Channel Capacity of Neural Networks

Gen Ye[1] and Tong Lin[1,2](✉)

[1] The Key Laboratory of Machine Perception(MOE), School of EECS, Peking University, Beijing, China
[2] Peng Cheng Laboratory, Shenzhen, China
{yegen,lintong}@pku.edu.cn

Abstract. Occam's Razor principle suggests preference for simpler models and triggers an enduring question: what is the proper definition of complexity of a model? In this work, we regard neural networks as communication channels and measure the complexity of neural networks by means of their channel capacity—the maximum information reserved in the output of a neural network. Furthermore, we show a connection between the L2-norm of the weight matrix of the linear model and its channel capacity through the singular values of the weight matrix. On image classification problems, we find regularizing different neural networks by constraining their channel capacity effectively boosts the generalization performance and outperforms other information-theoretic regularization methods.

Keywords: Neural network · Channel capacity · Regularization method.

1 Introduction

Inductive biases are necessary for machine learning and one of the most famous examples would be *Occam's Razor*: the simplest explanation is best. From the perspective of deep learning, a simpler model may refer to a network with fewer parameters/connections [17], a network with smaller norm [16], or a network with shorter minimal description length [12]. Following the simple intuition that the output of a simple model should contain fewer information about the input, in this work we regard neural networks or their modules/layers as communication channels and propose to view the channel capacity as a measure of model complexity.

In information theory [5], communication means messages/data/signals, denoted as a random variable W, successively get through an encoder, a communication channel, and a decoder. Analogous to this communication process, an input to a neural network is an instantiation of a class label, and after representation learning through a neural network (corresponding to a communication channel), the recovered label is obtained by a classifier (corresponding to a decoder).

This work was supported by NSFC Tianyuan Fund for Mathematics (No. 12026606), and National Key R&D Program of China (No. 2018AAA0100300).

The channel capacity is defined as the maximum mutual information between channel input X and channel output Y, maximized over all possible input distributions. In this work, we define two versions of **information complexity (IC)** of neural networks. **Maximum information complexity (MIC)** of a neural network means the channel capacity of the corresponding communication channel. For boundness of this quantity, we constrain the possible distributions of the input to be those from a certain subset of absolutely continuous distributions (corresponding to a continuous random variable) and assume an extra noise added to the output. By assuming that the input distribution itself is drawn from a measure space and then changing the maximization over all distributions to expectation, we get the second version, called **expected information complexity (EIC)**, of neural networks for practical use.

It needs to be clear that analysis of **mutual information (MI)** between the input and the representation is nothing new for machine learning. **InfoMax principle** [18] argues that the goal of representation learning should be to learn a representation $Z = g(X)$ such that the MI $I(X, g(X)$ is maximized. Some recent state-of-the-art self-supervised learning methods [24, 11] aim at maximizing the MI between features of different views of the input, while this objective can be treated as a lower bound of the InfoMax objective [27]. **Information bottleneck principle (IB)** [25] suggests that supervised learning should attempt to learn a representation Z being maximally expressive about the label Y while being maximally compressive about the input X, which has been recently rephrased in the context of deep learning [26]. The learning objective of IB is to minimize $L(p_\theta(z|x)) = I(Z, X) - \beta I(Z, Y)$, where model parameter θ belongs to a condition distribution and β controls the tradeoff. Following this principle, a thread of research [22, 21] analyzes the training of neural networks using information planes and discusses the relationship between generalization and compression. Also IB can be directly applied to train neural networks [15, 1]. Hafez-Kolahi and Kasaei [9] provides a survey about IB and its applications.

Previous works mostly draw lessons from data compression of information theory, whereas this work treats neural networks as communication channels. To verify the usefulness of this perspective, we train neural networks regularized with lower information complexity, supposing that a neural network with lower channel capacity can convey more task-relative information when it fits training data equally well as other models.

Our main contributions are as follows:

- We formally define two kinds of new complexity measures for neural networks.
- We design two regularization methods by penalizing the neural network of high channel capacity during training.
- Experiments on various settings show that our methods do improve classification performance and outperform other information-theoretic regularization methods.

