HOW (NOT) TO TRAIN YOUR GENERATIVE MODEL: SCHEDULED SAMPLING, LIKELIHOOD, ADVERSARY?

Ferenc Huszár
Balderton Capital LLP, London, UK
ferenc.huszar@gmail.com

ABSTRACT

Modern applications and progress in deep learning research have created renewed interest for generative models of text and of images. However, even today it is unclear what objective functions one should use to train and evaluate these models. In this paper we present two contributions.

Firstly, we present a critique of scheduled sampling, a state-of-the-art training method that contributed to the winning entry to the MSCOCO image captioning benchmark in 2015. Here we show that despite this impressive empirical performance, the objective function underlying scheduled sampling is improper and leads to an inconsistent learning algorithm.

Secondly, we revisit the problems that scheduled sampling was meant to address, and present an alternative interpretation. We argue that maximum likelihood is an inappropriate training objective when the end-goal is to generate natural-looking samples. We go on to derive an ideal objective function to use in this situation instead. We introduce a generalisation of adversarial training, and show how such method can interpolate between maximum likelihood training and our ideal training objective. To our knowledge this is the first theoretical analysis that explains why adversarial training tends to produce samples with higher perceived quality.

调度采样不是好的目标函数，导致不稳定算法

极大似然估计：如果目标是生成natural-looking样本，不是一个正确的目标函数。

对抗的目标函数很好

1 INTRODUCTION

Building sophisticated generative models that produce realistic-looking images or text is an important current frontier of unsupervised learning. The renewed interest in generative models can be attributed to two factors. Firstly, thanks to the active investment in machine learning by internet companies, we now have several products and practical use-cases for generative models: texture generation(Han et al., 2008), speech synthesis (Ou & Zhang, 2012), image caption generation (Lin et al., 2014; Vinyals et al., 2014), machine translation (Sutskever et al., 2014), conversation and dialogue generation (Vinyals & Le, 2015; Sordoni et al., 2015). Secondly, recent success in generative models, particularly those based on deep representation learning, have raised hopes that our systems may one day reach the sophistication required in these practical use cases.

While noticable progress has been made in generative modelling, in many applications we are still far from generating fully realistic samples. One of the key open questions is what objective functions one should use to train and evaluate generative models (Theis et al., 2015). The model likelihood is often considered the most principled training objective and most research in the past decades has focussed on maximum lieklihood(ML) and approximations thereof Hinton et al. (2006); Hyvärinen (2006); Kingma & Welling (2013). Recently we have seen promising new training strategies such as those based on adversarial networks (Goodfellow et al., 2014; Denton et al., 2015) and kernel moment matching (Li et al., 2015; Dziugaite et al., 2015) which are not — at least on the surface — related to maximum likelihood. Most of this departure from ML was motivated by the fact that the exact likelihood is intractable in the most models. However, some authors have recently observed that even in models whose likelihood is tractable, ML training leads to undesired behaviour, and introduced new training procedures that deliberately differ from maximum likelihood. Here we will focus on scheduled sampling (Bengio et al., 2015) which is an example of this.

In this paper we attempt to clarify what objective functions might work well for the generative scenario and which ones should one avoid. In line with (Theis et al., 2015) and (Lacoste-Julien

et al., 2011), we believe that the objective function used for training should reflect the task we want to ultimately use the model for. In the context of this paper, we focus on generative models that are created with the sole purpose of generating realistic-looking samples from. This narrower definition extends to use-cases such as image captioning, texture generation, machine translation and dialogue systems, but excludes tasks such as unsupervised pre-training for supervised learning, semisupervised learning, data compression, denoising and many others.

