

On the Information Bottleneck Theory of Deep Learning

Team No. 1

Anurag, Abhishek, Vikram, Mudit



Intro

Here we are reviewing and implementing the concepts in the paper -

“On the Information Bottleneck Theory of Deep Learning”.



Objective

Theoretical Progress is still playing catch up with the astounding success of Deep Learning Models.

Theory of Information Bottleneck is recently used to study DNNs.

Shwartz-Ziv and Tishby proposed the information bottleneck that expresses the tradeoff between the mutual information measures $I(X,T)$ and $I(T,Y)$.

This paper is an attempt to better understand in the IB theory.



Claims

Here we falsify 3 claims made under the IB theory of Deep Learning made by Shwartz-Ziv and Tishby :

1. Deep networks undergo two distinct phases which consists
 - initial fitting phase and
 - a subsequent compression phase.
2. The compression phase is causally related to the excellent generalization performance of deep networks
3. The compression phase occurs due to the diffusion-like behavior of stochastic gradient descent.



Why these claims are wrong?

We show that

1. Information oplan trajectory is purely function of neural non linearity used: tanh yield a compression phase as soon as neural activations enter the saturation regime . Relu does not show this.It gives a linear and single sided saturating non-linearity.
2. There is no causal connection between compression and generalization as discovered in the experiment
3. Compression phase doesn't arise from stochasticity in training (SGD) as same properties were observed by trying with full batch gradient.
4. When input contains relevant and irrelevant info, hidden representation compress irrelevant information.
5. While training, the compression happens concurrently with fitting(not subsequently).



Some Basic prerequisites to mention

- In this view, deep learning is a question of representation learning: each layer of a deep neural network can be seen as a set of summary statistics which contain some but not all of the information present in the input, while retaining as much information about the target output as possible.
- The amount of information in a hidden layer regarding the input and output can then be measured over the course of learning, yielding a picture of the optimization process in the information plane . This method holds the promise to serve as a general analysis that can be used to compare different architectures, using the common currency of **mutual information**.
- The elegant information bottleneck (IB) theory provides a fundamental bound on the amount of input compression and target output information that any representation can achieve .
- The IB bound thus serves as a method-agnostic ideal to which different architectures and algorithms may be compared



Definitions

Mutual Information:

In probability theory and information theory, the mutual information (MI) of two random variables is a measure of the mutual dependence between the two variables. More specifically, it quantifies the "amount of information" (in units such as shannons (bits), nats or hartleys) obtained about one random variable through observing the other random variable.



Entropy and Information example

Entropy: Measure of uncertainty before the flip.

Information: Knowledge you have to gain after the flip.

"5 letter word" or "5 flip of coins", which one contains more information?

1) To guess 5 letter word, we can guess each letter using binary search, at 4.7 questions.

Hence amount of information in 5 letter word = $(5 \times 4.7) = 23$ questions.

2) To guess 5 flip we have to 1 question to each flip, hence amount of information in 5 flip of coins is $5 \times 1 = 5$.

Hence amount of information in 5 letter word > 5 flip of coins.

Information too can be measured and compared called as entropy.



COMPRESSION AND NEURAL NONLINEARITIES

TANH

- A neural network with 7 fully connected hidden layers of width 12-10-7-5-4-3-2 is trained with stochastic gradient descent to produce a binary classification from a 12-dimensional input. In our replication we used 256 randomly selected samples per batch.
- The mutual information of the network layers with respect to the input and output variables is calculated by binning the neuron's tanh output activations into 30 equal intervals between -1 and 1. Discretized values for each neuron in each layer are then used to directly calculate the joint distributions, over the 4096 equally likely input patterns and true output labels.
- We see a transition between an initial fitting phase, during which information about the input increases, and a subsequent compression phase, during which information about the input decreases.

