Deep Neural Networks and the Information Bottleneck method.

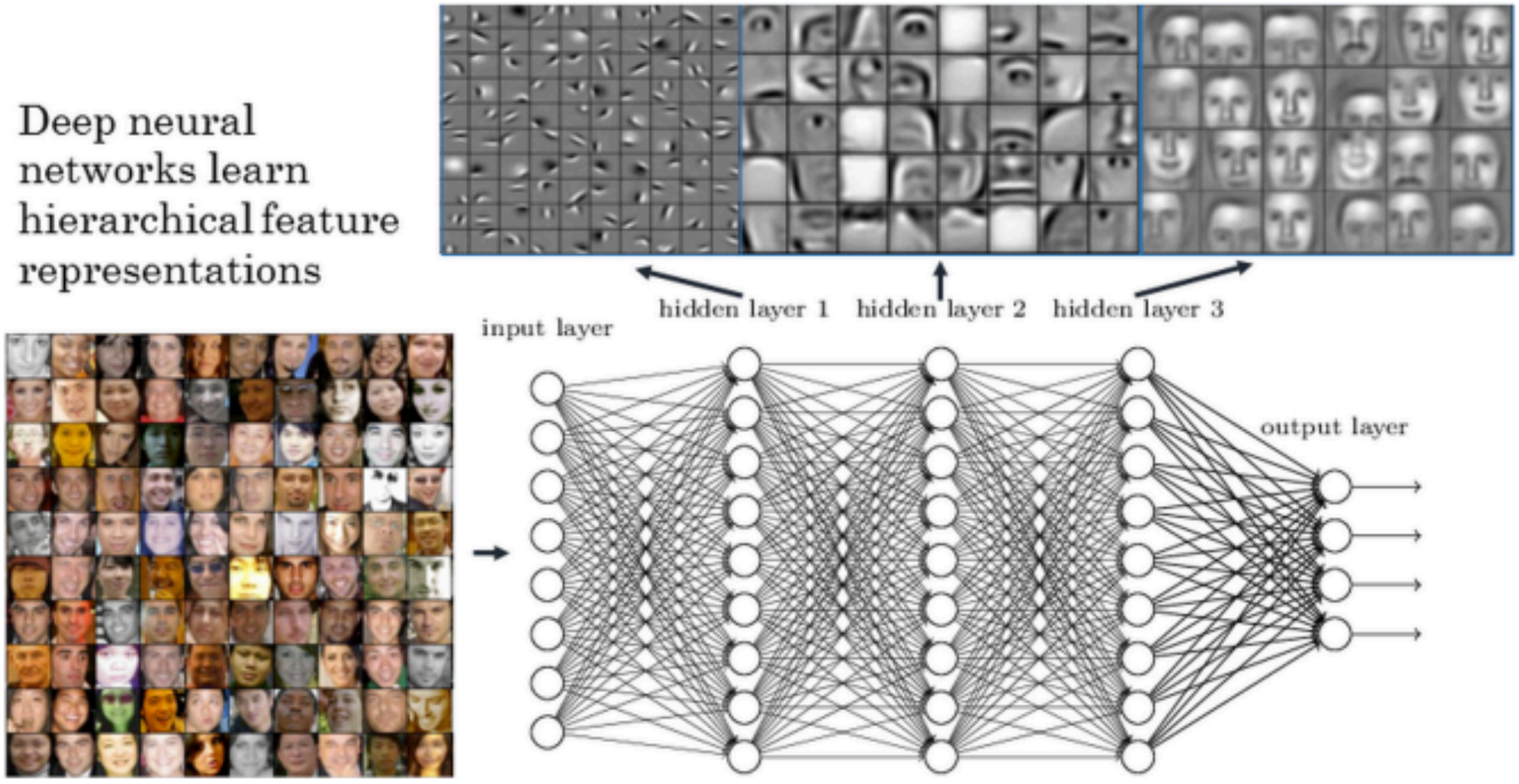
Piotr Warchoł

Theory of Complex Systems Department

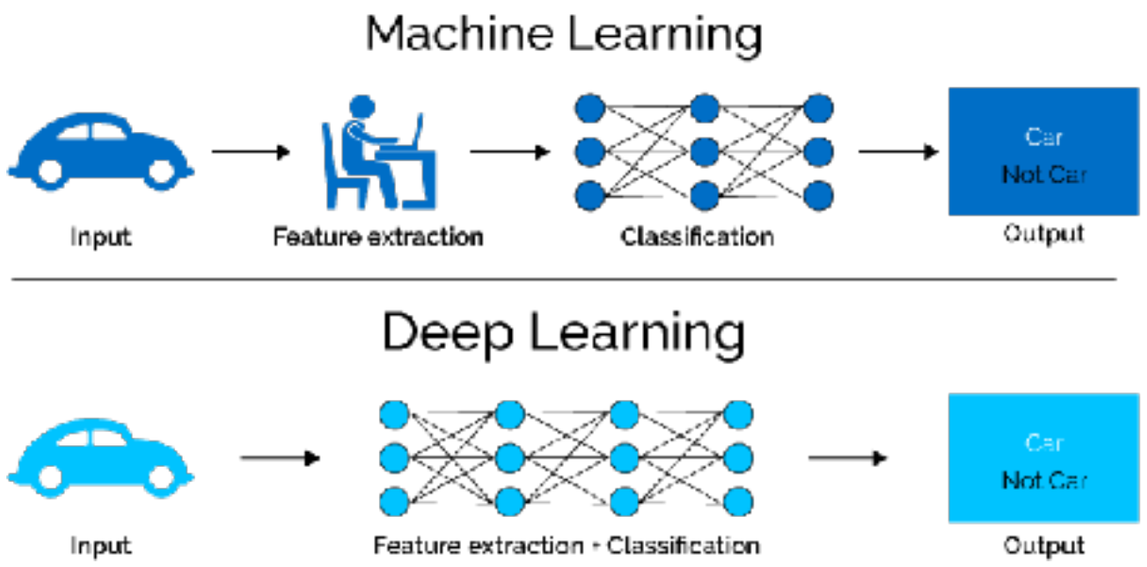
Faculty of Physics, Astronomy and Applied Computer Science

