2016 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 13–16, 2016, SALERNO, ITALY

FLOW OF RENYI INFORMATION IN DEEP NEURAL NETWORKS

Che-Wei Huang, Shrikanth (Shri) S. Narayanan

Signal Analysis and Interpretation Laboratory (SAIL)
University of Southern California
cheweihu@usc.edu, shri@sipi.usc.edu

ABSTRACT

We propose a rate-distortion based deep neural network (DNN) training algorithm using a smooth matrix functional on the manifold of positive semi-definite matrices as the non-parametric entropy estimator. The objective in the optimization function includes not only the measure of performance of the output layer but also the measure of information distortion between consecutive layers in order to produce a concise representation of its input on each layer. An experiment on speech emotion recognition shows the DNN trained by such method reaches comparable performance with an encoder-decoder system.

Index Terms— Deep learning, artificial neural network, Information-theoretic learning, Renyi entropy

1. INTRODUCTION

In the past decade, deep neural network systems have become extraordinarily successful in various machine learning and artificial intelligence tasks such as speech recognition, image recognition and natural language processing. Deep neural networks, as the simplest form among deep neural network systems, is a class of architectures comprised of connected units, called neurons, laid out in a layer-wise fashion. Despite its remarkable performance on many pattern recognition tasks, the pursuit for a theoretical understanding of DNNs is still an ongoing research effort.

A single neuron implements a hyperplane which is able to optimally classify conditionally independent input data. However, in general the data distribution does not possess such statistical independence. To have conditionally independent data entering into an output neuron for optimal classification, one option is to appropriately transform the input data in order to achieve the desired statistical property. A recent work by Mehta et al. [1] showed that there is an exact mapping between the variational renormalization group (RG) and DNNs based on restricted Boltzmann machines (RBM). The RG transformations iteratively integrate out irrelevant features from a microscopic scale to a larger scale while retaining the relevant ones. Such successive transformations effectively approximate conditional independence along the layers. A following work by Tishby et al. [2], based on the idea of coarse-graining, proposed to formulate the learning problem of a DNN as a tradeoff between compression and prediction. Information-theoretically speaking, they argued that the DNN learning problem can be posed as extracting the minimal sufficient statistics of input data with respect to the output, which can be viewed as a special case in the rate distortion theory: the information bottleneck (IB) method [3]. In addition, the authors claimed the phase transitions that are governed by the Lagrangian multiplier β play an important role in the optimal design of DNN architectures.

Khadivi et al. [4] investigated the flow of the discrete Shannon entropy across consecutive layers in a DNN and defined a new optimization problem for training a DNN based on the IB principle. Moreover, they demonstrated numerically that a DNN can successfully learn boolean functions (AND, OR, XOR) while achieving the minimal representation of the data.

In this paper, we consider the flow of entropy across consecutive layers for *continuous* random variables. This task is not trivial from many perspectives. First of all, the IB method works naturally well with discrete bottleneck random variables T, but the inference for its extension to the general solution of continuous random variables T is intractable except for the special case when the source X and relevance Y random variables are jointly Gaussian [5]. In addition, analytically computing the differential entropy of these transformed random variables starting from the input layer poses another challenge due to the often non-square connectivity matrices. Numerically, a straight-forward method, the so-called plug-in method, divides the entropy estimation into two steps. First, a density estimation based on the data samples is carried out to fit a pre-defined density model. The entropy estimator is obtained by plugging the estimated density into the definition of the information theoretic quantity. Another class of techniques can perform one-step estimation for entropy and mutual information based on entropic graphs. For example, Pál et al. [6] proposed an estimator using k-nearest-neighbor graphs with a high probability bounds on the estimation error. However, these graph based estimators are not differentiable and thus not suitable for gradient optimization except for the work by Faivishevsky et al. [7], who gave a smooth estimator of Shannon's differential entropy by averaging k-nearest-

978-1-5090-0746-2/16/$31.00

neighbor statistics for the all possible values of order statistics k.

Instead, we employ a recently developed approach to estimate the Renyi entropy [8] directly from data based on infinitely divisible kernels [9]. This smooth non-parametric estimation using operators in reproducing kernel Hilbert spaces is a nice result coming from the marriage of the information-theoretic learning and the kernel method, and has been shown to be effective for auto-encoder learning [10] and metric learning [11]. The fact that the matrix functional of Renyi entropy defined in [9] has a strong resemblance to the quantum Renyi entropy is another intriguing motivation as the previous work has pointed out a connection between RG and RBM based DNNs.

