MarginGAN: Adversarial Training in Semi-Supervised Learning

Jinhao Dong — Xidian University

Tong Lin* — Peking University

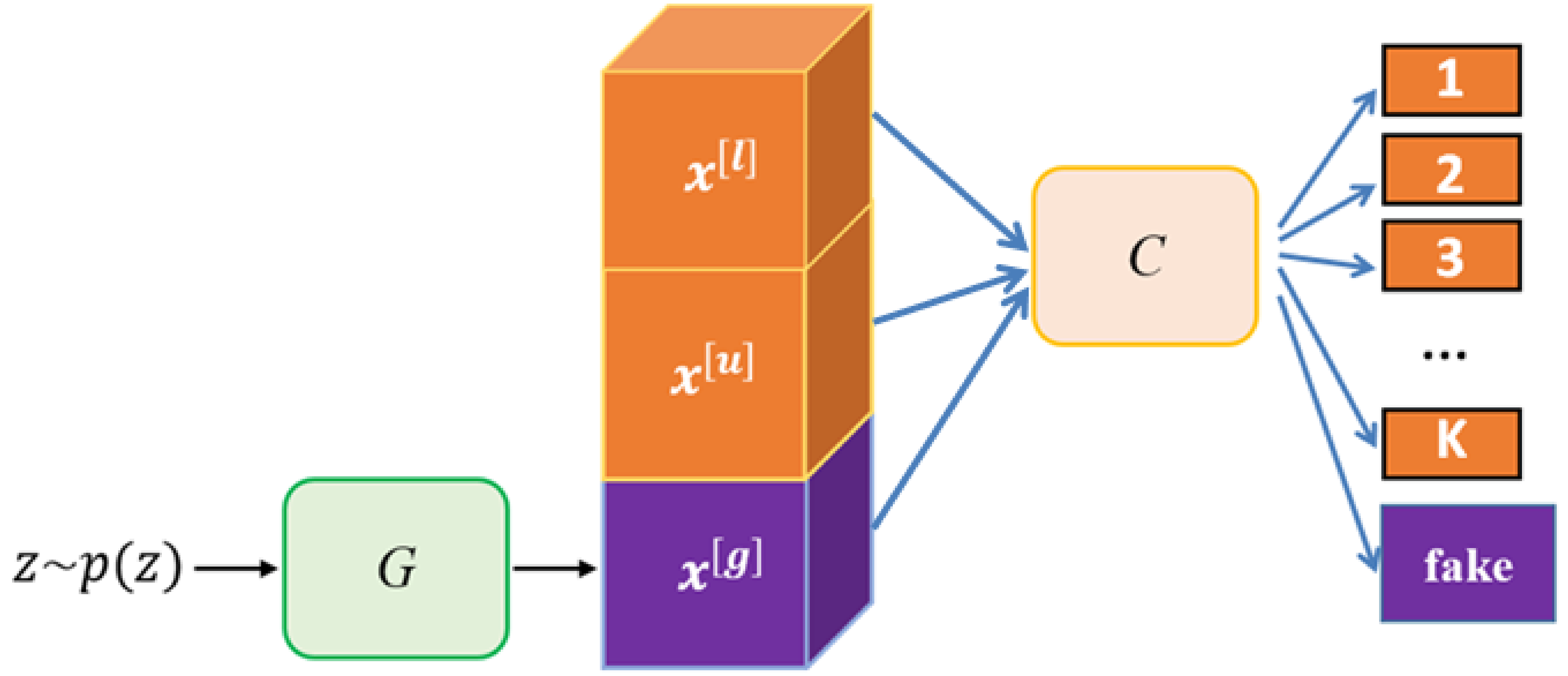
NEURAL INFORMATION PROCESSING SYSTEMS
VANCOUVER
DEC 8-14, 2019

Summary

- A three-player GAN model called MarginGAN is proposed for semi-supervised learning (SSL)
- The discriminator is trained as usual to distinguish real examples from fake examples produced by the generator
- The classifier aims at increasing the margin of real examples and decreasing the margin of fake examples
- The generator attempts to yield realistic and large-margin examples in order to fool the discriminator and the classifier simultaneously
- Pseudo labels are used for unlabeled and fake examples in training

