Information Bottleneck Methods on Convolutional Neural Networks

Junjie Li[1] and Ding Liu[1,*]

[1] *School of Computer Science and Technology, Tiangong University, Tianjin 300387, China*

Recent year, many researches attempt to open the black box of deep neural networks and propose a various of theories to understand it. Among them, information bottleneck theory (IB) claims that there are two distinct phases consisting of fitting phase and compression phase in the course of training. This statement attracts many attentions since its success in explaining the inner behavior of feedforward neural networks. In this paper, we employ IB theory to understand the dynamic behavior of convolutional neural networks (CNNs) and investigate how the fundamental features have impact on the performance of CNNs. In particular, through a series of experimental analysis on benchmark of MNIST and Fashion-MNIST, we demonstrate that the compression phase is not observed in all these cases. This show us the CNNs have a rather complicated behavior than feedforward neural networks.

I. INTRODUCTION.

In recent, the practical successes of deep neural networks have generated many attempts to explain the performance of deep learning [7, 14], especially in terms of the dynamics of the optimization [8, 9]. In this context, the information bottleneck (IB) theory provides a fundamental tool on this topic, and some preliminary empirical exploration of these ideas in deep feedforward neural networks has yielded striking findings [1, 6, 17]. Based on all these works, we investigate the IB theory using a analytical methods on Convolutional Neural Networks (CNNs) and observe quite different behaviours in contrast to those of feedforward neural networks.

In the series of original works [1, 5, 6], authors hold some core points that the distinct phases of the SGD optimization, drift and diffusion, which explain the empirical error minimization and the representation compression trajectories of the layers. These phases are characterized by very different signal to noise ratios of the stochastic gradients in every layer. This funding opens the black box of deep learning from the perspective of information theory and draw many attentions. Along this way, a further research offers some different views of IB theory and shows us some different behaviors on feedforward neural networks [2]. They say that fitting and compression phases in the course of training strongly depends on the nonlinear activation. The authors state that double saturating nonlinearities lead to compression and stochasticity in the training phase does not contribute to compression. Obviously, it is partly in contradiction with the initial idea in Ref. [1]. Moreover, Ref. [15] claim that the compression can happen even when using ReLu activation in their high dimensional experiments, and there is not a clear link between compression and generalization. Then lately, some works start to focus on exploring the inner organization of CNNs and autoencoders by using matrix-based Renyi's entropy [10, 16]. The authors propose that variability in the compression behavior is strongly depends on different estimators. By using matrix-based Renyi's entropy estimator and remove the redundant information in the MI, they observe compression phase during the training. Moreover, Ref. [18] find a new phenomenon – clustering emerging in the training phase. And they propose that the compression strongly rely on the clustering and may not causally related to generalization. So until now, based on all these previous works, compression and the relationship between it and generalization still remain elusive.

In this paper, different from the previous works, we observe no compression phase both on convolution layers and fully connected layers on standard CNNs, even with double saturating nonlinearity such as *tanh*. This observation partly supports the conclusion by Ref.[2] that compression is not the universal phase during the course of training. Moreover, from the perspective of IB theory, we investigate how the fundamental features such as convolutional layer width, network depth, kernel size, pooling layers etc. have an effect on the performance of CNNs. The experimental results verify the importance of these features in improving the generalization performence.

II. METHOD

The Information Bottleneck (IB) theory is introduced by Tishby et.al first time in the paper [5]. Afterwards, Ref. [6] [1] analyse the training phase of DNNs from the perspective of IB. Accordingly, IB suggests that each hidden layer will capture more useful information from the input variable, and the hidden layers are supposed to be the maximally compressed mappings of the input. There are several fundamental points to know about IB theory as follow:

A. Mutual Information

Mutual Information (MI) measures the mutual dependence of two random variables. Further, it quantifies

