Analyzing the Dynamics of Gated Auto-encoders

Daniel Jiwoong Im
School of Engineering
University of Guelph
Guelph, Ontario, Canada N1G2W1
imj@uoguelph.ca

Graham W. Taylor
School of Engineering
University of Guelph
Guelph, Ontario, Canada N1G2W1
gwtaylor@uoguelph.ca

Abstract

Auto-encoders are perhaps the best-known non-probabilistic methods for representation learning. They are conceptually simple and easy to train. Recent theoretical work has shed light on their ability to capture manifold structure, and drawn connections to density modeling. This has motivated researchers to seek ways of auto-encoder scoring, which has furthered their use in classification. Gated auto-encoders (GAEs) are an interesting and flexible extension of auto-encoders which can learn transformations among different images or pixel covariances within images. However, they have been much less studied, theoretically or empirically. In this work, we apply dynamical systems view to GAEs, deriving a means of GAE scoring, and drawing connections to RBMs and score matching. Experimenting on a set of deep learning benchmarks, we also demonstrate their effectiveness for classification.

1 Introduction

Representation learning algorithms are machine learning algorithms which involve the learning of features or explanatory factors. Deep learning techniques, which employ several layers of representation learning, have achieved much recent success in machine learning benchmarks and competitions, however, most of these successes have been achieved with purely supervised learning methods and have relied on large amounts of labeled data [8, 15]. Though progress has been slower, it is likely that unsupervised learning will be important to future advances in deep learning [1].

The most successful and well-known example of non-probabilistic unsupervised learning is the auto-encoder. Conceptually simple and easy to train via backpropagation, various regularized variants of the model have recently been proposed [13, 18, 14] as well as theoretical insights into their operation [17, 4].

In practice, the latent representation learned by auto-encoders has typically been used to solve a secondary problem, often classification. The most common setup is to train a single auto-encoder on data from all classes and then a classifier is tasked to discriminate among classes. However, this contrasts with the way probabilistic models have typically been used in the past: in that literature, it is more common to train one model per class and use Bayes' rule for classification. There are two challenges to classifying using per-class auto-encoders. First, up until very recently, it was not known how to obtain the score of data under an auto-encoder, meaning how much the model "likes" an input. Second, auto-encoders are non-probabilistic, so even if they can be scored, the scores do not integrate to 1 and therefore the per-class models need to be calibrated.

Kamyshanska and Memisevic [6] have recently shown how scores can be computed from an auto-encoder by interpreting it as a dynamical system. Although the scores do not integrate to 1, they show how one can combine the unnormalized scores into a generative classifier by learning class-specific normalizing constants from labeled data.

In this paper we turn our interest towards a variant of auto-encoders which are capable of learning higher-order features from data [10]. The main idea is to learn relations between pixel intensities rather than the pixel intensities themselves by structuring the model as a tri-partite graph which connects hidden units to pairs of images. If the images are different, the hidden units learn how the images transform. If the images are the same, the hidden units encode within-image pixel covariances. Learning such higher-order features can yield improved results on recognition and generative tasks.

We adopt a dynamical systems view of gated auto-encoders, demonstrating that they can be scored similarity to the classical auto-encoder. We develop theory which yields insights into the operation of gated auto-encoders, and we also show in our experiments that a classification model which uses gated auto-encoder scoring can outperform a number of other representation learning architectures, including scored classical auto-encoders.

2 Gated Auto-encoders

In this section, we review the gated auto-encoder (GAE). Due to space constraints, we will not review the classical auto-encoder. Instead, we direct the reader to the reviews in [10, 6] with which we share notation. Similar to the classical auto-encoder, the GAE consists of an encoder $h(\cdot)$ and decoder $r(\cdot)$. While the standard auto-encoder processes input $\mathbf{x}$, the GAE processes pairs of datapoints $(\mathbf{x}, \mathbf{y})$. The GAE is usually trained to reconstruct $\mathbf{y}$ given $\mathbf{x}$, though it can also be trained symmetrically, that is, to reconstruct both $\mathbf{x}$ from $\mathbf{y}$ and $\mathbf{y}$ from $\mathbf{x}$. Intuitively, the GAE learns *relations* between the inputs, rather than representations of the inputs themselves[1]. If $\mathbf{x} \neq \mathbf{y}$, for example, they represent sequential frames of a video, intuitively, the mapping units $\mathbf{h}$ learn *transformations*. In the case that $\mathbf{x} = \mathbf{y}$ (i.e. the input is copied), the mapping units learn pixel covariances.

In the simplest form of the GAE, the M hidden (mapping) units are given by a basis expansion of $\mathbf{x}$ and $\mathbf{y}$. However, this leads to a parameterization that it is at least quadratic in the number of inputs and thus, prohibitively large. Therefore, in practice, $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{h}$ are projected onto matrices or ("latent factors"), W^X, W^Y, and W^M, respectively. To further reduce the parameterization, we set $W^X = W^Y = W^F$, effectively learning a single set of parameters for the inputs. The number of factors, F, must be the same for X, Y, and H. Thus, the model is completely parameterized by $\theta = \{W^F, W^M\}$ such that W^F is an $F \times D$ matrix and W^M is an $M \times F$ matrix. The encoder function is defined by

$$h(\mathbf{x}, \mathbf{y}) = \sigma(W^M((W^F\mathbf{x}) \odot (W^F\mathbf{y}))) \tag{1}$$

where $\odot$ is element-wise multiplication and $\sigma(\cdot)$ is an activation function. The decoder function is defined by

$$r(\mathbf{x}|\mathbf{y}, h) = (W^F)^T((W^F\mathbf{y}) \odot (W^M)^T h(\mathbf{x}, \mathbf{y})), \tag{2}$$

$$r(\mathbf{y}|\mathbf{x}, h) = (W^F)^T((W^F\mathbf{x}) \odot (W^M)^T h(\mathbf{x}, \mathbf{y})). \tag{3}$$

Note that we have again reduced parameters by sharing the filters W^F and W^M between the encoder and decoder. However, these can be de-coupled and separate sets of filters learned. The choice of whether to apply a nonlinearity to the output, and the specific form of objective function will depend on the nature of the inputs, for example, binary, categorical, or real-valued. Here, we have assumed real-valued inputs for simplicity of presentation, therefore, Eqs. 2 and 3 are bi-linear functions of $\mathbf{h}$ and we use a squared-error objective:

$$J = \frac{1}{2}\|r(\mathbf{x}|\mathbf{y}) - \mathbf{x}\|^2. \tag{4}$$

We can also constrain the GAE to be a symmetric model by training it to reconstruct $\mathbf{x}$ given $\mathbf{y}$ and $\mathbf{y}$ given $\mathbf{x}$ at the same time [10]:

$$J = \frac{1}{2}\|r(\mathbf{x}|\mathbf{y}) - \mathbf{x}\|^2 + \frac{1}{2}\|r(\mathbf{y}|\mathbf{x}) - \mathbf{y}\|^2. \tag{5}$$

The symmetric objective can be thought of as the non-probabilistic analogue of modeling a *joint* distribution over $\mathbf{x}$ and $\mathbf{y}$ as opposed to a conditional [10].

