# GENERATIVE MODELING WITH ONE RECURSIVE NETWORK

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We propose to train a multilayer perceptron simultaneously as an encoder and a decoder in order to create a high quality generative model. In one call a network is optimized as either an encoder or decoder, and in a second recursive call the network uses its own outputs to learn the remaining corresponding function, allowing for the minimization of popular statistical divergence measures over a single feed-forward function. This new approach derives from a simple reformulation of variational bayes and extends naturally to the domain of Generative Adversarial Nets. Here we demonstrate a single network which learns a generative model via an adversarial minimax game played against itself. Experiments demonstrate comparable efficacy for the single-network approach versus corresponding multi-network formulations.

## 1 INTRODUCTION

The last half-century of artificial intelligence research has seen a profound shift from highly engineered expert systems to weakly-biased models (Gallant & Gallant, 1993). The algorithm Deep Blue, a canonical example, used expert-designed heuristics combined with tree search for superhuman play at chess. Twenty years later, AlphaGo Zero was able to *learn* superhuman play for the game of Go (or chess) given no prior knowledge of the game at all (Campbell et al., 2002; Silver et al., 2016; 2017).

Meanwhile, the field of generative modeling has recently seen significant advances. Kingma & Welling (2013) provide a parametrization of latent variable methods and a link to auto-encoders which has unlocked the power of deep learning in variational inference, namely the Variational Auto-encoder (VAE). Goodfellow et al. (2014) introduced Generative Adversarial Nets (GAN), which puts two networks in an adversarial game: one tries to distinguish real from fake data, and the other to fool this network by generating high quality examples. The GAN is an extremely succesful approach, having been used to generate photorealistic samples (Zhang et al., 2019).

Both the VAE and GAN methods minimize divergence measures, Kullback Leibler and Jensen Shannon respectively, which have been shown to be members of the $f$-divergence family (Nowozin et al., 2016; Nguyen et al., 2010). This statistical relationship, along with a relationship to the classic wake-sleep algorithm, has instigated significant effort to combine, relate, or unify the models (Hu et al., 2017). Examples include the Adversarial Generative Encoder and $\alpha$-GAN, which propose to combine the adversarial game with an additional reconstruction loss (Ulyanov et al., 2017; Rosca et al., 2017). Other proposals introduce a third auxiliary network to learn an additional divergence, such as Adversarial Autoencoders (Makhzani et al., 2015) and the VAE-GAN (Larsen et al., 2016).

We propose instead to remove a network for the VAE and GAN methods. In light of the trend towards increasingly unbiased systems, we aim to replace a small amount of human design, the distinction of separate encoder and decoder spaces, with a learning algorithm. Our first contribution is the derivation of a single-network VAE, based on our claim that the approximate variational parameter has been unnecessarily carried into modern deep learning practices. Our second contribution is the extension of this technique to the GAN, allowing for a single network to learn a generative function in a recursive self-adversarial game. Empirical results are presented for both algorithms. The single-network VAE is found comparable in efficiency to similar two-network implementations. Furthermore, the single-network GAN exhibits regularization which we argue aids in the algorithm's convergence and gives an edge over the two-network approach.

The single-network formulation of these methods opens new directions for deep learning research and for the unification of deep generative models.

## 2 METHOD

We would like to train a multilayer perceptron as a generative model. Due to the analytic complexity of the perceptron, the marginal likelihood of the networks parameters, denoted $\theta$, w.r.t. some data $X$, is intractable. We also stray from two-sample parametric tests, such as Maximum Mean Discrepancy, due to their sensitivity to Monte Carlo error (Dziugaite et al., 2015; Li et al., 2015). Finally, we do not introduce an auxiliary network.

Define a multilayer perceptron $F$ parametrized by $\theta$. In the following sections, the optimization of $F$ will induce the recursive interpretation, $F(z) = x$ and $F(F(z)) = F^2(z) = \hat{z}'$, where $z'$ may be from any probability space. This interpretation follows from the underlying graphical structure of the VAE and GAN algorithms. See 1. In the case of the VAE, $\hat{z}'$ will be the more constrained $\hat{z}$, which approximates $z \sim \mathcal{N}(0, I)$.

