f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization

Sebastian Nowozin, Botond Cseke, Ryota Tomioka

Light Background:

Traditionally Generative Neural Samplers are probabilistic models that sample using a feed forward network (random input vector -> probability distribution based of network weights)

Probabilistic generative models are great for samples and derivations and derivatives but fail in cases where generative adversarial models succeed. Namely, computing likelihoods and marginalization.

The paper expresses that GANS are essentially a more sophisticated version of generative models due to the type of neural network they use. Generative models can imitate this behavior through the introduction of the divergence functions.

Paper Concepts:

Probabilistic models that sample using a feed forward network (random input vector -> probability distribution based of network weights), GANs and other Generative Neural Samplers are able to do this one pass.

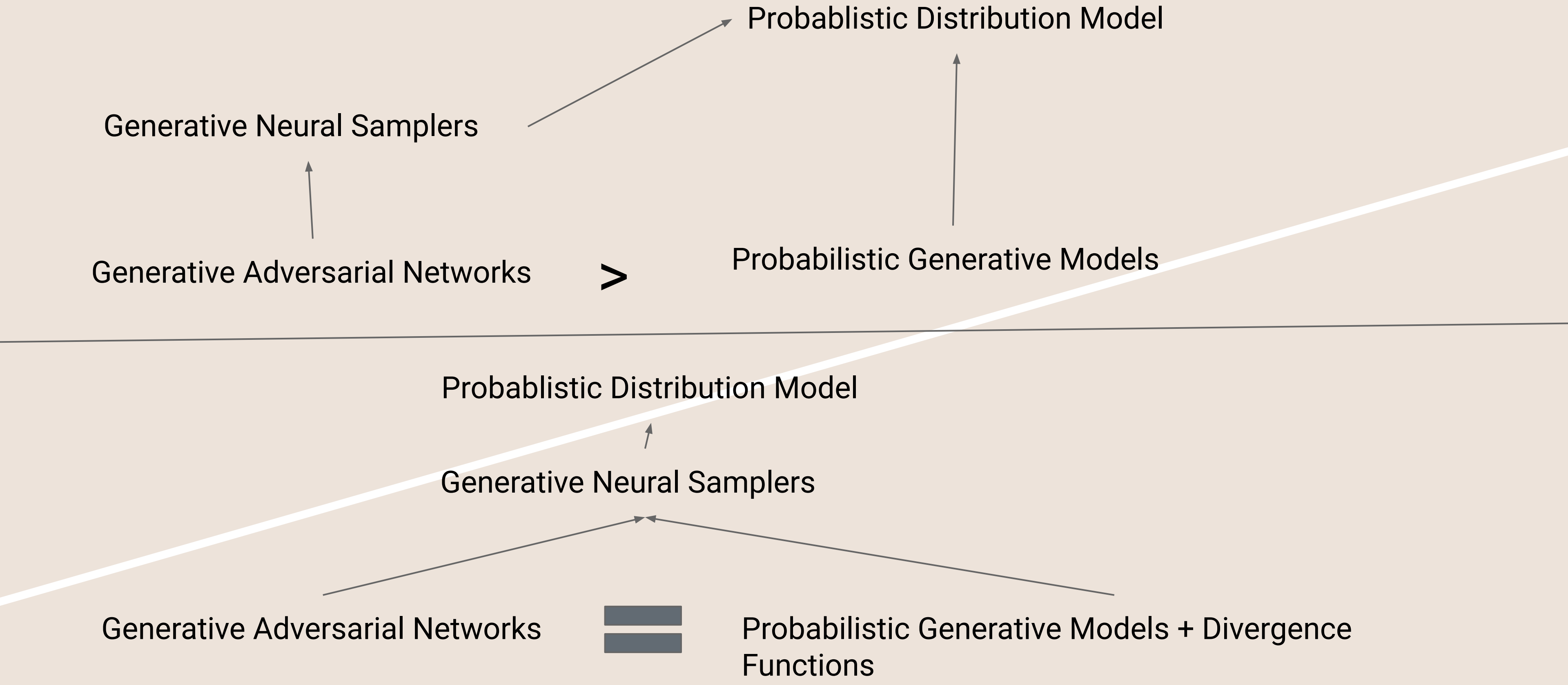
How can general probabilistic models improve their performance and application? Through imitating GANs and converting them into a Generative Neural Sampler type Model.