The outline of this paper is as follows. The next section will introduce some basics and notations from topics including DNN, rate-distortion theory, and the matrix-based Renyi entropy. In the third section, we will formulate the DNN learning problem as a rate-distortion optimization problem, followed by a section devoted to the experiments on speech emotion recognition to show the flow of Renyi entropy in a DNN when performing a real-world task. The last section will conclude this paper.

2. DNNS, RATE DISTORTION THEORY AND MATRIX-BASED RENYI ENTROPY

2.1. Deep Neural Networks (DNNs)

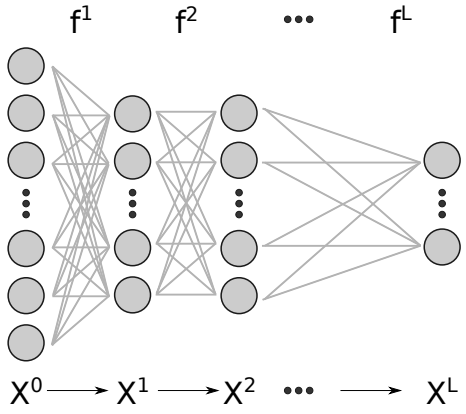

Fig. 1. An illustration of the DNN architecture.

Suppose $\{(\mathbf{x}_i, y_i)|i = 1, \cdots, N\}$ is the set of training data, where $\mathbf{x}_i \in \mathbb{R}^d$ are features and $y_i \in \{0, \cdots, C\}$ the corresponding label. The goal of discriminative learning is to approximate the conditional probability distribution $\mathbf{f} = [f^0, \cdots, f^C]^T$, where $f^y(\mathbf{x}) \triangleq p(y|\mathbf{x})$, from which

$$\mathbf{f} : \mathbb{R}^d \to [0, 1]^C$$

makes the minimum loss with respect to a target loss function $\mathcal{L}(\mathbf{f}, y)$.

A DNN is a network architecture laid out in a layer-wise fashion with a number of neurons on each layer. Each neuron transforms its input to an output according to its activation function. The representation on the l^{th} layer is the activation values on the layer. Suppose there are L layers. We denote the data represented by the layers as $\{\mathbf{X} = \mathbf{X}^0, \cdots, \mathbf{X}^l, \cdots, \mathbf{X}^L = \hat{\mathbf{Y}}\}$, where $\mathbf{X}$ is the data into the input layer and $\hat{\mathbf{Y}}$ the estimated output.

The relation between two consecutive layers can be expressed as

$$\mathbf{X}^l = \mathbf{f}^l(\mathbf{X}^{l-1}) = \mathbf{g}^l(\mathbf{W}^l\mathbf{X}^{l-1} + \mathbf{b}^l), \tag{1}$$

where $\mathbf{W}^l$ is the connectivity matrix from $(l-1)^{\text{th}}$ to l^{th} layers and $\mathbf{g}^l(\mathbf{x}) = \sigma^l \odot \mathbf{x}$ indicates the activation function on l^{th} layer acting element-wisely on the components of $\mathbf{x}$. Notice here the functions $\mathbf{f}^l$ are deterministic though unknown and need to be learned. The function $\mathbf{f}$ that a DNN desires to approximate is the result of multiple compositions and can be summarized succinctly as follows:

$$\mathbf{f} = \mathbf{f}^L \circ \cdots \circ \mathbf{f}^1. \tag{2}$$

Fig. 1 gives an illustration of a typical DNN architecture.

Since each layer depends only on the previous layer, the layers in a DNN form a Markov chain $\mathbf{Y} - \mathbf{X} - \mathbf{X}^1 - \cdots - \mathbf{X}^L$. According to the data processing inequality, the following holds for any $j \geq i$:

$$I(\mathbf{Y}; \mathbf{X}) \geq I(\mathbf{Y}; \mathbf{X}^i) \geq I(\mathbf{Y}; \mathbf{X}^j) \geq I(\mathbf{Y}; \hat{\mathbf{Y}}), \tag{3}$$

where $I(;)$ is the Shannon mutual information. The equality hold if and only if each layer is a sufficient statistics of its input.

2.2. Rate Distortion Theory

Suppose $\mathbf{X}$ is a discrete random variable and the resource for the representation of $\mathbf{X}$ is limited. An interesting question is to optimally quantize the random variable $\mathbf{X}$ subject to a constraint on the distortion between the original $\mathbf{X}$ and the quantized $\hat{\mathbf{X}}$ random variables. This question motivates the classical rate distortion theory.

