

---

# Rate-Distortion Auto-Encoders

---

**Luis G. Sanchez Giraldo**

Dept. of Electrical and Computer Engineering  
University of Florida  
Gainesville, Florida, USA  
sanchez@cnel.ufl.edu

**Jose C. Principe**

Dept. of Electrical and Computer Engineering  
University of Florida  
Gainesville, Florida, USA  
principe@cnel.ufl.edu

## Abstract

Interest in auto-encoder algorithms has been rekindled by recent work in deep learning. Current efforts have been directed towards effective training of auto-encoder architectures with a large number of coding units. Here, we propose a learning algorithm for auto-encoders based on a rate-distortion objective that minimizes the mutual information between the inputs and the outputs of the auto-encoder subject to a fidelity constraint. The goal is to learn a representation that is minimally committed to the input data, but that is rich enough to reconstruct the inputs up to certain level of distortion. Minimizing the mutual information acts as a regularization term whereas the fidelity constraint can be understood as a risk functional in the conventional statistical learning setting. The proposed algorithm uses a recently introduced measure of entropy based on infinitely divisible matrices that avoids the plug in estimation of densities. Experiments using over-complete bases show that the rate-distortion auto-encoders can learn a regularized input-output mapping in an implicit manner.

## 1 Introduction

Auto-encoders are unsupervised learning algorithms that capture the structure in data by finding an internal representation of the input patterns (encoding) from which they can be reconstructed at least approximately. By learning a transformation  $G$  (encoding) from the input  $x \in \mathcal{X}$  to  $z = G(x) \in \mathcal{Z}$  the auto-encoder tries to capture the structure of the input. To guarantee the transformation  $G$  preserves the information about  $x$ , a decoder  $\tilde{G}^{-1}$  is also learned along the way, such that a measure of fidelity  $E[D(X, \tilde{X})]$  in the reconstruction is optimized. Perhaps, the most conventional class of auto-encoders are the bottle-neck networks that transform the input data  $x \in \mathcal{X} \subseteq \mathbb{R}^d$  to a lower dimensional space  $\mathcal{Z} \subseteq \mathbb{R}^p$ , and then transform it back to the input space  $\mathcal{X}$ . It is expected that the lower dimensional space forces the encoder to capture the relations between the variables in the input space. While this is clear when one is restricted to linear mappings, the problem becomes less well-understood if nonlinear mappings are allowed.

Recent progress in deep learning has reignited the interest on developing procedures to train auto encoders. Nevertheless, research efforts are now focused on how to effectively train auto-encoder architectures with a large number of encoding units, possibly larger than the number of inputs. In this case, it is necessary to develop algorithms that can avoid trivial solutions. Approaches like sparse encoding, which can be applied to over-complete scenarios,  $\dim(\mathcal{X}) < \dim(\mathcal{Z})$ , use the concept of effective dimensionality. A sparsity constraint induces an active set of variables with an expected  $L^0$  norm smaller than the input space dimensionality. In general, sparse coding procedures require solving an optimization problem at inference time since the encoding mapping is not explicit. However, it has been shown that efficient inference can be made possible by training a nonlinear feed-forward network to mimic the output of a sparse encoding algorithm [1].

Another recent approach focus on the idea of robustness. De-noising auto-encoders [2] learn a mapping by simply minimizing the error between the reconstruction of the auto-encoder when a

noisy version of the input is feed to the mapper. Let  $X$  be the random variable representing the input, and  $\hat{X}$  be the noisy version of it. The goal of the de-noising auto-encoder is to learn a map  $f(\cdot) = \tilde{G}^{-1}(G(\cdot))$  such that  $E[D(f(\hat{X}), X)]$  is minimized. Contractive auto-encoders [3, 4] are another way of enforcing robustness on the learned encoder by penalizing the sensitivity of the representation  $G(x)$  to the input  $x$ . This is achieved by minimizing the expected Frobenius norm of the Jacobian  $E[J_G(X)]$  of  $G$  while maximizing fidelity of the reconstruction.

Here, we adopt an information theoretic view of the problem. In all the above examples, implicitly or explicitly, the concept of compression plays a fundamental role. The idea of rate-distortion has been explored for nondeterministic mappings by [5]. In this case the structure of the data is represented by a set of  $N$  points located along a manifold. The algorithm produces random map  $p(\mu|x)$  from  $\mathcal{X}$  to the embedding of a manifold denoted by  $\mathcal{M}$ . The manifold is assumed to be a less faithful description of the data. The objective is to minimize a trade-off between fidelity and information content. The information content correspond to the mutual information between the data and the manifold representation:

$$I(X; \mathcal{M}) = \int_{x \in \mathcal{X}} \int_{\mu \in \mathcal{M}} P(x, \mu) \log \frac{P(x, \mu)}{p(x)P_{\mathcal{M}}(\mu)} dx \quad (1)$$

The locations of the points are adjusted according to the distortion measure. A Boltzmann distribution containing the fidelity measure arises as solution to the above objective. The update equations for an algorithm based on the above trade off correspond to an iterated projection between convex sets. Note however, that a solution of this form arises from the considerations of a random mapping from  $\mathcal{X}$  to the embedded manifold.