* dingliu_thu@126.com

the amount of information got about one random variable through observing the other. For example, given two variables A and B, mutual information $I(A;B)$ is defined as:

$$I(A;B) = H(A) - H(A|B) \tag{1}$$

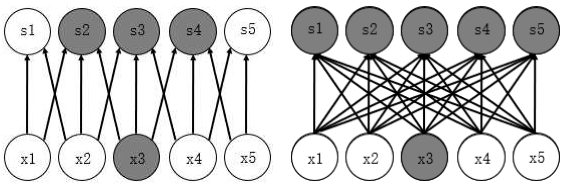
$$H(A) = -\sum_{a\in A} p(a)\log p(a) \tag{2}$$

$$\begin{aligned} H(A|B) &= -\sum_{b\in B} p(b) \sum_{a\in A} p(a|b)\log p(a|b) \\ &= -\sum_{a\in A}\sum_{b\in B} p(b,a)\log p(a|b) \end{aligned} \tag{3}$$

where $H(A)$ and $H(A|B)$ are entropy and conditional entropy respectively, and $p(b,a)$ denotes joint probability distribution.

B. Binning-based MI Estimator

The binning-based MI estimator is widely used in feedforward neural networks. And as we know, CNNs are characterized as sparse interactions (sparse connectivity) in compare with feedforward neural networks as shown in Fig. 1, which appears in Ref. [13]. In principle, the sparsity will not lead to the failure of binning-based estimator. So along this way, we also use it to evaluate the MI in CNNs. First, we reshape the output images of each channel of each convolutional layer into a vector, and splice these vectors into a long one h. Then, according to Ref. [2], we discretize the activation output by a fixed bin size, i.e. $T = bin(h)$. (Ref. [2] choose 0.5, while in this paper, we use the constant 0.67 as bin size. Because according to our experimental results, it is good for visualization and meets the results of kernel density estimation method in [11, 12].) We show the process in Fig. 2.



(a)sparse connectivity (b)dense connectivity

FIG. 1. Sparse connectivity and dense connectivity (fully connected) .

In this case, we use the fact that $H(T|X) = 0$. Then

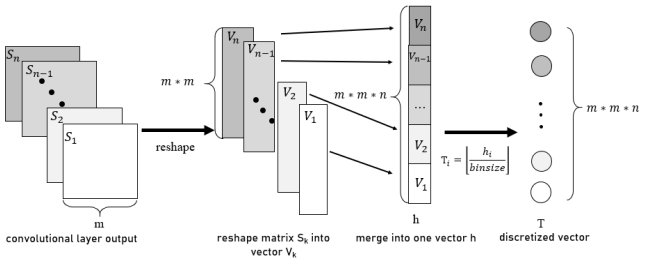


FIG. 2. The process of binning activity. Each channel of layer's output is firstly converted into a vector, and then these vectors are combined into a long one, i.e. h. Then, it is mapped into a new vector by dividing by the constant bin size.

$I(T;X)$ and $I(T;Y)$ can be rewritten respectively as:

$$\begin{aligned} I(T;X) &= H(T) - H(T|X) \\ &= H(T) \\ &= -\sum_{i=1}^{n} p_i \log p_i \end{aligned} \tag{4}$$

$$\begin{aligned} I(T;Y) &= H(T) - H(T|Y) \\ &= -\sum_{i=1}^{n} p_i \log p_i + \sum_{i=1}^{n} p_i H(T|Y=y_i) \end{aligned} \tag{5}$$

where p_i is the probability that a activation output lands in the ith interval.

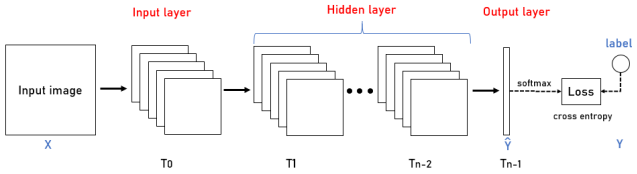
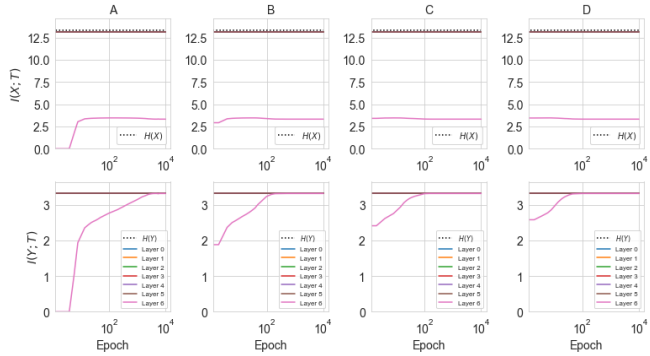