This paper is organised around the following main contributions:

scheduled sampling is improper: In the first half of this paper we focus on autoregressive models for sequence generation. These models are interesting for us mainly because exact maximum likelihood training is tractable, even in relatively complex models such as stacked LSTMs (Bengio et al., 2015; Sutskever et al., 2014; Theis & Bethge, 2015). However, it has been observed that autoregressive generative models trained via ML have some undesired behaviour when they are used to generate samples. We revisit a recent attempt to remedy these problems: scheduled sampling. We reexpress the scheduled sampling training objective in terms of Kullback-Leibler divergences, and show that it is in fact an improper training objective. Therefore we recommend to use scheduled sampling with care.

KL-divergence as a model of perceptual loss: In the latter part of the paper we seek an alternative solution to the problem scheduled sampling was meant to address. We uncover a more fundamental problem that applies to all generative models: that the likelihood is not the right training objective when the goal is to generate realistic samples. Maximum likelihood can be thought of as minimising the Kullback-Leibler divergence $KL[P\|Q]$ between the real data distribution P and the probabilistic model Q. We present a model that suggests generative models should instead be trained to minimise $KL[Q\|P]$, the Kullback-Leibler divergence in the opposite direction. The differences between minimising $KL[P\|Q]$ and $KL[Q\|P]$ are well understood, and explain the observed undesirable behaviour in autoregressive sequence models.

generalised adversarial training: Unfortunately, $KL[Q\|P]$ is even harder to optimise than the likelihood, so it is unlikely to yield a viable training procedure. Instead, we suggest to minimise an information quantity which we call generalised Jensen-Shannon divergence. We show that this divergence can effectively interpolate between the behaviour of $KL[P\|Q]$ and $KL[P\|Q]$, thereby containing both maximum likelihood, and our ideal perceptual objective function as a special case. We also show that generalisations of the adversarial training procedure proposed in (Goodfellow et al., 2014) can be employed to approximately minimise this divergence function. Our analysis also provides a new theoretical explanation for the success of adversarial training in producing qualitatively superior samples.

2 Autoregressive Models for Sequence Generation

In this section we will focus on a particularly useful class of probabilistic models, which we call autoregressive generative models (see e. g. Theis et al., 2012; Larochelle & Murray, 2011; Bengio et al., 2015). An autoregressive probabilistic model explicitly defines the joint distribution over a sequence of symbols $x_{1:N}$ recursively as follows:

$$Q_{1:N}(x_{1:N}) = \prod_{n=1}^{N} Q_n(x_n|x_{1:n-1};\theta). \tag{1}$$

We note that technically the above equation holds for all joint distributions $Q_{1:N}$, here we further assume that each of the component distributions $Q_n(x_n|x_{1:n-1};\theta)$. are tractable and easy to compute. Autoregressive models are considered relatively easy to train, as the model likelihood is typically tractable. This allows us to train even complicated deep models such as stacked LSTMs in the coherent and well understood framework of maximum likelihood estimation (Theis et al., 2012; 2015).

3 THE SYMPTOMS

Despite the elegance of a closed-form maximum likelihood training, Bengio et al. (2015) have observed out that maximum likelihood training leads to undesirable behaviour when the models are used to generate samples from. In this section we review these *symptoms*, and throughout this paper we will explore different strategies aimed at explaining and

Typically, when training an AR model, one minimises the log predictive likelihood of the nth symbol in each training sentence conditioned on all previous symbols in the sequence that we collectively call the prefix. This can be thought of as a special case of maximum likelihood learning, as the joint likelihood over all symbols in a sequence factorises into these conditionals via the chain rule of probabilities.

When using the trained model to generate sample sequences, we generate each new sequence symbol-by-symbol in a recursive fashion: Assuming we already generated a prefix of n sybols, we feed that prefix into the conditional model, and ask it to output the predictive distribution for the $n+1$st character. The $n+1$st character is then sampled from this distribution and added to the prefix.

Crucially, at training time the RNN only sees prefixes from real training sequences. However, at generation time, it can generate a prefix that is never seen in the training data. Once an unlikely prefix is generated, the model typically has a hard time recovering from the mistake, and will start outputting a seemingly random string of symbols ending up with a sample that has poor perceptual quality and is very unlikely under the true sequence distribution P.

