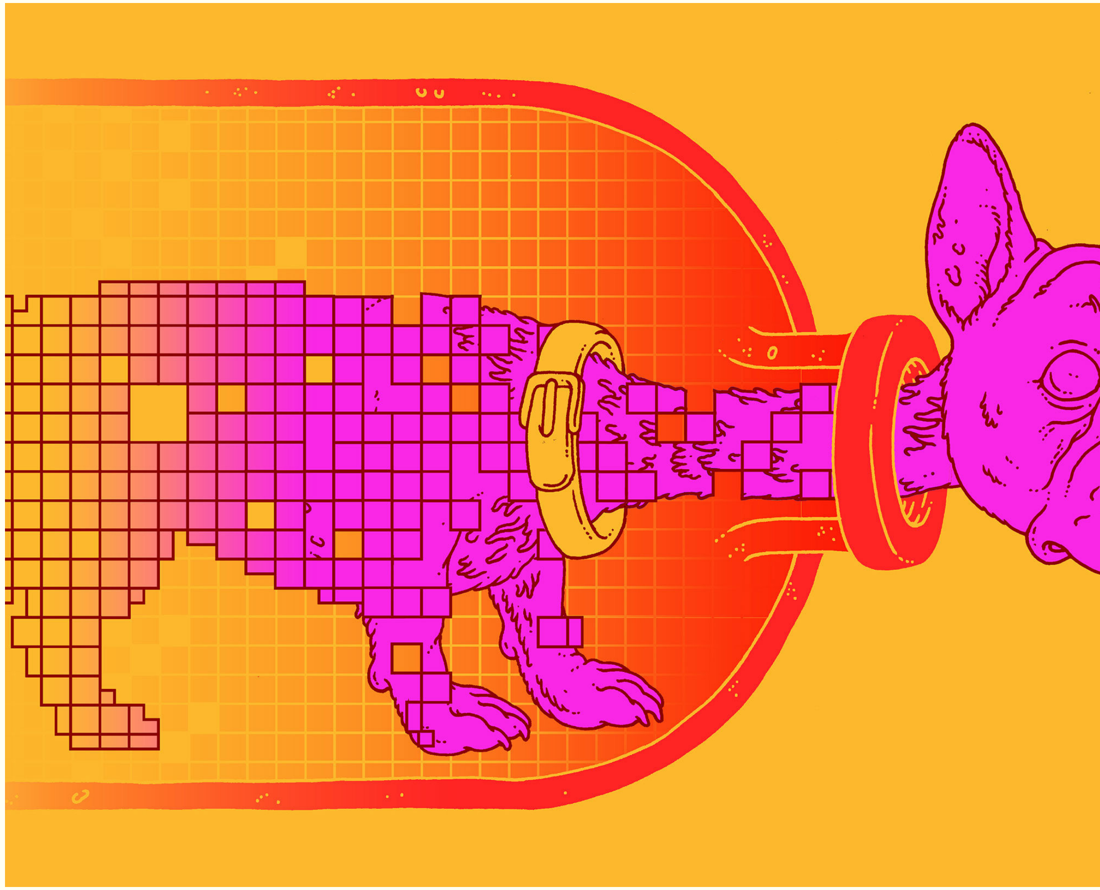
OPENING THE BLACK BOX OF DEEP NEURAL NETWORKS



Ravid Swhartz-Ziv

Advisor: Naftali Tishby

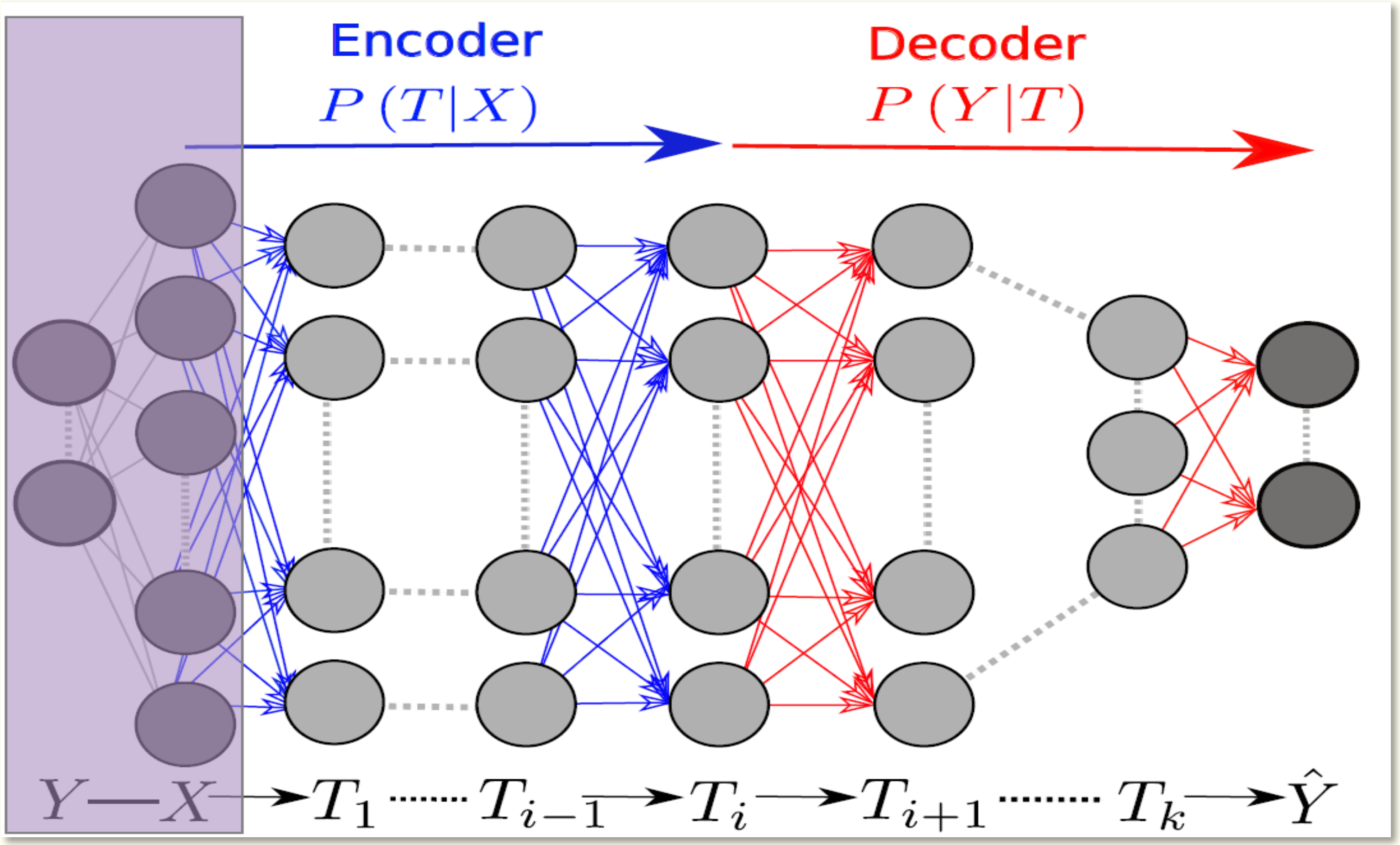
The Edmond & Lily Safra Center for Brain Sciences

Hebrew University, Jerusalem, Israel

We Need a Theory for Deep Learning…

- Why DNN’s work so well?
- What is “an optimal DNN”?
- Design principles
  - What determines the number & width of the layers?
  - What determines the connectivity and inter-layer connections?
- Interpretability
  - What do the layers/neurons capture/represent?
- Better learning algorithms
  - Is stochastic gradient descent the best we can do?