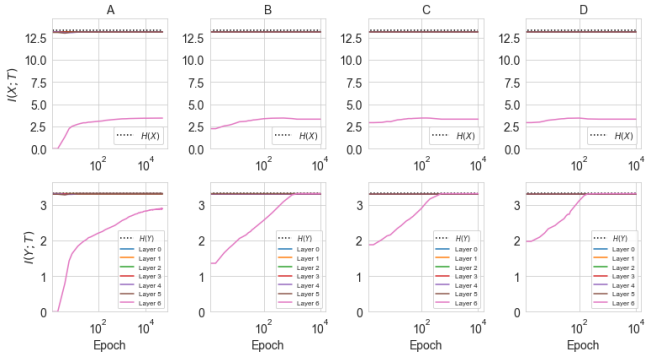
FIG. 3. Architecture of our convolutional network model. T_0 is input layer. $T_0 \sim T_{n-2}$ are convolutional layers. T_{n-1} is fully connected output layer.

C. Information Plane And Data Processing Inequality

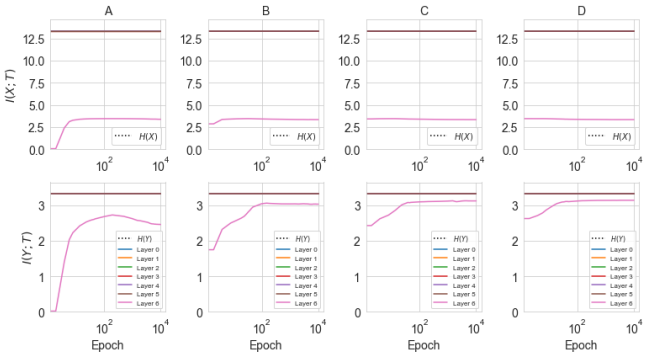
Information plane (IP) shows the dynamic behaviour of $I(Y;T)$ with respect to $I(X;T)$ [1]. They propose that the optimization of feedforward neural networks involve two phases, namely fitting phase and compression phase. In the fitting phase, the feedforward neural networks try to fit training samples into corresponding labels by increasing both $I(X;T)$ and $I(Y;T)$. In com-



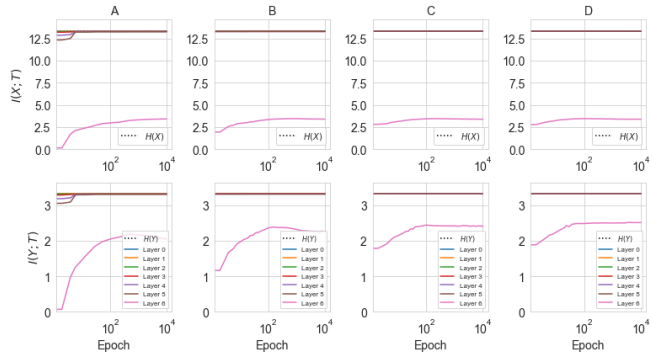
(a)MI path on training data of MNIST



(b)MI path on training data of Fashion-MNIST



(c)MI path on test data of MNIST



(d)MI path on test data of Fashion-MNIST

FIG. 4. (Colored online) MI path on CNNs with different convolutional layer widths. Colored lines mark each layer of network. As we describe in *Experiment* section, layer $0 \sim 5$ are convolutional layers, and layer 6 is final output layer (fully connected layer). Convolutional layer 5 (the final convolutional layer) covers all previous layers, $H(X)$ and $H(Y)$ since they all have the same value. The convoluntional layer width of 4 networks are (A) 1-1-1-1-1-1, (B) 3-3-3-3-3-3, (C) 6-6-6-6-6-6, (D) 12-12-12-12-12-12. The pink line represents the mutual information of final output layer, which grows until getting stable as the process of training.

pression phase, the feedforward neural networks discard redundant information by reducing $I(X;T)$. Based on these statements, people will observe these two apparent phases on IP.