2 Related Work

MacKay [19] viewed neural network learning as communication and fostered research on the capacity of a single neuron. Parameters of a neural network are treated as a message in [19] while corresponding to a communication channel in our work. Foggo and Yu [7] investigated the maximum MI $\sup_\theta I(X, Z_\theta)$, where θ is the set of parameters of a neural network, for various neural network architectures. In this work, the channel capacity of neural networks is also the maximum MI $\max_{p_X} I(X, Z)$, however, with respect to all possible input distributions.

Our work involves the calculation of MI between high dimensional random vectors, which is a notoriously hard problem. Classic techniques [8, 28] proposed to estimate MI through samples are hard to scale up to dimensionality encountered in deep learning. To overcome the difficulty, recent works [1, 3] develop MI estimators for DNNs with different variational bounds of MI.

In the context of supervised learning, our work is related to recently proposed information-theoretic regularization methods [23, 1, 20]. Szegedy et al. [23] tried to smooth the label to prevent models from assigning full probability to each training example. Similar to label smoothing, confidence penalty method of Pereyra et al. [20] proposed to regularize networks with an extra loss term, which penalizes networks for having low entropy predictive distributions. Alemi et al. [1] attempted to regularize neural networks by constraining the MI between the input and the representation.

3 Information Complexity of Neural Networks

We first introduce some concepts from information theory. We denote random vectors with capital letters such as X. All the logarithms in this paper are in base e. In this work, we use $\mathcal{AC}^n$ to denote the set of all n-dimensional continuous random vectors and $\mathcal{PD}^n$ to denote the set of all **probability density function (pdf)** of n-dimensional continuous random vectors. Beside, we use $X \sim \mathcal{N}(m, K)$ to indicate that X is a gaussian random vector with mean m and covariance matrix K. The differential entropy of the random vector X is $h(X) = -\int p(x) \log p(x) dx$, where $p(x)$ is the pdf of X. The mutual information between random vectors X and Y is $I(X, Y) = \int p(x, y) \log \frac{p(x,y)}{p(x)p(y)} dxdy$.

3.1 Information Complexity of Neural Networks

In this subsection, we give formal definitions of two version of information complexity of neural networks. In information theory, a discrete-time memoryless communication channel corresponds to a regular conditional probability mathematically, which can be used to construct a joint distribution of the input and the output. For a set $\mathcal{A}$ of all possible distributions of input, the **channel capacity** of a communication channel [5] is: $C = \max_{p_X \in \mathcal{A}} I(X, Y)$, where X is the input and Y is the output of the channel. Capacity describes the ability of the channel

to transport information, which is also intuitively important to understand neural networks. Treating a neural networks as a communication channel, we need to explain how the neural network defines a regular conditional probability and what the set of possible input distributions is.

Let a n-dimensional random vector $X \in \mathcal{AC}^n$ be the input of a neural network. Let θ be the parameter of the neural network, corresponding a continuous function $f_\theta(x) : \mathbb{R}^n \to \mathbb{R}^m$. Let the m-dimensional random vector Y be the output of the neural network. In this work, we assume the output $Y(X, \theta, \varepsilon) = f_\theta(X) + \varepsilon$ is a function of the random vector X, parameter of the network θ and an extra noise ε.

Add noises to the output of neural networks. A straightforward idea making connections between a neural network and a communication channel is to define $Y = f_\theta(X)$. Because $f_\theta(x)$ is continuous, it's easy to prove that this model relates to a communication channel. In this work, we analyze neural networks with extra noise $\varepsilon \neq 0$. When the distribution of the input is absolutely continuous with respect to some convex subset of $\mathbb{R}^n$, MI between the input and the representation is infinity for some most practical deterministic neural networks and almost every choice of weight matrices [2]. To avoid the infinity of MI, we add an extra noise $\varepsilon \in \mathcal{AC}^m$ to the output of a neural network following the suggestion of [2].

Limit the set of possible distributions of the input. It's a well-known fact [5] that the channel capacity of a gaussian channel is infinite if we don't constrain the signal-to-noise rate (SNR). In this work, we follow the maximum-input-power constraint from information theory and define the allowable input pdfs as those from $\mathcal{A}_P^n = \{p(x) \,\big|\, p \in \mathcal{PD}^n, \int_{\mathbb{R}^n} x^T x p(x) dx \leq P\}$. This constraint is reasonable because the input of neural networks often takes values in intervals, e.g., $[0, 1]$.