The paper expresses that GNS models are able to broaden and improve the performance and additional applications of GANs through the usage of divergence functions* for training GNS models.

*Divergence Functions are functions used to measure the difference between distributions.

The paper covers the choices of f-divergence, training complexity, and quality of the final result. This paper is different from the usual paper with one singular model. Rather this paper suggests a class of changes.

Visual Aid


Probablistic Distribution Model
Generative Neural Samplers
Generative Adversarial Networks
>
Probabilistic Generative Models
Probablistic Distribution Model
Generative Neural Samplers
Generative Adversarial Networks
=
Probabilistic Generative Models + Divergence Functions

Background- Probabilistic Generative Models

Recall probabilistic generative models describe a probability distribution over a given domain X.

Given Q, a generative model, attempting to describe class Q, you can perform one of the following tasks:

- Sampling
- Estimation
- Point-wise Likelihood Evaluation

- Sampling: inspecting samples or calculating a function on a set of samples -> able to get ideas about the distribution / resolve decisions

- Estimation: Given samples {x1, x2, . . . , xn} from an unknown distribution, Q is able to describe the distribution. *true distributions.

- Point-wise Likelihood Eval: Given sample x, able to evaluate likelihood

Background - GANs

GANs follow the mathematical model: Jensen-Shannon divergence, and specifically minimizing it.

$$D_{\mathrm{JS}}(P\|Q) = \frac{1}{2}D_{\mathrm{KL}}(P\|\frac{1}{2}(P+Q)) + \frac{1}{2}D_{\mathrm{KL}}(Q\|\frac{1}{2}(P+Q))$$

- D_{KL} = Kullback-Liebler Divergence, essentially a second discriminator
- These two divergence factors work together to find the overall distribution given enough samples

They are extremely efficient because when creating a single sample it requires one pass through its feed forward neural network.

These GAN properties are important to know because they are the properties that will be generalized and applied to the GNS through the divergence function.

Model - Generalizing GANs with F-Divergence

1. Derive the GAN training objectives for all f-divergences and provide as example additional divergence functions, including the Kullback-Leibler and Pearson divergences.

2. Simplify the saddle-point optimization procedure in new derived equation.

3. Provide experimental insight into which divergence function is suitable for estimating generative neural samplers for use case.

Take Nguyen DK Equation for general purpose divergence functions as a base for point (3).

$$D_K(\mathbb{P}, \mathbb{Q}) \quad := \quad \int p_0 \log \frac{p_0}{q_0} \, d\mu.$$

Model - Generalizing GANs with F-Divergence (2)

Taking the Nguyen and incorporating the dual divergence functions from GANS we come up with:

$$D_K(\mathbb{P}, \mathbb{Q}) \ := \ \int p_0 \log \frac{p_0}{q_0} \, d\mu. \longrightarrow D_f(P\|Q) = \int_{\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x,$$

There are some special properties for the custom function that we generated (on the right):

- Given two distributions the function aims to measure the difference between the two distributions and that is our F-Divergence.
- f: R_+ -> R, is a convex lower semicontinuous function where f(1) = 0. This is specific for the function above and would be different for different functions. (f(1) = 0, is important so D_f(P||P) = 0

Model - Variational Estimation of f-Divergences

The Nguyen derives a general variational method to estimate f-divergences given samples from P & Q. The equation estimates a function for P and Q as opposed to the desired result of estimating model parameters.

The paper introduces VDM (variational divergence minimization):

Taking the derived equation earlier with the properties:

f: R_+ -> R, is a convex lower semicontinuous function where f(1) = 0. This is specific for the function above and would be different for different functions.

- We can derive a new function: Every convex, lower-semicontinuous function f has a convex conjugate function f $*$, also known as the Fenchel conjugate

$$f^*(t) = \sup_{u \in \mathrm{dom}_f} \{ut - f(u)\}$$

Model – Side by Side Comparison of Functions

The Nguyen derives a general variational method to estimate f-divergences given samples from P & Q. The equation estimates a function for P and Q as opposed to the desired result of estimating model parameters.

