”Opening the black box of Deep Neural Networks via Information” by Schwarz-Ziv and Tishby

Report for ”Graphical Models and Statistical Inference” course at Skoltech

Bogdan Kirillov

October 29, 2017

Contents

1 Introduction

1.1 Short description of the problem

Recenlty attention of world's A.I. scientists is centered on Deep Neural Networks. The reason why is DNNs performance which is close to[1] and sometimes above human levels. DNNs can solve a lot of problems that were considered human-complete before with results that even surpasses human's abilities[2].

This power comes with the following critical issues:

Lack of theory Currently there is no real theory behind why do the DNNs work. DNNs are large non-linear dynamical systems, among the largest that humanity ever encountered. The complexity of DNNs stops attempts to unravel their inner mechanics by classical mathematical methods.

Black box Understanding why did the network put that particular label for this particular example is also very challenging. The DNNs are used today as black-box model without reliable possibility of interpretation. Graying this black box is an area of active research now[3].

Because of those facts the field has limitations in its applicability: one wouldn't really use an algorithm that one doesn't understand completely (and actually nobody does), especially in serious settings when human lives depend on the model, e.g. in medicine, disaster mitigation, criminal investigations etc.

The paper we are talking about[4] is written to shed light on the first issue. It provides an attempt to describe DNN's *modus operandi* theoretically by applying modern methods of pure mathematics.

1.2 Key points of the paper by Schwarz-Ziv and Tishby

The paper of interest[4] can be roughly summarized to the following list of key points:

- Most of the training time is spent on compression of the input to efficient representation rather than on fitting to labels;
- The compression begins when the training errors shrink and the SGD epochs go into random diffusion, constrained by the training error value;
- The converged layers lie very close to the 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. This mechanism is absent in one layer networks;

- The output of a neuron in hidden layer is irrelevant to the prediction of label *per se*. Only the layered structure counts;
- Adding more hidden layers reduces training time;
- Hidden layers are expected to converge to critical points for phase transition of Informational Bottleneck curve.

1.3 Structure of this report

This report is structured as follows:

1. The first part of the report is intended to describe the contributions of Schwarz-Ziv and Tishby I have considered significant;
    - Section ”Information theory of Deep Learning” describes how the authors state the ways to apply classical information-theoretic approaches for the field of Deep Learning;
    - Section ”Network behavior on Information Plane” speaks about what Information Plane is and why authors consider it significant for the problem of Deep Neural Network Interpretation;
    - Section ”Hidden layers” concentrates on explanation of hidden layers *modus operandi* provided by Schwarz-Ziv and Tishby;
2. The second part is left for description of my numerical experiments intended to reproduce some of the paper’s plots and computations on a different setting;
    - Section ”Numerical experiments setup” describes differences of my setup compared to Schwarz-Ziv and Tishby’s;
    - Section ”Results of numerical experiments” shows how my results differ from and how converge to Schwarz-Ziv and Tishby’s, also here I speculate about the reason and meaning of the differences.
3. The third part roughly reviews current applications of the paper and foreshadows possible future extensions of the field.

2 Information theory of Deep Learning

Main reason to do supervised learning is to get some representation of input data that aids prediction of a label (doesn’t matter whether it is integer as in Classification or real as in Regression) not only on the training set but also in general case on data previously not seen by the model. One would want to be able to learn such representation from experimental data given the joint distribution of input data and labels is unknown.

In their previous work[5], the authors suggested that DNNs form a Markov chain of such representations. They take form of network's hidden layer. Their first important insight is to consider output of a layer as single multidimensional random variable with its own decoder ($P(Y|T)$) and encoder ($P(T|X)$) distributions (as shown on figure below).