RELU

- We then modified the code to train deep networks using rectified linear activation functions ($f(x) = \max(0, x)$).
- While the activities of tanh networks are bounded in the range $[-1, 1]$, ReLU networks have potentially unbounded positive activities. To calculate mutual information, we first trained the ReLU networks, next identified their largest activity value over the course of training, and finally chose 100 evenly spaced bins between the minimum and maximum activity values to discretize the hidden layer activity.
- The resulting information plane dynamics are shown in Fig. 1B

Figure 1: Information plane dynamics and neural nonlinearities. (

The x-axis plots information between each layer and the input, while the y-axis plots information between each layer and the output.

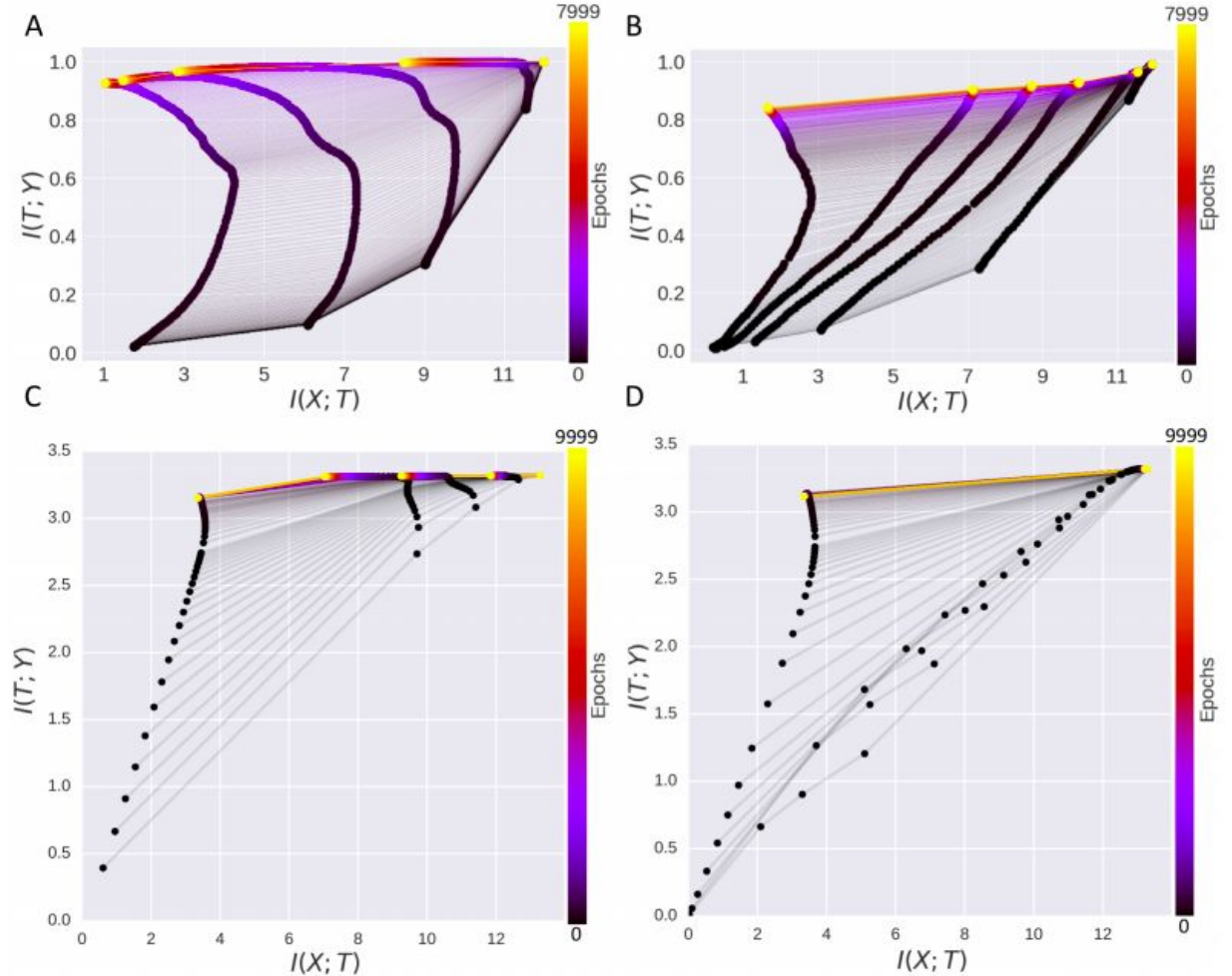
The color scale indicates training time in epochs. Each of the six layers produces a curve in the information plane with the input layer at far right, output layer at the far left.

