Neural Networks, Information Bottleneck and Unsupervised Learning

Fabio Massimo Zennaro[1]

Department of Informatics
University of Oslo

[1] fabiomz@ifi.uio.no

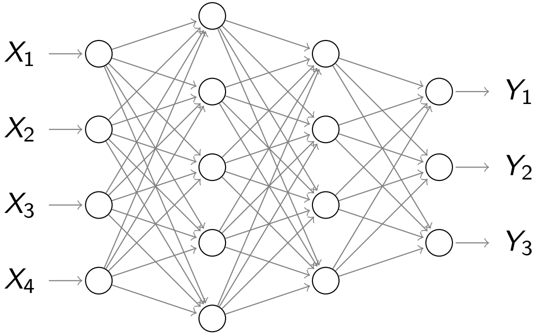
Content

1. *Neural networks (NN):* a brief intro on neural networks
2. *Understanding NNs:* questions on the dynamics of neural networks
3. *Information bottleneck (IB):* one framework to study neural networks
4. *Understanding unsupervised learning algorithms via IB:* a link to some of my work

1. Neural Networks

What is a neural network?

In general, a *model* (loosely inspired from biology) *for learning/fitting*.

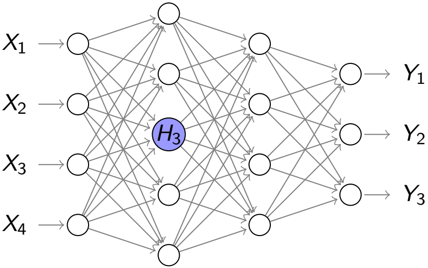


In particular, a *supervised feedforward NN* maps input X to output Y.

We can further characterized this answer in different way.

NN as a graphical model

From a *graphical point of view*, a neural network is a **layered weighted graphical model**.



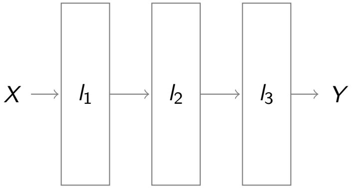
We can compute the activity of a node through a *linear combination* and an *element-wise non-linearity* f:

$$H_3 = g\left(\sum_{i=1}^{5} W_{i3}X_i + b_3\right)$$

This can be expressed more compactly in *matrix notation*.

NN as a composition of functions

From a *compositional* point of view, a neural network is a **composition of functions**.



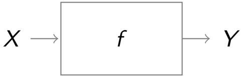
A network composes (or *stacks* in ML jargon) multiple layers:

$$Y = l_3 \circ l_2 \circ l_1(X) = l_3(l_2(l_1(X)))$$

This has been formalized in category-theoretical terms too [7].

NN as a function approximator

From a *functional* point of view, a neural network is a **function approximator** [5].



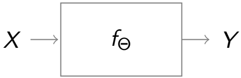
A network is simply a function:

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$
$$f : X \mapsto Y$$

This is often called the *black-box view*.

NN as a function fitter

From a *statistical* point of view, a neural network is a **function fitter**.



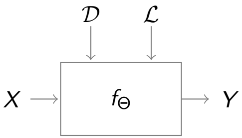
A network is now a parametrized function that approximates a function $f*$.

The parameters are *weights* and *biases*:

$$\Theta = \{W_l, b_l\}$$

NN as a learning model

From a *learning* point of view, a neural network is a **flexible trainable model**.



We learn a parametrized function f_Θ using the *data* $\mathcal{D}$ to optimized a *loss function* $\mathcal{L}$.

$$\min_\Theta \mathcal{L}\left(f_\Theta(X), Y\right)\Big|_{(X,Y)\in\mathcal{D}}$$

This optimization problem is defined in the *parameter space* (not in the *function space*).

Backpropagation

We learn by *gradient descent*:

$$\frac{\partial\, \mathcal{L}\left(f_{\Theta}(X), Y\right)|_{(X,Y)\in\mathcal{D}}}{\partial\Theta}$$

Weight updates are *backpropagated* through the layers via *chain rule*.

Notice that the *loss landscape* depends on the data $\mathcal{D}$.

Neural networks are instances of *differentiable programs*.

2. Understanding neural networks

A real-world instance of a neural network

Take as an example the historic *AlexNet* [9].

