Towards universal neural nets: Gibbs machines and ACE.

Galin Georgiev GALIN.GEORGIEV@GAMMADYNAMICS.COM
GammaDynamics, LLC

Abstract

We study a class of neural nets - *Gibbs machines* - which are a type of variational auto-encoders, designed for gradual learning. They offer an universal platform for incrementally adding newly learned features, including physical symmetries, and are directly connected to information geometry and thermodynamics. Combining them with classifiers, gives rise to a brand of universal generative neural nets - stochastic auto-classifier-encoders (ACE). ACE have state-of-the-art performance in their class, both for classification and density estimation for the MNIST data set.

1. Introduction.

1.1. Universality.

We buck the recent trend of building highly specialized neural nets by exploring nets which accomplish multiple tasks without compromising performance. An *universal* net can be tentatively described as one which, among other things: i) works for a variety of applications, i.e. visual recognition/reconstruction, speech recognition/reconstruction, natural language processing, etc; ii) performs various tasks: classification, generation, probability density estimation, etc; iii) is self-contained, i.e., does not use specialized external machine learning methods; iv) is biologically plausible.

1.2. Probabilistic and quantum viewpoint.

The input of a neural net is typically a set of P observations $\{\mathbf{x}_\mu\}_{\mu=1}^P$, which can be represented mathematically as row-vectors in the space spanned by *observables* $\{\mathbf{x}_i\}_{i=1}^N$, e.g., the N pixels on a screen. The net is then asked to perform classification, estimation, generation, etc, tasks on it. In *generative* nets, this is accomplished by randomly generating L latent observations $\{\mathbf{z}_\mu^{(\kappa)}\}_{\kappa=1}^L$ for every observation $\mathbf{x}_\mu$. This induced "uncertainty" of the μ-th state is modeled by a *model* conditional density $p(\mathbf{z}|\mathbf{x}_\mu)$. It is the copy-cat, in imaginary time/space, of the (squared) wave function from quantum mechanics[1], and fully describes the μ-th conditional state $\left(\mathbf{x}_\mu, \{\mathbf{z}_\mu^{(\kappa)}\}_{\kappa=1}^L\right)$. In statistical mechanics parlance, the *latents* are fluctuating microscopic variables, while the macroscopic observables are obtained from them via some aggregation. In the absence of physical time, observations are thus interpreted as *partial equilibria* of independent small parts of the expanded (by a factor of L) original data set.

The quality of the model conditional density, or more generally - the model joint density $p(\mathbf{z}, \mathbf{x}_\mu)$ - is judged by the "distance" from the implied marginal density $q(\mathbf{x}) := \int p(\mathbf{x}, \mathbf{z})d\mathbf{z}$ to the *empirical* marginal density $r(\mathbf{x})$. This distance is called *cross-entropy* or *negative log-likelihood* and denoted by $-\log \mathcal{L}(r||q) := \mathbf{E}(-\log q(\mathbf{x}))_{r(\mathbf{x})}$, where $\mathbf{E}()_{r()}$ is an expectation with respect to $r()$. Its minimization is the ultimate goal.

1.3. Equilibrium setting. Gibbs machines.

Einstein originated the theory of equilibrium i.e. small fluctuations in early 20-th century (Einstein, 2006). He used an exponential model density $p^{Exp}()$ in space and derived the Brownian diffusion i.e., Gaussian model density $p^G()$ in space/time. They are special cases of a broad class of densities - *Gibbs* or *exponential* densities - which form the foundation of classic statistical mechanics. Gibbs densities are variational *maximum-entropy* densities and hence optimal for modeling equilibria.

We argue in sub-sections 2.3, 3.1 that Gibbs densities are also optimal for modeling *fully-generative* equilibrium nets, and call the nets using them *Gibbs machines*. They were inspired by the first fully-generative nets - the *variational* auto-encoders (VAE) (Kingma & Welling, 2014), (Rezende et al., 2014) - and employ the same upper bound (3.4) for the negative log-likelihood (because of its univer-

[1] Strictly speaking, we will employ unbounded densities and hence stochastic analysis formalism and its centerpiece - the diffusion equation. But they are formally equivalent to the quantum-mechanical formalism and its centerpiece - the Schrodinger equation - in imaginary time/space coordinates.

sality for fully-generative equilibrium nets, section 3.1). Like their physics counterparts, Gibbs machines offer a platform for mimicking the gradual nature of learning: already learned *symmetry statistics* like space/time symmetries, can be added incrementally and accelerate learning, sub-sections 1.5, 2.3, 3.2.

1.4. The curse of Gaussianization.

Unlike physics and Brownian particles, human data is decidedly non-equilibrium in nature and exhibits large fluctuations and non-Gaussian behavior. Unfortunately, some of the key features of modern neural nets, like non-linear activation functions and dropout (Srivastava et al., 2014), come at the high price of *Gaussianizing* the data set.

Quantifying non-Gaussianity and "distance" from equilibrium is not easy when dealing with large number of observables N and observations P. Fortunately, there is a one-dimensional proxy for non-Gaussianity of a multidimensional data set: the non-Gaussianity of the negative Gaussian log-likelihoods $\{-\log p^G(\mathbf{z}_\mu)\}_\mu$. Here, $-\log p^G(\mathbf{z}_\mu) = \frac{1}{2}\mathbf{z}_\mu \mathbf{C}(\mathbf{z})^{-1}\mathbf{z}_\mu + const$, with a multivariate Gaussian $\mathcal{N}_{N_{lat}}(0, \mathbf{C}(\mathbf{z}))$ as model density and $\mathbf{C}(\mathbf{z})$ the empirical covariance[2]. These negative *observation entropies*, as interpreted by Einstein, are related to singular value decomposition and are central in theory of fluctuations (Landau & Lifshitz, 1980), chapter 12. Their second moment $\mathbf{E}\left((\log p^G(\mathbf{z}_\mu))^2\right)_r$ is proportional to the multivariate *kurtosis* measuring the "fatness" of the density.

The right *quantile-quantile* (Q-Q) plot in Figure 1 shows the Gaussianization effect of non-linearities and dropout for the MNIST data set (LeCun et al., 1998). Non-linearities Gaussianize because they are compressive in nature and "rectify" the unlikely (with large negative log-likelihoods) observations, which we refer to as *intricates*, see Figure 3. Dropout Gaussianizes because it drops latent variables and hence decreases the kurtosis.

1.4.1. Non-generative ACE.

It is precisely the intricates, which - because of their extreme non-Gaussianity, see Figure 2 - are ideal candidates for "feature vectors" in classification tasks (Hyvarinen et al., 2009), section 7.9, 7.10. Their conjugates are then the "receptive fields" or "feature detectors"[3] - see open problem 1 in section 5. We show on the top (respectively, bottom) plot in Figure 3 the most (respectively, least) 30 intricate images in MNIST, in descending order of Gaussian negative log-likelihood $-\log p^G(\mathbf{x}_\mu) \geq 0$, from the class corresponding to the digit 8.

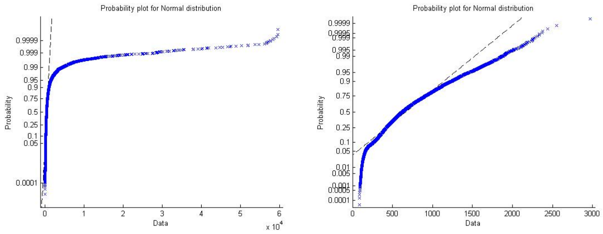

Figure 1. Q-Q plots against a Gaussian of the density of the negative log-likelihoods $\{-\log p^G(\mathbf{z}_\mu)\}_\mu$ of the 10000 MNIST test observations in layer 2 of a 5-layer standard feed-forward classifier net, see right branch of Figure 4 and Appendix A for implementation details. Layer sizes are 784-700-700-700-10. Learning rate = 0.0015, decay = 500 epochs, batch size = 10000. For the right plot, dropout is 0.2 in input layer and 0.5 in hidden layers. As an exception from the rules in Appendix A, a tanh() activation function is used in the first hidden layer. **Left.** No dropout and no non-linearity: highly non-Gaussian. **Right.** With dropout and non-linearity: severely Gaussianized, especially for the intricates towards the right (see text).

