Opening the black box of Deep Neural Networks via Information

Ravid Schwartz-Ziv, **Naftali Tishby**
The Hebrew University of Jerusalem, Israel

Presenter: Shilin HE

2019/01/03

Geoffrey Hinton's Opinion on Information Bottleneck

"It's extremely interesting," Hinton wrote. "I have to listen to it another 10,000 times to really understand it, but it's very rare nowadays to hear a talk with a really original idea in it that may be the answer to a really major puzzle."

Motivation

Neural Network

To optimize the Information Bottleneck (IB) tradeoff between compression and prediction, successively, for each layer. [Tishby and Zaslavsky' 15]

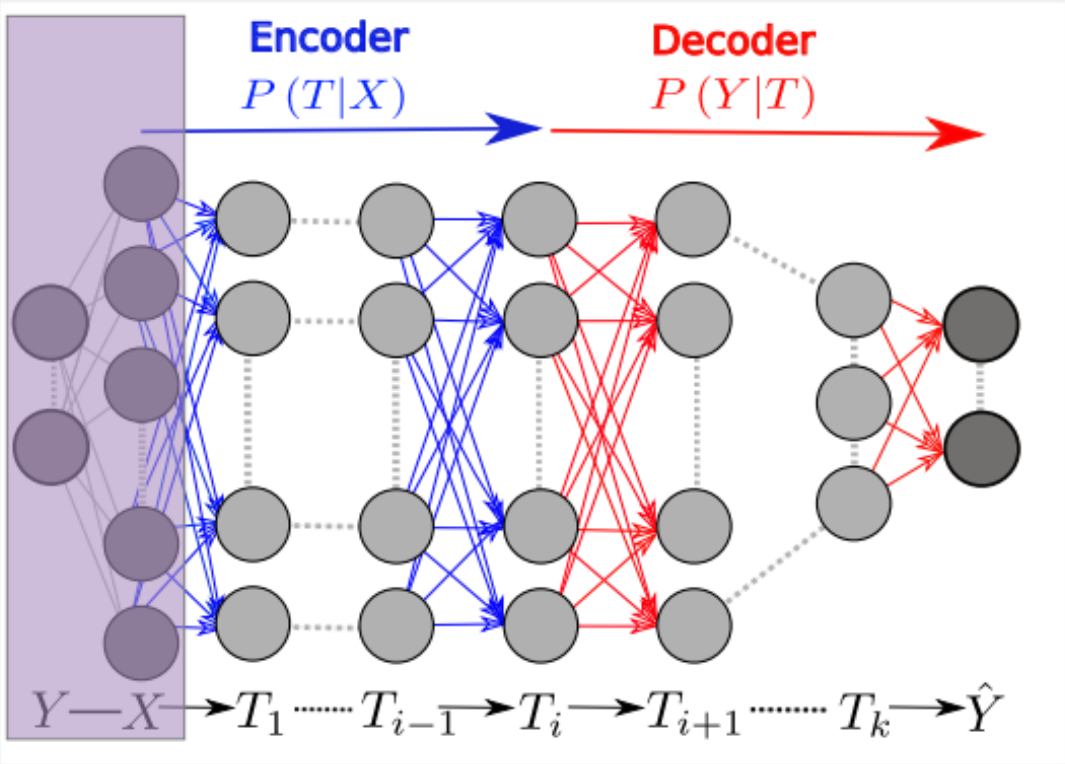
Goal of this paper:

demonstrate the effectiveness of the Information Plane visualization of DNNs

Contribution:

- Explore the neural networks from Information Bottleneck perspective
- Obtain several interesting findings/conclusions for a better understanding of training dynamics, learning processes and internal representations.

Information Theory of Deep Learning



X: input patters; **T(X):** representation; **Y:** label

Markov Chain of such representation: hidden layers

Treat T as a random variable

Background

Mutual Information: MI determines how similar the joint distribution p(x,y) is to the products of factored marginal distribution p(x)p(y)

$$I(X;Y) = D_{KL}[p(x,y)||p(x)p(y)] = \sum_{x\in X, y\in Y} p(x,y)\log\left(\frac{p(x,y)}{p(x)\,p(y)}\right)$$

$$= \sum_{x\in X, y\in Y} p(x,y)\log\left(\frac{p(x|y)}{p(x)}\right) = H(X) - H(X|Y)\,,$$

MI quantifies the **number of relevant bits** that the **input variable X** contains about the **label Y** on average

Background

Optimal Learning Problem then becomes:

Construction of an optimal encoder of that relevant information via an efficient representation ---- a *minimal sufficient statistic* of X with respect to Y

A *minimal sufficient statistic* can enable the decoding of the relevant information with the smallest number of binary questions (on average); i.e., an optimal code.

Two Properties of MI:

1) **Invariance to Invertible transformations**

2) **Data Processing Inequality (DPI)**

Background

Property 1: Invariance to Invertible transformations

$$I(X;Y) = I(\psi(X);\phi(Y)))$$

Toy example:

$$I(X;Y) = \sum_{y\in Y}\sum_{x\in X}\left(p(x,y)log(\frac{p(x,y)}{p(x)p(y)})\right)$$

$$Z_x = \alpha X \qquad Z_y = \alpha Y \qquad I(Z_x;Z_y) \text{ =? } I(X;Y)$$

Obviously $p(z_x) = p(x)$ $p(z_y) = p(y)$ $p(z_x, z_y) = p(x,y)$

Property 1: Invariance to Invertible transformations

$$I(X;Y) = I(\psi(X);\phi(Y)))$$

Proof 2:

If X′ = F(X) and Y ′ = G(Y) are homeomorphisms (smooth and uniquely invertible maps), and JX = ||∂X/∂X′ || and JY = ||∂Y/∂Y ′ || are the Jacobi determinants, then

$$\mu'(x',y') = J_X(x')J_Y(y')\mu(x,y)$$

$$\begin{aligned} I(X',Y') &= \iint dx'dy'\mu'(x',y')\log\frac{\mu'(x',y')}{\mu'_x(x')\mu'_y(y')} \\ &= \iint dxdy\ \mu(x,y)\ \log\frac{\mu(x,y)}{\mu_x(x)\mu_y(y)} \\ &= I(X,Y)\ . \end{aligned}$$

[1] Estimating Mutual Informationm, Alexander Kraskov, Harald Stögbauer, and Peter Grassberger Phys. Rev. E **69**, 066138 – Published 23 June 2004; Erratum Phys. Rev. E **83**, 019903 (2011)

Property 2: Data Processing Inequality (DPI)

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

Intuitive understanding:

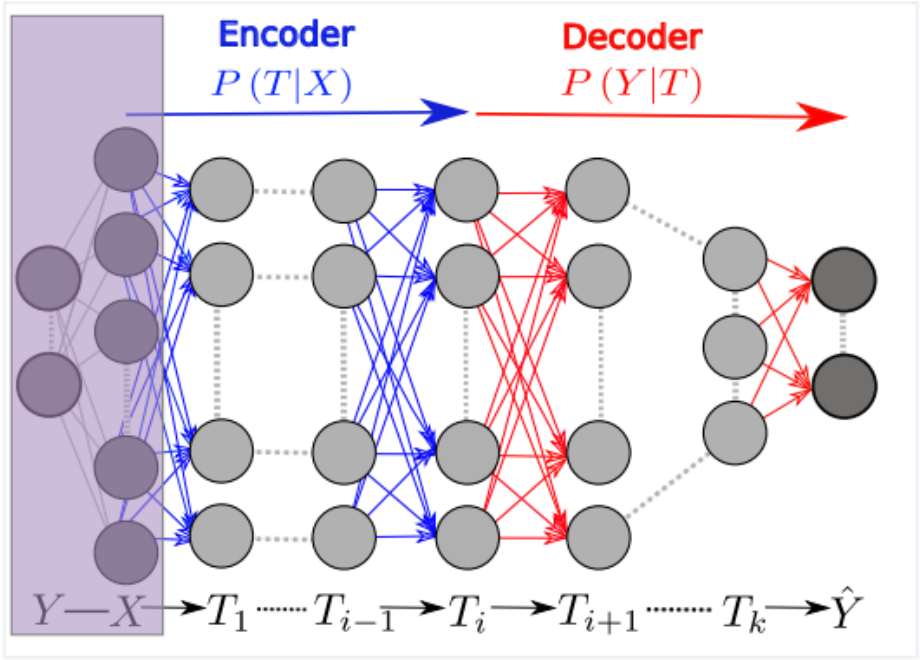
Information Loss

I(X,Y) = I(X,Z) if and only if X->Z->Y

Background

Information Plane:

Given P(X;Y), T is uniquely mapped to a point in the information plane with coordinate (I(X;T) , I(T;Y))



Information Path:

$$I(X;Y) \geq I(T_1;Y) \geq I(T_2;Y) \geq \ldots \geq I(T_k;Y) \geq I(\hat{Y};Y)$$
$$H(X) \geq I(X;T_1) \geq I(X;T_2) \geq \ldots \geq I(X;T_k) \geq I(X;\hat{Y}).$$

Due to the invertible re-parameterization, each information path corresponds to many different DNN's with possibly very different architectures.

Background

Information Bottleneck Optimal Representations

What is the optimal representations of X w.r.t Y?

Sufficient Statistics: S(X) that captures all the information that X has on Y, I(S(X),Y) = I(X,Y)

Minimal Sufficient Statistics: T(X) that are the simplest sufficient statistics

Formulate through Markov Chain: Y->X->S(X)->T(X)

$$T(X) = \arg \min_{S(X):I(S(X);Y)=I(X;Y)} I(S(X);X)$$

Solution: Information Bottleneck (IB) tradeoff

Experiments

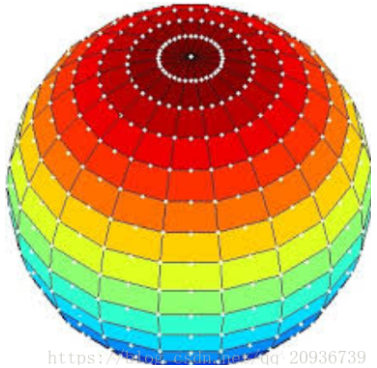
Task: Binary Decision Rules that are invariant under O(3) rotation of the sphere
Input: 12 binary inputs, uniformly distributed points on a 2D sphere
Output: 0/1
Model: 7 layers fully connect layers, 50 different random initialized network
2^12 = 4096 possible input patters

To calculate MI, in each layer, discretize output activations into 30 equal intervals, and get the discrete joint distributions

$$P(T_i, X) \qquad P(T_i, Y) = \sum_x P(x, Y) P(T_i|x)$$

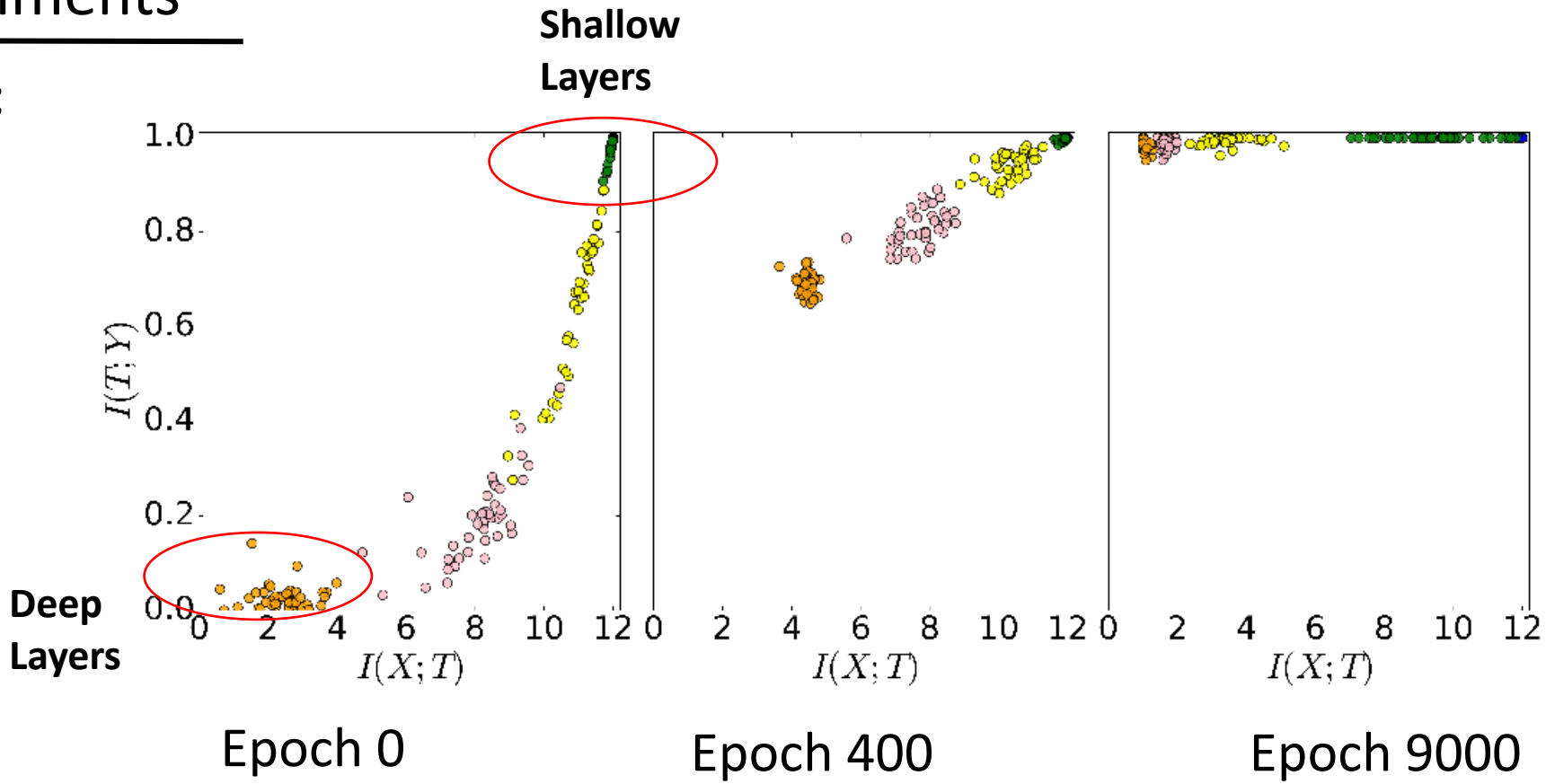
Based on the above, calculate the following for each hidden layer

$$I(X; T_i) \qquad I(T_i; Y)$$



Experiments

Results:

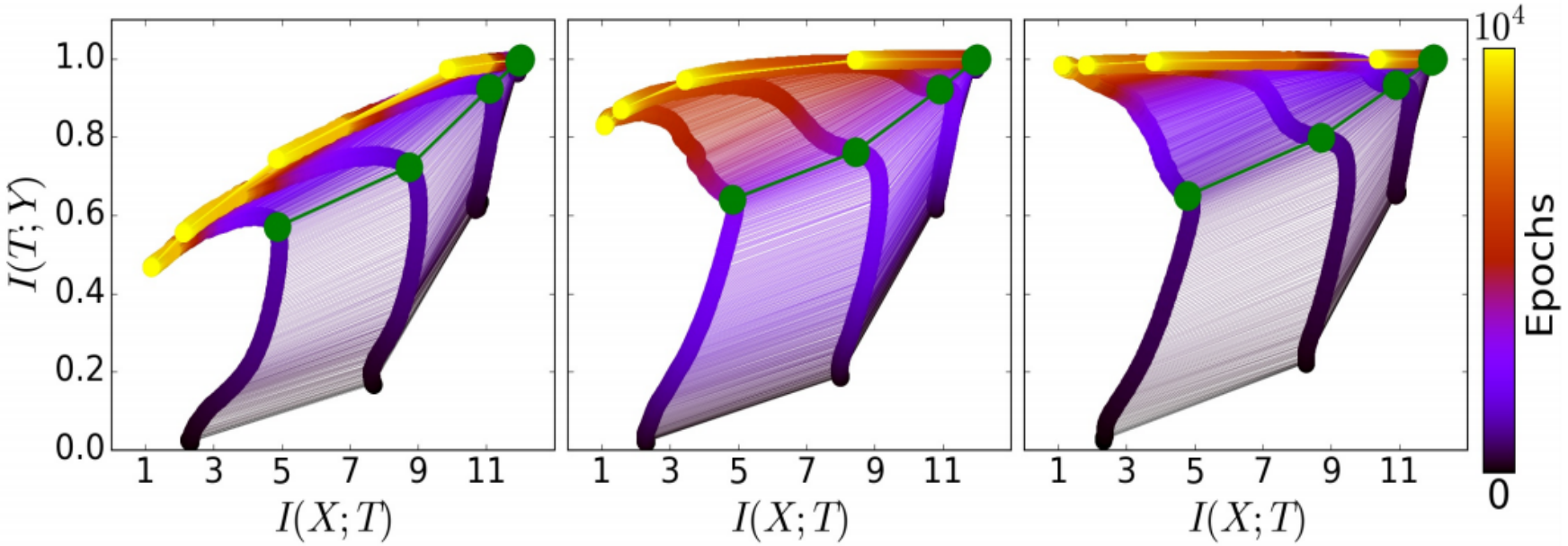


1. At the beginning, deeper layer fail to preserve relevant information
2. During SGD, I_y increase, and then I_x decrease => compress the representation
3. Different randomized networks show similar path, and converge to nearby points in information plane.

[1] https://www.youtube.com/watch?v=P1A1yNsxMjc&feature=youtu.be

Experiments

After Average



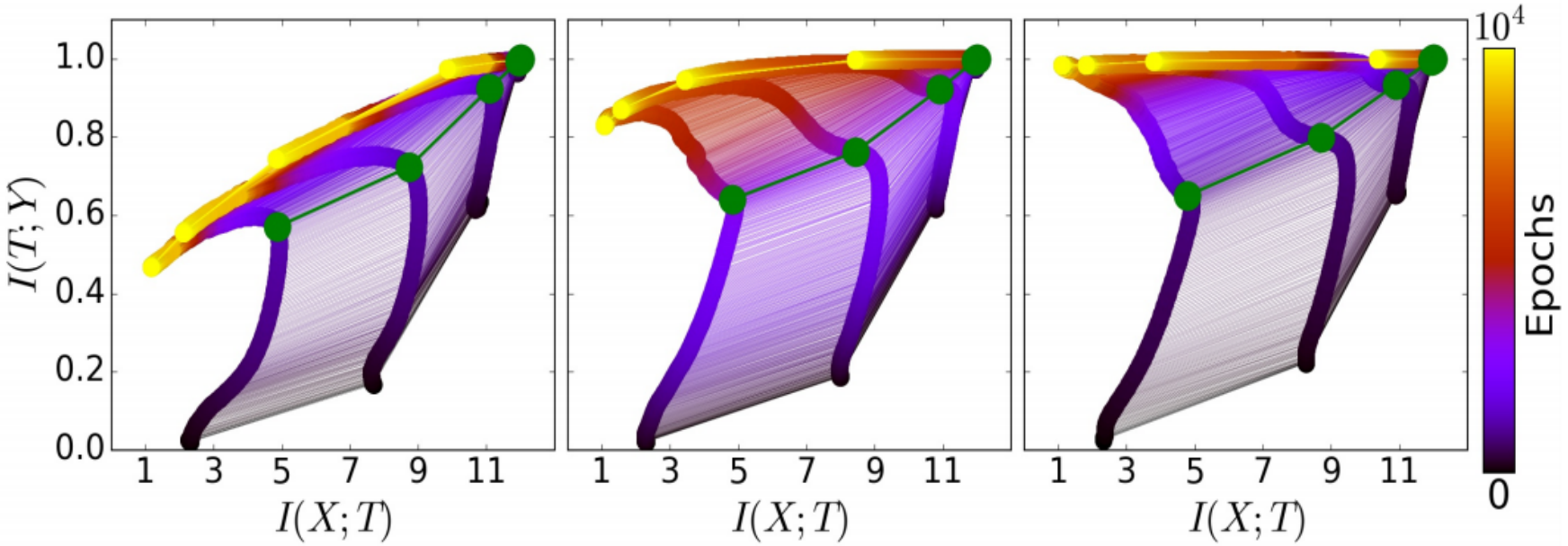
5% data 45% data 85%

Two phases:

1. **Empirical error minimization (ERM):** a few hundred epochs
   I_y increase, preserving the DPI order (lower layers have higher information)
   Explain: cross-entropy loss minimization
2. **Representation Compression:**
   I_x decreases and layers lose irrelevant information until convergence (yellow points)