4 SYMPTOMATIC TREATMENT: SCHEDULED SAMPLING

了解这个就知道这里讲的是什么

In (Bengio et al., 2015), the authors stipulate that the cause of the observed poor behaviour is the disconnect between how the model is trained (it's always fed prefixes from real data) and how it's used (it's always fed synthetic prefixes generated by the model itself). To address this, the authors propose an alternative training strategy called scheduled sampling (SS). In scheduled sampling, the network is sometimes given its own synthetic data as prefix instead of a real prefix at training time. This, the authors argue, simulates the environment in which the model is used when generating samples from it.

More specifically, we turn each training sequence into modified training sequence in a recursive fashion using the following procedure:

- for the nth symbol we draw from a Bernoulli distribution with parameter ϵ to decide whether we keep the original symbol or use one generated by the model
- if we decided to replace the symbol, we use the current model RNN to output the predictive distribution of the next symbol given the current prefix, and sample from this predictive distribution
- we add to the training loss the log predictive probability of the real nth symbol, given the prefix (the prefix at this point may already contain generated characters)
- depending on the coinflip above, the original or simulated character is added to the prefix and we continue with the recursion

The method is called scheduled sampling to describe the way the hyperparameter ϵ is annealed during training from an initial value of $\epsilon = 1$ down to $\epsilon = 0$. Here, we would like to understand the limiting behaviour of this training procedure, whether and why it is an appropriate way to address the shortcomings of maximum likelihood training.

4.1 SCHEDULED SAMPLING FORMULATED AS KL DIVERGENCE MINIMISATION

To keep notation simple, let us consider the case of learning sequences of length 2, that is pairs of random symbols x_1 and x_2. Our aim is to formulate a closed form training objective that corresponds to scheduled sampling.

If x_1 is kept original - rather than replaced by a sample - the scheduled sampling objective in fact remains the same as maximum likelihood. We can understand maximum likelihood as minimising the following KL divergence[1] between the true data distribution P and our approximation Q:

$$D_{ML}[P\|Q] = KL[P\|Q] \tag{2}$$
$$= KL[P_{x_1}\|Q_{x_1}] + \mathbb{E}_{z\sim P_{x_1}} KL[P_{x_2|x_1=z}\|Q_{x_2|x_1=z}] \tag{3}$$

Here, P_{x_1} and Q_{x_1} denote marginal distributions of the first symbol x_1 under P and Q respectively, while $Q_{x_2|x_1=z}$ and $P_{x_2|x_1=z}$ denote the conditional distributions of the second symbol x_2 conditioned on the value of the first symbol x_1 being z.

The other case we need to consider is when x_1 is replaced by a sample from the model, in this case Q_{x_1}. The training objective can now be expressed as the following divergence:

$$D_{alternative}[P\|Q] = KL[P_{x_1}\|Q_{x_1}] + \mathbb{E}_{y\sim P_{x_1}}\mathbb{E}_{z\sim Q_{x_1}} KL[P_{x_2|x_1=y}\|Q_{x_2|x_1=z}] \tag{4}$$
$$= KL[P_{x_1}\|Q_{x_1}] + \mathbb{E}_{z\sim Q_{x_1}} KL[P_{x_2}\|Q_{x_2|x_1=z}] \tag{5}$$

Notice how in the second term the KL divergence is now measured from P_{x_2} rather than the conditional, this is because the real value of the first symbol is never shown to the model, when it is asked to predict the second symbol x_2.