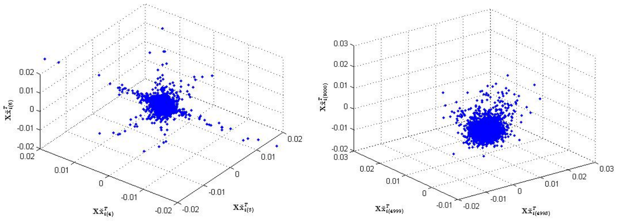

Figure 2. **Left.** The first 5000 MNIST training images, projected on three of least likely, i.e. most intricate conjugate images, ranked #3, #4, #6 in descending order of Gaussian negative log-likelihood $-\log p^G(\mathbf{x}_\mu)$. This is a highly non-Gaussian 3-dimensional distribution. **Right.** The same MNIST images, projected on the three most likely conjugate images, ranked # 4998, #4999, #5000 in Gaussian negative log likelihood. Much more Gaussian-looking.

In order to preserve the non-Gaussianity of the data, and improve performance significantly along the way, we will combine classifiers with *auto-encoders* - hence the name *auto-classifier-encoder* (ACE). Auto-encoders have a *reconstruction* error in their negative log-likelihood and thus force the net to be more faithful to the raw data. ACE si-

[2] If the latent observations $\{\mathbf{z}_\mu\}_{\mu=1}^P$ come from an N_{lat}-dimensional Gaussian distribution, the density of these negative Gaussian log-likelihoods is proportional to the familiar $F(N_{lat}, P - N_{lat})$ density (Mardia et al., 1979), sections 1.8, 3.5. For the typical case $P - N_{lat} \to \infty$, it is proportional to the chi-squared density $\chi^2_{N_{lat}}()$, which in turn converges to a rescaled Gaussian $\mathcal{N}(0, 1)$, as $N_{lat} \to \infty$.

[3] Recall that, for a given row-vector observation $\mathbf{x}_\mu$, its *conjugate* is $\check{\mathbf{x}}_\mu = \mathbf{x}_\mu \mathbf{C}^{-1}$, where $\mathbf{C}$ is the covariance matrix. Up to a constant, the Gaussian negative log-likelihood is thus the inner product of an observation and its conjugate, in the standard Euclidean metric: $-\log p^G(\mathbf{x}_\mu) = \frac{1}{2} < \check{\mathbf{x}}_\mu, \mathbf{x}_\mu >$.

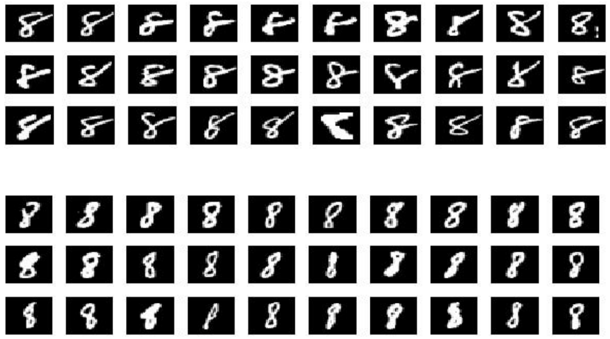

Figure 3. **Top.** The top 30 MNIST training images, ranked in descending order of Gaussian negative log-likelihood $-\log p^G(\mathbf{x}_\mu)$, from the class corresponding to the digit 8. They are quite intricate indeed. **Bottom.** The bottom 30 MNIST training images, in the same order, from the same class. Much more vanilla-looking.

multaneously classifies and reconstructs, assuming an independence between the two, and hence additivity of the respective negative log-likelihoods:

$$-log\mathcal{L}_{ACE} = -log\mathcal{L}_{AE} - log\mathcal{L}_C. \qquad (1.1)$$

In its first - non-generative - installment, ACE can do with a standard classifier and a shallow auto-encoder in the dual space of observations. It still beats handily the peers in its class, Figure 8, right.

1.4.2. Non-Gaussian densities.

In real-life data sets, the number of observations $P \to \infty$, while the dimension of observables N is fixed. An universal net will hence tend to work better when the dimension of latent layers $N_{lat} \geq N$, i.e. have the so-called *overcomplete* representation (Coates et al., 2011). When $N_{lat} >> N$, for any given N-dimensional observation $\mathbf{x}_\mu$, only a small number of latents $\{z_{\mu j}\}_{j=1}^{N_{lat}}$ deviate significantly from zero, as on the left plot of Figure 2. For these *sparse representations*, sampling from high-entropy Gaussian-like densities, as on the right plot of Figure 2, is flawed. Sampling instead from "fat-tail" densities offers a significant performance improvement for MNIST, Figure 9, right. As in mathematical finance, stochastic volatility and jumps are arguably the first natural source of non-Gaussianity, and are almost fully-tractable. The *q-Gibbs machines* offer another venue, sub-section 2.3.

1.4.3. Generative ACE.

An even greater issue for current nets is the spontaneous "clumping" or clusterization which is prevalent in real-life data sets. Statistical mechanics deals with it by introducing *higher-hierarchy* densities which are conditional on low-hierarchy densities. Clusterization aggregates the low-hierarchy partial equilibria - the observations - into higher-hierarchy partial equilibria - clusters, sub-section 1.2.

To mimic this universal phenomenon, in its second, generative, installment, ACE combines a classifier and a generative auto-encoder in the same space of observables, in a brand of an auto-encoder *supervision*, Figure 4. ACE generalizes the classic idea of using separate decoders for separate classes, (Hinton et al., 1995). In training, the conditional latent density $p(\mathbf{z}|\mathbf{x}_\mu)$ from sub-section 1.2 is generalized to $p(\mathbf{z}|\mathbf{x}_\mu, c_\mu)$, where c_μ is the class or label of the μ-th observation. Since, of course, classification labels can not be used on the testing set, the sampling during testing is from a mixture of densities, with class probabilities supplied by the classifier (hence the dashed line in Figure 4). Mixture densities in the posterior were also used in (Kingma et al., 2014), albeit in a different architecture. The ACE is universal in the sense of subsection 1.1 and achieves state-of-the-art performance, both as a classifier and a density estimator, Figure 9. For its relation with information geometry, see open problem 6 in section 5.

1.5. Symmetries in the latent manifold.

When the dimensionality N_{lat} of the ACE latent layer is low, traversing the latent dimension in some uniform fashion describes the latent manifold for a given class. Figure 5 shows the dominant dimension for each of the 10 classes in MNIST. This so-called *manifold learning* by modern feed-forward nets was pioneered by the *contractive* auto-encoders (CAE) (Rifai et al., 2012). A symmetry in our context is, loosely speaking, a one-dimensional parametric transformation, which leaves the log-likelihood unchanged. In probabilistic terms, this is equivalent to the existence of a one-parametric density, from which "symmetric" observations are sampled, see (1.2) below. Nets currently learn symmetries from the training data, after it is artificially *augmented*, e.g. by adding rotated, translated, rescaled, etc, images, in the case of visual recognition. But once a symmetry is learned, it does not make sense to re-learn it for every new data set.