Experiments

After Average

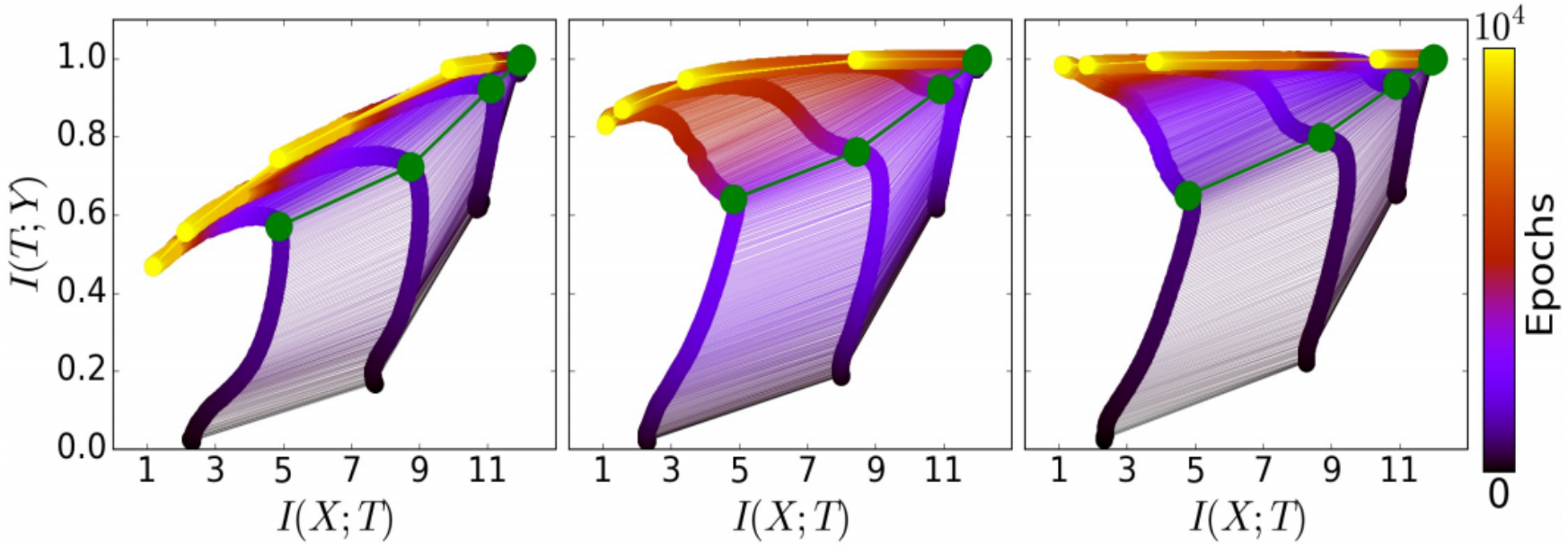


5% data 45% data 85%

Conclusion 1: Most of the training g epochs in standard DL are spent on compression of the input to efficient representation and not on fitting the training labels.

Experiments

After Average

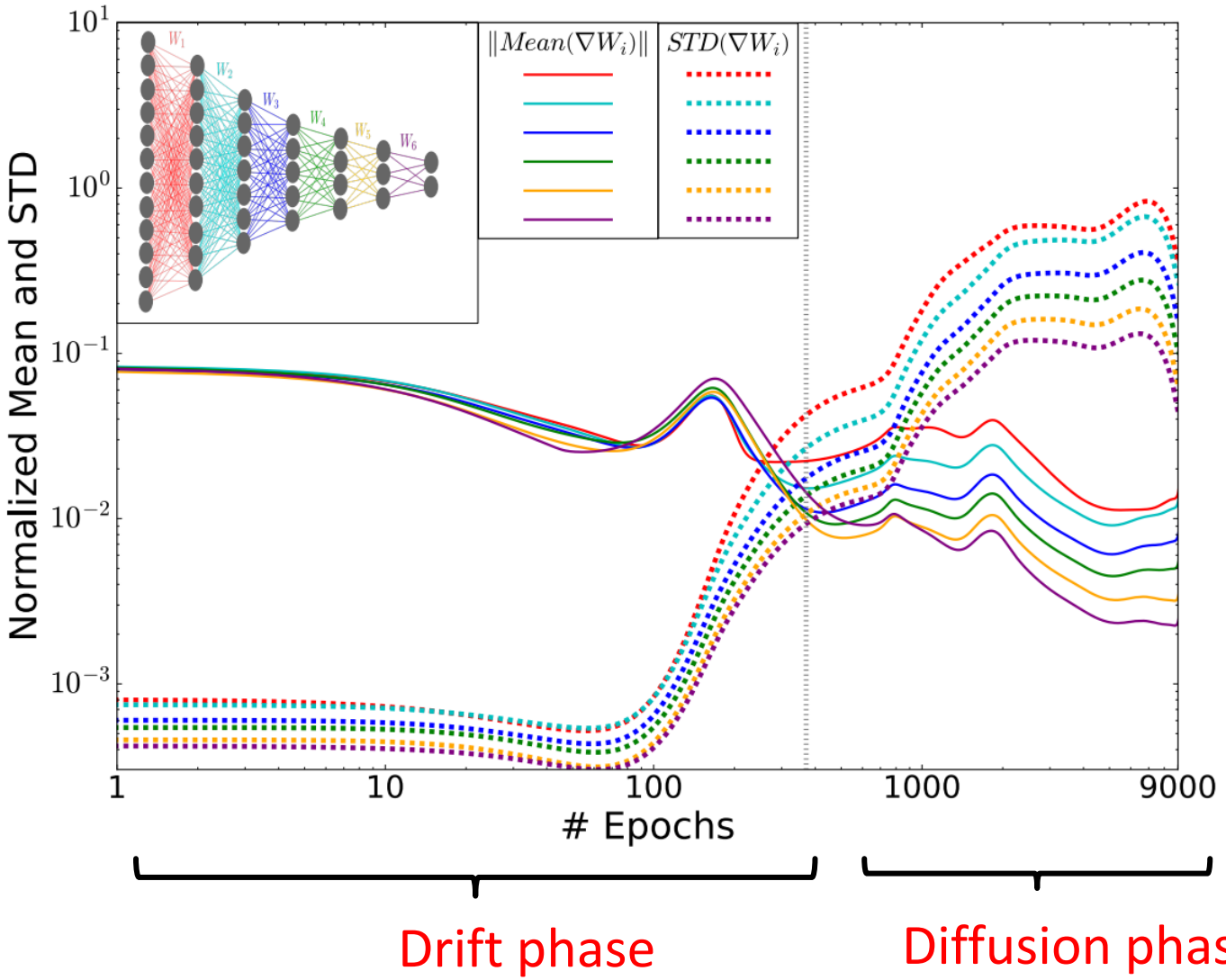


5% data 45% data 85%

ERM phases are similar but compression phases are different (5%, 45%, 85%)
Small data, I_y decrease, reduce layer's label info (overfitting!)
Large data, I_y increase, increase label info

Experiments

SGD: normalized mean and STD of gradients w.r.t epochs

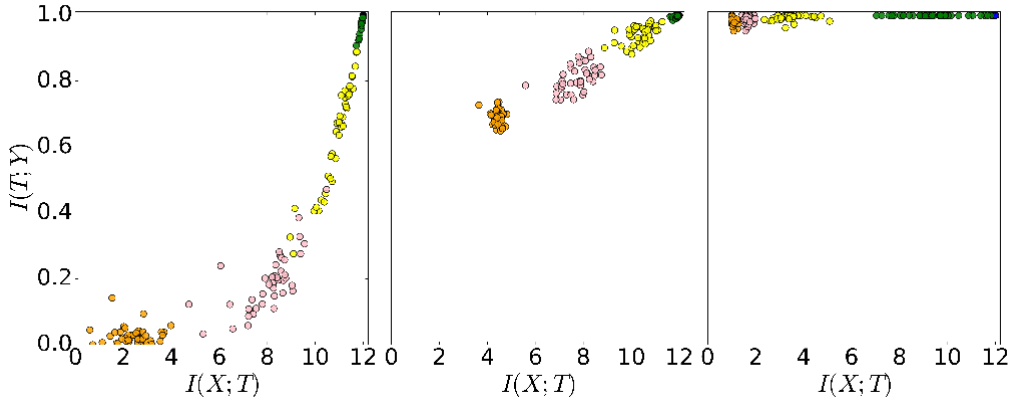


Drift phase:
Mean >> STD, small gradient stochasticity

Diffusion phase:
Mean < STD, behave like Gaussian noise with small mean
-> empirical error saturates and SGD is dominated by its fluctuation