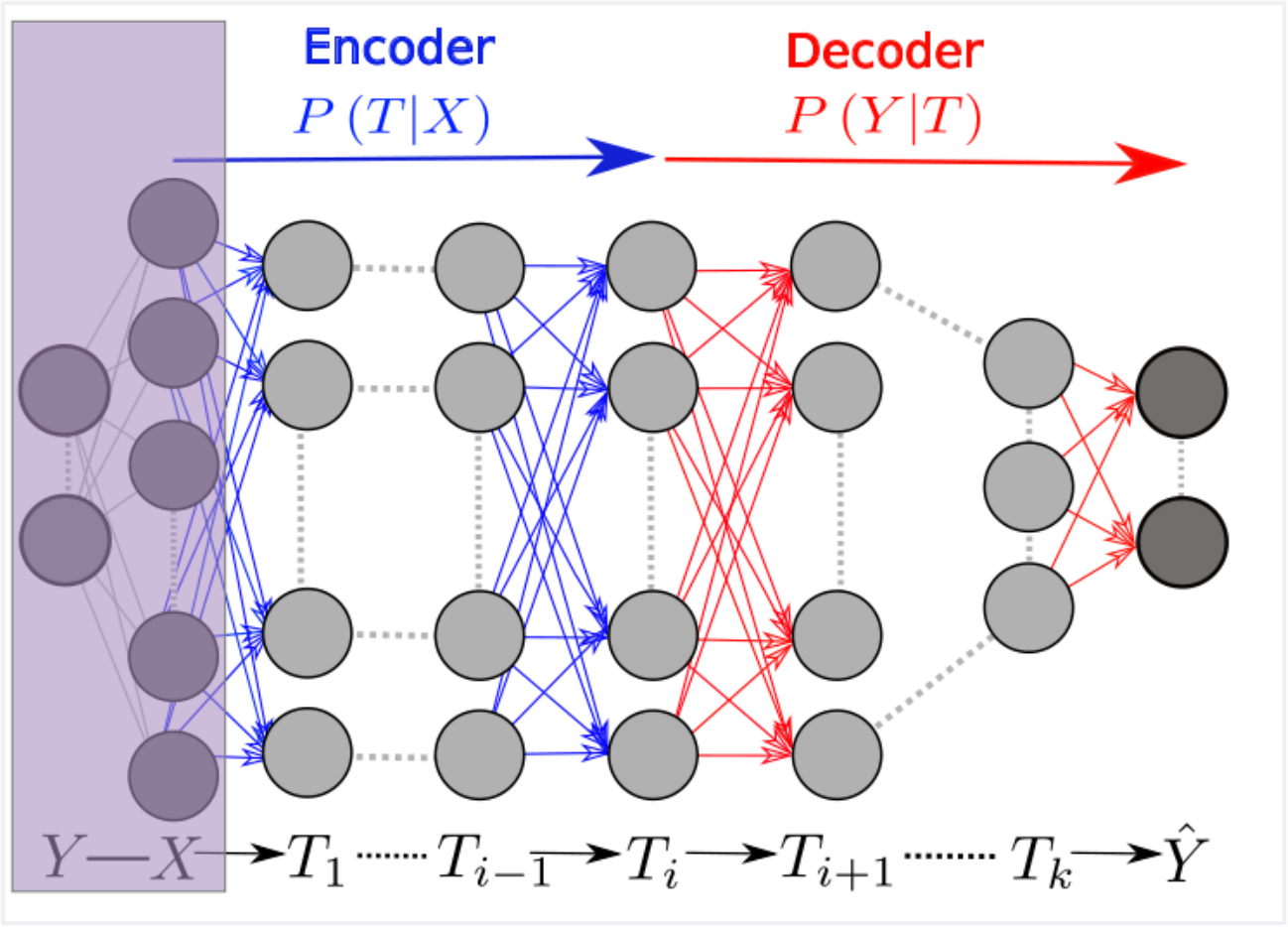

Figure 2.1: Markov chain of successive representations. T - representation, X - input, Y - label.

Before I define an Information Plane as the authors do, important things should be stated about the measures they use.

Their work is built around the Mutual Information:

$$I(Input; Label) = D_{KL}[p(i,l)||p(i)p(l)] = \sum_{l \in Label; i \in Input} p(i,l) log \frac{p(i,l)}{p(i)p(l)} \quad (2.1)$$

This formula basically means the following: if one receives two messages, the amount of redundant information one gets is equal to Kullback-Leibler divergency between their joint distribution and a multiplication of their distributions. By "redundant" I mean the information both messages share.

MI has two important properties (according to authors):

Invariance to invertible transformations

$$\forall \phi, \gamma : \exists \phi^{-1}, \gamma^{-1} \Rightarrow I(A; B) = I(\phi(A); \gamma(B)) \tag{2.2}$$

Mutual information remains unchanged if we apply to data some transformation that has an inverse.