- *Number of parameters:* $|\Theta| \approx 60 \cdot 10^6$
- *Number of data points:* $|\mathcal{D}| \approx 1.2 \cdot 10^6$

In 2012, this network set a breakthrough performance in image classification.

See more recent architectures/dataset online[2]: in general, $|\Theta| > |\mathcal{D}|$.

[2] `https://paperswithcode.com/sota/image-classification-on-imagenet`

The magic of learning

Given $|\Theta| > |\mathcal{D}|$, it is not surprising that NNs learn.

It is surprising that NNs **generalize** (as opposite to *memorizing* a dataset).

Generalization is empirically verified by measuring performances on test data unseen at training.

Although phenomena like *adversarial examples* suggest that generalization may be brittle or counterintuitive [23].

How come it works?

This raises some questions [28, 2, 10]:

- *Why don't we memorize?*
- *Why don't we learn noise?*
- *What happens during learning?*
- *Why don't we get stuck in a local minima?*

Standard statistical *learning theory* fails: bounds are meaningless.
Standard *regularization* hardly account for the success.

There are various hypothesis to explain the *effectiveness* and the *dynamics* of learning in NNs.

These questions are connected, but not the same as, *interpretable ML*.

Hypotheses from machine learning

- *Universality*: NNs are universal approximators [5]
- *Layering hypothesis*: layering increase NNs expressivity [6]
- *Prior hypothesis*: NNs have strong priors [3]
- *Abstraction hypothesis*: layers extract more and more abstract features [25]. Practical/anecdotal confirmation from observation or fine-tuning of architectures.

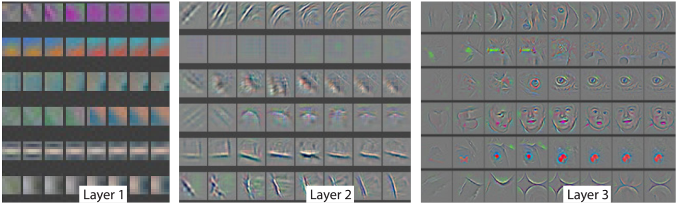


Image from [25]

- *Folding hypothesis*: NNs fold the space and apply piecewise linear functions [15]

Hypotheses from physics

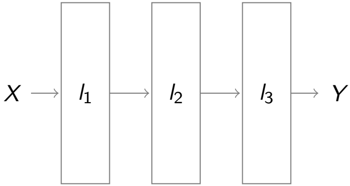
- *Renormalization group theory hypothesis:* NNs carry out the equivalent of a variational RNG. *(NN as stacked RBMs trained by contrastive divergence $\mapsto$ Kadanoff's variational RNG [14])*
- *Information distillation hypothesis:* NNs are successful because they model a generative process that is hierarchical, low-order, local and symmetric. [11] *(Connected to the prior hypothesis)*
- *Many almost-optimal minima hypothesis:* most local minima are equivalent. (*Under assumptions, NNs are related to spin-glass models and analyzed using random matrix theory.*) [4]

3. Information Bottleneck

Hypotheses from information theory

Information bottleneck [24] proposes an *information-theoretic* interpretation to the dynamics of a NN.

Let's reinterpret the compositional perspective of a NN



as a *Markov chain*:

$$X \rightarrow Z_1 \rightarrow Z_2 \rightarrow Z_3 \rightarrow Y$$

Let us call Z_i an *intermediate representation*.

The Information Bottleneck

Good intermediate representations Z_i:

- encodes efficiently X (*compression*)
- eases mapping onto Y (*relevance*)

Ideally Z_i to contain **all and only** the information relevant to Y.

In information-theoretic terms:

- We maximize the compression by *minimizing the mutual information between X and Z*
- We maximize the relevance by *maximizing the mutual information between Z and Y*

This connect to *rate-distortion theory* and the computation of *sufficient statistics*.

The Information Bottleneck (2)

We can re-express this objective as a *single optimization problem*:

$$\arg\min_{Z_i} I\left[X; Z_i\right] - \beta I\left[Z_i; Y\right]$$

where β is a Lagrangian multiplier and trades off compression and relevance.

This optimization has an analytic solution using *Blahut-Arimoto algorithm* [24], but practically estimating mutual informations is hard [8].

This principle has been used both to try to *explain* learning [18] and to *direct* learning [1].