Here, we propose an algorithm that incorporates the idea to learn a parametrized mapping, encoder and decoder function, that optimizes a trade off between compression and fidelity in the reconstruction. The intuition behind this procedure is that the auto-encoder should avoid trivial solutions if it is forced to map the input data to a compressed version of it.

## 2 The rate-distortion function

For the transmission of information through a channel, it was shown by Shannon that the noise and bandwidth inherent to the channel determine the rate at which information can be transferred, and that any attempt to transfer information above the limit imposed by the channel will incur in some error in the recovered message. This result known as the channel capacity problem is concerned with achieving zero error in the transmission. A complementary problem is to determine what is the minimum possible rate at which information can be transferred such that the error in the recovered message does not surpasses a certain level. Answering this question gives rise to the rate-distortion function. The information rate-distortion function  $R(D)$  is defined as:

$$R(D) = \min_{q(\hat{x}|x)} I(X; \hat{X}) \text{ subject to } E[d(X, \hat{X})] \leq D \quad (2)$$

where  $d(\cdot, \cdot)$  is a distortion function that can be related to the loss incurred when the exact value of the input cannot be recovered from the output and  $E[d(X, \hat{X})] = \sum_{x, \hat{x}} p(x)q(\hat{x}|x)d(x, \hat{x})$ . The above problem (2) is a variational problem for which we search for  $q(\hat{x}|x) \geq 0$  subject to the regularity condition:

$$\sum_{\hat{x} \in \mathcal{X}} q(\hat{x}|x) = 1 \text{ for all } x \in \mathcal{X} \quad (3)$$

This is a standard minimization problem of a convex function over a convex set [6] with a know numerical solution in the case of finite input/output alphabets. How could this function be utilized to find a representation of the input variables?

### 2.1 Motivation Example: rate-distortion and PCA

The relation between rate-distortion and PCA arises from the assumption of a Gaussian source. Consider the Gaussian random vector  $X \in \mathbb{R}^d$  with zero mean and covariance  $\Sigma$ . We can find similarity transformation  $\mathbf{U}$  to a Gaussian random vector  $Z \in \mathbb{R}^d$  with zero mean and diagonal covariance matrix  $\Gamma$ , such that  $\mathbf{U}^T \Sigma \mathbf{U} = \Gamma$ . The transformation matrix  $\mathbf{U}$  is unitary which implies  $\det(\mathbf{U}) = 1$ ,

and thus,  $h(X) = h(Z)$  and the mapping is a bijection<sup>1</sup>. The following theorem from [7] is used on stating the result.

**Theorem 2.1 (Rate distortion for a parallel Gaussian source):** *Let  $X_i \sim \mathcal{N}(0, \gamma_i^2)$ ,  $i = 1, \dots, d$ , be independent Gaussian random variables, and the distortion measure  $d(x, \hat{x}) = \sum_{i=1}^d (x_i - \hat{x}_i)^2$ . Then the rate distortion function is given by*

$$R(D) = \sum_{i=1}^d \frac{1}{2} \log \frac{\gamma_i}{D_i} \quad (4)$$

where

$$D_i = \begin{cases} \lambda & \text{if } \lambda < \gamma_i^2 \\ \gamma_i^2 & \text{if } \lambda \geq \gamma_i^2 \end{cases} \quad (5)$$

and  $\lambda$  is chosen so that  $\sum_{i=1}^d D_i = D$ .

Applying the result to the random vector  $Z$  yields the conclusion that the rate for components with variance less than  $\lambda$  will be zero, and thus, no effort in representing such components should be made. This is the way components are selected in PCA where a variance threshold is set and only those components above the threshold are retained. On the other hand,  $\lambda$  plays the role of the observation noise in the generative model for PCA, but the main difference is that  $\hat{Z}$  is not assumed to be  $\mathcal{N}(0, \mathbf{I})$ . Note however that for the above conclusion, the similarity transformation was given in advance rather than derived from the rate-distortion objective. Our goal is to learn a map  $f: \mathcal{X} \mapsto \mathcal{X}$  based on the principle of minimization of mutual information. The numerical procedure known as the Blahut-Arimoto algorithm provides a means to compute a stochastic map  $q(\hat{x}|x)$  that achieves the minimum rate under a given distortion constraint. Nevertheless, this method was developed for finite input/output alphabets. We are interested in an approximation that uses a deterministic map  $f(\cdot)$  on continuous input/output spaces.

