Interpretability of Deep Learning

(draft)

Bingbing Hu

January 3, 2019

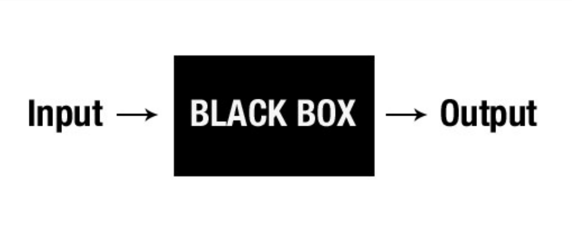
SIST, ShanghaiTech

Outline

1. Motivation
2. Interpretability
    - Model Transparency
    - Model Functionality
3. Information Bottleneck

Motivation

Motivation



Motivation

- Deep Learning (DL) is hot but quite a blackbox
- Interpretability is demanding
  self-driving car, healthcare, criminal justice ...
- Why shall we trust it

We should understand & be able to reason about model output.

Notion of Interpretability

- Model transparency
  - simulatability
  - decomposability
  - algorithm transparency
- Model functionality
  - textual description
  - visualization
  - local explanation

Interpretability

Model Transparency

[1]Visualizing the response of individual units in unsupervised deep belief networks [EBCV09].

- analyze units in any layer
- [2]extended by [ZF14] to supervised CNN for higher-layer analysis
- visualize to guide modifications

[1]Erhan et al., University of Montreal, 2009
[2]Zeiler et al., ECCV, 2014

Model Transparency

[1] Investigate information in different layer [MV15].

- investigate image representation at different CNN layers
- reval deeper layers learn more abstract representation of a image

[1] Mahendran et al., Proceedings of CVPR, 2015

Model Transparency

[1]Generate model-preferred inputs [SVZ13].

- generate images by maxmizing output score
- qualitatively demonstrate the features most representation each class

[1]Simonyan et al., arXiv preprint, 2013

Model Transparency

[1]Generate perferred images to particular neurons in CNN [NYC16].

- use Deep Generator Network to generate images
- producing very realistic images
- try to understand what the network has learned

[1]Nguyen et al., arXiv preprint, 2016

Model Transparency

[1]How a model's predictions would differ if a data point were altered, or not seen during training [KL17]?

- use statistical influence functions to approximate the disturb of data points without retrain the model
- assess the importance of particular training point
- generate adversarial examples to attack the trained model

[1]Koh et al., arXiv preprint, 2017

Model Transparency

[1]Analyze deep networks using information theory [SZT17].

- calculate how info. is preserved on each layer's in/out-puts
- learn how SGD optimizes the network
- the depth of the network is consistent with IB optimality

[1]Shwartz-Ziv et al., arXiv preprint, 2017

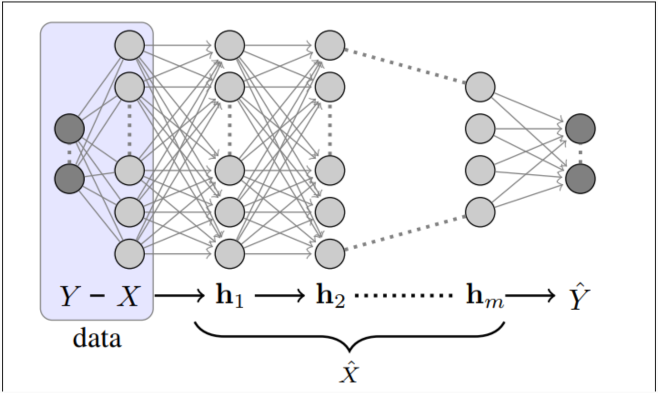
Model Functionality

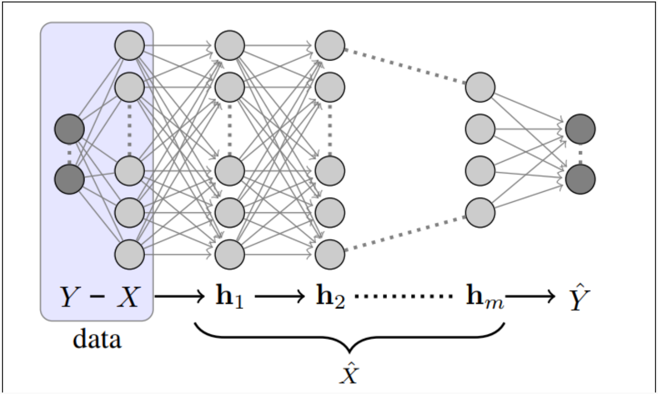
Model functionality can be explained by post-hoc interpretations of what the model has done.

