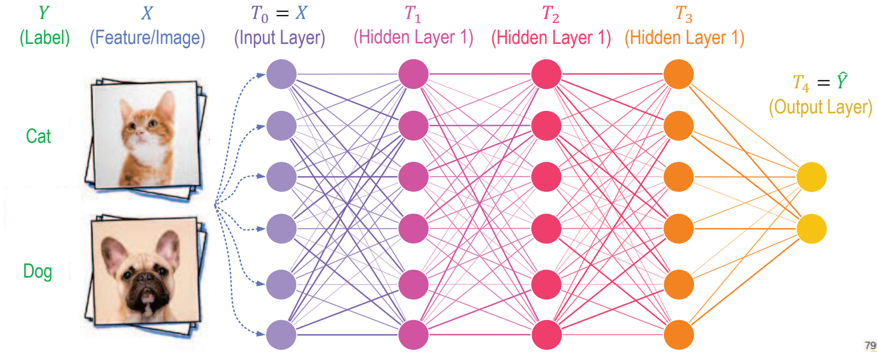


IB Theory Claim: Training comprises 2 phases

- **Fitting:** $I(Y;T_\ell)$ & $I(X;T_\ell)$ rise (short)
- **Compression:** $I(X;T_\ell)$ slowly drops (long)

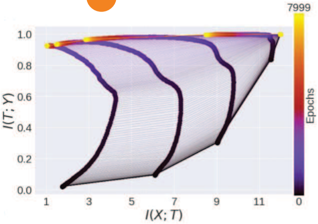
Setup and Preliminaries

Feedforward DNN for Classification:



IB Theory Claim: Training comprises 2 phases

- **Fitting:** $I(Y;T_\ell)$ & $I(X;T_\ell)$ rise (short)
- **Compression:** $\boldsymbol{I(X;T_\ell)}$ slowly drops (long)



Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$$\Longrightarrow \ I(X;T_\ell) \text{ is independent of the DNN parameters}$$

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$$\Longrightarrow \ I(X;T_\ell) \text{ is independent of the DNN parameters}$$

Why?

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$$\Longrightarrow \ I(X;T_\ell) \text{ is independent of the DNN parameters}$$

Why? Formally...

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$$\Longrightarrow \; I(X;T_\ell) \textit{ is independent of the DNN parameters}$$

Why? Formally...

- **Continuous $\boldsymbol{X}$:**

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$\implies I(X;T_\ell)$ *is independent of the DNN parameters*

Why? Formally...

- **Continuous $\boldsymbol{X}$:** $I(X;T_\ell) = h(T_\ell) - h(T_\ell|X)$

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$\implies$ $I(X;T_\ell)$ *is independent of the DNN parameters*

Why? Formally...

- **Continuous $\boldsymbol{X}$:** $I(X;T_\ell) = h(T_\ell) - \boldsymbol{h(T_\ell|X)}$

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$$\implies I(X;T_\ell) \text{ is independent of the DNN parameters}$$

Why? Formally...

- **Continuous $\boldsymbol{X}$:** $I(X;T_\ell) = h(T_\ell) - \boldsymbol{h\big(\tilde{f}_\ell(X)\big|X\big)}$

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$\implies$ $I(X;T_\ell)$ *is independent of the DNN parameters*

Why? Formally...

- **Continuous $\boldsymbol{X}$:** $I(X;T_\ell) = h(T_\ell) - \underbrace{\boldsymbol{h(\tilde{f}_\ell(X)|X)}}_{=-\infty}$

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$$\implies I(X;T_\ell) \text{ is independent of the DNN parameters}$$

Why? Formally...

- **Continuous X:** $$I(X;T_\ell) = h(T_\ell) - h\big(\tilde{f}_\ell(X)\big|X\big) = \infty$$

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$\implies I(X;T_\ell)$ *is independent of the DNN parameters*

Why? Formally...

- **Continuous X:** $I(X;T_\ell) = h(T_\ell) - h\big(\tilde{f}_\ell(X)\big|X\big) = \infty$
- **Discrete X:**

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$\Longrightarrow$ $I(X;T_\ell)$ *is independent of the DNN parameters*

Why? Formally...

- **Continuous $\boldsymbol{X}$:** $$I(X;T_\ell) = h(T_\ell) - h\big(\tilde{f}_\ell(X)\big|X\big) = \infty$$
- **Discrete $\boldsymbol{X}$:** The map $X \mapsto T_\ell$ is injective*

$\star$ For almost all weight matrices and bias vectors

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$$\implies I(X;T_\ell) \textit{ is independent of the DNN parameters}$$

Why? Formally...

- **Continuous $\boldsymbol{X}$:** $I(X;T_\ell) = h(T_\ell) - h\big(\tilde{f}_\ell(X)\big|X\big) = \infty$
- **Discrete $\boldsymbol{X}$:** The map $X \mapsto T_\ell$ is injective* $\implies I(X;T_\ell) = H(X)$

$\star$ For almost all weight matrices and bias vectors

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$$\implies I(X;T_\ell) \text{ is } \textbf{\textit{independent of the DNN parameters}}$$

Why? Formally...

- **Continuous $\boldsymbol{X}$:** $I(X;T_\ell) = h(T_\ell) - h\big(\tilde{f}_\ell(X)\big|X\big) = \infty$
- **Discrete $\boldsymbol{X}$:** The map $X \mapsto T_\ell$ is injective* $\implies I(X;T_\ell) = \boldsymbol{H(X)}$

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$\Longrightarrow I(X;T_\ell)$ *is* ***independent of the DNN parameters***

Why? Formally...

- **Continuous $\boldsymbol{X}$:** $I(X;T_\ell) = h(T_\ell) - h\big(\tilde{f}_\ell(X)\big|X\big) = \boldsymbol{\infty}$
- **Discrete $\boldsymbol{X}$:** The map $X \mapsto T_\ell$ is injective* $\Longrightarrow I(X;T_\ell) = \boldsymbol{H(X)}$

Intuition:

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$\implies I(X;T_\ell)$ *is* ***independent of the DNN parameters***

Why? Formally...

- **Continuous $\boldsymbol{X}$:** $I(X;T_\ell) = h(T_\ell) - h\big(\tilde{f}_\ell(X)\big|X\big) = \boldsymbol{\infty}$
- **Discrete $\boldsymbol{X}$:** The map $X \mapsto T_\ell$ is injective* $\implies I(X;T_\ell) = \boldsymbol{H(X)}$

Intuition: Encoding all info. about X is arbitrarily fine variations of T_ℓ

Meaningless Mutual Information

Observation

Det. DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid)

$\Longrightarrow$ $I(X;T_\ell)$ *is* ***independent of the DNN parameters***

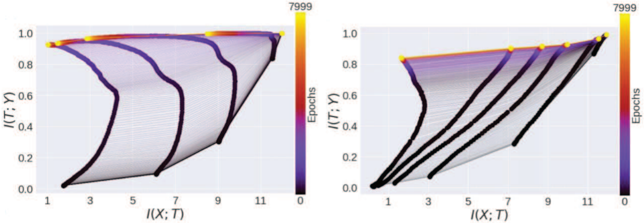
Why? Formally...

- **Continuous $\boldsymbol{X}$:** $I(X;T_\ell) = h(T_\ell) - h\big(\tilde{f}_\ell(X)\big|X\big) = \infty$
- **Discrete $\boldsymbol{X}$:** The map $X \mapsto T_\ell$ is injective* $\Longrightarrow$ $I(X;T_\ell) = \boldsymbol{H(X)}$

Intuition: Encoding all info. about X is arbitrarily fine variations of T_ℓ

Past Works:

[Schwartz-Ziv&Tishby'17, Saxe *et al.* '18]



What is going on here?

- Plots via binning-based estimator of $I(X;T_\ell)$, for $X \sim \mathrm{Unif}(\text{dataset})$

What is going on here?

- Plots via binning-based estimator of $I(X;T_\ell)$, for $X \sim \mathsf{Unif}(\text{dataset})$

$$\implies \text{Plotted values are} \quad I\big(X;\mathsf{Bin}(T_\ell)\big)$$

What is going on here?

- Plots via binning-based estimator of $I(X;T_\ell)$, for $X \sim \mathsf{Unif}(\text{dataset})$

$$\implies \text{ Plotted values are } \quad I\big(X;\mathsf{Bin}(T_\ell)\big) \overset{??}{\approx} I(X;T_\ell)$$

What is going on here?

