Penalized Loss-Sensitive Generative Adversarial Network

Emma Ambrosini
Department of Computer Science
Simmons College
emma.ambrosini@simmons.edu

Marzieh Edraki
Department of Computer Science
University of Central Florida
M.Edraki@knights.ucf.edu

Guo-Jun Qi
Department of Computer Science
University of Central Florida
GuoJun.Qi@ucf.edu

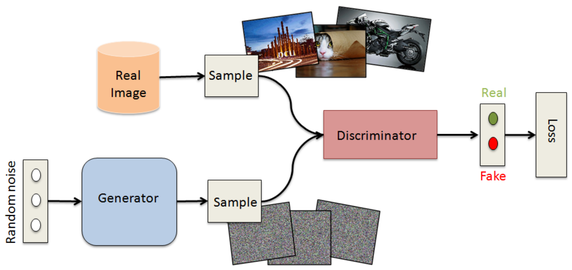
Abstract

▪ The **Loss-Sensitive Generative Adversarial Network Penalized with Gradient (LS-GAN-PG)** is an improvement of LS-GAN that exploits the norm of gradient of loss function with respect to its domain as a mechanism to **reduce the complexity** of generative models and to **decrease the chance of being over-fitted** to the few training examples. The LS-GAN *proposes a novel paradigm for training the classical GAN model that* utilizes a **loss function** to *quantify the quality of generated samples with constraint that* ***cost of real images must be smaller than the cost of generated samples*** *at least by amount of an adaptable margin between fake and real samples so the* model can focus on improving poor ones (those with a higher loss margin value)

▪ This is something the original GAN was not capable of doing – its learning was based on probability and a *discriminator* rather than a unique loss calculation.

Background

Generative Adversarial Networks – method of training generative models that utilizes a *generator* and *discriminator* which train & work against each other in a "minimax two-player game" [1]



- **Discriminator** –
  - Reads in images
    1) actual training samples
    2) 'false' samples created by generator
  - Classifies them as 'real' (inputs) or 'fake' (generated) based off learning
- **Generator** –
  - Tries to create images that will be labeled 'real' by the discriminator – this becomes problematic in GAN learning
  - In LS-GAN, samples are improved based on *G* output loss margin; higher/lower loss value
- **Loss –** value calculated between real & fake samples, quality level of generated samples
- Our model "**penalizes**"/minimizes this value to better its accuracy, learning, convergence, & to regularize

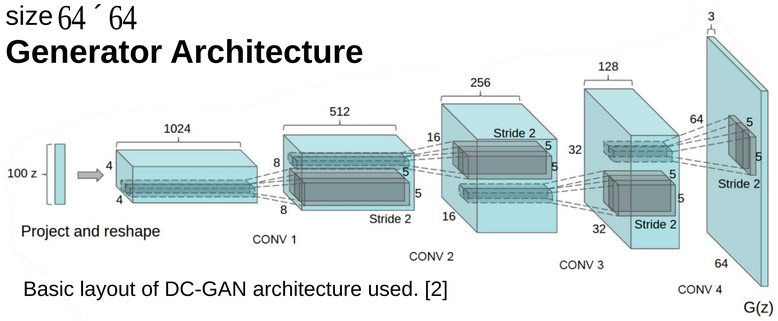
Methods

- **Main Idea of LS-GAN**:

$$L_\theta(x) \le L_\theta(G_\varphi(z)) - \Delta(x, G_\varphi(z))$$

Difference between real input (x) and generated sample ($G\varphi(z)$)

- This constraint causes the model to prioritize samples with a higher loss value in improving generation quality
- **LS-GAN-PG** applies *gradient norm* $\|\nabla_x L(x)\|$ as regularizer for the cost function to control the complexity of the generative model
- We applied LS-GAN-PG to generate images of size 64×64
- **Generator Architecture**



Basic layout of DC-GAN architecture used. [2]

- **Cost model Architecture** - Sequence of five layers of convolution with *stride* of 2 followed by *batch normalization* and *Rectified linear function* as nonlinearity
- **Training process**
  - We used **celebA** dataset, a massive collection celebrity face photos.
  - **Adam optimizer** has been used to train the generator and cost model iteratively
  - Batch size of 64
  - Maximum number of training epochs is 24

Findings

Discussion

- Both the Tensorflow LS-GAN and LS-GAN-PG were able to run, learn, produce decent quality samples
- Faster convergence in model with gradient penalty applied than without
- Sample quality in the penalized LS-GAN evened out and began to slightly decrease at around the eleventh epoch
  - Could be due to the penalty, regularization working

Related Works

- **Generative Adversarial Network** [1] – proposal of generative network method of training in which a discriminative model works against the generative model in training, discerning between 'real' and 'fake' samples. The generator attempts to fool the discriminator by improving sample quality.
- **DC-GAN** [2] – Deep Convolutional GAN that applies the batch normalization in generator to alleviate the vanishing gradient problem on deep models. Has a focus on unsupervised learning and use D as a feature extractor.

References

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.
- [2] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434,* 2015.
  - Image: https://github.com/carpedm20/DCGAN-tensorflow
- [3] Guo-Jun Qi, "Loss-Sensitive Generative Adversarial Networks on Lipschitz Densities", *arXiv preprint arXiv:1701.06264*, 2017

Acknowledgments - *The support for this work was provided by the National Science Foundation REU program under Award No. 1560302. Any opinions, findings, and conclusions and recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.*