Information Bottleneck

Opening the Black Box of Deep Neural Networks

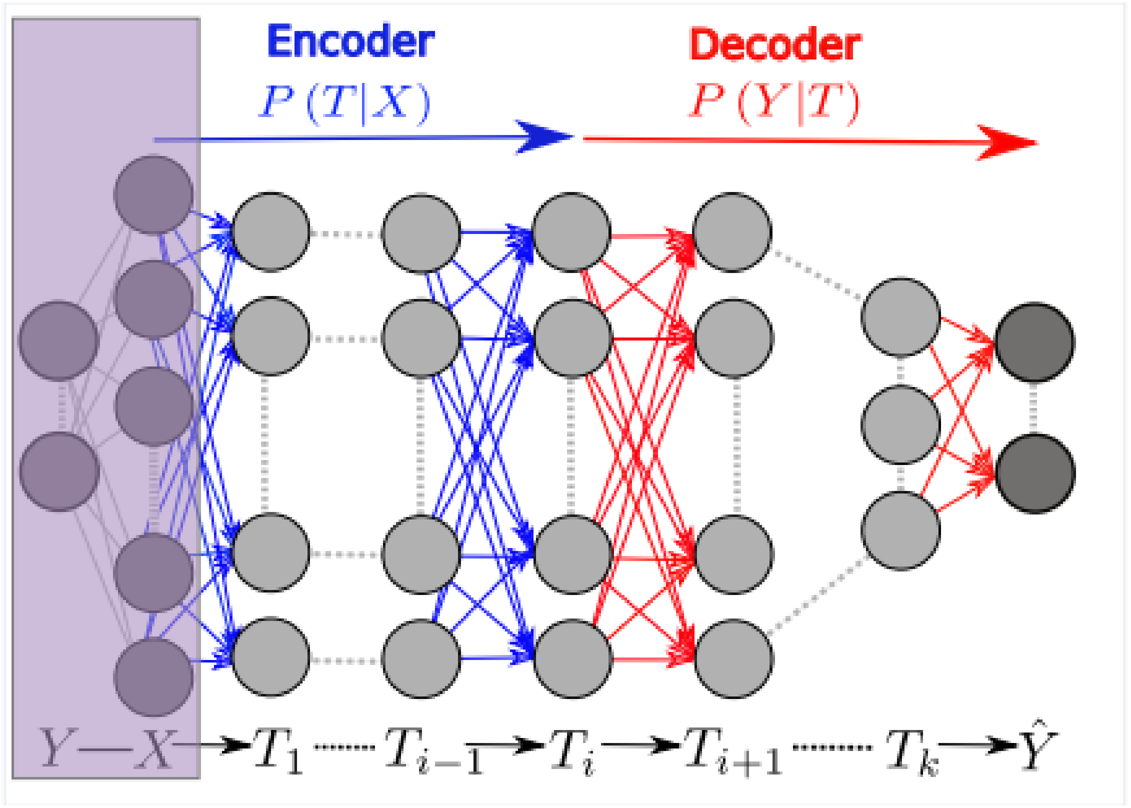
Penglong Zhai

2018/11/17

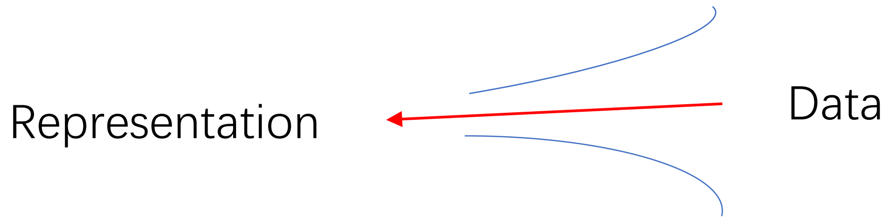
Intuition

- Deep learning algorithms have made remarkable progress on numerous machine learning tasks and dramatically improved the state-of-the-art in many practical area.
- Despite their great success, there is still no comprehensive understanding of the optimization process or the internal organization of DNNs .
- They are often criticized for being used as mysterious ‘black boxes’