[1] Relational features can be mixed with standard features by simply adding connections that are not gated.

3 Gated AutoEncoder Scoring

In [6], the authors showed that data could be scored under an auto-encoder by interpreting the model as a *dynamical system*. In contrast to the probabilistic views based on score matching [14, 17, 4] and regularization, the dynamical systems approach permits scoring under models with either linear (real-valued data) or sigmoid (binary data) outputs, as well as arbitrary hidden unit activation functions. The method is also agnostic to the learning procedure used to train the model, meaning that it is suitable for the various types of regularized auto-encoders which have been proposed recently. In this section, we demonstrate how the dynamical systems view can be extended to the GAE.

3.1 Vector field representation

Similar to [6], we will view the GAE as a dynamical system with the vector field defined by

$$F(\mathbf{y}|\mathbf{x}) = r(\mathbf{y}|\mathbf{x}) - \mathbf{y}.$$

The vector field represents the linear transformation that $\mathbf{y}|\mathbf{x}$ undergoes as a result of applying the reconstruction function $r(\mathbf{y}|\mathbf{x})$. Repeatedly applying the reconstruction function to an input $\mathbf{y}|\mathbf{x} \rightarrow r(\mathbf{y}|\mathbf{x}) \rightarrow r(r(\mathbf{y}|\mathbf{x})|\mathbf{x}) \rightarrow \cdots r(r \cdots r(\mathbf{y}|\mathbf{x})|\mathbf{x})$ yields a trajectory whose dynamics, from a physics perspective, can be viewed as a force field. At any point, the potential force acting on a point is the gradient of some potential energy (negative goodness) at that point. In this light, the GAE reconstruction may be viewed as pushing pairs of inputs $\mathbf{x}, \mathbf{y}$ in the direction of lower energy.

Our goal is to derive the energy function, which we call a scoring function, and which measures how much a GAE "likes" a given pair of inputs $(\mathbf{x}, \mathbf{y})$ up to normalizing constant. In order to find an expression for the potential energy, the vector field must be able to written as the derivative of a scalar field [6]. To check this, we can submit to Poincaré's integrability criterion: For some open, simple connected set $\mathcal{U}$, a continuously differentiable function $F : \mathcal{U} \rightarrow \Re^m$ defines a gradient field if and only if

$$\frac{\partial F_i(\mathbf{y})}{\partial y_j} = \frac{\partial F_j(\mathbf{y})}{\partial y_i}, \ \forall i, j = 1 \cdots n.$$

Considering the GAE, note that i^{th} component of the decoder $r_i(\mathbf{y}|\mathbf{x})$ can be rewritten as

$$r_i(\mathbf{y}|\mathbf{x}) = (W^F_{.i})^T(W^F\mathbf{x} \odot (W^M)^T h(\mathbf{y}, \mathbf{x})) = (W^F_{.i} \odot W^F\mathbf{x})^T(W^M)^T h(\mathbf{y}, \mathbf{x}).$$

The derivatives of $r_i(\mathbf{y}|\mathbf{x}) - y_i$ with respect to y_j are

$$\begin{aligned}
\frac{\partial r_i(\mathbf{y}|\mathbf{x})}{\partial y_j} =& (W^F_{.i} \odot W^F\mathbf{x})^T(W^M)^T \frac{\partial h(\mathbf{x}, \mathbf{y})}{\partial y_j} = \frac{\partial r_j(\mathbf{y}|\mathbf{x})}{\partial y_i} \\
\frac{\partial h(\mathbf{y}, \mathbf{x})}{\partial y_j} =& \frac{\partial h(\mathbf{u})}{\partial \mathbf{u}} W^M(W^F_{.j} \odot W^F\mathbf{x})
\end{aligned} \tag{6}$$

where $\mathbf{u} = W^M((W^F\mathbf{y}) \odot (W^F\mathbf{x}))$. By substituting Equation 6 into $\frac{\partial F_i}{\partial y_j}, \frac{\partial F_j}{\partial y_i}$, we have

$$\frac{\partial F_i}{\partial y_j} = \frac{\partial r_i(\mathbf{y}|\mathbf{x})}{\partial y_j} - \delta_{ij} = \frac{\partial r_j(\mathbf{y}|\mathbf{x})}{\partial y_i} - \delta_{ij} = \frac{\partial F_j}{\partial y_i}$$

where $\delta_{ij} = 1$ for $i = j$ and 0 for $i \neq j$. Similarly, the derivatives of $r_i(\mathbf{y}|\mathbf{x}) - y_i$ with respect to x_j are

$$\begin{aligned}
\frac{\partial r_i(\mathbf{y}|\mathbf{x})}{\partial x_j} =& (W^F_{.i} \odot W^F_{.j})^T(W^M)^T h(\mathbf{x}, \mathbf{y}) + (W^F_{.i} \odot W^F\mathbf{x})(W^M)^T \frac{\partial h}{\partial x_j} = \frac{\partial r_j(\mathbf{y}|\mathbf{x})}{\partial x_i}, \\
\frac{\partial h(\mathbf{y}, \mathbf{x})}{\partial y_j} =& \frac{\partial h(\mathbf{u})}{\partial \mathbf{u}} W^M(W^F_{.j} \odot W^F\mathbf{x}).
\end{aligned} \tag{7}$$

By substituting Equation 7 into $\frac{\partial F_i}{\partial x_j}, \frac{\partial F_j}{\partial x_i}$, this yields

$$\frac{\partial F_i}{\partial x_j} = \frac{\partial r_i(\mathbf{x}|\mathbf{y})}{\partial x_j} = \frac{\partial r_j(\mathbf{x}|\mathbf{y})}{\partial x_i} = \frac{\partial F_j}{\partial x_i}.$$

The GAE satisfies Poincaré's integrability criterion. Furthermore, this also applies to the GAE with a symmetric objective function (Eq. 5) by setting the input as $\boldsymbol{\xi}|\boldsymbol{\gamma}$ such that $\boldsymbol{\xi} = [\mathbf{y}; \mathbf{x}]$ and $\boldsymbol{\gamma} = [\mathbf{x}; \mathbf{y}]$ and following the exact same procedure.