We hence propose the *reverse* approach: add the symmetry explicitly to the latent layer, alongside its Noether invariant, (Gelfand & Fomin, 1963). As an example, consider translational symmetries in a two-dimensional system with coordinates $(z^{(h)}, z^{(v)}) \in \mathbb{R}^2$. They imply the conservation of the horizontal and vertical *momenta* $(-i\hbar\partial/\partial z^{(h)}, -i\hbar\partial/\partial z^{(v)}) = (p^{(h)}, p^{(v)}) \in \mathbb{R}^2$ and a quantum mechanical wave function $\sim e^{\frac{i}{\hbar}p^{(h)}(z^{(h)}-h)+\frac{i}{\hbar}p^{(v)}(z^{(v)}-v)}$, where (h, v) are offsets, (Landau & Lifshitz, 1977), section 15. After switching to imaginary time/space as in sub-section 1.2, and setting $\hbar = 1$, the corresponding conditional model density for a

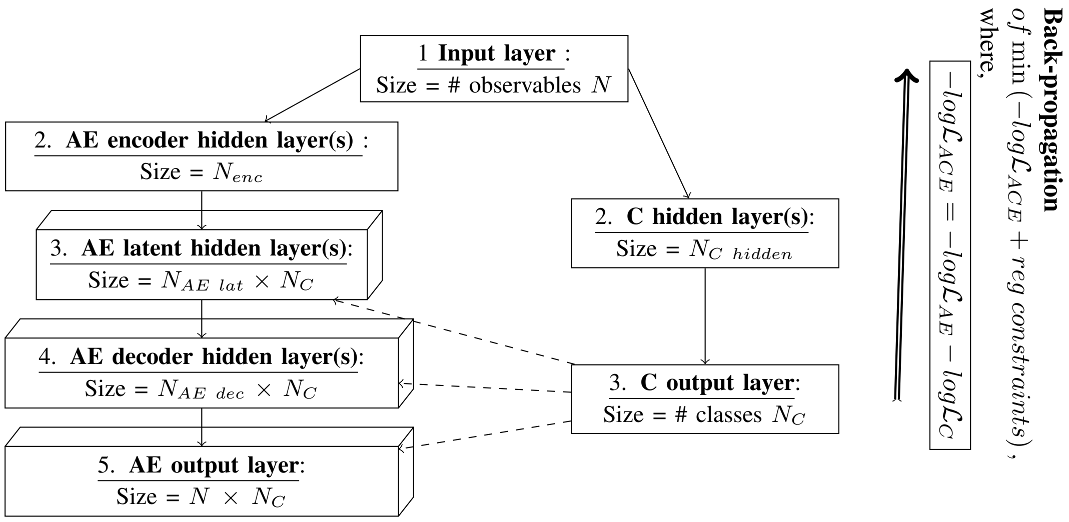
1 Input layer :
Size = # observables N
2. AE encoder hidden layer(s) :
Size = N_{enc}
3. AE latent hidden layer(s):
Size = $N_{AE\ lat} \times N_C$
4. AE decoder hidden layer(s):
Size = $N_{AE\ dec} \times N_C$
5. AE output layer:
Size = $N \times N_C$
2. C hidden layer(s):
Size = $N_{C\ hidden}$
3. C output layer:
Size = # classes N_C
Back-propagation
of $\min(-log\mathcal{L}_{ACE} + reg\ constraints)$,
where,
$-log\mathcal{L}_{ACE} = -log\mathcal{L}_{AE} - log\mathcal{L}_C$

Figure 4. ACE architecture: $\mathbf{AE}$ stands for "auto-encoder", $\mathbf{C}$ stands for "classifier". Training is supervised i.e. labels are used in the auto-encoder and each class has a separate decoder, with unimodal sampling in the latent layer. The sampling during testing is instead from a mixture of densities $p(\mathbf{z}|\mathbf{x}_\mu) = \sum_{c=1}^{N_c} \omega_{\mu,c} p(\mathbf{z}|\mathbf{x}_\mu, c)$, with class probabilities $\{\omega_{\mu,c}\}_{c=1}^{N_c}$ provided by the classifier, hence the dashed lines.

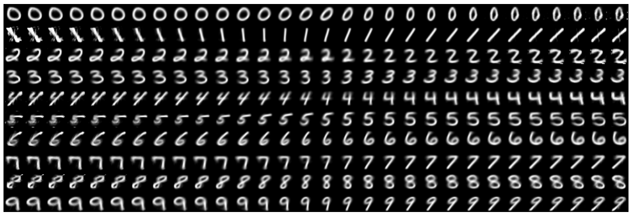

Figure 5. Dominant dimension for each of MNIST ten classes, with each row corresponding to a separate class. While rotational symmetry dominates most classes, size i.e. scaling symmetry, clearly dominates the class of digit 5. The net is an ACE in creative regime as in sub-section 3.1, with an equally spaced deterministic grid in the latent layer $\{\sigma_s\}_{s=1}^{30}$, $-6 \le \sigma_s \le 6$. Layer sizes 784-700-(1 x10)-(700x10)-(784x10) for the AE branch and 784-700-700-700-10 for the C branch, Figure 4 and Appendix A, learning rate = 0.0002, decay = 500 epochs, batch size = 1000.

given observation/state μ is:

$$p(\mathbf{z}|\mathbf{x}_\mu) \sim e^{-2p_\mu^{(h)}|z^{(h)} - h_\mu| - 2p_\mu^{(v)}|z^{(v)} - v_\mu|}, \qquad (1.2)$$

i.e. a two-dimensional Laplacian which fits [4] in the Gibbs machine paradigm (2.3). We demonstrate in sub-section 3.2 how to build-in translational, scaling and rotational symmetry in a net, by computing the *symmetry statistics* like $\{h_\mu, v_\mu\}$ explicitly and estimating the invariants with the rest of the net parameters. In general, they have to be refined via an optimization, as e.g. in (Jadeberg et al., 2015). For more details, see (Georgiev, 2015b).

2. Theoretical background.

2.1. Conditional densities.

When reconstruction is required, the conditional density $p(\mathbf{z}|\mathbf{x}_\mu)$ is central in the generative neural net formalism. For discrete data, the minimization of the negative log-likelihood is equivalent to minimizing the *Kullback-Leibler divergence* $\mathcal{D}(r||q) = \sum_\mathbf{x} r(\mathbf{x}) \log \frac{r(\mathbf{x})}{q(\mathbf{x})}$ between *empirical* $r()$ and model $q()$ densities, since $-\log \mathcal{L}(r||q)$ = $\mathcal{S}(r)$ + $\mathcal{D}(r||q)$, where $\mathcal{S}(r) = \mathbf{E}(-\log r)_r$ is the *entropy* of $r()$. Latents are not a priori given but rather sampled from a closed-form conditional model density $p(\mathbf{z}|\mathbf{x})$. The optimization target hence is the *mixed empirical density* $r(\mathbf{x}, \mathbf{z}) = \frac{1}{P}\sum_\mu \delta(\mathbf{x} - \mathbf{x}_\mu) p(\mathbf{z}|\mathbf{x}_\mu)$ (Kulhavỳ, 1996), section 2.3;(Cover & Thomas, 2006), problem 3.12. The marginal empirical densities are $r(\mathbf{x}) = \frac{1}{P}\sum_\mu \delta(\mathbf{x} - \mathbf{x}_\mu)$, $r(\mathbf{z}) = \frac{1}{P}\sum_\mu p(\mathbf{z}|\mathbf{x}_\mu)$. Hence, the negative log-likelihood of $q(\mathbf{x}) := \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}$ is an arithmetic average[5] across observations $-\mathcal{L}(r||q) = -\frac{1}{P}\sum_\mu \log q(\mathbf{x}_\mu)$. From the Bayes identity, we have in terms of the joint density:

$$\begin{aligned} -\log \mathcal{L}(r||q) &= \mathbf{E}(-\log q(\mathbf{x}))_{r(\mathbf{x})} = \\ &= \mathbf{E}(-\log p(\mathbf{x}, \mathbf{z}))_{r(\mathbf{x},\mathbf{z})} + \mathbf{E}(\log p(\mathbf{z}|\mathbf{x}))_{r(\mathbf{x},\mathbf{z})}. \qquad (2.1) \end{aligned}$$

[4] Technically, Laplacian is not in the exponential class, but it is a sum of two exponential densities in the domains $(-\infty, \mu)$, $[\mu, \infty)$ defined by its mean μ, and those densities are in the exponential class in their respective domains. Laplacian is biologically-plausible because it is a bi-product of squaring Gaussians.