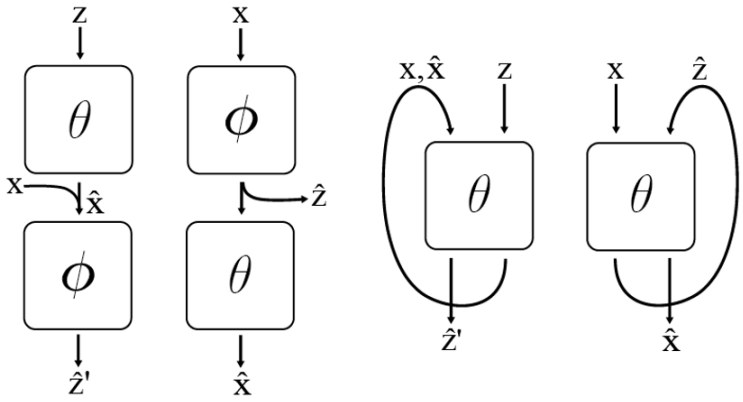


Figure 1: Computational graphs. From left to right: two-network GAN, two-network VAE, one-network GAN, one-network VAE

One may consider the single-network formulation a weakly-informative prior for our desired model, since it is intentionally less engineered than our full knowledge may permit (the correspondence of functions to distinct topologies).

We claim that the recursive multilayer perceptron for the GAN and VAE algorithms yields:

1. A feed-forward generator without a separate auxiliary network.
2. A single parameter space for data generation or inference, which may allow deeper understanding of the parametric relationship between the encoder and decoder. In the case of variational bayesian inference, the addition of a second space is an assumption used to approximate the true posterior. By using one space we are able to remove this assumption for the VAE. This change is discussed further momentarily.
3. A single-step optimization algorithm for the VAE and GAN.
4. An expanded scope of sequence models with respect to conditional distributions and generative modeling.

### 2.1 A NON-VARIATIONAL LOWER BOUND

Consider an i.i.d. dataset $X = \{x_1, ..., x_n\}$. A common technique in Bayesian statistics is to relate $X$ to a latent variable $z$, typically with a simple prior $p(z)$, via some conditional or joint distribution $p(x, z)$. Define the family of models $p_\theta(x, z) = p_\theta(x|z)p(z)$ with $\theta$ some finite dimensional parameter. It is of interest to perform a maximum likelihood estimation or maximum a posteriori

estimate of the value $\theta$ such that our model closely resembles the true distribution $p(x, z)$. Consider the lower bound of the marginal log-likelihood

$$\log p_\theta(x) \geq -D_{KL}(p_\theta(z|x)||p(z)) + \mathbb{E}_{p_\theta(z|x)} \log p_\theta(x|z) = \mathcal{L}(\theta; x) \tag{1}$$

a consequence of Jensen's Inequality. Note that this is exactly the variational lower bound or ELBO, except that we have used $p_\theta(z|x)$ in place of the variational distribution $q_\phi(z|x) \approx p(z|x)$. Variational methods introduce $q_\phi$ as a seperate parameter space from $\theta$ in order to approximate the optimal $\theta^*$. Classicaly $q_\phi$ is assumed to come from a mean-field variational family which contains approximate posterior densities to the true distribution, as is the case in the lower-bound derivation from Jordan et al. (1999) or the original Expectation Maximization algorithm (Dempster et al., 1977). In modern deep learning practices, backpropogation over complex network topologies is used in place of mean-field equations, yet the variational approximation $q_\phi$ is still used (Kingma & Welling, 2013; Zhao et al., 2017). We aim to move beyond the variational assumption, and approximate the optimal $p_{\theta^*}(x, z)$ directly with the non-variational lower bound.