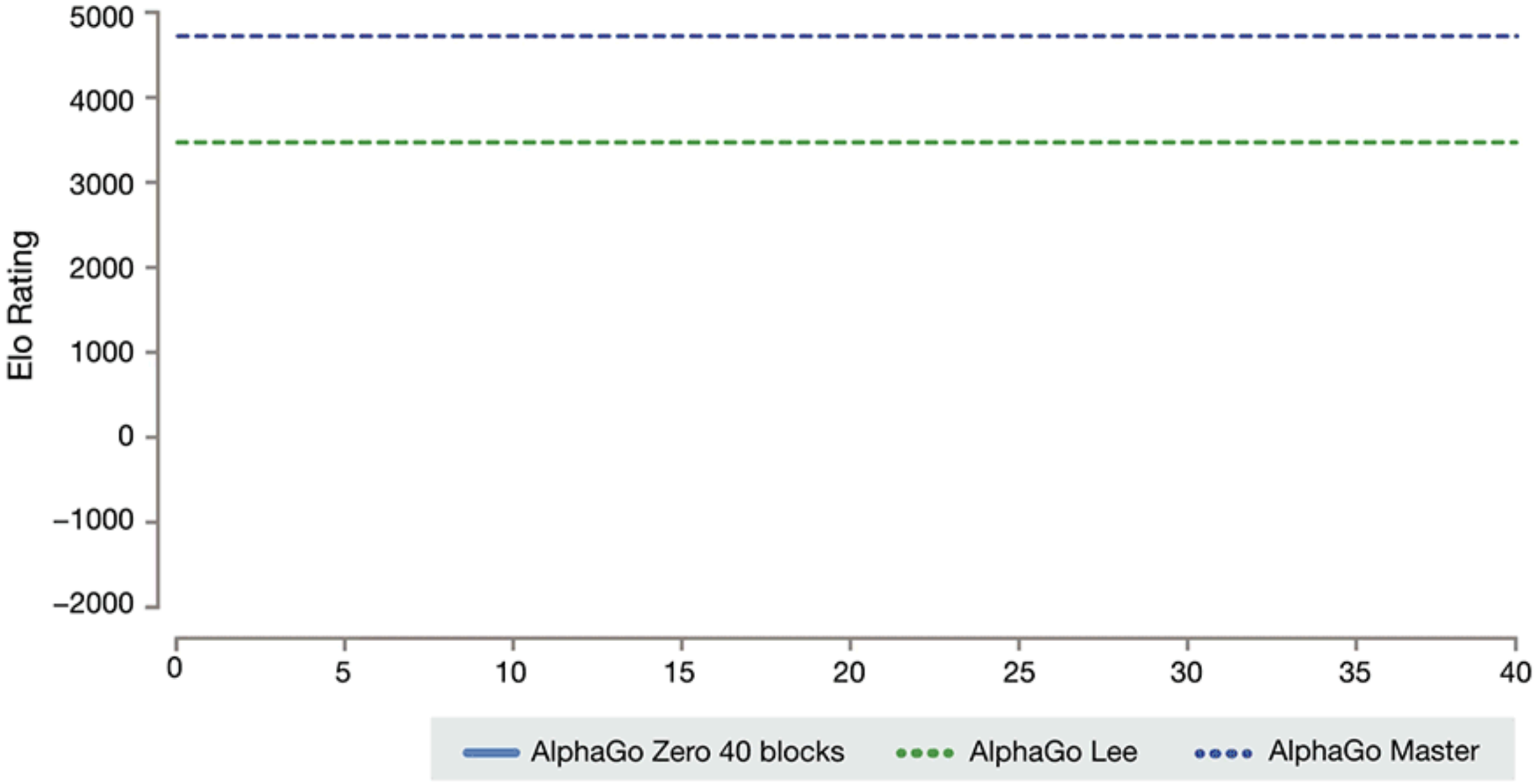
JAGIELLONIAN UNIVERSITY
IN KRAKOW

Deep Learning algorithms - a class of machine learning algorithms which uses a cascade of multiple layers that learn levels of representations that correspond to different levels of abstraction



Can use raw data!




Elo Rating
5000
4000
3000
2000
1000
0
−1000
−2000
0
5
10
15
20
25
30
35
40
AlphaGo Zero 40 blocks
AlphaGo Lee
AlphaGo Master

A
B
C
D
E
F

Real-Time User-Guided Image Colorization with Learned Deep Priors

Richard Zhang* Jun-Yan Zhu* Phillip Isola
Xinyang Geng Angela S. Lin Tianhe Yu Alexei A. Efros

Speech recognition

Music Generation

Generating visual indicated sounds

Reenacting people talking (visual and sound)

Pixel restoration

Real-time multi-person pose estimation

Beating games

Lip reading

Topic classification

Sentiment analysis

Language translation

Object detection

Text generation

Speech generation

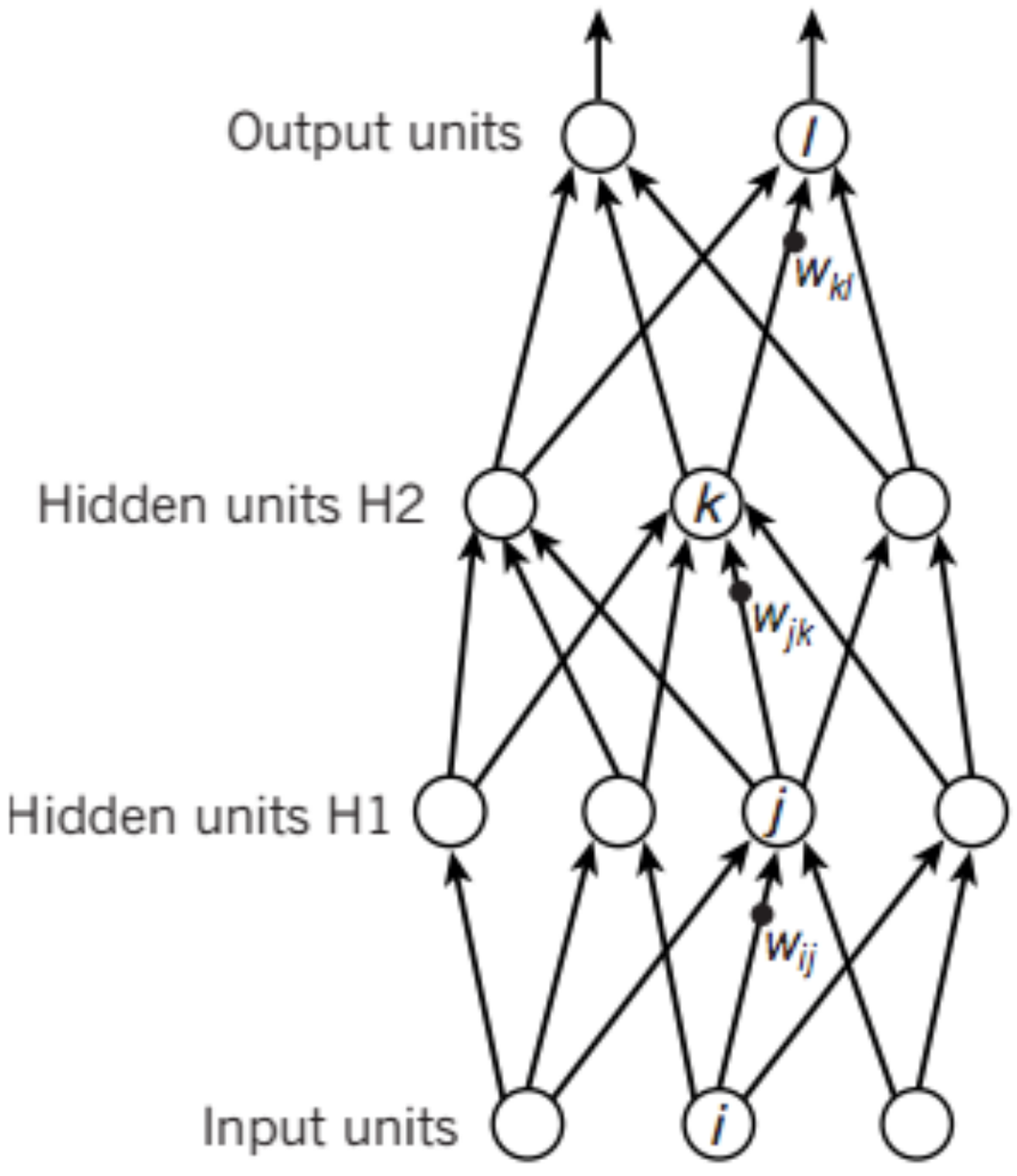
Image segmentation

Image caption generation

Image generation and complementation

Data analysis in science

Drug discovery



$$y_l = f(z_l)$$

$$z_l = \sum_{k \,\varepsilon\, \mathrm{H2}} w_{kl}\, y_k$$

$$y_k = f(z_k)$$

$$z_k = \sum_{j \,\varepsilon\, \mathrm{H1}} w_{jk}\, y_j$$

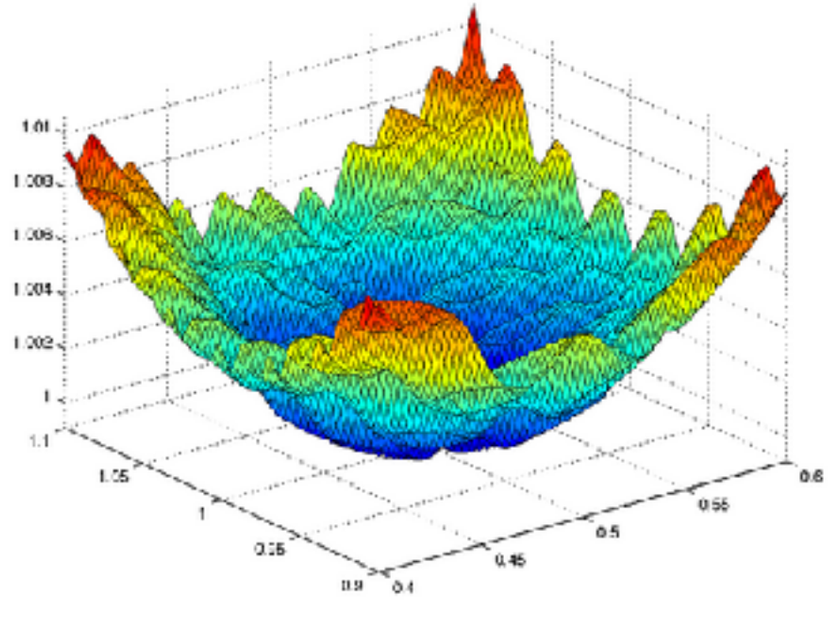
$$y_j = f(z_j)$$

$$z_j = \sum_{i \,\varepsilon\, \mathrm{Input}} w_{ij}\, x_i$$

- f(z) - some nonlinear, continuous function
- E(w) - cost function which we minimize
- How do we change w to make E smaller?