Introduction


Encoder
$P(T|X)$
Decoder
$P(Y|T)$
$Y — X \rightarrow T_1 \cdots T_{i-1} \rightarrow T_i \rightarrow T_{i+1} \cdots T_k \rightarrow \hat{Y}$

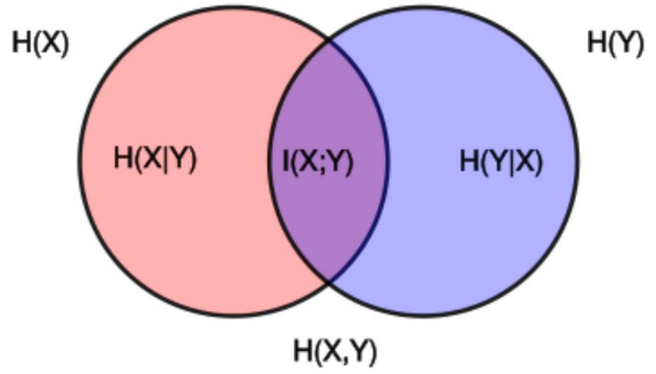
- The DNN layers form a Markov chain of successive internal representations of the input layer *X*.
- representation of the input, T , is defined through an encoder, $P\,(T|X)$, and the prediction $\hat{Y}$, through a decoder $P\,(\hat{Y}|T\,)$.
- We formalize this problem as that of finding a short code for X that preserves the maximum information about Y and eliminates the redundant information of X relevant to Y.
- we squeeze the information that X provides about Y through a 'bottleneck' formed by a limited set of codewords $\hat{X}$.



Information Theory Definitions

- Entropy is the uncertainty of a random variable.
- Joint entropy is the entropy of a pair of r.v.s.
- Mutual Information is the KL-divergence between the joint distribution and the product distribution which can be interpreted as reduction in uncertainty of due to knowledge of another r.v.

$$
\begin{aligned}
I(X;Y) &= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \\
&= H(X) - H(X|Y).
\end{aligned}
$$



Optimal Representation: Minimal Sufficient Statistics

- Minimal sufficient statistics, $T(X)$, are the simplest sufficient statistics to inference Y and induce the coarsest sufficient partition on X.

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

- Exact minimal sufficient statistics only exist for very special distributions.
- Replace it with approximate minimal sufficient statistics, or the optimal tradeoff between compression of X and prediction of Y .

A possible solution: Information Bottleneck

- From 'rate distortion theory': how well we can represent a r.v. X using a compressed representation $\hat{X}$
- we squeeze the information that X provides about Y through a 'bottleneck' formed by a limited set of codewords $\hat{X}$.
- There is tradeoff between compressing the representation and preserving meaningful information.
- The idea can be formulated as :

$$\boxed{\mathcal{L}[p(\tilde{x}|x)] = I(\tilde{X};X) - \beta I(\tilde{X};Y)}$$

[1].N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, 1999, pp. 368–377.

Minimization method

- There are two parameters we need to specify in the equations.
- The distribution $\mathrm{p}(\hat{x}|x)$ that minimizes $\mathrm{I}(X;\hat{X})$ can be calculated using Lagrange multipliers by minimizing :

$$
\begin{aligned}
\mathcal{F}[p(\tilde{x}|x)] &= I(X;\tilde{X}) - \beta I(\tilde{X},Y) - \sum_{x,\tilde{x}} \lambda(x) p(\tilde{x}|x) \\
&= \sum_{x,\tilde{x}} p(x,\tilde{x}) \log \frac{p(\tilde{x}|x)}{p(\tilde{x})} \\
&\quad - \beta \sum_{\tilde{x},y} p(\tilde{x},y) \log \frac{p(\tilde{x},y)}{p(\tilde{x})p(y)} + \sum_{x,\tilde{x}} \lambda(x) p(\tilde{x}|x)
\end{aligned}
$$