- Using t-SNE to visualize model output
- ......

Information Bottleneck

Deep Learning and the Information Bottleneck Principle (Naftali TIshby & Noga Zaslavsky)





X : input, low-level representation of data

Y : desired output, lower dimension

The most entropy of X is not very informative about Y.

IB Principle

Given input $X \in \mathcal{X}$, output $Y \in \mathcal{Y}$, we want to compress X while preserve the information about Y.

IB Principle

Given input $X \in \mathcal{X}$, output $Y \in \mathcal{Y}$, we want to compress X while preserve the information about Y.

An optimial representation will compress X by dismissing the irrelevant parts which give no info. about Y.

IB Principle

Given input $X \in \mathcal{X}$, output $Y \in \mathcal{Y}$, we want to compress X while preserve the information about Y.

An optimial representation will compress X by dismissing the irrelevant parts which give no info. about Y.

Summary

Namely, find the relevant parts $\hat{X}$ inside X w.r.t. Y to minimize

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

IB vs. Rate-Distortion

Rate distortion: $X \in \mathcal{X}$, find a representation $\hat{X}$ of X
Goal: minimize the "distance between" X and $\hat{X}$

IB vs. Rate-Distortion

Rate distortion: $X \in \mathcal{X}$, find a representation $\hat{X}$ of X

Goal: minimize the "distance between" X and $\hat{X}$

Need a distance measure: $d_{IB}(x, \hat{x}) = D_{KL}[p(y|x) \| p(y|\hat{x})]$

$\implies D_{IB} = \mathbb{E}[d_{IB}(x, \hat{x})] = I(X; Y|\hat{X})$

IB vs. Rate-Distortion

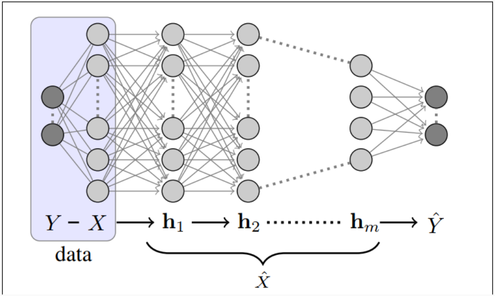
Rate distortion: $X \in \mathcal{X}$, find a representation $\hat{X}$ of X
Goal: minimize the "distance between" X and $\hat{X}$
Need a distance measure: $d_{IB}(x, \hat{x}) = D_{KL}[p(y|x) \| p(y|\hat{x})]$
$\implies D_{IB} = \mathbb{E}[d_{IB}(x, \hat{x})] = I(X; Y|\hat{X})$

To minimize

$$\tilde{\mathcal{L}}[p(\hat{x}|x)] = I\left(X; \hat{X}\right) + \beta I\left(X; Y|\hat{X}\right)$$

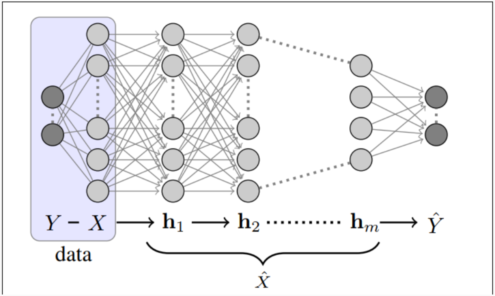
Note: $I(X; Y|\hat{X})$ can be viewed as the relevant information *not* captured by $\hat{X}$.