Deep Neural Networks and Information Theory



Information Theory Basics

- **The KL-distribution divergence:**

$$D_{KL}[p(x)||q(x)] = \sum_{x \in X} p(x) log \frac{p(x)}{q(x)} \geq 0$$

- **The Mutual Information:**

$$I(X;Y) = D_{KL}[p(x,y)||p(x)p(y)] = H(X) - H(X|Y)$$

- **Data Processing Inequality (DPI):**

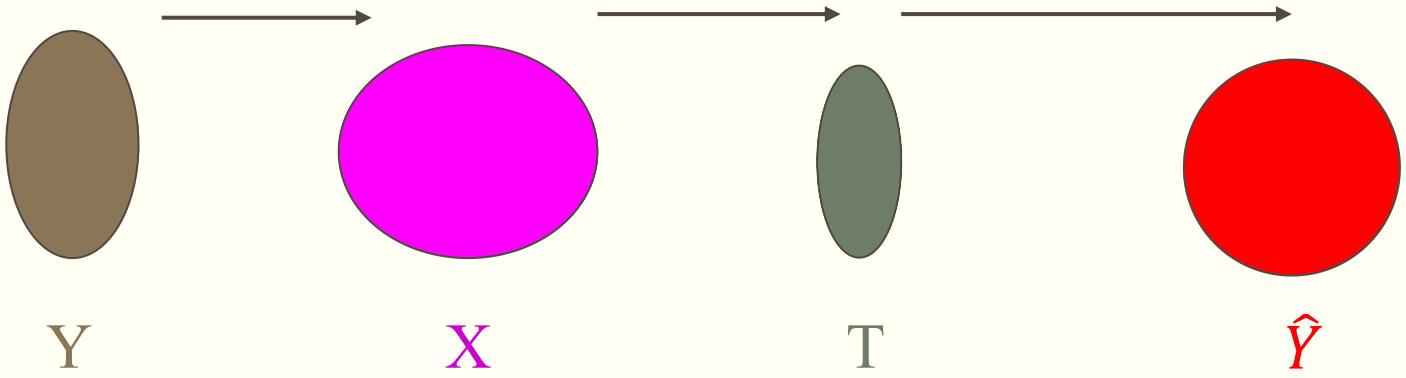
For any Markov chain $X \to Y \to Z$

$$I(X;Y) \geq I(X;Z)$$

The Information Bottleneck Method

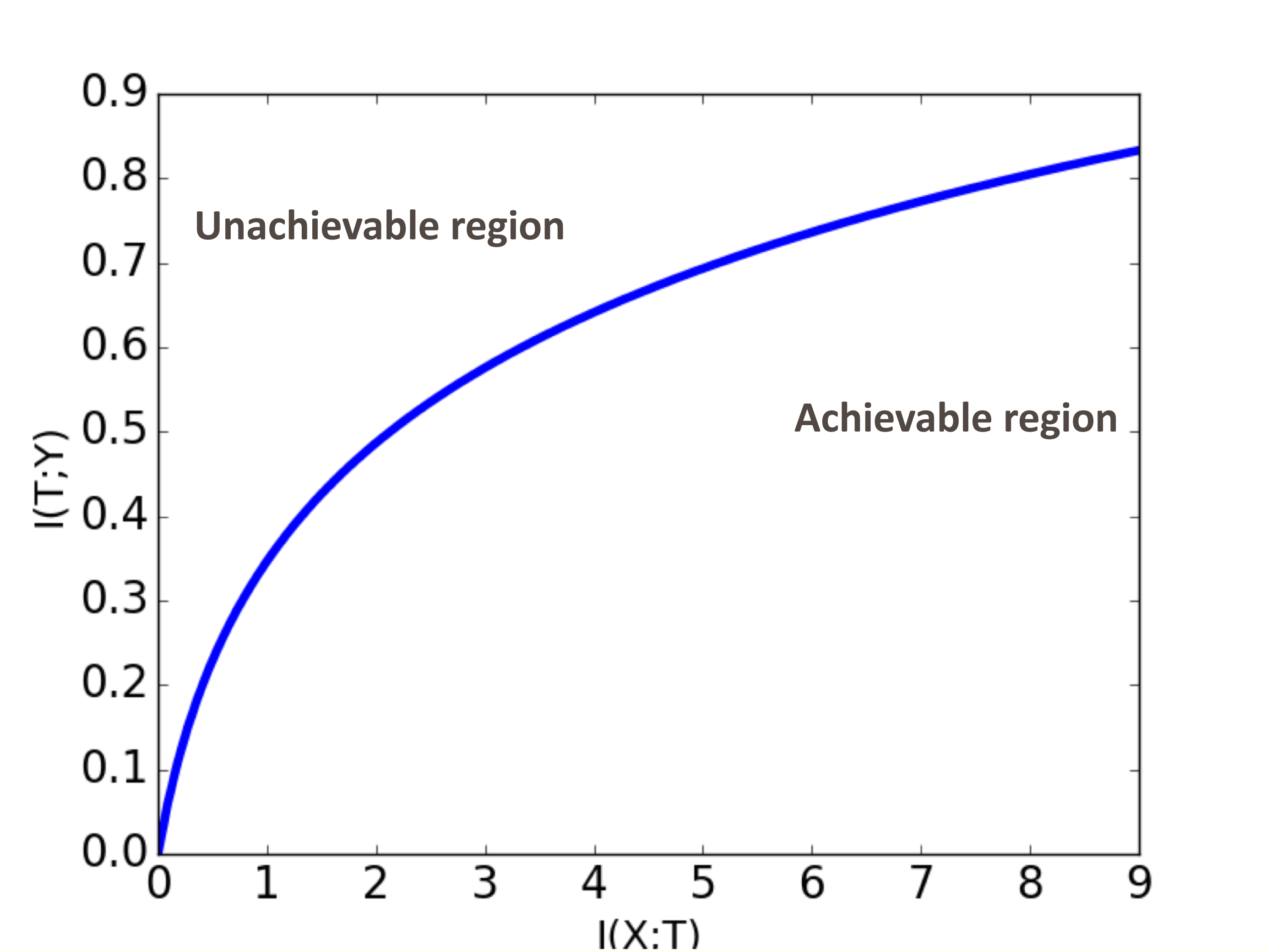
(Tishby, Pereira, Bialek, 1999)

- We would like to find relevant partitioning T that compress X as much as possible, and to capture as much as possible information about Y



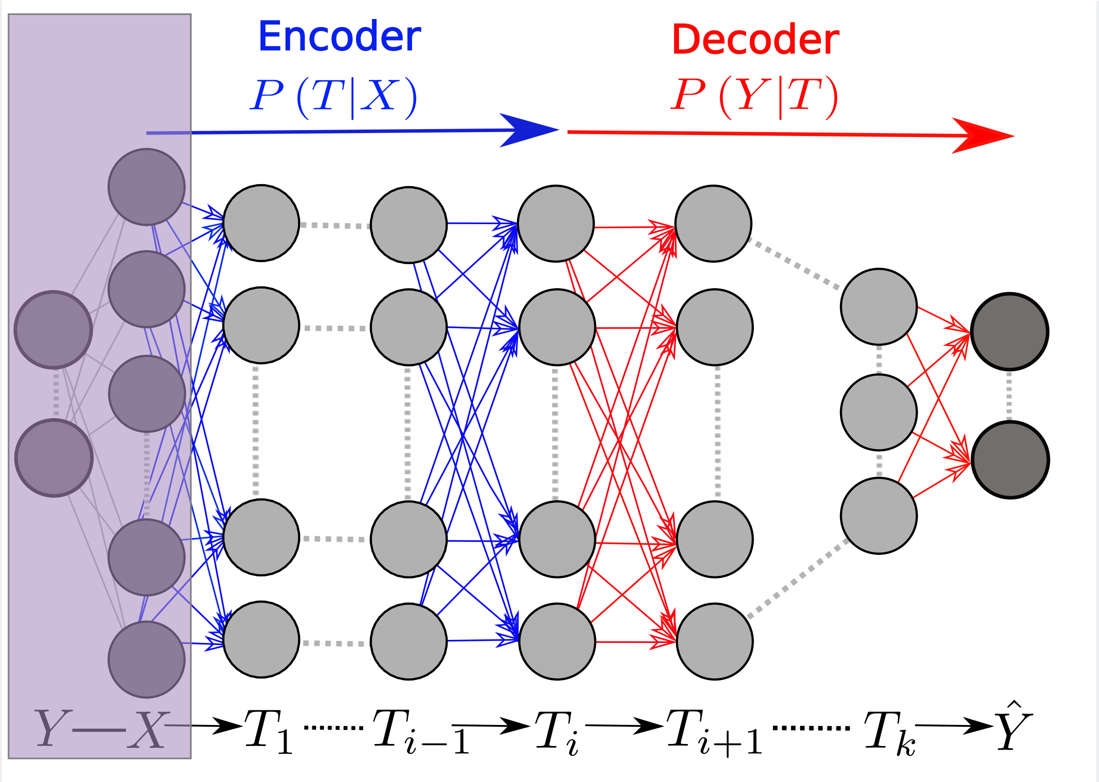
$$\min_{p(t|x)} I(T;X) - \beta I(T;Y), \beta > 0$$

The Information Plane



DNN's and the Information Bottleneck

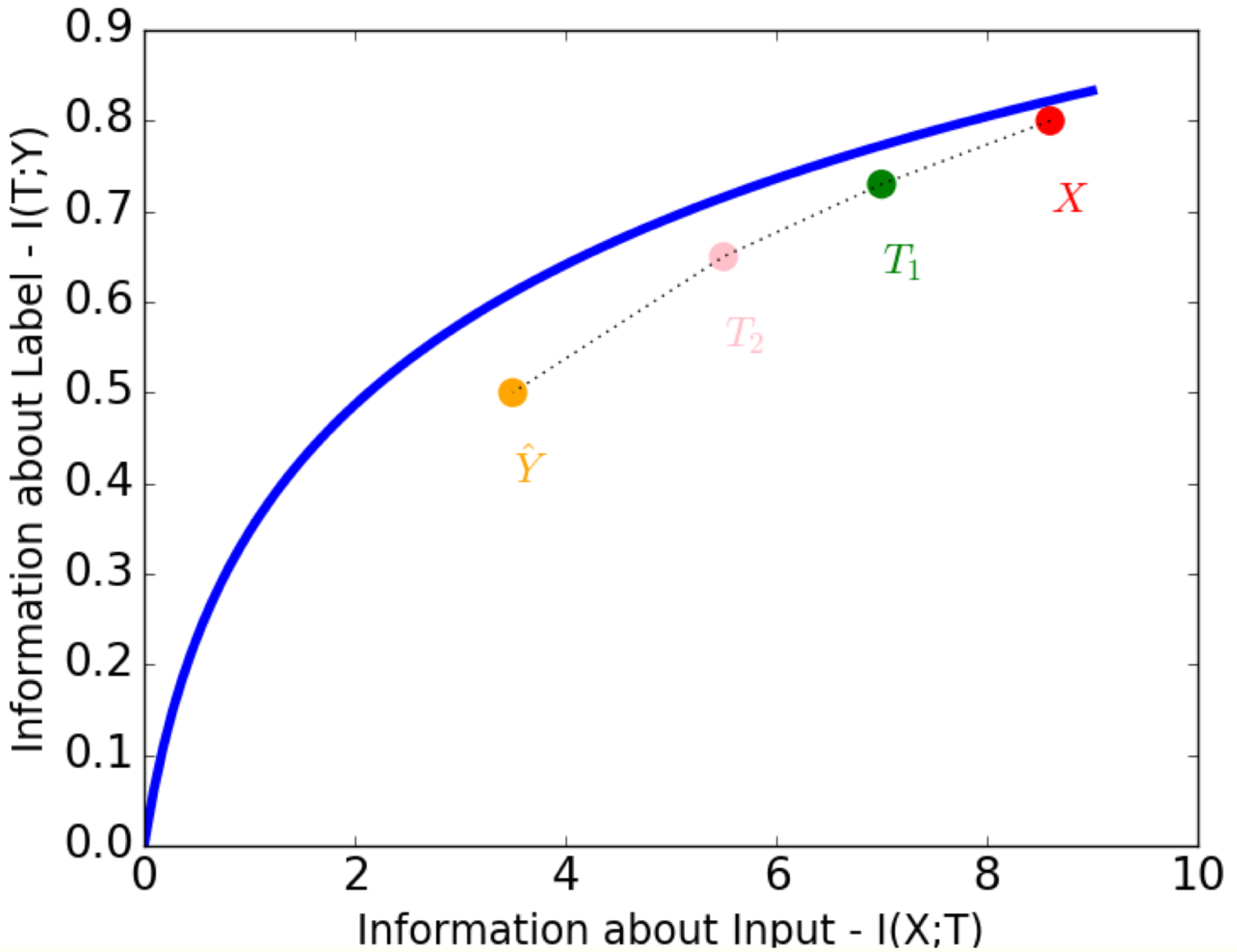
- Markov chain of intermediate representations.
- $I(X;Y) \geq I(h_j;Y) \geq I(h_{j+1};Y) \ldots \geq I(\hat{Y};Y)$
- $H(X) \geq I(X;h_j) \geq I(X;h_{j+1}) \geq I(X;h_{j+1}) \ldots$