### 2.2 The pseudo-variational auto-encoder

The VAE algorithm uses the original variational lower bound with approximate posterior $q_\phi$

$$\log p_\theta(x) \geq -D_{KL}(q_\phi(z|x)||p_\theta(z|x)) + \mathbb{E}_{q_\theta(z|x_i)} \log p_\theta(x|z) = \mathcal{L}(\theta, \phi; x) \tag{2}$$

where $\phi$ and $\theta$ parametrize seperate multilayer perceptron models. While one may like to maximize the lower bound w.r.t. these parameters, the standard Monte Carlo gradient estimator is known to exhibit extremely high variance, making appropriate gradient updates infeasible. As a solution Kingma & Welling (2013) propose to parametrize $q_\phi(z|x)$ by a differentiable function $g_\phi(\epsilon, x)$ where $\epsilon$ is an independent noise variable $\epsilon \sim p(\epsilon)$. In the case of $z \sim \mathcal{N}(0, I)$, $g$ is defined such that $g_\phi(\epsilon, x) = \mu + \sigma \odot \epsilon$ where $\odot$ denotes the element-wise product, and $\mu$ and $\sigma$ are the output of a multilayer perceptron parametrized by $\phi$. That is they propose to maximize

$$\widetilde{\mathcal{L}}(\theta, \phi; x_i) = -D_{KL}(q_\phi(z|x_i)||p(z)) + \sum_{i=1}^{L} \log p_\theta(x_i|\tilde{z}_i) \tag{3}$$

where $\tilde{z}_i = g_\phi(\epsilon, x_i)$.

Let $F$ be some multilayer perceptron parametrized by $\theta$. We would like to optimize the non-variational lower bound 1 w.r.t. $\theta$. In the case where $F$ is an auto-encoder, we desire the distribution $z \sim \mathcal{N}(0, I)$ to be from a lower dimensional space than $X$. It follows that $F$ must output the dimension of $X$ in one call, and the dimension of $z$ embedded in the dimension of $X$ in the second call. Define the projection $\pi_z(\hat{z}) = P\hat{z}$ where $P$ is a square matrix and $P_{i,j} = 1$ if $i = j \leq \dim(z)$ and 0 otherwise. We now combine our projection and the reparametrization trick from Kingma & Welling (2013). Let $p_\theta(z|x)$ in 1 be computed by the function $\tilde{z} = g_\theta(\epsilon, x) = \pi_z(\mu + \sigma \odot \epsilon)$ where $\mu$ and $\sigma$ are the outputs from $F_\theta$ and $\epsilon$ is sampled from an isotropic Gaussian. For the calculation of the KL divergence, $z$ may be embedded in the $X$ dimension or $\tilde{z}$ may have its extra zeros squeezed. Finally we may define the function

$$\widetilde{\mathcal{L}}(\theta; x_i) = -D_{KL}(p_\theta(z|x_i)||p(z)) + \sum_{i=1}^{L} \log p_\theta(x_i|\tilde{z}_i) \tag{4}$$

which may be differentiated and maximized with respect to $\theta$. Optimzing this lower bound on the marginal log-likelihood is equivalent to training a single network VAE. The effect of different projection (latent space) dimensions is explored in the experiments.

**Algorithm 1** Single-VAE gradient update

1: **function** UPDATEVAE($\theta^t, \alpha$)
2: Sample: $\{x_1, ..., x_M\}$ with $x_i \sim X$, $\{z_1, ..., z_M\}$ with $z_i \sim p(z)$
3: Update: $\theta^{t+1} = \theta^t + \alpha \nabla_\theta \tilde{\mathcal{L}}(\theta; x_i)$
4: **end function**

### 2.3 RECURSIVE GENERATIVE ADVERSARIAL NETWORK

Introduced by Goodfellow et al. (2014), the GAN plays the minimax game for the value function $V$ .

$$\min_G \max_D V(G, D) := E_{x\sim p(x)}[log(D(x))] + E_{z\sim p(z)}[log(1 - D(G(z)))] \tag{5}$$

where $x$ and $z'$ are sampled from their respective distributions. We let $G$ and $D$ be models parametrized by some $\theta'$ and $\phi$ respectively, such that $V$ may be differentiated and optimized w.r.t. these parameters. Define a new function $F$, parametrized by $\theta$, such that

$$F(\cdot) = \begin{cases} G(z) & z \sim \mathcal{N}(0, I) \\ D(x) & x \sim X \end{cases} \tag{6}$$