Moreover, the MI among all feedforward neural networks layers form a Markov chain, which leads to Data Processing Inequality (DPI). It can be depicted as:

$$\begin{aligned} H(X) \geq I(X;T_0) \cdots I(X;T_{n-2}) \geq I(X;T_{n-1}) \\ H(Y) \geq I(Y;T_0) \cdots I(Y;T_{n-2}) \geq I(Y;T_{n-1}) \end{aligned} \tag{6}$$

where T_0 is input layer, $T_1 \cdots T_{n-2}$ are hidden layers, and T_{n-1} denotes the final output layer.

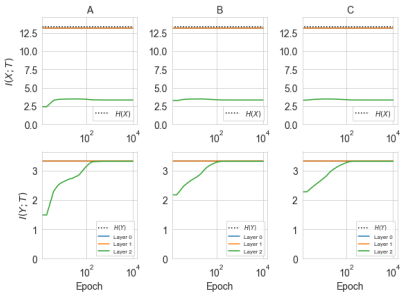
III. EXPERIMENTS

In order to investigate the impact of IB theory on CNNs , we perform a series of experiments on MNIST and Fashion-MNIST datasets. Our code is available on the Github [19]. For simplicity, we select 10,000 training samples randomly as training dataset and 10,000 test samples as test dataset. The networks are trained by using Adam algorithm and cross-entropy loss function with batch of 1000 samples. In addition, we set the learning rate as 10^{-3}, and use *tanh* activation except for final output layer with softmax. Our model is shown in Fig3. The MI is evaluated on both training dataset and test dataset respectively. Therefore, in this case, $H(X)$ for both training dataset and test dataset equals to $\log_2 10^4$. Then, we analyse the impact of some crucial features such as convolutional layer width, network depth, kernel size and pooling layer on CNNs from view point of IB theory. Specifically, we discuss the compression phase on CNNs architecture.

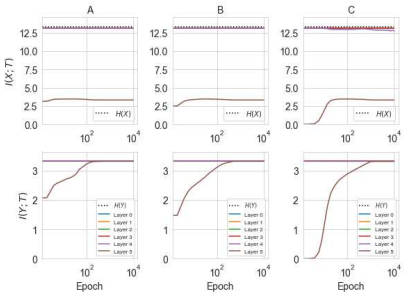
A. Convolutional Layer Width

The convolutional layer width is crucial on the way to understand representation power of neural networks. To study the effect of convolutional layer width from the perspective of IB theory, we train 4 different CNNs with various of convolutional layer widths (number of channels).

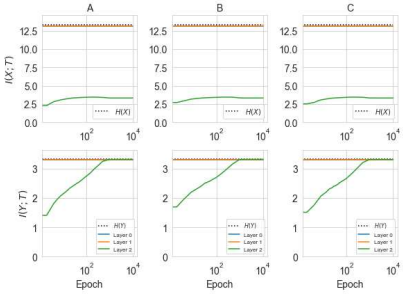
Fig 4 shows the $I(Y;T)$ and $I(X;T)$ paths on these



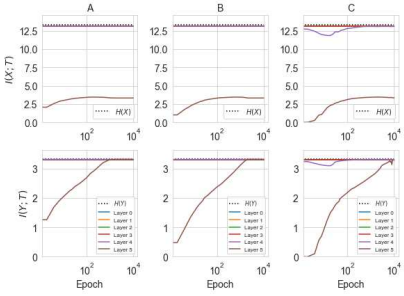
(a)MI path on training data of MNIST. All networks with fixed depth = 3. The sizes of kernel are (A) 3 x 3 (B) 7 x 7 (C) 11 x 11



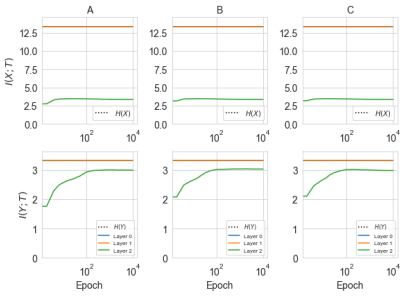
(b)MI path on training data of MNIST. All networks with fixed depth = 6. The sizes of kernel are (A) 3 x 3 (B) 5 x 5 (C) 7 x 7



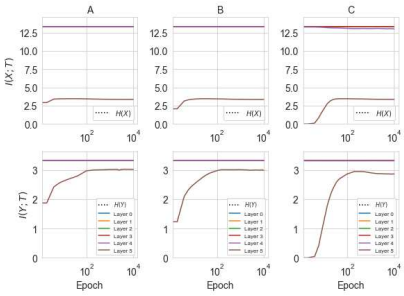
(c)MI path on training data of Fashion-MNIST. All networks with fixed depth = 3. The sizes of kernel are (A) 3 x 3 (B) 7 x 7 (C) 11 x 11