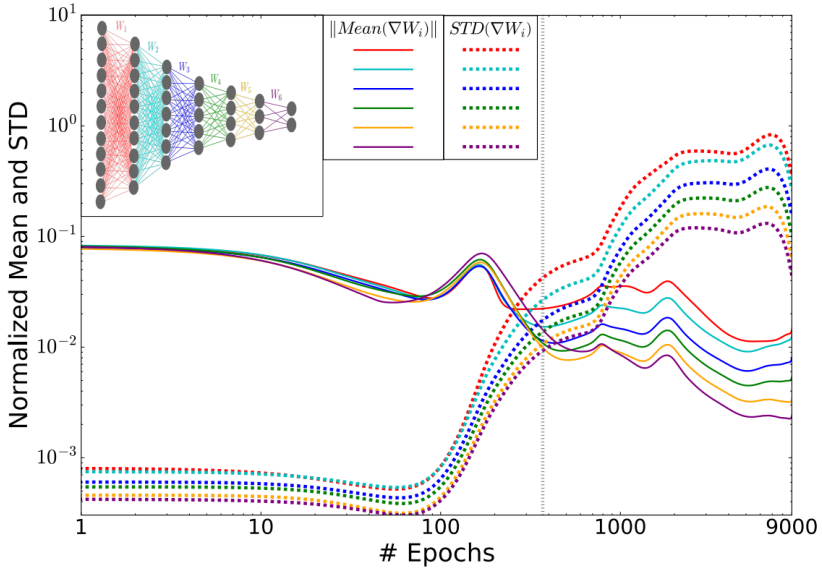
Claim: these two phases correspond and explain the ERM and compression phases

Experiments



Drift phase:
Increase I_y since it quickly reduces empirical error

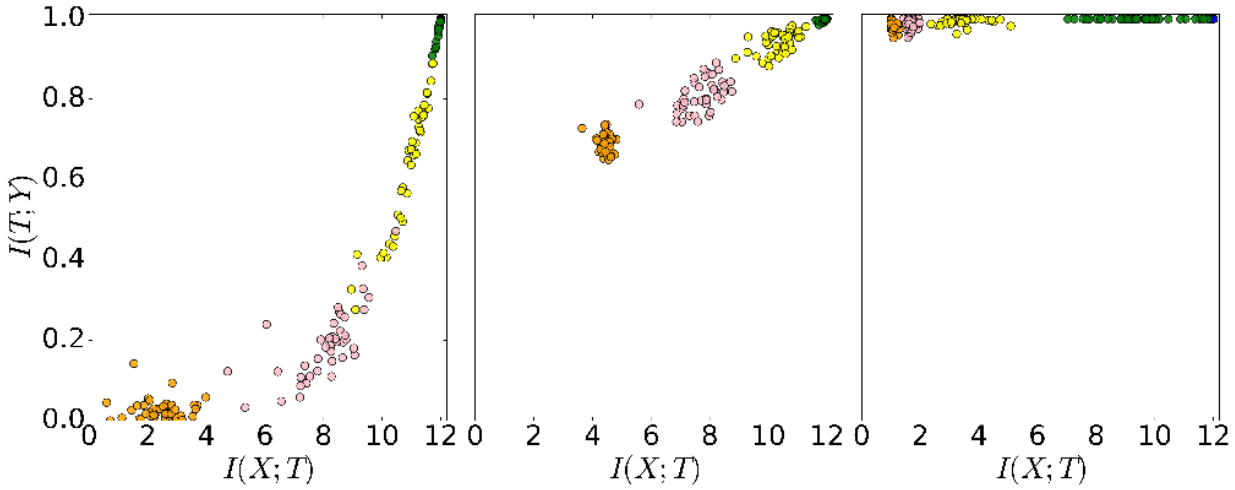
Diffusion phase:
1. Add random noises to the weights
2. Minimize I_x for every layer to get more compressed representation (Focker-Planck equation)

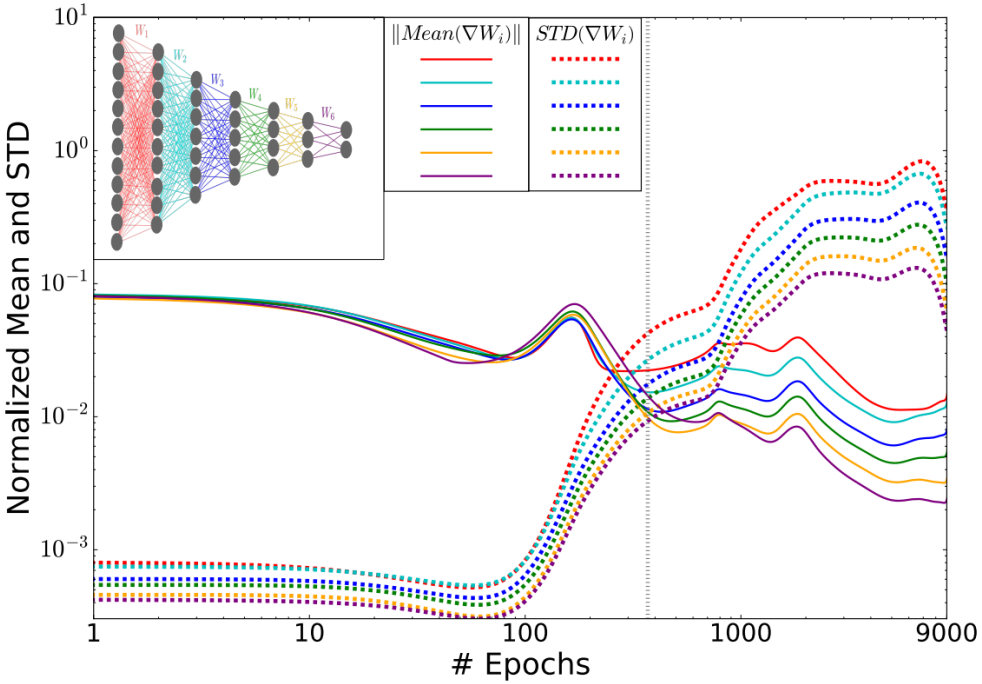


Observation: No indication for vanishing connections or norm decrease near convergence

Finding: Consistent with previous work:
Explicit forms of regularization, such as weight decay, dropout, and data augmentation, do not adequately explain the generalization error of DNNs