Note that by letting $z$ have the same number of dimensions as $X$, we may avoid the definition of a projection map. Let $F^*$ be an alias for $F$, with both functions parametrized by $\theta$. By substituting $F$ into 5, we obtain the new value function

$$\min_{F^*} \max_F V(F^*, F) := E_{x\sim p(x)}[log(F(x))] + E_{z\sim p(z)}[log(1 - F(F^*(z)))] \tag{7}$$

This formulation induces a cyclic computational graph as seen in 1. In practice Goodfellow et al. (2014) recommend maximizing $\log(D(G(z)))$ for the generator, as opposed to minimizing $\log(1 - D(G(z)))$ as it provides more robust gradients during training. By reformulating and simplifying, we may define an alternative function

$$V(F) := E_{x\sim p(x)}[log(F(x))] + E_{z\sim p(z)}[log(1 - F(\bar{F}(z)))] + E_{z\sim p(z)}[log(\bar{F}(F(z)))] \tag{8}$$

where $\bar{F}$ denotes $F$ with fixed parametrization $\bar{\theta}$. $V(F)$ is then maximized with respect to $\theta$.

**Algorithm 2** Single-GAN gradient update

1: **function** UPDATEGAN($\theta^t, \alpha$)
2: Sample: $\{x_1, ..., x_M\}$ with $x_i \sim X$, and $\{F(z_1), ..., F(z_M)\}$ with $z_i \sim p_\theta(z)$
3: Update: $\theta^{t+1} = \theta^t + \alpha \nabla_\theta V(F_{\theta^t})$
4: **end function**

**Theorem 1** *The optimal $F(z)$ has distribution $p_{true}(x)$*

Define $F$ as in 6 and $\bar{\theta}$ as fixed $\theta$ for optimization. We define $V$ as in 7. Observe that we may use the change of variables as shown by Goodfellow et al. (2014)

$$V(F, F) = \int_x p_{true}(x) log(F(x)) dx + \int_z p_z(z) log(1 - F(p_{\bar{\theta}})) dz$$

$$V(F, F) = \int_x p_{true}(x) log(F(x)) dx + p_{\bar{\theta}}(x) log(1 - F(x)) dx$$

It follows that the supremum is achieved at $\frac{p_{true}(x)}{p_{true}(x)+p_\theta(x)}$ for $F(x)$, the discriminator. Thus it can be shown that the optimal generator $F(z)$ minimizes the Jensen Shannon Divergence between $p_{true}(x)$ and $p_\theta(x)$.

Like Goodfellow et al. (2014) we do not present results for convergence of $V$ in the case when the models $G$, $D$ or $F$ represent multilayer perceptrons.

## 3 RELATED WORK

Numerous methods have been offered for the estimation of a generative function using only a single parametrized model. The Expectation Maximization (EM) algorithm proposes to estimate the maximum likelihood or maximum a posteriori for a parameter space by iteratively estimating the

log-likelihood and updating the distribution to improve this expectation. The EM algorithm relies on mean-field equations and has been shown to be equivalent to iterative variational methods over two parameter spaces (Dempster et al., 1977; Neal & Hinton, 1998). Another example, Noise Contrastive Estimation iteratively updates a model to improve upon its discrepancy with a data distribution by using the model itself as a discriminative function. The approach relies on analytically evaluating the corresponding probability densities, making it infeasible for complex models such as the MLP (Gutmann & Hyvärinen, 2010). More recently, Generative Moment Matching Networks (GMMN) have been proposed, which train by minimizing the maximum mean discrepancy (MMD) between an MLP output and a data distribution. MMD is a non-parametric two-sample measure of the distance between all moments of the given distributions, made tractable through the kernel trick. GMMN has not been shown to empirically match other models such as the GAN (Li et al., 2015).