3.2 Scoring the GAE

As mentioned in Section 3.1, our goal is to find an energy surface, so that we can express the energy for a specific pair $(\mathbf{x}, \mathbf{y})$. From the previous section, we showed that Poincaré's criterion is satisfied and this implies that we can write the vector field as the derivative of a scalar field. Moreover, it illustrates that this vector field is a conservative field and this means that the vector field is a gradient of some scalar function, which in this case is the energy function of a GAE:

$$r(\mathbf{y}|\mathbf{x}) - \mathbf{y} = \bigtriangledown E$$

Hence, by integrating out the trajectory of the GAE $(\mathbf{x}, \mathbf{y})$, we can measure the its energy along a path. Moreover, the line integral of a conservative vector field is path independent, which allows us to take the anti-derivative of the scalar function:

$$\begin{aligned} E(\mathbf{y}|\mathbf{x}) =& \int (r(\mathbf{y}|\mathbf{x}) - \mathbf{y}) d\mathbf{y} \\ =& \int W^F \left(\left(W^F \mathbf{x} \right) \odot W^M h(\mathbf{u}) \right) d\mathbf{y} - \int \mathbf{y} d\mathbf{y} \\ =& W^F \left(\left(W^F \mathbf{x} \right) \odot W^M \int h\left(\mathbf{u} \right) d\mathbf{y} \right) - \int \mathbf{y} d\mathbf{y}, \end{aligned} \tag{8}$$

where $\mathbf{u}$ is an auxiliary variable such that $\mathbf{u} = W^M((W^F\mathbf{y}) \odot (W^F\mathbf{x}))$ and $\frac{d\mathbf{u}}{d\mathbf{y}} = W^M(W^F \odot (W^F\mathbf{x} \otimes \mathbf{1}_D))$, and $\otimes$ is the Kronecker product. Moreover, note that decoder can be re-formulated as

$$\begin{aligned} r(\boldsymbol{y}|\boldsymbol{x}) &= (W^F)^T (W^F \boldsymbol{x} \odot (W^M)^T h(\boldsymbol{y}, \boldsymbol{x})) \\ &= \left((W^F)^T \odot (W^F \boldsymbol{x} \otimes \mathbf{1}_D) \right) (W^M)^T h(\boldsymbol{y}, \boldsymbol{x}). \end{aligned}$$

Re-writing Eq. 8 in terms of the auxiliary variable $\mathbf{u}$, we get

$$\begin{aligned} E(\mathbf{y}|\mathbf{x}) &= \left((W^F)^T \odot (W^F\mathbf{x} \otimes \mathbf{1}_D) \right) (W^M)^T \int h(\mathbf{u}) \left(W^M \left(W^F \odot (W^F\mathbf{x} \otimes \mathbf{1}_D) \right) \right)^{-1} d\mathbf{u} - \int \mathbf{y} d\mathbf{y} \\ &= \int h(\mathbf{u}) d\mathbf{u} - \frac{1}{2}\mathbf{y}^2 + \text{const.} \end{aligned} \tag{9}$$

A more detailed derivation from Eq. 8 to Eq. 9 is provided in Appendix A.1. Identical to [6], if $h(\mathbf{u})$ is an element-wise activation function and we know its anti-derivative, then it is very simple to compute $E(\mathbf{x}, \mathbf{y})$.

4 Relationship to Restricted Boltzmann Machines

In this section, we relate GAEs through the scoring function to other types of Restricted Boltzmann Machines, such as the Factored Gated Conditional RBM [16] and the Mean-covariance RBM [12].

4.1 Gated Auto-encoder and Factored Gated Conditional Restricted Boltzmann Machines

Kamyshanska et al. [6] showed that several hidden activation functions defined gradient fields, including sigmoid, softmax, tanh, linear, rectified linear function (ReLU), modulus, and squaring. Hence, these activation functions are applicable to GAEs as well.

In the case of the sigmoid activation function, $\sigma = h(\mathbf{u}) = \frac{1}{1+\exp(-\mathbf{u})}$, our energy function becomes

$$\begin{aligned} E_\sigma =& 2 \int (1 + \exp -(\mathbf{u}))^{-1} d\mathbf{u} - \frac{1}{2}(\mathbf{x}^2 + \mathbf{y}^2) + \text{const}, \\ =& 2 \sum_k \log \left(1 + \exp \left(W^M_{k\cdot} (W^F\mathbf{x} \odot W^F\mathbf{y}) \right) \right) - \frac{1}{2}(\mathbf{x}^2 + \mathbf{y}^2) + \text{const}. \end{aligned}$$

Note that if we consider the conditional GAE we reconstruct $\mathbf{x}$ given $\mathbf{y}$ only, this yields

$$E_\sigma(\mathbf{y}|\mathbf{x}) = \sum_k \log \left(1 + \exp \left(W^M (W^F_{k\cdot}\mathbf{y} \odot W^F_{k\cdot}\mathbf{x}) \right) \right) - \frac{\mathbf{y}^2}{2} + \text{const}. \tag{10}$$

This expression is identical, up to a constant, to the free energy in a Factored Gated Conditional Restricted Boltzmann Machine (FCRBM) with Gaussian visible units and Bernoulli hidden units. We have ignored biases for simplicity. A derivation including biases is shown in Appendix B.1.

4.2 Gated Auto-encoder and Mean-covariance Restricted Boltzmann Machines

The Covariance auto-encoder (CAE) was introduced by [10]. It is a specific form of symmetrically trained auto-encoder whose inputs are equal, $\mathbf{x} = \mathbf{y}$ and input weights are tied. It maintains a set of relational mapping units to model covariance between pixels. One can introduce a separate set of mapping units connected pairwise to only one of the inputs which model the mean intensity. In this case, the model becomes a Mean-covariance auto-encoder (mcAE).

Theorem 1. *Consider a CAE with encoder and decoder:*

$$h(\mathbf{x}) = h(W^M((W^F\mathbf{x})^2) + \mathbf{b})$$
$$r(\mathbf{x}|h) = (W^F)^T(W^F\mathbf{x} \odot (W^M)^T h(\mathbf{x})) + \mathbf{a},$$

where $\theta = \{W^F, W^M, \mathbf{a}, \mathbf{b}\}$ are the parameters of the model, and $h(\cdot) = \frac{1}{1+\exp(-\cdot)}$ is a sigmoid function. Moreover, consider a Covariance Restricted Boltzmann Machine [12] with Gaussian-distributed visibles and Bernoulli-distributed hiddens, such that its energy function is defined by

$$E^c(\mathbf{x}, \mathbf{h}) = \frac{(\mathbf{a} - \mathbf{x})^2}{\sigma^2} - \sum_f P\mathbf{h}(C\mathbf{x})^2 - \mathbf{bh}.$$

Then the energy function of the CAE with dynamics $r(\mathbf{x}|\mathbf{y}) - \mathbf{x}$ is equivalent to the free energy of Covariance RBM up to a constant:

$$E(\mathbf{x}, \mathbf{x}) = \sum_k \log\left(1 + \exp\left(W^M(W^F\mathbf{x})^2 + \mathbf{b}\right)\right) - \frac{\mathbf{x}^2}{2} + \text{const} \tag{11}$$

The proof is given in Appendix B.2. We can extend this analysis to the mcAE by using the above theorem and the results from [6].