Suppose the pdf of a fixed noise is denoted as $p_\varepsilon(\cdot)$. The joint distribution of X and Y is totally determined by the pdf of input p_X and the parameter of neural network θ, which corresponds to a conditional pdf of Y given X, $p_\theta(y|x) = p_\varepsilon(y - f_\theta(x))$. Consequently, the MI between X and Y is determined purely by p_X and θ, which can written as:

$$\begin{aligned} I(X,Y) &= \int_{\mathbb{R}^n \times \mathbb{R}^m} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy \\ &= \int_{\mathbb{R}^n \times \mathbb{R}^m} p_\theta(y|x)p(x) \log \frac{p_\theta(y|x)}{p(y)} dxdy \\ &= \int_{\mathbb{R}^n \times \mathbb{R}^m} p_\theta(y|x)p(x) \log \frac{p_\theta(y|x)}{\int_{\mathbb{R}^n} p(x)p_\theta(y|x)dx} dxdy. \end{aligned}$$

Therefore we also write $MI(p_X, \theta) = I(X, Y)$ to denote MI between the input and the output. Beside a fixed noise, we restrict possible input random vectors X to those correspond to pdfs $p_X \in \mathcal{A}_P^n$. Now we are ready to define the maximum information complexity (MIC) of neural networks.

Definition 1 (Maximum information complexity)**.** *Given a random vector $\varepsilon \in \mathcal{AC}^m$ and a neural network $f_\theta(x) : \mathbb{R}^n \to \mathbb{R}^m$, the maximum information*

complexity with noise ε of the neural network is:

$$MIC_\varepsilon(\theta) := \max_{p_X \in \mathcal{A}_P^n} MI(p_X, \theta).$$

Calculation of MIC requires solving an optimization problem over a function space, which makes it difficult to be used in practice. In order to overcome this difficulty, we define a similar concept called expected information complexity. Let $\mathcal{A}$ be the set of all possible pdfs of input random vectors. Assuming $\mathcal{A}$ belongs to a probability space $(\mathcal{A}, \mathcal{F}_\mathcal{A}, \mathcal{P}_\mathcal{A})$, we can define expected information complexity of neural networks in a general way:

Definition 2 (Expected information complexity)**.** *Let a random vector $\varepsilon \in \mathcal{AC}^m$ and let $(\mathcal{A}, \mathcal{F}_\mathcal{A}, \mathcal{P}_\mathcal{A})$ be a probability space that satisfies:* (1) *the corresponding sample space $\mathcal{A}$ is a subset of $\mathcal{PD}^n$ and* (2) *the functional $MI(\cdot, \theta) : \mathcal{A} \to [0, \infty)$ is a random variable for all θ. We define the expected information complexity with noise ε and the probability space $(\mathcal{A}, \mathcal{F}_\mathcal{A}, \mathcal{P}_\mathcal{A})$ of a neural network $f_\theta(x) : \mathbb{R}^n \to \mathbb{R}^m$ to be:*

$$EIC_{\varepsilon, \mathcal{P}_\mathcal{A}}(\theta) := \mathbb{E}[MI(p_X, \theta)] = \int MI(p_X, \theta) d\mathcal{P}_\mathcal{A}.$$

3.2 Special Case: Single Layer Neural Networks without Activation

In this subsection, we derive the close-form expression of MIC of a simple neural network which is just a linear function $f_\theta(x) = Wx$, where $W \in \mathbb{R}^{n \times m}$ is the weight matrix. Let X be the input whose pdf belongs to $\mathcal{A}_P^n$ and let $\varepsilon \sim \mathcal{N}(0, I_m)$ be the noise injected to the latent representation of the neural network $\hat{Y} = f_\theta(X)$. The output of the neural network is $Y = f_\theta(X) + \varepsilon$. Known as gaussian vector channel, this kind of models is well studied in the context of network information theory [6]. The next theorem gives the value of the channel capacity of a gaussian vector channel, which is also MIC of the corresponding neural network.