Another key example, the Multi-Prediction Deep Boltzmann Machine from Goodfellow et al. (2013) proposes to train a Deep Boltzman Machine (DBM) as a single probabilitistic model. Typically the DBM is optimized in conjunction with a seperate model (often a multilayer perceptron) and requires greedy layer-wise pre-training of the Boltzmann Machine (Salakhutdinov & Hinton, 2009). Given some set of variables, the Multi-Prediction DBM takes as input a subset of the variables and is trained to predict the remaining unseen subset. This process allows a single training step over a single parameter space. Goodfellow et. al. argue that a single model for all inference tasks is ideal compared to multiple disjoint models. One may note this proposal's similarity to ours; both methods use a multi-prediction scheme for inferring the complements of a given variable set, and both methods may be interpreted as recurrent models.

Finally, we note the induced recursive interpretation of the single-network models, as shown in 1. Introduced by Pollack in 1990, recursive neural networks are a generalization of Recurrent Nets (RNNs). Traditionally recursive networks structure inputs in a tree, as opposed to an RNN list, in order to generalize to variant input-vector length and to exploit hierarchical structure in certain data (Pollack, 1990). The recursive network calls one multilayer perceptron recursively over nodes of the tree, encoding implicit bias in the computational graph. This fact has hindered recursive networks in the past due to indeterminate or restrictive graph structures (Chinea, 2009; dag). We also recognize the universal function approximation theorem for neural networks, which guarantees their ability to approximate continuous functions over compact euclidean sets (Cybenko, 1989), and that recurrent neural networks have been shown to be Turing universal under certain assumptions (Siegelmann & Sontag, 1995; Kilian & Siegelmann, 1996).

## 4 EXPERIMENTS

Results compare the single-network methods to corresponding two-network methods. Models presented are trained on MNIST (LeCun et al., 2010), Fashion MNIST (Xiao et al., 2017), and SVHN (Netzer et al., 2011) datasets. Code is written in Tensorflow. Overall the single-network VAE exhibits comparable performance to the two-network baseline, though it converges more slowly. The single-network GAN presents high quality samples, and seems to automatically regularize its hidden activations, which likely aids in the quality of samples.

### 4.1 SINGLE NETWORK VARIATIONAL AUTO-ENCODER

We compare a single network trained to minimize the non-variational lower bound 4 to a two-network model trained to minimize the variational lower bound 3. Both models use a learning rate of 0.001 and momentum with the same hyperparameters. In all tests the single-network has hidden-layers of size 800 each and a projection matrix as described before, which determines the number of output neurons to be used for data encoding. We give the baseline the closest possible number of trainable parameters, always favoring it over the single-network if a tie is not possible. For example, for 2D MNIST the baseline uses two networks each with hidden-layers of size 657, giving it about 4,000 more trainable parameters than the single-network. Both models use sigmoid and relu activations. The output activation for the single-network GAN is a sigmoid for the decoder, and is toggled to a linear activation in the encoder call. Similarly the baseline uses a sigmoid for the decoder output and a linear activation for the encoder output.

The corresponding two dimensional manifolds are also shown below.