Corollary 1.1. *The energy function of a Mean-covariance auto-encoder and the free energy of a Mean-covariance RBM (mcRBM) with Gaussian-distributed visibles and Bernoulli-distributed hiddens are equivalent up to a constant. The energy of the mcAE is:*

$$E = \sum_k \log\left(1 + \exp\left(-W^M(W^F\mathbf{x})^2 - \mathbf{b}\right)\right) + \sum_k \log\left(1 + \exp\left(W\mathbf{x} + \mathbf{c}\right)\right) - \mathbf{x}^2 + \text{const} \tag{12}$$

where $\theta = \{W, \mathbf{c}\}$ parameterize the mean mapping units and $\theta = \{W^F, W^M, \mathbf{a}, \mathbf{b}\}$ parameterize the covariance mapping units.

Proof. The proof is very simple. Let $E_{mc} = E_m + E_c$, where E_m is the energy of the mean auto-encoder, E_c is the energy of the covariance auto-encoder, and E_{mc} is the energy of the mcAE. We know from Theorem 1 that E_c is equivalent to the free energy of a covariance RBM, and the results from [6] show that that E_m is equivalent to the free energy of mean (classical) RBM. As shown in [12], the free energy of a mcRBM is equal to the sum of the free energy of a mean RBM and the free energy of a covariance RBM.

□

5 Regularized Gated Auto-encoder Scoring

In order to learn auto-encoders with *over-complete* representations, that is, more hidden units than input units, it is necessary to regularize during training. A number of different criteria for regularized auto-encoders have been proposed in the literature such as denoising, contraction, and sparsity. Such regularizers make learned representations sensitive to directions in which the density of data is concentrated and less sensitive to lower density regions [4]. Previous work has shown that the denoising auto-encoder and a form of contractive auto-encoder are closely related to each other, and both attempt to capture the underlying data manifold. In this section, we analyze the regularized GAE as a dynamical system[2].

[2]Note that for regularized auto-encoders and gated auto-encoders, the dynamics of training and at test time are different since regularization is not present at test time.

We will consider the same scoring mechanism for Gated auto-encoders, but with a regularizer, such as denoising, contraction, or sparsity. First, we will examine the denoising criterion for the GAE [18]. Additionally, we will analyze from the perspective of GAE training, since we do not add noise to the input at test time. Consider Gaussian corruption noise ϵ on the input $\mathbf{x}$, such that $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Let the vector field be $F = r(\tilde{\mathbf{y}}|\mathbf{y}) - \tilde{\mathbf{y}}$, which satisfies Poincaré's integrability criterion. The proof is shown in Appendix C. We can rewrite the reconstruction function by a Taylor series expansion with respect to $\tilde{\mathbf{y}}$:

$$r(\tilde{\mathbf{y}}|\mathbf{x}) = \tilde{\mathbf{y}} + \sigma^2 \frac{\partial r(\tilde{\mathbf{y}}|\mathbf{x})}{\partial \tilde{\mathbf{y}}} + o(\sigma^2) \simeq \tilde{\mathbf{y}} + \sigma^2 \frac{\partial r(\tilde{\mathbf{y}}|\mathbf{x})}{\partial \tilde{\mathbf{y}}}. \tag{13}$$

where we assume that σ^2 is sufficiently small. Through the use of score matching [5], Swersky et al. [14] have shown that AE models are equivalent to several energy-based models. Applying score matching to the GAE, we can derive another interesting relationship. Substituting Eq. 13 into the score matching objective, we get

$$\begin{aligned}
&\mathbb{E}[\| \frac{\partial r(\tilde{\mathbf{y}}|\mathbf{x})}{\partial \mathbf{x}} - \frac{\partial \log p(\tilde{\mathbf{y}}|\mathbf{y})}{\partial \tilde{\mathbf{y}}} \|^2], \\
&= \frac{1}{\sigma^2} \mathbb{E}[\| r(\tilde{\mathbf{y}}|\mathbf{x}) - \tilde{\mathbf{y}} - \frac{\partial \log \mathcal{N}(\tilde{\mathbf{y}}|\mathbf{y}|\sigma^2)}{\partial \tilde{\mathbf{y}}} \|^2], \\
&= \frac{1}{\sigma^2} \mathbb{E}[\| \left(r(\tilde{\mathbf{y}}|\mathbf{x}) - \tilde{\mathbf{y}} \right) - \left(\mathbf{y} - \tilde{\mathbf{y}} \right) \|^2],
\end{aligned} \tag{14}$$

which is the expected difference of the regularized GAE vector field and the ideal vector field $\mathbf{y} - \tilde{\mathbf{y}}$. Reducing further, we obtain

$$\frac{1}{\sigma^2} \mathbb{E}[\| \left(r(\tilde{\mathbf{y}}|\mathbf{x}) - \tilde{\mathbf{y}} \right) - \left(\mathbf{y} - \tilde{\mathbf{y}} \right) \|^2] = \frac{1}{\sigma^2} \mathbb{E}[\| r(\tilde{\mathbf{y}}|\mathbf{x}) - \mathbf{y} \|^2],$$

which is the objective function for the denoising GAE. Previously, Vincent has shown the connection between score matching and DAEs with a particular form [17]. Additionally, Alain has also shown the relation between score matching and DAEs, but with a more general form [4]. Our derivation is similar to the derivation of [4], but extended to the GAE. However, we have interpreted Equation 14 as another objective function that tries to minimize the difference between two dynamical systems. Hence, one can interpret this as: trying to minimize the reconstruction error under a noised input is equivalent to optimizing the vector field of the GAE towards its ideal form.