Different layers at the same epoch are connected by fine lines.

A) for a network with tanh nonlinearities (except for the final classification layer which contains two sigmoidal neurons).

(B) Information plane dynamics with ReLU nonlinearities (except for the final layer of 2 sigmoidal neurons). Here no compression phase is visible in the ReLU layers. F

(C) Information plane dynamics for a tanh network of size $784 - 1024 - 20 - 20 - 20 - 10$ trained on MNIST, estimated using the non-parametric kernel density mutual information estimator ,no compression is observed except in the final classification layer with sigmoidal neurons. See Appendix B for the KDE MI method applied to the original Tishby dataset; additional results using a second popular nonparametric k-NN-based method (Kraskov et al., 2004); and results for other neural nonlinearities.



Nonlinear compression in a minimal model

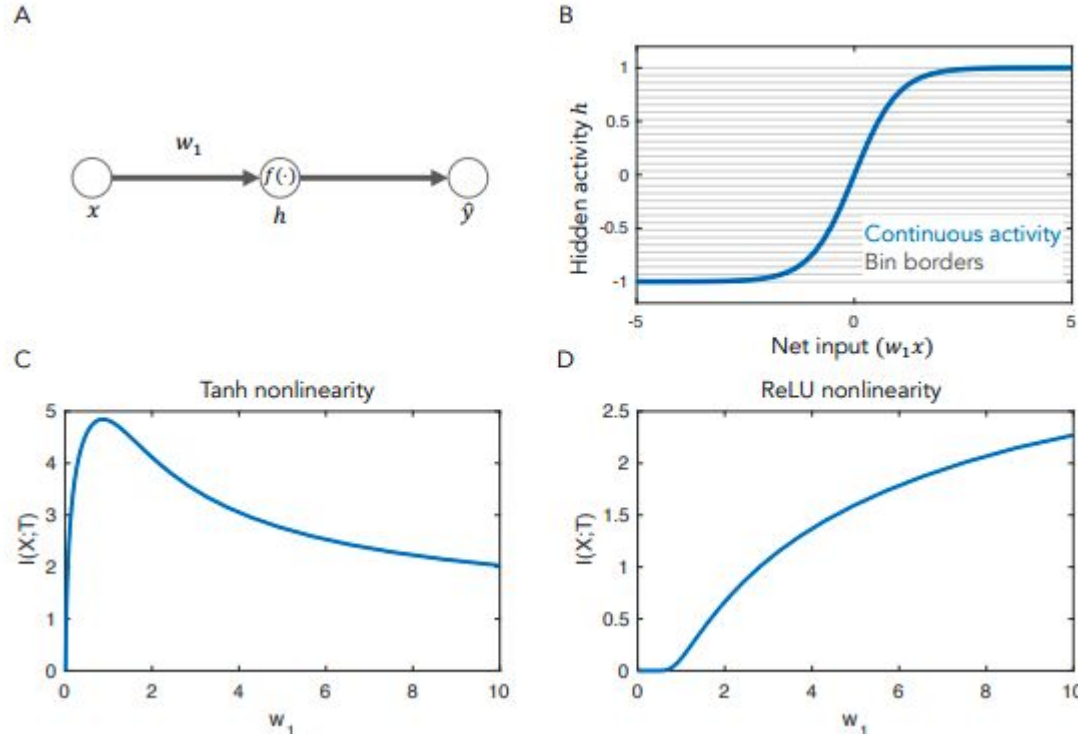


Figure 2: Nonlinear compression in a minimal model. (A) A three neuron nonlinear network which receives Gaussian inputs x , multiplies by weight w_1 , and maps through neural nonlinearity $f(\cdot)$ to produce hidden unit activity h . (B) The continuous activity h is binned into a discrete variable T for the purpose of calculating mutual information. Blue: continuous tanh nonlinear activation function. Grey: Bin borders for 30 bins evenly spaced between -1 and 1. Because of the saturation in the sigmoid, a wide range of large magnitude net input values map to the same bin. (C) Mutual information with the input as a function of weight size w_1 for a tanh nonlinearity. Information increases for small w_1 and then decreases for large w_1 as all inputs land in one of the two bins corresponding to the saturation regions. (D) Mutual information with the input for the ReLU nonlinearity increases without bound. Half of all inputs land in the bin corresponding to zero activity, while the other half have information that scales with the size of the weights.