[5] This decomposition does not imply independence of observations: the latent variables can in general contain information from more than one observation, as for example in the case of time series auto-regression.

From the explicit form of $r(\mathbf{x},\mathbf{z})$, for the μ-th observation:

$$-\log q(\mathbf{x}_\mu) = \mathbf{E}(-\log p(\mathbf{x}_\mu,\mathbf{z}))_{p(\mathbf{z}|\mathbf{x}_\mu)} - \mathcal{S}(p(\mathbf{z}|\mathbf{x}_\mu)), \tag{2.2}$$

where $\mathcal{S}(p(\mathbf{z}|\mathbf{x}_\mu)) = \mathbf{E}(-\log p(\mathbf{z}|\mathbf{x}_\mu))_{p(\mathbf{z}|\mathbf{x}_\mu)}$ is the entropy of the model distribution conditional on a given visible observation $\mathbf{x}_\mu$. If we sample the latent observables only once per observation, the right-hand side reduces to the familiar $-\log p(\mathbf{x}_\mu,\mathbf{z}_\mu) + \log p(\mathbf{z}_\mu|\mathbf{x}_\mu)$.

2.2. Conditional independence.

The hidden/latent observables $\mathbf{z} = \{\mathbf{z}_j\}_{j=1}^{N_{lat}}$ are *conditionally independent* if, for a given observation $\mathbf{x}_\mu$, one has $p(\mathbf{z}|\mathbf{x}_\mu) = \prod_{j=1}^{N_{lat}} p(\mathbf{z}_j|\mathbf{x}_\mu)$. From the independence bound of entropy $\mathcal{S}(p(\mathbf{z}|\mathbf{x}_\mu)) \leq \sum_{j=1}^{N_{lat}} \mathcal{S}(p(\mathbf{z}_j|\mathbf{x}_\mu))$, (Cover & Thomas, 2006), Chapter 2, conditional independence minimizes the negative entropy term on the right-hand side of (2.2). Everything else being equal, conditional independence is hence optimal for nets.

2.3. Gibbs and q-Gibbs model densities.

There is a broad class of probability density families - *Gibbs* or *exponential* families - which dominate the choices of model densities, both in physics and neural nets. This class includes a sufficiently large number of density families: Gaussian, Bernoulli, exponential, gamma, etc. Their general closed form is:

$$p_{\boldsymbol{\lambda}}(\mathbf{z}) = \frac{p(\mathbf{z})}{\mathcal{Z}} e^{-\sum_{s=1}^{M} \lambda_s \mathcal{M}_s(\mathbf{z})}, \tag{2.3}$$

where $p(\mathbf{z})$ is an arbitrary *prior* density, $\boldsymbol{\lambda} = \{\lambda_s\}$ are *Lagrange* multipliers, $\mathcal{M}_j(\mathbf{z})$ are so-called *sufficient statistics*, and $\mathcal{Z} = \int p_{\boldsymbol{\lambda}}(\mathbf{z})d\mathbf{z}$ is the normalizing *partition function*. The sufficient statistics in physics form a complete set of state variables like energy, momenta, number of particles, etc, fully describing the μ-th conditional state, sub-section 1.2, see (Landau & Lifshitz, 1980), sections 28,34,35,110. In probability, the sufficient statistics are typically monomials like $\mathcal{M}_1(\mathbf{z}) = \mathbf{z}$, $\mathcal{M}_2(\mathbf{z}) = \mathbf{z}^2$, etc, whose expectations form the moments m_1, m_2, etc, of a given multi-dimensional density. As proposed in sub-section 1.5, one can add to the list the symmetry statistics, see subsection 3.2 for details.

The Gibbs class is special because it is the variational *maximum entropy* class: it is the unique functional form which maximizes the entropy $\mathcal{S}_p(f)$, computed with respect to the reference measure $p()$, across the universe $\{f(\mathbf{z})\}$ of all densities with fixed expectations of the sufficient statistics $\mathbf{E}(\mathcal{M}_s(\mathbf{z}))_f = m_s$, $s = 1,...,M$, see (Cover & Thomas, 2006), chapter 12. The Lagrange multipliers $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\mathbf{m})$ are computed so as to satisfy these constraints and are hence functions of the vector $\mathbf{m} = \{m_s\}_{s=1}^{M}$.

The Gibbs class is special in even stronger sense: it is a *minimum divergence* class. For an arbitrary prior density $p(\mathbf{z})$, the Kullback-Leibler divergence $\mathcal{D}(p_{\boldsymbol{\lambda}}(\mathbf{z})||p(\mathbf{z})))$ minimizes the divergence $\mathcal{D}(f(\mathbf{z})||p(\mathbf{z}))$ across the universe $\{f(\mathbf{z})\}$ of all densities with fixed expectations of the sufficient statistics $\mathbf{E}(\mathcal{M}_s(\mathbf{z}))_f = m_s$, $s = 1,...,M$. This follows from the *variational Pythagorean theorem*, Figure 6, (Chentsov, 1968), (Kulhavỳ, 1996), section 3.3:

$$\mathcal{D}(f(\mathbf{z})||p(\mathbf{z})) = \mathcal{D}(f(\mathbf{z})||p_{\boldsymbol{\lambda}}(\mathbf{z})) + \mathcal{D}(p_{\boldsymbol{\lambda}}(\mathbf{z})||p(\mathbf{z})) \geq \\ \geq \mathcal{D}(p_{\boldsymbol{\lambda}}(\mathbf{z})||p(\mathbf{z})). \tag{2.4}$$

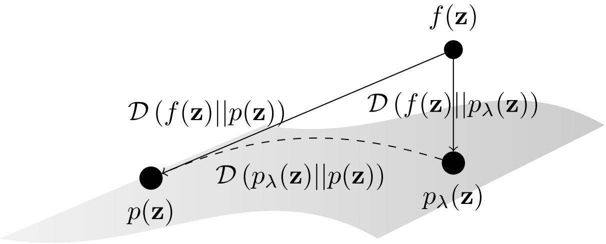
$f(\mathbf{z})$
$\mathcal{D}\left(f(\mathbf{z})||p(\mathbf{z})\right)$
$\mathcal{D}\left(f(\mathbf{z})||p_\lambda(\mathbf{z})\right)$
$\mathcal{D}\left(p_\lambda(\mathbf{z})||p(\mathbf{z})\right)$
$p(\mathbf{z})$
$p_\lambda(\mathbf{z})$

Figure 6. A naive visualization of the probabilistic (variational) Pythagorean theorem from (2.4).

In our context, for a given observation $\mathbf{x}_\mu$, choosing a wave function from the Gibbs type is equivalent to choosing a conditional model density:

$$p(\mathbf{z}|\mathbf{x}_\mu) = \frac{p(\mathbf{z})}{\mathcal{Z}} e^{-\sum_{s=1}^{M} \lambda_s(\mathbf{m}(\mathbf{x}_\mu))\mathcal{M}_s(\mathbf{z})}. \tag{2.5}$$

We made explicit the indirect dependence of λ_s on $\mathbf{x}_\mu$ via the constraints vector $\mathbf{m} = \mathbf{m}(\mathbf{x}_\mu)$. As we will see in sub-section 3.1, minimizing the divergence $\mathcal{D}(f(\mathbf{z})||p(\mathbf{z}))$ across an unknown a priori family of conditional distributions $p(\mathbf{z}|\mathbf{x}_\mu) = f(\mathbf{z})$, is crucial for the quality of a generative net. The minimum divergence property (2.4) implies that we are always better off choosing $p(\mathbf{z}|\mathbf{x}_\mu)$ from a Gibbs class, as in (2.5), hence the name Gibbs machines.