IB self-consistent equations

- Computing the partial of $\mathcal{F}$ w.r.t. $p(\hat{x}|x)$ yields

$$\frac{\partial \mathcal{F}}{\partial p(\tilde{x}|x)} = p(x)\left[\log \frac{p(\tilde{x}|x)}{p(\tilde{x})} - \beta \sum_y p(y|x) \log \frac{p(y|\tilde{x})}{p(y)} - \frac{\lambda(x)}{p(y)}\right]$$

- Setting this equal to 0 and solving for $p(\hat{x}|x)$ yields

$$\begin{aligned} p(\hat{x}|x) &= \frac{p(\hat{x})}{Z(x;\beta)} \exp\left(-\beta D\left[p(y|x) \| p(y|\hat{x})\right]\right) \\ p(y|\hat{x}) &= \sum_x p(y|x)\, p(x|\hat{x}) \\ p(\hat{x}) &= \sum_x p(x)\, p(\hat{x}|x) \end{aligned}$$

$\xrightarrow{\text{algorithm}}$

```
while IB ≤ $\epsilon$:
$p_t(\tilde{x}|x) = p_t(\tilde{x})/Z_t(x,\beta)\exp(-\beta d(x,\tilde{x}))$;
$p_{t+1}(\tilde{x}) = \sum_x p(x) p_t(\tilde{x}|x)$;
$p_{t+1}(y|\tilde{x}) = \sum_y p(y|x) p_t(x|\tilde{x})$;
$t = t + 1$
```

Information Path In The Networks

- Relevant information captured by the network can be quantified by:

$$\boxed{I(Y;\hat{Y})/I(X;Y)}$$

- After training, the network receives an input X, and successively processes it through the layers, which form a Markov chain to the predicted output Y(Data Processing Inequality).

$$I(Y;X) \geq I(Y;\mathbf{h}_j) \geq I(Y;\mathbf{h}_i) \geq I\left(Y;\hat{Y}\right)$$

$$I(X;h_j) \geq I(X;h_i) \geq I(X;\hat{Y})$$

- Representation of Every layer can be quantified in the above formulation and can be used to evaluate the optimality of each hidden layer.
- Our two order parameters, $I(h_i;\ X)$ and $I(h_i;\ Y\ )$,allow us to visualize and compare different network architectures in terms of their efficiency in preserving the relevant information in $P(X;\ Y\ )$.

Dynamics of the training in networks

- By visualizing the paths of different networks in the *information path* we explore the following fundamental issues:

1. The SGD layer dynamics in the *Information plane*.
2. The effect of the training sample size on the layers.
3. What is the benefit of the hidden layers?
4. What is the final location of the hidden layers?
5. Do the hidden layers form optimal IB representations?

[3]. Naftali Tishby et.al **Opening the black box of Deep Neural Networks via Information**

Experiment Setting

Dataset

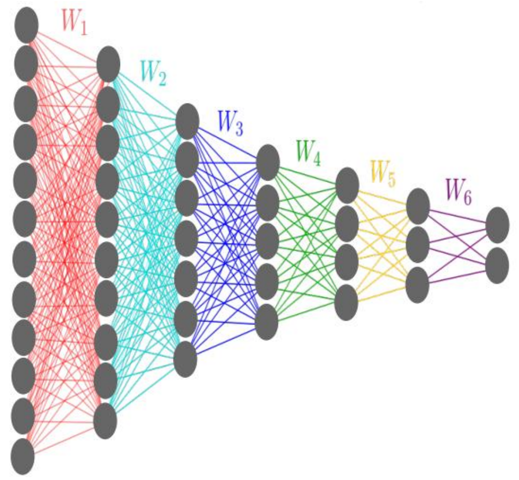
✓12D binary dataset.

✓4096 data points.