6 Experiments

For the experiments, we followed the same experimental setup as [11] where we used a standard set of "Deep Learning Benchmarks" [9]. As well, we extended the auto-encoder classification algorithm of [6] to the gated auto-encoder and mean-covariance auto-encoder. Denoting $E_i^M(\mathbf{x})$ as the energy of the auto-encoder (modelling mean) for class i and $E_i^C(\mathbf{x})$ as the energy of gated-encoder (modelling covariance) for class i. And let B_i^M and B_i^C be the constant term for the mean and covariance autoencoder, where these constant parameters are surrogate normalizing constants which calibrate the per-class auto-encoders. The slight change of the objective function from [6] for the GAE and mcAE are

$$P_{GAE}(y_i|\mathbf{x}) = \frac{\exp(E_i^C(\mathbf{x}) + B_i)}{\sum_j \exp(E_j^C(\mathbf{x}) + B_j)}, P_{mcAE}(y_i|\mathbf{x}) = \frac{\exp(E_i^M(\mathbf{x}) + E_i^C(\mathbf{x}) + B_j)}{\sum_j \exp(E_j^M(\mathbf{x}) + E_j^C(\mathbf{x}) + B_j)} \tag{15}$$

which has a straight forward interpretation. We call these classifiers "Gated Autoencoder Scoring" (GAES) and "Mean Covariance Autoencoder Scoring" (MCAES), respectively. The training procedure is as follows:

1. Train a (denosing/contractive) mean covariance (gated) autoencoder for each class with tied input weights and tied inputs on gated version.
2. Train the mean covariance (gated) autoencoder scoring coefficients based on Equation 15.

Note that while training the GAE, we set $\mathbf{x} = \mathbf{y}$, which means that GAE models pixel covariances. Hence, one may think of GAES (AES) as score based on an AE that models only covariance (mean), and finally mcAES as score based on an AE that models both mean and covariance.

We used mini-batch stochastic gradient descent to optimize parameters during training. The hyper-parameters consisting of: number of hiddens, number of factors, corruption level, learning rate, weight-decay, momentum rate, and batch sizes were chosen based on a held-out validation set. Corruption levels and weight-decay were selected from $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, and number of hidden and factors were selected from 100,300,500. We selected the learning rate and weight-decay from the range $(0.001, 0.0001)$.

Classification error results are shown in Table 1. First, the error rates of auto-encoder scoring variant methods illustrate that across all datasets AES outperforms GAES and mcAES outperforms both AES and GAES. AES models mean and GAE models covariance, while mcAE models both mean and covariance, making it naturally more powerful. Moreover, GAES and mcAES achieve lower error rates by a large margin on rotated MNIST with backgrounds (final row). On the other hand, both GAES and mcAES performed poorly on MNIST with random white noise background (second row from bottom). We believe this phenomenon is due to the inability to model covariance in this dataset. In MNIST with random white noise the pixels are typically uncorrelated, where in rotated MNIST with backgrounds the correlations are present and consistent.

DATA	SVM RBF	RBM	DEEP SAA_3	GSM	AES	GAES	mcAES
RECT	2.15	4.71	2.14	0.56	0.84	0.61	**0.54**
RECTIMG	24.04	23.69	24.05	22.51	21.45	22.85	**21.41**
CONVEX	19.13	19.92	18.41	**17.08**	21.52	21.6	20.63
MNISTROT	11.11	14.69	**10.30**	11.75	11.25	16.5	13.42
MNISTRAND	14.58	9.80	11.28	10.48	**9.70**	18.65	16.73
MNISTROTIM	55.18	52.21	51.93	55.16	47.14	39.98	**35.52**

Table 1: Classification error rates on the Deep Learning Benchmark dataset. SAA_3 stands for three-layer Stacked Auto-encoder. SVM and RBM results are from [17], DEEP and GSM are results from [11], and AES is from [6].

7 Conclusion

There have been many theoretical and empirical studies on auto-encoders [18, 13, 14, 17, 4, 6], however, the theoretical study of gated auto-encoders is limited apart from [10, 3]. The GAE has several intriguing properties that a classical auto-encoder does not, based on its ability to model relations among pixel intensities rather than just the intensities themselves. This opens up a broader set of applications. In this paper, we derive some theoretical results for the GAE that enable us to gain more insight and understanding of its operation.

We cast the GAE as a dynamical system driven by a vector field in order to analyze the model. In the first part of the paper, by following the same procedure as [6], we showed that the GAE could be scored according to an energy function. From this perspective, we demonstrated the equivalency of the GAE energy to the free energy of a FCRBM with Gaussian visible units, Bernoulli hidden units, and sigmoid hidden activations. One interesting observation is that Gaussian-Bernoulli RBMs have been reported to be difficult to train [7, 2], and the success of training RBMs is highly dependent on the training setup [19]. Auto-encoders that can be scored therefore are a more robust alternative. In the same manner, we also showed that the covariance auto-encoder can be formulated in a way such that its energy function is the same as the free energy of a covariance RBM, and this naturally led to a connection between the mean-covariance auto-encoder and mean-covariance RBM.

In Section 5, we explored the relationship between optimizing vector fields with score matching and a L_2 reconstruction objective function, again based on GAE dynamics. Finally, we conducted an empirical investigation showing that on several datasets, Mean-covariance auto-encoders outperformed other classical representation learning methods in the classification setting.

Acknowledgements

The authors acknowledge NSERC for financial support.

References

[1] Yoshua Bengio and Éric Thibodeau-Laufer. Deep generative stochastic networks trainable by backprop. *arXiv preprint arXiv:1306.1091*, 2013.

[2] Kyunghyun Cho, Alexander Ilin, and Tapani Raiko. Improved learning of gaussian-bernoulli restricted boltzmann machines. In *Artificial Neural Networks and Machine Learning (ICANN)*, pages 10–17, 2011.

[3] Alain Droniou and Olivier Sigaud. Gated autoencoders with tied input weights. In *Proceedings of the 28th International Conference of Machine Learning (ICML)*, 2013.

[4] Alain Guillaume and Yoshua Bengio. What regularized auto-encoders learn from the data generating distribution. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

[5] Aaron Hyvrinen. Estimation of non-normalized statistical models using score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

[6] Hanna Kamyshanska. On autoencoder scoring. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 720–728, 2013.

[7] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[9] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007.

[10] Roland Memisevic. Gradient-based learning of higher-order image features. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2011.

[11] Roland Memisevic, Christopher Zach, Geoffrey Hinton, and Marc Pollefeys. Gated softmax classification. In *Neural Information Processing Systems (NIPS).*, 2010.

[12] MarcAurelio Ranzato and Geoffery E. Hinton. Modeling pixel means and covariances using factorized third-order boltzmann machines. In *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR)*, 2010.

[13] Salah Rifai. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011.

[14] Kevin Swersky, MarcAurelio Ranzato, David Buchman, Nando D. Freitas, and Benjamin M. Marlin. On autoencoders and score matching for energy based models. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 1201–1208, 2011.