(for all inputs)

- Efficiency! - Stochastic gradient descent



$$\frac{\partial E}{\partial W_{ik}} \approx \frac{E(W + \epsilon e_{ik}) - E(W)}{\epsilon}$$

Inefficient to calculate separately for every weight

Backpropagation - using the chain rule

$y_l = f(z_l)$

$z_l = \sum_{k \,\varepsilon\, \text{H2}} w_{kl}\, y_k$

$y_k = f(z_k)$

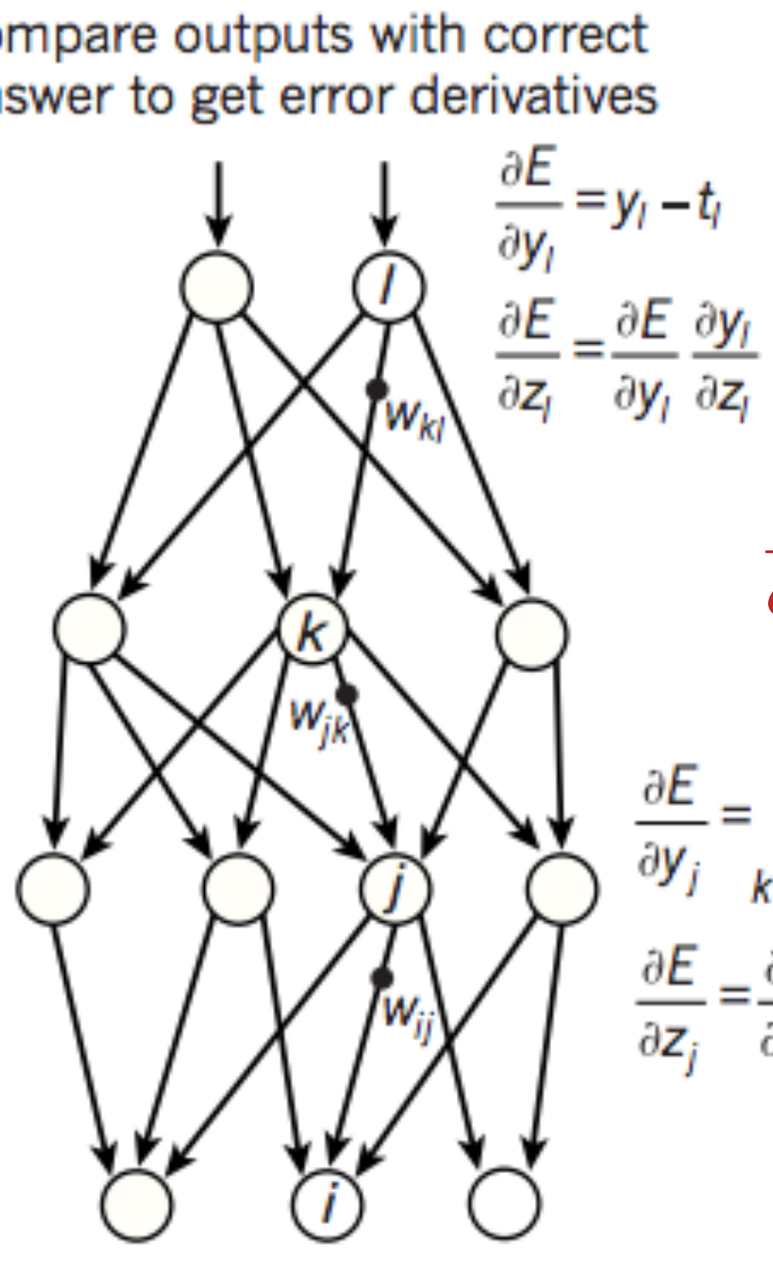
$z_k = \sum_{j \,\varepsilon\, \text{H1}} w_{jk}\, y_j$

$y_j = f(z_j)$

$z_j = \sum_{i \,\varepsilon\, \text{Input}} w_{ij}\, x_i$

$$\frac{\partial E}{\partial y_k} = \sum_{l \,\varepsilon\, \text{out}} w_{kl} \frac{\partial E}{\partial z_l}$$

$$\frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial z_k}$$



$$\frac{\partial E}{\partial W_{kl}} = y_k \frac{\partial E}{\partial z_l}$$

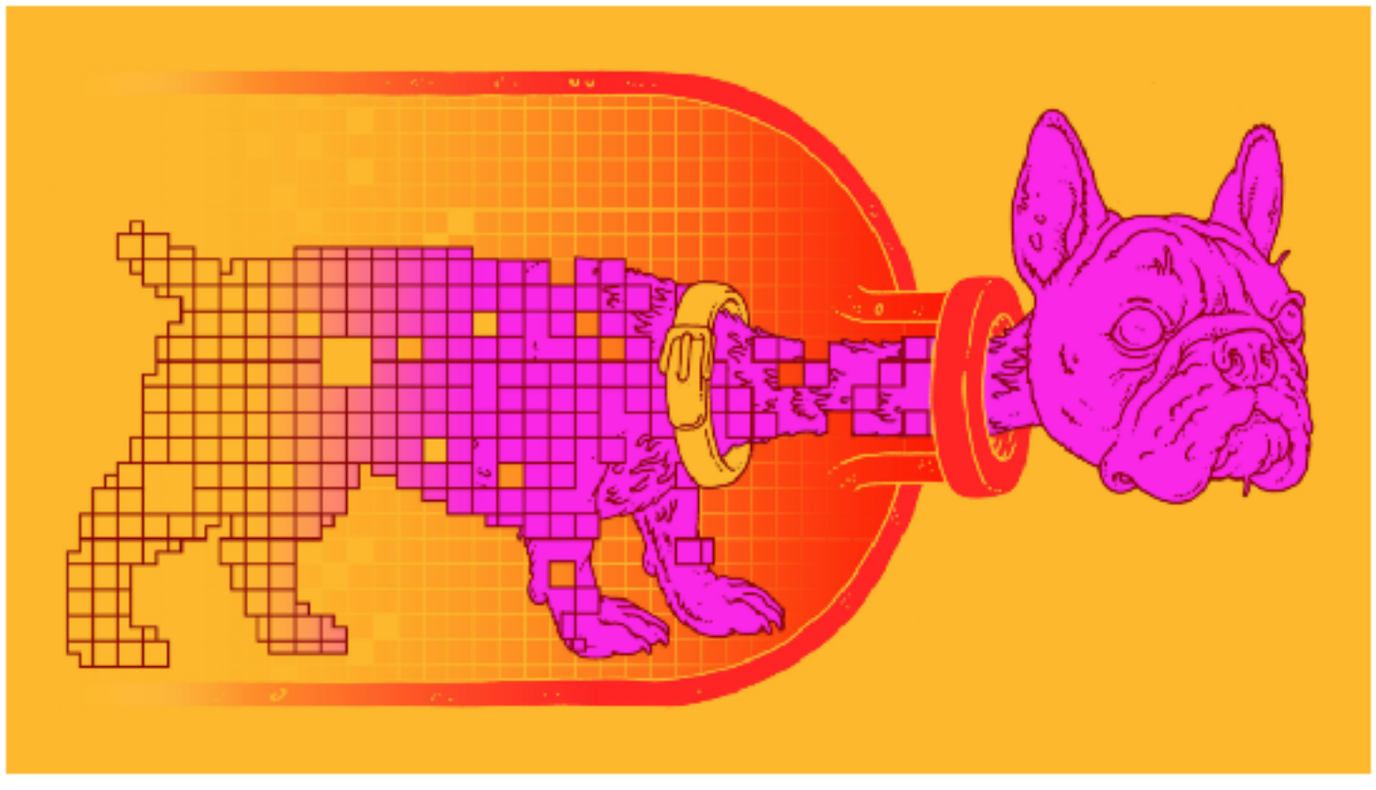
$$\frac{\partial E}{\partial y_j} = \sum_{k \,\varepsilon\, \text{H2}} w_{jk} \frac{\partial E}{\partial z_k}$$

$$\frac{\partial E}{\partial z_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial z_j}$$

Opening the black box of Deep Neural Networks via Information.

29.03.2017

David Schwartz-Ziv, Naftali Tishby



New Theory Cracks Open the Black Box of Deep Learning - Quanta Magazine

- Expressivity/expressibility - what class of functions can the neural network express, with what amount of resources (trainable parameters)?

- Trainability/Learnability - how rapidly and to what extent, can the network learn? What amount of train data is needed?

- Generalization - how come no overfitting?