Data-Processing Inequality

$$\forall X, Y, Z : (X \to Y \to Z) \Rightarrow I(X; Y) \geq I(X; Z) \tag{2.3}$$

where $A \to B... \to Z$ - Markov chain, X, Y, Z - random variable.
Playing with data will never increase information.

In the context of Schwarz-Ziv and Tishby's paper, the MI is computed between representation of input and label and shows amount of information relevant to predict the label. They state the idea of *optiman encoder* that efficiently represents relevant information.

The key concept here is that Machine Learning is just a form of Data Compression: when one tries to train a model, he/she is actually trying to make an algorithm for lossy compression that extracts all the information relevant for predicting a label and loses all irrelevant.

This notion is captured in Information Bottleneck tradeoff:

$$\min_{p(r|i), p(l|r), p(r)} I(Input; Representation) - \beta I(Representation; Label) \tag{2.4}$$

β is a Lagrange multiplier and determines the level of relevant information captured by the representation. The implicit solution for IB tradeoff is given by self-consistent equations:

$$\begin{cases} p(r|i) = \frac{p(r)}{Z(i;\beta)} exp(-\beta D_{KL}[p(l|i)||p(l|r)]) \\ p(r) = \sum_i p(r|i)p(i) \\ p(l|r) = \sum_i p(l|i)p(i|r) \end{cases} \tag{2.5}$$

where $Z(i; \beta)$ is a partition function.

3 Network behavior on Information Plane

Information Plane is a plane given by computations of $I(Input|Representation)$ (X-axis) and $I(Representation|Label)$(Y-axis). It provides a visual way of looking at

the training process. When model trains, its infoplane coordinates shift. Each layer of the model has a point on the plane (as shown below, the plot is from [4]).

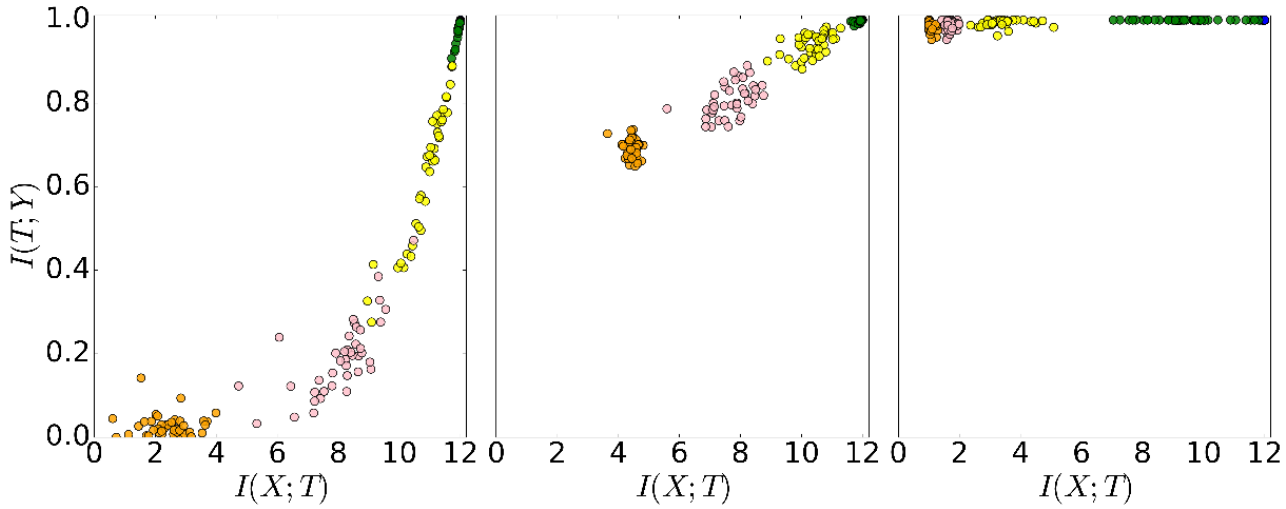

Figure 3.1: Snapshots of layers of 50 randomized networks: left - with the initial weights; center - at 400 epochs (ERM); right - after 9000 epochs (RC); layers are denoted by color.

