Lecture Notes for

Neural Networks and Machine Learning

BigGAN

Logistics and Agenda

- Logistics
  - **Student Presentation**: None
- Agenda
  - Town Hall
  - BigGAN
    - Big GAN? or Biggin?
  - Move onto: GAN-Zooks

GAN Town Hall

BigGAN

In a field with 1000's of competing papers, BigGAN is here to use the most meaningful Portions of each paper and put them into One BIG paper.

LARGE SCALE GAN TRAINING FOR HIGH FIDELITY NATURAL IMAGE SYNTHESIS

Andrew Brock[*, †]
Heriot-Watt University
ajb5@hw.ac.uk

Jeff Donahue[†]
DeepMind
jeffdonahue@google.com

Karen Simonyan[†]
DeepMind
simonyan@google.com

BigGAN Overview

- This is an agglomeration of GAN Knowledge from *2013-2020*
- Training is hard, so use heuristics
  - large batches, feature matching
  - use hinge loss (max margin)
- Use attention, conditional classes, spectral normalization, moving average of weights, orthogonal weight initialization, skip connections, orthogonal regularizers
- **Truncation trick:** sample a wide range during training $\sigma = \lambda$, then truncate for evaluation $\sigma = \frac{\lambda}{2}$

- Large Scale GAN Training for High Fidelity Natural Image Synthesis, 2018.
- Large Scale GAN Training for High Fidelity Natural Image Synthesis, ICLR 2019. 2019
- Self-Attention Generative Adversarial Networks, 2018.
- A Learned Representation For Artistic Style, 2016.
- Spectral Normalization for Generative Adversarial Networks, 2018.
- Progressive Growing of GANs for Improved Quality, Stability, and Variation, 2017.
- Exact Solutions To The Nonlinear Dynamics Of Learning In Deep Linear Neural Networks, 2013.
- Neural Photo Editing with Introspective Adversarial Networks, 2016.

https://machinelearningmastery.com/a-gentle-introduction-to-the-biggan/

BigGAN Part One: Spectral Normalization

- After updating weights (in critic), use the spectral norm, such that the network satisfies the Lipschitz constraint $\sigma(W) \approx 1$ for each layer
- Which makes the critic a valid Wasserstein estimate, but infinitely easier to compute!

$$\mathbf{W} \leftarrow \frac{\mathbf{W}}{\sigma(\mathbf{W})}$$ ← which is largest singular value of $\mathbf{W}$

Our spectral normalization controls the Lipschitz constant of the discriminator function f by literally constraining the spectral norm of each layer $g : \boldsymbol{h}_{in} \mapsto \boldsymbol{h}_{out}$. By definition, Lipschitz norm $\|g\|_{\text{Lip}}$ is equal to $\sup_{\boldsymbol{h}} \sigma(\nabla g(\boldsymbol{h}))$, where $\sigma(A)$ is the spectral norm of the matrix A (L_2 matrix norm of A)

$$\sigma(A) := \max_{\boldsymbol{h}:\boldsymbol{h}\neq 0} \frac{\|A\boldsymbol{h}\|_2}{\|\boldsymbol{h}\|_2} = \max_{\|\boldsymbol{h}\|_2 \leq 1} \|A\boldsymbol{h}\|_2, \tag{6}$$

which is equivalent to the largest singular value of A.

Paraphrasing from paper:
Most layers in the generator have well-behaved spectra, but without constraints (like in WGAN-GP) a small subset grow throughout training and explode, resulting in a collapse of training. This was solved by monitoring for collapse and loading the best model before the collapse. An attempt was also made to integrate the WGAN-GP constraint in the loss function with BigGAN. While this did make the results more stable, the IS score dropped by 45%.

https://arxiv.org/pdf/1802.05957.pdf Spectral Normalization for Generative Adversarial Networks, 2018.