Theorem 1 (MIC of linear neural networks [6])**.** *Given a linear neural network $f_\theta(x) = Wx$ and a gaussion noise $\varepsilon \sim \mathcal{N}(0, I_m)$. If the rank of W is d (> 0) and the positive singular values of W are $\gamma_1, \ldots, \gamma_d$ in descenting order, the maximum information complexity is*

$$MIC_\varepsilon(W) = \frac{1}{2} \log(\lambda^k \prod_{i=1}^{k} \gamma_i^2),$$

where λ is chosen such that $\sum_{i=1}^{d} \max\{\lambda - \frac{1}{\gamma_i^2}, 0\} = P$ (where P is the constant of $\mathcal{A}_P^n$) and k is the number of singular values γ_i that satisfies $\lambda > \frac{1}{\gamma_i^2}$.

Proof. See the discussion in section 9.1 of [6]

The MIC of linear neural networks is only related to the singular values of the weight matrix. For comparison, weight decay method [16] constrains the L2-norm of the weight matrix by introducing an extra loss $\frac{1}{2}\|W\|_F^2 = \frac{1}{2}tr(W^TW) = \frac{1}{2}\sum_{i=1}^{d}\gamma_i^2$, where $\|\cdot\|_F$ is the Frobinus norm. Yoshida and Miyato [29] proposed to constrain the spectral norm of the weight matrix by introducing an extra loss $\frac{1}{2}\|W\|_2^2 = \frac{1}{2}\gamma_1^2$, where $\|\cdot\|_2$ is the spectral norm. Note that weight decay method and spectral norm regularization show some connections to MIC of the model:

$$\begin{aligned}\log(\lambda^k\prod_{i=1}^{k}\gamma_i^2) &\le \prod_{j=1}^{k}(\frac{1}{k}(P+\sum_{i=1}^{k}\frac{1}{\gamma_i^2})\gamma_j^2) - 1\\ &\le \prod_{j=1}^{k}(P\gamma_j^2+1) - 1\\ &\le (P\|W\|_F^2+1)^d - 1 \le (Pd\|W\|_2^2+1)^d - 1.\end{aligned}$$

The first inequality is obtained because $\log(x) \le x-1$ and $\lambda = \frac{1}{k}(P+\sum_{i=1}^{k}\frac{1}{\gamma_i^2})$. The second inequality holds because $\frac{1}{\gamma_i^2} - \frac{1}{\gamma_j^2} < P$ for all $1 \le i,j \le k$. We can see that the MIC measure is tighter than the Frobinus norm and the spectral norm of W.

3.3 Information Complexity Regularization

In this subsection, we describe how to use information complexity to regularize neural networks.

A general learning framework. Suppose the set of learnable parameters of a neural network is θ and the origin learning objective is to minimize a loss function $L(\theta, S)$ where S is the training dataset. Information complexity regularization methods introduce extra terms to construct a new learning objective:

$$\min_{\theta} L(\theta,S) + \beta_e EIC_{\varepsilon,\mathcal{P}_{\mathcal{A}}}(\theta) + \beta_m MIC_{\varepsilon}(\theta),$$

where the β_e and β_m controls the strength of regularization.

A version of EIC. At first, we show how to construct a probability space on which we can define EIC. In general, a basic idea is to parameterize the set of all possible pdfs of the input random vector and then define a prior distribution on the parameter space. We show a simple example about this strategy and use this version of EIC in our experiments. Suppose the random vector V is uniformly distributed on the n-dimensional hyperrectangle HR : $HR = \{x \in \mathbb{R}^n \,\big|\, \frac{1}{2} \le x_i \le 1 \text{ for all } 1 \le i \le n\}$. Let $G(v) = p_v : HR \to \mathcal{PD}^n$ map a vector v to the pdf of the gaussian random vector with mean 0 and covariance matrix $I_n v$. With the random vector V and the map $G(v)$, it's straightforward to construct a probability space on $\mathcal{NA} = G(HR)$, which we denote $(\mathcal{NA}, \mathcal{F}_{\mathcal{NA}}, \mathcal{P}_{\mathcal{NA}})$. In this case, the EIC of a neural network is : $EIC_{\varepsilon,\mathcal{P}_{\mathcal{NA}}}(\theta) = \mathbb{E}[MI(G(V),\theta)]$, where $G(V)$ is the pdf of the input random vector.