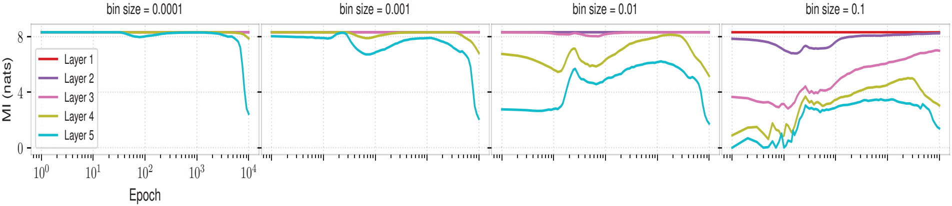
- Plots via binning-based estimator of $I(X;T_\ell)$, for $X \sim \mathsf{Unif}(\mathsf{dataset})$

$\implies$ Plotted values are $\quad I(X;\mathsf{Bin}(T_\ell)) \overset{??}{\approx} I(X;T_\ell)$ **No!**

What is going on here?

- Plots via binning-based estimator of $I(X;T_\ell)$, for $X \sim \mathsf{Unif}(\text{dataset})$

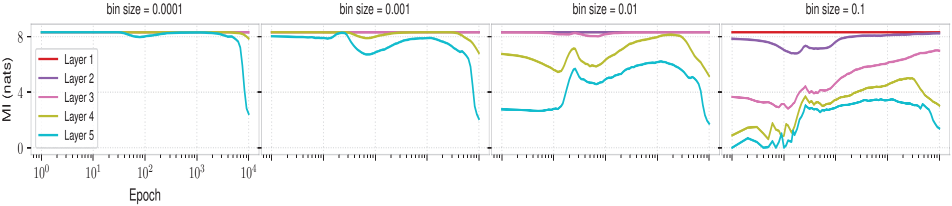
$$\Longrightarrow \text{ Plotted values are } \quad I(X;\mathsf{Bin}(T_\ell)) \overset{??}{\approx} I(X;T_\ell) \qquad \textbf{No!}$$



What is going on here?

- Plots via binning-based estimator of $I(X;T_\ell)$, for $X \sim \mathsf{Unif}(\text{dataset})$

$$\implies \text{ Plotted values are } \quad I(X;\mathsf{Bin}(T_\ell)) \overset{??}{\approx} I(X;T_\ell) \qquad \textbf{No!}$$

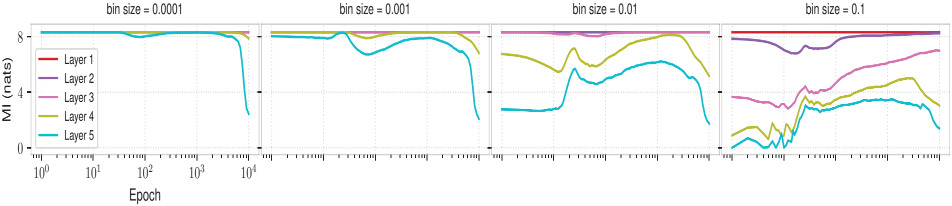


- Smaller bins $\implies$ Closer to truth: $\quad I(X;T_\ell) = \ln(2^{12}) \approx 8.31$

What is going on here?

- Plots via binning-based estimator of $I(X;T_\ell)$, for $X \sim \mathrm{Unif}(\text{dataset})$

$$\implies \text{Plotted values are} \quad I(X;\mathrm{Bin}(T_\ell)) \overset{??}{\approx} I(X;T_\ell) \qquad \textbf{No!}$$

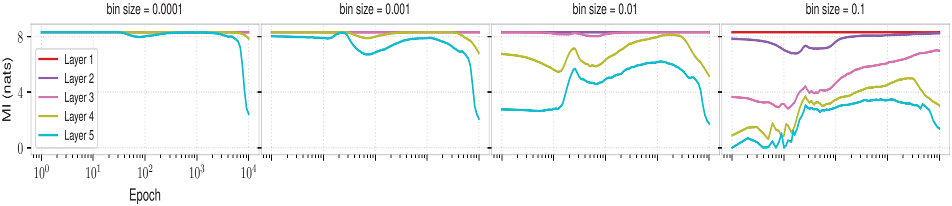


- Smaller bins $\implies$ Closer to truth: $I(X;T_\ell) = \ln(2^{12}) \approx 8.31$
- Binning introduces "noise" into estimator (not present in the DNN)

What is going on here?

- Plots via binning-based estimator of $I(X;T_\ell)$, for $X \sim \text{Unif}(\text{dataset})$

$$\implies \text{Plotted values are} \quad I(X;\text{Bin}(T_\ell)) \overset{??}{\approx} I(X;T_\ell) \qquad \textbf{No!}$$

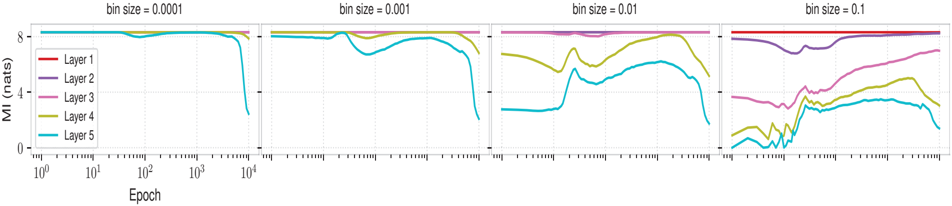


- Smaller bins $\implies$ Closer to truth: $I(X;T_\ell) = \ln(2^{12}) \approx 8.31$
- Binning introduces "noise" into estimator (not present in the DNN)
- Plots showing estimation errors

What is going on here?

- Plots via binning-based estimator of $I(X;T_\ell)$, for $X \sim \text{Unif}(\text{dataset})$

$$\Longrightarrow \text{ Plotted values are } \quad I(X;\text{Bin}(T_\ell)) \overset{??}{\approx} I(X;T_\ell) \qquad \textbf{No!}$$



- Smaller bins $\Longrightarrow$ Closer to truth: $I(X;T_\ell) = \ln(2^{12}) \approx 8.31$
- Binning introduces "noise" into estimator (not present in the DNN)
- Plots showing estimation errors

⊛ **Real Problem:** $I(X;T_\ell)$ is meaningless for studying the DNN

Noisy Deep Neural Networks

Proposed Fix: Inject (small) Gaussian noise to neurons' output

Noisy Deep Neural Networks

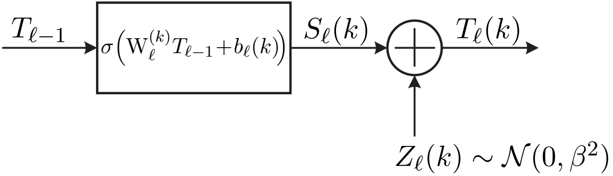
Proposed Fix: Inject (small) Gaussian noise to neurons' output

- **Formally:** $T_\ell = f_\ell(T_{\ell-1}) + Z_\ell$, where $Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$

Noisy Deep Neural Networks

Proposed Fix: Inject (small) Gaussian noise to neurons' output

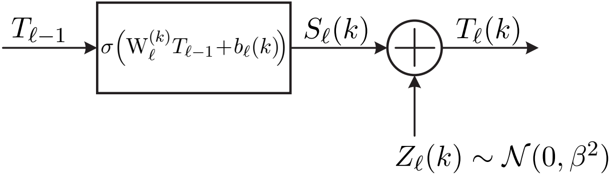
- **Formally:** $T_\ell = f_\ell(T_{\ell-1}) + Z_\ell$, where $Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$



Noisy Deep Neural Networks

Proposed Fix: Inject (small) Gaussian noise to neurons' output

- **Formally:** $T_\ell = f_\ell(T_{\ell-1}) + Z_\ell$, where $Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$

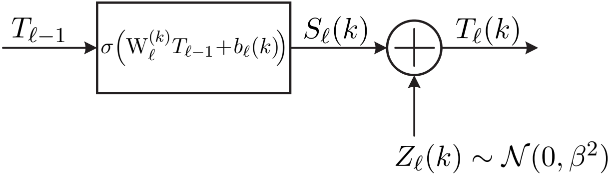


$\implies X \mapsto T_\ell$ is a **parametrized channel** that depends on DNN param.!