Let $\mathbf{X}$ a random variable distributed by some discrete probability mass function $p(x)$, $x \in \mathcal{X}$, where $\mathcal{X}$ is the set of alphabets. Denote $\hat{\mathcal{X}}$ the set of alphabets for the quantized random variables. Suppose $\mathcal{D} : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbf{R}^+$ is a distortion measure which maps a pair of random variables into a non-negative real number. The rate distortion problem can be formulated as

$$R(D) \triangleq \min_{p(\hat{\mathbf{x}}|\mathbf{x}):\sum_{\mathbf{x},\hat{\mathbf{x}}} p(\mathbf{x})p(\hat{\mathbf{x}}|\mathbf{x})\mathcal{D}(\mathbf{x}|\hat{\mathbf{x}}) \leq D_c} I(\mathbf{X}; \hat{\mathbf{X}}), \tag{4}$$

where $p(\hat{\mathbf{x}}|\mathbf{x})$ is the encoding probability and D_c the constraint on the distortion. The problem in Eq. (4) can be re-

formulated as an unconstrained optimization problem by introducing the Lagrangian multiplier

$$\min_{p(\hat{\mathbf{x}}|\mathbf{x})} \mathcal{F}[p(\hat{\mathbf{x}}|\mathbf{x})] = I(\mathbf{X};\hat{\mathbf{X}}) + \beta \sum_{\mathbf{x},\hat{\mathbf{x}}} p(\mathbf{x})p(\hat{\mathbf{x}}|\mathbf{x})\mathcal{D}(\mathbf{x}|\hat{\mathbf{x}}). \tag{5}$$

Solving $\mathcal{F}[p(\hat{\mathbf{x}}|\mathbf{x})]$ variationally would yield a set of self-consistent equations. The Blahut-Arimoto algorithm [12] then iteratively searches for the solution. The rate distortion problem tries to minimize the irrelevant information of $\mathbf{X}$ while keeping the optimal amount of relevant one subject to a constraint. The quantized random variable $\hat{\mathbf{X}}$ can be viewed as a representation of $\mathbf{X}$.

One issue with the rate distortion is the choice of the distortion measure implicitly determines the form of the representation, which is often not completely known. The IB method [3] offers a slightly relaxed formulation. Rather than assuming the knowledge of the representation, the authors assumed there is a relevance random variable $\mathbf{Y}$ correlated to $\mathbf{X}$, i.e., $I(\mathbf{X};\mathbf{Y}) > 0$. Minimizing the variational problem $\mathcal{F}[p(\hat{\mathbf{x}}|\mathbf{x})] = I(\mathbf{X};\hat{\mathbf{X}}) - \beta I(\hat{\mathbf{X}};\mathbf{Y})$ is analogous to passing the information in $\mathbf{X}$ about $\mathbf{Y}$ through a bottleneck $\hat{\mathbf{X}}$, where $\min I(\mathbf{X};\hat{\mathbf{X}})$ plays a similar role in reducing the relevant information while $\min -I(\hat{\mathbf{X}};\mathbf{Y}) = \max I(\hat{\mathbf{X}};\mathbf{Y})$ maximizes the relevant one. It turns out the IB method is a special case of the rate distortion theory where the choice of the distortion measure is the Kullback-Leibler divergence between $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{y}|\hat{\mathbf{x}})$ [3].

Recently, Tishby et al. [2] proposed to analyse DNNs using the principle of the IB method. The information distortion measure introduces extra terms to the objective function. Information distortion between consecutive layers is incorporated in addition to the usual loss function between the output layer and the input. The motivation for maximizing $I(\mathbf{X}^L;\mathbf{Y})$ stems from the fact that $I(\mathbf{X}^L;\mathbf{Y})$ bounds the prediction error in classification tasks [13], while minimizing $I(\mathbf{X};\mathbf{X}^l)$ could capture the most concise representation. Furthermore, the authors also demonstrated the bifurcation points on the information curve to a sub-optimal curve at the critical values of β, and made a conjecture that these critical points could help in the design of the optimal architecture. Khadivi et al. [4] characterized the discrete entropy change between consecutive layers based on the IB principle. They formulated DNN training as the following constrained optimization problem:

$$\min_{I(\mathbf{X};\mathbf{Y}|\mathbf{X}^L)\leq\epsilon} \sum_{l=1}^{L} I(\mathbf{X};\mathbf{X}^l). \tag{6}$$

However, when applying to learning the booleans functions, the problem in Eq. (6) contains infeasible solutions because the constraint does penalize such solutions. The authors demonstrated a modified problem can perfectly learns these boolean functions while achieving the optimal compression bound.