In scheduled sampling, we choose randomly between the above two cases, so the full SS objective can be described as a convex combination of D_{ML} and $D_{alternative}$ above:

$$D_{SS}[P\|Q] = KL[P_{x_1}\|Q_{x_1}] + \epsilon\mathbb{E}_{z\sim P_{x_1}} KL[P_{x_2|x_1=z}\|Q_{x_2|x_1=z}] + (1-\epsilon)\mathbb{E}_{z\sim Q_{x_1}} KL[P_{x_2}\|Q_{x_2|x_1=z}] \tag{6}$$

It is worth noting at this point that this divergence is an idealised form of the scheduled sampling. In the actual algorithm, expectations over Q_{x_1} and $Bernoulli(\epsilon)$ would be implemented by sampling[2]. This divergence describes the method's limiting behaviour in the limit of infinite training data.

By rearranging terms we can further express the SS objective as the following KL divergence:

$$D_{SS}[P\|Q] = KL[P_{x_1}\|Q_{x_1}] + \mathbb{E}_{z\sim P_{x_1}} KL\left[\epsilon P_{x_1|x_1=z} + \frac{Q_{x_1}(z)}{Q_{x_1}(z)} P_{x_2} \middle\| Q_{x_2|x_1=z}\right] + C_{P,\epsilon} \tag{7}$$
$$= KL\left[P_{x_1}\left(\epsilon P_{x_1|x_1} + (1-\epsilon)\frac{Q_{x_1}P_{x_2}}{P_{x_1}}\right) \middle\| Q_{x_1,x_2}\right] + C_{P,\epsilon} \tag{8}$$

A very natural requirement for any divergence function used to assess goodness of fit in probabilistic models is that it is minimised when $Q = P$. In statistics, this property is referred to as strictly proper scoring rule estimation (Gneiting & Raftery, 2007). Working with strictly proper divergences guarantees consistency, i. e. that the training procedure can ultimately recover the true P, assuming the model class is flexible enough and enough training data is provided. What the above analysis shows us is that scheduled sampling is not a consistent estimation strategy. As $\epsilon \to 0$, the divergence is globally minimised at the factorised distribution $Q = P_{x_1}P_{x_2}$, rather than at the correct joint distribution P. The model is still inconsistent when intermediate values $0 < \epsilon < 1$ are used, in this case the divergence has a global optimum that is somewhere between the true joint P and the factorised distribution $P_{x_1}P_{x_2}$.

Based on this analysis we suggest that scheduled sampling works by pushling models towards a trivial solution of memorising distribution of symbols conditioned on their position in the sequence,

[1] more precisely, maximum likelihood minimises the cross-entropy $KL[P\|Q] + H[P]$, where $H[P]$ is the differential entropy of training data.

[2] The authors also propose taking argmax of each distribution instead of sampling, this case is harder to analyse but we think our general observations still hold.

rather than on the prefix of preceding symbols. In recurrent neural network (RNN) terminology, this would means that the optimal architecture under SS uses its hidden states merely to implement a simple counter, and learns to pay no attention whatsoever to the content of the sequence prefix. While this may indeed lead to models that are more likely to recover from mistakes, we believe it fails to address the limitations of maximum likelihood the authors initially set out to solve.

How could an inconsistent training procedure still achieve state-of-the-art performance in the image captioning challenge? There are multiple possible explanations to this. We speculate that the optimisation was not run until full convergence, and perhaps an improvement over the maximum likelihood solution was found as a coincidence due to the the interplay between early stopping, random restarts, the specific structure of the model class and the annealing schedule for ϵ.

5 THE DIAGNOSIS

After discussing scheduled sampling, a method proposed to remedy the symptoms explained in section 3, we now seek a better explanation of why those symptoms exist in the first place. We will now leave the autoregressive model class, and consider probabilistic generative models in their full generality.

The symptoms outlined in Section 3 can be attributed to a mismatch between the loss function used for training (likelihood) and the loss used for evaluating the model (the perceptual quality of samples produced by the model). To fix this problem we need a training objective that more closely matches the perceptual metric used for evaluation, and ideally one that allows for a consistent statistical estimation framework.