- Interpretability- insight in terms of what is the process in which the prediction is made

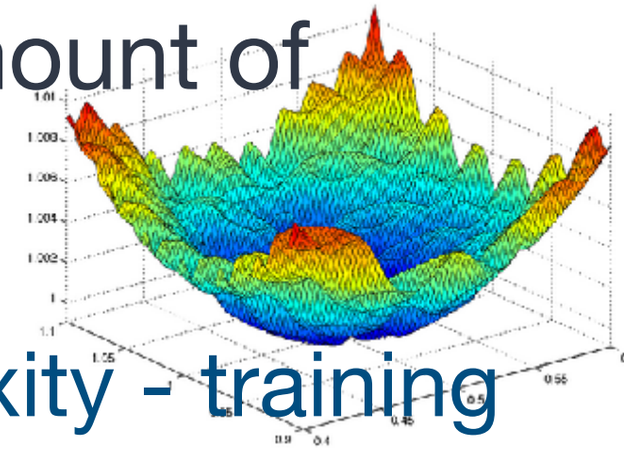
- Architecture - what is the optimal architecture for a given task?

- Expressivity/expressibility - what class of functions can the neural network express, with what amount of resources (trainable parameters)

The universal approximation theorem - feed forward neural networks with a single hidden layer (with some non-linear activation function) can be used to approximate any continuous function to any desired precision.

- First version - 1989 - George Cybenko
- precise lower limits on size and number of layers still a research area

- Trainability/Learnability - how rapidly and to what extent, can the network learn? What amount of train data is needed?



Rephrased in terms of computational complexity - training even a simple 2-layer, 3-node neural network to global optimality is NP-complete in the worst case.

- Showed by Avrim Blum and Ronald Rivest in 1988

However, there is empirical evidence that depth of neural networks makes the performance of local minima close to that of global minima…

- Choromanska et al. by mapping to spherical spin-glass

- Generalization - how come no overfitting?

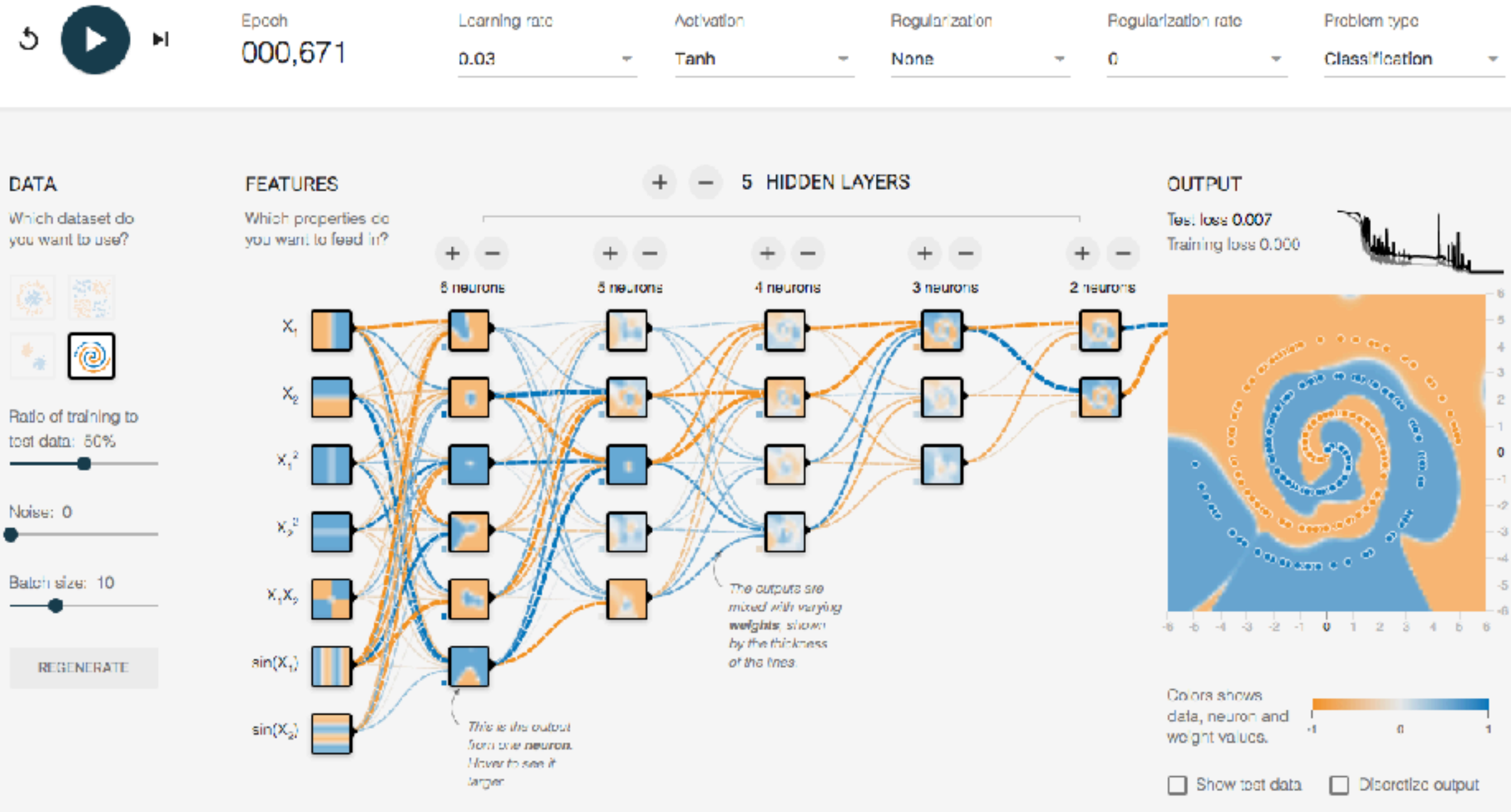
"Aparent paradox" - modern deep neural networks are able to fit randomly labeled data. "…simple depth two neural networks already have perfect finite sample expressivity as soon as the number of parameters exceeds the number of data points as it usually does in practice".

- Zhang et al. 2017 on image classification

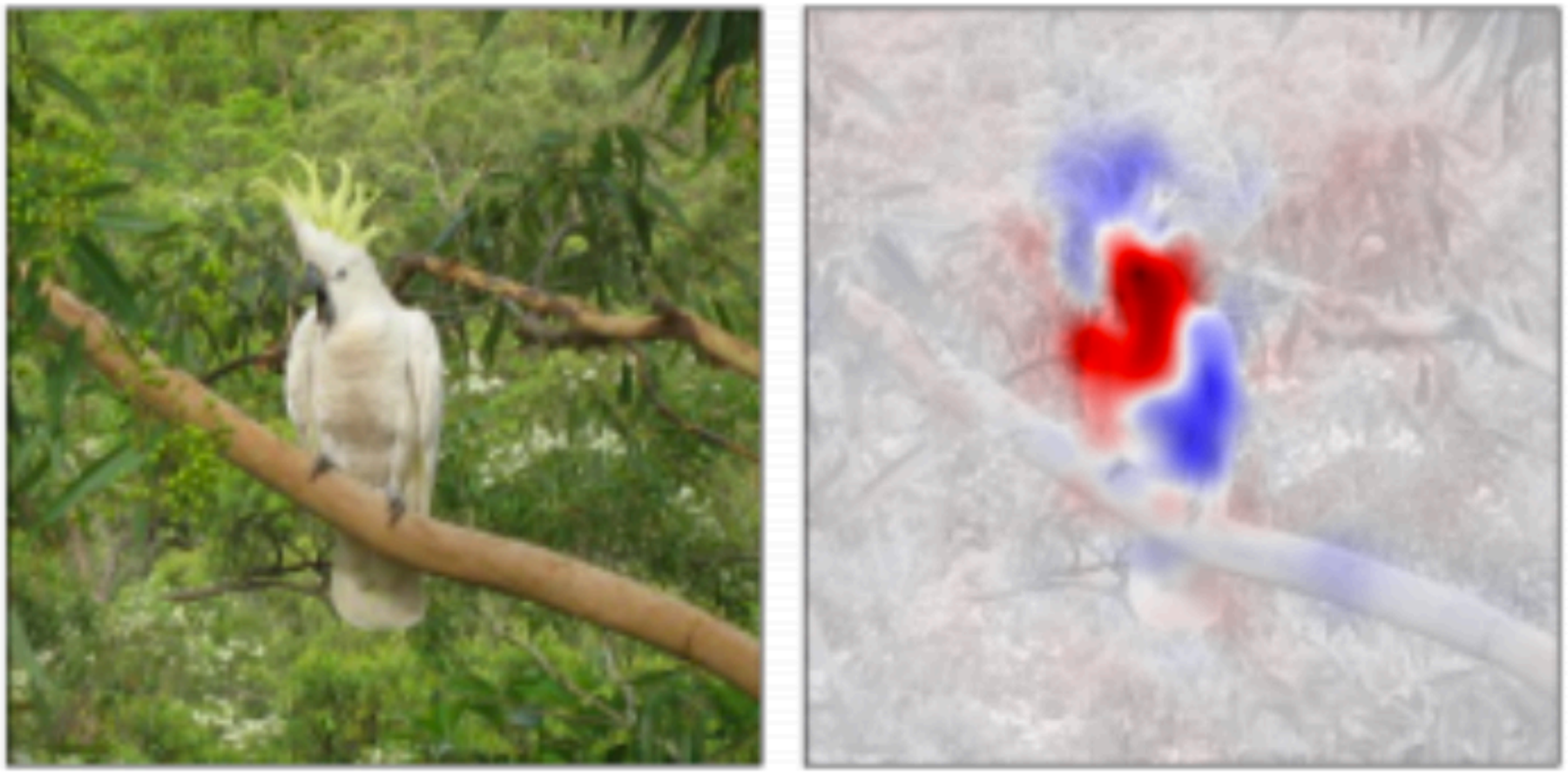
Generalization in deep learning: "This paper explains why deep learning can generalize well…"