Observation of network's movement on infoplane allowed authors to find out two distinct stages of Stochastic Gradient Descent training:

ERM

- takes few hundred epochs;
- layers increase information on labels;
- order of layers is preserved;

Representation compression

- takes a lot of time;
- layers drop irrelevant information on input data;

Authors try to further investigate ERM and representation-compression phases by looking at behavior of of the stochastic gradients along the epochs (figure below[4]):

The transition between two phases of SGD is clearly seen on the plot (marked by gray line). The first is a *drift* phase, where the gradient means are much larger than their standard deviations, indicating small gradient stochasticity and in the second phase called the *diffusion* phase, the gradient means are very small compared to their batch to batch fluctuations, and the gradients behave like Gaussian noise with very small means, for each layer.

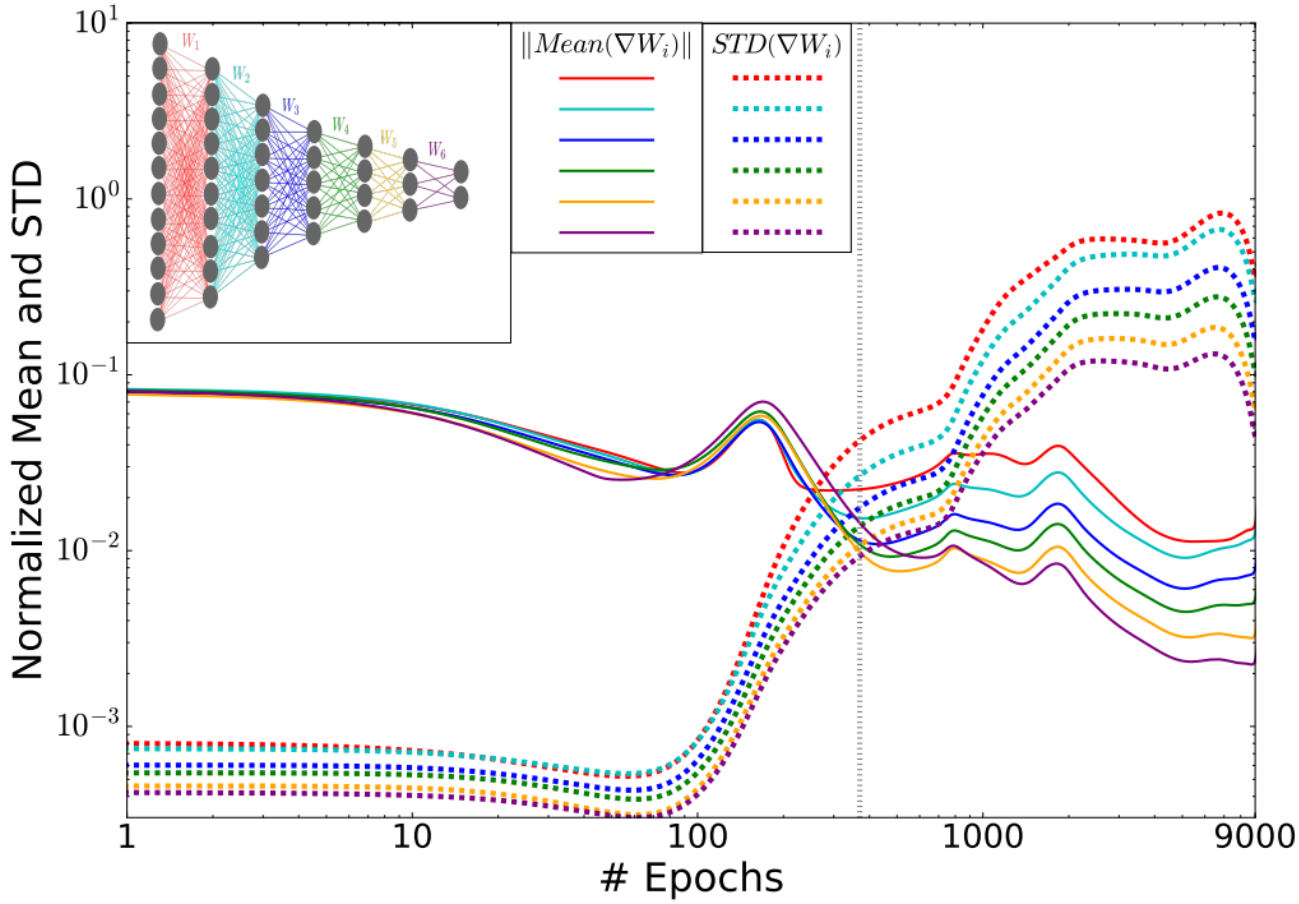

Figure 3.2: The layers' Stochastic Gradients distributions during the optimization process. The norm of the means and standard deviations of the weights gradients for each layer, as function of the number of training epochs (in log-log scale).