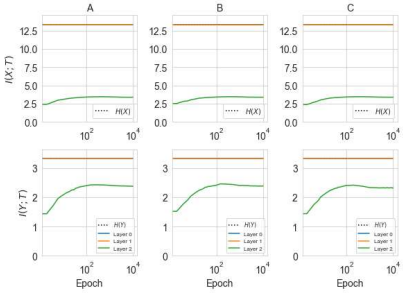
(d) MI path on training data of FASHION-NNIST. All networks with fixed depth = 6. The sizes of kernel are (A) 3 x 3 (B) 5 x 5 (C) 7 x 7



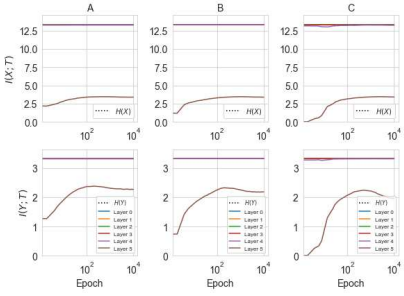
(e)MI path on test data of MNIST. All networks with with fixed depth = 3. The sizes of kernel are (A) 3 x 3 (B) 7 x 7 (C) 11 x 11



(f)MI path on test data of MNIST. All networks with fixed depth size = 6. The sizes of kernel are (A) 3 x 3 (B) 5 x 5 (C) 7 x 7

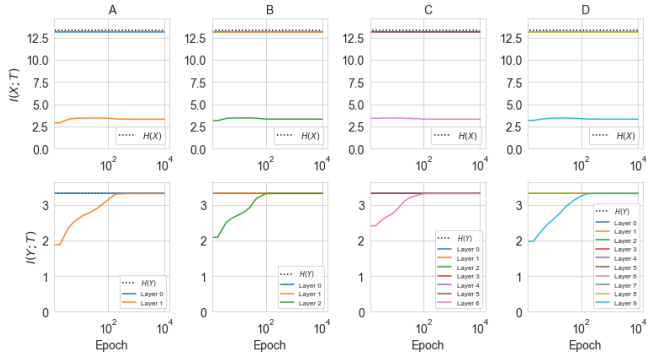


(g)MI path on test data of Fashion-MNIST. All networks with fixed depth = 3. The sizes of kernel are (A) 3 x 3 (B) 7 x 7 (C) 11 x 11

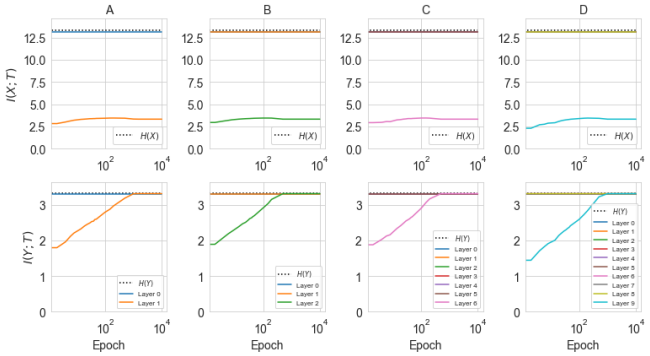


(h)MI path on test data of Fashion-MNIST. All networks with fixed depth = 6. The sizes of kernel are (A) 3 x 3 (B) 5 x 5 (C) 7 x 7

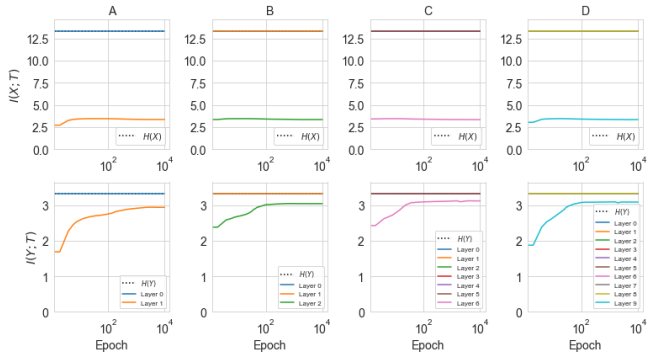
FIG. 5. (Colored online) MI path on CNNs with different convolutional kernel sizes.