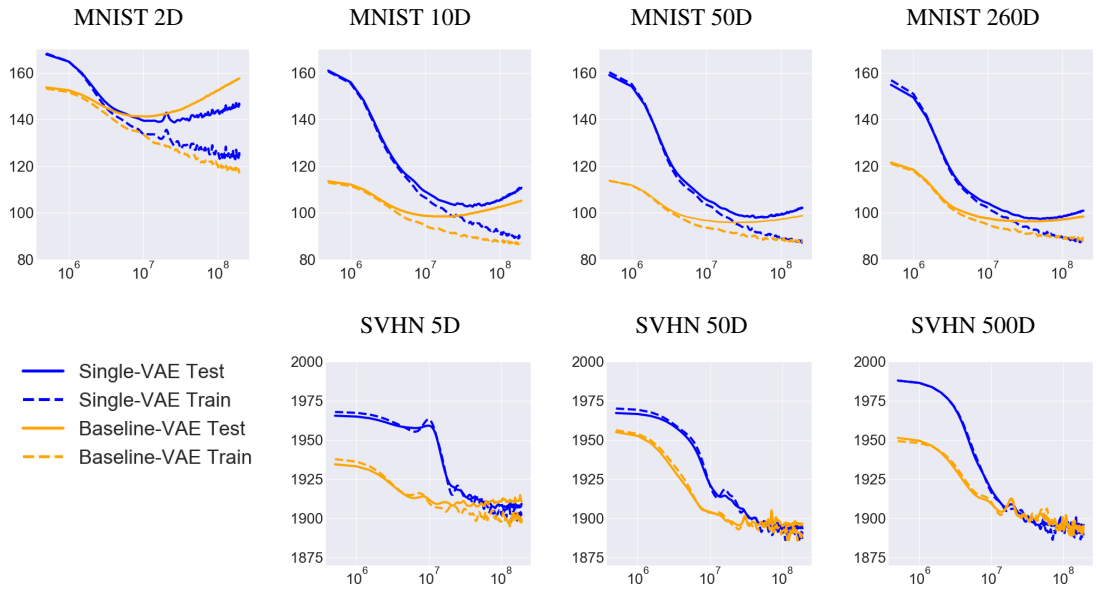


Figure 2: The correpsonding negative lower bound $-\widetilde{\mathcal{L}}$ over total number of samples seen. The single-network and baseline exhibit similar performance across samples, though the single-network is always slower to converge.

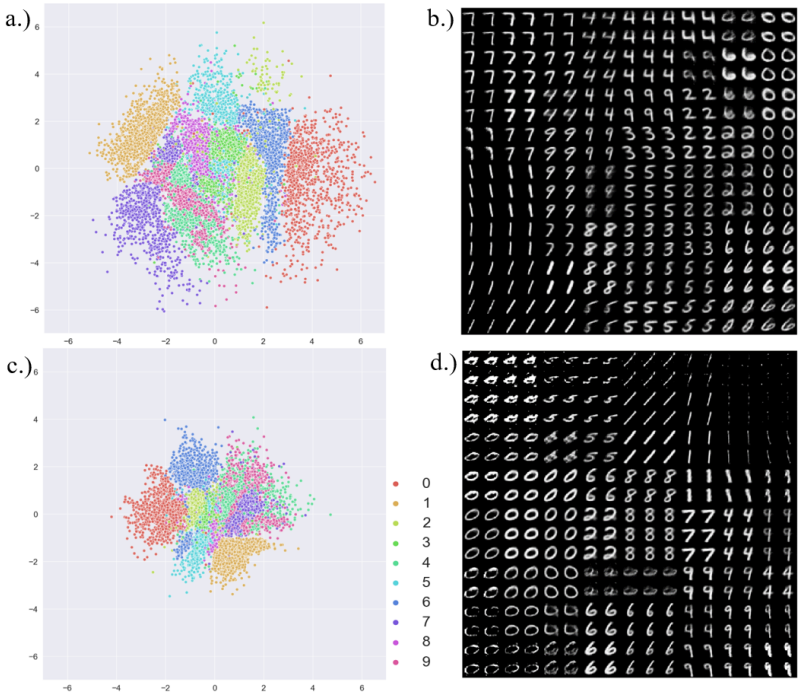


Figure 3: The top row (a & b) is generated by the single-network VAE, the bottom row (c & d) from the baseline two-network VAE. The first column (a & c) is the plot of the 2D encoded space $\hat{z}$ from the full MNIST test data. Y labels are depicted. The second column (b & d) is generated by a grid-sample of the decoders.

### 4.2 THE SINGLE NETWORK GAN

The following images are sampled from the single-network GAN and baseline GAN. For Fashion MNIST and MNIST the single-network GAN uses two hidden-layers of size 512 each. The baseline GAN uses two networks each with two hidden-layers of size 330. As before, this setup favors the baseline slightly, with about 1000 more trainable parameters. For SVHN the single-net has two hidden-layers of size 800. The baseline has layers of size 423, favoring the baseline by about 4000 trainable parameters. We fixed the parameter number then fine tuned each model, including the use of momentum and batch normalization. All models add a small $L1$ loss to the discriminator outputs, which is multiplied by a fine tuned constant. Momentum was used for all baseline models, and only used on SVHN for the single-network. Batch norm is used for all baseline models, and for no single-network models (though we surprisingly found it effective for the single-net).