DNN's and the Information Bottleneck

- For each layer there is an optimal point on the information plane.

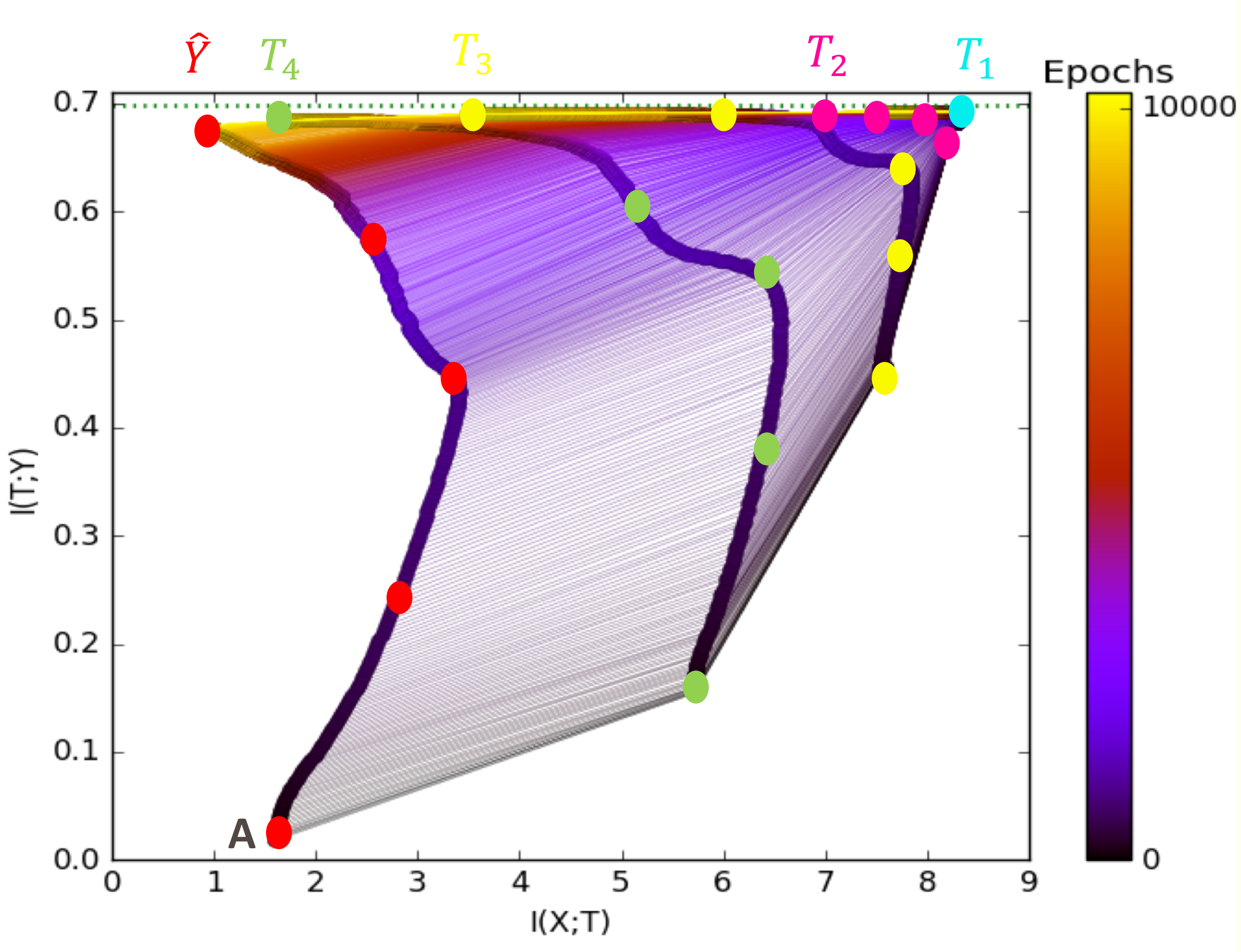
- The goal of the network is to find the best trade-off between compression and prediction for each layer.



THE INFORMATION IN DNNS DURING THE TRAINING

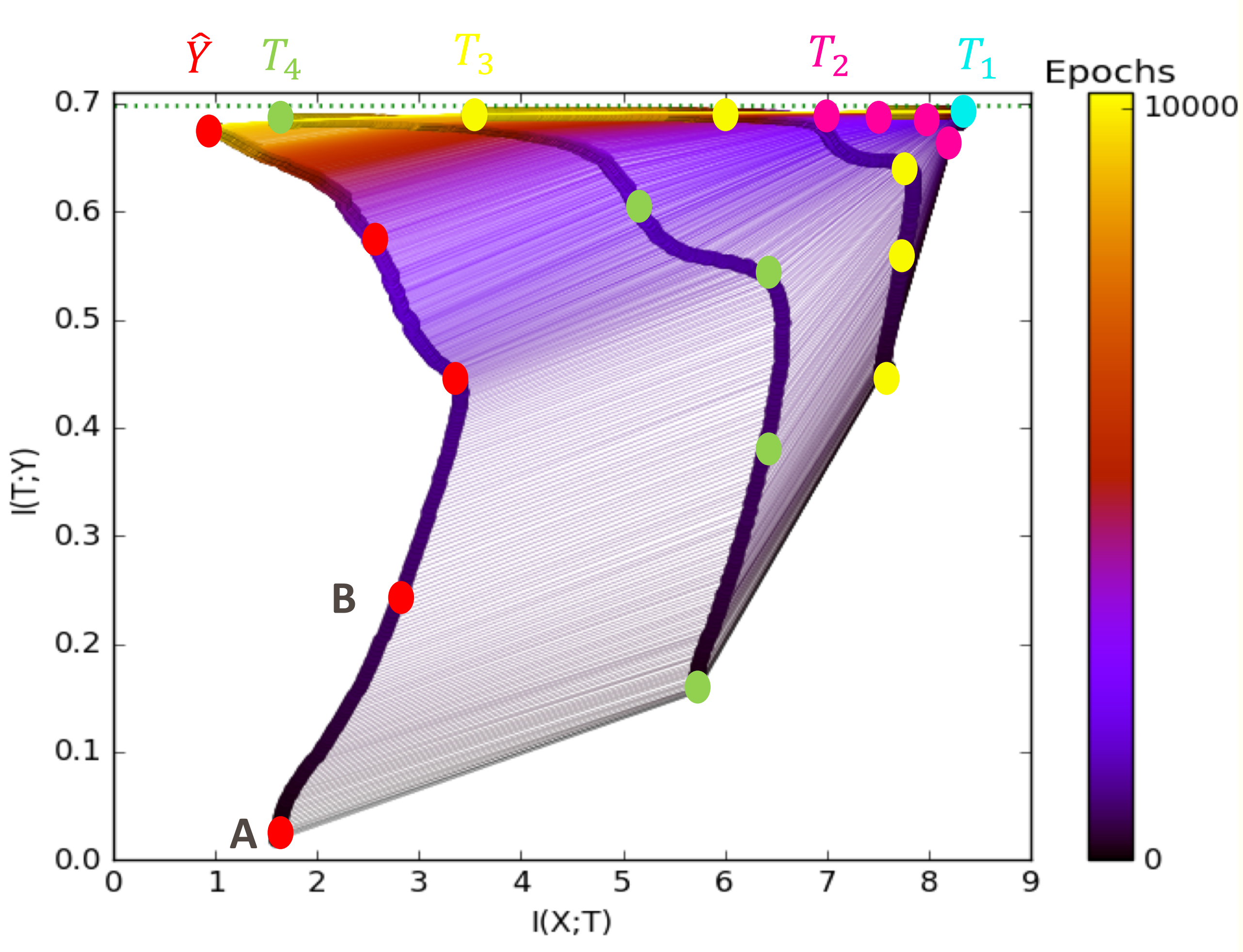
The Information during the Learning

- A – Initial State:
  - The neurons in layer 1 encode everything about the input.
  - The neurons in the highest layers are in a nearly random state.