✓2 classes.

model

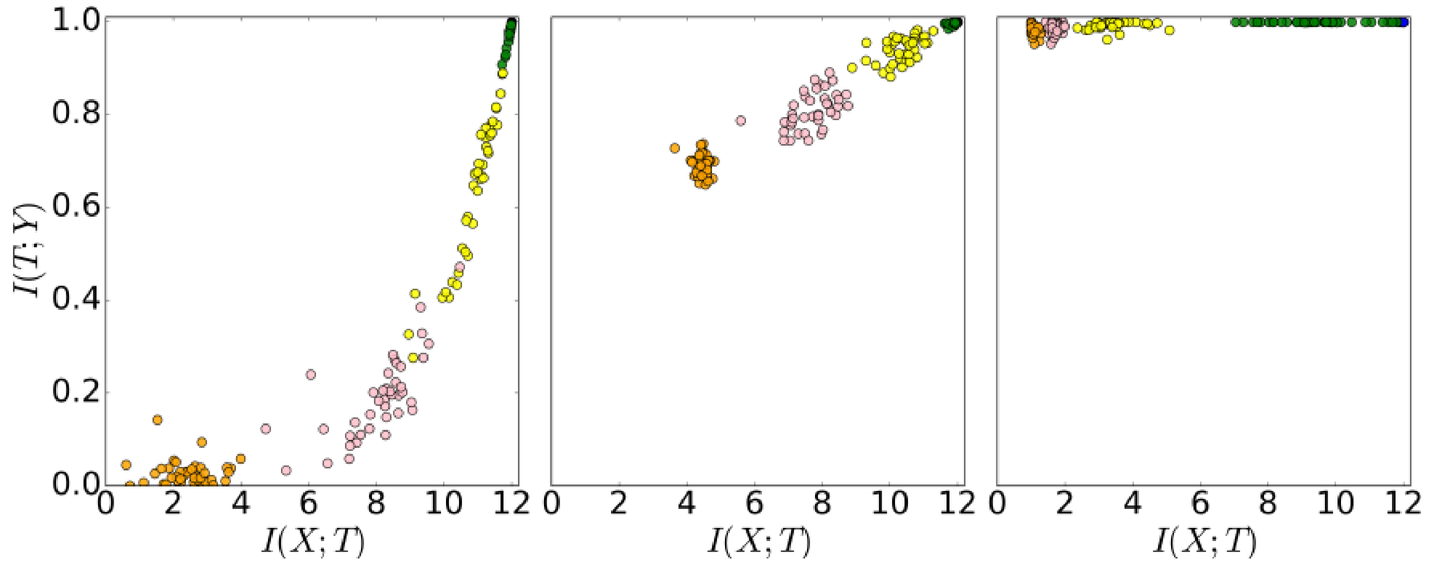
✓Architecture :



✓7 layers

✓Tanh or Sigmoid activation.

Training dynamics of the layers.



Snapshots of layers (different colors) of 50 randomized networks during the SGD optimization process in the *information plane* (in bits): **left** - with the initial weights; **center** - at 400 epochs; **right** - after 9000 epochs.

- During the SGD optimization the layers first increase *IY* , and later significantly decrease *IX*, thus compressing the representation.