Noisy Deep Neural Networks

Proposed Fix: Inject (small) Gaussian noise to neurons' output

- **Formally:** $T_\ell = f_\ell(T_{\ell-1}) + Z_\ell$, where $Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$



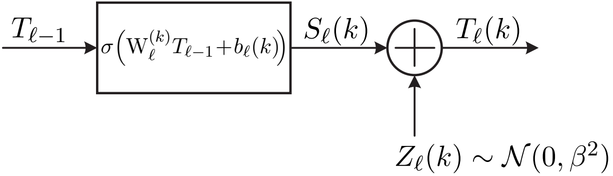
$\Longrightarrow$ $X \mapsto T_\ell$ is a **parametrized channel** that depends on DNN param.!

- **Operational Perspective:**

Noisy Deep Neural Networks

Proposed Fix: Inject (small) Gaussian noise to neurons' output

- **Formally:** $T_\ell = f_\ell(T_{\ell-1}) + Z_\ell$, where $Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$



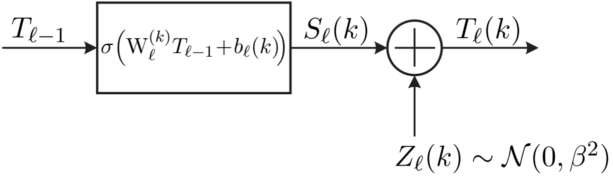
$\Longrightarrow$ $X \mapsto T_\ell$ is a **parametrized channel** that depends on DNN param.!

- **Operational Perspective:**
  - Performance & learned representations similar to det. DNNs ($\beta \approx 10^{-1}$)

Noisy Deep Neural Networks

Proposed Fix: Inject (small) Gaussian noise to neurons' output

- **Formally:** $T_\ell = f_\ell(T_{\ell-1}) + Z_\ell$, where $Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$



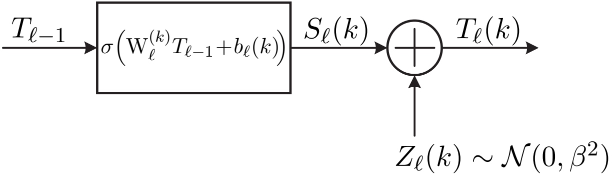
$\implies$ $X \mapsto T_\ell$ is a **parametrized channel** that depends on DNN param.!

- **Operational Perspective:**
  - Performance & learned representations similar to det. DNNs ($\beta \approx 10^{-1}$)
  - Noise masks fine variations – MI represents relevant/distinguishable info.

Noisy Deep Neural Networks

Proposed Fix: Inject (small) Gaussian noise to neurons' output

- **Formally:** $T_\ell = f_\ell(T_{\ell-1}) + Z_\ell$, where $Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$



$\Longrightarrow$ $X \mapsto T_\ell$ is a **parametrized channel** that depends on DNN param.!

- **Operational Perspective:**
  - Performance & learned representations similar to det. DNNs ($\beta \approx 10^{-1}$)
  - Noise masks fine variations – MI represents relevant/distinguishable info.
  - Dropout & quantized DNNs widely used in practice $\approx$ internal noise

Mutual Information (Estimation) in Noisy DNNs

- **Layer ℓ:** Denote $S_\ell \triangleq f_\ell(T_{\ell-1})$

Mutual Information (Estimation) in Noisy DNNs

- **Layer ℓ:** Denote $S_\ell \triangleq f_\ell(T_{\ell-1}) \implies T_\ell = S_\ell + Z_\ell,\ Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$

Mutual Information (Estimation) in Noisy DNNs

- **Layer ℓ:** Denote $S_\ell \triangleq f_\ell(T_{\ell-1}) \implies T_\ell = S_\ell + Z_\ell,\ Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$
- **Assume:** $X \sim \mathsf{Unif}(\mathcal{X})$, where $\mathcal{X} \triangleq \{x_i\}_{i=1}^m$ is empirical dataset

Mutual Information (Estimation) in Noisy DNNs

- **Layer ℓ:** Denote $S_\ell \triangleq f_\ell(T_{\ell-1}) \implies T_\ell = S_\ell + Z_\ell,\ Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$
- **Assume:** $X \sim \mathsf{Unif}(\mathcal{X})$, where $\mathcal{X} \triangleq \{x_i\}_{i=1}^m$ is empirical dataset
- **Mutual Information:** $I(X;T_\ell) = h(T_\ell) - \frac{1}{m}\sum_{i=1}^m h(T_\ell | X = x_i)$

Mutual Information (Estimation) in Noisy DNNs

- **Layer ℓ:** Denote $S_\ell \triangleq f_\ell(T_{\ell-1}) \implies T_\ell = S_\ell + Z_\ell,\ Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$
- **Assume:** $X \sim$ Unif$(\mathcal{X})$, where $\mathcal{X} \triangleq \{x_i\}_{i=1}^m$ is empirical dataset
- **Mutual Information:** $I(X;T_\ell) = h(T_\ell) - \frac{1}{m}\sum_{i=1}^m h(T_\ell | X = x_i)$

⊛ Distribution of S_ℓ is **extremely** complicated to compute/evaluate

Mutual Information (Estimation) in Noisy DNNs

- **Layer ℓ:** Denote $S_\ell \triangleq f_\ell(T_{\ell-1}) \implies T_\ell = S_\ell + Z_\ell,\ Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$
- **Assume:** $X \sim \mathsf{Unif}(\mathcal{X})$, where $\mathcal{X} \triangleq \{x_i\}_{i=1}^m$ is empirical dataset
- **Mutual Information:** $I(X;T_\ell) = h(T_\ell) - \frac{1}{m}\sum_{i=1}^m h(T_\ell|X=x_i)$

⊛ Distribution of S_ℓ is **extremely** complicated to compute/evaluate

⊛ But, P_{S_ℓ} and $P_{S_\ell|X=x_i}$ are **easily** sampled from via DNN fwd. pass

Mutual Information (Estimation) in Noisy DNNs

- **Layer ℓ:** Denote $S_\ell \triangleq f_\ell(T_{\ell-1}) \implies T_\ell = S_\ell + Z_\ell,\ Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$
- **Assume:** $X \sim$ Unif$(\mathcal{X})$, where $\mathcal{X} \triangleq \{x_i\}_{i=1}^m$ is empirical dataset
- **Mutual Information:** $I(X;T_\ell) = h(T_\ell) - \frac{1}{m}\sum_{i=1}^m h(T_\ell|X = x_i)$

⊛ Distribution of S_ℓ is **extremely** complicated to compute/evaluate

⊛ But, P_{S_ℓ} and $P_{S_\ell|X=x_i}$ are **easily** sampled from via DNN fwd. pass

$\implies$ Estimate MI from samples & Exploit noisy DNN structure

Mutual Information (Estimation) in Noisy DNNs

- **Layer ℓ:** Denote $S_\ell \triangleq f_\ell(T_{\ell-1}) \implies T_\ell = S_\ell + Z_\ell,\ Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$
- **Assume:** $X \sim \text{Unif}(\mathcal{X})$, where $\mathcal{X} \triangleq \{x_i\}_{i=1}^m$ is empirical dataset
- **Mutual Information:** $I(X;T_\ell) = h(T_\ell) - \frac{1}{m}\sum_{i=1}^m h(T_\ell | X = x_i)$

⊛ Distribution of S_ℓ is **extremely** complicated to compute/evaluate

⊛ But, P_{S_ℓ} and $P_{S_\ell|X=x_i}$ are **easily** sampled from via DNN fwd. pass

$\implies$ Estimate MI from samples & Exploit noisy DNN structure

Differential Entropy Estimation under Gaussian Convolutions

Estimate $h(S+Z)$ using n i.i.d. samples from $P_S \in \mathcal{F}_d$ (nonparametric class) and knowing that $Z \sim \mathcal{N}(0, \beta^2 \mathrm{I}_d)$ independent of S.