[15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[16] Graham W. Taylor and Geoffrey E. Hinton. Factored conditional restricted boltzmann machines for modeling motion style. In *Proceedings of the 26th annual international conference on machine learning*, pages 1025–1032, 2009.

[17] Pascal Vincent. A connection between score matching and denoising auto-encoders. *Neural Computation*, 23(7):1661–1674, 2010.

[18] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and P.A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 2008.

[19] Nan Wang, Jan Melchior, and Laurenz Wiskott. Gaussian-binary restricted boltzmann machines on modeling natural image statistics. Technical report, Institut fur Neuroinformatik Ruhr-Universitat Bochum, Bochum, 44780, Germany, 2014.

A Gated Auto-encoder Scoring

A.1 Measuring the score in terms of energy

Integrating out the GAE's trajectory, we have

$$\begin{aligned} E(\mathbf{y}|\mathbf{x}) &= \int_{\mathcal{C}} (r(\mathbf{y}|\mathbf{x}) - \mathbf{y}) d\mathbf{y} \\ &= \int W^Y \left((W^X \mathbf{x}) \odot W^M h(\mathbf{u}) \right) d\mathbf{y} - \int \mathbf{y} d\mathbf{y} \\ &= W^Y \left((W^X \mathbf{x}) \odot W^M \int h(\mathbf{u}) \, d\mathbf{u} \right) - \int \mathbf{y} d\mathbf{y}, \end{aligned} \tag{16}$$

where $\mathbf{u}$ is an auxiliary variable such that $\mathbf{u} = W^M((W^Y\mathbf{y}) \odot (W^X\mathbf{x}))$ and $\frac{d\mathbf{u}}{d\mathbf{y}} = W^M(W^Y \odot (W^X\mathbf{x} \otimes \mathbf{1}_D))$, where $\otimes$ is the kronecker product. Consider symmetric objective function, which is defined in Equation 5. Then we have to also consider vector field system where the both symmetric cases $\mathbf{x}|\mathbf{y}$ and $\mathbf{y}|\mathbf{x}$ is valid. As mentioned in Section 3.1, let $\xi = [\mathbf{x}; \mathbf{y}]$ and $\gamma = [\mathbf{y}; \mathbf{x}]$. As well, let $W^\xi = \text{diag}(W^X, W^Y)$ and $W^\gamma = \text{diag}(W^Y, W^X)$. Consequently, the vector field becomes

$$F(\boldsymbol{\xi}|\boldsymbol{\gamma}) = r(\boldsymbol{\xi}|\boldsymbol{\gamma}) - \boldsymbol{\xi} \tag{17}$$

and the energy function becomes

$$\begin{aligned} E(\boldsymbol{\xi}|\boldsymbol{\gamma}) &= \int (r(\boldsymbol{\xi}|\boldsymbol{\gamma}) - \boldsymbol{\xi}) d\boldsymbol{\xi} \\ &= \int (W^\xi)^T ((W^\gamma \boldsymbol{\gamma}) \odot (W^M)^T h(\boldsymbol{u})) d\boldsymbol{\xi} - \int \boldsymbol{\xi} d\boldsymbol{\xi} \\ &= (W^\xi)^T ((W^\gamma \boldsymbol{\gamma}) \odot (W^M)^T \int h(\boldsymbol{u}) d\boldsymbol{u}) - \int \boldsymbol{\xi} d\boldsymbol{\xi} \end{aligned}$$

where $\boldsymbol{u}$ is an auxiliary variable such that $\boldsymbol{u} = W^M \left((W^\xi \boldsymbol{\xi}) \odot (W^\gamma \boldsymbol{\gamma}) \right)$. Then

$$\frac{d\boldsymbol{u}}{d\boldsymbol{\xi}} = W^M \left(W^\xi \odot (W^\gamma \boldsymbol{\gamma} \otimes \mathbf{1}_D) \right)$$

Moreover, note that decoder can be re-formulated as

$$\begin{aligned} r(\boldsymbol{\xi}|\boldsymbol{\gamma}) &= (W^\xi)^T (W^\gamma \boldsymbol{\gamma} \odot (W^M)^T h(\boldsymbol{\xi}, \boldsymbol{\gamma})) \\ &= \left((W^\xi)^T \odot (W^\gamma \boldsymbol{\gamma} \otimes \mathbf{1}_D) \right) (W^M)^T h(\boldsymbol{\xi}, \boldsymbol{\gamma}) \end{aligned}$$

Re-writing the first term of Equation 16 in terms of the auxiliary variable $\boldsymbol{u}$, we get

$$\begin{aligned} E(\boldsymbol{\xi}|\boldsymbol{\gamma}) &= \left((W^\xi)^T \odot (W^\gamma \boldsymbol{\gamma} \otimes \mathbf{1}_D) \right) (W^M)^T \int h(\boldsymbol{u}) \left(W^M (W^\xi \odot (W^\gamma \boldsymbol{\gamma} \otimes \mathbf{1}_D)) \right)^{-1} d\boldsymbol{u} - \int \boldsymbol{\xi} d\boldsymbol{\xi} \\ &= \left((W^\xi)^T \odot (W^\gamma \boldsymbol{\gamma} \otimes \mathbf{1}_D) \right) (W^M)^T \left(W^M (W^\xi \odot (W^\gamma \boldsymbol{\gamma} \otimes \mathbf{1}_D)) \right)^{-1} \int h(\boldsymbol{u}) d\boldsymbol{u} - \int \boldsymbol{\xi} d\boldsymbol{\xi} \\ &= \int h(\boldsymbol{u}) d\boldsymbol{u} - \int \boldsymbol{\xi} d\boldsymbol{\xi} \\ &= \int h(\boldsymbol{u}) d\boldsymbol{u} - \frac{1}{2} \boldsymbol{\xi}^2 + \text{const} \end{aligned}$$

B Relation to other types of Restricted Boltzmann Machines

B.1 Gated Auto-encoder and Factored Gated Conditional Restricted Boltzmann Machines