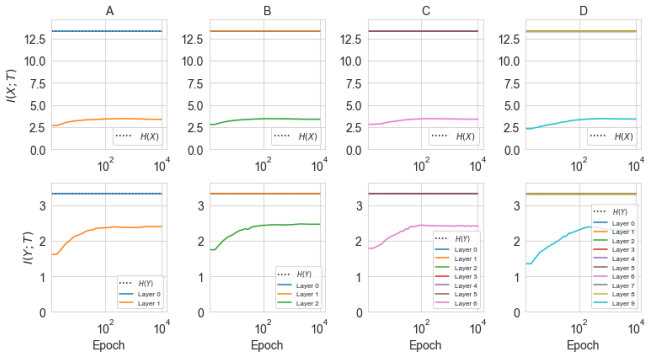
(a)MI path on training data of MNIST. (A) depth=2, (B) depth=3, (C) depth=7, (D) depth=10



(b)MI path on training data of Fashion-MNIST. (A) depth=2, (B) depth=3, (C) depth=7, (D) depth=10



(c)MI path on test data of MNIST. (A) depth=2, (B) depth=3, (C) depth=7, (D) depth=10



(d)MI path on test data of Fashion-MNIST. (A) depth=2, (B) depth=3, (C) depth=7, (D) depth=10

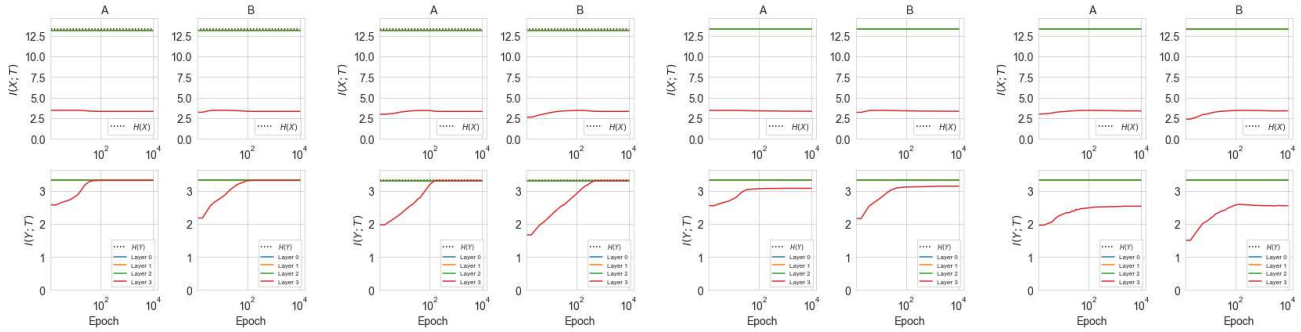
FIG. 6. (Colored online) MI path on CNNs with different depths.

networks during the training and test phase. By using DPI we introduce earlier, the theoretical upper bound of $I(X;T)$ of each layer is $H(X)$. Similarly, the theoretical upper bound of $I(Y;T)$ equals to $H(Y)$. Therefore, in this figure, we observe the MI on all convolutional layers reach the upper bound which means they capture almost all information on input X and label Y. This is due to we treat the whole image as a single variable, then all images are basically different. So according to Eq. 4, $I(X;T)$ can be represented by $H(T)$. Moreover, $H(T)$ is equal to $\log_2 10^4$ i.e. $H(X)$. In the same view, $I(Y;T)$ on convolutional layer is closely equal to $H(Y)$.

For final output layer, the starting value of $I(X;T)$ and $I(Y;T)$ increase apparently with the expending of width. And also, with wider convolutional layer, the model reach the upper bound faster. So wide CNNs can perform better with less training epochs. Specifically, in panel (c) and (d), we observe larger maximum value of $I(Y;T)$ for final output layer with increasing of width. Base on these observations, we believe that wide network is capable of capturing more information, which is beneficial to have better generalization.