2.3. Matrix-based Renyi Entropy

Suppose $\mathbf{X}$ is a random variable distributed by a continuous probability density function $p(\mathbf{x})$. The continuous Shannon entropy of $\mathbf{X}$ is the differential entropy

$$h(\mathbf{x}) = -\int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x}. \tag{7}$$

Despite sharing some commonalities with the discrete Shannon entropy, the differential entropy is different from its discrete counterpart in a number of properties. One of them that complicates a direct extension from the previous information theoretical analysis of DNNs to a continuous setting is the scaling property.

Suppose $\mathbf{Y} = m(\mathbf{X})$ is a random variable transformed from $\mathbf{X}$ via the map $m : \mathbb{R}^p \to \mathbb{R}^q$. If $p = q$, then the following inequality holds [14] :

$$h(\mathbf{Y}) \leq h(\mathbf{X}) + \int p(\mathbf{x}) \log \left| \frac{\partial m}{\partial \mathbf{x}} \right| d\mathbf{x}, \tag{8}$$

where $\frac{\partial m}{\partial \mathbf{x}}$ is the Jacobian of the map m. The equality holds when m is a non-singular transformation. When $p \neq q$ the Jacobian matrix is not square and there is no systematic method to derive the differential entropy of $\mathbf{Y}$.

The major difficulty of analytical analysis comes from the fact that transformations $\mathbf{f}^l$ between layers in a DNN are often non-square. Suppose we further impose an additional assumption that the input data distribution is Gaussian. Even though the linear connectivity matrix $\mathbf{W}^1$ preserves the Gaussianity (perhaps in a degenerate form), the non-linearity of the activation function σ^1 would destroy the Gaussianity and the analysis would be difficult beyond the first hidden layer. The non-linearity is a key component to the very success of deep learning. Therefore there is no reason to further assume linear activation functions.

One of the numerical approaches for estimating entropy based on the data samples is the plug-in method. However, drawbacks of the plug-in method include the accumulated error for estimating the density models and the need to choose a parametric model. On the other hand, non-parametric entropic graph based methods, though capable of directly estimating the information theoretic quantities and possessing nice convergence properties, are mostly non-differentiable and not suitable as the objective for gradient optimization except for the work by Faivishevsky et al. [7].

A recent framework of non-parametric estimation for the matrix based Renyi entropy was developed based on the information theoretical learning and the reproducing kernel Hilbert space [9]. This new estimator is a smooth matrix functional on the manifold of the positive definite matrices over the real numbers, and has been shown to be effective in auto-encoders learning [10] and metric learning [11]. In this paper, we will use the matrix-based Renyi entropy to formulate the DNN training problem based on the rate-distortion theory.

Definition 1 *Suppose* ***PSD***(n) *denotes the set of all positive semi-definite matrices over real numbers. Let* $\mathbf{K} \in$ ***PSD***(n). *The matrix functional, parametrized by a real number* $\alpha > 0$,

$$S_\alpha(\mathbf{K}) = \frac{1}{1-\alpha} \log_2 \mathrm{Tr}(\mathbf{K}^\alpha) \tag{9}$$

satisfies the axioms for a matrix functional to be a measure of entropy, and is called the matrix based Renyi entropy.

The definition in Eq.(9) is similar to the quantum Renyi entropy, $S_\alpha(\rho) = \frac{1}{1-\alpha} \log \mathrm{Tr}(\rho^\alpha)$, where ρ is a density operator, which is equivalent to a positive-semi definite matrix when an orthonormal basis is fixed.

For a positive definite matrix $\mathbf{K}$, the fractional power of $\mathbf{K}$, denoted by $\mathbf{K}^{\circ r}$ for some $r \in \mathbb{R}^+$, is not necessarily positive semi-definite, where $(\mathbf{K}^{\circ r})_{ij} = \mathbf{K}_{ij}^r$.

Definition 2 *Suppose* $\mathbf{K}$ *is positive semi-definite and every entry in* $\mathbf{K}$ *is non-negative.* $\mathbf{K}$ *is said to be infinitely divisible if* $\mathbf{K}^{\circ r}$ *is positive semi-definite for every non-negative* r.