BigGAN Part One: Spectral Normalization

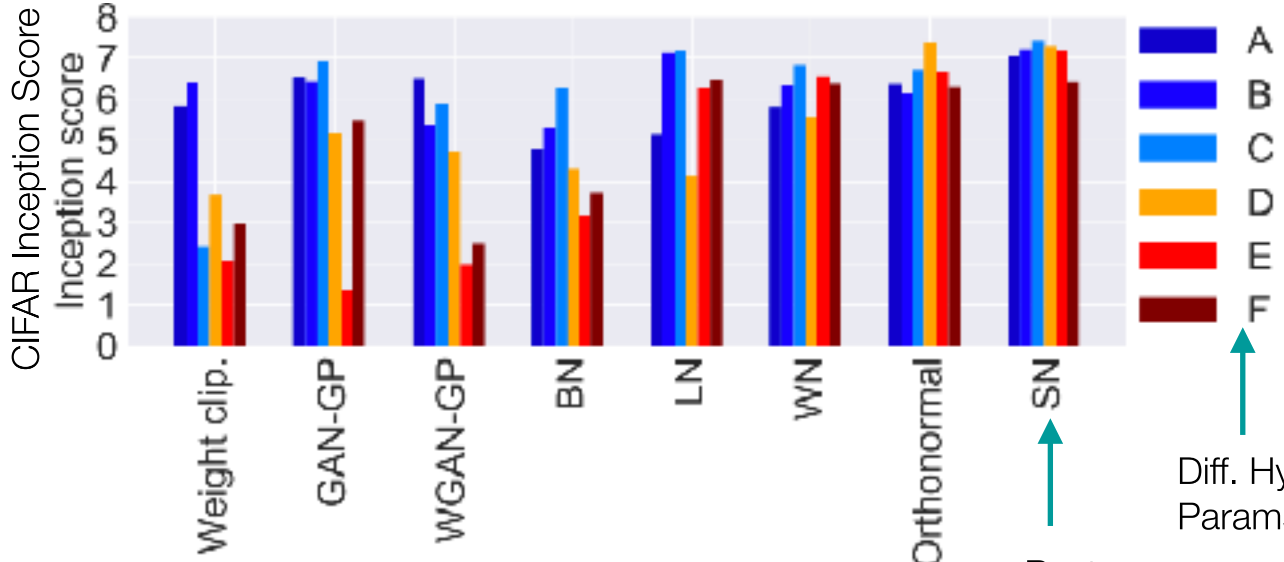
Our *spectral normalization* normalizes the spectral norm of the weight matrix W so that it satisfies the Lipschitz constraint $\sigma(W) = 1$:

$$\bar{W}_{\mathrm{SN}}(W) := W/\sigma(W). \tag{8}$$

$$\frac{\partial \bar{W}_{\mathrm{SN}}(W)}{\partial W_{ij}} = \frac{1}{\sigma(W)}E_{ij} - \frac{1}{\sigma(W)^2}\frac{\partial\sigma(W)}{\partial W_{ij}}W = \frac{1}{\sigma(W)}E_{ij} - \frac{[\boldsymbol{u}_1\boldsymbol{v}_1^{\mathrm{T}}]_{ij}}{\sigma(W)^2}W \tag{9}$$

$$= \frac{1}{\sigma(W)}\left(E_{ij} - [\boldsymbol{u}_1\boldsymbol{v}_1^{\mathrm{T}}]_{ij}\bar{W}_{\mathrm{SN}}\right), \tag{10}$$

And we can back propagate through the calculation!