## 2.2 Rate-distortion as an objective for learning a map

Shannon's definition of mutual information can be decomposed in terms of the marginal and conditional entropies as:

$$I(X; Y) = H(X) - H(X|Y) \quad (6)$$

where  $H(X)$  and  $H(X|Y)$  are the marginal and conditional entropies of  $X$  and  $X$  given  $Y$ . Since the entropy of the input  $X$  is constant, we only need to consider  $H(X|Y)$  when dealing with maps from  $X$  to  $Y$ . We can thus reformulate (2) in terms of the conditional entropy as:

$$\underset{f(X)=\hat{X}}{\text{maximize}} H(X|\hat{X}) \text{ subject to } \mathbb{E}[d(X, \hat{X})] \leq D. \quad (7)$$

Writing the Lagrangian of (7),

$$\mathcal{L}(f, \mu) = \mathbb{E}[d(X, \hat{X})] - \frac{1}{\mu} H(X|\hat{X}), \quad (8)$$

where  $\mu$  determines the distortion level  $D$ , shows that the objective can be understood as a regularized risk minimization problem. In (8),  $d(X, \hat{X})$  plays the role of the loss functional and  $H(X|\hat{X})$  acts as a regularization parameter. However, this form of regularization is not explicit on the parameters of the function  $f(\cdot)$ . Informally, we can see that maximizing the conditional entropy  $H(X|\hat{X}) = H(X; \hat{X}) - H(\hat{X})$  can have the effect of lowering the entropy of the output variable  $\hat{X}$ , and thus, we can think of the mapping  $f$  as a contraction.

## 3 Rate-Distortion Auto-Encoder Algorithm

On the previous section, we motivate the use of the rate-distortion objective to learn a map that is minimally committed in terms of retaining information about the inputs, yet able to provide a reconstruction with a desired level of distortion. However, solving the above objective function

<sup>1</sup>Note, we are not using mutual information to relate  $X$  and  $Z$ , cause in this case it is infinite

would require the distribution of  $X$  to be known. Since we only have access to a set of *i.i.d.* samples  $\{\mathbf{x}_i\}_{i=1}^N$  drawn from  $P_X(x)$ , a suitable estimator of both conditional entropy, and expected distortion is required. For the expected distortion we employ the empirical estimator

$$D_{\text{emp}}(f) = \frac{1}{N} \sum_{i=1}^N d(\mathbf{x}_i, f(\mathbf{x}_i)). \quad (9)$$

For the conditional entropy term, we use an alternative definition of entropy introduced in [8].

### 3.1 Matrix-Based Entropy Functional

Let  $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N \subset \mathcal{X}$  and  $\kappa: \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$  be a real valued positive definite kernel that is also infinitely divisible. The Gram matrix  $K$  obtained from evaluating a positive definite kernel  $\kappa(\cdot, \cdot)$  on all pairs of exemplars, that is  $(K)_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ , can be employed to define a quantity with properties similar to those of an entropy, for which the probability distribution of  $X$  does not need to be estimated.

A matrix-based analogue to Renyi's  $\alpha$ -entropy for a positive definite matrix  $A$  of size  $N \times N$ , such that  $\text{tr}(A) = 1$ , is given by the functional:

$$S_\alpha(A) = \frac{1}{1-\alpha} \log_2 [\text{tr}(A^\alpha)] = \frac{1}{1-\alpha} \log_2 \left[ \sum_{i=1}^N \lambda_i(A)^\alpha \right], \quad (10)$$

where  $\lambda_i(A)$  denotes the  $i$ th eigenvalue of the  $A$  matrix<sup>2</sup>. The matrix-based entropy estimate from the sample  $\mathbf{X}$  can be obtained by evaluating (10) on following normalized version of  $K$ :

$$A_{ij} = \frac{1}{N} \frac{K_{ij}}{\sqrt{K_{ii}K_{jj}}}. \quad (11)$$

Furthermore, based on the the product kernel, we can define the notion of joint-entropy as:

$$S_\alpha \left( \frac{A \circ B}{\text{tr}(A \circ B)} \right), \quad (12)$$

where  $A \circ B$  denotes the Hadamard product between the matrices  $A$  and  $B$ . For a set of pairs  $\{(x_i, y_i)\}_{i=1}^n$  such that  $x_i \in \mathcal{X}$  and  $y_i \in \mathcal{Y}$ , let  $\kappa_{\mathcal{X}}$  and  $\kappa_{\mathcal{Y}}$  be positive definite kernels defined on  $\mathcal{X} \times \mathcal{X}$  and  $\mathcal{Y} \times \mathcal{Y}$ , respectively. If  $A_{ij} = \kappa_{\mathcal{X}}(x_i, x_j)$  and  $B_{ij} = \kappa_{\mathcal{Y}}(y_i, y_j)$ , the Hadamard product  $A \circ B$  is nothing but the Gram matrix using the product kernel  $\kappa((x_i, y_i), (x_j, y_j)) = \kappa_{\mathcal{X}}(x_i, x_j) \cdot \kappa_{\mathcal{Y}}(y_i, y_j)$  [9]. It can also be shown that if the Gram matrices  $A$  and  $B$  are constructed using normalized infinitely divisible kernels (based on (11)), such that  $A_{ii} = B_{ii} = 1/n$ , (12) is never smaller than any of the individual entropies  $S_\alpha(A)$  or  $S_\alpha(B)$ . This allows us to define a matrix notion of conditional entropy as follows:

$$S_\alpha(A|B) = S_\alpha \left( \frac{A \circ B}{\text{tr}(A \circ B)} \right) - S_\alpha(B), \quad (13)$$

### 3.2 Maximum Matrix-Based Conditional Entropy Auto-encoder

Using the kernel matrix measure of entropy, we can formulate an objective function based on (13). For a set of points  $\{\mathbf{x}_i\}_{i=1}^n$ , where  $\mathbf{x}_i \in \mathbb{R}^{d_x}$  we define a parametrized encoding mapping  $\hat{\mathbf{z}}_i = G_{\mathbf{W}, \mathbf{c}}(\mathbf{x}_i) = g(\mathbf{W}\mathbf{x}_i + \mathbf{c})$ , as well as a decoding mapping  $\tilde{\mathbf{x}}_i = \tilde{G}_{\mathbf{A}, \mathbf{b}}^{-1}(\hat{\mathbf{z}}_i) = \mathbf{A}\hat{\mathbf{z}}_i + \mathbf{b}$ . Notice that there is no imposed constraint on the mapping such as tied weights or explicit weight decay in our formulation. Let  $\mathbf{K}_X$  and  $\mathbf{K}_{\hat{X}}$  denote the normalized Gram matrices for  $\{\mathbf{x}_i\}_{i=1}^n$  and  $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ , respectively. Our goal is to find the set of pairs  $(\mathbf{W}, \mathbf{c})$  and  $(\mathbf{A}, \mathbf{b})$  that maximize the matrix-based conditional entropy of the inputs  $\mathbf{x}_i$  given the reconstructions  $\tilde{\mathbf{x}}_i = \tilde{G}_{\mathbf{A}}^{-1}(G_{\mathbf{W}}(\mathbf{x}_i))$ . This can be posed as the following optimization problem:

$$\begin{aligned} & \underset{(\mathbf{W}, \mathbf{c}, \mathbf{A}, \mathbf{b}) \in \Theta}{\text{maximize}} && S_\alpha(\mathbf{K}_X | \mathbf{K}_{\hat{X}}) \\ & \text{subject to} && \frac{1}{N} \sum_{i=1}^N d(\mathbf{x}_i, \tilde{\mathbf{x}}_i) \leq D. \end{aligned} \quad (14)$$

<sup>2</sup>All eigenvalues are considered along with their multiplicities

Let  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)^T$  denote the input data matrix,  $\mathbf{Z} = \mathbf{X}\mathbf{W}^T + \mathbf{1}_N\mathbf{c}^T$  be the affine transform of the input,  $\hat{\mathbf{Z}} = g(\mathbf{Z})$  denote the encoder output after the nonlinearity  $g(\cdot)$ ,  $\mathbf{G}' = g'(\mathbf{Z})$  be the derivatives of the encoder nonlinearity evaluated at  $\mathbf{Z}$ , and  $\hat{\mathbf{X}} = \hat{\mathbf{Z}}\mathbf{A}^T + \mathbf{1}_N\mathbf{b}^T$  be the decoder output. The gradients of  $S_\alpha(\mathbf{K}_{\hat{\mathbf{X}}})$  and  $S_\alpha(N\mathbf{K}_X \circ \mathbf{K}_{\hat{\mathbf{X}}})$  with respect to  $\mathbf{K}_{\hat{\mathbf{X}}}$  are given by:

$$\frac{\partial S_\alpha(\mathbf{K}_{\hat{\mathbf{X}}})}{\partial \mathbf{K}_{\hat{\mathbf{X}}}} = \frac{\alpha}{(1-\alpha)\text{tr}(\mathbf{K}_{\hat{\mathbf{X}}}^\alpha)} \mathbf{U}\mathbf{\Lambda}^{\alpha-1}\mathbf{U}^T \text{ and} \quad (15)$$

$$\frac{\partial S_\alpha(N\mathbf{K}_X \circ \mathbf{K}_{\hat{\mathbf{X}}})}{\partial \mathbf{K}_{\hat{\mathbf{X}}}} = \frac{\alpha}{(1-\alpha)\text{tr}[(N\mathbf{K}_X \circ \mathbf{K}_{\hat{\mathbf{X}}})^\alpha]} N\mathbf{K}_X \circ (\mathbf{V}\mathbf{\Gamma}^{\alpha-1}\mathbf{V}^T), \quad (16)$$