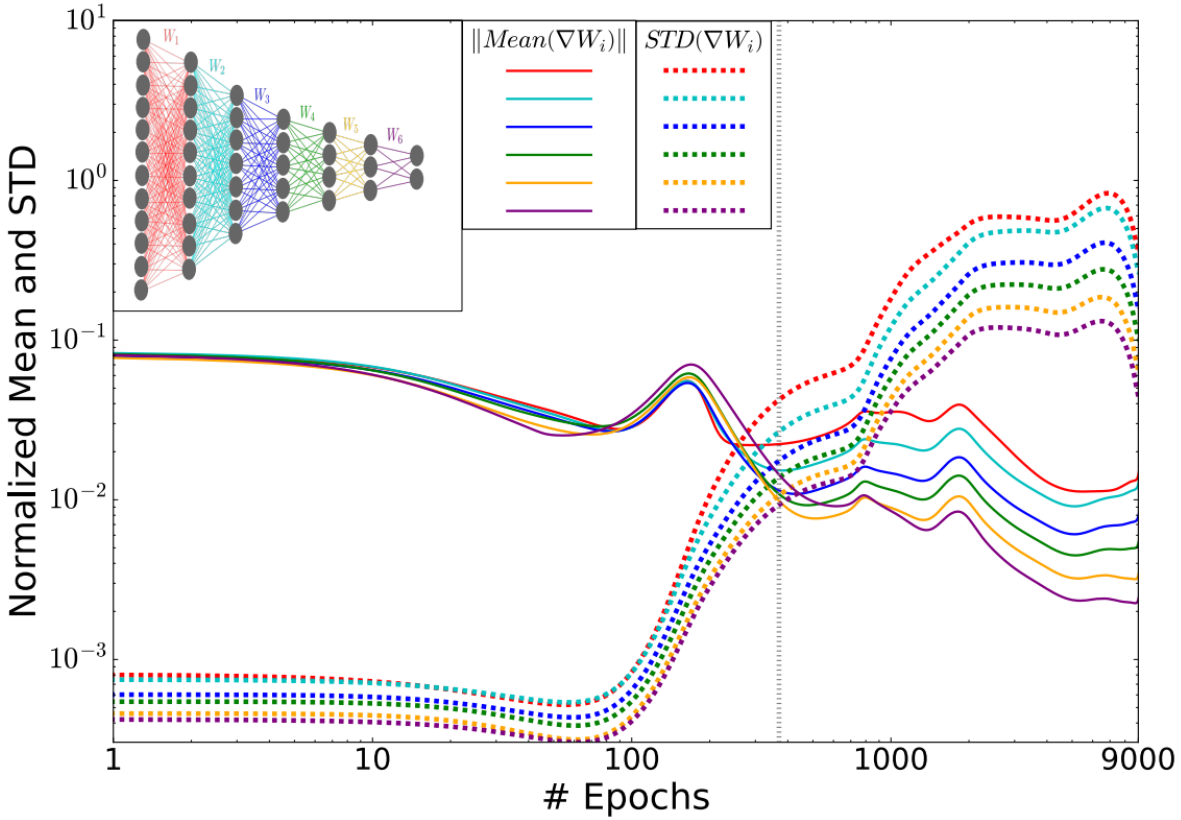
Experiments





Conclusion 2: The representation compression phase begins when the training errors becomes small and the Stochastic Gradient Decent (SGD) epochs change from a fast drift to smaller training error into a stochastic relaxation, or random diffusion, constrained by the training error value

Experiments



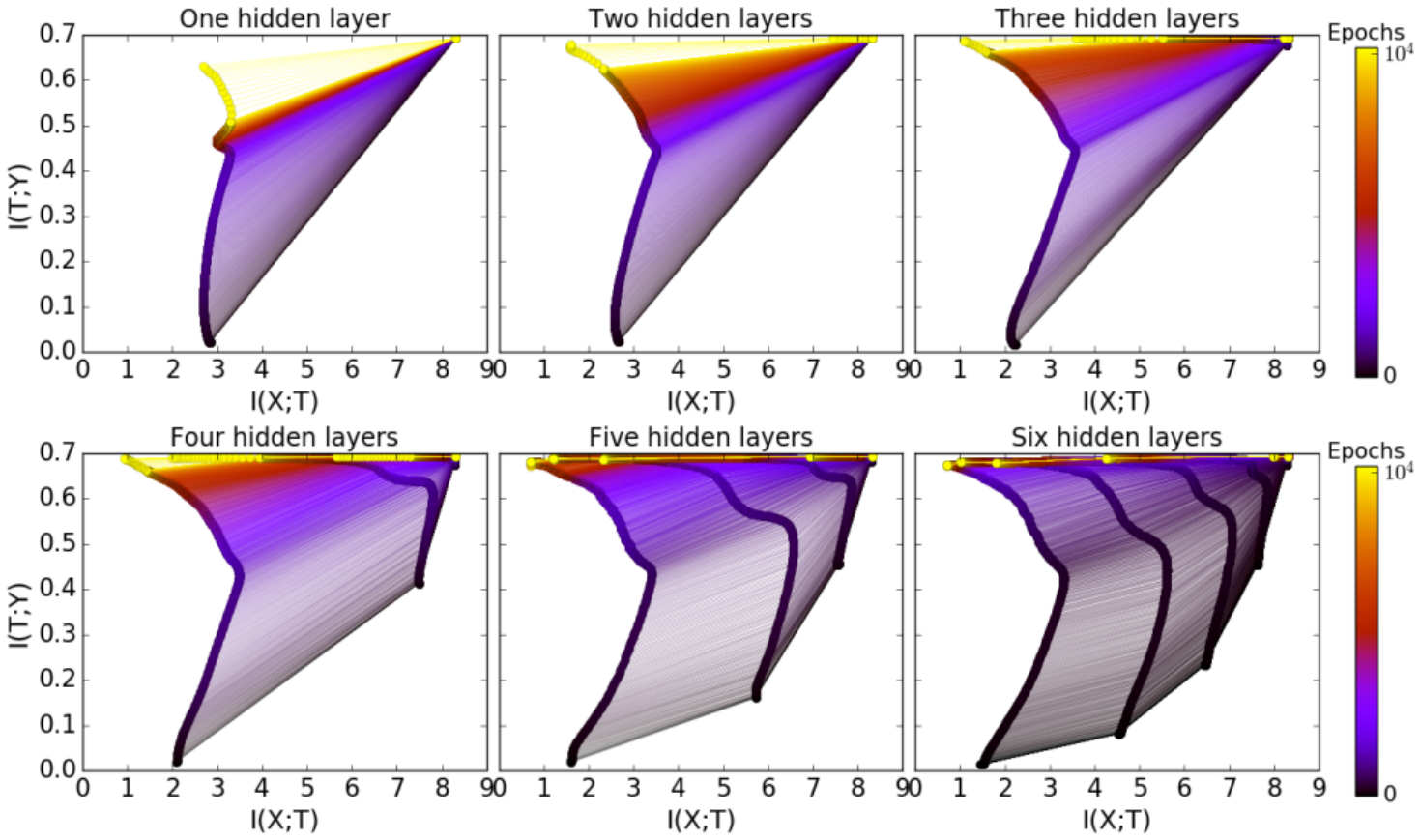
Observation: Correlations between weights of different neurons in the same layer, which converge to essentially the same point in the plane, was very small

Finding: A huge number of different networks with essentially optimal performance, and *attempts to interpret single weights or even single neurons in such networks can be meaningless*.

Experiments

Questions: What is the benefit of the hidden layers?