5.0.1 A MODEL OF NO-REFERENCE PERCEPTUAL QUALITY ASSESSMENT

When researchers evaluate their generative models for perceptual quality, they draw samples from it, then - for lack of a better word - *eyeball* the samples. In visual information processing this is often referred to as no-reference perceptual quality assessment (see e. g. Wang et al., 2002). When using the model in an application like caption generation, we typically draw a sample from a conditional model $\mathbf{y}|\mathbf{x} \sim Q_{\mathbf{y}|\mathbf{x}}$, where $mathbfx$ represents the context of the query, and present it to a human observer. We would like each sample to pass a Turing test. We want the human observer to feel like $\mathbf{y}$ is a plausible naturally occurring response, within the context of the query $\mathbf{x}$.

In this section, we will propose that the KL divergence $KL[Q\|P]$ can be used as an idealised objective function to describe the no-reference perceptual quality assessment scenario. First of all, we make the assumption that the perceived quality of each sample is related to the *surprisal* $-\log Q_{human}(x)$ under the human observers' subjective prior of stimuli $Q_{human}(\mathbf{x})$ CITE. We further assume that the human observer has learnt an accurate model of the natural distribution of stimuli, thus, $Q_{human}(\mathbf{x}) = P(\mathbf{x})$. These two assumptions suggest that in order to optimise our chances in the Turing test scenario, we need to minimise the following cross-entropy or perplexity term:

$$-\mathbb{E}_{\mathbf{x}\sim Q} \log P(x) \tag{9}$$

Note that this perplexity is the exact opposite average negative log likelihood $-\mathbb{E}_{\mathbf{x}\sim P} \log Q(x)$, with the role of P and Q changed.

However, the objective in Eqn. 9 would be maximised by a model Q that deterministically picks the most likely stimulus. To enforce diversity one can simultaneously try to maximise the entropy of Q. This leaves us with the following KL divergence to optimise:

$$KL[Q\|P] = -\mathbb{E}_{\mathbf{x}\sim Q} \log P(x) + \mathbb{E}_{\mathbf{x}\sim Q} \log Q(x) \tag{10}$$

It is known that $KL[Q\|P]$ is minimised when $P = Q$, therefore minimising it would correspond to a consistent estimation strategy. However, it is only well-defined when P is positive and bounded in the full support of Q, which is not the case when P is an empirical distribution of samples and Q is a smooth probabilistic model. For this reason, $KL[Q\|P]$ is not viable as a practical training objective

in statistical esimation. Still, we can use it as our idealised perceptual quality metric to motivate our choice of practical objective functions.

5.0.2 How does this explain the symptoms?

The differences in behaviour between $KL[Q\|P]$ and $KL[P\|Q]$ are well understood and exploited for example in the context of approximate Bayesian inference (Lacoste-Julien et al., 2011; MacKay, 2003; Minka, 2001). The differences are most visible when model underspecification is present: imagine trying to model a multimodal P with a simpler, unimodal model Q. Minimising $KL[P\|Q]$ corresponds to moment matching and has a tendency to find models Q that cover all the modes of P, at the cost of placing probability mass where P has none. Minimising $KL[Q\|P]$ in this case leads to a mode-seeking behaviour: the optimal Q will typically concentrate around the largest mode of P, at the cost of completely ignoring smaller modes. These differences are illustrated visually in Figure 1, panels B and D.

In the context of generative models this means that minimising $KL[P\|Q]$ often leads to models that overgeneralise, and sometimes produce samples that are very unlikely under P. This would explain why recurrent neural networks trained via maximum likelihood also have a tendency to produce completely unseen sequences. Minimising $KL[P\|Q]$ will aim to create a model that can generate all the behaviour that is observed in real data, at the cost of introducing behaviours that are never seen. By contrast, if we train a generative model by minimising $KL[Q\|P]$, the model will very conservatively try to avoid any behaviour that is unlikely under P. This comes at the cost of ignoring modes of P completely, unless those additional modes can be modelled without introducing probability mass in regions where P has none.