- Kawaguchi et al. 16.10.2017

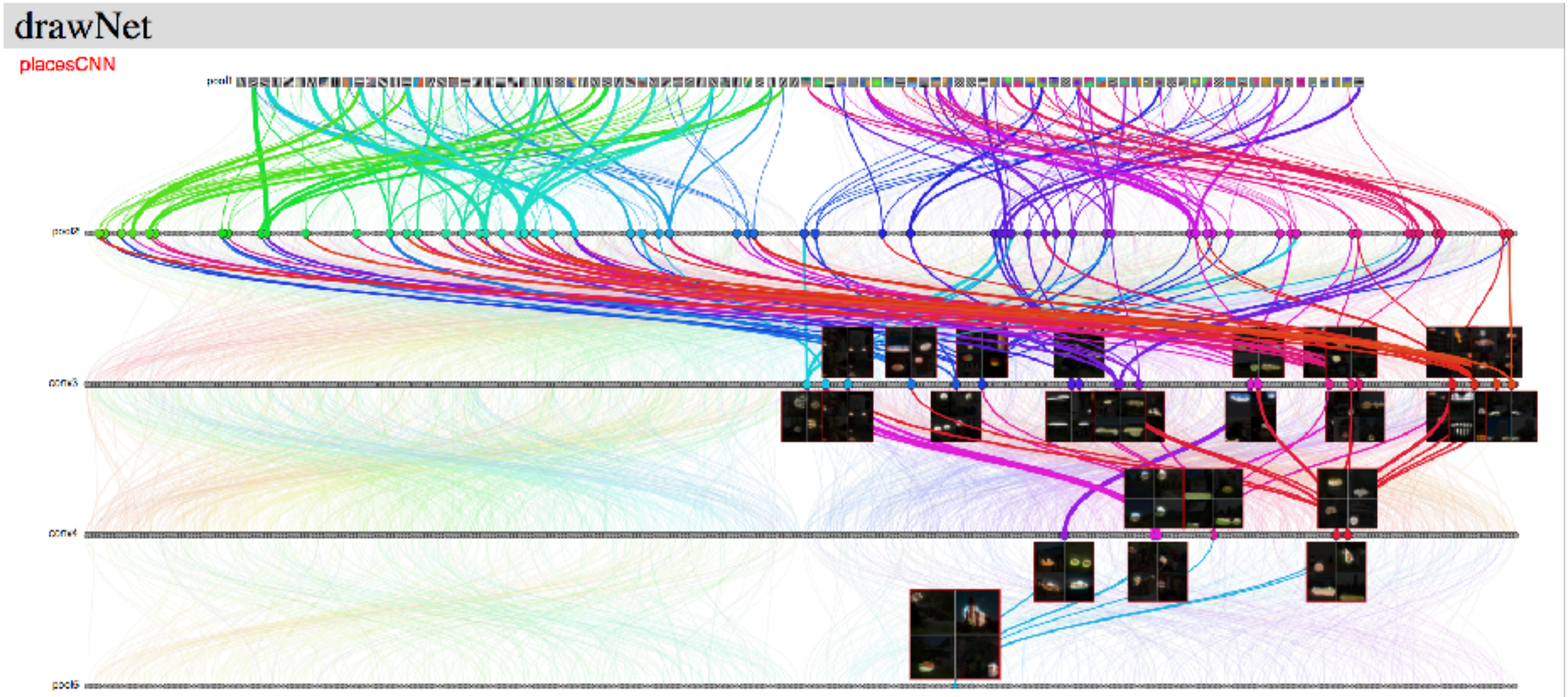
- Insight in terms of what is the process in which the prediction is made



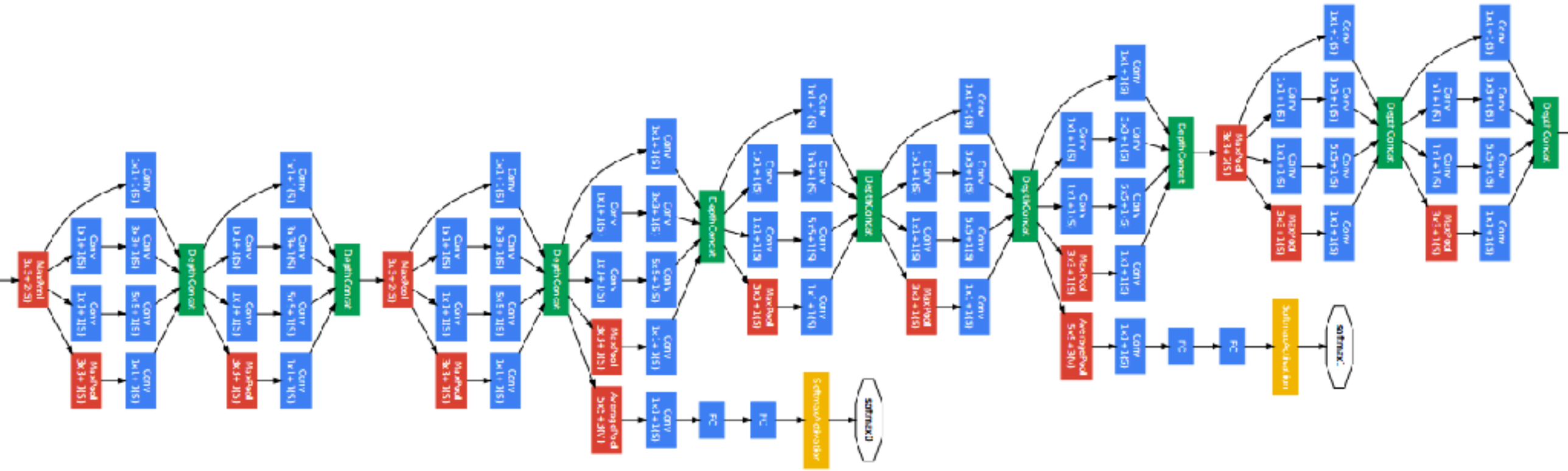
- Insight in terms of what is the process in which the prediction is made



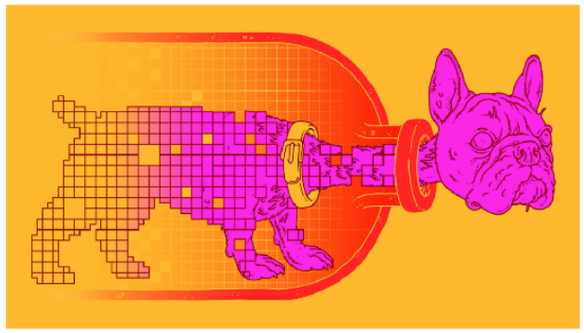
- Insight in terms of what is the process in which the prediction is made



- Architecture - what is the optimal architecture for a given task?



- GoogleLeNet 2014



What is learning (formally)?
The information Bottleneck method ():

data

X

true labels

Y

representation of X

T

estimated labels

$\hat{Y}$

cat

dog

The information Bottleneck method ():

data

X

true labels

Y

representation of X

T

estimated labels

$\hat{Y}$

cat

dog

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

Tishby et al. (1999)]

The optimal learning problem can be cast as the construction of an optimal encoder p(tlx) of that relevant information via an efficient representation - a minimal sufficient statistic of X with respect to Y - if such can be found. A minimal sufficient statistic can enable the decoding p(ylt) of the relevant information with the smallest number of binary questions (on average); i.e., an optimal code.