Prior Work

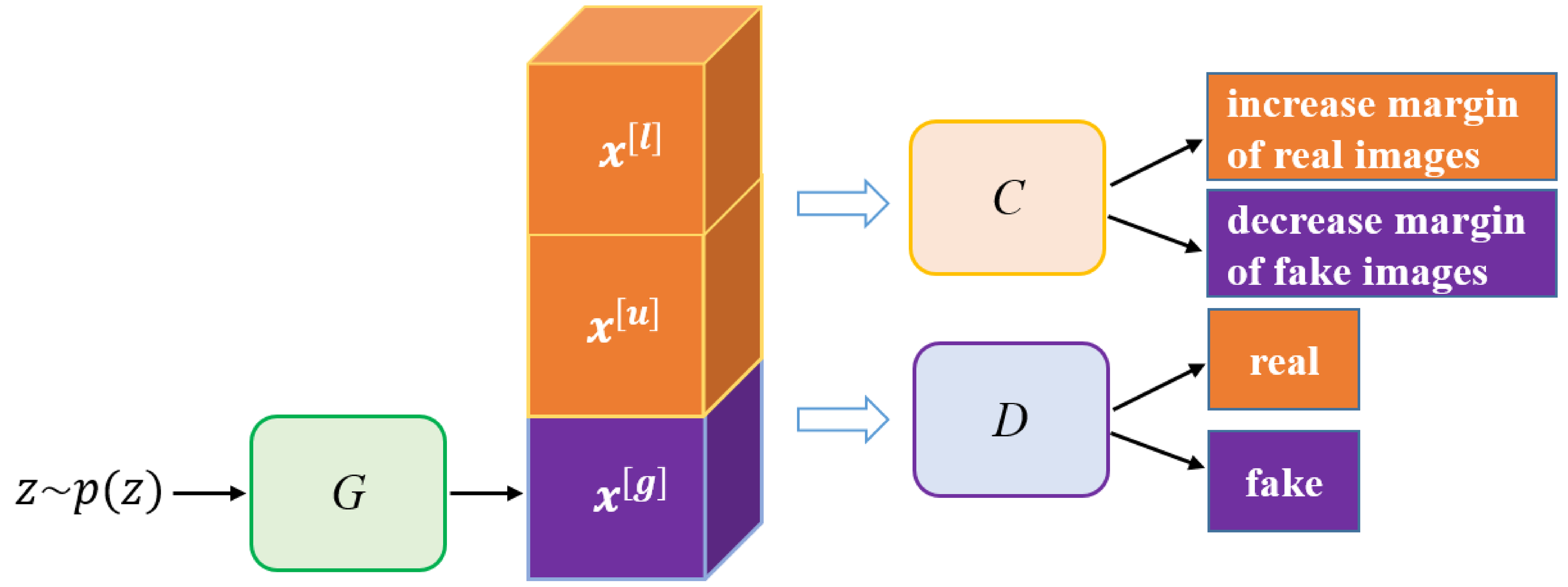
- Feature Matching GANs (Salimans et al., NeurIPS 2016)



- Good SSL requires a “bad” GAN (Dai et al., NeurIPS 2017)
- Question: How to design a SSL model composed of a classifier, a discriminator, and a “bad” generator?