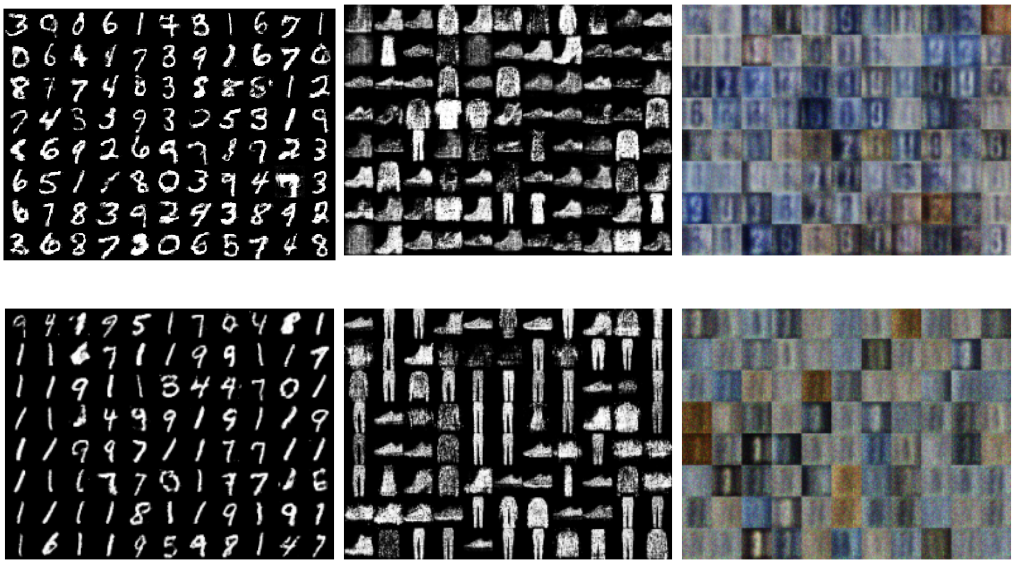

Figure 4: GAN comparison. Top row: samples from the single-network GAN. Bottom row: samples from the comparable two-network baseline.

We use kernel density estimation to compare generative models. 10,000 photos are sampled from each model and three-fold cross-validation is used to estimate the Parzen window. The final model is used to calculate the average log-likelihood of the 10,000 sample MNIST test set. We note that kernel desnity estimation is quite unreliable and has been shown to exhibit high variance between different trials (Theis et al., 2015). Generally one expects that the results correlate to some goodness-of-fit to the true distribution.

| Model | KDE $LL$ (nats) | $\pm$SEM |
|---|---|---|
| Baseline VAE | 410 | 7.6 |
| Single-Network VAE | 416 | 7.6 |
| Baseline GAN | 73 | 11.9 |
| Single-Network GAN | 399 | 2.9 |
| GAN (Goodfellow et al., 2014) | 225 | 2 |
| GMMD (Li et al., 2015) | 147 | 2 |
| Deep GSN (Bengio et al., 2014) | 214 | 1.1 |

Table 1: Log-likelihood of full MNIST test set based on Parzen window estimates. The VAE models reported use a 50 dimensional latent space. Note: Deep Generative Stochastic Networks (DSN) uses an estimated Markov chain to train a single network (Bengio et al., 2014).

We find that the single-network GAN outperforms the baseline log-likelihood on this test.

In an attempt to understand the effect of the single parameter formulation, we analyze the activations of the hidden-layers across training. In the following chart, both models are trained with stochastic gradient descent, a small fine-tuned $L1$ regularization as before, and no batch normalization. We find that the single-network GAN has a regularizing effect over its activations compared to the baseline.