Once again, both $KL[P\|Q]$ and $KL[Q\|P]$ define consistent estimation strategies. They differ in the kind of errors they make under severe model misspecification particularly in high dimensions.

6 Generalised Adversarial Training

We theorised that $KL[Q\|P]$ may be a more meaningful training objective if our aim was to improve the perceptual quality of generative models, but it is impractical as an objective function.

Here we show that a generalised version of adversarial training (Goodfellow et al., 2014) can be used to approximate training based on $KL[Q\|P]$. Adversarial training can be described as minimising an approximation to the Jensen-Shannon divergence between P and Q (Goodfellow et al., 2014; Theis et al., 2015). The JS divergence between P and Q is defined by the following formula:

$$JSD[P\|Q] = JSD[P\|Q] = \frac{1}{2}KL\left[P\middle\|\frac{P+Q}{2}\right] + \frac{1}{2}KL\left[Q\middle\|\frac{P+Q}{2}\right] \tag{11}$$

Unlike KL divergence, the JS divergence is symmetric in its arguments, and can be understood as being somewhere between $KL[Q\|P]$ and $KL[P\|Q]$ in terms of its behaviour. One can therefore hope that JSD would behave a bit more like $KL[Q\|P]$ and therefore ultimately tend to produce more realistic samples. Indeed, the behaviour of JSD minimisation under model misspecification is more similar to $KL[Q\|P]$ than $KL[P\|Q]$ as illustrated in Figure 1. Empirically, methods built on adversarial training do tend to produce appealing samples (Goodfellow et al., 2014; Denton et al., 2015).

However, we can even formally show that JS divergence is indeed an interpolation between the two KL divergences in the following sense. Let us consider a more general definition of Jensen-Shannon divergence, parametrised by a non-trivial probability $0 < \pi < 1$:

$$JS_\pi[P\|Q] = \pi \cdot KL[P\|\pi P + (1-\pi)Q] + (1-\pi)KL[Q\|\pi P + (1-\pi)Q]. \tag{12}$$

For any given value of π this generalised Jensen-Shannon divergence is not symmetric in its arguments P and Q anymore, instead the following weaker notion of symmetry holds:

$$JS_\pi[P\|Q] = JS_{1-\pi}[Q\|P] \tag{13}$$

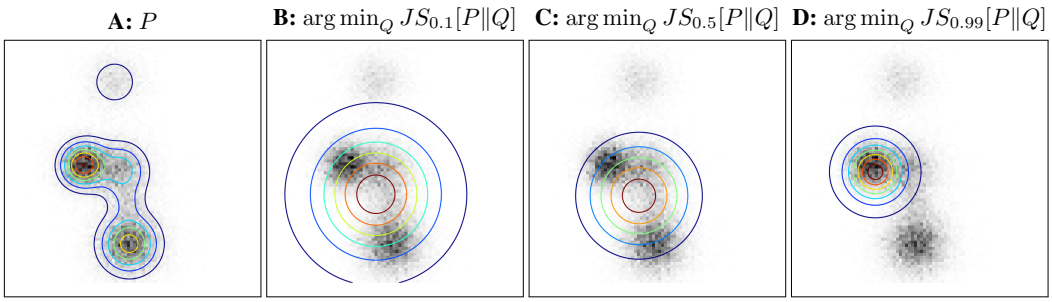

Figure 1: Illustrating the behaviour of the generalised JS divergence under model underspecification for a range of values of π. Data is drawn from a multivariate Gaussian distribution P **(A)** and we aim approximate it by a single isotropic Gaussian **(B-D)**. Contours show level sets the approximating distribution, overlaid on top of the 2D histogram of observed data. For $\pi = 0.1$, JS divergence minimisation behaves like maximum likelihood **(B)**, resulting in the characteristic moment matching behaviour. For $\pi = 0.99$ **(D)**, the behaviour becomes more akin to the mode-seeking behaviour of minimising $KL[Q\|P]$. For the intermediate value of $\pi = 0.5$ **(C)** we recover the standard JS divergence approximated by adversarial training. To produce this illustraiton we used software made available by Theis et al. (2015).