Definition 3 *The joint Renyi entropy of two positive semi-definite matrices* $\mathbf{K}_1$ *and* $\mathbf{K}_2$ *is defined through the Hadamard product* $\circ$ *of these two matrices by*

$$S_\alpha(\mathbf{K}_1, \mathbf{K}_2) = S_\alpha \left(\frac{\mathbf{K}_1 \circ \mathbf{K}_2}{\mathrm{Tr}(\mathbf{K}_1 \circ \mathbf{K}_2)} \right). \tag{10}$$

This is well-defined since the set of positive semi-definite matrices is closed under the Hadamard product, and the use of infinitely divisible kernels makes sure $S_\alpha(\mathbf{K}^{\circ r} / \mathrm{Tr}(\mathbf{K}^{\circ r}))$ is also well-defined.

Proposition 1 *[9] Let* $\mathbf{K}_1, \mathbf{K}_2 \in$ ***PSD***(n) *and* $\mathrm{Tr}(\mathbf{K}_1) = \mathrm{Tr}(\mathbf{K}_2) = 1$ *with* $(\mathbf{K}_1)_{ii} = \frac{1}{n}$ *for all* i. *Then*

1. $S_\alpha \left(\frac{\mathbf{K}_1 \circ \mathbf{K}_2}{\mathrm{Tr}(\mathbf{K}_1 \circ \mathbf{K}_2)} \right) \geq S_\alpha(\mathbf{K}_2)$.
2. $S_\alpha \left(\frac{\mathbf{K}_1 \circ \mathbf{K}_2}{\mathrm{Tr}(\mathbf{K}_1 \circ \mathbf{K}_2)} \right) \leq S_\alpha(\mathbf{K}_1) + S_\alpha(\mathbf{K}_2)$.

The second item in Proposition 1 generalizes the chain rule and provides a partial guarantee for the non-negativeness of the mutual information defined below.

Definition 4 *Let* $\mathbf{K}_1, \mathbf{K}_2 \in$ ***PSD***(n), $\mathrm{Tr}(\mathbf{K}_1) = \mathrm{Tr}(\mathbf{K}_2) = 1$ *and* $(\mathbf{K}_m)_{ij} \geq 0$ *for all* m, i, j *such that* $(\mathbf{K}_1)_{ii} = (\mathbf{K}_2)_{ii} = \frac{1}{n}$ *for all* i. *The matrix based Renyi mutual information is defined as*

$$I_\alpha(\mathbf{K}_1; \mathbf{K}_2) = S_\alpha(\mathbf{K}_1) + S_\alpha(\mathbf{K}_2) - S_\alpha(\mathbf{K}_1, \mathbf{K}_2). \tag{11}$$

The radial basis function kernel (RBF) kernel $\mathbf{K}$ with $\mathbf{K}_{ij} = \exp(-\frac{|\mathbf{x}^i - \mathbf{x}^j|^2}{2\sigma_x^2})$ belongs to the class of infinitely divisible kernel and the trace normalized RBF kernel satisfies all of the premises in Definition 4.

3. RATE-DISTORTION DEEP NEURAL NETWORKS

The principle of IB analysis for DNNs proposed in [2] suggested that in addition to optimization of the performance of the output layer, it is information-theoretically meaningful to minimize the propagation of irrelevant information through hidden layers. The Shannon mutual information $I(\mathbf{X}; \mathbf{Y})$ for any two random variables can be decomposed into two terms by $I(\mathbf{X}; \mathbf{Y}) = -\mathbf{CE}(\mathbf{Y}, \mathbf{X}) + \mathrm{H}(\mathbf{Y})$, the cross-entropy between $\mathbf{Y}$ and $\mathbf{X}$ and the entropy of $\mathbf{Y}$. Moreover, $I(\mathbf{X}; \mathbf{Y}|F(\mathbf{X})) = I(\mathbf{Y}; \mathbf{X}) - I(\mathbf{Y}; F(\mathbf{X}))$ for some deterministic map $\mathbf{F}$. By analogy, the rate-distortion based training of a DNN can be formulated as

$$\begin{aligned} & \min_{\mathbf{f}: I(\mathbf{X};\mathbf{Y}|\mathbf{f}(\mathbf{X})) \leq \epsilon} \sum_{l=1}^{L} I_\alpha(\mathbf{X}; \mathbf{X}^l) \\ \equiv \quad & \min_{\mathbf{f}} \sum_{l=1}^{L} I_\alpha(\mathbf{X}; \mathbf{X}^l) - \beta I_\alpha(\mathbf{f}(\mathbf{X}); \mathbf{Y}) \\ \equiv \quad & \min_{\mathbf{f}} \sum_{l=1}^{L} I_\alpha(\mathbf{X}; \mathbf{X}^l) + \beta \mathbf{CE}(\mathbf{Y}, \mathbf{f}(\mathbf{X})), \end{aligned} \tag{12}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are the input and output random variables, and $\mathbf{f}$ is a function a DNN tries to learn.

