Semi-Supervised Learning with Generative Adversarial Networks

Augustus Odena   AUGUSTUS.ODENA@GMAIL.COM

Abstract

We extend Generative Adversarial Networks (GANs) to the semi-supervised context by forcing the discriminator network to output class labels. We train a generative model G and a discriminator D on a dataset with inputs belonging to one of N classes. At training time, D is made to predict which of N+1 classes the input belongs to, where an extra class is added to correspond to the outputs of G. We show that this method can be used to create a more data-efficient classifier and that it allows for generating higher quality samples than a regular GAN.

1. Introduction

Work on generating images with Generative Adversarial Networks (GANs) has shown promising results (Goodfellow et al., 2014). A generative net G and a discriminator D are trained simultaneously with conflicting objectives. G takes in a noise vector and outputs an image, while D takes in an image and outputs a prediction about whether the image is a sample from G. G is trained to maximize the probability that D makes a mistake, and D is trained to minimize that probability. Building on these ideas, one can generate good output samples using a cascade (Denton et al., 2015) of convolutional neural networks. More recently (Radford et al., 2015), even better samples were created from a single generator network. Here, we consider the situation where we try to solve a semi-supervised classification task and learn a generative model simultaneously. For instance, we may learn a generative model for MNIST images while we train an image classifier, which we'll call C. Using generative models on semi-supervised learning tasks is not a new idea - Kingma et al. (2014) expand work on variational generative techniques (Kingma & Welling, 2013; Rezende et al., 2014) to do just that. Here, we attempt to do something similar with GANs. We are not the first to use GANs for semi-supervised learning. The CatGAN (Springenberg, 2015) modifies the objective function to take into account mutual information between observed examples and their

---

predicted class distribution. In Radford et al. (2015), the features learned by D are reused in classifiers.

The latter demonstrates the utility of the learned representations, but it has several undesirable properties. First, the fact that representations learned by D help improve C is not surprising - it seems reasonable that this should work. However, it also seems reasonable that learning a good C would help to improve the performance of D. For instance, maybe images where the output of C has high entropy are more likely to come from G. If we simply use the learned representations of D after the fact to augment C, we don't take advantage of this. Second, using the learned representations of D after the fact doesn't allow for training C and G simultaneously. We'd like to be able to do this for efficiency reasons, but there is a more important motivation. If improving D improves C, and improving C improves D (which we know improves G) then we may be able to take advantage of a sort of feedback loop, in which all 3 components (G,C and D) iteratively make each other better.

In this paper, inspired by the above reasoning, we make the following contributions:

- First, we describe a novel extension to GANs that allows them to learn a generative model and a classifier simultaneously. We call this extension the Semi-Supervised GAN, or SGAN.

- Second, we show that SGAN improves classification performance on restricted data sets over a baseline classifier with no generative component.

- Finally, we demonstrate that SGAN can significantly improve the quality of the generated samples and reduce training times for the generator.

2. The SGAN Model

The discriminator network D in a normal GAN outputs an estimated probability that the input image is drawn from the data generating distribution. Traditionally this is implemented with a feed-forward network ending in a single sigmoid unit, but it can also be implemented with a softmax output layer with one unit for each of the classes [REAL, FAKE]. Once this modification is made, it's simple to see that D could have N+1 output units corresponding to

Algorithm 1 SGAN Training Algorithm

Input: I: number of total iterations
for $i = 1$ **to** I **do**
　Draw m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
　Draw m examples $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ from data generating distribution $p_d(x)$.
　Perform gradient descent on the parameters of D w.r.t. the NLL of D/C's outputs on the combined minibatch of size $2m$.
　Draw m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
　Perform gradient descent on the parameters of G w.r.t. the NLL of D/C's outputs on the minibatch of size m.
end for

[CLASS-1, CLASS-2, ...CLASS-N, FAKE]. In this case, D can also act as C. We call this network D/C.

Training an SGAN is similar to training a GAN. We simply use higher granularity labels for the half of the minibatch that has been drawn from the data generating distribution. D/C is trained to minimize the negative log likelihood with respect to the given labels and G is trained to maximize it, as shown in Algorithm 1. We did not use the modified objective trick described in Section 3 of Goodfellow et al. (2014).

Note: in concurrent work, (Salimans et al., 2016) propose the same method for augmenting the discriminator and perform a much more thorough experimental evaluation of the technique.

3. Results

The experiments in this paper were conducted with https://github.com/DoctorTeeth/supergan, which borrows heavily from https://github.com/carpedm20/DCGAN-tensorflow and which contains more details about the experimental setup.