where  $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$  and  $\mathbf{V}\mathbf{\Gamma}\mathbf{V}^T$  are the eigenvalue decompositions of  $\mathbf{K}_{\hat{\mathbf{X}}}$  and  $N\mathbf{K}_X \circ \mathbf{K}_{\hat{\mathbf{X}}}$ , respectively. In our work, we use the normalized Gaussian kernel,  $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{n} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_x^2}\right)$ , for both input and output points. The partial derivatives of the conditional entropy with respect to the parameters of the auto-encoder are given by:

$$\frac{\partial S_\alpha(\mathbf{K}_{\hat{\mathbf{X}}})}{\partial \mathbf{A}} = -4(\mathbf{A}\hat{\mathbf{Z}}^T) [\mathcal{D}(\mathbf{P}) - \mathbf{P}] \hat{\mathbf{Z}}, \quad (17)$$

$$\frac{\partial S_\alpha(\mathbf{K}_{\hat{\mathbf{X}}})}{\partial \mathbf{b}} = \mathbf{0}, \quad (18)$$

$$\frac{\partial S_\alpha(\mathbf{K}_{\hat{\mathbf{X}}})}{\partial \mathbf{W}} = -4 \left( (\mathbf{A}^T \mathbf{A} \hat{\mathbf{Z}}^T [\mathcal{D}(\mathbf{P}) - \mathbf{P}]) \circ \mathbf{G}'^T \right) \mathbf{X}, \quad (19)$$

$$\frac{\partial S_\alpha(\mathbf{K}_{\hat{\mathbf{X}}})}{\partial \mathbf{c}} = -4 \left( (\mathbf{A}^T \mathbf{A} \hat{\mathbf{Z}}^T [\mathcal{D}(\mathbf{P}) - \mathbf{P}]) \circ \mathbf{G}'^T \right) \mathbf{1}_N, \quad (20)$$

where  $\mathbf{P} = \frac{\partial S_\alpha(\mathbf{K}_{\hat{\mathbf{X}}})}{\partial \mathbf{K}_{\hat{\mathbf{X}}}} \circ \frac{1}{2\sigma_x^2} \mathbf{K}_{\hat{\mathbf{X}}}$  and  $\mathcal{D}(\mathbf{P}) = \text{diag}(\mathbf{P}\mathbf{1}_N)$ . The partial derivatives of the joint entropy

term have the same form of the above, we only need to replace  $\mathbf{P}$  by  $\mathbf{Q} = \frac{\partial S_\alpha(N\mathbf{K}_X \circ \mathbf{K}_{\hat{\mathbf{X}}})}{\partial \mathbf{K}_{\hat{\mathbf{X}}}} \circ \frac{1}{2\sigma_x^2} \mathbf{K}_{\hat{\mathbf{X}}}$ .

From the above set of derivatives, (17), (19), and (20), we can see that the term  $\mathcal{D}(\mathbf{P}) - \mathbf{P}$  appears in all of them. Noticing that  $\mathbf{P}$  is positive definite and that indeed can be regarded as an affinity matrix, we can think of  $\mathcal{D}(\mathbf{P}) - \mathbf{P}$  as the graph Laplacian, where  $\mathbf{P}$  is the weight matrix for the set of edges and  $\mathcal{D}(\mathbf{P})$  the degree of the graph. Notice also that  $\mathbf{P}$  corresponds to a data dependent kernel since its values depend on the spectrum of  $\mathbf{K}_{\hat{\mathbf{X}}}$ .

The derivatives with respect to the distortion measure  $D_{\text{emp}}$  are given by:

$$\frac{\partial D_{\text{emp}}}{\partial \mathbf{A}} = -\frac{2}{N} (\mathbf{X} - \hat{\mathbf{X}})^T \hat{\mathbf{Z}}, \quad (21)$$

$$\frac{\partial D_{\text{emp}}}{\partial \mathbf{b}} = -\frac{2}{N} (\mathbf{X} - \hat{\mathbf{X}})^T \mathbf{1}_N, \quad (22)$$

$$\frac{\partial D_{\text{emp}}}{\partial \mathbf{W}} = -\frac{2}{N} [((\mathbf{X} - \hat{\mathbf{X}})\mathbf{A}^T) \circ \mathbf{G}']^T \mathbf{X}, \quad (23)$$

$$\frac{\partial D_{\text{emp}}}{\partial \mathbf{c}} = -\frac{2}{N} [((\mathbf{X} - \hat{\mathbf{X}})\mathbf{A}^T) \circ \mathbf{G}']^T \mathbf{1}_N, \quad (24)$$