In practice, in order for $p(\mathbf{z}|\mathbf{x}_\mu)$ to be tractable, $p(\mathbf{z})$ has to be from a specific parametric family within the Gibbs class: then both $p(\mathbf{z})$ and $p(\mathbf{z}|\mathbf{x}_\mu)$ will be tractable and in the same family, e.g. Gaussian, exponential, etc.

Except for the symmetry statistics, sub-sections 1.5, 3.2, the conditional moments $\mathbf{m} = \mathbf{m}(\mathbf{x}_\mu)$ for the μ-th quantum state are free parameters. Together with the Lagrange multipliers of the symmetry statistics, they can be thought of as *quantum numbers* distinguishing the partial equilibrium states from one another, section 1.2. These quantum numbers are added to the rest of the free net parameters, to be optimized by standard methods, like back-propagation/ stochastic gradient descent, etc.

Gibbs densities are only a special case (for $q = 1$) of the broad class of q-Gibbs (or q-exponential) densities. The corresponding *nonextensive* statistical mechanics (Tsallis, 2009), describes more adequately long-range-interacting many-body systems like typical human-generated data sets. Many of the properties of the exponential class remain true for the q-exponential class (Amari & Ohara, 2011), see open problem 4 in section 5.

3. Application to neural nets.

3.1. Fully-generative nets. Gibbs machines.

In order to have a fully-generative net, one needs to choose a marginal model density $p(\mathbf{z})$ and sample directly from it, unencumbered by observations $\{\mathbf{x}_\mu\}$. We will hence distinguish two separate regimes of operation:

- **creative**: Latent observables $\mathbf{z} = \{\mathbf{z}_j\}_{j=1}^{N_{lat}}$ are sampled from a closed-form model density $p(\mathbf{z})$, unencumbered by observations $\{\mathbf{x}_\mu\}$. A closed-form *reconstruction* model density $p^{rec}(\mathbf{x}|\mathbf{z})$ is also chosen, which defines a joint density $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p^{rec}(\mathbf{x}|\mathbf{z})$. Unfortunately, the implied posterior conditional $q(\mathbf{z}|\mathbf{x}) := p(\mathbf{x}, \mathbf{z})/\int p(\mathbf{x}, \mathbf{z})d\mathbf{z}$ $= p(\mathbf{x}, \mathbf{z})/q(\mathbf{x})$ is generally intractable.

- **non-creative**: Latent observables are sampled from a closed-form model conditional density $p(\mathbf{z}|\mathbf{x})$ with observations $\{\mathbf{x}_\mu\}$ attached to the net, as is common in training, validation or testing. The same closed-form densities as above are used but of course $p(\mathbf{z}|\mathbf{x}) \neq q(\mathbf{z}|\mathbf{x})$. The empirical densities are the same as in sub-section 2.1.

Expanding the creative joint density in both Bayesian directions, one gets for the negative log-likelihood of $q(\mathbf{x})$ $:= \int p(\mathbf{x}, \mathbf{z})d\mathbf{z}$ vs. the non-creative empirical density:

$$-\log \mathcal{L}(r||q) = \mathbf{E}(-\log p^{rec}(\mathbf{x}|\mathbf{z}))_{r(\mathbf{x},\mathbf{z})} - \mathbf{E}(-\log p(\mathbf{z}))_{r(\mathbf{z})} + \mathbf{E}(\log q(\mathbf{z}|\mathbf{x}))_{r(\mathbf{x},\mathbf{z})}. \qquad (3.1)$$

From the explicit form of $r(\mathbf{x}, \mathbf{z})$, for the μ-th observation:

$$-\log q(\mathbf{x}_\mu) = \mathbf{E}(-\log p^{rec}(\mathbf{x}_\mu|\mathbf{z}))_{p(\mathbf{z}|\mathbf{x}_\mu)} - \mathbf{E}(-\log p(\mathbf{z}))_{p(\mathbf{z}|\mathbf{x}_\mu)} + \mathbf{E}(\log q(\mathbf{z}|\mathbf{x}_\mu))_{p(\mathbf{z}|\mathbf{x}_\mu)}. \qquad (3.2)$$

Adding and subtracting $\mathcal{S}(p(\mathbf{z}|\mathbf{x}_\mu))$, this can be re-written:

$$-\log q(\mathbf{x}_\mu) = \overbrace{\mathbf{E}(-\log p^{rec}(\mathbf{x}_\mu|\mathbf{z}))_{p(\mathbf{z}|\mathbf{x}_\mu)}}^{reconstruction\ error} + + \underbrace{\mathcal{D}(p(\mathbf{z}|\mathbf{x}_\mu)||p(\mathbf{z}))}_{generative\ error} - \underbrace{\mathcal{D}(p(\mathbf{z}|\mathbf{x}_\mu)||q(\mathbf{z}|\mathbf{x}_\mu))}_{variational\ error}. \qquad (3.3)$$

The *reconstruction error* measures the negative likelihood of getting $\mathbf{x}_\mu$ back, after the transformations and randomness inside the net. The *generative error* is the divergence between the generative densities in the non-creative and creative regimes. Crucially, it can be interpreted as the hypotenuse in the variational Pythagorean theorem (2.4). Minimizing the variational error is hard for real-life data sets, because of the intractability of $q(\mathbf{z}|\mathbf{x}_\mu)$, see open problem 7 in section 5. Dropping it yields an upper bound for the negative log-likelihood[6]:

$$-\log q(\mathbf{x}_\mu) \le \mathcal{U}(-\log q(\mathbf{x}_\mu)) := \underbrace{\mathbf{E}(-\log p^{rec}(\mathbf{x}_\mu|\mathbf{z}))_{p(\mathbf{z}|\mathbf{x}_\mu)}}_{reconstruction\ error} + \underbrace{\mathcal{D}(p(\mathbf{z}|\mathbf{x}_\mu)||p(\mathbf{z}))}_{generative\ error}. \qquad (3.4)$$

It is clear from the above derivation that (3.3), (3.4) are universal for fully-generative nets and were first used in the first fully-generative nets, the VAE-s (Kingma & Welling, 2014), (Rezende et al., 2014). The VAE-s owe their name to the variational error term and were introduced in the context of very general sampling densities. From the variational Pythagorean theorem (2.4), under certain constraints, sampling densities (2.5) from the Gibbs class minimize the generative error. Hence we call the respective nets Gibbs machines. While the variational error is due to an approximation, the variational principle from which the Gibbs class is derived, is fundamental to statistical mechanics.

Without offering a rigorous proof, we believe that the argument from sub-section 2.2 can be generalized to this context: For a given reconstruction error, the generative error in (3.4) is minimized when the latent variables are conditionally independent. For Gaussian multi-variate sampling, this follows from the explicit form of the generative error (Gil et al., 2013), table 3, and *Hadamard's* inequality (Cover & Thomas, 2006), chapter 8.

3.2. Latent symmetry statistics. Momenta.

We will show for brevity only how to build spatial invariances in a 2-dim square visual recognition model. For real-life data sets, the symmetry statistics have to be computed via another net, see (Georgiev, 2015b).