Setting: 6 different architectures with 1-6 hidden layers



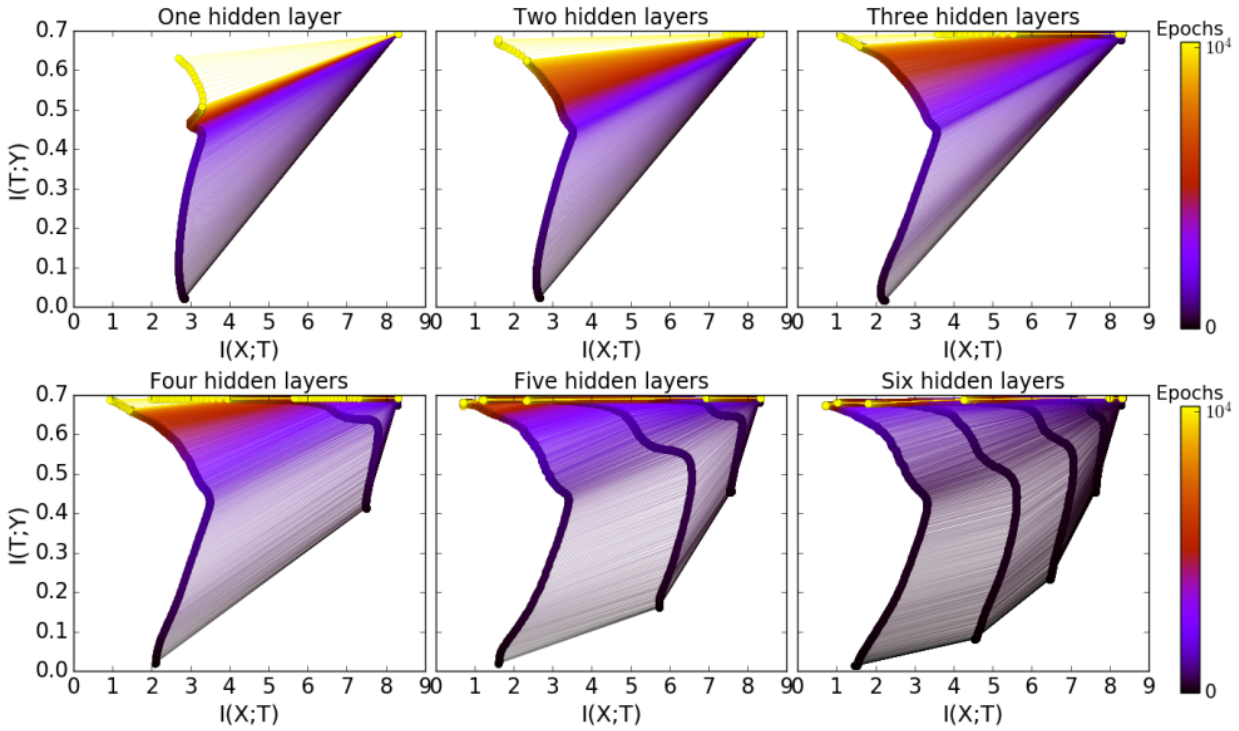
Conclusions:

1. Adding hidden layers dramatically reduces the number of training epochs for good generalization

2. The compression phase of each layer is shorter when it starts from a previous compressed layer

3. The compression is faster for the deeper (narrower and closer to the output) layers.

Experiments

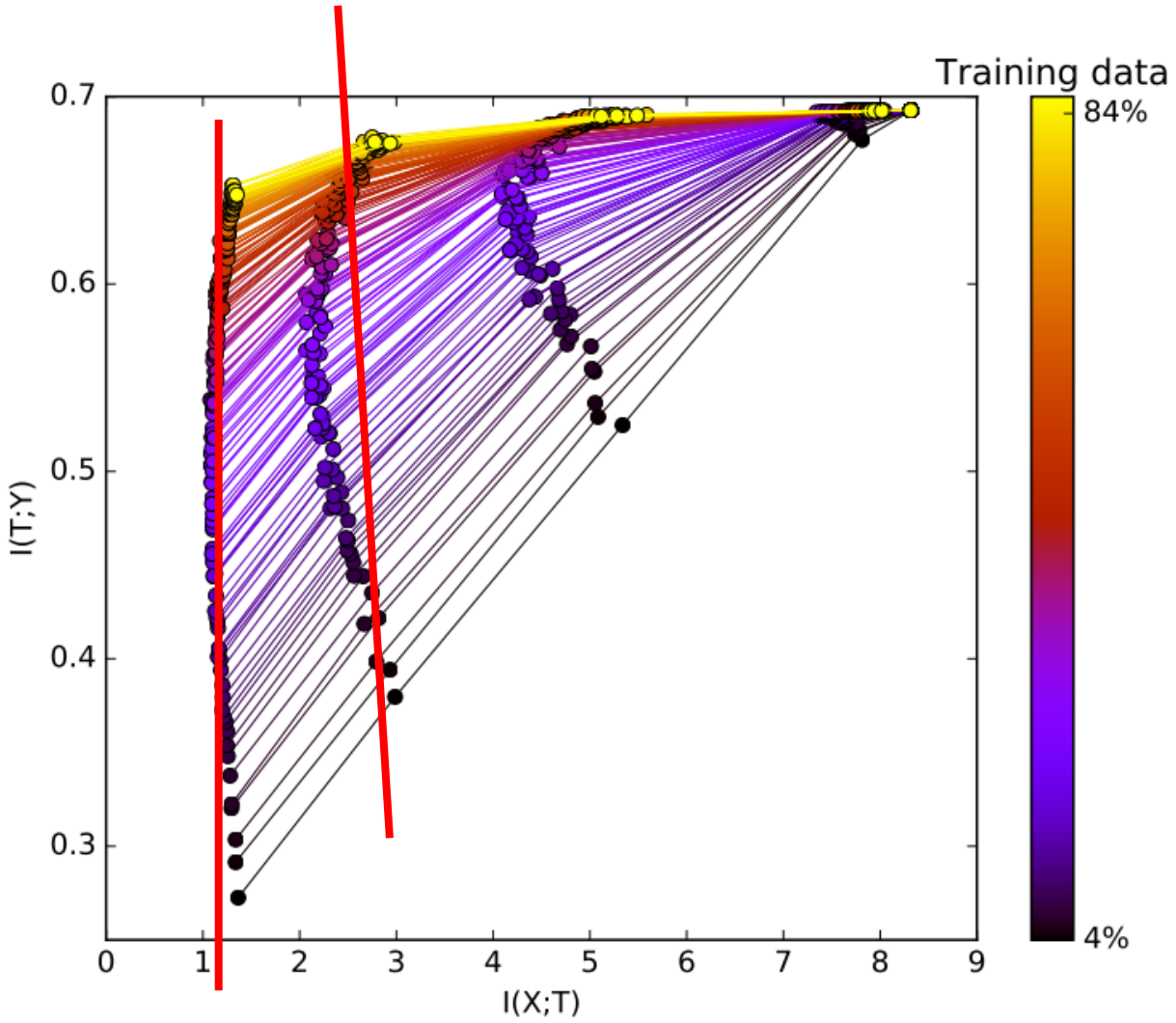


Conclusion 3: The converged layers lie on or very close to the Information Bottleneck (IB) theoretical bound, and the maps from the input to any hidden layer and from this hidden layer to the output satisfy the IB self-consistent equations.

Conclusion 4: The training time is dramatically reduced when adding more hidden layers. Thus the main advantage of the hidden layers is computational.