Mutual Information (Estimation) in Noisy DNNs

- **Layer ℓ:** Denote $S_\ell \triangleq f_\ell(T_{\ell-1}) \implies T_\ell = S_\ell + Z_\ell,\ Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$
- **Assume:** $X \sim$ Unif$(\mathcal{X})$, where $\mathcal{X} \triangleq \{x_i\}_{i=1}^m$ is empirical dataset
- **Mutual Information:** $I(X;T_\ell) = h(T_\ell) - \frac{1}{m}\sum_{i=1}^m h(T_\ell | X = x_i)$

⊛ Distribution of S_ℓ is **extremely** complicated to compute/evaluate

⊛ But, P_{S_ℓ} and $P_{S_\ell|X=x_i}$ are **easily** sampled from via DNN fwd. pass

$\implies$ Estimate MI from samples & Exploit noisy DNN structure

Differential Entropy Estimation under Gaussian Convolutions

Estimate $h(S+Z)$ using n i.i.d. samples from $P_S \in \mathcal{F}_d$ (nonparametric class) and knowing that $Z \sim \mathcal{N}(0, \beta^2 \mathrm{I}_d)$ independent of S.

Results [ZG-Greenewald-Polyanskiy'18]:

Mutual Information (Estimation) in Noisy DNNs

- **Layer ℓ:** Denote $S_\ell \triangleq f_\ell(T_{\ell-1}) \implies T_\ell = S_\ell + Z_\ell,\ Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$
- **Assume:** $X \sim$ Unif$(\mathcal{X})$, where $\mathcal{X} \triangleq \{x_i\}_{i=1}^m$ is empirical dataset
- **Mutual Information:** $I(X;T_\ell) = h(T_\ell) - \frac{1}{m}\sum_{i=1}^m h(T_\ell | X = x_i)$

⊛ Distribution of S_ℓ is **extremely** complicated to compute/evaluate

⊛ But, P_{S_ℓ} and $P_{S_\ell|X=x_i}$ are **easily** sampled from via DNN fwd. pass

$\implies$ Estimate MI from samples & Exploit noisy DNN structure

Differential Entropy Estimation under Gaussian Convolutions

Estimate $h(S+Z)$ using n i.i.d. samples from $P_S \in \mathcal{F}_d$ (nonparametric class) and knowing that $Z \sim \mathcal{N}(0, \beta^2 \mathrm{I}_d)$ independent of S.

Results [ZG-Greenewald-Polyanskiy'18]:

- Sample complexity is exponential in d

Mutual Information (Estimation) in Noisy DNNs

- **Layer ℓ:** Denote $S_\ell \triangleq f_\ell(T_{\ell-1}) \implies T_\ell = S_\ell + Z_\ell,\ Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I})$
- **Assume:** $X \sim \mathsf{Unif}(\mathcal{X})$, where $\mathcal{X} \triangleq \{x_i\}_{i=1}^m$ is empirical dataset
- **Mutual Information:** $I(X;T_\ell) = h(T_\ell) - \frac{1}{m}\sum_{i=1}^m h(T_\ell|X=x_i)$

⊛ Distribution of S_ℓ is **extremely** complicated to compute/evaluate

⊛ But, P_{S_ℓ} and $P_{S_\ell|X=x_i}$ are **easily** sampled from via DNN fwd. pass

$\implies$ Estimate MI from samples & Exploit noisy DNN structure

Differential Entropy Estimation under Gaussian Convolutions

Estimate $h(S+Z)$ using n i.i.d. samples from $P_S \in \mathcal{F}_d$ (nonparametric class) and knowing that $Z \sim \mathcal{N}(0, \beta^2 \mathrm{I}_d)$ independent of S.

Results [ZG-Greenewald-Polyanskiy'18]:

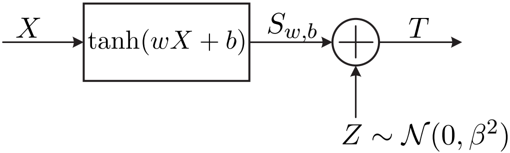
- Sample complexity is exponential in d
- Absolute-error minimax risk is $O\left((\log n)^{d/4}/\sqrt{n}\right)$ (all const. explicit)

$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:

$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:

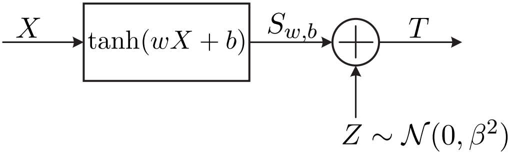


$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:

- **Input:** $X \sim \mathsf{Unif}\left(\mathcal{X}_{-1} \cup \mathcal{X}_1\right)$

  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$

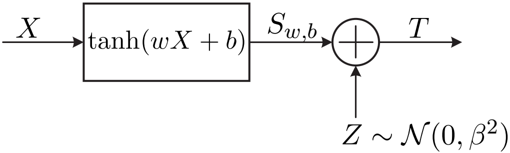


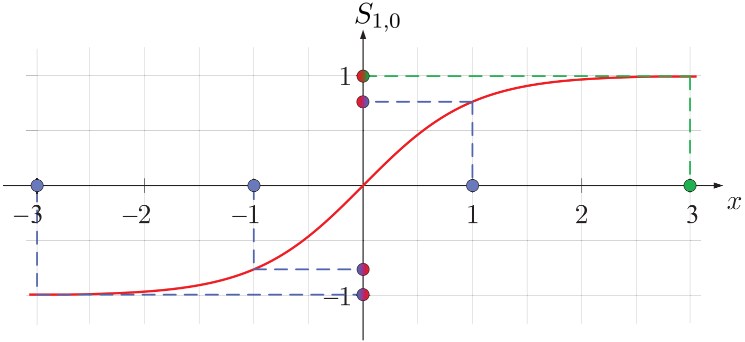
$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:

- **Input:** $X \sim \mathsf{Unif}\big(\mathcal{X}_{-1} \cup \mathcal{X}_1\big)$

  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$



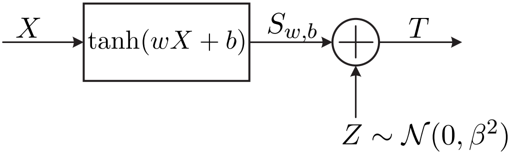


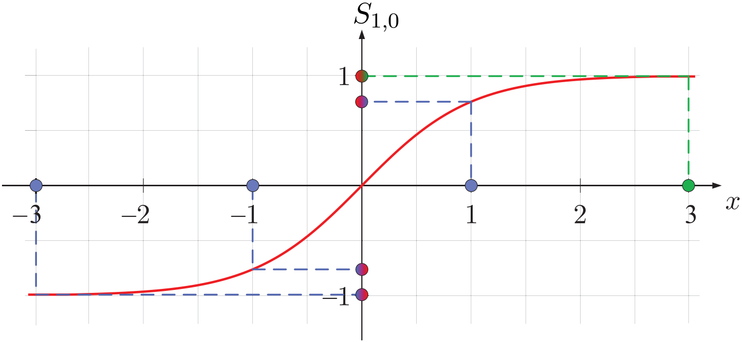
$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:

- **Input:** $X \sim \mathsf{Unif}\left(\mathcal{X}_{-1} \cup \mathcal{X}_1\right)$

  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$



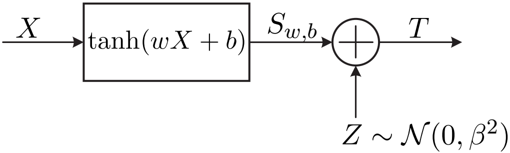


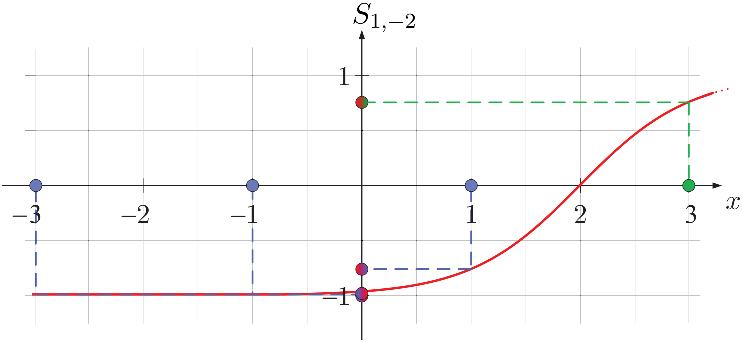
Move $\tanh$ center $x=2$ ($\Longleftrightarrow$ $b=-2$)

$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:

- **Input:** $X \sim \mathsf{Unif}\left(\mathcal{X}_{-1} \cup \mathcal{X}_1\right)$
  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$