The above derivatives can be employed to search for the parameters for instance using gradient ascent. In our work we implemented gradient ascent for the following Lagrangian:

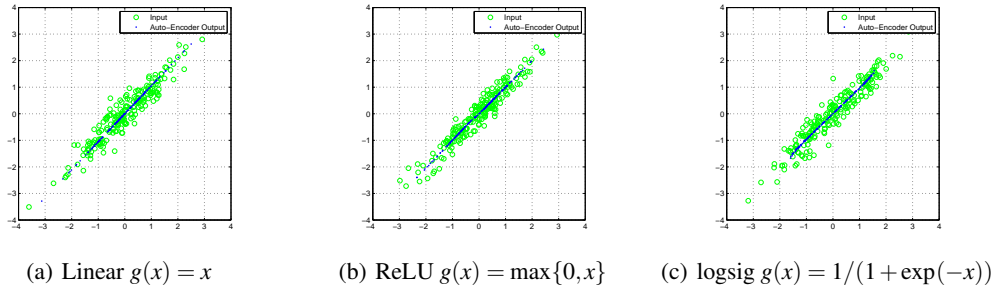
$$\mathcal{L}(\mathbf{W}, \mathbf{c}, \mathbf{A}, \mathbf{b}) = S_\alpha(N\mathbf{K}_X \circ \mathbf{K}_{\hat{\mathbf{X}}}) - S_\alpha(\mathbf{K}_{\hat{\mathbf{X}}}) - \mu \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2, \quad (25)$$

where  $\mu$  is fixed depending on the desired distortion level<sup>3</sup>.

## 4 Experiments

This section describe some of the experiments we have carried out with the rate-distortion auto-encoders. We qualitatively illustrate the regularization property of the rate-distortion auto-encoder

<sup>3</sup>Note that by fixing the value of  $\mu$  we are implicitly enforcing an equality constraint on  $D_{\text{emp}}$  rather than an inequality constraint since the positive multiplier represents an active constraint.



**Figure 1. Outputs for different activation functions of the rate-distortion auto-encoder with an over-complete representation when the inputs are Gaussian distributed.**

with a simple set of examples involving an over-complete representations. First, we show how the rate-distortion objective regularizes different types of units implicitly. Algorithms such as contractive auto-encoders rely on explicit calculation of the Jacobian, which depends on the type of units selected. Following these experiments, we show some resulting basis after learning on the MNIST dataset.

#### 4.1 Synthetic Data

**Gaussian distributed data:** The first example corresponds to a set of data points drawn from a bivariate Gaussian distribution with zero mean and covariance matrix

$$\Sigma_x = \begin{pmatrix} 1 & 0.95 \\ 0.95 & 1 \end{pmatrix}. \quad (26)$$

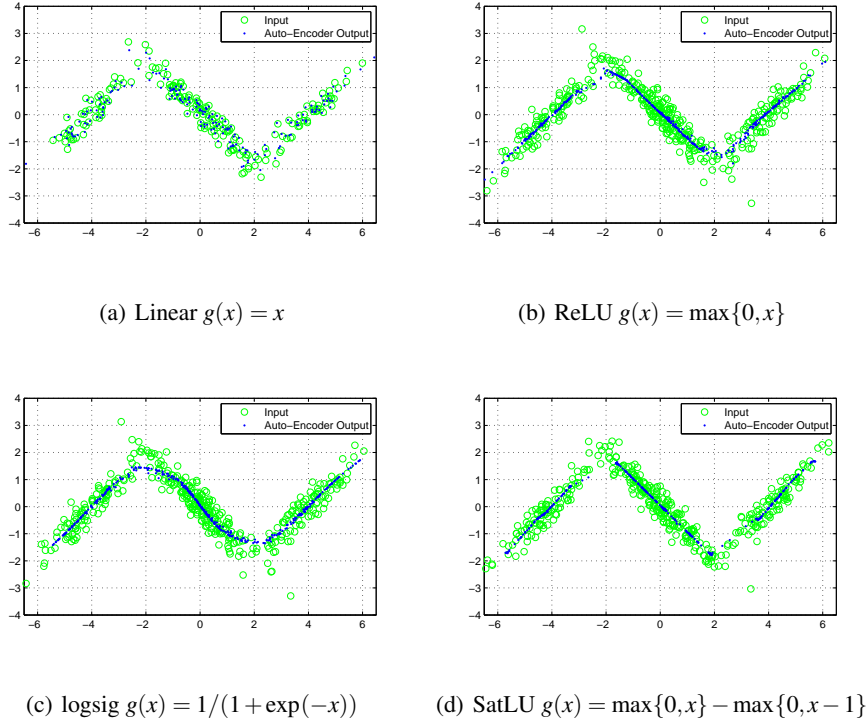
We use a linear encoding, that is  $g(x) = x$ , that is also over-complete since the encoder projects the 2-dimensional data points to 10 different directions. The conventional auto-encoder would overfit by being able to achieve zero reconstruction error, but it won't be able to implicitly retain what is thought to be the structure in the data. We also compare this output to the outputs of two nonlinear auto-encoders, one uses the logsig units  $g(x) = 1/(1 + \exp(-x))$ , and the other a rectified linear units (ReLU)  $g(x) = \max\{0, x\}$ . Figure 1 shows the outputs of the three auto-encoders on the Gaussian distributed data. It can be seen that the outputs approximately align with what corresponds roughly to the first principal component of the data. Notice, that no bottleneck or shrinkage was explicitly defined. The parameters of our cost function are  $\mu = 1.5$  for the distortion trade-off, and  $\sigma_x = 0.2\sqrt{2} = \sigma_{\hat{x}}$  for the kernel size.