Experiments

Evolution of the layers with training sample size



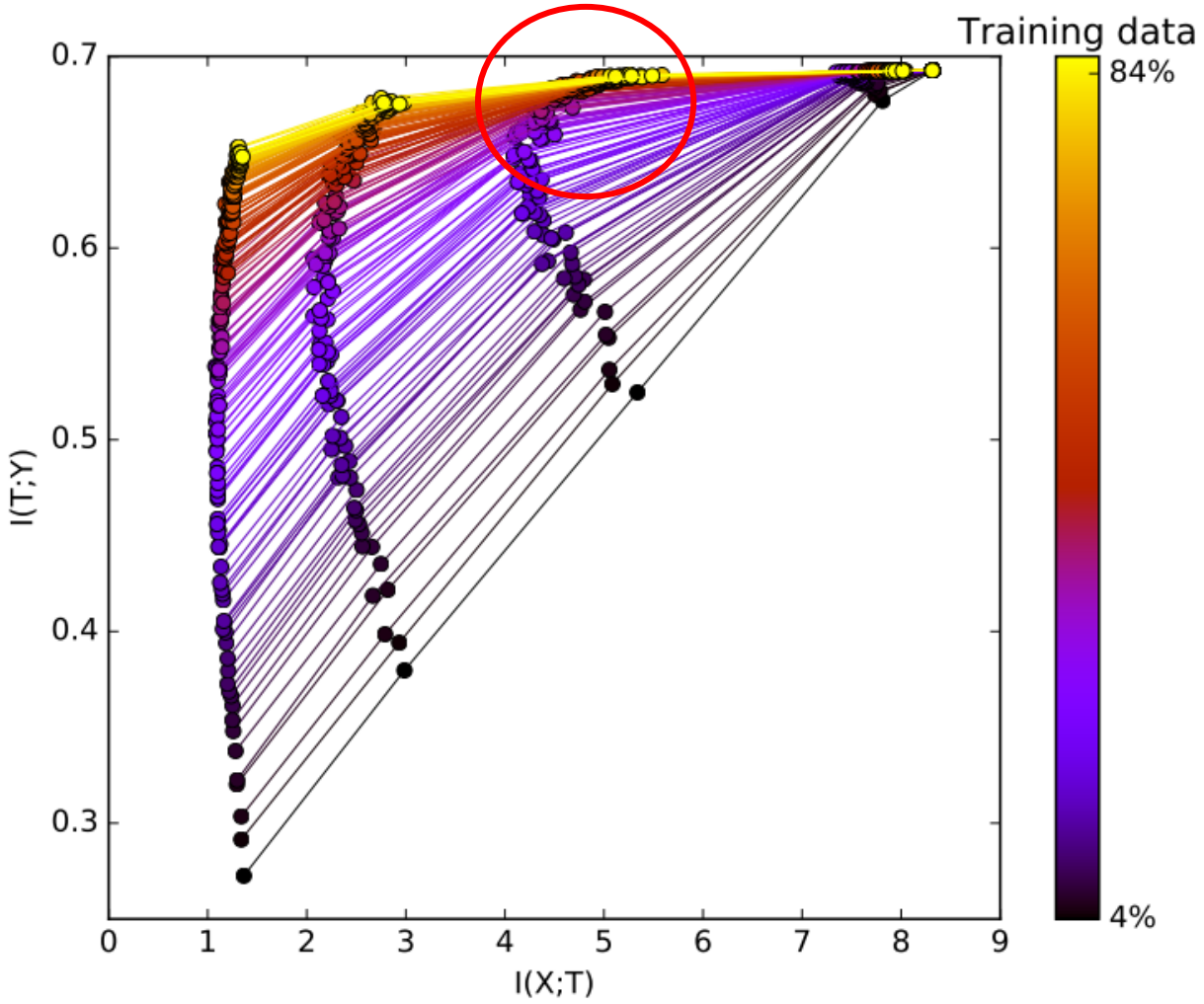
with increasing training size, I_y is pushed up and gets closer to the theoretical IB bound

Besides,
the converged layers for different training sizes lie on a smooth line for each layer

the layers converge to specific points on the finite sample information curve

Experiments

Evolution of the layers with training sample size



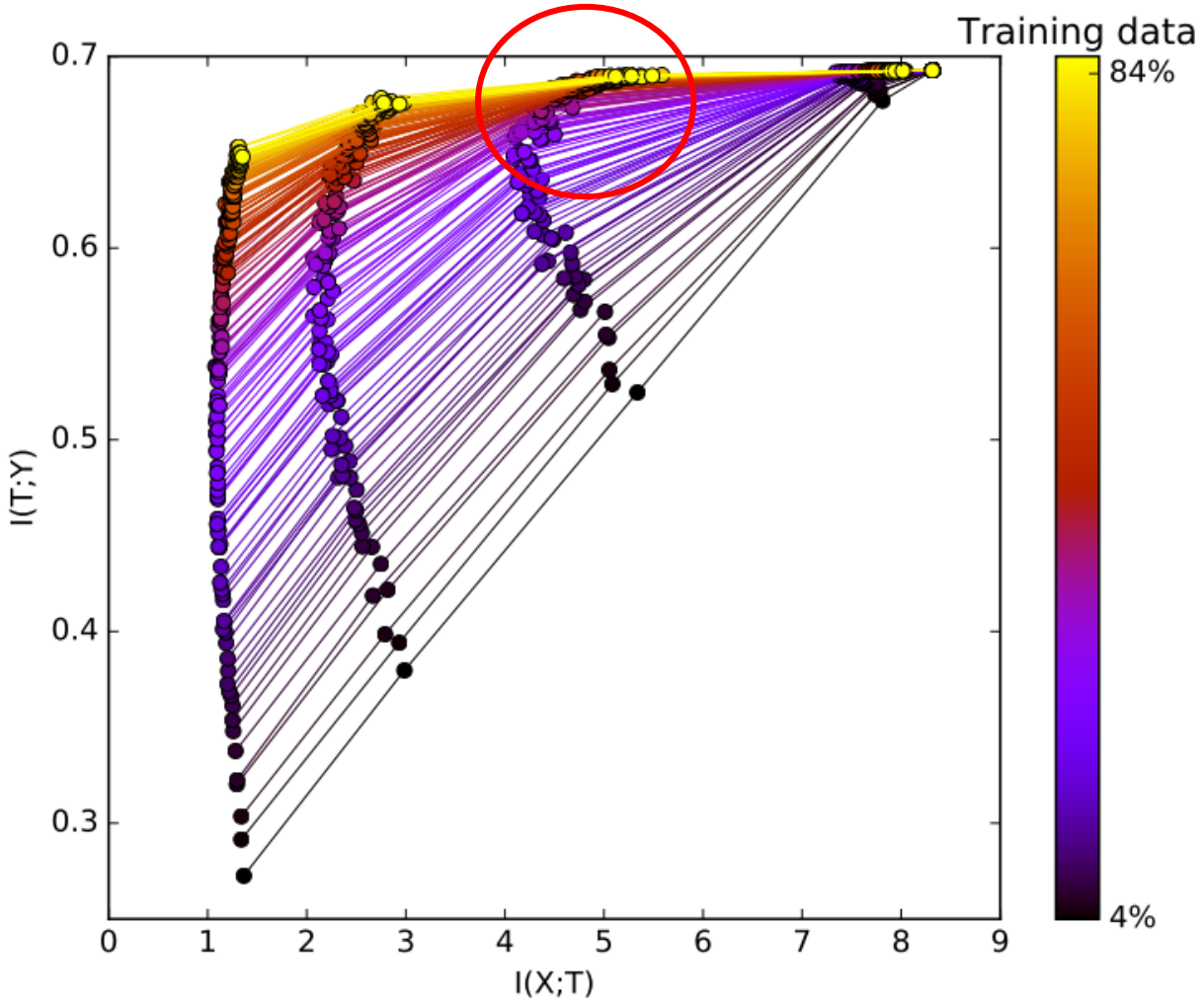
In the lower layers, training size hardly changes the information at all, since random weights keep most mutual information on both X and Y.

However, deeper layers learn to preserve more of the information on Y and better compress the irrelevant information in X.

With larger training samples more details on X become relevant for Y => there is a shift to higher I_x in the middle layers

Experiments

Evolution of the layers with training sample size



Conclusion 5:

The hidden layers appear to lie close to critical points on the IB bound, which can be explained by critical slowing down of the stochastic relaxation process.


THANKS