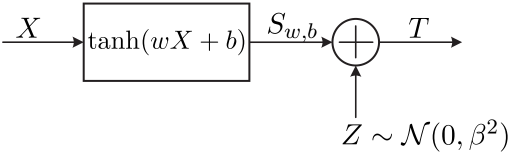


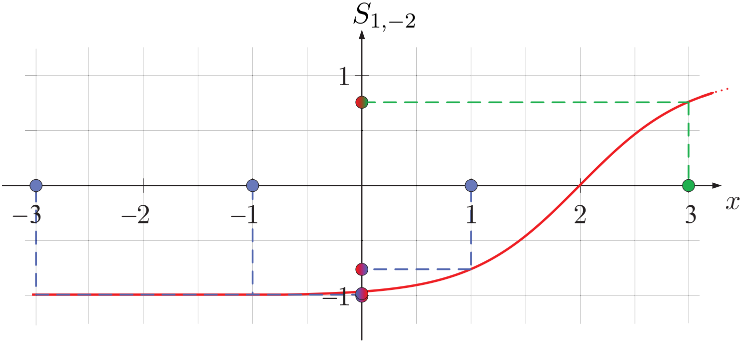
$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:

- **Input:** $X \sim \text{Unif}\big(\mathcal{X}_{-1} \cup \mathcal{X}_1\big)$

  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$





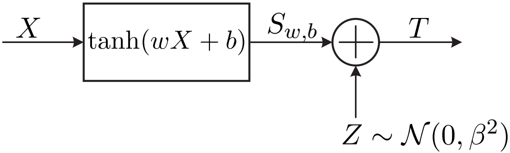
⊛ Sharpen $\tanh$ transition ($\iff$ increase w and keep $b=-2w$)

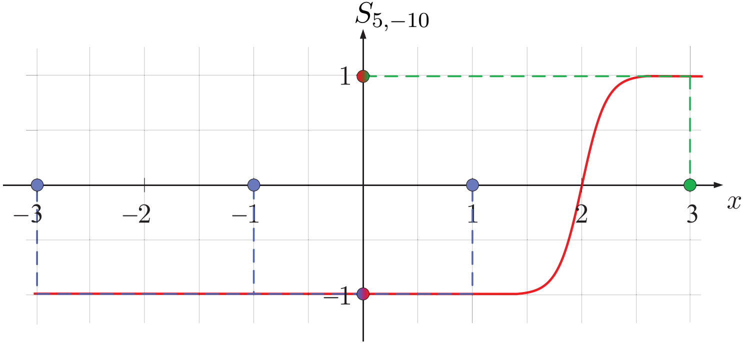
$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:

- **Input:** $X \sim \mathsf{Unif}\left(\mathcal{X}_{-1} \cup \mathcal{X}_1\right)$

  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$

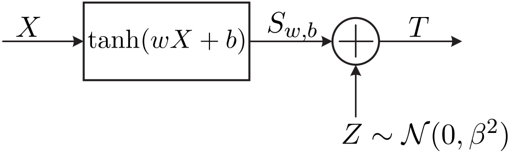


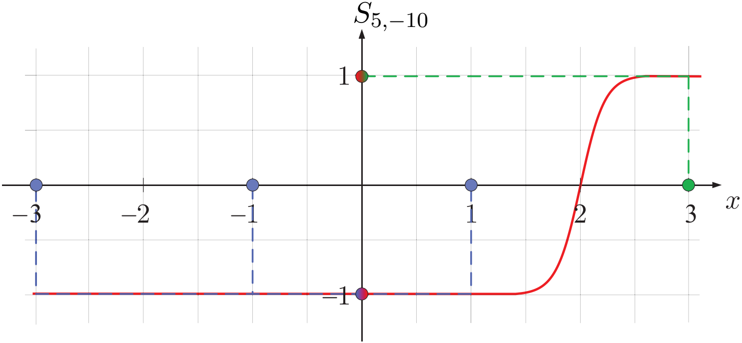


$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:

- **Input:** $X \sim \mathsf{Unif}\left(\mathcal{X}_{-1} \cup \mathcal{X}_1\right)$
  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$

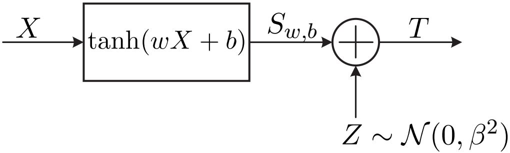




✓ Correct classification performance

$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

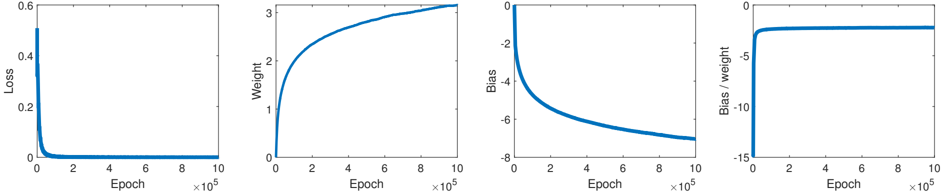
Single Neuron Classification:



- **Input:** $X \sim \mathsf{Unif}\left(\mathcal{X}_{-1} \cup \mathcal{X}_1\right)$

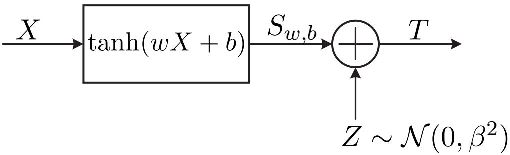
  $\mathcal{X}_{-1} \triangleq \{-3, -1, 1\}$, $\mathcal{X}_1 \triangleq \{3\}$

- **Empirical Results:**



$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:



- **Input:** $X \sim \mathsf{Unif}(\mathcal{X}_{-1} \cup \mathcal{X}_1)$

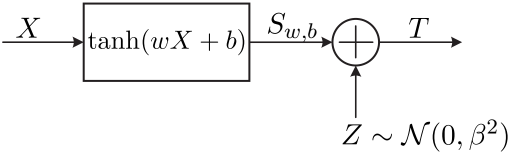
  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$

- **Mutual Information:**

  $I(X;T)$

$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:



- **Input:** $X \sim \mathsf{Unif}\big(\mathcal{X}_{-1} \cup \mathcal{X}_1\big)$

  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$

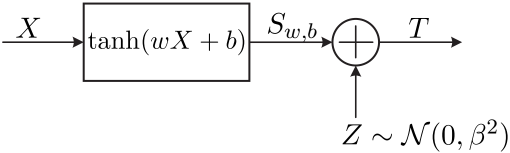
- **Mutual Information:**

  $$I(X;T) = I(X; S_{w,b} + Z)$$

$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:

- **Input:** $X \sim \mathsf{Unif}\big(\mathcal{X}_{-1} \cup \mathcal{X}_1\big)$
  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$

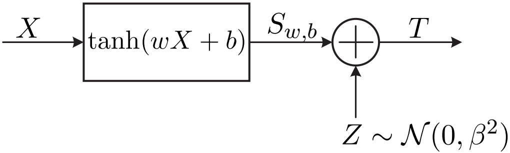


- **Mutual Information:**

$$I(X;T) = I(X;S_{w,b}+Z) = I\big(\tanh(wX+b);S_{w,b}+Z\big)$$

$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:



- **Input:** $X \sim \mathsf{Unif}\big(\mathcal{X}_{-1} \cup \mathcal{X}_1\big)$

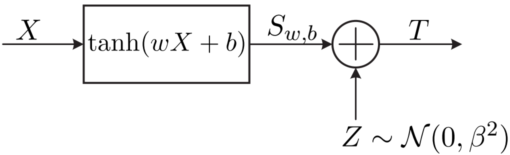
  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$

- **Mutual Information:**

$$I(X;T) = I(X;S_{w,b}+Z) = I\big(\tanh(wX+b);S_{w,b}+Z\big) = I(S_{w,b};S_{w,b}+Z)$$

$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:



- **Input:** $X \sim \mathsf{Unif}\left(\mathcal{X}_{-1} \cup \mathcal{X}_1\right)$