An upper bound of EIC of neural networks. In this work, we use a sampling method to minimize an upper bound of EIC in practice instead of minimizing the exact value of EIC. We can derive an upper bound of EIC:

$$\begin{aligned} EIC_{\varepsilon,\mathcal{P}_{\mathcal{A}}}(\theta) &= \mathbb{E}[I(X,Y)] = \mathbb{E}[h(Y) - h(Y|X)] = \mathbb{E}[h(Y)] - C \\ &\leq \mathbb{E}\left[\frac{1}{2}\log((2\pi e)^m det(K_Y))\right] - C = \mathbb{E}\left[\frac{1}{2}\log det(K_Y)\right] - C^{'} \\ &\leq \frac{1}{2}\mathbb{E}[tr(K_Y)] - C^{''}, \end{aligned}$$

where det means the determinant of a matrix and tr means the trace of a matrix. The first inequality holds because the maximum-entropy distribution for a given covariance is Gaussian with the same covariance [5]. The second inequality is true because $det(K_Y)$ is the product of singular values of K_Y and $tr(K_Y)$ is the sum of the singular values.

In order to minimize $\mathbb{E}[tr(K_Y)]$, we sample n pdfs $(p_X^{(1)}, \ldots, p_X^{(n)})$ of the input random vector based on $(\mathcal{A}, \mathcal{F}_{\mathcal{A}}, \mathcal{P}_{\mathcal{A}})$. Then we use each pdf $p_X^{(i)}$ to get k samples of the input $(\hat{x}_{i1}, \ldots, \hat{x}_{ik})$, which are fed into the neural network to get the samples of the output $(\hat{y}_{i1}, \ldots, \hat{y}_{ik})$. The sampled outputs are used to approximate $\mathbb{E}[tr(K_Y)]$:

$$\begin{aligned} \mathbb{E}[tr(K_Y)] &= \int \mathbb{E}_{Y\sim p_Y} \|Y - \mathbb{E}_{Y\sim p_Y}[Y]\|^2 d\mathcal{P}_{\mathcal{A}} \\ &\approx \frac{1}{kn}\sum_{i=1}^{n}\sum_{j=1}^{k} \|\hat{y}_{ij} - \frac{1}{k}\sum_{l=1}^{k}\hat{y}_{il}\|^2 \\ &\triangleq \widehat{tr}, \end{aligned}$$

where $\|\cdot\|$ means the L2-norm of a vector. It's easy to calculate the gradient of $\widehat{tr}$ with respect to model parameters.

A surrogate of MIC of neural networks. We also use a sampling method to minimize a surrogate of MIC in practice. As mentioned before, the MIC of linear neural networks has an upper bound $\prod_{j=1}^{k}(\frac{1}{k}(P + \sum_{i=1}^{k}\frac{1}{\gamma_i^2})\gamma_j^2)$, which is a polynomial of singular values and the ratios between them.

Suppose the singular value decomposition of W is $U\Sigma V^T$. Let $X \sim \mathcal{N}(0, I_n)$ and let $X^{'} = V^T X$, $X^{''} = \frac{X^{'}}{\|X^{'}\|}$. Meanwhile, we have the following equations:

$$\frac{\|Wx\|}{\|x\|} = \frac{\|U\Sigma V^T V x^{'}\|}{\|Vx^{'}\|} = \frac{\|U\Sigma x^{'}\|}{\|x^{'}\|} = \|U\Sigma x^{''}\| = \sqrt{\sum_{i=1}^{d}\gamma_i^2 (x_i^{''})^2}.$$

It's straightforward to see that $X^{''}$ is a random vector having uniform distribution on $\{x \in \mathbb{R}^n \mid \|x\| = 1\}$. Given $X \sim \mathcal{N}(0, I_n)$, we minimize $\mathbb{E}\left[\frac{\|Wx\|}{\|x\|}\right]$ to constrain

the singular values and minimize $\mathbb{V}\left[\frac{\|Wx\|}{\|x\|}\right]$, where $\mathbb{V}[\cdot]$ means the variance, to limit the diversity of singular values.

Furthermore, we hypothesize that minimizing the surrogate of MIC of general neural networks f_θ:

$$MICS = \mathbb{E}\left[\frac{\|f_\theta(x)\|}{\|x\|}\right] + \mathbb{V}\left[\frac{\|f_\theta(x)\|}{\|x\|}\right],$$

is an effective way to constrain the MIC. The sampling method to estimate $MICS$ is straightforward and we minimize the estimator $\widehat{MICS}$ in our experiments.