B. Kernel Size and Network Depth

Ref.[3] points out network with a large kernel size can be replaced by a deep network with small kernel size. From the perspective of information theory, how does the kernel size and depth affect MI in CNNs? We evaluate MI with various choices of depth and kernel size. By comparing information paths in Fig5 and Fig6, we find that both larger kernel size and deeper network can promote the starting value of $I(X;T)$ and $I(Y;T)$ on the final output layer, which implies network capture more information with less training. However, if we continuously increase kernel size or depth , the starting point cannot increase anymore or even becomes worse. So we propose that the larger kernel size and depth can drive network capturing more information with less training epochs. But over-large kernel size and depth will need more training epochs to capture the same amount of information. Furthermore, unlike convolutional layer width, in Fig.5 (e), (f), (g), (h) as well as in Fig.6 (c) and (d), we observe that they all reach the same maximum value of MI for final output layer, which implies that a small kernel size and shallow depth are good enough to have a better generalization performance in these simple cases.



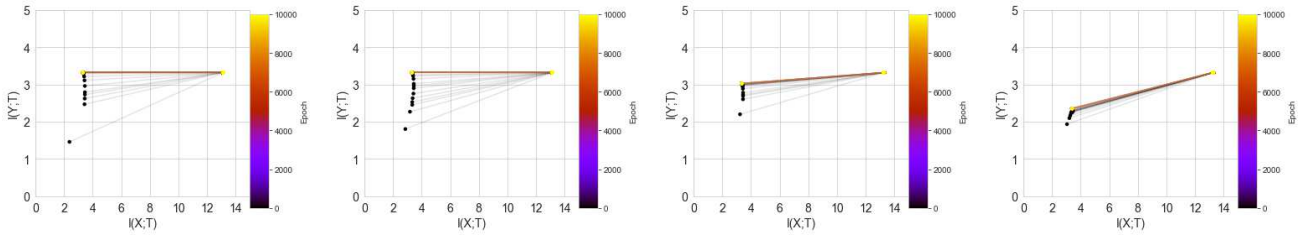
(a)MI on training data of MNIST. (A) without pooling layer (B) with pooling layer

(b)MI on training data of Fashion-MNIST. (A) without pooling layer (B) with pooling layer

(c)MI on test data of MNIST. (A) without pooling layer (B) with pooling layer

(d)MI on test data of Fashion-MNIST. (A) without pooling layer (B) with pooling layer

FIG. 7. (Colored online) MI path on CNNs with pooling layer.



(a)MI on training data of MNIST

(b)MI on training data of Fashion-MNIST

(c)MI on test data of MNIST

(d)MI on test data of Fashion-MNIST

FIG. 8. Information plane for multi-fully connected layers on CNNs.

C. Pooling Layers and Multi-Fully Connected Layers

CNNs almost always include some forms of pooling layer such as max pooling and average pooling etc. The pooling layer always discards parts of data in order to improve the generalization and reduce computational complexity. In order to investigate the role of the pooling layer, we take the max pooling as example. In Fig 7, we observe that the curves of MI grow in different ways. Panel (c) and (d) show the networks with pooling layer reach a little bit larger value than networks without pooling layer, which implies that pooling layer is beneficial to have better generalization.

We also design different CNNs with multi-fully connected layers to study whether double-sided saturating nonlinearities like *tanh* yield compression phase in CNNs. (Ref. [2] propose that double-sided saturating nonlinearities yield a compression phase while linear activation function can not). So, we use a network with 5 convolutional layers (convolutional layer width=3-3-3-3-3 and kernel size=3-3-3-3-3) and 4 fully connected layers (500-1024-500-10). Fig 8 shows these layers information plane (IP) paths. From this we observe that all convolutional layers and fully connected layers (except for final output fully connected layers) converge to a point. In addition, the MI of final output layers grows during both training phase and test phase. So, we observe there are no compression phase occurs in the CNNs.

IV. DISCUSSION

Information bottleneck theory provides a interesting analytic tool to explore the inner behaviour of deep neural network, and based on this, people try to understand why deep learning works well. Along this way, this paper try to extend the study to CNNs and investigate how the fundamental features have impact on the performance of CNNs. Based on our cases, we summarize some key observations and draw conclusions as:

1. Convolutional layers can capture almost all information on input towards label. The MI between convolutional layers and input/output keep close to their upper bound.