Opening the Black Box of DNN via IB (1)

Can we explain learning in deep neural networks using IB? [18]

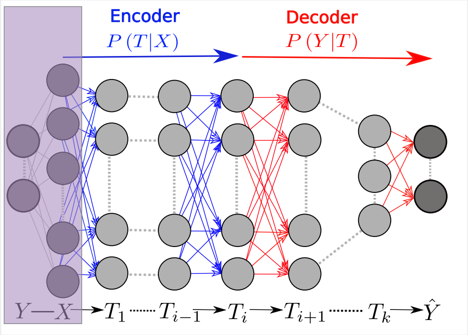


Image from [18] where $T_i = Z_i$

Opening the Black Box of DNN via IB (2)

Can we explain learning in deep neural networks using IB? [18]

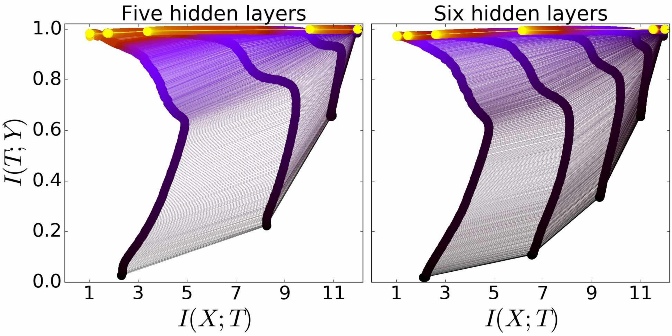


Image from [18] where $T = Z$

- Trajectory in the information plane agrees with IB theory
- (Two different learning phases may be identified)
- (There are some criticisms of this analysis [17])

4. Understanding Unsupervised Learning Algorithms

Unsupervised Learning (UL)

So far, we have dealt with *supervised neural networks* learning from data a mapping $X \mapsto Y$.

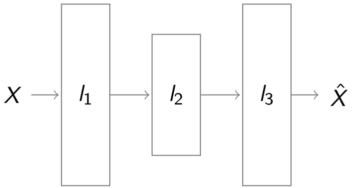
In *unsupervised learning* we only have data X but no target/image Y.

How do we relate *neural networks* to *unsupervised learning*?

- Some algorithms **are** neural networks (*with properly engineered labels*)
- Some algorithms **can be seen** as neural networks (*with properly engineered loss function*)

Auto-encoders

Auto-encoders are NNs that explicitly *compress* and *decompress* the data X.



Data X works as input and as label.

$$\mathcal{L}(f_\Theta(X), X) = D\left[f_\Theta(X), X\right]$$

for some distance measure $D[\cdot,\cdot]$ between the data X and the reconstruction $\hat{X}$.

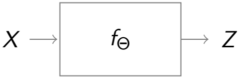
Sparse filtering

Sparse filtering (SF) [16] transform the data X as:

$$Z = f_{\Theta}(X) = \left\| \left\| g\left(WX\right) \right\|_{L2,row} \right\|_{L2,column}$$

where $\Theta = \{W\}$ and *sparsity* is optimized by having $\|Z\|_{L1}$ minimized.

This can be presented as a neural network:



with loss:

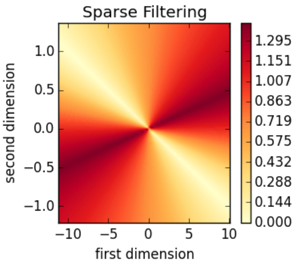
$$\mathcal{L}(f_{\Theta}(X)) = \|Z\|_{L1}$$

optimized by backpropagation.

Aside: Why does sparse filtering

As in the case of NN, SF empirically works well (although not always).
It is surprising that SF learns *good representations* of the data X optimizing a function that just maximizes sparsity.

Why maximizing sparsity work? When does it not? [26]
SF assumes a *structure explained by cosine distance:*

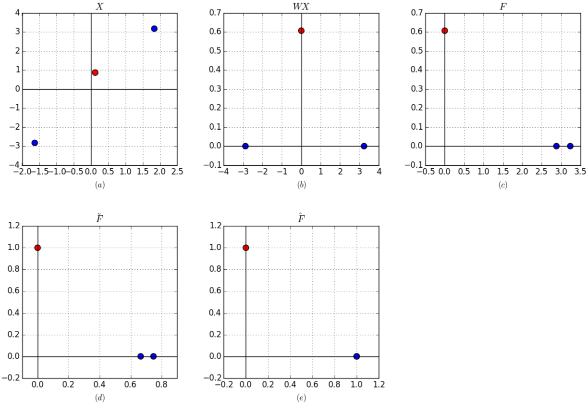


(a)

$$D_{cos}\left[X_1, X_2\right] < \epsilon$$
$$\Rightarrow D_{Eucl}\left[Z_1, Z_2\right] < \delta(\epsilon)$$