It is easy to show that JS_π divergence converges to 0 in the limit of both $\pi \to 0$ and $\pi \to 1$. Crucially, it can be shown that the gradients with respect to π at these two extremes recover $KL[Q\|P]$ and $KL[P\|Q]$, respectively. A proof of this property can be obtained by considering the Taylor-expansion $KL[Q\|Q + a] \approx a^T H a$, where H is the positive definite Hessian and substituting $a = \pi(P - Q)$ as follows:

$$\lim_{\pi \to 0} \frac{JSD[P\|Q;\pi]}{\pi} = \lim_{\pi \to 0} \left\{ KL[P\|\pi P + (1-\pi)Q] + \frac{1-\pi)}{\pi} KL[Q\|\pi P + (1-\pi)Q] \right\} \quad (14)$$

$$= KL[P\|Q] + \lim_{\pi \to 0} \frac{1}{\pi} \pi^2 (P-Q)^T H (P-Q) \quad (15)$$

$$= KL[P\|Q] \quad (16)$$

Therefore, we can say that for infinitisemally small values of π, JS_π is approximately proportional to $KL[P\|Q]$:

$$\frac{JS_\pi[P\|Q]}{\pi} \approx KL[P\|Q]. \quad (17)$$

And by symmetry in Eqn. 13 we also have that for small values of π

$$\frac{JS_{1-\pi}[P\|Q]}{1-\pi} \approx KL[Q\|P] \quad (18)$$

This limiting behaviour also implies that for small values of π, JS_π has the same optima as $KL[P\|Q]$. For values of π close to 1, JS_π has the same optima as $KL[Q\|P]$. Thus, by minimising JS_π divergence for a range of $\pi \in (0, 1)$ allows us to interpolate between the behaviour of $KL[P\|Q]$ and $KL[Q\|P]$.

We note that JS_π can also be approximated via adversarial training as described in (Goodfellow et al., 2014). The practical meaning of the parameter π is the ratio of labelled samples the adversarial disctiminator network receives from Q and P in the training procedure. $\pi = \frac{1}{2}$ implies that the adversarial network is faced with a balanced classification problem as it is the case in the original algorithm, for $\pi < \frac{1}{2}$ samples from the real distribution P are overrepresented. Similarly, when $\pi > \frac{1}{2}$, the classification problem is biased towards Q. Thus, the generality of generative adversarial

networks can be greatly improved by incorporating a minor change to the procedure. We note that this change may have adverse effects on the convergence properties of the algorithm, which we have not investigated.

7 CONCLUSIONS

In this paper our goal was to understand which objective functions work and which ones don't in the context of generative models. Here we were only interested in models that are created for the purpose of drawing samples from, and we excluded other use-cases such as semi-supervised feature learning.

Our findings and recommendations can be summarised as follows:

1. Maximum likelihood should not be used as the training objective if the end goal is to draw realistic samples from the model. Models trained via maximum likelhiood have a tendency to overgeneralise and generate unplausible samples.
2. Scheduled sampling, designed to overcome the shortcomings of maximum likelihood, fails to address the fundamental problems, and we showed it is an inconsistent training strategy.
3. We theorise that $KL[Q\|P]$ could be used as an idealised objective function to describe the no-reference perceptual quality assessment scenario, but it is impractical to use in practice.
4. We propose the generalised Jensen-Shannon divergence as a promising, more tractable objective function that can effectively interpolate between maximum likelihood and $KL[Q\|P]$-minimisation.
5. Our analysis suggests that adversarial training strategies are a the best choice for generative modelling, and we propose a more flexible algorithm based on our generalised JS divergence.