Applied to DNN



$$I(Y;X) \geq I(Y;\boldsymbol{h}_{i-1}) \geq I(Y;\boldsymbol{h}_i) \geq I(Y;\hat{Y})$$

Applied to DNN



$$I(Y;X) \geq I(Y;\boldsymbol{h}_{i-1}) \geq I(Y;\boldsymbol{h}_i) \geq I(Y;\hat{Y})$$

At each layer,

- maximize $I(Y;\boldsymbol{h}_i)$: view $\boldsymbol{h}_i$ as $\hat{X}$
- minimize $I(\boldsymbol{h}_{i-1};\boldsymbol{h}_i)$: view $\boldsymbol{h}_{i-1}$ as X

Applied to DNN

From $I(X;\hat{X}) + \beta I(X;Y|\hat{X})$, by defining $\boldsymbol{h}_0 = X$ and $\boldsymbol{h}_{m+1} = \hat{Y}$ we get

$$I(\boldsymbol{h}_{i-1};\boldsymbol{h}_i) + \beta I(Y;\boldsymbol{h}_{i-1}|\boldsymbol{h}_i),$$

which gives a optimal rule for training.

Applied to DNN

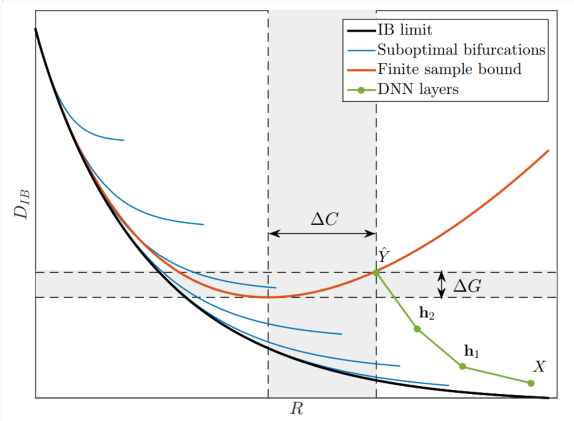
However, the joint distribution $p(X, Y)$ is unknown in general!

- representational complexity $K = |\hat{\mathcal{X}}|$
- finite sample distribution $\hat{p}(x, y)$
- empirical mutual info. $\hat{I}(X, Y)$

[SST10] gives

$$I\left(\hat{X}; Y\right) \leq \hat{I}\left(\hat{X}; Y\right) + O\left(\frac{K|\mathcal{Y}|}{\sqrt{n}}\right)$$
$$I\left(\hat{X}; X\right) \leq \hat{I}\left(\hat{X}; X\right) + O\left(\frac{K}{\sqrt{n}}\right)$$

Applied to DNN



Applied to DNN

Some insight:

- input layer has a bad generalization since it's complexity
- hidden layer compressed the input for a better generalization

References i

[EBCV09] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent, *Visualizing higher-layer features of a deep network*, University of Montreal **1341** (2009), no. 3, 1.

[KL17] Pang Wei Koh and Percy Liang, *Understanding black-box predictions via influence functions*, arXiv preprint arXiv:1703.04730 (2017).

[MV15] Aravindh Mahendran and Andrea Vedaldi, *Understanding deep image representations by inverting them*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 5188–5196.

[NYC16] Anh Nguyen, Jason Yosinski, and Jeff Clune, *Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks*, arXiv preprint arXiv:1602.03616 (2016).

[SST10] Ohad Shamir, Sivan Sabato, and Naftali Tishby, *Learning and generalization with the information bottleneck*, Theoretical Computer Science **411** (2010), no. 29-30, 2696–2711.

[SVZ13] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, *Deep inside convolutional networks: Visualising image classification models and saliency maps*, arXiv preprint arXiv:1312.6034 (2013).

[SZT17] Ravid Shwartz-Ziv and Naftali Tishby, *Opening the black box of deep neural networks via information*, arXiv preprint arXiv:1703.00810 (2017).

[ZF14] Matthew D Zeiler and Rob Fergus, *Visualizing and understanding convolutional networks*, European conference on computer vision, Springer, 2014, pp. 818–833.

Thank you!