3.1. Generative Results

We ran experiments on the MNIST dataset (LeCun et al., 1998) to determine whether an SGAN would result in better generative samples than a regular GAN. Using an architecture similar to that in Radford et al. (2015), we trained an SGAN both using the actual MNIST labels and with only the labels REAL and FAKE. Note that the second configuration is semantically identical to a normal GAN. Figure 1 contains examples of generative outputs from both GAN and SGAN. The SGAN outputs are significantly more clear than the GAN outputs. This seemed to hold true across different initializations and network architectures, but it is

Table 1. Classifier Accuracy

EXAMPLES	CNN	SGAN
1000	0.965	0.964
100	0.895	0.928
50	0.859	0.883
25	0.750	0.802

hard to do a systematic evaluation of sample quality for varying hyperparameters.
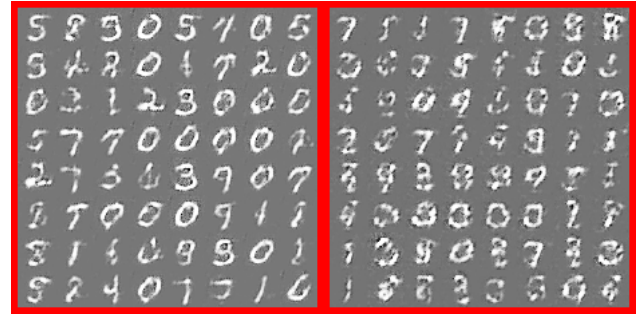

Figure 1. Output samples from SGAN and GAN after 2 MNIST epochs. SGAN is on the left and GAN is on the right.

3.2. Classifier Results

We also conducted experiments on MNIST to see whether the classifier component of the SGAN would perform better than an isolated classifier on restricted training sets. To train the baseline, we train SGAN without ever updating G. SGAN outperforms the baseline in proportion to how much we shrink the training set, suggesting that forcing D and C to share weights improves data-efficiency. Table 1 includes detailed performance numbers. To compute accuracy, we took the maximum of the outputs not corresponding to the FAKE label. For each model, we did a random search on the learning rate and reported the best result.

4. Conclusion and Future Work

We are excited to explore the following related ideas:

- Share some (but not all) of the weights between D and C, as in the dual autoencoder (Sutskever et al., 2015). This could allow some weights to be specialized to discrimination and some to classification.

- Make GAN generate examples with class labels (Mirza & Osindero, 2014). Then ask D/C to assign one of $2N$ labels [REAL-ZERO, FAKE-ZERO, ...,REAL-NINE, FAKE-NINE].

- Introduce a ladder network (Rasmus et al., 2015) L in place of D/C, then use samples from G as unlabeled data to train L with.

Acknowledgements

References

Abadi, Martín, Agarwal, Ashish, Barham, Paul, Brevdo, Eugene, Chen, Zhifeng, Citro, Craig, Corrado, Gregory S., Davis, Andy, Dean, Jeffrey, Devin, Matthieu, Ghemawat, Sanjay, Goodfellow, Ian J., Harp, Andrew, Irving, Geoffrey, Isard, Michael, Jia, Yangqing, Józefowicz, Rafal, Kaiser, Lukasz, Kudlur, Manjunath, Levenberg, Josh, Mané, Dan, Monga, Rajat, Moore, Sherry, Murray, Derek Gordon, Olah, Chris, Schuster, Mike, Shlens, Jonathon, Steiner, Benoit, Sutskever, Ilya, Talwar, Kunal, Tucker, Paul A., Vanhoucke, Vincent, Vasudevan, Vijay, Viégas, Fernanda B., Vinyals, Oriol, Warden, Pete, Wattenberg, Martin, Wicke, Martin, Yu, Yuan, and Zheng, Xiaoqiang. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467, 2016. URL http://arxiv.org/abs/1603.04467.

Denton, Emily L., Chintala, Soumith, Szlam, Arthur, and Fergus, Robert. Deep generative image models using a laplacian pyramid of adversarial networks. *CoRR*, abs/1506.05751, 2015. URL http://arxiv.org/abs/1506.05751.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Networks. *ArXiv e-prints*, June 2014.

Kingma, D. P and Welling, M. Auto-Encoding Variational Bayes. *ArXiv e-prints*, December 2013.

Kingma, Diederik P., Rezende, Danilo Jimenez, Mohamed, Shakir, and Welling, Max. Semi-supervised learning with deep generative models. *CoRR*, abs/1406.5298, 2014. URL http://arxiv.org/abs/1406.5298.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Mirza, Mehdi and Osindero, Simon. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. URL http://arxiv.org/abs/1411.1784.

Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. URL http://arxiv.org/abs/1511.06434.

Rasmus, Antti, Valpola, Harri, Honkala, Mikko, Berglund, Mathias, and Raiko, Tapani. Semi-supervised learning with ladder network. *CoRR*, abs/1507.02672, 2015. URL http://arxiv.org/abs/1507.02672.

Rezende, D., Mohamed, S., and Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *ArXiv e-prints*, January 2014.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved Techniques for Training GANs. *ArXiv e-prints*, June 2016.

Springenberg, J. T. Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks. *ArXiv e-prints*, November 2015.

Sutskever, I., Jozefowicz, R., Gregor, K., Rezende, D., Lillicrap, T., and Vinyals, O. Towards Principled Unsupervised Learning. *ArXiv e-prints*, November 2015.