INFORMATION PLANE DYNAMICS IN DEEP LINEAR NETWORKS

- The preceding section investigates the role of nonlinearity in the observed compression behavior, tracing the source to double-saturating nonlinearities and the binning methodology used to calculate mutual information. However, other mechanisms could lead to compression as well.
- Even without nonlinearity, neurons could converge to highly correlated activations, or project out irrelevant directions of the input
- In a student-teacher setting, one “student” neural network learns to approximate the output of another “teacher” neural network.
- We consider a scenario where a linear teacher neural network generates input and output examples which are then fed to a deep linear student network to learn.
- we assume multivariate Gaussian inputs $X \sim N(0, I_{N_i})$ and a scalar output Y . The output is generated by the teacher network according to $Y = W_o X + o$, where $o \sim N(0, \sigma_o^2)$ represents aspects of the target function which cannot be represented by a neural network.
- To train the student network, a dataset of P examples is generated using the teacher. The student network is then trained to minimize the mean squared error between its output and the target output using standard (batch or stochastic) gradient descent on this dataset.
- Fig. 3 shows example training and test dynamics over the course of learning in panel C, and The linear network behaves qualitatively like the ReLU network, and does not exhibit compression.
- Nevertheless, it learns a map that generalizes well on this task and shows minimal overtraining. Hence, in the setting we study here, generalization performance can be acceptable without any compression phase.