The effect of the training sample size on the layers

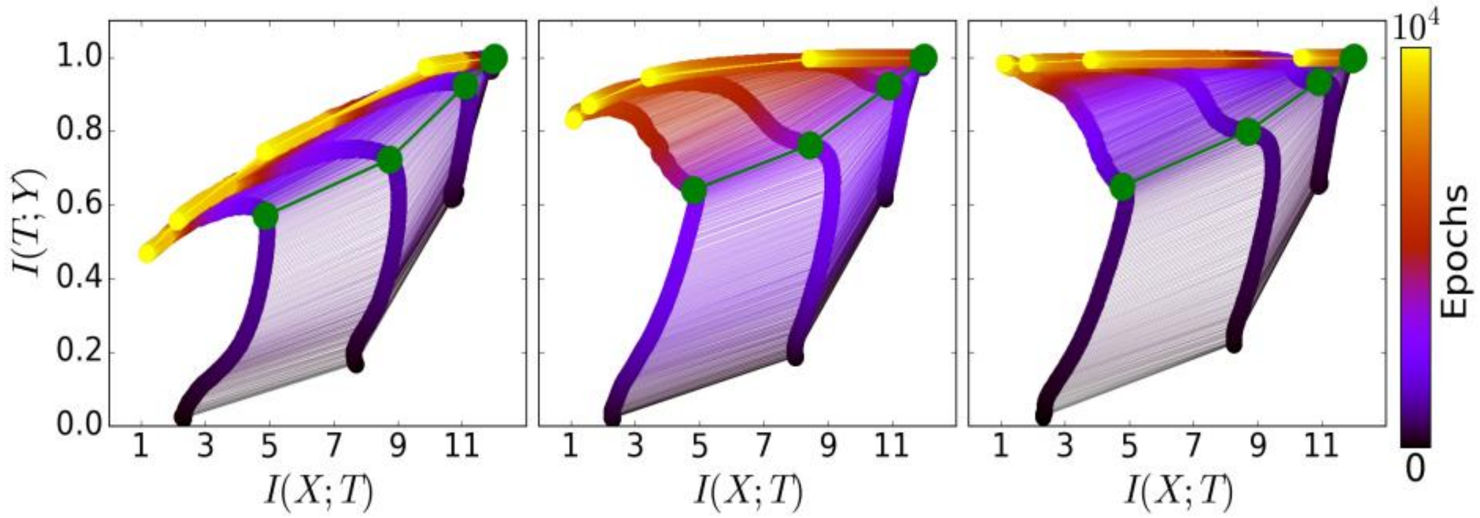
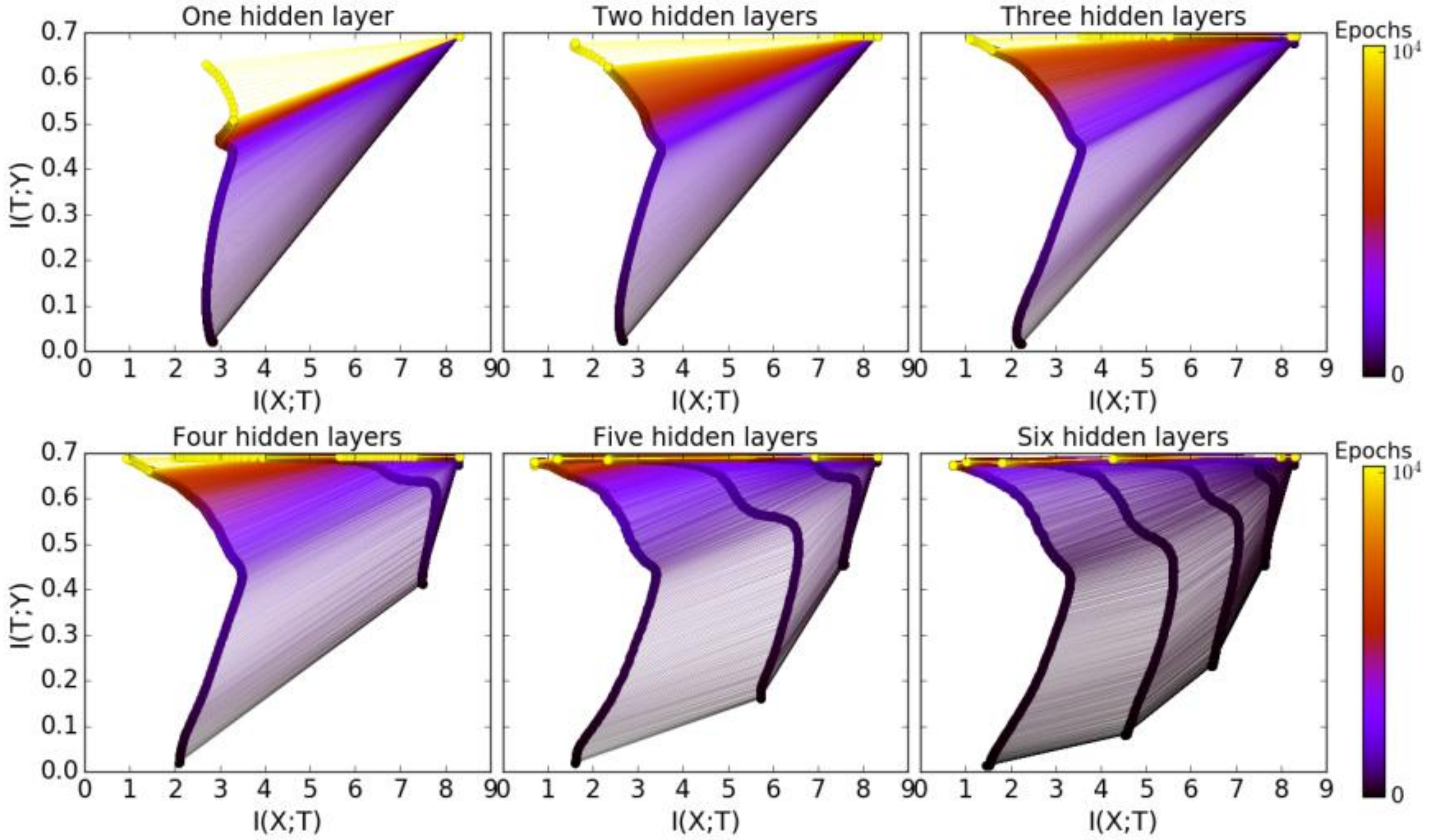