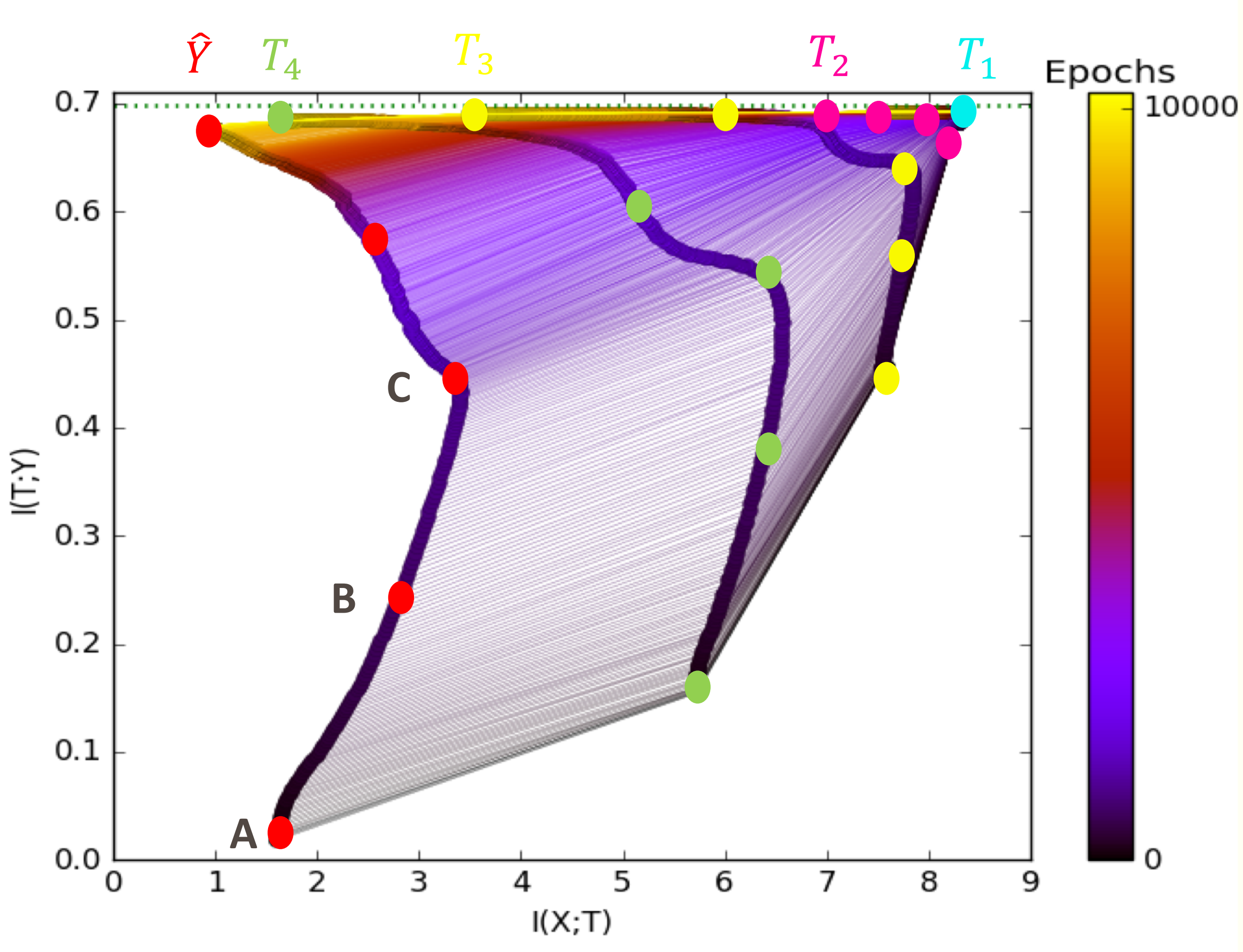
The Information during the Learning

- B - Fitting Phase:
  - The higher layers gain information about the input.
  - They are fitting to the labels.



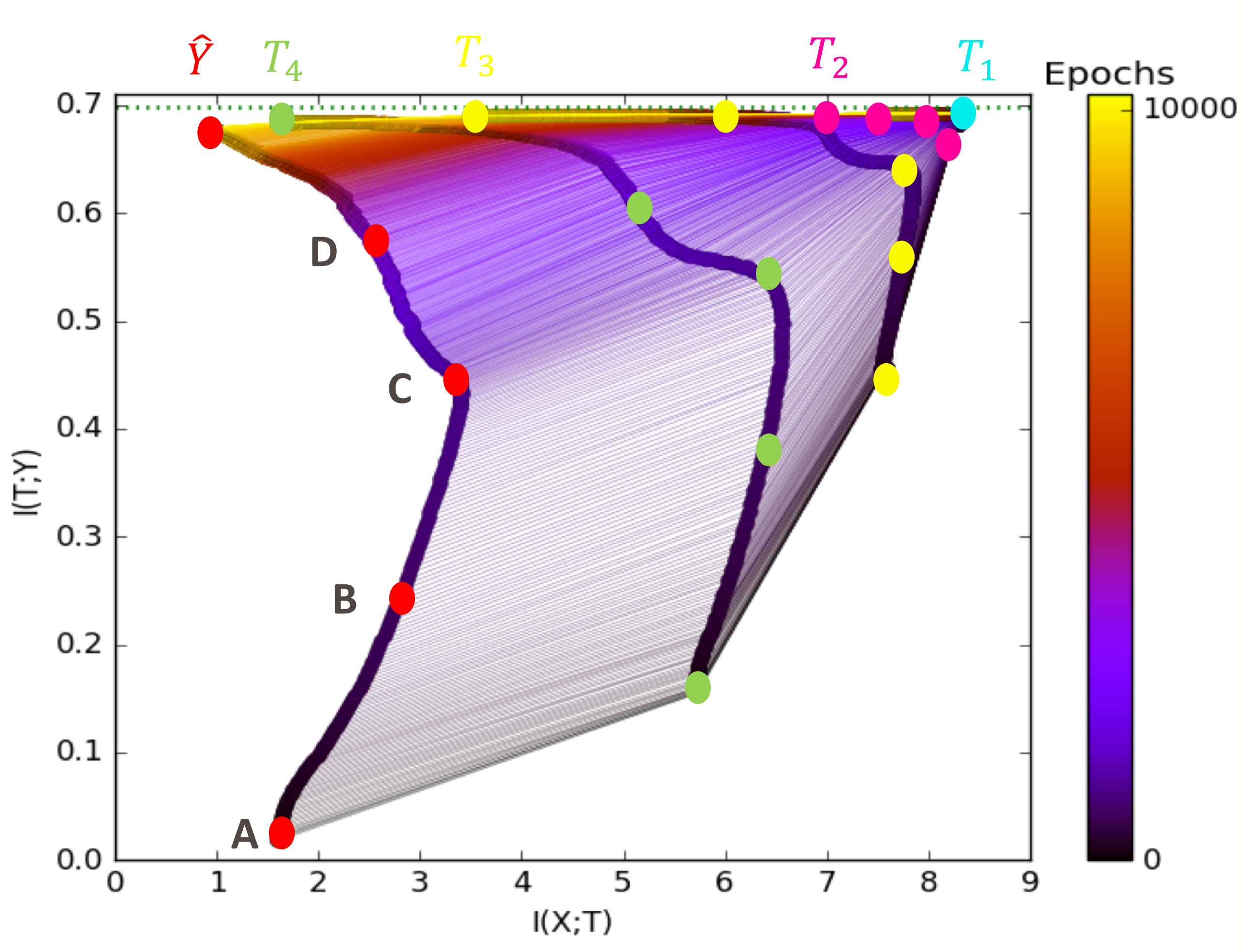
The Information during the Learning

- C – Phase change:
  - The layers stop fitting.
  - They start to forget information about the input.