Generalization and information plane dynamics in deep linear networks

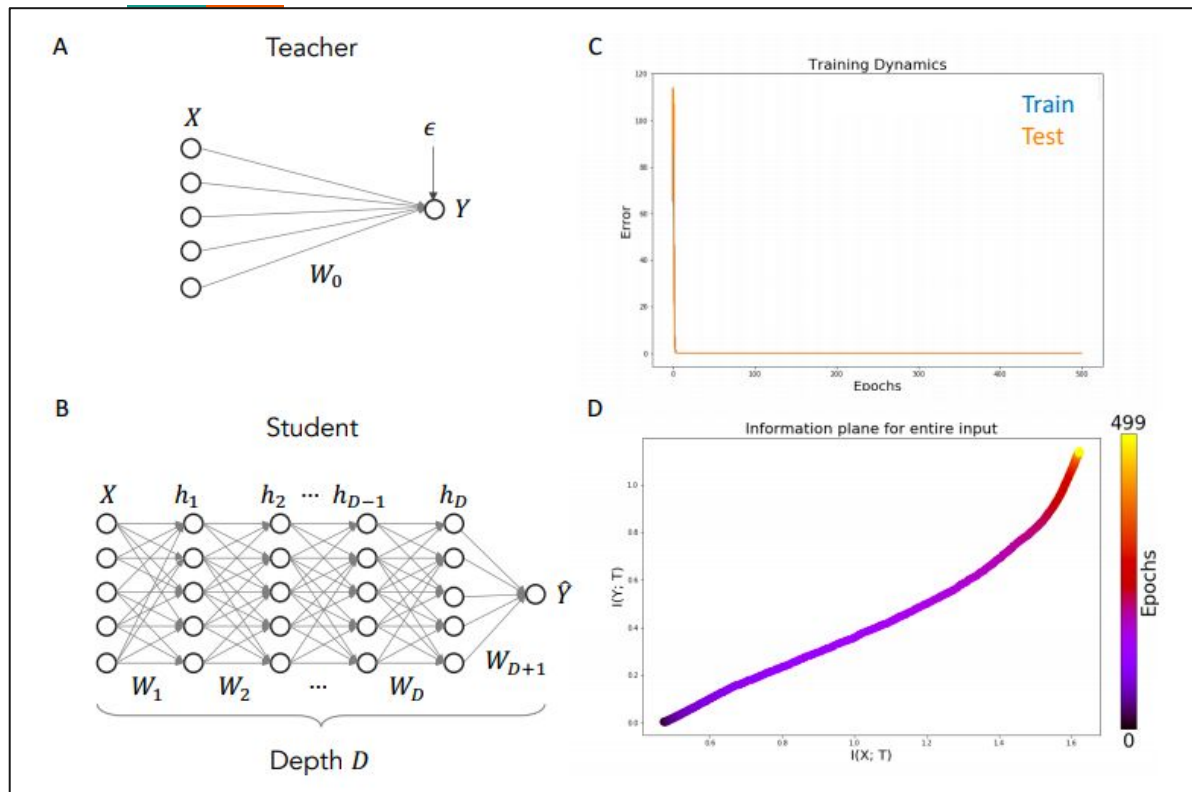


Figure 3: Generalization and information plane dynamics in deep linear networks. (A) A linear teacher network generates a dataset by passing Gaussian inputs X through its weights and adding noise. (B) A deep linear student network is trained on the dataset (here the network has 1 hidden layer to allow comparison with Fig. 4A, see Supplementary Figure 18 for a deeper network). (C) Training and testing error over time. (D) Information plane dynamics. No compression is observed.



Overtraining and information plane dynamics.

- Fig. 4 shows learning dynamics with the number of samples matched to the size of the network. Here overfitting is substantial, and again no compression is seen in the information plane.
- Hence, in this linear analysis of a generic setting, there do not appear to be additional mechanisms that cause compression over the course of learning; and generalization behavior can be widely different for networks with the same dynamics of information compression regarding the input.
- Here the tanh networks show substantial compression, despite exhibiting overtraining. This establishes a dissociation between behavior in the information plane and generalization dynamics: networks that compress may (Fig. 1A) or may not (Fig. 4C-D) generalize well, and networks that do not compress may (Figs. 1B, 3A-B) or may not (Fig. 4A-B) generalize well.