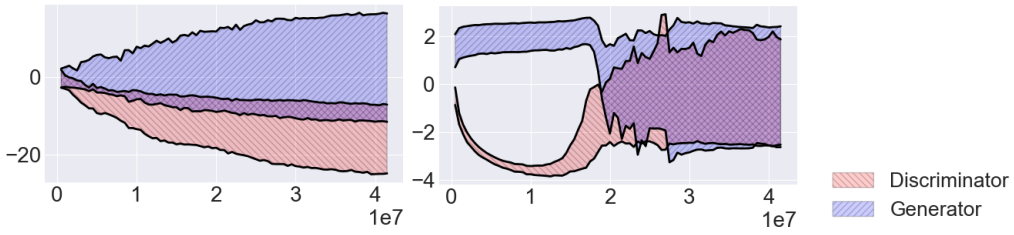


Figure 5: Distribution of 2nd hidden-layer neuron activations across total MNIST examples seen. The 15th to 85th percentile activation ranges are shown. Left: single-network GAN. Right: baseline GAN.

The same percentile-activation test has been conducted on the baseline GAN with momentum and batch normalization. We found in this case the activations to be more stable than with SGD, but not as monotonic as the single-network GAN.

## CONCLUSION

We have proposed to train a multilayer perceptron simultaneously as an encoder and a decoder to create a high quality generative model. Our approach is derived from a reformulation of the variational lower bound and extends to Generative Adversarial Nets. Both the single-network VAE and GAN have been shown to be comparable to their baseline counterparts. Most importantly, we have shown that the explicit parametric distinction between encoders and decoders may be automated by backpropgation over a single space.

We leave as open problems for future work to

- Apply new or existing computer-vision techniques to the models presented.
- Apply other recurrent models and optimizations in order to exploit the cyclic computational graph induced by our methods.
- Provide a formal result on the Minimum Description Length (MDL) of the models presented. The MDL principle is the claim that the best hypothesis is the shortest one with the best compression of the data (Rissanen, 1978).
- Use the single-network formulations to better understand the parameter space $\theta$, especially w.r.t. the relationship between encoding and decoding network pathways.
- Explore potential new directions for the unification of generative models.

## REFERENCES

Yoshua Bengio, Eric Laufer, Guillaume Alain, and Jason Yosinski. Deep generative stochastic networks trainable by backprop. In *International Conference on Machine Learning*, pp. 226–234, 2014.

Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134 (1-2):57–83, 2002.

Alejandro Chinea. Understanding the principles of recursive neural networks: a generative approach to tackle model complexity. In *International Conference on Artificial Neural Networks*, pp. 952–963. Springer, 2009.

G. Cybenko. Approximation by superpositions of a sigmoidal function, 1989.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–22, 1977.

Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*, 2015.

Stephen I Gallant and Stephen I Gallant. *Neural network learning and expert systems*. MIT press, 1993.

Ian Goodfellow, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Multi-prediction deep boltzmann machines. In *Advances in Neural Information Processing Systems*, pp. 548–556, 2013.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric P Xing. On unifying deep generative models. *arXiv preprint arXiv:1706.00550*, 2017.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Joe Kilian and Hava T Siegelmann. The dynamic universality of sigmoidal neural networks. *Information and computation*, 128(1):48–56, 1996.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pp. 1558–1566. PMLR, 2016.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pp. 1718–1727, 2015.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pp. 355–368. Springer, 1998.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, Nov 2010. ISSN 1557-9654. doi: 10.1109/tit.2010.2068870. URL `http://dx.doi.org/10.1109/TIT.2010.2068870`.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pp. 271–279, 2016.

Jordan B Pollack. Recursive distributed representations. *Artificial Intelligence*, 46(1-2):77–105, 1990.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and Shakir Mohamed. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017.

Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pp. 448–455, 2009.

Hava T Siegelmann and Eduardo D Sontag. On the computational power of neural nets. *Journal of computer and system sciences*, 50(1):132–150, 1995.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. It takes (only) two: Adversarial generator-encoder networks. 2017.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL `http://arxiv.org/abs/1708.07747`.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pp. 7354–7363. PMLR, 2019.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Towards deeper understanding of variational autoencoding models. *arXiv preprint arXiv:1702.08658*, 2017.