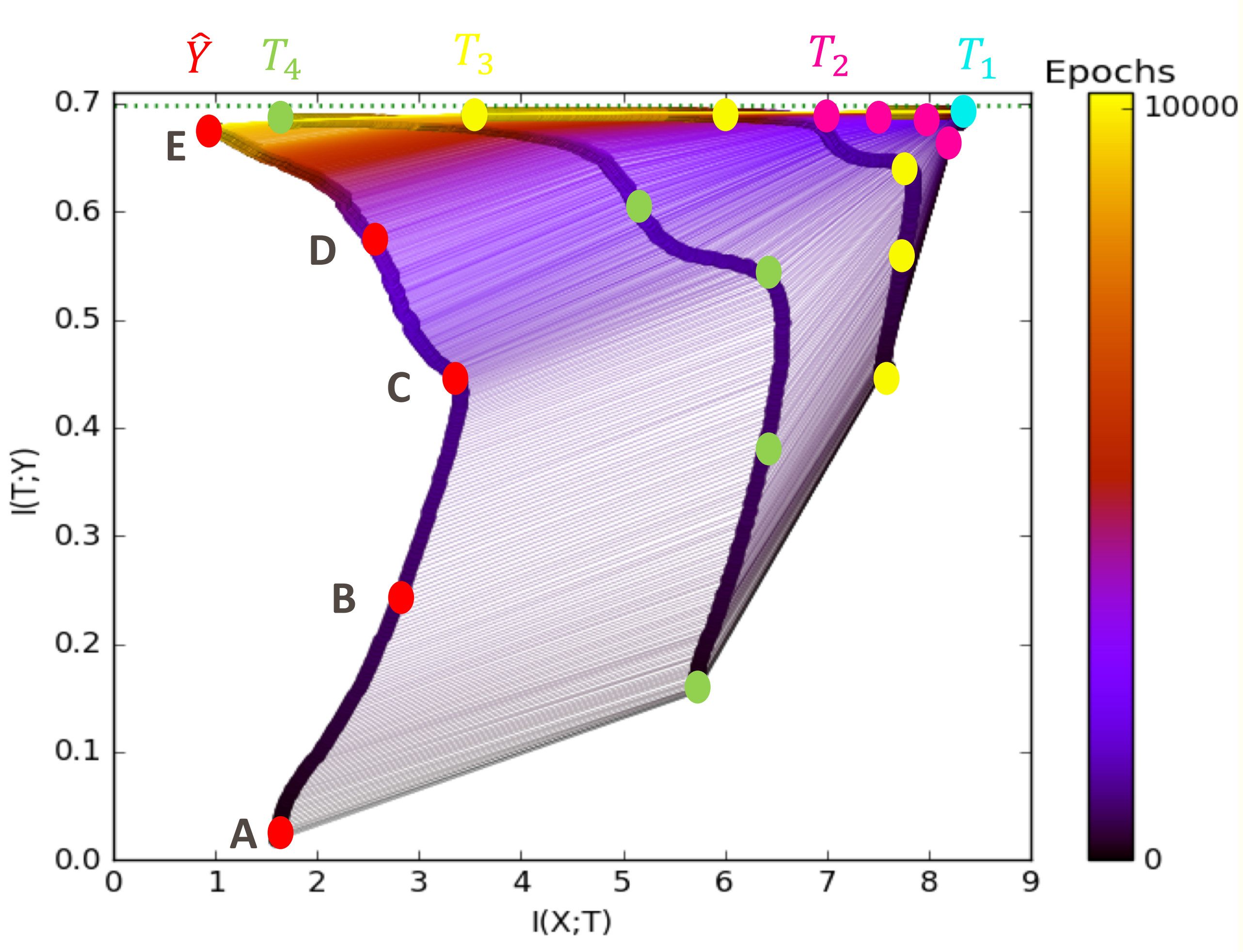
The Information during the Learning

- D – Compression phase:
  - The layers compress their representation.
  - They keep only the relevant information about the labels.

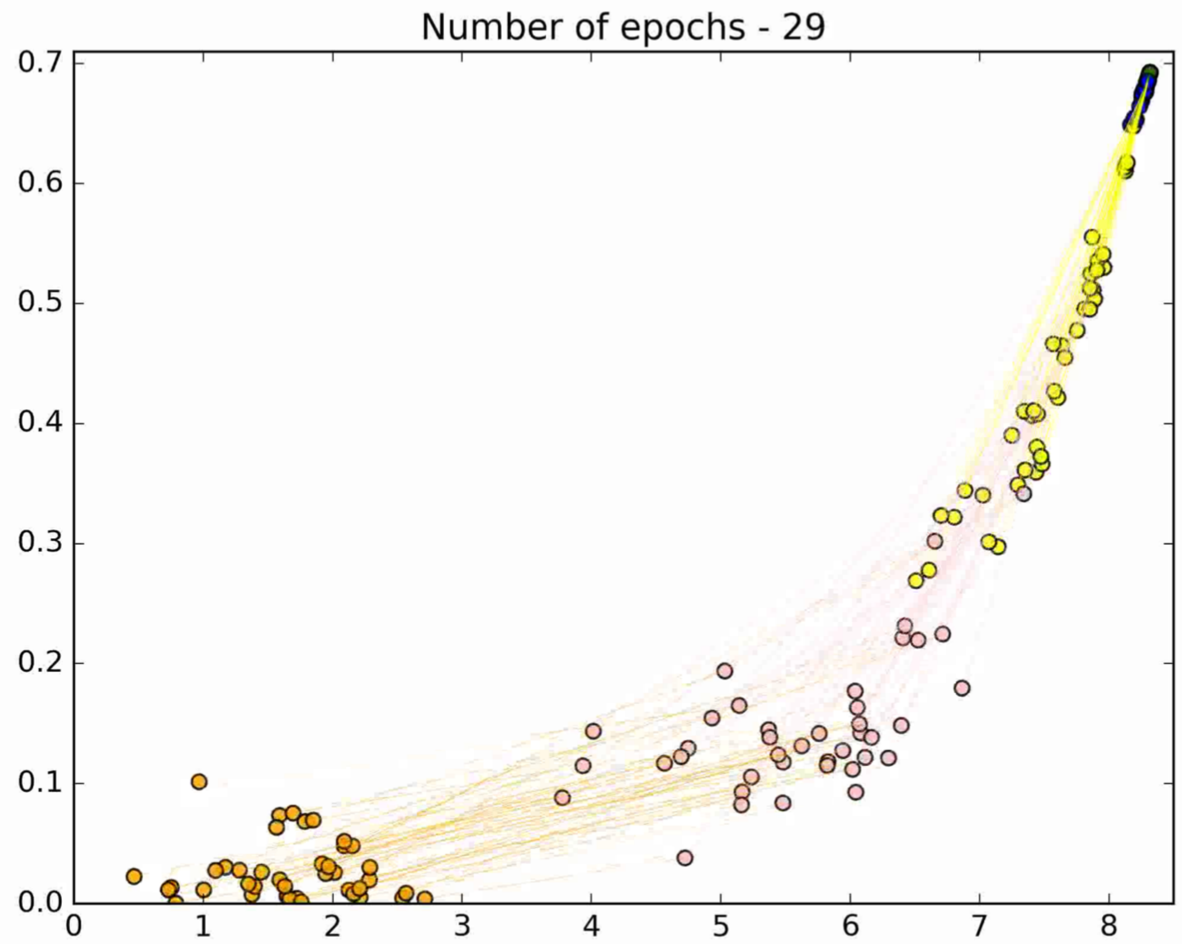


The Information during the Learning

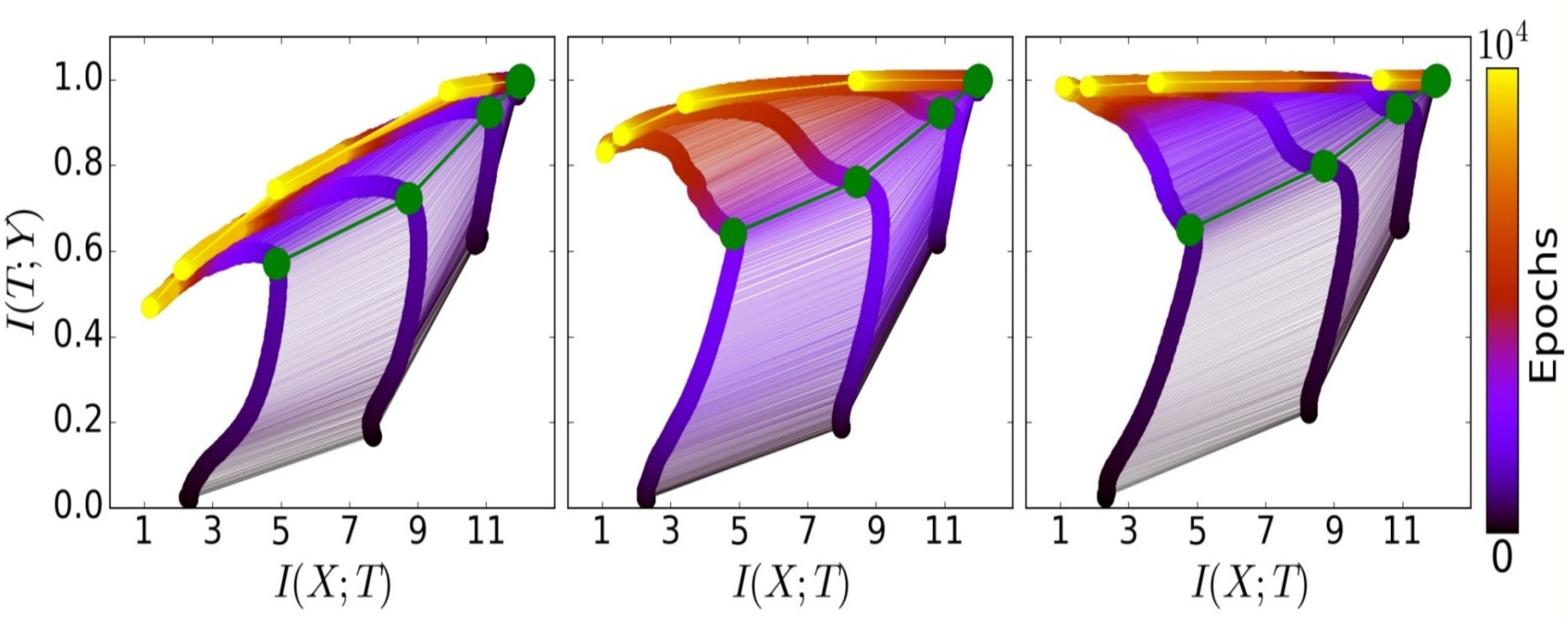
- E – Final State
  - The layers converge to an optimal balance between accuracy and predication.