Every observable i.e. pixel $\mathbf{x}_i$, $i = 0, ..., N$, can be assigned a horizontal h_i and a vertical v_i integer coordinates on the screen, e.g., $h_i, v_i \in \{1, ..., \sqrt{N}\}$. In these coordinates, a row-observation $\mathbf{x}_\mu = \{x_{\mu i}\}_{i=1}^{N}$ becomes a matrix-observation $\{x_{\mu,h_i,v_i}\}$ and a net layer of size N becomes a layer of size $\sqrt{N} \times \sqrt{N}$. The *center of mass* (h_μ, v_μ):

$$h_\mu := \frac{\sum_{i=1}^{N} x_{\mu i} h_i}{\sum_i x_{\mu i}}, \quad v_\mu := \frac{\sum_{i=1}^{N} x_{\mu i} v_i}{\sum_i x_{\mu i}}. \qquad (3.5)$$

of every observation defines latent symmetry statistics $\mathbf{h}$ $= \{h_\mu\}_{\mu=1}^{P}$ and $\mathbf{v} = \{v_\mu\}_{\mu=1}^{P}$, see sub-sections 1.5, 2.3. Without loss of generality, we assumed here that $x_{\mu i} \ge 0$, hence $0 \le h_\mu, v_\mu \le \sqrt{N}$. In the coordinate system cen-

[6] This is an expanded version of the textbook *variational inequality* (Cover & Thomas, 2006), Exercise 8.6, hence the name *variational error*.

tered by (h_μ, v_μ), we have for every μ the new coordinates:

$$(\hat{h}_{\mu i}, \hat{v}_{\mu i}) := (h_i - h_\mu, v_i - v_\mu), \tag{3.6}$$

$-\sqrt{N} \le \hat{h}_{\mu i}, \hat{v}_{\mu i} \le \sqrt{N}$. In this coordinate system, every pixel has polar coordinates $r_{\mu i} := \sqrt{\hat{h}^2_{\mu i} + \hat{v}^2_{\mu i}}$, $\varphi_{\mu i}$ $:= atan2(\hat{v}_{\mu i}, \hat{h}_{\mu i}) \in (-\pi, \pi]$, hence the new symmetry statistics: *scale* $\mathbf{r} = \{r_\mu\}_{\mu=1}^P$ and *angle* $\boldsymbol{\varphi} = \{\varphi_\mu\}_{\mu=1}^P$:

$$r_\mu := \frac{\sum_{i=1}^N x_{\mu i} r_{\mu i}}{\sum_i x_{\mu i}}, \quad \varphi_\mu := \frac{\sum_{i=1}^N x_{\mu i} \varphi_{\mu i}}{\sum_i x_{\mu i}}, \tag{3.7}$$

$0 < r_\mu \le \sqrt{2N}$, $-\pi < \varphi_\mu \le \pi$. In order to un-scale and un-rotate an image, change coordinates one more time to:

$$(\tilde{h}_{\mu i}, \tilde{v}_{\mu i}) := \frac{C}{r_\mu} (\hat{h}_{\mu i}, \hat{v}_{\mu i}) \begin{pmatrix} \cos(-\varphi_\mu) & \sin(-\varphi_\mu) \\ \sin(\varphi_\mu) & \cos(-\varphi_\mu) \end{pmatrix}, \tag{3.8}$$

$-M \le \tilde{h}_{\mu i}, \tilde{v}_{\mu i} \le M$, for some constants C, M depending[7] on $\min_\mu r_\mu$, $\max_\mu r_\mu$. After rounding and a shift, we thus have for any observation μ, an index mapping:

$$\begin{aligned} \{1, ..., N\} &\to \{1, ..., (2M+1)^2\} \\ i &\to \tilde{h}_{\mu i} + (\tilde{v}_{\mu i} - 1)(2M+1). \end{aligned} \tag{3.9}$$

When $2M+1 > \sqrt{N}$, the $(2M+1-\sqrt{N})^2$ indexes which are not in the mapping image, correspond to identically zero observables for that observation. In the coordinates (3.8), a layer of size N becomes a layer of size $(2M+1)^2$.

In summary: i) for auto-encoders, apply (3.9) at the input and its inverse at the output of the net; ii) for classifiers, apply (3.9) at the input only; iii) for both, include in addition the symmetry statistics $\mathbf{h}, \mathbf{v}, \mathbf{r}, \boldsymbol{\varphi}$ in the latent layer, if needed, see (Georgiev, 2015b). The prior model density $p()$ of symmetry statistics can be assumed equal to their parametrized posterior $p(|\mathbf{x}_\mu)$, but there are other options.

When sampling the symmetry statistics from independent Laplacians as in (1.2) e.g., the respective density means are set to be $h_\mu, v_\mu, r_\mu, \varphi_\mu$ from (3.5), (3.7). The density scales $\sigma^h_\mu, \sigma^v_\mu, \sigma^r_\mu, \sigma^\varphi_\mu$ on the other hand are free parameters, and can in principle be optimized in the non-creative regime, alongside the rest of the net parameters, sub-section 2.3. As argued in sub-section 1.5, the inverted scales are the scaled *momenta*. In the creative regime, when sampling e.g. from $\mathbf{h}$ alone, one will get horizontally shifted identical replicas. See open problem 2, section 5.

4. Experimental results.

The Theano (Bastien et al., 2012) code used for experiments is in (Georgiev, 2015a), see also (Popov, 2015).

4.1. Non-generative ACE.

The motivation for the non-generative ACE comes from the Einstein observation entropies $\{-\log p^G(\mathbf{x}_\mu)\}_\mu$, as in sub-section 1.4, and their relation to singular value decomposition (SVD). Recall that the SVD of the $B \times N$ data matrix $\mathbf{X}$ with B observations and N observables is $\mathbf{X}$ $= \mathbf{V}\boldsymbol{\Lambda}\mathbf{W}^T$. The $B \times B$ matrix $\mathbf{V}\mathbf{V}^T$ is i) a projection mapping; ii) its diagonals are up to a constant the negative Gaussian log-likelihoods $\{-\log p^G(\mathbf{x}_\mu)\}_\mu$; and iii) it is invariant on $\mathbf{X}$, i.e. $\mathbf{X} = \mathbf{V}\mathbf{V}^T\mathbf{X}$.

Let us consider a shallow auto-encoder, Figure 7, left, and its dual in the space of observations, Figure 7, right. It can be shown that for tied weights $\mathbf{V}^{(2)} = \mathbf{V}^{(1)T}$, in the absence of non-linearities, the optimal $B \times N_{lat}$ hidden layer solution $\mathbf{H}_o$ on the left is $\mathbf{H}_o = \mathbf{V}^{(2)} = \mathbf{V}$ (Georgiev, 2015c). The divergence of $\mathbf{V}\mathbf{V}^T\mathbf{X}$ from $\mathbf{X}$ is thus the reconstruction error in the dual space and minimizing it gets us closer to the optimal $\mathbf{H}_o$. If we treat the first hidden layer $\mathbf{H}$ of a classifier as the rescaled dual weight matrix $\sqrt{B}\mathbf{V}^{(2)}$, we arrive at the dual reconstruction error :

$$-\log \mathcal{L}'_{recon} = \frac{1}{B} \sum_{i=1}^N \mathbf{E}(-\log \varphi(\mathbf{H}\mathbf{H}^T\mathbf{x}_i/B))_{\mathbf{x}_i}, \tag{4.1}$$

for a given column-vector observable $\mathbf{x}_i = \{x_{\mu i}\}_{\mu=1}^B$ and sigmoid non-linearity $\varphi()$, see Appendix A.

The non-generative ACE has as minimization target the composite negative log-likelihood (1.1), with $-\log \mathcal{L}_{AE}$ replaced by the dual reconstruction error (4.1). The orthogonality $\mathbf{V}^T\mathbf{V} = \mathbf{I}_{N_{lat}}$ of $\mathbf{V}$ implies the need for an additional *batch normalization*, similar to (Ioffe & Szegedy, 2015), see Appendix A.

The best known results for the test classification error of feed-forward, non-convolutional nets, without artificial data augmentation, are in the 0.9-1% handle (Srivastava et al., 2014), table 2. As shown on the right of Figure 8, the non-generative ACE offers a 20-30% improvement.

4.2. Generative ACE.

The architecture is in Figure 4, the minimization target is the ACE negative log-likelihood (1.1), with $-\log \mathcal{L}_{AE}$ replaced by the upper bound (3.4). Laplacian sampling density is used in training and the mixed Laplacian in testing, with the explicit formulas for the generative error in Appendix B. The generative ACE produces similarly outstanding classification results as the non-generative ACE on the regular MNIST data set, Figure 9, left. Even without tweaking hyper-parameters, it also produces outstanding results for the density estimation of the binarized MNIST data set, Figure 9, right. An upper bound for the negative log-likelihood in the 86-87 handle is in the ballpark of the best non-recurrent nets, (Gregor et al., 2015), table 2.