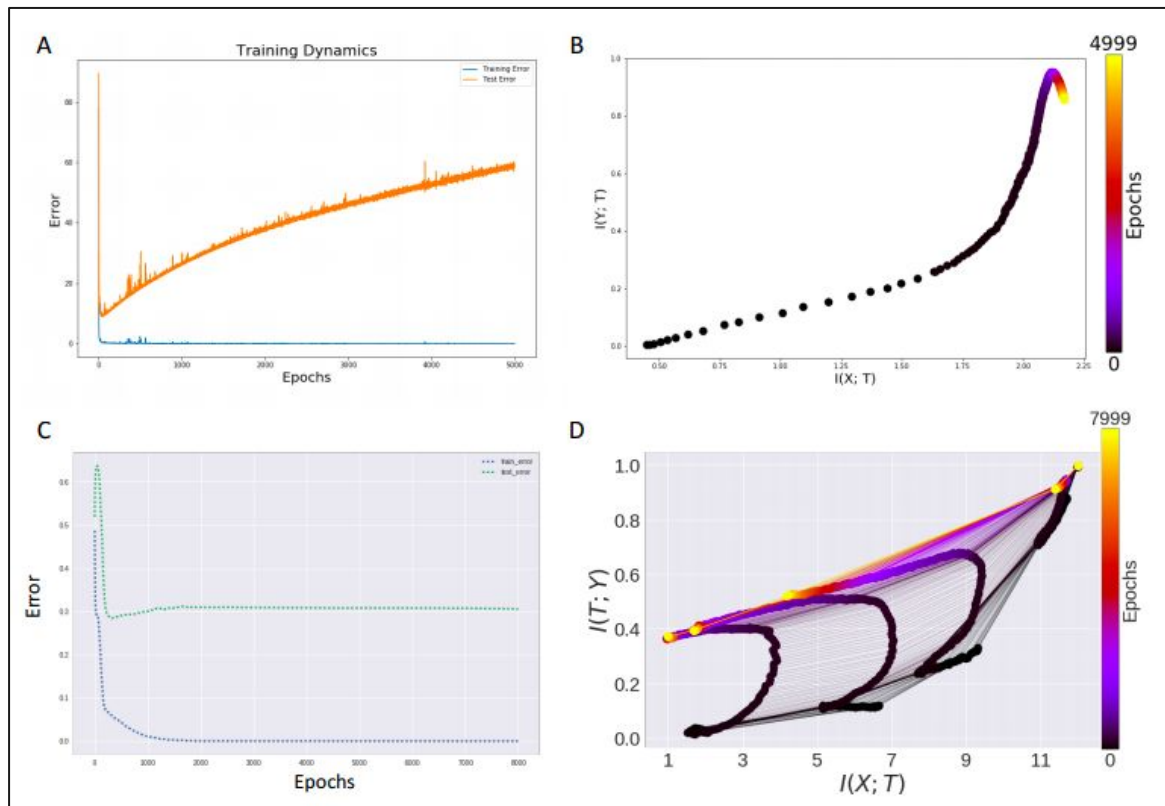


Figure 4: Overtraining and information plane dynamics. (A) Average training and test mean square error for a deep linear network trained with SGD. Overtraining is substantial. Other parameters: $N_i = 100$, $P = 100$, Number of hidden units = 100, Batch size = 5 (B) Information plane dynamics. No compression is observed, and information about the labels is lost during overtraining. (C) Average train and test accuracy (% correct) for nonlinear tanh networks exhibiting modest overfitting ($N = 8$). (D) Information plane dynamics. Overfitting occurs despite continued compression.




COMPRESSION IN BATCH GRADIENT DESCENT AND SGD

Theoretical Claim :

A theoretical claim of the information bottleneck theory of deep learning, namely that randomness in stochastic gradient descent is responsible for the compression phase. This is because the choice of input samples in SGD is random, the weights evolve in a stochastic way during training.

- Here we distinguish two phases of SGD optimization: in the first “drift”(generalization) phase and “diffusion”(Compression) phase.
- Under the small training error, the weights drawn from this stationary distribution will maximize the entropy of inputs given hidden layer activity, $H(X|T)$
- The result of the diffusion dynamics will be to minimize $I(X; T) := H(X) - H(X|T)$ for a given value of $I(T; Y)$ reached at the end of the drift phase.


Note: Assumption is the the constraint on the training error are equivalent to constraint on mutual information.



But the above doesn't hold up theoretically and empirically

There is no general reason that a given set of weights sampled from this distribution (i.e., the weight parameters found in one particular training run) will maximize $H(X | T)$, the entropy of inputs given hidden layer activity.

In particular, $H(X | T)$ reflects (conditional) uncertainty about inputs drawn from the data-generating distribution, rather than uncertainty about any kind of distribution across different training runs.



Now after showing the compression phase doesn't always drive the distribution of the weight to maximum entropy distribution subject to training error constraint, Now we will show that the empirically that the stochasticity of the SGD is not necessary for compression.