2. Wide convolutional layers network is able to improve the generalization performance. Furthermore, wide network need less training epochs to reach its optimal performance than narrow one.

3. In general case, larger kernel size and deeper architecture drive network capturing more information with less training epochs. But, a over-large kernel size and over deep architecture will need much more training epochs to capture the same amount of information. This

implies that people should balance the kernel size and network depth while design the deep architecture. Furthermore, it shows us the extremely deep neural network is probably not the right way to do deep learning.

4. In some simple cases, our results also reveal that there is no compression whether in convolutional layers or fully connected layers, even using double-sided saturating nonlinearities in CNNs. Hence, we tend to think the compression probably happen in some specific cases, but not a universal mechanism in deep learning, and further, the relationship between it and generalization needs more experimental verification.

In the future work, we plan to verify the above conclusions on some more complicated datasets such as ImageNet, and on some more complicated deep architectures such as Generative Adversarial Networks (GANs). We believe it will provide more experimental evidence to verify the IB theory and help us to understand deep learning.

V. ACKNOWLEDGMENTS

Our research is supported by the National Natural Science Key Foundation of China (61433015), the Science & Technology Development Fund of Tianjin Education Commission for Higher Education (2018KJ217).

[1] Shwartz-Ziv, Ravid and Tishby, Naftali, Opening the black box of deep neural networks via information, arXiv preprint arXiv:1703.00810, 2017.
[2] Saxe, Andrew Michael and Bansal, Yamini and Dapello, Joel et.al, On the information bottleneck theory of deep learning, 2018.
[3] Szegedy, Christian and Vanhoucke, Vincent and Ioffe, Sergey et.al, Rethinking the inception architecture for computer vision, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, 2818–2826.
[4] Cao, Xudong, A practical theory for designing very deep convolutional neural networks, Technical Report, 2015.
[5] Tishby, Naftali and Pereira, Fernando C and Bialek, William et.al., The information bottleneck method, arXiv preprint physics/0004057, 2000.
[6] Tishby, Naftali and Zaslavsky, Noga, Deep learning and the information bottleneck principle, 2015 IEEE Information Theory Workshop (ITW), IEEE, 2015:1–5.
[7] Kadmon, Jonathan and Sompolinsky, Haim, Optimal architectures in a solvable model of deep networks, Advances in Neural Information Processing Systems, 2016:4781–4789.
[8] Saxe, Andrew M and McClelland, James L and Ganguli, Surya, Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, arXiv preprint arXiv:1312.6120, 2013.
[9] Advani, Madhu S and Saxe, Andrew M, High-dimensional dynamics of generalization error in neural networks, arXiv preprint arXiv:1710.03667, 2017.
[10] Yu, Shujian and Wickstrøm, Kristoffer and Jenssen, Robert et.al., Understanding Convolutional Neural Networks with Information Theory: An Initial Exploration, arXiv preprint arXiv:1804.06537, 2018.
[11] Kolchinsky, Artemy and Tracey, Brendan, Estimating mixture entropy with pairwise distances, Entropy, 2017.
[12] Kolchinsky, Artemy and Tracey, Brendan D and Wolpert, David H, Estimating mixture entropy with pairwise distances, Entropy, 2017.
[13] Goodfellow, Ian and Bengio, Yoshua and Courville, Aaron, Deep learning, MIT press, 2016.
[14] Guidotti, Riccardo and Monreale, Anna and Ruggieri, Salvatore et.al, A survey of methods for explaining black box models, ACM, 2019.
[15] Gabrié, Marylou and Manoel, Andre and Luneau, Clément et.al, Entropy and mutual information in models of deep neural networks, Advances in Neural Information Processing Systems, 2018.
[16] Yu, Shujian and Principe, Jose C, Understanding autoencoders with information theoretic concepts, Neural Networks, Elsevier, 2019.
[17] Amjad, Rana Ali and Geiger, Bernhard Claus, Learning representations for neural network-based classification using the information bottleneck principle, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE, 2019.
[18] Goldfeld, Ziv and Van Den Berg, Ewout and Greenewald, Kristjan, Estimating information flow in deep neural networks, Proceedings of the 36th International Conference on Machine Learning, 2019.
[19] https://github.com/mrJayLee/IB_ON_CNN.