  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$

- **Mutual Information:**

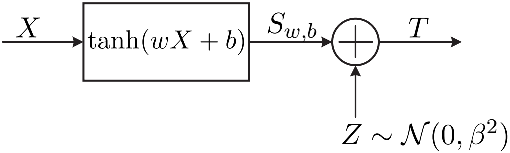
$$I(X;T) = I(X;S_{w,b}+Z) = I\big(\tanh(wX+b);S_{w,b}+Z\big) = I(S_{w,b};S_{w,b}+Z)$$

$\Longrightarrow$ $I(X;T)$ is the aggregate info. transmitted over AWGN w. symbols

$$\mathcal{S}_{w,b} \triangleq \big\{\tanh(-3w+b),\tanh(-w+b),\tanh(w+b),\tanh(3w+b)\big\}$$

$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:



- **Input:** $X \sim \mathsf{Unif}\left(\mathcal{X}_{-1} \cup \mathcal{X}_1\right)$

  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$

- **Mutual Information:**

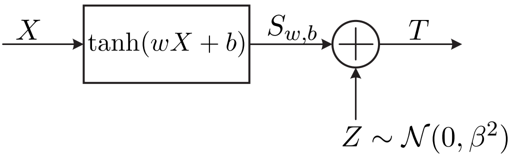
$$I(X;T) = I(X;S_{w,b}+Z) = I\big(\tanh(wX+b);S_{w,b}+Z\big) = I(S_{w,b};S_{w,b}+Z)$$

$\Longrightarrow$ $I(X;T)$ is the aggregate info. transmitted over AWGN w. symbols

$$\mathcal{S}_{w,b} \triangleq \{\tanh(-3w+b), \tanh(-w+b), \tanh(w+b), \tanh(3w+b)\} \longrightarrow \{\pm 1\}$$

$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:



- **Input:** $X \sim \mathsf{Unif}\left(\mathcal{X}_{-1} \cup \mathcal{X}_1\right)$

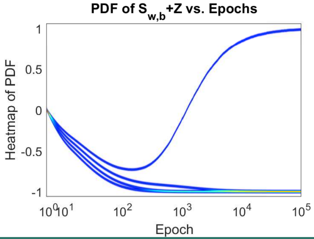
  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$

- **Mutual Information:**

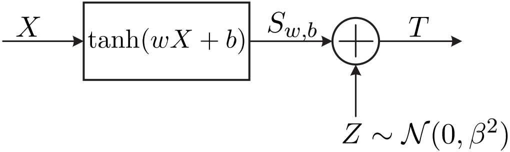
$$I(X;T)=I(X;S_{w,b}+Z)=I\big(\tanh(wX+b);S_{w,b}+Z\big)=I(S_{w,b};S_{w,b}+Z)$$

$\Longrightarrow$ $I(X;T)$ is the aggregate info. transmitted over AWGN w. symbols

$$\mathcal{S}_{w,b} \triangleq \{\tanh(-3w+b),\tanh(-w+b),\tanh(w+b),\tanh(3w+b)\} \longrightarrow \{\pm 1\}$$



Single Neuron Classification:

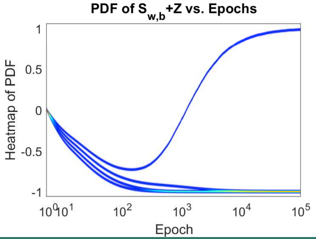


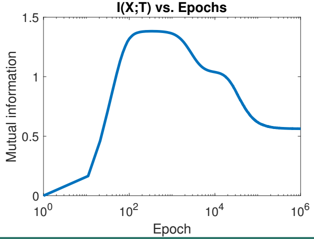
- **Input:** $X \sim \mathsf{Unif}\left(\mathcal{X}_{-1} \cup \mathcal{X}_1\right)$
  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$
- **Mutual Information:**

$$I(X;T)=I(X;S_{w,b}+Z)=I\big(\tanh(wX+b);S_{w,b}+Z\big)=I(S_{w,b};S_{w,b}+Z)$$

$\Longrightarrow$ $I(X;T)$ is the aggregate info. transmitted over AWGN w. symbols

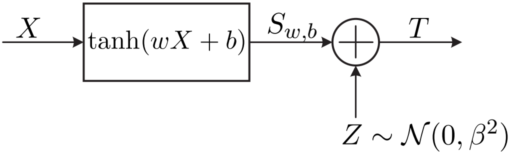
$$\mathcal{S}_{w,b}\triangleq\{\tanh(-3w+b),\tanh(-w+b),\tanh(w+b),\tanh(3w+b)\}\longrightarrow\{\pm1\}$$





$I(X;T_\ell)$ Dynamics - Illustrative Minimal Example

Single Neuron Classification:



- **Input:** $X \sim \mathsf{Unif}(\mathcal{X}_{-1} \cup \mathcal{X}_1)$

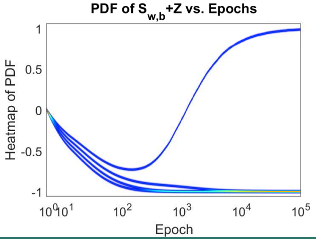
  $\mathcal{X}_{-1} \triangleq \{-3,-1,1\}$, $\mathcal{X}_1 \triangleq \{3\}$

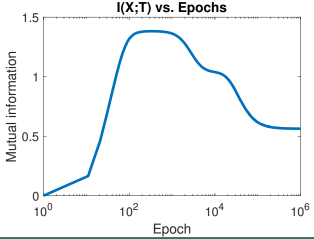
- **Mutual Information:**

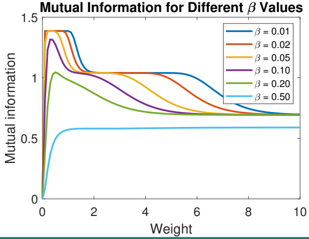
$$I(X;T) = I(X;S_{w,b}+Z) = I\big(\tanh(wX+b);S_{w,b}+Z\big) = I(S_{w,b};S_{w,b}+Z)$$

$\Longrightarrow$ $I(X;T)$ is the aggregate info. transmitted over AWGN w. symbols

$$\mathcal{S}_{w,b} \triangleq \{\tanh(-3w+b),\tanh(-w+b),\tanh(w+b),\tanh(3w+b)\} \longrightarrow \{\pm 1\}$$







Clustering of Representations - Larger Networks

Noisy version of DNN from [Schwartz-Ziv&Tishby'17]:

Clustering of Representations - Larger Networks

Noisy version of DNN from [Schwartz-Ziv&Tishby'17]:

- **Binary Classification:** 12-bit input & 12–10–7–5–4–3–2 MLP arch.

Clustering of Representations - Larger Networks

Noisy version of DNN from [Schwartz-Ziv&Tishby'17]:

- **Binary Classification:** 12-bit input & 12–10–7–5–4–3–2 MLP arch.
- **Noise std.:** Set to $\beta = 0.1$

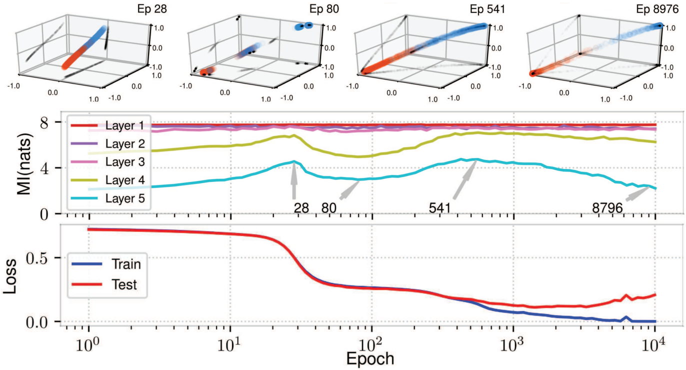
Clustering of Representations - Larger Networks

Noisy version of DNN from [Schwartz-Ziv&Tishby'17]:

- **Binary Classification:** 12-bit input & 12–10–7–5–4–3–2 MLP arch.
- **Noise std.:** Set to $\beta = 0.1$

Noisy version of DNN from [Schwartz-Ziv&Tishby'17]:

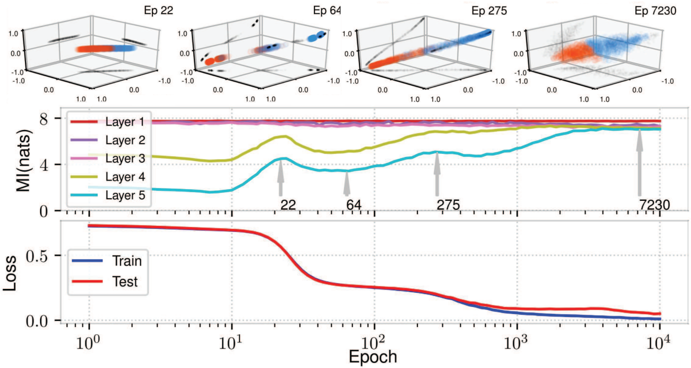
- **Binary Classification:** 12-bit input & 12–10–7–5–4–3–2 MLP arch.
- **Noise std.:** Set to $\beta = 0.1$



Clustering of Representations - Larger Networks

Noisy version of DNN from [Schwartz-Ziv&Tishby'17]:

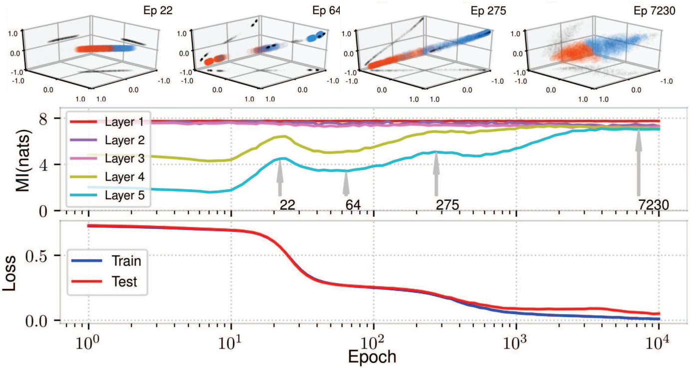
- **Binary Classification:** 12-bit input & 12–10–7–5–4–3–2 MLP arch.
- **Noise std.:** Set to $\beta = 0.1$



Clustering of Representations - Larger Networks

Noisy version of DNN from [Schwartz-Ziv&Tishby'17]:

- **Binary Classification:** 12-bit input & 12–10–7–5–4–3–2 MLP arch.
- **Noise std.:** Set to $\beta = 0.1$



$\Longrightarrow$ Compression of $I(X;T_\ell)$ driven by clustering of representations

Circling back to Deterministic DNNs

- $I(X;T_\ell)$ is constant

Circling back to Deterministic DNNs

- $I(X;T_\ell)$ is constant $\implies$ Doesn't measure clustering

Circling back to Deterministic DNNs

- $I(X;T_\ell)$ is constant $\implies$ Doesn't measure clustering
- Alternative measures for clustering (det. and noisy DNNs):

Circling back to Deterministic DNNs

- $I(X;T_\ell)$ is constant $\implies$ Doesn't measure clustering
- Alternative measures for clustering (det. and noisy DNNs):
  - Scatter plots (up to 3D layers)

Circling back to Deterministic DNNs

- $I(X;T_\ell)$ is constant $\implies$ Doesn't measure clustering
- Alternative measures for clustering (det. and noisy DNNs):
  - Scatter plots (up to 3D layers)
  - Within-class & In-between-class pairwise distance distribution

Circling back to Deterministic DNNs

- $I(X;T_\ell)$ is constant $\implies$ Doesn't measure clustering
- Alternative measures for clustering (det. and noisy DNNs):
  - Scatter plots (up to 3D layers)
  - Within-class & In-between-class pairwise distance distribution
  - Binned entropy $H\big(\mathsf{Bin}(T_\ell)\big)$

Circling back to Deterministic DNNs

- $I(X;T_\ell)$ is constant $\implies$ Doesn't measure clustering
- Alternative measures for clustering (det. and noisy DNNs):
  - Scatter plots (up to 3D layers)
  - Within-class & In-between-class pairwise distance distribution
  - Binned entropy $H\big(\mathsf{Bin}(T_\ell)\big)$
- **Noisy DNNs:** $I(X;T_\ell)$ and $H\big(\mathsf{Bin}(T_\ell)\big)$ highly correlated!*

Circling back to Deterministic DNNs

- $I(X;T_\ell)$ is constant $\implies$ Doesn't measure clustering
- Alternative measures for clustering (det. and noisy DNNs):
  - Scatter plots (up to 3D layers)
  - Within-class & In-between-class pairwise distance distribution
  - Binned entropy $H\big(\mathsf{Bin}(T_\ell)\big)$
- **Noisy DNNs:** $I(X;T_\ell)$ and $H\big(\mathsf{Bin}(T_\ell)\big)$ highly correlated![*]
- **Det. DNNs:** $H\big(\mathsf{Bin}(T_\ell)\big)$ compresses (resolution wrt bins size)

Circling back to Deterministic DNNs

- $I(X;T_\ell)$ is constant $\implies$ Doesn't measure clustering
- Alternative measures for clustering (det. and noisy DNNs):
  - Scatter plots (up to 3D layers)
  - Within-class & In-between-class pairwise distance distribution
  - Binned entropy $H\big(\mathsf{Bin}(T_\ell)\big)$
- **Noisy DNNs:** $I(X;T_\ell)$ and $H\big(\mathsf{Bin}(T_\ell)\big)$ highly correlated!*
- **Det. DNNs:** $H\big(\mathsf{Bin}(T_\ell)\big)$ compresses (resolution wrt bins size)

⊛ **Past Works:** Estimated $I(X;T_\ell)$ by $I\big(X;\mathsf{Bin}(T_\ell)\big)$

Circling back to Deterministic DNNs

- $I(X;T_\ell)$ is constant $\implies$ Doesn't measure clustering
- Alternative measures for clustering (det. and noisy DNNs):
  - Scatter plots (up to 3D layers)
  - Within-class & In-between-class pairwise distance distribution
  - Binned entropy $H\big(\mathsf{Bin}(T_\ell)\big)$
- **Noisy DNNs:** $I(X;T_\ell)$ and $H\big(\mathsf{Bin}(T_\ell)\big)$ highly correlated!*
- **Det. DNNs:** $H\big(\mathsf{Bin}(T_\ell)\big)$ compresses (resolution wrt bins size)

⊛ **Past Works:** Estimated $I(X;T_\ell)$ by $I\big(X;\mathsf{Bin}(T_\ell)\big) = H\big(\mathsf{Bin}(T_\ell)\big)$

Circling back to Deterministic DNNs

- $I(X;T_\ell)$ is constant $\implies$ Doesn't measure clustering
- Alternative measures for clustering (det. and noisy DNNs):
  - Scatter plots (up to 3D layers)
  - Within-class & In-between-class pairwise distance distribution
  - Binned entropy $H\big(\text{Bin}(T_\ell)\big)$
- **Noisy DNNs:** $I(X;T_\ell)$ and $H\big(\text{Bin}(T_\ell)\big)$ highly correlated!*
- **Det. DNNs:** $H\big(\text{Bin}(T_\ell)\big)$ compresses (resolution wrt bins size)

⊛ **Past Works:** Estimated $I(X;T_\ell)$ by $I\big(X;\text{Bin}(T_\ell)\big) = H\big(\text{Bin}(T_\ell)\big)$

✗ Incapable of accurately estimating MI values

Circling back to Deterministic DNNs

- $I(X;T_\ell)$ is constant $\implies$ Doesn't measure clustering
- Alternative measures for clustering (det. and noisy DNNs):
  - Scatter plots (up to 3D layers)
  - Within-class & In-between-class pairwise distance distribution
  - Binned entropy $H\big(\mathsf{Bin}(T_\ell)\big)$
- **Noisy DNNs:** $I(X;T_\ell)$ and $H\big(\mathsf{Bin}(T_\ell)\big)$ highly correlated!*
- **Det. DNNs:** $H\big(\mathsf{Bin}(T_\ell)\big)$ compresses (resolution wrt bins size)

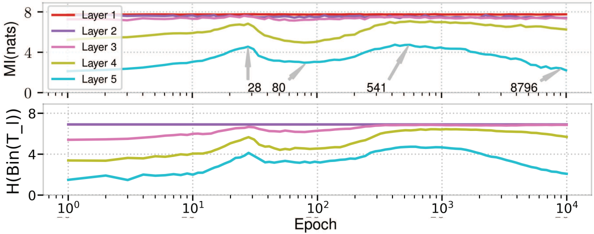
⊛ **Past Works:** Estimated $I(X;T_\ell)$ by $I\big(X;\mathsf{Bin}(T_\ell)\big) = H\big(\mathsf{Bin}(T_\ell)\big)$

✗ Incapable of accurately estimating MI values

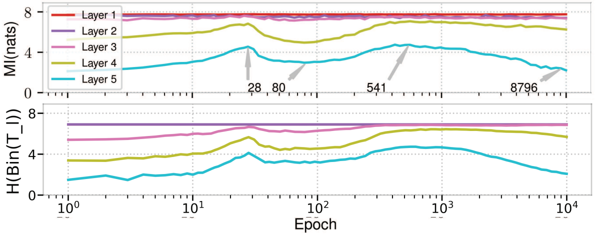
✓ Still, simple to compute & follows MI in tracking clustering!