DNNs in the Information Plane



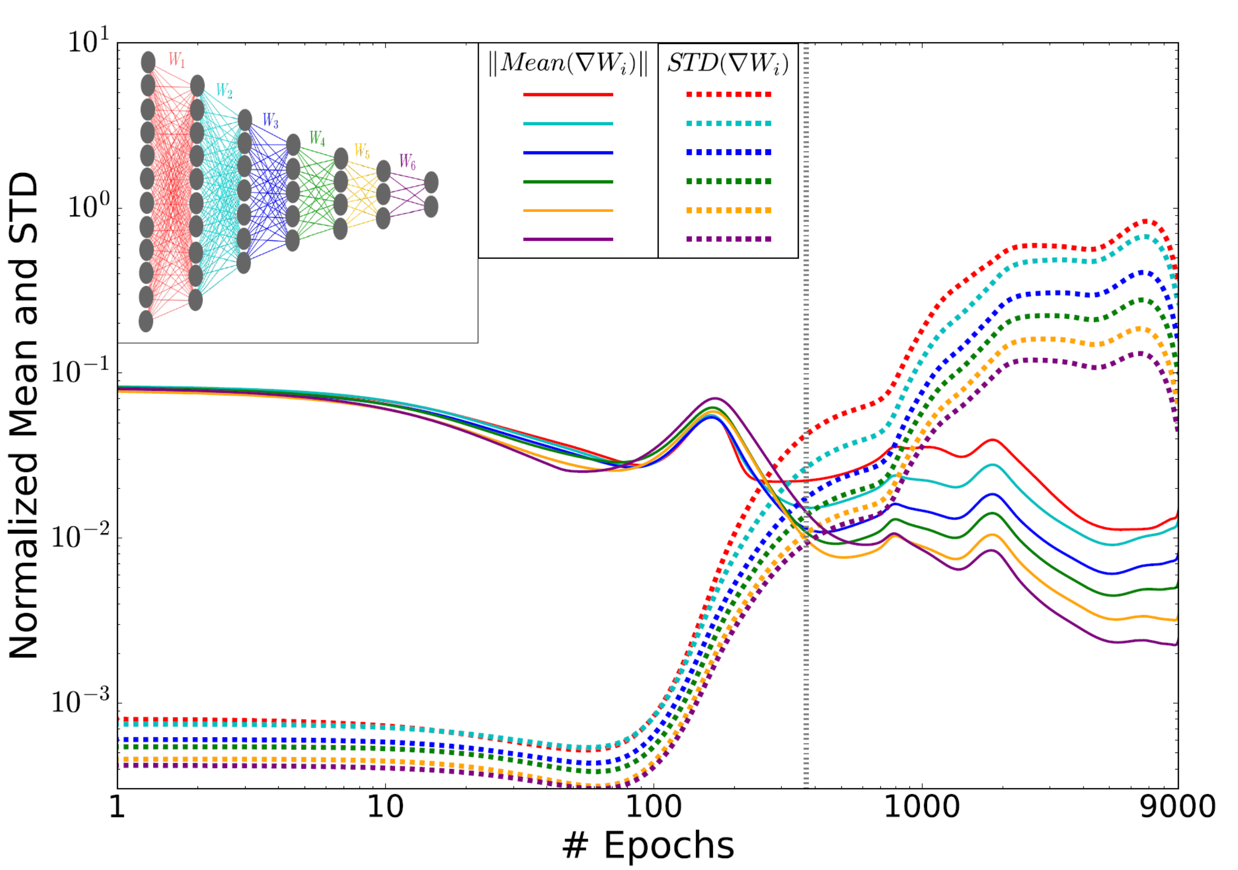
Different Amount of Training Data



5% **45%** **80%**

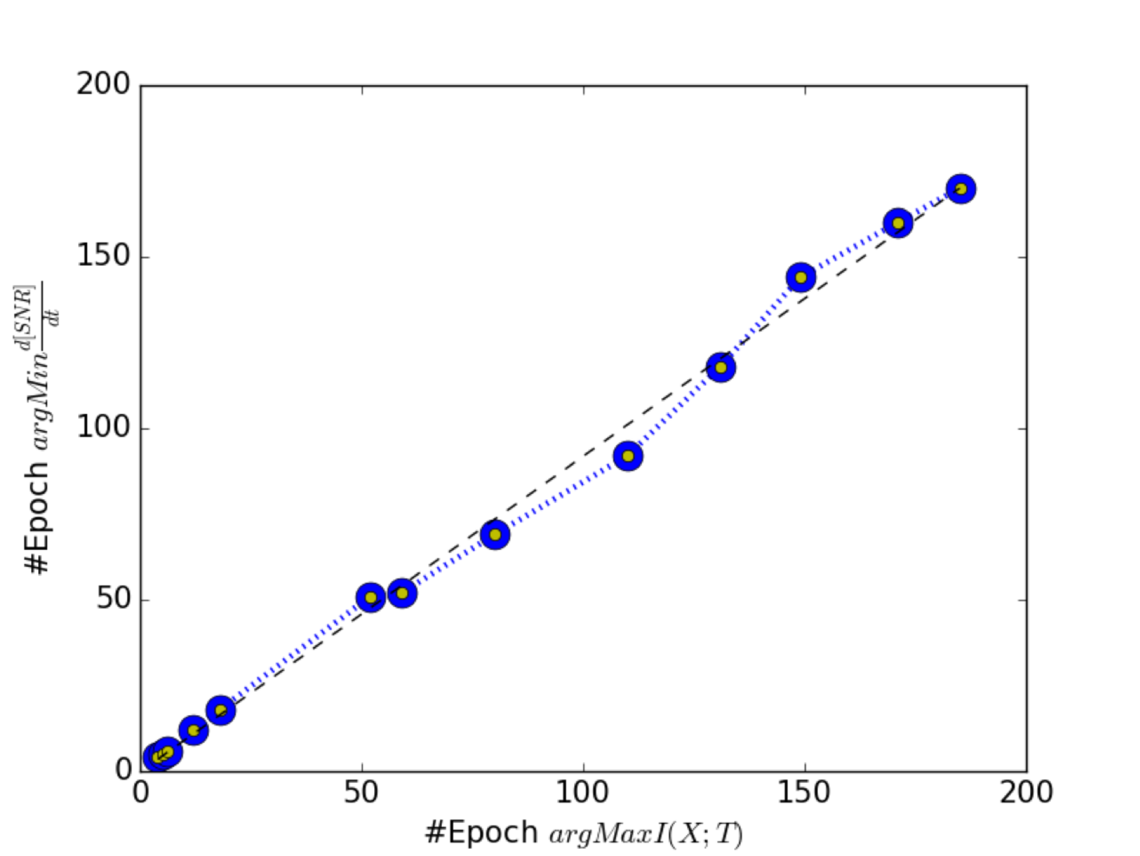
The Layers' Gradients

- **First phase** - large gradient means and small variance (*drift*, high SNR).
- **Second phase** - large fluctuations and small means (*diffusion*, low SNR).
- The phase transition in the information accrues at the minimum of the derivative of the SNR of the gradients.