Aside: Why does sparse filtering

A perfect learning instance (all points are mapped onto *bases*)



Applying IB to unsupervised learning

IB can not be straightforwardly applied to unsupervised learning:

$$\arg\min_{Z} I\left[X; Z\right] - \cancel{\beta I\left[Z; Y\right]}$$

- We do not have *label information* to anchor too
- Without it, it does not make sense to minimize MI with the input

An alternative general formulation is:

$$\arg\min_{Z} I\left[X; Z\right] - \beta \mathcal{F}\left(Z\right)$$

where $\mathcal{F}$ accounts for some form of structure in Z [20, 19].

Sparse Filtering and IB

In [26], we also conjectured that SF implicitly optimizes a information-theoretic:

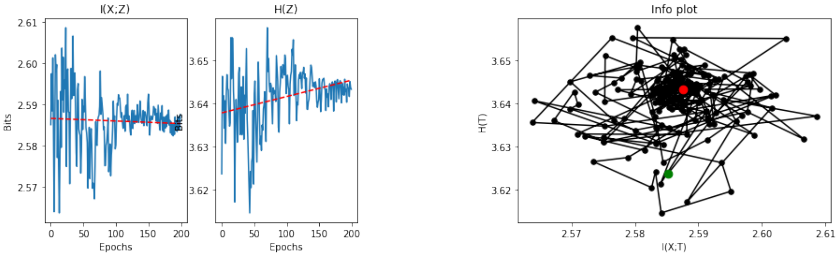
$$\arg\max_{Z} I\left[X;Z\right] - H\left[Z\right]$$

- $I\left[X;Z\right]$: we want SF to preserve information in the input X (Infomax principle [12])
  We assume it coded in the algorithm
- $H\left[Z\right]$: sparsity acts as a proxy for entropy
  We assume it expressed in the loss $\mathcal{L}$

We run some simulations in [27]; however, results are affected by a computational problem and will be soon superseded.

Sparse Filtering and IB

We re-implemented the algorithm in *tensorflow*, and run new simulations[3]:



Information-theoretic objective is challenging:

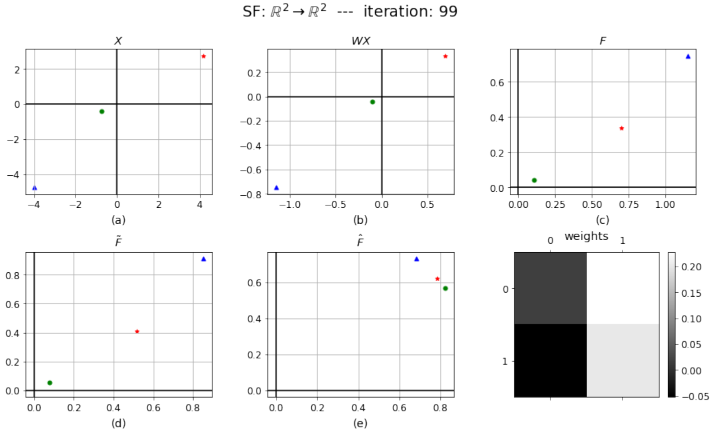
$$I[X;Z] - H[Z] = H[Z] - H[Z|X] - H[Z] \quad (1)$$
$$= -H[Z|X] \quad (2)$$

Assessment of $H[Z|X]$ is noisy if $H[f_\Theta(X)|X]$

[3] https://github.com/FMZennaro/SF-IB/tree/master/v2

Sparse Filtering and IB

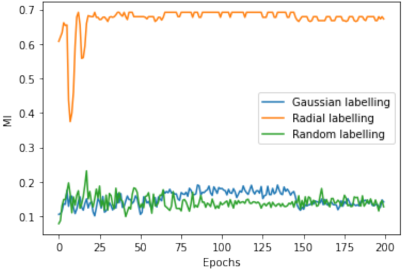
Replacing the sparsity proxity with explicit entropy minimization leads to a collapse of the representations:



Sparsity is a proxy for entropy minimization only locally?