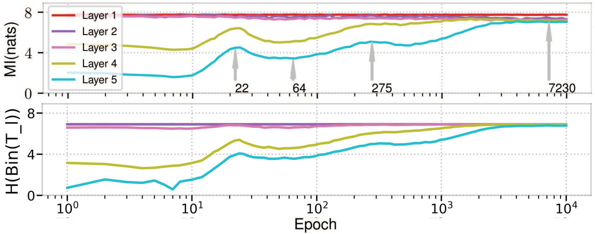
Circling back to Deterministic DNNs (Cntd.)

Comparing to Previously Shown MI Plots:

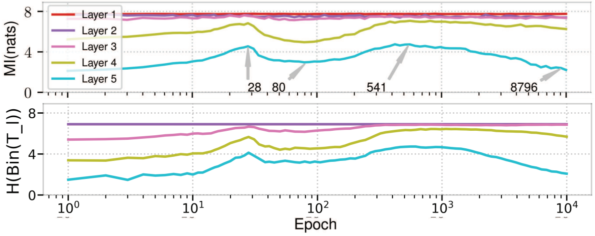


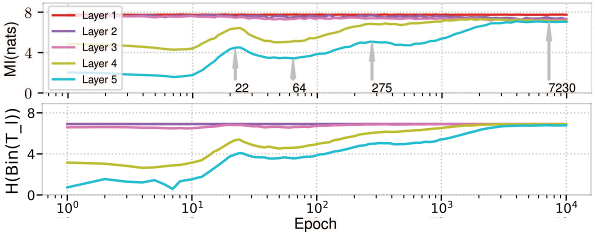
Comparing to Previously Shown MI Plots:





Comparing to Previously Shown MI Plots:





$\Longrightarrow$ Past works we not showing MI but clustering (via binned-MI)!

Summary

- **Reexamined Information Bottleneck Compression:**

Summary

- **Reexamined Information Bottleneck Compression:**
  - $I(X;T)$ fluctuations in det. DNNs are theoretically impossible

Summary

- **Reexamined Information Bottleneck Compression:**
  - $I(X;T)$ fluctuations in det. DNNs are theoretically impossible
  - Yes, past works presented $I(X;T)$ dynamics during training

Summary

- **Reexamined Information Bottleneck Compression:**
  - $I(X;T)$ fluctuations in det. DNNs are theoretically impossible
  - Yes, past works presented $I(X;T)$ dynamics during training
- **Noisy DNN Framework:** Studying IT quantities over DNNs

Summary

- **Reexamined Information Bottleneck Compression:**
  - $I(X;T)$ fluctuations in det. DNNs are theoretically impossible
  - Yes, past works presented $I(X;T)$ dynamics during training
- **Noisy DNN Framework:** Studying IT quantities over DNNs
  - Toolkit for accurate MI estimation over this framework

Summary

- **Reexamined Information Bottleneck Compression:**
  - $I(X;T)$ fluctuations in det. DNNs are theoretically impossible
  - Yes, past works presented $I(X;T)$ dynamics during training
- **Noisy DNN Framework:** Studying IT quantities over DNNs
  - Toolkit for accurate MI estimation over this framework
  - Clustering of the learned representations is the source of compression

Summary

- **Reexamined Information Bottleneck Compression:**
  - $I(X;T)$ fluctuations in det. DNNs are theoretically impossible
  - Yes, past works presented $I(X;T)$ dynamics during training
- **Noisy DNN Framework:** Studying IT quantities over DNNs
  - Toolkit for accurate MI estimation over this framework
  - Clustering of the learned representations is the source of compression
  - Methods to track clustering in det. DNNs (incl. $H\big(\mathsf{Bin}(T_\ell)\big)$)

Summary

- **Reexamined Information Bottleneck Compression:**
  - $I(X;T)$ fluctuations in det. DNNs are theoretically impossible
  - Yes, past works presented $I(X;T)$ dynamics during training
- **Noisy DNN Framework:** Studying IT quantities over DNNs
  - Toolkit for accurate MI estimation over this framework
  - Clustering of the learned representations is the source of compression
  - Methods to track clustering in det. DNNs (incl. $H\big(\text{Bin}(T_\ell)\big)$)

⊛ **Det. DNNs cluster representations**

Summary

- **Reexamined Information Bottleneck Compression:**
  - $I(X;T)$ fluctuations in det. DNNs are theoretically impossible
  - Yes, past works presented $I(X;T)$ dynamics during training
- **Noisy DNN Framework:** Studying IT quantities over DNNs
  - Toolkit for accurate MI estimation over this framework
  - Clustering of the learned representations is the source of compression
  - Methods to track clustering in det. DNNs (incl. $H\big(\text{Bin}(T_\ell)\big)$)

⊛ **Det. DNNs cluster representations** $\implies$ Clarify past observations

Summary

- **Reexamined Information Bottleneck Compression:**
  - $I(X;T)$ fluctuations in det. DNNs are theoretically impossible
  - Yes, past works presented $I(X;T)$ dynamics during training
- **Noisy DNN Framework:** Studying IT quantities over DNNs
  - Toolkit for accurate MI estimation over this framework
  - Clustering of the learned representations is the source of compression
  - Methods to track clustering in det. DNNs (incl. $H\big(\text{Bin}(T_\ell)\big)$)

⊛ **Det. DNNs cluster representations** $\implies$ Clarify past observations

- **Future Research:**

Summary

- **Reexamined Information Bottleneck Compression:**
  - $I(X;T)$ fluctuations in det. DNNs are theoretically impossible
  - Yes, past works presented $I(X;T)$ dynamics during training
- **Noisy DNN Framework:** Studying IT quantities over DNNs
  - Toolkit for accurate MI estimation over this framework
  - Clustering of the learned representations is the source of compression
  - Methods to track clustering in det. DNNs (incl. $H\big(\mathrm{Bin}(T_\ell)\big)$)

⊛ **Det. DNNs cluster representations** $\Longrightarrow$ Clarify past observations

- **Future Research:**
  - Curse of dimensionality: How to track clustering in high-dimensions?

Summary

- **Reexamined Information Bottleneck Compression:**
  - $I(X;T)$ fluctuations in det. DNNs are theoretically impossible
  - Yes, past works presented $I(X;T)$ dynamics during training
- **Noisy DNN Framework:** Studying IT quantities over DNNs
  - Toolkit for accurate MI estimation over this framework
  - Clustering of the learned representations is the source of compression
  - Methods to track clustering in det. DNNs (incl. $H\big(\mathsf{Bin}(T_\ell)\big)$)

⊛ **Det. DNNs cluster representations** $\Longrightarrow$ Clarify past observations

- **Future Research:**
  - Curse of dimensionality: How to track clustering in high-dimensions?
  - Is compression necessary? Desirable?

Summary

- **Reexamined Information Bottleneck Compression:**
  - $I(X;T)$ fluctuations in det. DNNs are theoretically impossible
  - Yes, past works presented $I(X;T)$ dynamics during training
- **Noisy DNN Framework:** Studying IT quantities over DNNs
  - Toolkit for accurate MI estimation over this framework
  - Clustering of the learned representations is the source of compression
  - Methods to track clustering in det. DNNs (incl. $H\big(\text{Bin}(T_\ell)\big)$)

⊛ **Det. DNNs cluster representations** $\implies$ Clarify past observations

- **Future Research:**
  - Curse of dimensionality: How to track clustering in high-dimensions?
  - Is compression necessary? Desirable?
  - Build on findings to improve DNN training alg. and architectures