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

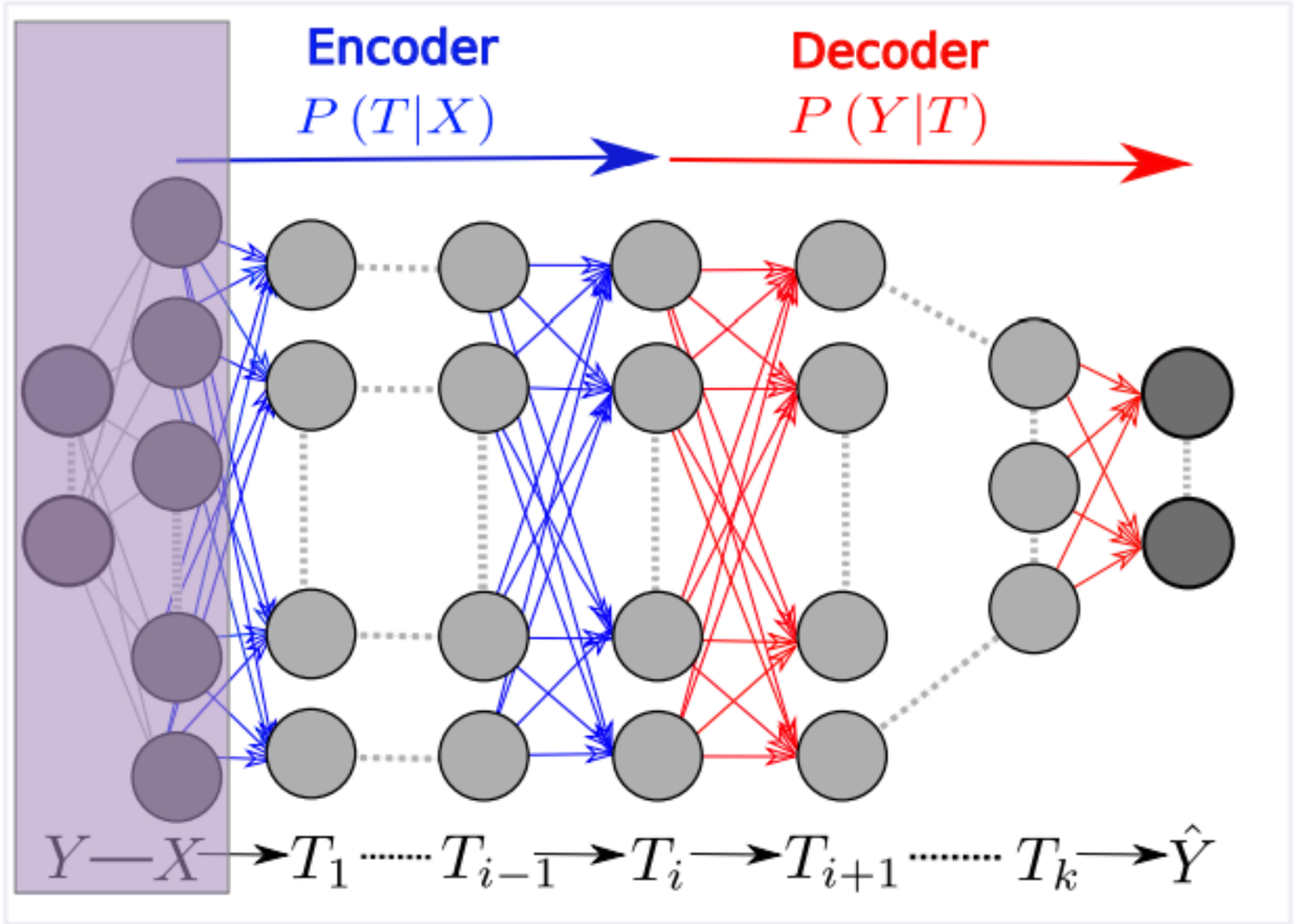
$$I(X;Y) = D_{KL}[p(x,y)||p(x)p(y)] = \sum_{x\in X, y\in Y} p(x,y)\log\left(\frac{p(x,y)}{p(x)\,p(y)}\right)$$

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

There is an implicit solution given by (P(X,Y) has to known):

$$\begin{cases} p(t|x) = \frac{p(t)}{Z(x;\beta)}\exp\left(-\beta D_{KL}\left[p(y|x)\parallel p(y|t)\right]\right) \\ p(t) = \sum_x p(t|x)\,p(x) \\ p(y|t) = \sum_x p(y|x)\,p(x|t)\ , \end{cases}$$

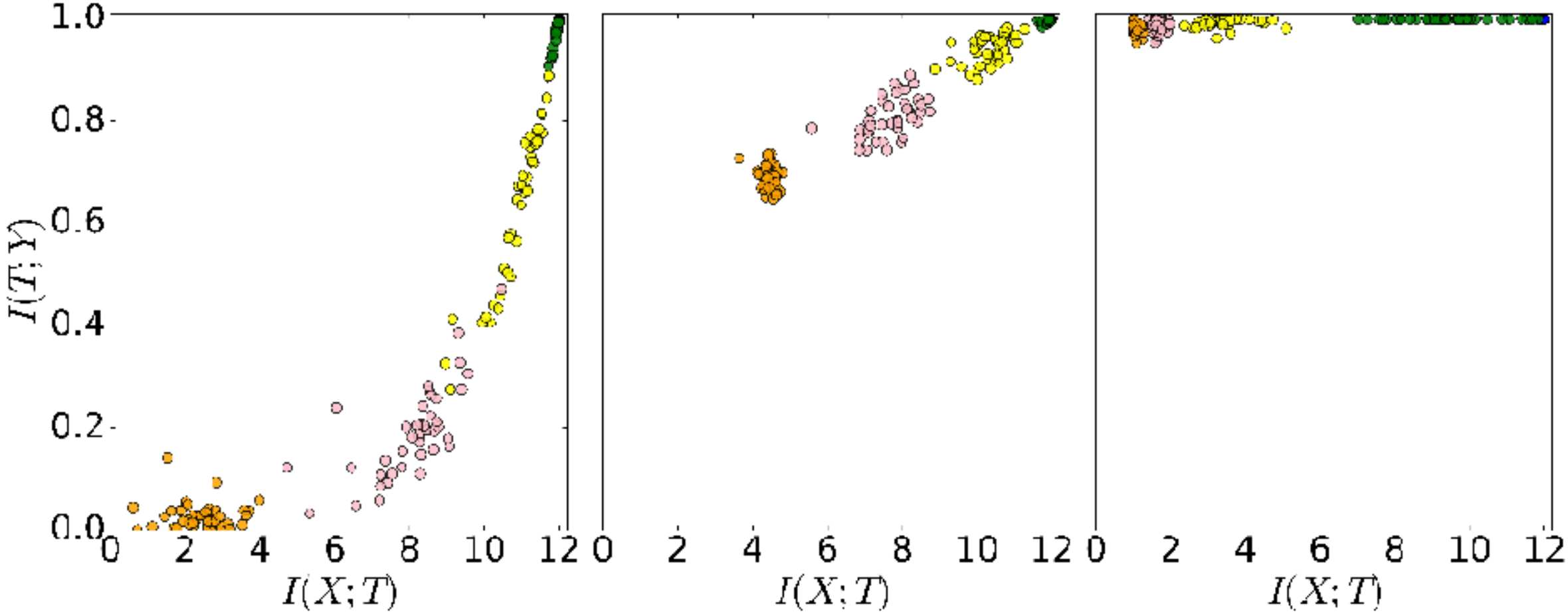
Tishby et al. (1999)]



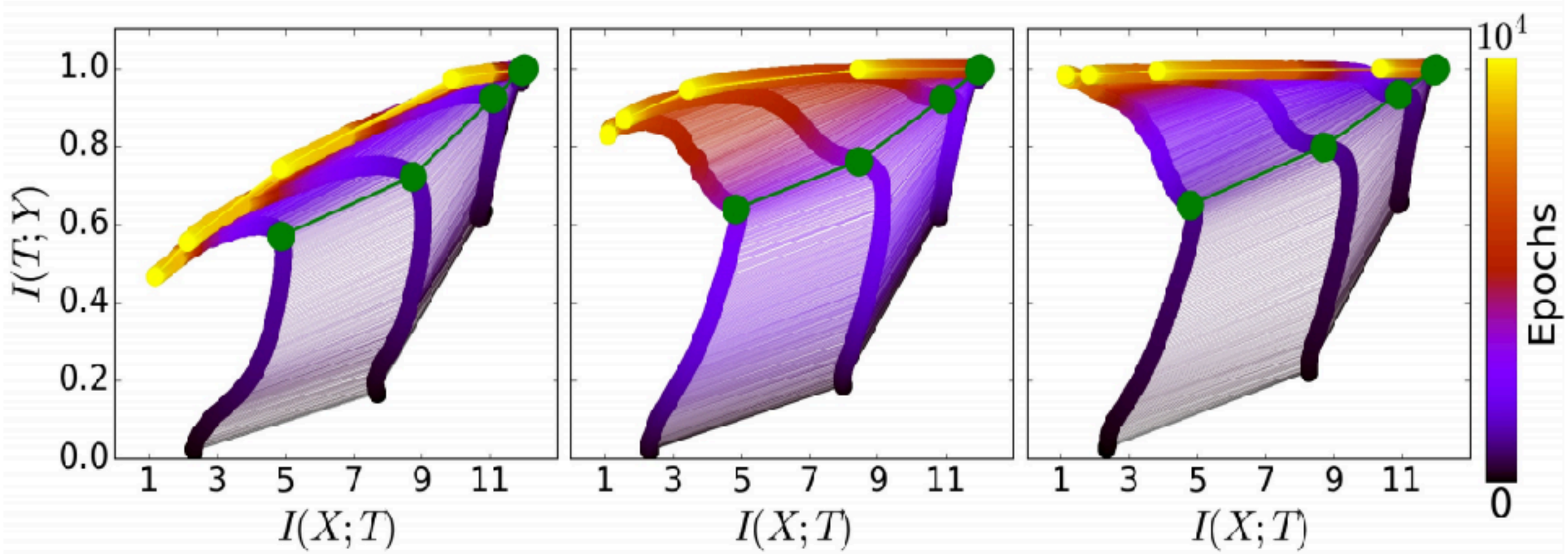
Setup:

- activation function: hyperbolic function; last layer sigmoidal
- SGD, loss function: cross-entropy
- at most 12-10-7-5-4-3-2 layers
- inputs - 12 binary
- output - 1 binary label
- a given function maps the uniformly distributed input to some label and defines the joint probability distribution.

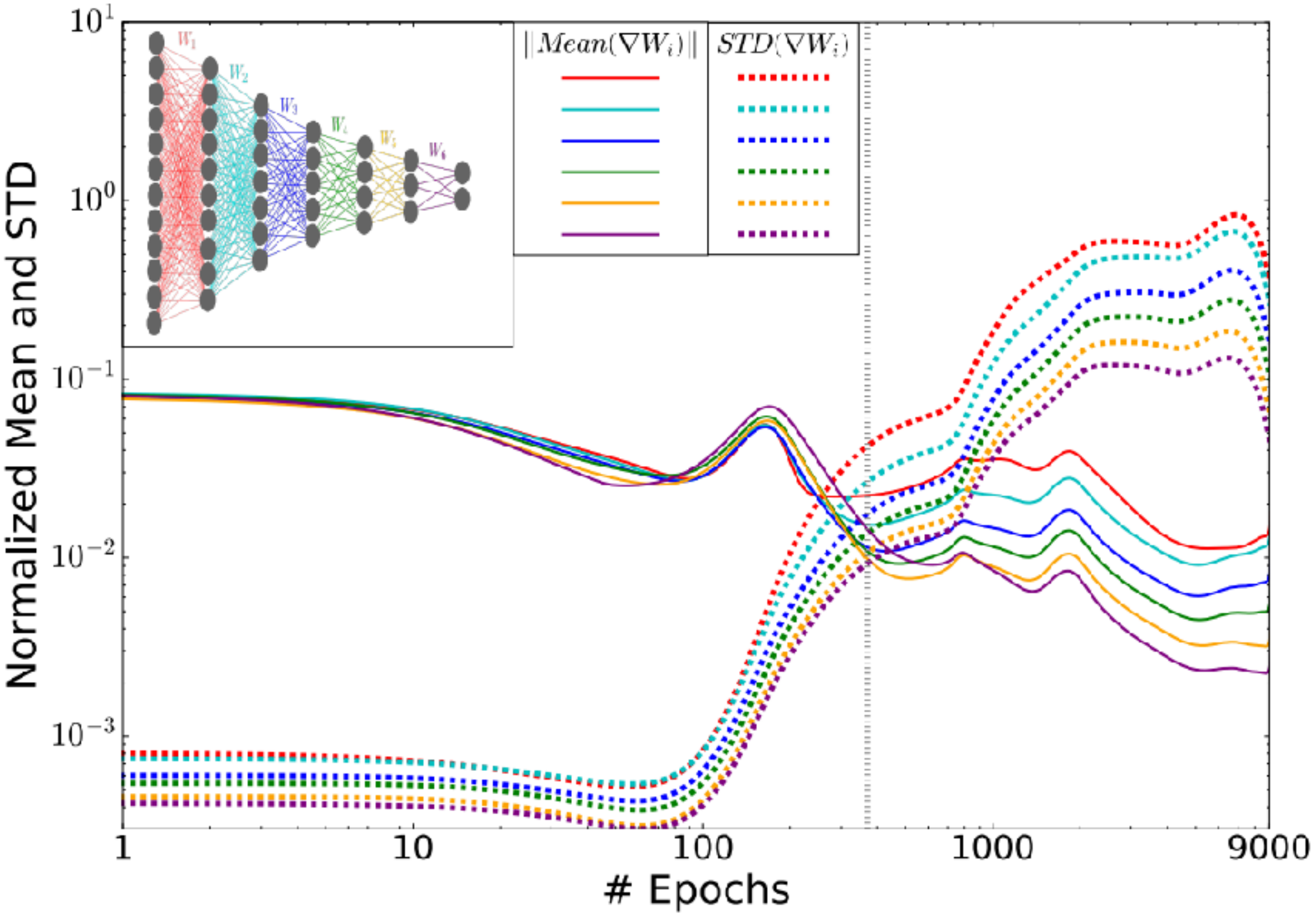
Tishby et al. (2016, 2017)]



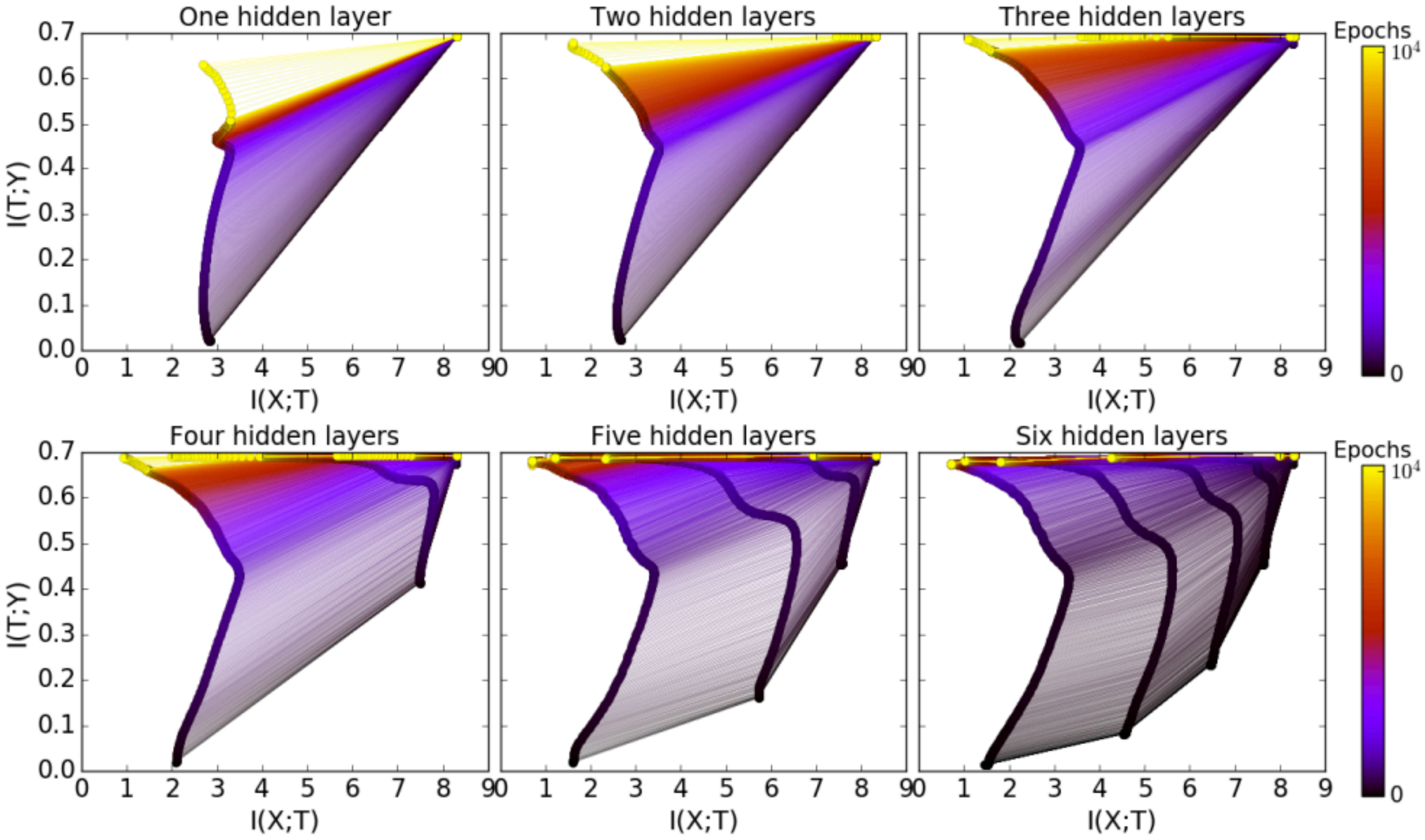
Tishby et al. (2016, 2017)]