Suppose that hidden activation functon is sigmoid function. Moreover, we define our Gated Auto-encoder to be consists of encoder $h(\cdot)$ and decoder $r(\cdot)$ such that

$$\begin{aligned} h(\mathbf{x}, \mathbf{y}) &= h(W^M((W^X\mathbf{x}) \odot (W^Y\mathbf{y})) + \mathbf{b}) \\ r(\mathbf{x}|\mathbf{y}, h) &= (W^X)^T ((W^Y\mathbf{y}) \odot (W^M)^T h(\mathbf{x}, \mathbf{y})) + \mathbf{a} \end{aligned}$$

where $\theta = \{W^M, W^X, W^Y, \mathbf{b}\}$ is the parameters of the model. Note that the weights are not tied in this case. Then we have the energy function for Gated Auto-encoder as follows:

$$\begin{aligned} E_\sigma(\mathbf{x}|\mathbf{y}) &= \int (1 + \exp\left(-W^M(W^X\mathbf{x}) \odot (W^Y\mathbf{y}) - \mathbf{b}\right))^{-1} d\mathbf{u} - \frac{\mathbf{x}^2}{2} + \mathbf{a}\mathbf{x} + \text{const} \\ &= \sum_k \log\left(1 + \exp\left(-W^M_{k\cdot}(W^X\mathbf{x}) \odot (W^Y\mathbf{y}) - b_k\right)\right) - \frac{\mathbf{x}^2}{2} + \mathbf{a}\mathbf{x} + \text{const} \end{aligned}$$

Now we consider the free energy function for Factored Gated Conditional Restricted Boltzmann Machines (FCRBM). We will assume that the reader is familiar with Restricted Boltzmann Machines and we will be dealing with the RBM that is consists of Gaussian distribution over the visible units and Bernoulli distribution over hidden units.

The energy function of FCRBM is defined by

$$E(\mathbf{x}, \mathbf{h}|\mathbf{y}) = \frac{(\mathbf{a} - \mathbf{x})^2}{2\sigma^2} - \mathbf{b}\mathbf{h} - \sum_f W^X_{f\cdot}\mathbf{x} \odot W^Y_{f\cdot}\mathbf{y} \odot W^H_{f\cdot}\mathbf{h} \tag{18}$$

The probability distribution over data $\mathbf{x}$ given $\mathbf{y}$ of FCRBM is

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &= \frac{\sum_{\mathbf{h}} \exp -(E(\mathbf{x}, \mathbf{h}|\mathbf{y}))}{Z(\mathbf{y})} = \frac{\exp\left(-F(\mathbf{x}|\mathbf{y})\right)}{Z(\mathbf{y})} \\ -F(\mathbf{x}|\mathbf{y}) &= \log\left(\sum_{\mathbf{h}} \exp\left(-E(\mathbf{x}, \mathbf{h}|\mathbf{y})\right)\right) \end{aligned}$$

where $Z(\mathbf{y}) = \sum_{\mathbf{x},\mathbf{h}} \exp\left(E(\mathbf{x}, \mathbf{h}|\mathbf{y})\right)$ is the partition function and $F(\mathbf{x}|\mathbf{y})$ is the free energy function. Expending free energy function, we get

$$\begin{aligned} -F(\mathbf{x}|\mathbf{y}) &= \log \sum_{\mathbf{h}} \exp\left(-E(\mathbf{x}, \mathbf{h}|\mathbf{y})\right) \\ &= \log \sum_{\mathbf{h}} \exp\left(\frac{-(\mathbf{a} - \mathbf{x})^2}{2\sigma^2} + \mathbf{b}\mathbf{h} + \sum_f W^X_{f\cdot}\mathbf{x} \odot W^Y_{f\cdot}\mathbf{y} \odot W^H_{f\cdot}\mathbf{h}\right) \\ &= -\frac{(\mathbf{a} - \mathbf{x})^2}{2\sigma^2} + \log\left(\sum_{\mathbf{h}} \exp\left(\mathbf{b}\mathbf{h} + \sum_f W^X_{f\cdot}\mathbf{x} \odot W^Y_{f\cdot}\mathbf{y} \odot W^H_{f\cdot}\mathbf{h}\right)\right) \\ &= -\frac{(\mathbf{a} - \mathbf{x})^2}{2\sigma^2} + \log\left(\sum_{\mathbf{h}} \prod_k \exp\left(b_k h_k + \sum_f (W^X_{f\cdot}\mathbf{x} \odot W^Y_{f\cdot}\mathbf{y}) \odot W^H_{fk} h_k\right)\right) \\ &= -\frac{(\mathbf{a} - \mathbf{x})^2}{2\sigma^2} + \sum_k \log\left(1 + \exp\left(b_k + \sum_f \left((W^H_{fk})^T(W^X\mathbf{x} \odot W^Y\mathbf{y})\right)\right)\right) \end{aligned}$$

Note that we can center the data by subtracting mean of $\mathbf{x}$ and dividing by its variance in order to make $\sigma^2 = 1$.

$$\begin{aligned} -F(\mathbf{x}|\mathbf{y}) &= -\frac{(\mathbf{a} - \mathbf{x})^2}{2} + \sum_k \log\left(1 + \exp\left(-b_k - \sum_f (W^H_{fk})^T(W^X\mathbf{x} \odot W^Y\mathbf{y})\right)\right) \\ &= \sum_k \log\left(1 + \exp\left(b_k + \sum_f (W^H_{fk})^T(W^X\mathbf{x} \odot W^Y\mathbf{y})\right)\right) - \mathbf{a}^2 + \mathbf{a}\mathbf{x} - \frac{\mathbf{x}^2}{2} \\ &= \sum_k \log\left(1 + \exp\left(b_k + \sum_f (W^H_{fk})^T(W^X\mathbf{x} \odot W^Y\mathbf{y})\right)\right) + \mathbf{a}\mathbf{x} - \frac{\mathbf{x}^2}{2} + \text{const} \end{aligned}$$

Letting $W^M = (W^H)^T$, we get

$$= \sum_k \log\left(1 + \exp\left(b_k + \sum_f W^M_{kf}(W^X\mathbf{x} \odot W^Y\mathbf{y})\right)\right) + \mathbf{ax} - \frac{\mathbf{x}^2}{2} + \text{const}$$

Hence, the Conditional Gated Auto-encoder and FCRBM are equal up to a constant.