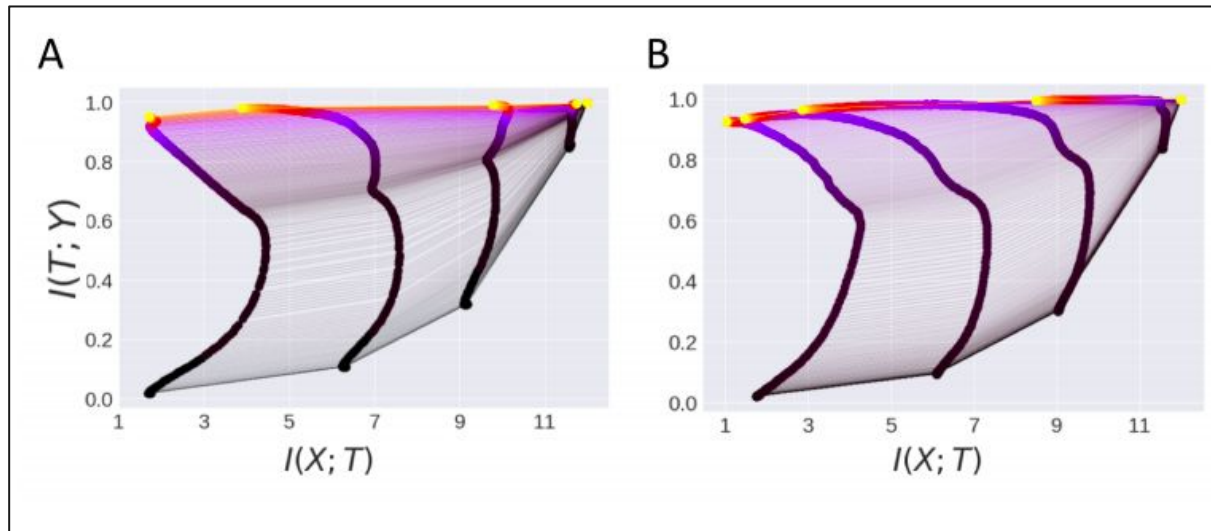
Now, we will prove our claim by the following the experiment shown below

We will take two distinct training procedure here :

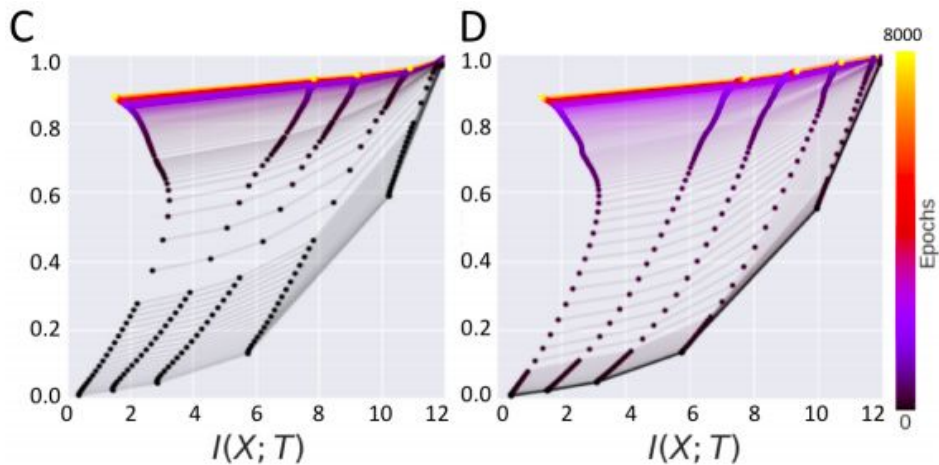
Offline stochastic gradient descent (SGD): It learns from a fixed-size dataset, and updates weights by repeatedly sampling a single example from the dataset and calculating the gradient of the error with respect to that single sample (the typical procedure used in practice)

Batch gradient descent (BGD): which learns from a fixed-size dataset, and updates weights using the gradient of the total error across all examples.

Note: Batch gradient descent uses the full training dataset and, crucially, therefore has no randomness or diffusion-like behavior in its updates.



- Now we apply the tanh network for both methods we can find that the robust compression in the both methods.
- Thus we can say that the randomness in the training process does not appear to contribute substantially to compression of information about the input.