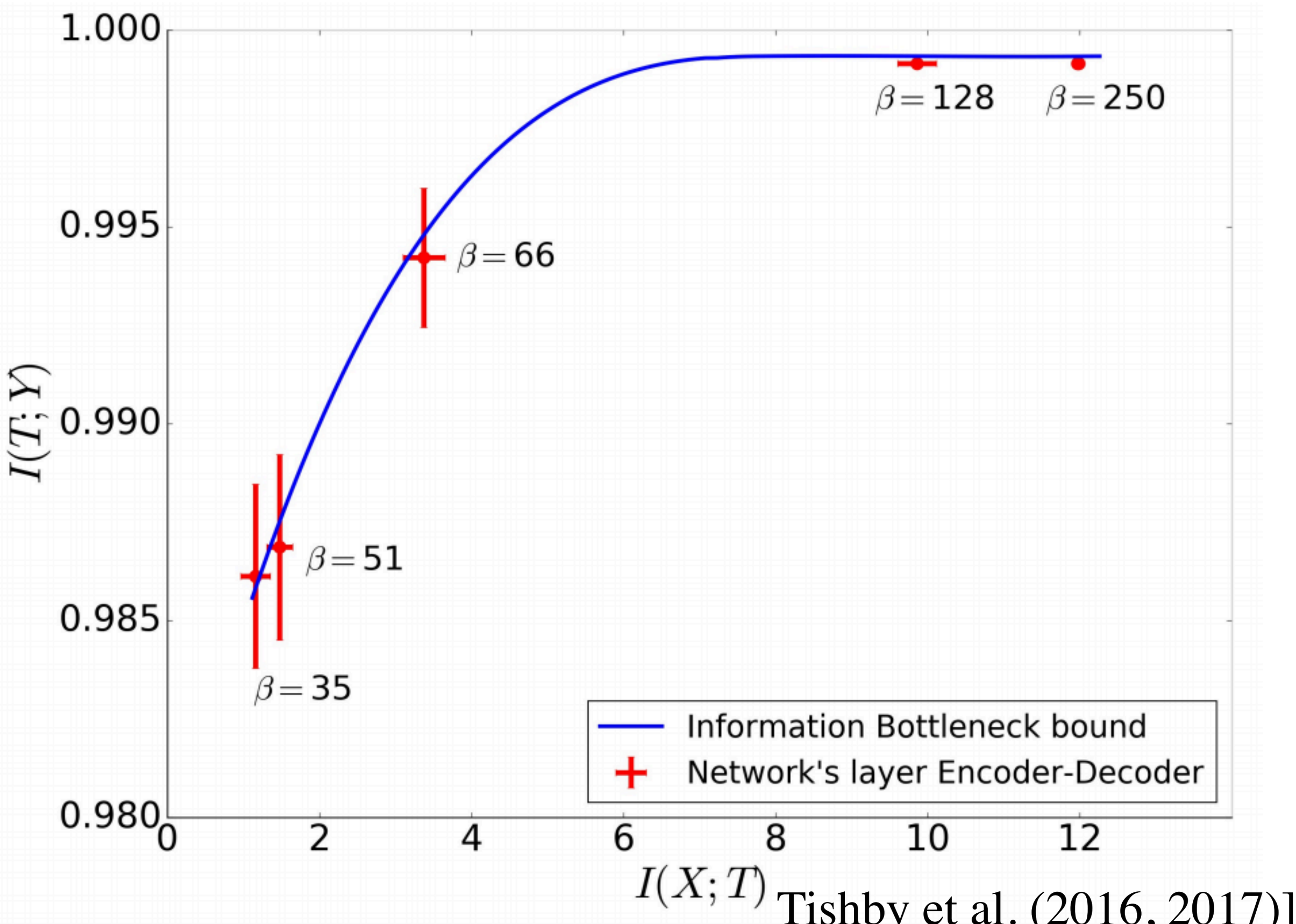
Tishby et al. (2016, 2017)]



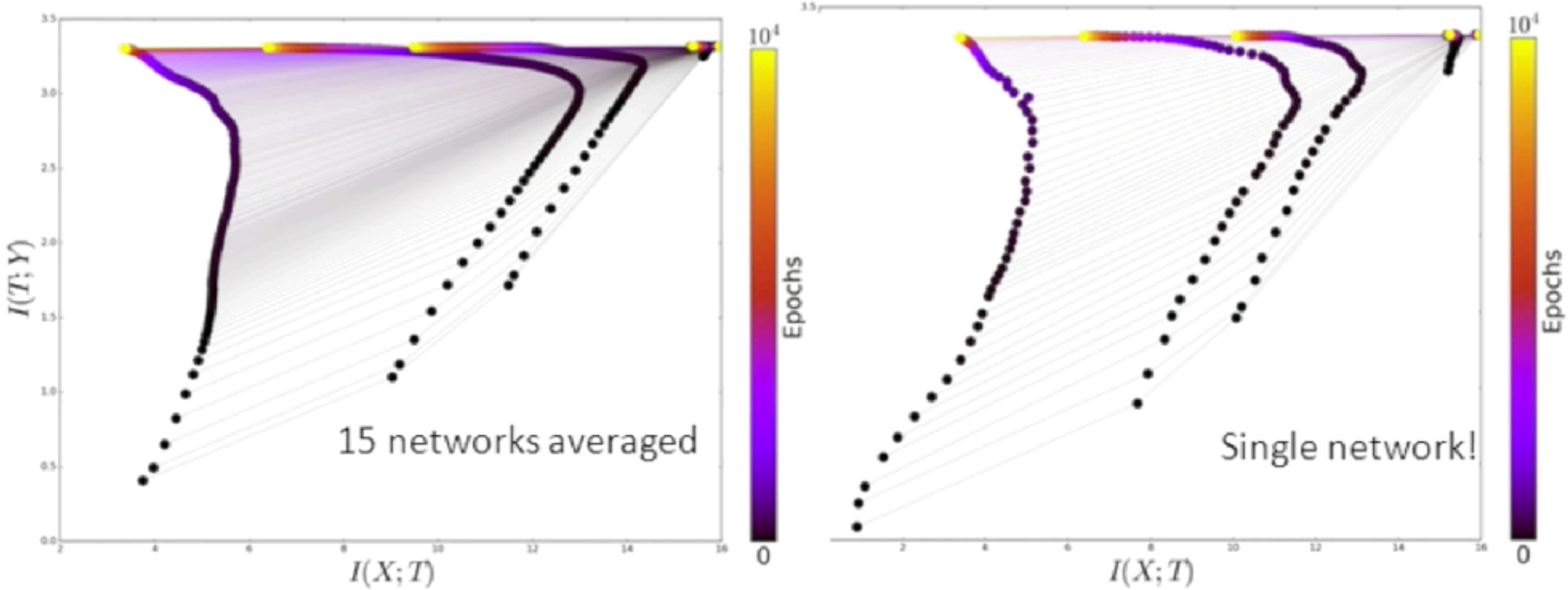
Tishby et al. (2016, 2017)]



Tishby et al. (2016, 2017)]



Tishby et al. (2016, 2017)]



MNIST handwriting digit recognition with a CNN architecture

Tishby et al. (2016, 2017)]

- SGD has to phases: Empirical Error Minimization (EEM) phase and a Representation Compression (RC) phase

- EEM - mostly fitting (short, dominated by drift), RM mostly reducing I(T,X) (long, dominated by diffusion)

- converged layers lie close to the IB theoretical bound, and P(T|X), satisfy the IB self-consistent optimality conditions

- hidden layers reduce the EEM (stochastic relaxation time)

- compressing the representation allows for generalisation

- Claim: generalizes to other architectures / types of deep models

- Claim: SGD in the diffusion phase is an overkill and faster methods can be used

Deep Learning - Emerging connections to physics

- Renormalization Group
- Spin Glasses
- Tensor networks
- Holography
- Random matrix Theory
- Field Theory and Spontaneous Symmetry breaking

Thank you for attention.

Some reading:

- Yann LeCun Yoshua Bengio & Geoffrey Hinton, Deep Learning, Nature Review;