Discussion. Note that we don't need to access the training data to calculate $\widehat{tr}$ and $\widehat{MICS}$. It's worth mentioning that our method is not limited to constrain the information complexity of the whole network. By viewing each block (e.g. first n layers of a neural network network) as a tiny neural network, we can impose multiple information complexity regularizer terms for more flexibility.

4 Experiments

In this section, we evaluate our methods on various image classification datasets: MNIST, Kuzushiji-MNIST, SVHN and CIFAR10 [3]. All models are implemented using PyTorch and trained on a single NVIDIA GeForce RTX 2080TI GPU.

4.1 Benchmark Experiments

We first experiment our regularization methods on various image classification datasets: MNIST, Kuzushiji-MNIST, SVHN, and CIFAR-10.

Settings. For MNIST and Kuzushiji-MNIST, we train three-layer MLPs with fully connected layers of the form 784–1024–1024–10 and the ReLU activation function. The batchsize is set as 100. For MNIST and Kuzushiji-MNIST, all models are trained using ADAM optimizer [14] with an initial learning rate 0.001 for 200 epochs and we decay the learning rate by a factor of 0.97 every 2 epochs. We train ResNet-20 from [10] for SVHN and ResNet-44 for CIFAR10. The batchsize is set as 128. ResNets are trained using Nesterov's accelerated gradient descent [4] with momentum 0.9 for 160 epochs. For SVHN and CIFAR10, we set the initial learning rate as 0.1 and decay the learning rate by a factor of 0.1 once the half and the three quarters of the training process have passed. For all datasets, we also report results by adding weight decay (WD).

Our methods include MIC regularization method (train networks with an extra loss $\widehat{MICS}$) and EIC regularization method (train networks with an extra loss $\widehat{tr}$). We impose constraints on both the IC of whole network and the IC of each

[3] The code is available at https://github.com/IanyePKU/IC-Regularization-methods

Table 1. Experimental results on benchmark datasets.The average and standard deviation of the accuracy of each method over 3 trials. We compare "baseline", "+MIC" and "+EIC" and the best one is shown in **boldface**. The best result for each dataset is shown with **underline**.

Dataset	Model & Setup	baseline	+MIC	+EIC
MNIST	MLP	98.56%(0.03)	**98.87%(0.03)**	98.86%(0.05)
MNIST	MLP +WD	98.69%(0.02)	**98.78%(0.05)**	**98.78%(0.03)**
Kuzushiji	MLP	93.53%(0.08)	93.94%(0.13)	**94.16%(0.11)**
Kuzushiji	MLP +WD	93.40%(0.09)	93.74%(0.07)	**93.85%(0.22)**
SVHN	ResNet20	95.45%(0.12)	95.59%(0.09)	**95.60%(0.04)**
SVHN	ResNet20 +WD	95.73%(0.09)	95.89%(0.09)	**95.93%(0.05)**
CIFAR10	ResNet44	85.76%(0.23)	86.11%(0.17)	**86.56%(0.33)**
CIFAR10	ResNet44 +WD	87.91%(0.08)	87.95%(0.10)	**88.10%(0.17)**

layer and assume that all noises obey the gaussian distribution. In all experiments, we perform the hyper-parameter search for β_e and β_m with candidates from $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$. For EIC method, we use the probability space $(\mathcal{NA}, \mathcal{F}_{\mathcal{NA}}, \mathcal{P}_{\mathcal{NA}})$ to estimate $\widehat{tr}$, implying that we expect fewer information is preserved in the outputs of the neural network when the inputs are from a gaussian distribution.