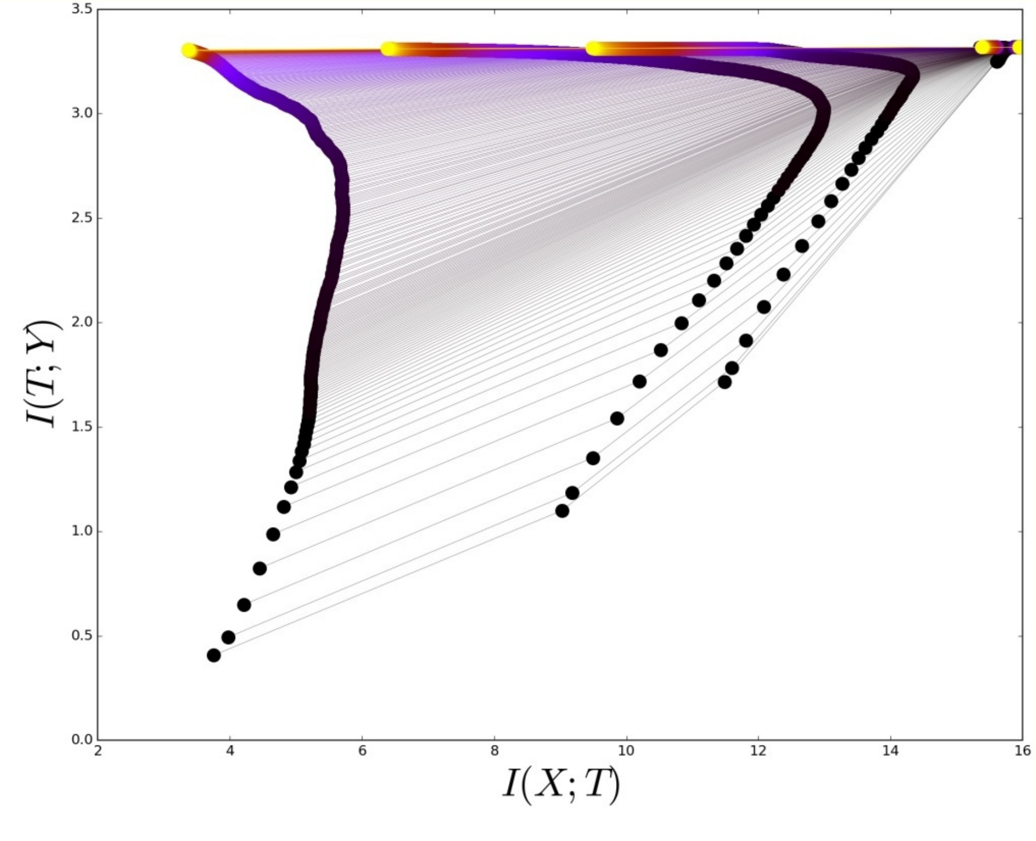


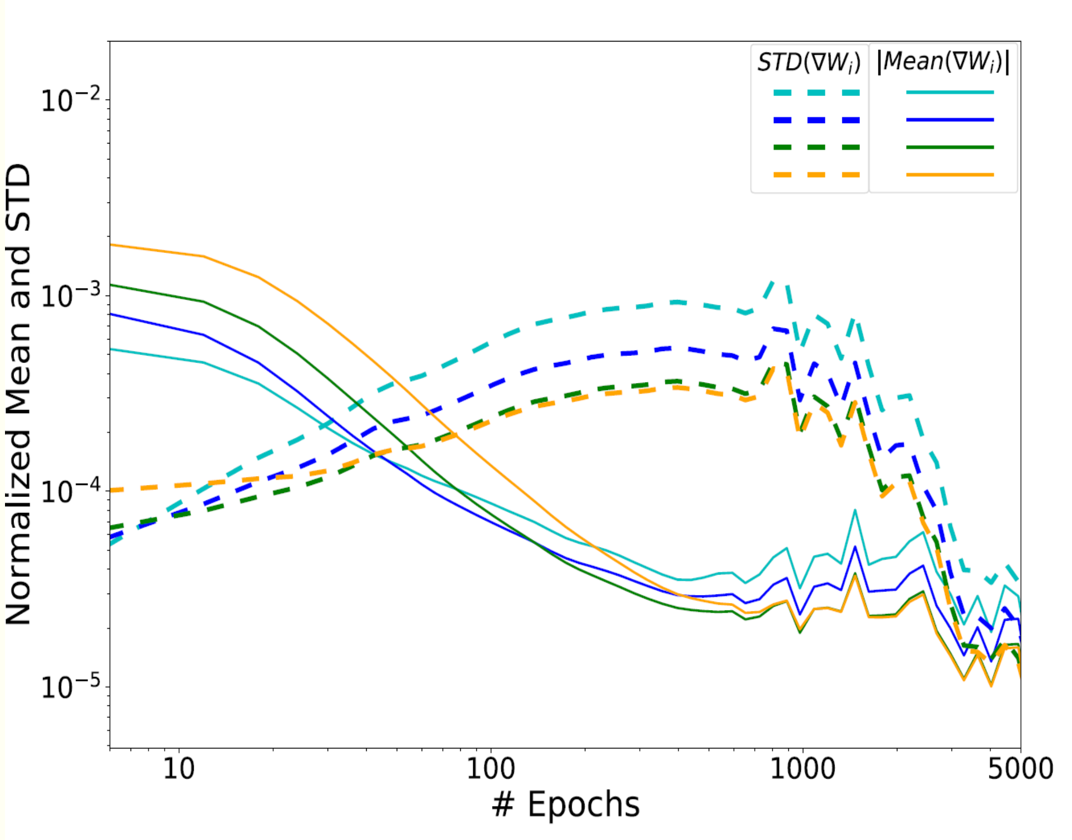
The Phase Transition

- The phase transition in the information plane occurs at the same time as the transition in the gradients.