Each term in Eq. (12) is differentiable. Here we derived the gradient for a DNN with one hidden layer. For a more general case, the derivation is similar. We assumed the activation on the output layer is the softmax function, while the activation on the hidden layer is any differentiable activation function acting element-wisely. We refer readers to [15] for the partial gradients of the cross-entropy. Following are the partial gradients of entropies with respect to the network parameters:

$$\begin{aligned} \frac{\partial S_\alpha(\mathbf{K}^2)}{\partial \mathbf{W}^2} &= 4 \left[\mathbf{B}^2 \mathbf{X}^2 - \mathbf{A}^2 \circ \mathbf{X}^2 \right]^T \mathbf{X}^1, \\ \frac{\partial S_\alpha(\mathbf{K}^2)}{\partial \mathbf{b}^2} &= 4 \left[\mathbf{B}^2 \mathbf{X}^2 - \mathbf{A}^2 \circ \mathbf{X}^2 \right]^T \mathbf{1}, \\ \frac{\partial S_\alpha(\mathbf{K}^2)}{\partial \mathbf{W}^1} &= 4 \big[\mathbf{B}^2((\mathbf{X}^2 \mathbf{W}^2) \circ \mathbf{G}^1) \\ &\quad - ((\mathbf{A}^2 \circ \mathbf{X}^2)\mathbf{W}^2) \circ \mathbf{G}^1 \big]^T \mathbf{X}^1, \\ \frac{\partial S_\alpha(\mathbf{K}^2)}{\partial \mathbf{b}^1} &= 4 \big[\mathbf{B}^2((\mathbf{X}^2 \mathbf{W}^2) \circ \mathbf{G}^1) \\ &\quad - ((\mathbf{A}^2 \circ \mathbf{X}^2)\mathbf{W}^2) \circ \mathbf{G}^1 \big]^T \mathbf{1}, \\ \frac{\partial S_\alpha(\mathbf{K}^1)}{\partial \mathbf{W}^1} &= -4 \left[(\mathbf{D}^1 \mathbf{X}^1) \circ \mathbf{G}^1 \right]^T \mathbf{X}, \\ \frac{\partial S_\alpha(\mathbf{K}^1)}{\partial \mathbf{b}^1} &= -4 \left[(\mathbf{D}^1 \mathbf{X}^1) \circ \mathbf{G}^1 \right]^T \mathbf{1}, \end{aligned}$$

where $\mathbf{A}^2 = \mathbf{D}^2 \mathbf{X}^2$, $\mathbf{B}^2 = \mathrm{diag}(\mathrm{diag}(\mathbf{D}^2 \mathbf{X}^2 \mathbf{X}^{2\mathrm{T}}))$, $\mathbf{G}^1 = \mathbf{g}^{1\prime}(\mathbf{W}^1 \mathbf{X} + \mathbf{b}^1)$, $\mathbf{D}^k = \mathrm{diag}(\mathbf{P}^k \mathbf{1}) - \mathbf{P}^k$ and $\mathbf{P}^k = \frac{\partial S_\alpha(\mathbf{K}^k)}{\partial \mathbf{K}^k} \circ \frac{1}{2\sigma_x^2} \mathbf{K}^k$. The trace normalized RBF kernels $\mathbf{K}$, $\mathbf{K}^1$ and $\mathbf{K}^2$ were computed using the infinitely divisible RBF

kernel based on $\mathbf{X}$, $\mathbf{X}^1$ and $\mathbf{X}^2$, respectively. By substituting $\mathbf{Q}^k = \frac{\partial S_\alpha(\mathbf{K},\mathbf{K}^k)}{\partial \mathbf{K}^k} \circ \frac{1}{2\sigma_x^2}\mathbf{K}^k$ for $\mathbf{P}^k$, we can get the gradient of joint entropies $S_\alpha(\mathbf{K},\mathbf{K}^k)$ with respect to each parameter.