**Mixture of Gaussians:** The second example, employs a mixture of three Gaussian distributions to show the output of the rate-distortion auto-encoder in a nonlinear scenario, where an over-complete representation followed by a nonlinearity can be advantageous. The means and covariances of the mixture components are

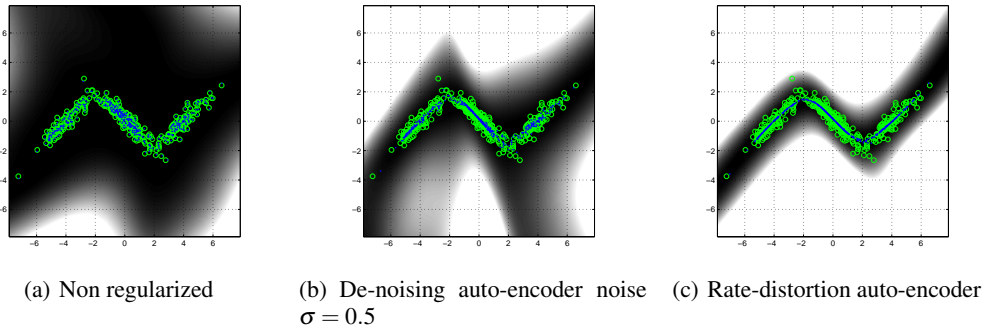
$$\begin{aligned} \mu_1 &= \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \mu_2 = \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \mu_3 = \begin{pmatrix} 6 \\ -2 \end{pmatrix}; \text{ and} \\ \Sigma_1 &= \begin{pmatrix} 1 & -0.95 \\ -0.95 & 1 \end{pmatrix}, \Sigma_2 = \Sigma_3 = \begin{pmatrix} 1 & 0.95 \\ 0.95 & 1 \end{pmatrix}, \end{aligned} \quad (27)$$

respectively, and the mixing weights are  $p_1 = 0.5$ , and  $p_2 = p_3 = 0.25$ .

Figure 2 shows the outputs of the four auto-encoders on the mixture of Gaussian distributions. The auto-encoders employ: linear, rectified linear, sigmoidal, and saturated linear units. It can be seen that the outputs approximately align with what can be thought of as the principal curves of the data. Again, we want to stress that no bottleneck or shrinkage was explicitly defined. In this case each of the auto-encoders has 20 units for encoding, which would easily overfit the data in the absence of any regularization or ad-hoc constraints such as tied weights. The parameters of our cost function are  $\mu = 1.5$  for the distortion trade-off, and  $\sigma_x = 0.2\sqrt{2} = \sigma_{\hat{x}}$  for the kernel size. The linear units seem to fit the data, but as we previously mentioned they favor the principal components. Lowering the value of  $\mu$  would collapse the reconstructed points into a line. This is not necessarily



**Figure 2. Outputs for different activation functions of the rate-distortion auto-encoder with an over-complete representation when the inputs are a mixture of Gaussian distributions.**



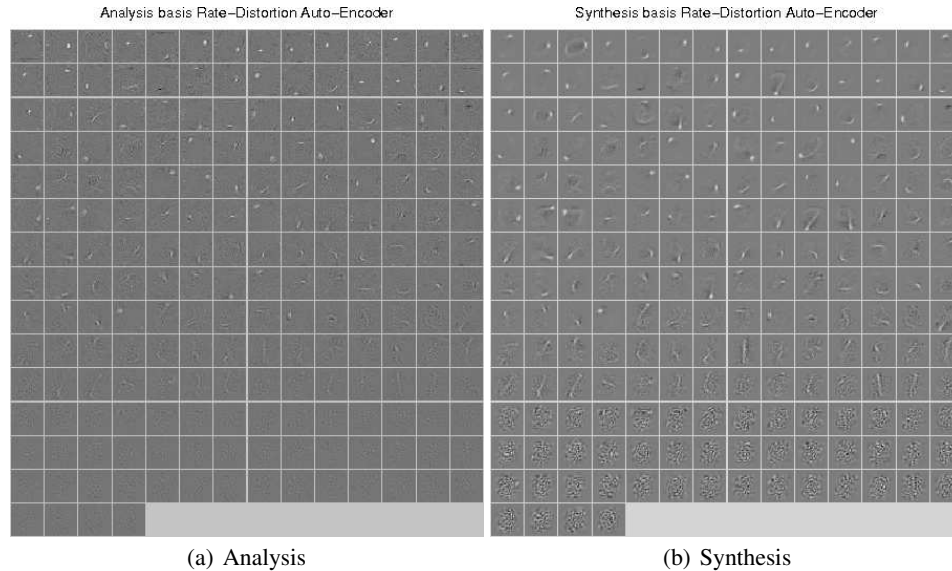
**Figure 3. Energy  $\|x - \hat{x}\|^2$  landscapes for different auto-encoder algorithms**