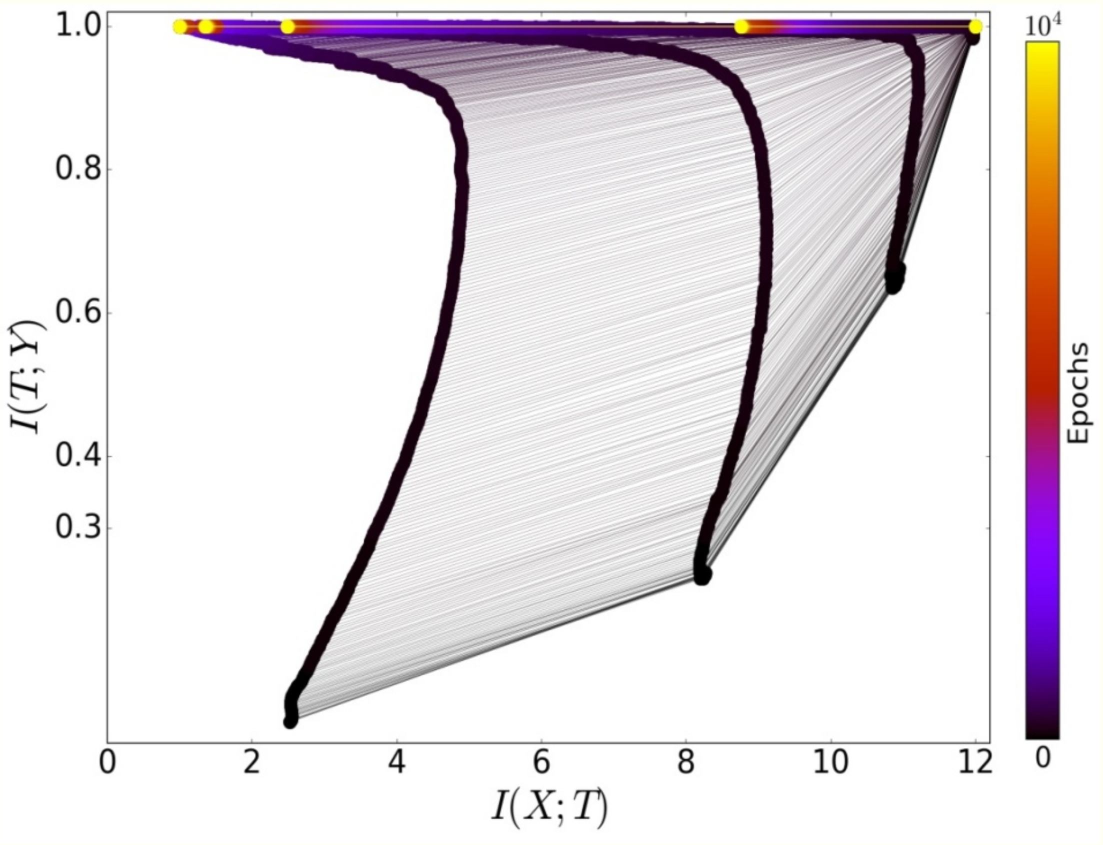


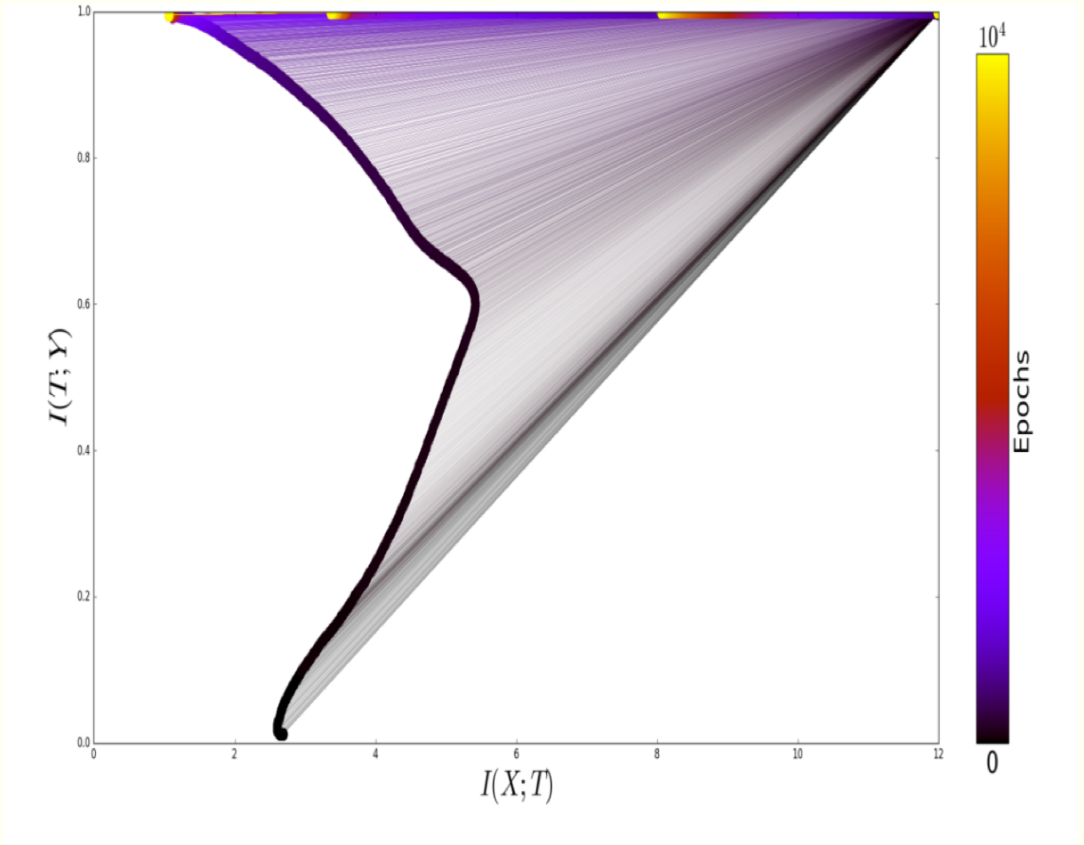
MNIST with CNN Architecture





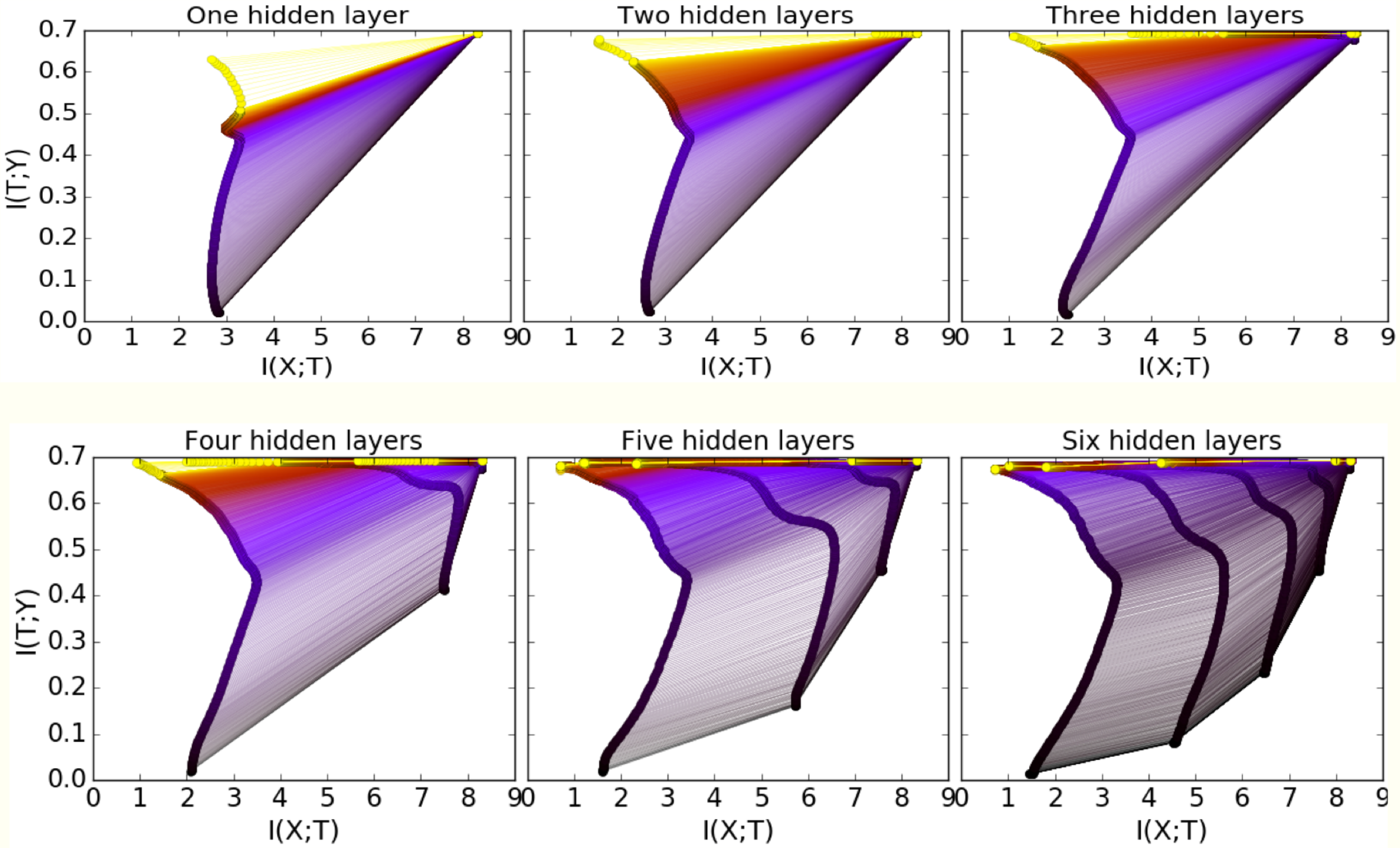
Different Problems and Architectures





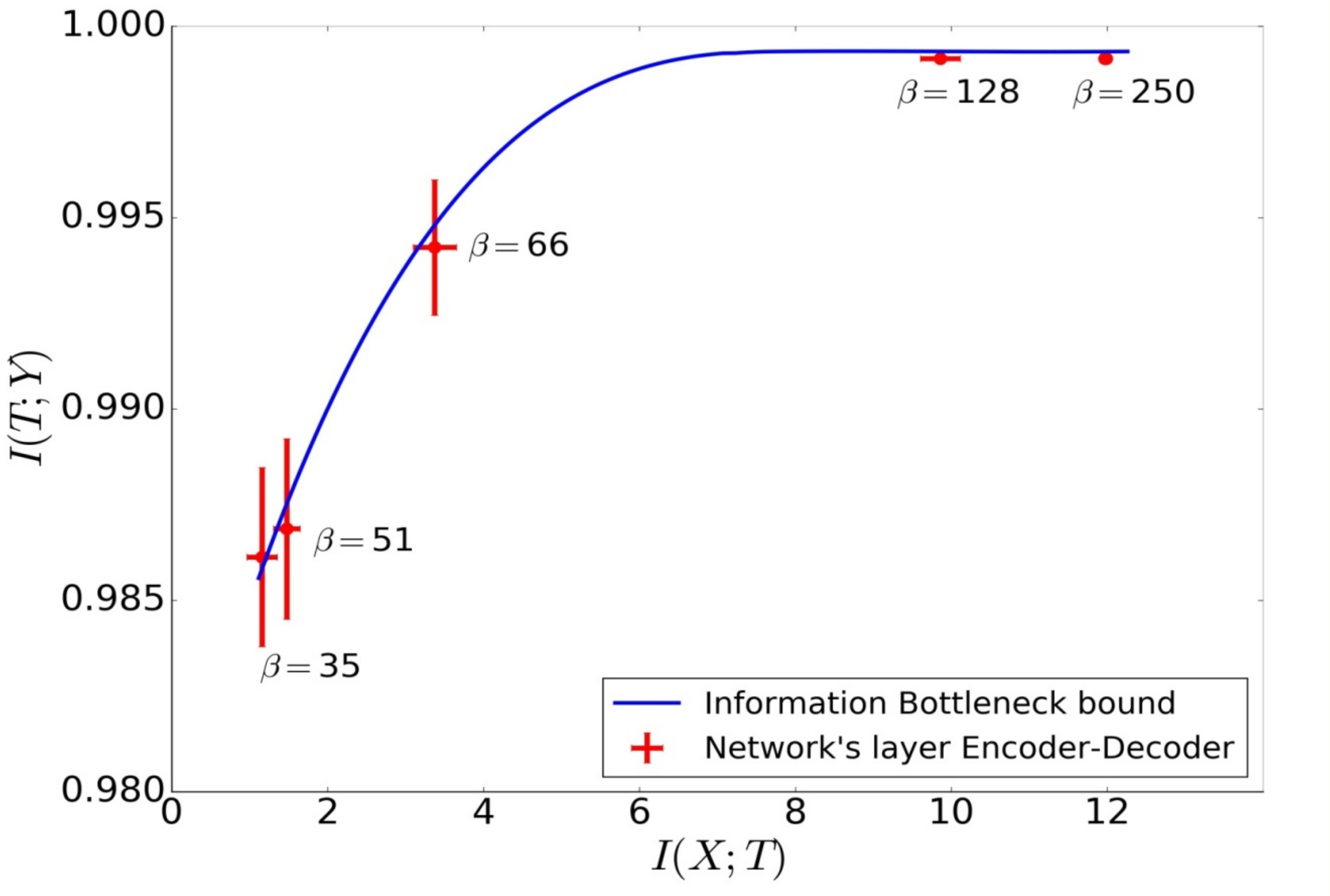
The Benefit of the Hidden Layers

- More layers take much fewer training epochs.
- The optimization time depend super-linearly on the compressed information for each layer.



The Optimality of the Network

- Layers of optimal DNN converge to the optimal IB limit information curve.
- The DNN encoder & decoder for each layer satisfy the IB self-consistent equations.



STOCHASTIC RELAXATION AND REPRESENTATION COMPRESSION.

The SGD Algorithm Converged to the IB Bound

$$dw_k\ (t) = -\nabla L(w_k)dt + \sqrt{\beta_k^{-1} D_k(x)}dW_k(t)$$

- $D_k(x)$ is the variance of the error function
- $W_k(t)$ is a Brownian motion
- β_k is the noise level of the layer

The SGD Algorithm Converged to the IB Bound

- SGD Converged to Gibbs distribution for each layer

  - $p(W_k) \approx \exp(\beta_k L(W_k))$

- It's the global minimizer of the free energy functional

$$F(p) = \mathbb{E}[L(W_k)] - \beta^{-1} H(p)$$

- Maximize the entropy under the constrain of the potential function.

The SGD Algorithm Converged to the IB Bound

- Max entropy over the weights $\rightarrow max\ H(X|T_k)$
- Since $I(X;T_k) = H(X) - H(X|T_k)$
- The SGD brings $I(X;T_k)$ to minimum under the constrain of the error
- When the loss function is the D_{KL},

$$F(p) = -I(Y;T_k) + \beta_k^{-1} I(X;T_k)$$

-> SGD converges to the optimal IB bound

Summary

- How the DNN layers converge?
  - The ERM and Representation Compression phases
  - The Drift and Diffusion phases of the SGD optimization
- What is the benefit of the Hidden Layers?
  - Computational benefit – boosting the compression
- Interpretability
  - Generally, only full layers can be interpreted
- Stochastic relaxation and representation compression
  - The SGD algorithm converges to the optimal IB bound

Questions?