- Now we apply the ReLU network for both methods we can find that the robust compression in the both methods.
- Thus we can say that the randomness in the training process does not appear to contribute substantially to compression of information about the input.
- Note that compression arises predominantly from the double saturating nonlinearity.

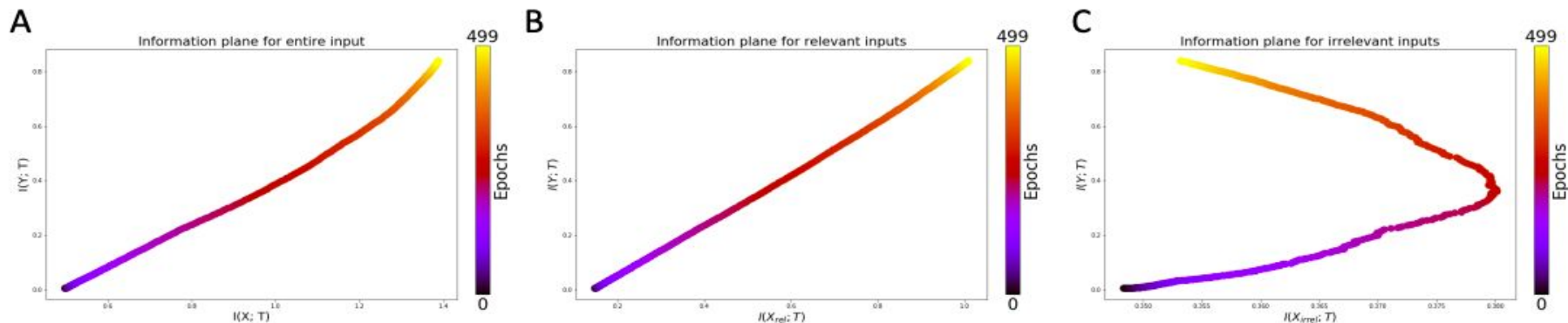


Simultaneous Fitting and Compression

The authors claim that generalization can occur without compression. In order to prove this, we consider a student-teacher setup. We partition the input X into a set of task-relevant inputs X_{rel} and a set of task-irrelevant inputs X_{irrel} , and alter the teacher network so that the teacher's weights to the task-irrelevant inputs are all zero. Hence the inputs X_{irrel} contribute only noise, while the X_{rel} contain signal. We then calculate the information plane dynamics for the whole layer, and for the task-relevant and task-irrelevant inputs separately.

Simultaneous Fitting and Compression

While the overall dynamics show no compression phase, the information specifically about the task-irrelevant subspace does compress over the course of training. This compression process occurs at the same time as the fitting to the task-relevant information





Conclusions

The results suggest that

1. Compression dynamics in the information plane are not a general feature of deep networks, but are critically influenced by the nonlinearities employed by the network.
2. Double saturating nonlinearities lead to compression, if mutual information is estimated by binning activations or by adding homoscedastic noise, while single-sided saturating nonlinearities like ReLUs do not compress in general.
3. Stochasticity in the training process does not contribute to compression in the cases we investigate.
4. Compression still may occur within a subset of the input dimensions if the task demands it. This compression, however, is interleaved rather than in a secondary phase.
5. The results suggests that this may not be the best scheme to link IB theory and DL Networks, however there is a possibility for finding other ways to use IB theory to devise fundamentally new training mechanisms that are inherently stochastic and where compression is explicitly encouraged with appropriate regularization terms



Analysis and Critiques

Authors have countered the claim of Tishby, stating that this compression phenomenon in DNNs is not comprehensive, and it depends on the particular activation function.

In particular, they claimed that the compression does not happen with ReLu activation functions.

Tishby disputed these claims, arguing that Saxe et al had not observed compression due to weak estimates of the mutual information [3].



References

[On the Information Bottleneck Theory of Deep Learning](#)

[Opening the Black Box of Deep Neural Networks via Information](#)

[OpenReview Forum](#)



Thank You!

Q & A