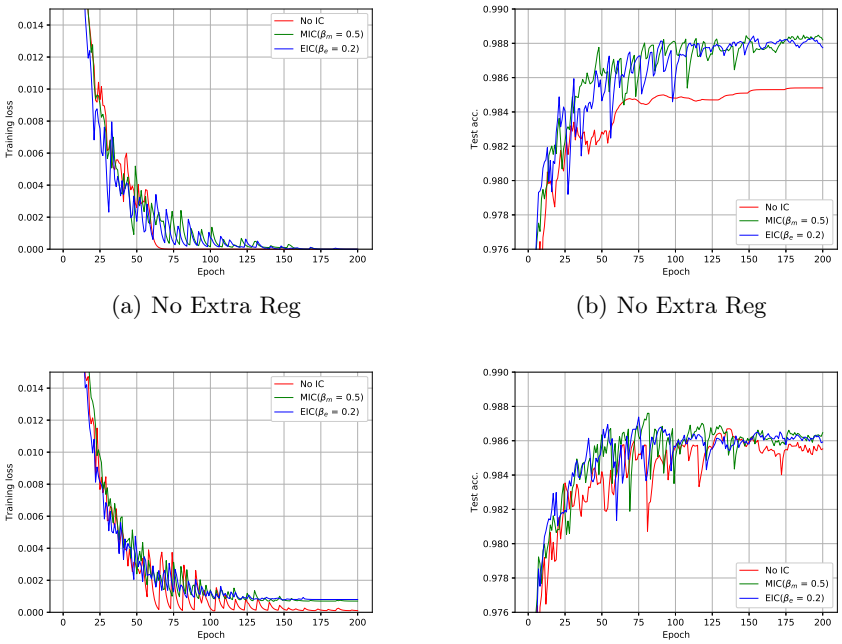
Results. The test error rate obtained by each method is summarized in Table 1. The best result for each dataset is always achieved by the models trained with IC regularization methods. For SVHN and CIFAR10, combining our methods with weight decay has complementary effects. Introducing an extra IC regularization term always improves performance, which justify the usefulness of our methods.

As shown in Fig. 1, training curves of models trained with our methods are usually serrated, which means that our models frequently escape from the regions with low training loss. We think that this feature of our methods helps models leave local minimum and achieve better generalization performance. The phenomena is potentially related to a recent observation [13] that zero training loss is not the final goal of the training process.

4.2 Comparison Experiment with other regularizaion methods

In order to verify the compatibility between our method with other regularization methods and compare our method with them, we train MLPs on MNIST using various combinations of regularizers. All settings of hyperparameters are same as the previous benchmark experiments. Other regularization methods used for comparison include:

– Weight Decay: Add an extra loss to constrain the L2-norm of parameters of the neural network. We set the regularization factor $\beta_{L2} = 10^{-4}$.



(a) No Extra Reg (b) No Extra Reg

(c) With Weight Decay (d) With Weight Decay

Fig. 1. Training curves of MLPs on MNIST. The top two figures show the training curves of MLPs using only our regularization terms. The bottom two figures show the training curves of MLPs using our regularization terms and weight decay.

- Confidence Penalty (CP): Penalize low entropy output distributions. We search for the best regularization factor β_{CP} from $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ as [20].
- Label Smoothing (LS): Minimize the KL divergence between uniform distribution and the network's predicted distribution.We search for the best regularization factor β_{LS} from $\{0.1, 0.3, 0.5, 1.0, 2.0, 4.0, 8.0\}$ as [20].
- Variational Information Bottleneck (VIB): Train neural networks using variational information bottleneck principle and minimize an upper bound of MI between the input of the network and the corresponding representation. We search for the best regularization factor β_{VIB} from $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$, which is a reasonable set according to [1].

Result. The test accuracy obtained by each method is summarized in Table 2. Setup of using only MIC (average of accuracy is 98.87%, see Table 1) outperforms setups of combining other regularization terms (first two column of results from Table 2). Combining IC regularization terms further improves the performance and the model trained using CP and MIC achieves the best performance (average

Table 2. Experimental results on MNIST. The average and standard deviation of the accuracy of each method over 3 trials. For each regularization method, we compare "No other reg", "+WD", "+MIC" and "+EIC" and the best one is shown in **boldface**.

Method	No other reg	+WD	+MIC	+EIC
CP	98.65%(0.02)	98.86%(0.03)	**98.97%(0.02)**	98.95%(0.04)
LS	98.79%(0.07)	98.85%(0.05)	**98.93%(0.05)**	**98.93%(0.01)**
VIB	98.66%(0.04)	98.80%(0.05)	**98.88%(0.01)**	98.81%(0.01)

of accuracy is 98.97%). This experiment shows that introducing an information complexity term boosts the generalization performance of trained neural networks and is compatible with other regularization methods.