The paper introduces VDM (variational divergence minimization):

Taking the derived equation earlier with the properties:

f: R_+ -> R, is a convex lower semicontinuous function where f(1) = 0. This is specific for the function above and would be different for different functions.

- We can derive a new function: Every convex, lower-semicontinuous function f has a convex conjugate function f $*$, also known as the Fenchel conjugate. This in turn is again convex and lower-semicontinuous and the pair (f, f $*$) is dual to another in the sense that f $**$ = f. Thus, f = f(u)

$$f^*(t) = \sup_{u \in \mathrm{dom}_f} \{ut - f(u)\}$$

$$f(u) = \sup\nolimits_{t \in \mathrm{dom}_{f^*}} \{tu - f^*(t)\}$$

Full Derivation from Start to Finish

$$
\begin{aligned}
D_f(P\|Q) &= \int_{\mathcal{X}} q(x) \sup_{t \in \mathrm{dom}_{f^*}} \left\{ t \frac{p(x)}{q(x)} - f^*(t) \right\} \mathrm{d}x \\
&\geq \sup_{T \in \mathcal{T}} \left(\int_{\mathcal{X}} p(x)\, T(x)\, \mathrm{d}x - \int_{\mathcal{X}} q(x)\, f^*(T(x))\, \mathrm{d}x \right) \\
&= \sup_{T \in \mathcal{T}} \left(\mathbb{E}_{x \sim P}\left[T(x)\right] - \mathbb{E}_{x \sim Q}\left[f^*(T(x))\right] \right),
\end{aligned}
$$

$$
T^*(x) = f'\left(\frac{p(x)}{q(x)} \right)
$$

Variational Divergence Minimization (VDM)

VDM application to the function derived from Nguyen: $D_f(P\|Q) = \int_{\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x,$

- All the previous work was necessary to get the variational lower bound on the f-divergence in order to estimate the generative model Q given distribution P.

- Similar to a GAN, we use two neural networks. Q is the generative model and T is the variational function.
  - Q takes a random vector and outputs a sample
  - T takes a sample and returns a scalar.

- Q is generated through the following f-GAN objective function that finds a saddle point to get Q.
  - Theta is a vector pertaining to Q. w is the vector pertaining to T.

$$F(\theta, \omega) = \mathbb{E}_{x\sim P}\left[T_\omega(x)\right] - \mathbb{E}_{x\sim Q_\theta}\left[f^*(T_\omega(x))\right].$$

VDM - Finding Optimal F

$$F(\theta, \omega) = \mathbb{E}_{x \sim P}\left[T_\omega(x)\right] - \mathbb{E}_{x \sim Q_\theta}\left[f^*(T_\omega(x))\right].$$

The expected values are approximated through minibatches.

- The first term is sampled from the training set without replacement.
- The second term is taken from current Q_{Theta}.

Running Single-Step Gradient we are able to find a saddle point where Theta is strongly convex and W is strongly concave. Essentially maximising Theta and minimizing W.

There can be many satisfying saddle points and so it comes down to setting a custom metric as to what you want to accept.

$$\nabla_\theta F(\theta^*, \omega^*) = 0, \quad \nabla_\omega F(\theta^*, \omega^*) = 0, \quad \nabla_\theta^2 F(\theta, \omega) \succeq \delta I, \quad \nabla_\omega^2 F(\theta, \omega) \preceq -\delta I.$$