Authors note the very significant consequence of representation compression by diffusion: there are a lot of different networks with optimal performance so any attempt to interpret a weight or an output of neuron is meaningless. Next time the network will not converge to the same optimal set of weights but to different one.

4 Hidden layers

To understand why the network needs hidden layers they have conducted a series of experiments (results are shown below[4]) as follows:

1. 6 different architectures with 1 to 6 hidden layers;
2. Repeat each experiment 50 times with randomized initial weights and training samples.

They have reached the following results:

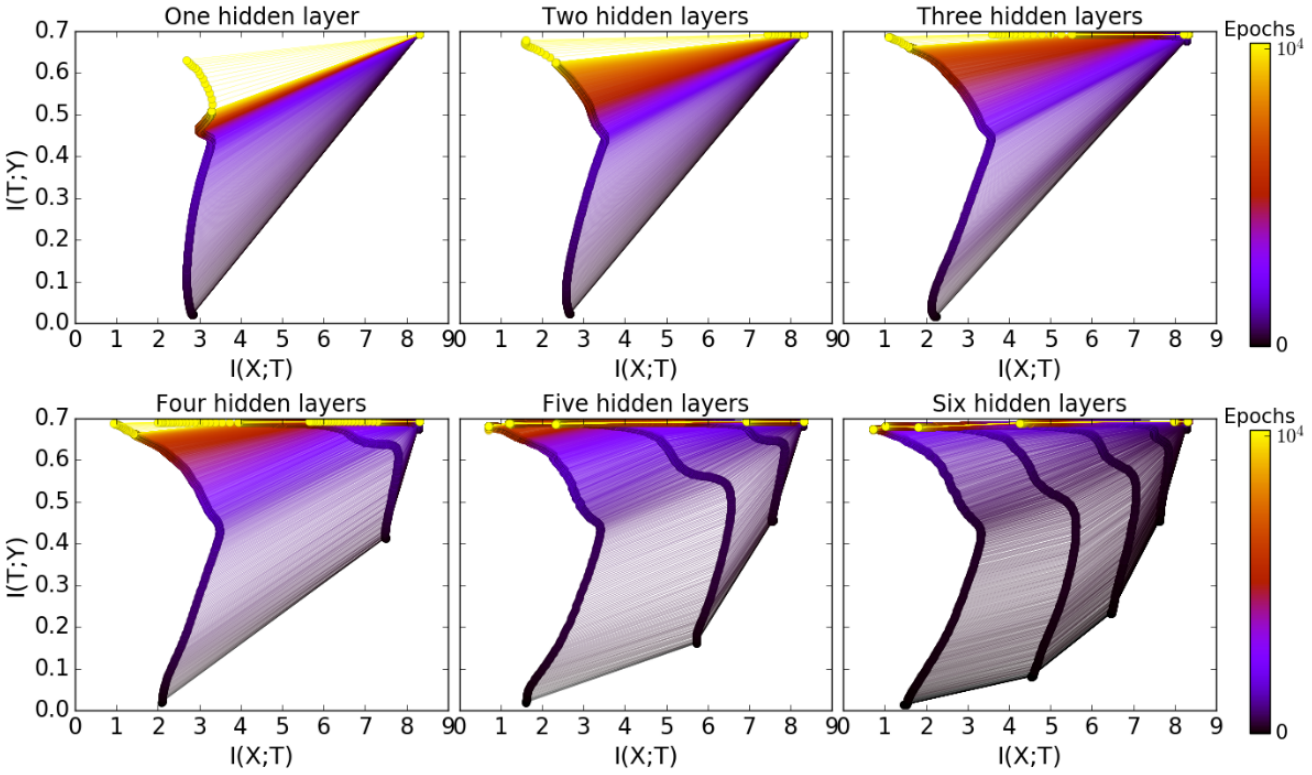

Figure 4.1: The layers information paths during the SGD optimization for different architectures. Each panel is the information plane for a network with a different number of hidden layers. The width of the hidden layers start with 12, and each additional layer has 2 fewer neurons. The final layer with 2 neurons is shown in all panels. The line colors correspond to the number of training epochs.