Our MarginGAN Model

- Architecture Overview



- The Discriminator

$$Loss(D) = -\{E_{x\sim p^{[l]}(x)}[\log(D(x))] + E_{\widetilde{x}\sim p^{[u]}(\widetilde{x})}[\log(D(\widetilde{x}))] + E_{z\sim p(z)}[\log(1-D(G(z)))]\}$$

- The Classifier: *min cross-entropy = max multiclass margins*

$$Loss(C) = Loss(C^{[l]}) + Loss(C^{[u]}) + Loss(C^{[g]})$$

labeled: $$Loss(C^{[l]}) = E_{(x,y)\sim p^{[l]}(x,y)}\left[-\sum_{i=1}^{k} y_i \log\left(C(x)_i\right)\right]$$

unlabeled: $$Loss(C^{[u]}) = E_{\widetilde{x}\sim p^{[u]}(\widetilde{x})}\left[-\sum_{i=1}^{k} \boxed{\widetilde{y}_i^{[u]}} log(C(\widetilde{x})_i)\right]$$

generated: $$Loss(C^{[g]}) = E_{z\sim p(z)}\left[-\sum_{i=1}^{k} \boxed{\widetilde{y}_i^{[g]}} log(1-C(G(z))_i)\right]$$

pseudo labels

- The Generator

$$Loss(G) = -E_{z\sim p(z)}\left[\log\left(D(G(z))\right)\right] + E_{z\sim p(z)}\left[Loss_{CE}\left(\boxed{\widetilde{y}^{[g]}}, C(G(z))\right)\right]$$

- Minimax Game

$$\min_G \max_{D,C} J(G,D,C)$$
$$= \left\{E_{x\sim p^{[l]}(x)}[\log(D(x))] + E_{\widetilde{x}\sim p^{[u]}(\widetilde{x})}[\log(D(\widetilde{x}))] + E_{z\sim p(z)}[\log(1-D(G(z)))]\right\}$$
$$+ \left\{E_{(x,y)\sim p^{[l]}(x,y)}\left[\mathrm{Margin}(x,y)\right] + E_{\widetilde{x}\sim p^{[u]}(\widetilde{x})}\left[\mathrm{Margin}(\widetilde{x},\widetilde{y}^{[u]})\right]\right.$$
$$\left. + E_{z\sim p(z)}\left[1-\mathrm{Margin}(G(z),\widetilde{y}^{[g]})\right]\right\},$$

if we redefine $\mathrm{Margin}(x,y) \doteq \langle y, \log C(x)\rangle$ and $1-\mathrm{Margin}(x,y) \doteq \langle y, \log(1-C(x))\rangle$.

Experiments

Error rates on MNIST

# of labels	100	600	1000	3000
NN	25.81	11.44	10.70	6.04
SVM	23.44	8.85	7.77	4.21
CNN	22.98	7.68	6.45	3.35
TSVM	16.81	6.16	5.38	3.45
DBN-rNCA	—	8.70	—	3.30
EmbedNN	16.86	5.97	5.73	3.59
CAE	13.47	6.30	4.77	3.22
MTC	12.03	5.13	3.64	2.57
dropNN	21.89	8.57	6.59	3.72
+PL	16.15	5.03	4.30	2.80
+PL+DAE	10.49	4.01	3.46	2.69
MarginGAN (ours)	3.53 ± 0.57	3.03 ± 0.60	2.87 ± 0.71	2.06 ± 0.20

Ablation study on MNIST

	Settings	Error Rates (%)	Training Time (sec.)
Normal Training	L	8.21 ± 0.82	408.41 ± 26.17
	L + U	4.54 ± 0.41	1305.64 ± 495.18
	L + U + G	3.20 ± 0.62	367.79 ± 82.82
Extreme Training	U	89.53 ± 0.81	—
	U + G	7.40 ± 5.01	886.83 ± 193.98

METHODS	SVHN (500 labels)	CIFAR-10 (1000 labels)	CIFAR-10 (4000 labels)
Ladder [18]	—	—	20.04 ± 0.47
CatGAN [14]	—	—	19.58 ± 0.58
FM GANs [8]	—	—	18.63 ± 2.32
Triple-GAN [15]	—	—	18.82 ± 0.32
SGAN [17]	—	—	17.26 ± 0.69
∏ model [7]	6.83 ± 0.66	27.36 ± 1.20	13.20 ± 0.27
MarginGAN (ours)	6.07 ± 0.43	10.39 ± 0.43	6.44 ± 0.10

(a) MNIST

(b) SVHN

(c) CIFAR-10