[7] The scale r_μ typically needs to have a lower bound, in order to ensure that M is of the same order of magnitude as $\sqrt{N}$.

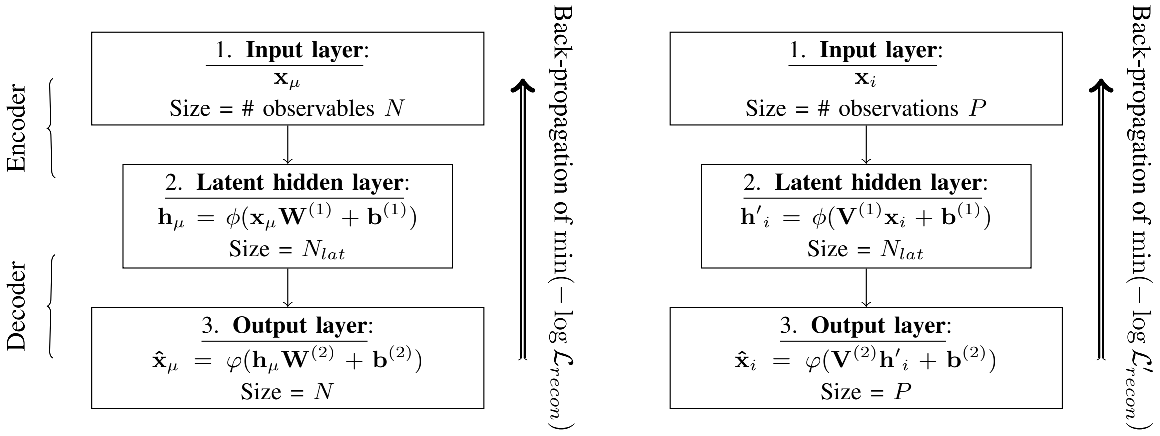
Encoder
Decoder
1. Input layer:
$\mathbf{x}_\mu$
Size = # observables N
2. Latent hidden layer:
$\mathbf{h}_\mu = \phi(\mathbf{x}_\mu \mathbf{W}^{(1)} + \mathbf{b}^{(1)})$
Size = N_{lat}
3. Output layer:
$\hat{\mathbf{x}}_\mu = \varphi(\mathbf{h}_\mu \mathbf{W}^{(2)} + \mathbf{b}^{(2)})$
Size = N
Back-propagation of $\min(-\log \mathcal{L}_{recon})$
1. Input layer:
$\mathbf{x}_i$
Size = # observations P
2. Latent hidden layer:
$\mathbf{h'}_i = \phi(\mathbf{V}^{(1)}\mathbf{x}_i + \mathbf{b}^{(1)})$
Size = N_{lat}
3. Output layer:
$\hat{\mathbf{x}}_i = \varphi(\mathbf{V}^{(2)}\mathbf{h'}_i + \mathbf{b}^{(2)})$
Size = P
Back-propagation of $\min(-\log \mathcal{L}'_{recon})$

Figure 7. Shallow auto-encoder in the space of observables (left) and observations (right). Minimization targets are the reconstruction errors in the respective spaces $-\log \mathcal{L}_{recon}$ and $-\log \mathcal{L}'_{recon}$ defined for binarized data in Appendix A, ϕ, φ are non-linearities.

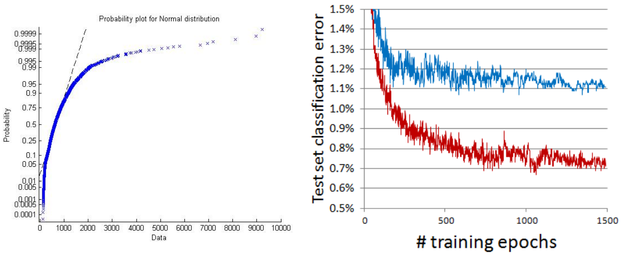

Figure 8. **Left.** The same Q-Q plot as on the right of Figure 1, but for the non-generative ACE, sub-section 4.1, with the same hyper-parameters as on the right of of Figure 1. **Right.** Classification error for the MNIST 10000 test set, as a function of training *epochs*, i.e., one full swipe over all training observations. The top line is the standard classifier as on the right of Figure 1. The bottom line is the classification error of the non-generative ACE with the same hyper-parameters.

5. Open problems.

1. Use the freely available intricates, sub-section 1.4, directly as feature detectors, in lieu of artificially computed Independent Component Analysis (ICA) features (Hyvarinen et al., 2009).
2. Test empirically the performance of ACE, with the symmetry statistics added as in 3.2, and computed either in closed form, or from specialized nets, as in (Jadeberg et al., 2015). For the distorted MNIST and CIFAR10 datasets, see (Georgiev, 2015b).
3. Deepen and make generative the shallow dual encoder of the non-generative ACE, sub-sections 1.4, 4.1.
4. Test empirically q-Gibbs machines, with negative log-likelihoods replaced by their q-equivalents and sampling from q-Gibbs densities, subsection 2.3.
5. Improve the upper bound for the generative error of mixture densities in Appendix B, by using variational methods as in (Hershey & Olsen, 2007).

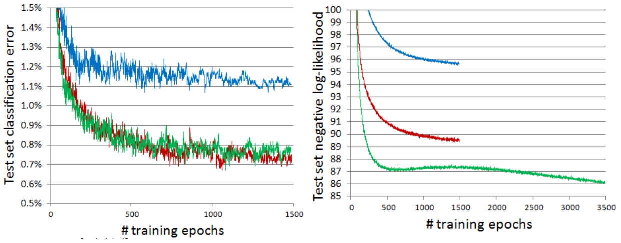

Figure 9. **Left.** Classification error for the MNIST 10000 test set. Top line is from a standard classifier net as on the right of Figure 1. Bottom lines are from generative ACE in classification mode: Gaussian sampling (red) and Laplacian (green). Layer sizes 784-700-(100x10)-(700x10)-(784x10) for the AE branch and 784-700-700-700-10 for the C branch, Figure 4 and Appendix A, learning rate = 0.0015, decay = 500 epochs, batch size = 10000. The dual reconstruction error $-\log \mathcal{L}'_{recon}$ from sub-section 4.1 is added to the overall cost. **Right.** Upper bound (3.4) of the negative log-likelihood for the binarized MNIST 10000 test set. The top line is the standard Gibbs machine with Gaussian sampling, layer sizes 784-700-400-700-784 and other hyper-parameters as below. The middle line is the same net but with Laplacian sampling. The bottom line is generative ACE with Laplacian mixture sampling. Layer sizes 784-700-(400x10)-(700x10)-(784x10) for the AE branch and 784-700-700-700-10 for the C branch, Figure 4 and Appendix A, learning rate = 0.0002, decay = 500 epochs, batch size = 1000.

6. How is the ACE blend of exponential and mixture densities related to the beautiful duality between these two families, underlying information geometry (Amari & Nagaoka, 2000), section 3.7?
7. Minimize the variational error in (3.4), by exploiting the special variational Bayes properties of the exponential class, (Gelman et al., 2003), section 2.4.

Acknowledgments

We appreciate motivating discussions with Ivaylo Popov and Nikola Toshev. Credit goes to Christine Haas for coining the terms *intricates* and *creative/non-creative* regimes.

References

Amari, S. and Nagaoka, H. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical monographs*. Oxford University Press, 2000.

Amari, Shun-ichi and Ohara, Atsumi. Geometry of q-exponential family of probability distributions. *Entropy*, 13(6):1170, 2011.