Sparse Filtering and IB

We may still use IB with *virtual labels*:



This may be a generic empirical approach to check out assumptions behind unsupervised learning algorithms?

Conclusions

- IB is a very general theory of learning
- There are alternative information bottleneck formulations [20, 22]
- This is not the only information-theoretic principle we can use for learning [21]
- Application to UL may be very interesting!

Thanks

Thank you for listening!

References I

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[2] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. *arXiv preprint arXiv:1706.05394*, 2017.

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[4] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pages 192–204, 2015.

References II

[5] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[6] Olivier Delalleau and Yoshua Bengio. Shallow vs. deep sum-product networks. In *Advances in neural information processing systems*, pages 666–674, 2011.

[7] Brendan Fong, David I Spivak, and Rémy Tuyéras. Backprop as functor: A compositional perspective on supervised learning. *arXiv preprint arXiv:1711.10455*, 2017.

[8] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

References III

[10] David Krueger, Nicolas Ballas, Stanislaw Jastrzebski, Devansh Arpit, Maxinder S Kanwal, Tegan Maharaj, Emmanuel Bengio, Asja Fischer, and Aaron Courville. Deep nets don't learn via memorization. 2017.

[11] Henry W Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247, 2017.

[12] Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

[13] David J.C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

[14] Pankaj Mehta and David J Schwab. An exact mapping between the variational renormalization group and deep learning. *arXiv preprint arXiv:1410.3831*, 2014.

References IV

[15] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pages 2924–2932, 2014.

[16] Jiquan Ngiam, Zhenghao Chen, Sonia A Bhaskar, Pang W Koh, and Andrew Y Ng. Sparse filtering. In *Advances in Neural Information Processing Systems*, pages 1125–1133, 2011.

[17] Andrew Michael Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan Daniel Tracey, and David Daniel Cox. On the information bottleneck theory of deep learning. 2018.

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

References V

[19] Noam Slonim, Nir Friedman, and Naftali Tishby. Unsupervised document classification using sequential information maximization. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 129–136. ACM, 2002.

[20] Noam Slonim and Naftali Tishby. Agglomerative information bottleneck. In *Advances in Neural Information Processing Systems*, pages 617–623, 2000.

[21] Greg Ver Steeg. Unsupervised learning via total correlation explanation. *arXiv preprint arXiv:1706.08984*, 2017.

[22] DJ Strouse and David J Schwab. The deterministic information bottleneck. *Neural computation*, 29(6):1611–1630, 2017.

[23] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

References VI

[24] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[25] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks (2013). *arXiv preprint arXiv:1311.2901*, 2013.

[26] Fabio Massimo Zennaro and Ke Chen. Towards understanding sparse filtering: A theoretical perspective. *Neural Networks*, 98:154–177, 2018.

[27] Fabio Massimo Zennaro and Ke Chen. Towards further understanding of sparse filtering via information bottleneck. *arXiv preprint arXiv:1910.08964*, 2019.

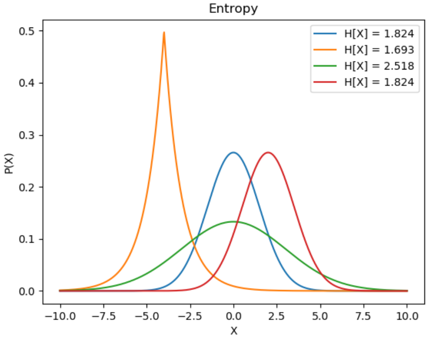
[28] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Entropy

Entropy of a random variable X:

$$H[X] = -\sum_{x} p(x) \log p(x)$$

- Statistical descriptor
- Domain-insensitive
- Measure of information
- Measure of uncertainty
- Measure of concentration



Mutual Information

Mutual information of two random variables X, Y:

$$I[X;Y] = H[X] - H[X|Y] = H[Y] - H[Y|X]$$

- Invariant to invertible reparametrization
- Measure of shared information
- Measure of reduction of uncertainty

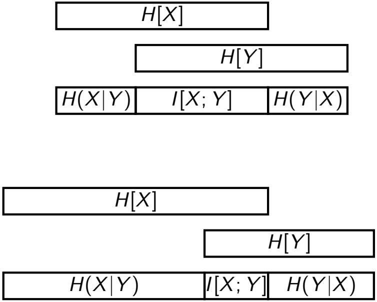


Diagram from [13]