B.2 Gated Auto-encoder and mean covariance Restricted Boltzmann Machines

Theorem 2. *Consider a covariance auto-encoder with an encoder and decoder,*

$$h(\mathbf{x}, \mathbf{x}) = h(W^M((W^F\mathbf{x})^2) + \mathbf{b})$$
$$r(\mathbf{x}|\mathbf{y} = \mathbf{x}, h) = (W^F)^T(W^F\mathbf{y} \odot (W^M)^T h(\mathbf{x}, \mathbf{y})) + \mathbf{a},$$

where $\theta = \{W^F, W^M, \mathbf{a}, \mathbf{b}\}$ are the parameters of the model. Moreover, consider a covariance Restricted Boltzmann Machine with Gaussian distribution over the visibles and Bernoulli distribution over the hiddens, such that its energy function is defined by

$$E^c(\mathbf{x}, \mathbf{h}) = \frac{(\mathbf{a} - \mathbf{x})^2}{\sigma^2} - \sum_f P\mathbf{h}(C\mathbf{x})^2 - \mathbf{bh},$$

where $\theta = \{P, C, \mathbf{a}, \mathbf{b}\}$ are its parameters. Then the energy function for covariance Auto-encoder with the dynamics of $r(\mathbf{x}|\mathbf{y}) - \mathbf{x}$ is equivalent to the free energy of a covariance Restricted Boltzmann Machines. And the energy function of the covariance Auto-encoder is

$$E(\mathbf{x}, \mathbf{x}) = \sum_k \log(1 + \exp(W^M(W^F\mathbf{x})^2 + \mathbf{b})) - \mathbf{x}^2 + \textit{const} \tag{19}$$

Proof. Note that the covariance auto-encoder is the same as a regular Gated Auto-encoder, but setting $\mathbf{y} = \mathbf{x}$ and making the weights on the factor layer the same. Then applying the general energy equation for GAE, Equation 9, to the covariance auto-encoder, we get

$$\begin{aligned} E(\mathbf{x}, \mathbf{x}) &= \int h(\mathbf{u})d\mathbf{u} - \frac{1}{2}\mathbf{x}^2 + \text{const} \\ &= \sum_k \log(1 + \exp(W^M(W^F\mathbf{x})^2 + \mathbf{b})) - \mathbf{x}^2 + \mathbf{ax} + \text{const}, \end{aligned} \tag{20}$$

where $\mathbf{u} = W^M(W^F\mathbf{x})^2 + \mathbf{b}$.

Now we consider the free energy function for mean covariance Restricted Boltzmann Machines (mcRBM). We will assume that the reader is familiar with Restricted Boltzmann Machines and we will be dealing with the RBM that is consists of Gaussian distribution over the visible units and Bernoulli distribution over hidden units.

$$\begin{aligned} -F(\mathbf{x}|\mathbf{y}) &= \log \sum_{\mathbf{h}} \exp\left(-E(\mathbf{x}, \mathbf{h}|\mathbf{y})\right) \\ &= \log \sum_h \exp\left(-\frac{(\mathbf{a} - \mathbf{x})^2}{\sigma^2} + (P\mathbf{h})(C\mathbf{x})^2 + \mathbf{bh}\right) \\ &= \log \sum_h \prod_k \exp\left(-\frac{(\mathbf{a} - \mathbf{x})^2}{\sigma^2} + \sum_f (P_{fk}h_k)(C\mathbf{x})^2 + b_k h_k\right) \\ &= \sum_k \log\left(1 + \exp\left(\sum_f (P_{fk}h_k)(C\mathbf{x})^2\right)\right) - \frac{(\mathbf{a} - \mathbf{x})^2}{\sigma^2} \end{aligned}$$

Note that we can center the data by subtracting mean of $\mathbf{x}$ and dividing by its variance in order to make $\sigma^2 = 1$.

$$= \sum_k \log\left(1 + \exp\left(\sum_f (P_{fk}h_k)(C\mathbf{x})^2\right)\right) - (\mathbf{a} - \mathbf{x})^2 \tag{21}$$

Letting $W^M = P^T$ and $W^F = C$, we get

$$= \sum_k \log\left(1 + \exp\left(\sum_f (P_{fk}h_k)(C\mathbf{x})^2\right)\right) - \mathbf{x}^2 + \mathbf{a}\mathbf{x} + \text{const} \tag{22}$$

Therefore, the two equations are equivalent. □

C Regularized Gated AutoEncoder Scoring

Using the same dynamic system as GAE, define the dynamics of regularized GAE to be

$$F(\tilde{\mathbf{y}}|\mathbf{x}) = r(\tilde{\mathbf{y}}|\mathbf{x}) - \tilde{\mathbf{y}}.$$

We show that $F(\tilde{\mathbf{y}}|\mathbf{x})$ satisfies the Poincaré criterion, Equation 6.

$$\begin{aligned}
\frac{F_i(\tilde{\mathbf{y}}|\mathbf{x})}{y_j} =& \frac{\partial(r_i(\tilde{\mathbf{y}}|\mathbf{x}) - \tilde{y}_i)}{\partial y_j} \\
=& \frac{\partial(r_i(\mathbf{y}+\epsilon|\mathbf{x}) - \tilde{y}_i)}{\partial y_j} \\
=& \frac{\partial(r_i(\mathbf{y}|\mathbf{x}) + \epsilon\frac{\partial r_i(\mathbf{y}|\mathbf{x})}{\partial \mathbf{y}} + o(\epsilon^2) - y_i - \epsilon_i)}{\partial y_j} \\
=& \frac{\partial(r_i(\mathbf{y}|\mathbf{x}) - y_i)}{\partial y_j} + \epsilon\frac{\partial^2 r_i(\mathbf{y}|\mathbf{x})}{\partial y_j \partial \mathbf{y}}
\end{aligned} \tag{23}$$

Notice that first component of the Equation 23 is symmetric in partial derivatives from Section 3.1. In fact, we can observe that second part of the equation is zero.

$$\begin{aligned}
\frac{\partial}{\partial y_j}\frac{\partial h(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}} =& W^M(W^F\mathbf{1} \odot W^F\mathbf{x})\frac{\partial}{\partial y_j}\frac{\partial h(\mathbf{u})}{\partial \mathbf{u}} = 0 \\
\epsilon\frac{\partial^2 r_i(\mathbf{y}|\mathbf{x})}{\partial y_j \partial \mathbf{y}} =& (W^F_{.i} \odot W^F\mathbf{x})(W^M)^T\frac{\partial}{\partial y_j}\frac{\partial h(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}} = 0
\end{aligned}$$

where $\mathbf{u} = W^M\left((W^F\mathbf{x}) \odot (W^F\mathbf{y})\right)$. Consequently, substituting Equation 23 to $\frac{F_i(\tilde{\mathbf{y}}|\mathbf{x})}{y_j}$ and $\frac{F_i(\tilde{\mathbf{y}}|\mathbf{x})}{y_i}$, we get

$$\frac{F_i}{\partial y_j} = \frac{\partial r_i(\mathbf{y}|\mathbf{x})}{\partial y_j} - \delta_{ij} = \frac{\partial r_j(\mathbf{y}|\mathbf{x})}{\partial y_i} - \delta_{ji} = \frac{F_j}{\partial y_i}$$

where $\delta_{ij} = 1$ for $i = j$ and 0 for $i \neq j$. Thus, the regularized GAE can be written as in terms of a scalar field, and the vector field can be integrated.