While our analysis highlighted the merits of adversarial training, it should be noted that the method is still very new, and has serious practical limitations. Firstly, the generative adversarial network algorithm is based on sampling from the approximate model Q, which is highly inefficient in high dimensional spaces. This limits the applicability of these methods to low-dimensional problems, and increases sensitivity to hyperparameters. Secondly, it is unclear how to employ adversarial traning on discrete probabilistic models, where the sampling process cannot be described as a differentiable operation. To make adversarial training practically viable these limitations need to be addressed in future work.

ACKNOWLEDGMENTS

I would like to thank Lucas Theis and Hugo Larochelle for useful discussions. I would like to thank authors Theis, van den Oord, and Bethge (2015) for kindly providing the source code for creating illustrations in Figure 1.

REFERENCES

Bengio, Samy, Vinyals, Oriol, Jaitly, Navdeep, and Shazeer, Noam M. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems, NIPS*, 2015. URL `http://arxiv.org/abs/1506.03099`.

Denton, Emily, Chintala, Soumith, Szlam, Arthur, and Fergus, Rob. Deep generative image models using a laplacian pyramid of adversarial networks. *arXiv preprint arXiv:1506.05751*, 2015.

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

Gneiting, Tilmann and Raftery, Adrian E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Han, Charles, Risser, Eric, Ramamoorthi, Ravi, and Grinspun, Eitan. Multiscale texture synthesis. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*, 27(3):51:1–51:8, 2008.

Hinton, Geoffrey E, Osindero, Simon, and Teh, Yee-Whye. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Hyvärinen, Aapo. Consistency of pseudolikelihood estimation of fully visible boltzmann machines. *Neural Computation*, 18(10):2283–2292, 2006.

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

Lacoste-Julien, Simon, Huszár, Ferenc, and Ghahramani, Zoubin. Approximate inference for the loss-calibrated bayesian. In *International Conference on Artificial Intelligence and Statistics*, pp. 416–424, 2011.

Larochelle, Hugo and Murray, Iain. The neural autoregressive distribution estimator. In *International Conference on Artificial Intelligence and Statistics*, pp. 29–37, 2011.

Li, Yujia, Swersky, Kevin, and Zemel, Richard. Generative moment matching networks. *arXiv preprint arXiv:1502.02761*, 2015.

Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pp. 740–755. Springer, 2014.

MacKay, David JC. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Minka, Thomas P. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

Ou, Zhijian and Zhang, Yang. Probabilistic acoustic tube: a probabilistic generative model of speech for speech analysis/synthesis. In *International Conference on Artificial Intelligence and Statistics*, pp. 841–849, 2012.

Sordoni, Alessandro, Galley, Michel, Auli, Michael, Brockett, Chris, Ji, Yangfeng, Mitchell, Margaret, Nie, Jian-Yun, Gao, Jianfeng, and Dolan, Bill. A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714*, 2015.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc VV. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Theis, Lucas and Bethge, Matthias. Generative image modeling using spatial lstms. *arXiv preprint arXiv:1506.03478*, 2015.

Theis, Lucas, Hosseini, Reshad, Bethge, Matthias, and Hsiao, Chuhsing Kate. Mixtures of conditional gaussian scale mixtures applied to multiscale image representations. *PloS one*, 7(7):e39857, 2012.

Theis, Lucas, van den Oord, Aaron, and Bethge, Matthias. A note on the evaluation of generative models. arXiv:1511.01844, Nov 2015. URL `http://arxiv.org/abs/1511.01844`.

Vinyals, Oriol and Le, Quoc. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

Vinyals, Oriol, Toshev, Alexander, Bengio, Samy, and Erhan, Dumitru. Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*, 2014.

Wang, Zhou, Sheikh, Hamid R, and Bovik, Alan C. No-reference perceptual quality assessment of jpeg compressed images. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pp. I–477. IEEE, 2002.