- More hidden layers - good generalization is reached faster;
- Compression phase of a layer is shorter if it is started from previously compressed layer;
- For the deeper layers the compression is faster;
- Wide layers also eventually converge in diffusion.

5 Numerical experiments setup

In this section I describe my experimental setup. I have tried to reproduce some of Schwarz-Ziv and Tishby results on a different setup. The idea is to see whether those results are invariant to:

- the task at hand - Schwarz-Ziv and Tishby used a toyish task that was easy to solve and analyze. I've tried more difficult real data task - identification of digits on MNIST[6];
- the way to estimate Mutual Information - Schwarz-Ziv and Tishby used a very direct method that is constructed easily for their toy task (I omit the description of

their method here, look for it at [4]). I use NPEET library[7] for non-parametric general purpose MI estimation;

- training algorithm - they used plain SGD, but I used Adam[8] since it is the most used variation of SGD for now.

MNIST (Mixed National Institute of Standards and Technology) is a database that contains a lot of hand-written digits. It is a very popular benchmark that allows to compare the performance of different Machine Learning models.

Typical MNIST workflow is as follows:

1. Split the MNIST into test and train sets;
2. Train the model using raw pixels as an input and 10 classes as a label;
3. Test the model on hold-out set;

For this report to simplify things I use only 1000 numbers from MNIST as train set and 100 numbers as test set. Also I'll train the models not to classify label of a digit, but to regress an actual digit. As loss function I use mean absolute error:

$$MAE = \frac{1}{n}\sum_i |y_{prediction} - y_{real}| \tag{5.1}$$

For this report I am not interested in performance of the models in terms of accuracy or mean absolute error, I only want to reproduce the plots similar to **??**.

All code to reproduce my experiments can be obtained from [9].

6 Results of numerical experiments

The network used is fully-connected with 4 [256,128,64,32] hidden layers, trained to perform Regression. Activations are ELU everywhere except the output layer which is Linear. The layer is marked by color in the following way: black, red, green, blue, yellow.

As you can see below the plot complies with the respective plot of Schwarz-Ziv and Tishby. We see that after 400 and 9000 epochs MI between representation and label went up, the clear order of layers started to emerge (from black(input) to yellow(output)).