For the rate-distortion DNN training, the cost function was $c(\mathbf{W}^1,\mathbf{b}^1,\mathbf{W}^2,\mathbf{b}^2) = S_\alpha(\mathbf{K}^1) + S_\alpha(\mathbf{K}^2) - S_\alpha(\mathbf{K},\mathbf{K}^1) - S_\alpha(\mathbf{K},\mathbf{K}^2) + \beta\mathbf{CE}(\mathbf{X}^2,\mathbf{Y})$, and we optimized it with stochastic gradient descent:

$$\min_{\mathbf{W}^1,\mathbf{b}^1,\mathbf{W}^2,\mathbf{b}^2} c(\mathbf{W}^1,\mathbf{b}^1,\mathbf{W}^2,\mathbf{b}^2). \tag{13}$$

4. EXPERIMENTS

To evaluate the effectiveness of the proposed algorithm, we performed our experiments on the Interactive Emotional Dyadic Motion Capture (IEMOCAP) database [16]. IEMOCAP consists of rich information about speech, facial expressions and hand gestures of ten actors in dyadic sessions. The actors were asked to perform selected emotional scripts and pre-defined improvised scenarios. There are five sessions in the corpus with two actors, one from each gender, in each session. The total amount of data in this modest sized corpus amounts to roughly 12 hours. For speech emotion recognition, we only considered the audio tracks labelled as one of the four categorical emotion types, including *Angry*, *Happy*, *Sad* and *Neutral* as they are the majority of the categorical emotion types, where the numbers of utterances in each category are 1103, 595, 1084 and 1708, respectively, with a sum of 4490. In the experiments, we followed a leave-one-speaker-out approach for cross validation. Specifically, we took four sessions as the training data, while in the remaining one session, one speaker is used for validation, model selection and parameter tuning and the other for testing.

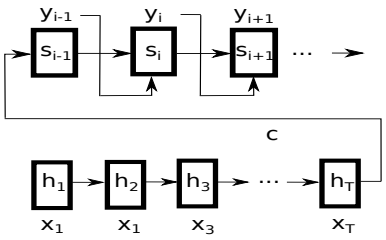

Fig. 2. A diagram of the encoder-decoder framework.

The low-level descriptors (LLDs) comprised of the 13-dimensional MFCC including the zero-th order coefficient, the pitch and their first order derivatives. Therefore, each frame had a dimensionality of 28. Since human emotion is context sensitive with long-range time dependencies [17], rather than working with these LLDs we extracted the utterance representations via the encoder-decoder framework [18, 19]. The encoder-decoder framework first maps an input sequence to an intermediate representation called the context vector, and forwards it to the decoder for it to generate the desired output. One advantage of the encoder-decoder framework is to model sequence-to-sequence learning, in particular, when the input and output sequence have different lengths. A schematic diagram of the encoder-decoder system is depicted in Fig.2, where $\mathbf{x}_i$ and $\mathbf{y}_j$ are input and output sequences, respectively, $\mathbf{h}_i$ and $\mathbf{s}_j$ are state outputs from the encoder and the decoder, respectively, and the context vector $\mathbf{c}$ serves to convey content and structural information of the input sequence from the encoder to the decoder. In practice, the encoder and the decoder are implemented via recurrent neural network (RNN) with a long short-term memory gating mechanism (LSTM). For the encoder, we used a bidirectional LSTM (BLSTM) to extracted the context vector $\mathbf{c}$. Because the emotion label has a length of one, a DNN is sufficient as the generator.

We trained a BLSTM-MLP encoder-decoder system to perform speech emotion recognition. The parameters were tuned based on cross validation in favor of the un-weighted accuracy (UA). The optimal architecture was found to be a BLSTM with 128 cells for each direction, and a MLP with a single hidden layer consisting of 128 neurons. The context vectors here is the time average of all hidden states of the encoder. The UA of the encoder-decoder system is 52.86%. Afterwards, we extracted the context vectors $\mathbf{c}$ as the input vectors in our experiment for rate-distortion based training of DNNs. The performance of the rate-distortion based DNN reached a comparable performance 52.54%. In the experiment we kept the value of α equal to 2. Other parameters were tuned by cross validation, including $\beta = 37$, $\sigma_x^2 = 0.2$, the number of neurons in the hidden layer being 64 and the batch size being 64.