the case when nonlinear units are considered.

Finally, in Figure 3 we show the resulting energy  $\|x - \hat{x}\|^2$  landscapes for the over-complete auto-encoder with soft rectified linear units after being trained with: Non-regularized, de-noising auto-encoder using isotropic Gaussian noise, and the proposed rate distortion function. The rate-distortion objective makes the auto-encoder carve well-defined ravines in the energy landscape at the points where majority of the data lies.

## 4.2 Handwritten Digits

Here we use a subset of 20000 samples from MNIST to train a rate-distortion auto-encoder. Unlike conventional stochastic gradient, that uses mini-batches of randomly sample data to compute the



**Figure 4. Analysis and Synthesis basis encountered using the Rate-Distortion Auto-Encoder algorithm**

estimate of the gradient at each step, we define mini-batches by pre-clustering the data. The reason behind this procedure is that the Gram matrix employed to compute the entropy measure is approximately block diagonal after reordering the samples by clustering. Therefore, the influence of the eigenvectors would be also local around the defined clusters. At first glance, this approach seems to add the computational overhead of the clustering. However, there are cases where samples are naturally clustered, for example, sequences of images or processes that are assumed to be piecewise stationary. Figure 4 shows an example of the learned analysis and synthesis basis, after training an auto-encoder with 200 logsig units in the representation layer (hidden layer). The trade off parameter has been set to 0.5 and the kernel size for the entropy measure is 8.4. Pixels values were scaled from 0 to 1 and then data matrix was centered. In the figure, it can be seen how the auto-encoder not only learns localized blobs as with tied weights and weight decay, but it also learns pen strokes without a explicit sparsity constraint or predefined corruption model as presented in the work on de-noising auto-encoders. Varying the trade off parameter gives different regimes in the resulting features. For smaller reconstruction errors features tend to be more localized, whereas for larger tolerance the features go from blobs, to pen strokes, to even full digits. Weight decay or adding Gaussian noise to the training set can be used to zero-out weights for non active inputs. Recall that no tied-weights were employed.

## 5 Conclusions

We presented an algorithm for auto-encoders based on a rate-distortion objective that tries to minimize the mutual information between the inputs and outputs subject to a fidelity constraint. As a motivation example, we showed that for multivariate Gaussian distributed data, PCA can be understood as an optimal mapping in the rate-distortion sense. Moreover, we described how the rate-distortion optimization problem can be understood as a learning objective where the fidelity constraint plays the role of a risk functional and the mutual information acts as a regularization term. To provide a training algorithm, we employed a recently introduced measure of entropy based on infinitely divisible matrices that avoids the plug in estimation of densities. Experiments using over-complete bases showed that the auto-encoder was able to learn useful encoding mapping (representation) and can also learn a regularized input-output in an implicit manner. As future work, we would like to investigate the relation between the information theoretic objective and the graph Laplacian that arose from the parameter update rules.



## References

- [1] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model," in *Neural Information Processing Systems*, 2006.
- [2] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1096–1103. [Online]. Available: <http://doi.acm.org/10.1145/1390156.1390294>
- [3] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin, and X. Glorot, "Higher order contractive auto-encoder," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2011.
- [4] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *28th International Conference on Machine Learning*, 2011.
- [5] D. Chigirev and W. Bialek, "Optimal manifold representation of data: An information theoretic approach," in *Neural Information Processing Systems*, 2004.
- [6] T. Berger, *Rate Distortion Theory: Mathematical Basis for Data Compression*. Prentice-Hall, 1971.
- [7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley-Interscience, 2006.
- [8] L. G. Sanchez Giraldo and J. C. Principe, "Information theoretic learning with infinitely divisible kernels," in *International Conference on Learning Representations*, 2013.
- [9] C. Berg, J. P. R. Christensen, and P. Ressel, *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*, ser. Graduate Texts in Mathematics. Springer-Verlag, 1984, no. 100.