Figure 3: The evolution of the layers with the training epochs in the information plane, for different training samples. On the left - 5% of the data, middle - 45% of the data, and right - 85% of the data. The colors indicate the number of training epochs with Stochastic Gradient Descent from 0 to 10000. The network architecture was fully connected layers, with widths: input=12-10-8-6-4-2-1=output. The examples were generated by the spherical symmetric rule described in the text. The green paths correspond to the SGD drift-diffusion phase transition - grey line on Figure 4

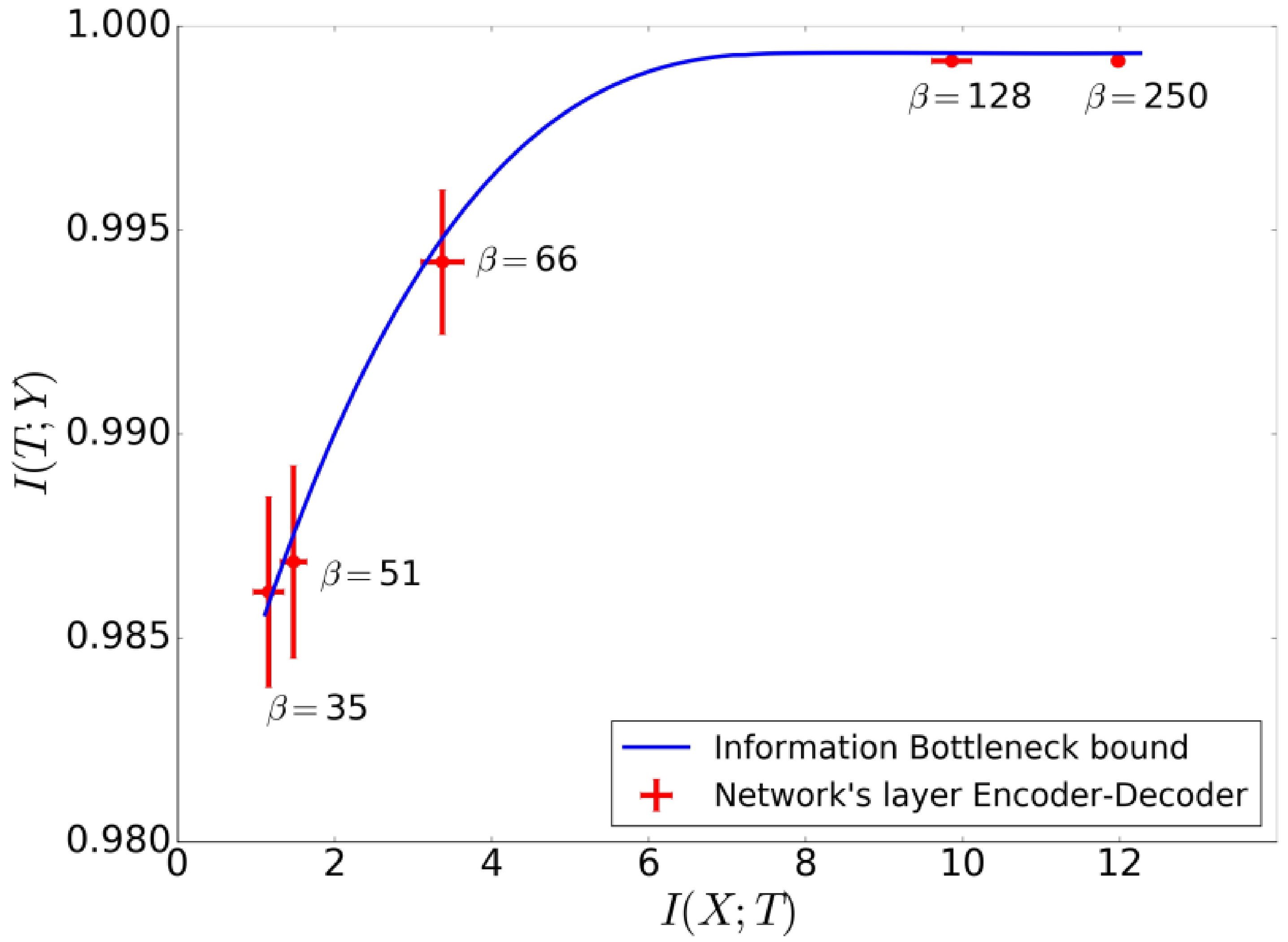
- During the fast ERM - phase, which takes a few hundred epochs, the layers increase the information on the labels (increase I_Y) while preserving the DPI order (lower layers have higher information).
- In the second and much longer compression phase the layers' information on the input, Ix, decreases and the layers lose irrelevant information until convergence .
- The compression phase significantly reduced the layers' label information in the small sample case, but with large samples the label information mostly increased.

Computational benefit of the hidden layers



- ✓ *Adding hidden layers dramatically reduces the number of training epochs for good generalization.*
- ✓ *The compression phase of each layer is shorter when it starts from a previous compressed layer .*
- ✓ *The compression is faster for the deeper (narrower and closer to the output) layers .*

Convergence to the layers to the IB bound



$$\beta_i^\star = \arg\min_{\beta} \mathbb{E}_x D_{KL}\left[p_i\left(t|x\right)||p_\beta^{IB}\left(t|x\right)\right]$$

- the DNN layers' encoder-decoder distributions satisfy the IB self-consistent equations within our numerical precision, with decreasing β as we move to deeper layers.

Conclusion

- We develop a new principle to evaluate the representations of our networks.
- most of the training epochs in standard DL are spent on compression of the input to efficient representation and not on fitting the training labels.
- The representation compression phase begins when the training errors becomes small.
- 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.
- The training time is dramatically reduced when adding more hidden layers. Thus the main advantage of the hidden layers is computational.

Limitation of the method

- In the experiment the dataset and the class of the data is pretty small.
- They just compute a pretty small net and dataset with a special activation.
- In recent study, we can’ t get the same conclusion if we replace the activation with others , for example , relu.
- Computation of MI and IB of bigger networks and datasets is pretty hard.

Thanks!!