Bastien, Frédéric, Lamblin, Pascal, Pascanu, Razvan, Bergstra, James, Goodfellow, Ian J., Bergeron, Arnaud, Bouchard, Nicolas, and Bengio, Yoshua. Theano: new features and speed improvements, 2012.

Chentsov, N.N. Nonsymmetrical distance between probability distributions, entropy and the theorem of Pythagoras. *Mathematical notes of the Academy of Sciences of the USSR*, 4(3):686–691, 1968.

Coates, A., Lee, H., and Ng, A. Y. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

Cover, Thomas M. and Thomas, Joy A. *Elements of Information Theory, 2nd Edition*. Wiley-Interscience, 2006.

Einstein, Albert. On Boltzmanns principle and some immediate consequences thereof. In Damour, Thibault, Darrigol, Olivier, Duplantier, Bertrand, and Rivasseau, Vincent (eds.), *Einstein, 19052005*, volume 47 of *Progress in Mathematical Physics*, pp. 183–199. Birkhuser Basel, 2006.

Gelfand, I.M. and Fomin, S.V. *Calculus of Variations*. Prentice-Hall, 1963.

Gelman, Andrew, Carlin, John B., Stern, Hal S., and Rubin, Donald B. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2003.

Georgiev, Galin. ACE, 2015a. URL `https://github.com/galinngeorgiev/ACE`.

Georgiev, Galin. Symmetries and control in generative neural nets., 2015b. arXiv:1511.02841.

Georgiev, Galin. Duality between observables and observations in neural nets., 2015c. to appear.

Gil, M, Alajaji, F, and Linder, T. Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences*, 249:124–131, 2013.

Goodfellow, Ian J., Warde-Farley, David, Mirza, Mehdi, Courville, Aaron, and Bengio, Yoshua. Maxout networks, 2013. arXiv:1302.4389.

Gregor, Karol, Danihelka, Ivo, Graves, Alex, and Wierstra, Daan. DRAW: A recurrent neural network for image generation, 2015. arXiv:1502.04623.

Hershey, J.R. and Olsen, P.A. Approximating the Kullback Leibler divergence between Gaussian mixture models. In *ICASSP*, volume 4, pp. 317–320, April 2007.

Hinton, Geoffrey E., Revow, Michael, and Dayan, Peter. Recognizing handwritten digits using mixtures of linear models. In *AINIPS*, volume 7, pp. 1015–1022, 1995.

Hyvarinen, Aapo, Hurri, Jamo, and Hoyer, Patrick O. *Natural Image Statistics: A probabilistic approach to early computational vision*. Springer-Verlag, 2009.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. arXiv:1502.03167.

Jadeberg, Max, Symonyan, Karen, Zisserman, Andrew, and Kavukcuoglu, Koray. Spatial transformer networks, 2015. arXiv:1412.6980.

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Kingma, Durk P. and Welling, Max. Auto-encoding variational Bayes. In *ICLR*, 2014.

Kingma, Durk P., Rezende, Danilo J., Mohamed, Shakir, and Welling, Max. Semi-supervised learning with deep generative models. 2014. arXiv:1406.5298.

Kulhavỳ, R. *Recursive Nonlinear Estimation: A Geometric Approach*. Lecture Notes in Control And Information Sciences. 1996.

Landau, L.D. and Lifshitz, E.M. *Quantim Mechanics, Nonrelativistic theory, 3rd edition*. Pergamon Press, 1977.

Landau, L.D. and Lifshitz, E.M. *Statistical Physics, Part 1, 3rd edition*. Elsevier Science, 1980.

LeCun, Yann, Cortes, Corinna, and Burges, Christopher J.C. MNIST handwritten digit database, 1998. URL `http://yann.lecun.com/exdb/mnist/`.

Mardia, K. V., Kent, J. T., and Bibby, J. M. *Multivariate Analysis*. Academic Press, 1979.

Popov, Ivaylo. Theano-Lights, 2015. URL `https://github.com/Ivaylo-Popov`.

Rezende, Danilo J., Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *JMLR*, volume 32, 2014.

Rifai, Salah, Bengio, Yoshua, Dauphin, Yann, and Vincent, Pascal. A generative process for sampling contractive auto-encoders. In *ICML*, 2012.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

Tsallis, Constantino. *Introduction to Nonextensive Statistical Mechanics: Approaching a Complex World /*. Springer New York,, 2009.

Appendices

A. Software and implementation.

The optimizer used is Adam (Kingma & Ba, 2015) stochastic gradient descent back-propagation. Specific hyper-parameters are in the text. We used only two standard sets of hyper-parameters, one for the classifier branch and one for the auto-encoder branch, no optimizations.

For reconstruction error of a binarized μ-th row-observation $\mathbf{x}_\mu = \{x_{\mu i}\}_{i=1}^N$ and its reconstruction $\hat{\mathbf{x}}_\mu$, we use the standard binary cross-entropy (Bastien et al., 2012): $\mathbf{E}(-\log \hat{\mathbf{x}}_\mu)_{\mathbf{x}_\mu} = \sum_i (-x_{\mu i} \log \hat{x}_{\mu i} - (1-x_{\mu i})(1-\log \hat{x}_{\mu i}))$, with $\hat{\mathbf{x}}_\mu$ the image of a sigmoid non-linearity φ. The batch negative log-likelihood is $-\log \mathcal{L}_{recon} = \frac{1}{B} \sum_{\mu=1}^B \mathbf{E}(-\log \hat{\mathbf{x}}_\mu)_{\mathbf{x}_\mu}$. In the space of observations, as on right plot of Figure 7, the dual reconstruction error is the same binary cross-entropy $\mathbf{E}(-\log \hat{\mathbf{x}}_i)_{\mathbf{x}_i}$, but for the i-th observable $\mathbf{x}_i = \{x_{\mu i}\}_{\mu=1}^B$ and a sum over μ instead of i. The batch negative log-likelihood is $-\log \mathcal{L}'_{recon} = \frac{1}{B} \sum_{i=1}^N \mathbf{E}(-\log \hat{\mathbf{x}}_i)_{\mathbf{x}_i}$, with a normalization factor conforming to the space of observables.

The non-linearities are $\tanh()$ in the auto-encoder branch and two-unit *maxout* (Goodfellow et al., 2013) in the classifier branch. Weight matrices of size $P \times N$ are initialized as random Gaussian matrices, normalized by the order of magnitude $\sqrt{P} + \sqrt{N}$ of their largest eigen-value . As discussed in sub-section 4.1, hidden observables in the first and last hidden layer of classifiers are batch-normalized i.e. de-meaned and divided by their second moment. Unlike (Ioffe & Szegedy, 2015), batch normalization is enforced identically in both the train and test set, hence test results depend slightly on the test batch.

B. Generative error formulas.

When sampling from a Laplacian, in order to have an unity variance in the prior, we choose for the independent one-dimensional latents a prior $p(z) = p^{Lap}(z; 0, \sqrt{0.5})$, where $p^{Lap}(z; \mu, b) = exp(-|z - \mu|/b)/(2b)$ is the standard Laplacian density with mean μ and scale b. In order to have zero generative error when $(\mu, \sigma) \to (0, 1)$, we parametrize the conditional posterior as $p(z|.) = p^{Lap}(z; \mu, \sigma\sqrt{0.5})$. The generative error in (3.4) equals: $-\log \sigma + |\mu|/\sqrt{0.5} + \sigma exp\left(-|\mu|/(\sigma\sqrt{0.5})\right) - 1$, see (Gil et al., 2013), table 3.

For the divergence between a mixture prior $\sum_s \alpha_s p_s(z)$ and a mixture posterior $\sum_s \alpha_s p_s(z|.)$ with the same weights $\{\alpha_s\}_s$, we use the upper bound $\sum_s \alpha_s \mathcal{D}\left(p_s(z|.)||p_s(z)\right)$ implied by the log sum inequality (Cover & Thomas, 2006). For improvements, see open problem 5 in section 5.