VDM vs. GAN Structure

F-GAN Model

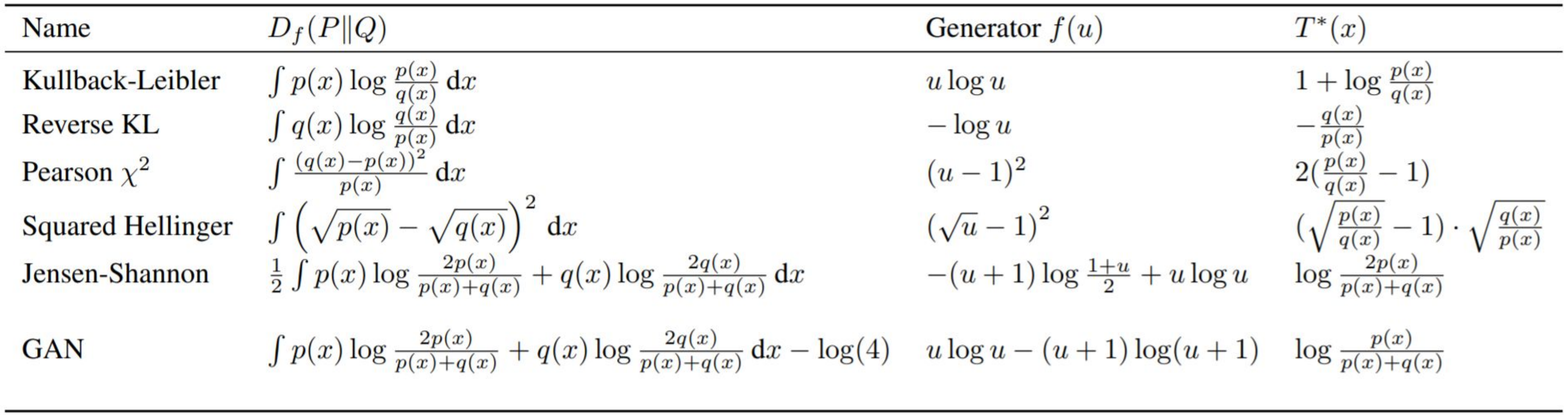
$$F(\theta, \omega) = \mathbb{E}_{x \sim P}\left[T_\omega(x)\right] - \mathbb{E}_{x \sim Q_\theta}\left[f^*(T_\omega(x))\right]$$

GAN Model

$$F(\theta, \omega) = \mathbb{E}_{x \sim P}\left[\log D_\omega(x)\right] + \mathbb{E}_{x \sim Q_\theta}\left[\log(1 - D_\omega(x))\right]$$

Key Differences: The D function for GANs is a discriminat function that is more complex than the T lower bound designator function. Also in terms of training speed, maximizing the second term in the F-Gan model is easier than minimizing the second term in the GAN model.

Side by Side Comparisons of f-Divergences



Name	$D_f(P\|Q)$	Generator $f(u)$	$T^*(x)$
Kullback-Leibler	$\int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x$	$u \log u$	$1 + \log \frac{p(x)}{q(x)}$
Reverse KL	$\int q(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$	$-\log u$	$-\frac{q(x)}{p(x)}$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)} \, \mathrm{d}x$	$(u-1)^2$	$2(\frac{p(x)}{q(x)} - 1)$
Squared Hellinger	$\int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \, \mathrm{d}x$	$(\sqrt{u}-1)^2$	$(\sqrt{\frac{p(x)}{q(x)}} - 1) \cdot \sqrt{\frac{q(x)}{p(x)}}$
Jensen-Shannon	$\frac{1}{2} \int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \, \mathrm{d}x$	$-(u+1) \log \frac{1+u}{2} + u \log u$	$\log \frac{2p(x)}{p(x)+q(x)}$
GAN	$\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \, \mathrm{d}x - \log(4)$	$u \log u - (u+1) \log(u+1)$	$\log \frac{p(x)}{p(x)+q(x)}$

Applications - Different f-Divergences

Experiment

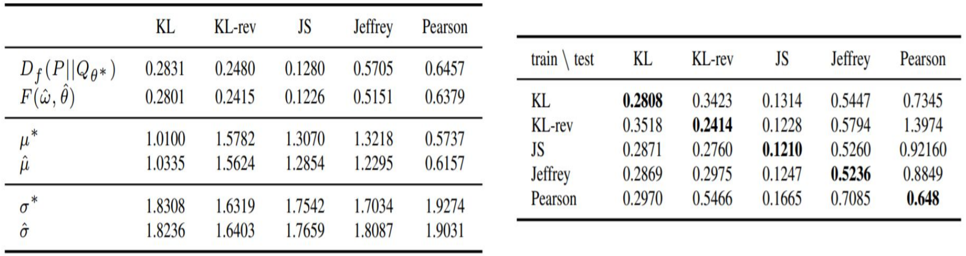
Experiment with differnet divergence functions:

Have models learn a Gaussian distribution and find the most optimal parameters to describe the Gaussian distribution.

When Q, our model, is turned into a linear function that receives: $z \sim \mathcal{N}(0, 1)$. Q outputs: $G_\theta(z) = \mu + \sigma z$

Theta in this case is a combination of (mu and sigma).