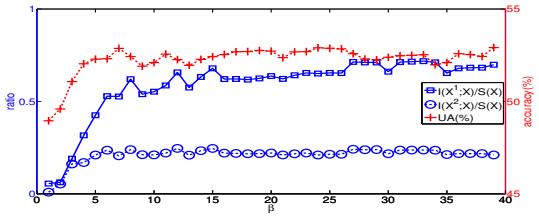

Fig. 3. The curves of UA and the ratio of the mutual information between representation $\mathbf{X}^k$ and $\mathbf{X}$ to the entropy of $\mathbf{X}$ for different values of the Lagrangian multiplier β

Fig. 3 summarizes the influence of the Lagrangian multiplier β on the UA and the amount of mutual information between the input and each layer. It is clear that when β is small, the compression effect is stronger and the ratios $I_\alpha(\mathbf{X}^k;\mathbf{X})/S_\alpha(\mathbf{K})$ remain small. Whereas when β increases the prediction power gradually increases accordingly at the cost of the compression power. Another observation is that the deeper the layer is the smaller the mutual information between the representation and the input, which is exactly the manifestation of the data processing inequality. This experiment empirically shows a trade-off between the compression and the prediction and the information flow between consecutive layers. The rate-distortion based analysis for recurrent neural network would be an interesting future work

to gain a deeper understanding behind the recent success in sequence-to-sequence learning.

5. CONCLUSION

We proposed a rate-distortion based DNN training algorithm based on the recently developed Renyi entropy estimation. In particular, our work focused on the continuous setting. The proposed training is shown effective and the trained DNN performed similar to an encoder-decoder system. For the experimental purpose, we extracted the intermediate context vectors from an encoder-decoder system as out input to a DNN training. In the future, we would like to study the rate-distortion based recurrent neural network training and to investigate the change of information flow over time. The minimization of information distortion in a sense serves as the regularization to the training, and it would be interesting to compare it with other techniques such as the $l1/l2$ or the dropout regularization.

6. REFERENCES

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

[2] Naftali Tishby and Noga Zaslavsky, "Deep learning and the information bottleneck principle," in *2015 IEEE Information Theory Workshop, ITW 2015, Jerusalem, Israel, April 26 - May 1, 2015*, 2015, pp. 1–5.

[3] Naftali Tishby, Fernando C. Pereira, and William Bialek, "The information bottleneck method," 1999, pp. 368–377.

[4] Pejman Khadivi, Ravi Tandon, and Naren Ramakrishnan, "Flow of information in feed-forward deep neural networks," in *arXiv:1603.06220*, 2016.

[5] Gal Chechik, Amir Globerson, Naftali Tishby, and Yair Weiss, "Information bottleneck for gaussian variables.," *Journal of Machine Learning Research*, vol. 6, pp. 165–188, 2005.

[6] Dávid Pál, Barnabás Póczos, and Csaba Szepesvári, "Estimation of renyi entropy and mutual information based on generalized nearest-neighbor graphs," in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, 2010, pp. 1849–1857.

[7] Lev Faivishevsky and Jacob Goldberger, "Ica based on a smooth estimation of the differential entropy," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., pp. 433–440. Curran Associates, Inc., 2009.

[8] A. Renyi, "On measures of information and entropy," in *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, 1960, pp. 547–561.

[9] Luis Gonzalo Sánchez Giraldo, Murali Rao, and José C. Príncipe, "Measures of entropy from data using infinitely divisible kernels," *IEEE Trans. Information Theory*, vol. 61, no. 1, pp. 535–548, 2015.

[10] Luis Gonzalo Sánchez Giraldo and Jose C. Principe, "Rate-distortion auto-encoders," *International Conference on Learning Representations*, 2014.

[11] Luis G. Sanchez Giraldo and Jose C. Principe, "Information theoretic learning with infinitely divisible kernels," in *arXiv:1301.3551*, 2013.

[12] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing*, Wiley-Interscience, 2006.

[13] Ohad Shamir, Sivan Sabato, and Naftali Tishby, "Learning and generalization with the information bottleneck," *Theor. Comput. Sci.*, vol. 411, no. 29-30, pp. 2696–2711, June 2010.

[14] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill Higher Education, 4 edition, 2002.

[15] Christopher M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Inc., New York, NY, USA, 1995.

[16] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette Chang, Sungbok Lee, and Shrikanth S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Journal of Language Resources and Evaluation*, vol. 42, no. 4, pp. 335–359, Dec. 2008.

[17] Angeliki Metallinou, Martin Wollmer, Athanasios Katsamanis, Florian Eyben, Bjorn Schuller, and Shrikanth Narayanan, "Context-sensitive learning for enhanced audiovisual emotion classification," *IEEE Trans. Affect. Comput.*, vol. 3, no. 2, pp. 184–198, Apr. 2012.

[18] Kyunghyun Cho, Bart Van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Oct. 2014, pp. 1724–1734.

[19] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, pp. 3104–3112. 2014.