5 Conclusion

In this paper, we have defined two kinds of new complexity measures for neural networks by linking each neural network to a communication channel. We showed a connection between the MIC of a single layer linear neural network and the L2-norm of its weight matrix. We also designed two new regularization methods using EIC and MIC. We conducted experiments on image classification datasets and showed the usefulness of our new regularization terms empirically.

References

1. Alemi, A.A., Fischer, I., Dillon, J.V., Murphy, K.: Deep variational information bottleneck. arXiv preprint arXiv:1612.00410 (2016)
2. Amjad, R.A., Geiger, B.C.: Learning representations for neural network-based classification using the information bottleneck principle. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(9), 2225–2239 (2019)
3. Belghazi, M.I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., Hjelm, D.: Mutual information neural estimation. In: International Conference on Machine Learning. pp. 531–540. PMLR (2018)
4. Bengio, Y., Boulanger-Lewandowski, N., Pascanu, R.: Advances in optimizing recurrent networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 8624–8628. IEEE (2013)
5. Cover, T.M.: Elements of information theory. John Wiley & Sons (1999)
6. El Gamal, A., Kim, Y.H.: Network information theory. Cambridge university press (2011)
7. Foggo, B., Yu, N.: On the maximum mutual information capacity of neural architectures. arXiv preprint arXiv:2006.06037 (2020)
8. Gao, S., Ver Steeg, G., Galstyan, A.: Efficient estimation of mutual information for strongly dependent variables. In: Artificial Intelligence and Statistics. pp. 277–286. PMLR (2015)
9. Hafez-Kolahi, H., Kasaei, S.: Information bottleneck and its applications in deep learning. arXiv preprint arXiv:1904.03743 (2019)

10. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European Conference on Computer Vision. pp. 630–645. Springer (2016)
11. Henaff, O.: Data-efficient image recognition with contrastive predictive coding. In: International Conference on Machine Learning. pp. 4182–4192. PMLR (2020)
12. Hinton, G.E., van Camp, D.: Keeping the neural networks simple by minimizing the description length of the weights. In: Proceedings of the Sixth Annual Conference on Computational Learning Theory. p. 5–13 (1993)
13. Ishida, T., Yamane, I., Sakai, T., Niu, G., Sugiyama, M.: Do we need zero training loss after achieving zero training error? arXiv preprint arXiv:2002.08709 (2020)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
15. Kolchinsky, A., Tracey, B.D., Wolpert, D.H.: Nonlinear information bottleneck. Entropy **21**(12), 1181 (2019)
16. Krogh, A., Hertz, J.A.: A simple weight decay can improve generalization. In: Advances in Neural Information Processing Systems. pp. 950–957 (1992)
17. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Computation **1**(4), 541–551 (1989)
18. Linsker, R.: Self-organization in a perceptual network. Computer **21**(3), 105–117 (1988)
19. MacKay, D.J.: Information theory, inference and learning algorithms. Cambridge university press (2003)
20. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., Hinton, G.: Regularizing neural networks by penalizing confident output distributions. In: International Conference on Learning Representations Workshop (2017)
21. Saxe, A.M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B.D., Cox, D.D.: On the information bottleneck theory of deep learning. Journal of Statistical Mechanics: Theory and Experiment **2019**(12), 124020 (2019)
22. Shwartz-Ziv, R., Tishby, N.: Opening the black box of deep neural networks via information. arXiv preprint arXiv:1703.00810 (2017)
23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2818–2826 (2016)
24. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. arXiv preprint arXiv:1906.05849 (2019)
25. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. In: Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing. pp. 368–377 (1999)
26. Tishby, N., Zaslavsky, N.: Deep learning and the information bottleneck principle. In: 2015 IEEE Information Theory Workshop (ITW). pp. 1–5. IEEE (2015)
27. Tschannen, M., Djolonga, J., Rubenstein, P.K., Gelly, S., Lucic, M.: On mutual information maximization for representation learning. In: International Conference on Learning Representations (2020)
28. Walters-Williams, J., Li, Y.: Estimation of mutual information: A survey. In: International Conference on Rough Sets and Knowledge Technology. pp. 389–396. Springer (2009)
29. Yoshida, Y., Miyato, T.: Spectral norm regularization for improving the generalizability of deep learning. arXiv preprint arXiv:1705.10941 (2017)