Results



	KL	KL-rev	JS	Jeffrey	Pearson
$D_f(P\|\|Q_{\theta^*})$	0.2831	0.2480	0.1280	0.5705	0.6457
$F(\hat{\omega}, \hat{\theta})$	0.2801	0.2415	0.1226	0.5151	0.6379
μ^*	1.0100	1.5782	1.3070	1.3218	0.5737
$\hat{\mu}$	1.0335	1.5624	1.2854	1.2295	0.6157
σ^*	1.8308	1.6319	1.7542	1.7034	1.9274
$\hat{\sigma}$	1.8236	1.6403	1.7659	1.8087	1.9031

train \ test	KL	KL-rev	JS	Jeffrey	Pearson
KL	**0.2808**	0.3423	0.1314	0.5447	0.7345
KL-rev	0.3518	**0.2414**	0.1228	0.5794	1.3974
JS	0.2871	0.2760	**0.1210**	0.5260	0.92160
Jeffrey	0.2869	0.2975	0.1247	**0.5236**	0.8849
Pearson	0.2970	0.5466	0.1665	0.7085	**0.648**

Table 3: Gaussian approximation of a mixture of Gaussians. Left: optimal objectives, and the learned mean and the standard deviation: $\hat{\theta} = (\hat{\mu}, \hat{\sigma})$ (learned) and $\theta^* = (\mu^*, \sigma^*)$ (best fit). Right: objective values to the true distribution for each trained model. For each divergence, the lowest objective function value is achieved by the model that was trained for this divergence.

Applications & Results – MNIST Digits

Model Design and Training

Two components to the VDM: Generative Model and Variational Function.

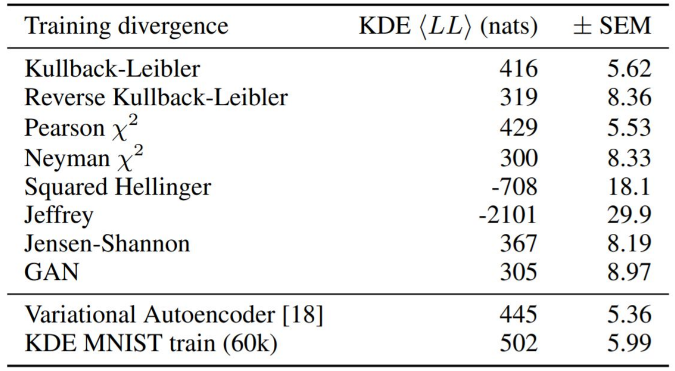
Very simple generative model with batch normalization, ReLU, and sigmoid activation functions. The variational function is also simple with a few linear layers and exponential in between.

The VDM is trained as described by sampling without replacement from batches of size 4096. The model is then trained on that information for one hour.

Compared against variational autoencoders with 20 latent dimensions.

Results

The results are calculated using kernel density estimation. The goal is to measure the average log-likelihood of the performance.



Training divergence	KDE $\langle LL \rangle$ (nats)	$\pm$ SEM
Kullback-Leibler	416	5.62
Reverse Kullback-Leibler	319	8.36
Pearson χ^2	429	5.53
Neyman χ^2	300	8.33
Squared Hellinger	-708	18.1
Jeffrey	-2101	29.9
Jensen-Shannon	367	8.19
GAN	305	8.97
Variational Autoencoder [18]	445	5.36
KDE MNIST train (60k)	502	5.99

Summary of Concepts & Conclusion

The paper was focused more on the structure of GANs and figuring out ways to improve upon them.

The paper came up with the concept of f-divergence, hence the name of the paper f-GAN.

Utilizing the proper divergence function leads to the optimal model and better performance.

The paper contributed the VDM model and the general idea behind the usage of f-Divergence with an application towards GNS / GANs.

Thus, GANs can be generalized to an arbitrary divergence function and the algorithm behind GANs is simplified as well.

(a) GAN (b) KL (c) Squared Hellinger

References

- 2016 (NeurIPS): S. Nowozin, B. Cseke, R. Tomioka.f-gan: Training generative neural samplers using variational divergence minimization. NeurIPS, 2016.