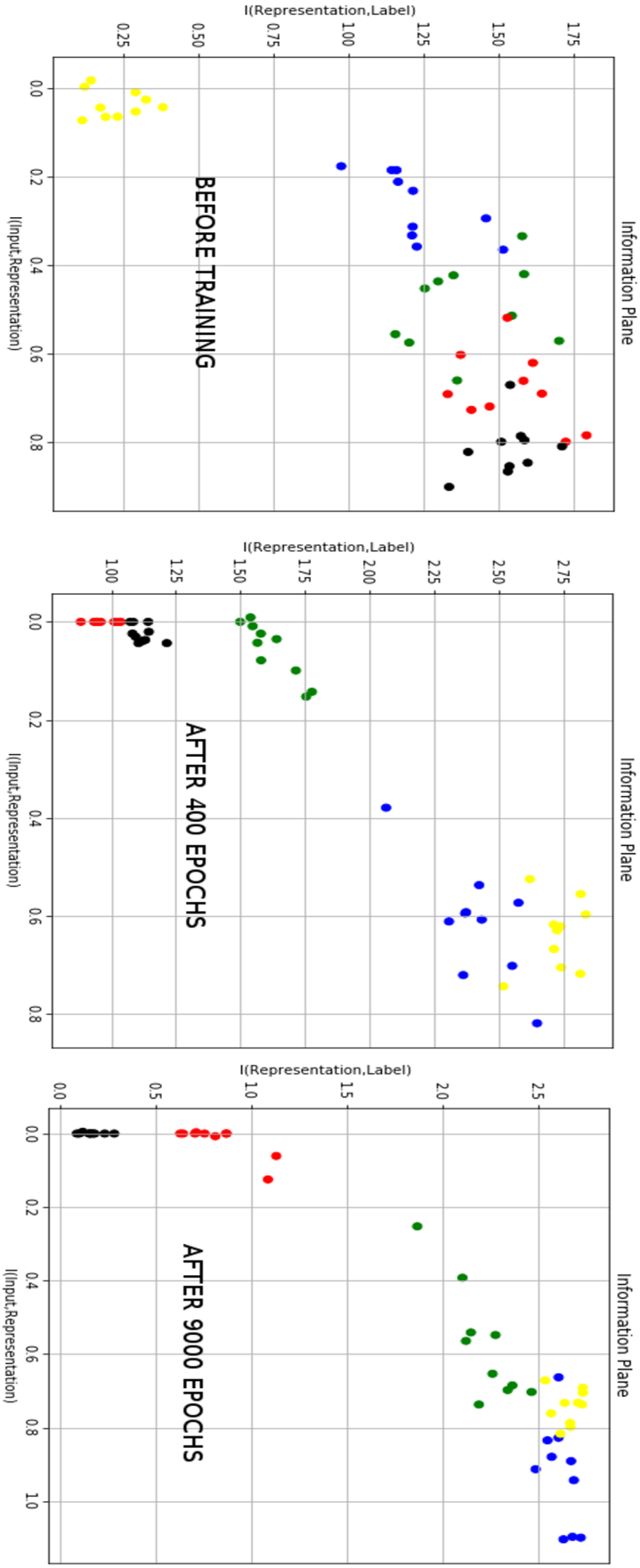

Figure 6.1: MNIST Information Plane snapshots

7 Discussion

For me, the one most important thing from this paper is that it is possible to reason about model training in terms of moving around the infoplane. It offers quite a range of possibilities:

- Easy visual comparison of the model's performance. If one has two models A and B and A is higher on Y-axis and farther at X-axis from (0,0) than B, one can say that A is better trained because is able to extract more from the data and predicts the label better;
- IB tradeoff can be used as a promising loss function. If one wants a good model, one can add the IB tradeoff to existing loss function as a regularization term, or just use the tradeoff instead of typical loss;
- Further investigation in this direction can lead to new insight into inner mechanisms of DNNs.

Schwarz-Ziv and Tishby provide the library called IDNNs [10] that can be used to visualize information plane and network behavior.

Information Bottleneck as loss function is already implemented in [11]. They parametrize IB as Neural Network, leverage the reparameterization trick for efficient training and got 1.13% error on MNIST. As a byproduct they have found out that IB loss improves robustness to adversarial inputs. This direction also is worth investigating

The other inspiring thing is that they results actually are invariant to changes I descirbed in previous two sections. Scientific result should be the same regardless of method it was obtained by. It implies that IB-approach could be of good use for Deep Learning practitioners in both industry and science.

References

1. Dermatologist-level classification of skin cancer with deep neural networks / A. Esteva [et al.] // Nature. — 2017. — Vol. 542, no. 7639. — Pp. 115–118.
2. *Lu C., Tang X.* Surpassing Human-Level Face Verification Performance on LFW with GaussianFace. // AAAI. — 2015. — Pp. 3811–3819.
3. Towards better analysis of machine learning models: A visual analytics perspective / S. Liu [et al.] // Visual Informatics. — 2017. — Vol. 1, no. 1. — Pp. 48–56.
4. *Shwartz-Ziv R., Tishby N.* Opening the Black Box of Deep Neural Networks via Information // arXiv preprint arXiv:1703.00810. — 2017.
5. *Tishby N., Zaslavsky N.* Deep learning and the information bottleneck principle // Information Theory Workshop (ITW), 2015 IEEE. — IEEE. 2015. — Pp. 1–5.
6. *LeCun Y.* The MNIST database of handwritten digits // http://yann. lecun. com/exdb/mnist/. — 1998.
7. *Ver Steeg G.* Non-parametric entropy estimation toolbox (npeet). — 2000.
8. *Kingma D., Ba J.* Adam: A method for stochastic optimization // arXiv preprint arXiv:1412.6980. — 2014.
9. *Kirillov B.* Numerical experiments for the report. — URL: `https://github.com/bakirillov/GM_Skoltech`.
10. *Schwarz-Ziv R.* IDNNs. — URL: `https://github.com/ravidziv/IDNNs`.
11. Deep Variational Information Bottleneck / A. A. Alemi [et al.] // arXiv preprint arXiv:1612.00410. — 2016.