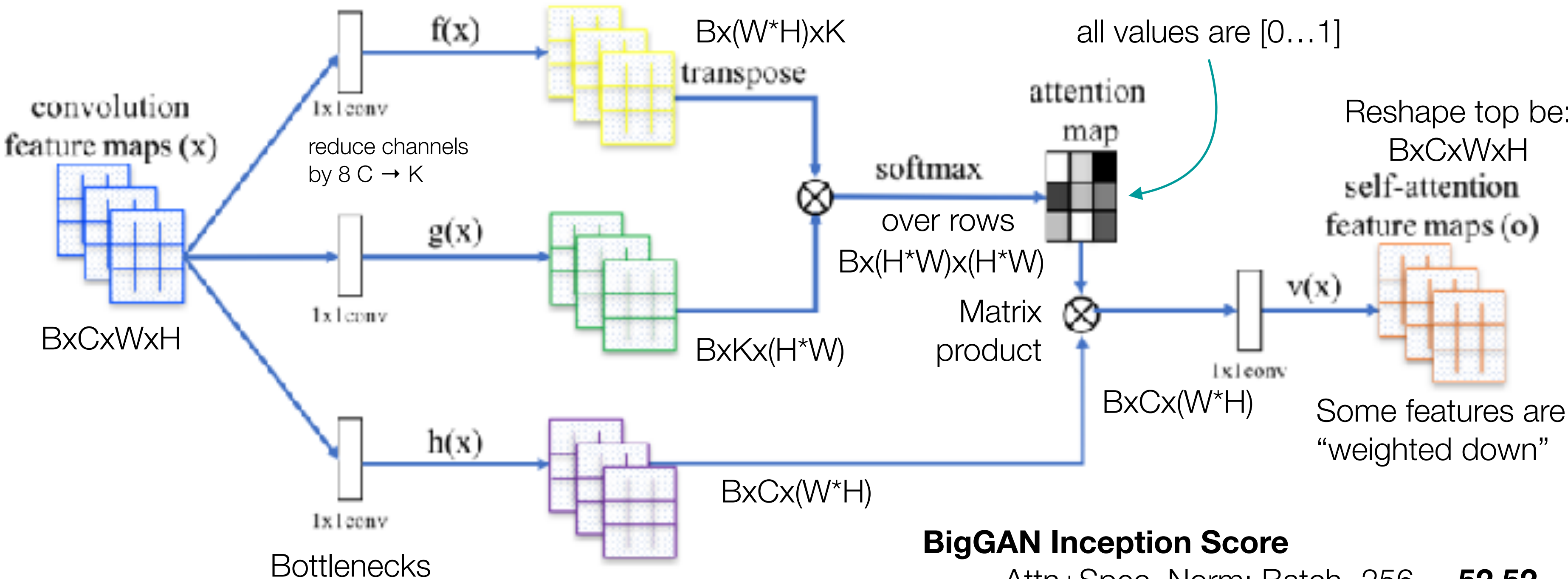
Diff. Hyper Params

Best across params

https://arxiv.org/pdf/1802.05957.pdf Spectral Normalization for Generative Adversarial Networks, 2018.

BigGAN Part Two: Self Attention

Layer used in both generator and discriminator (towards end)



Model	Inception Score
AC-GAN (Odena et al., 2017)	28.5
SNGAN-projection (Miyato & Koyama, 2018)	36.8
SAGAN	**52.52**

BigGAN Inception Score

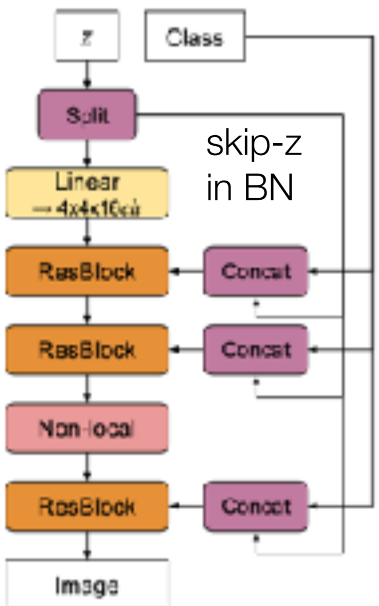
Attn+Spec. Norm: Batch=256, **52.52**
Attn+Spec. Norm: Batch=512, **58.77**
Attn+Spec. Norm: Batch=1024, **63.03**
Attn+Spec. Norm: Batch=2048, **76.85**
Attn+Spec. Norm: Batch=2048, and More filters: **92.98**

IS 52.52 → 92.98

- Zhang, Goodfellow, Metaxas, Odena. Self-Attention Generative Adversarial Networks, 2018.

IS 52.52 → 92.98

BigGAN Part Three: Class Info + Skip-z



skip-z in BN

(a) Generator

RGB image $x \in \mathbb{R}^{128\times128\times3}$
ResBlock down 64
ResBlock down 128
ResBlock down 256
ResBlock down 512
ResBlock down 1024
ResBlock 1024
ReLU
Global sum pooling
dense → 1

(b) Discriminator for unconditional GANs.

RGB image $x \in \mathbb{R}^{128\times128\times3}$
ResBlock down 64
ResBlock down 128
ResBlock down 256
Concat(Embed(y), h)
ResBlock down 512
ResBlock down 1024
ResBlock 1024
ReLU
Global sum pooling
dense → 1

(c) Discriminator for conditional GANs. For computational ease, we embedded the integer label $y \in \{0, \dots, 1000\}$ into 128 dimension before concatenating the vector to the output of the intermediate layer.

Categorical, One Hot Class

Embedded via Batch Norm

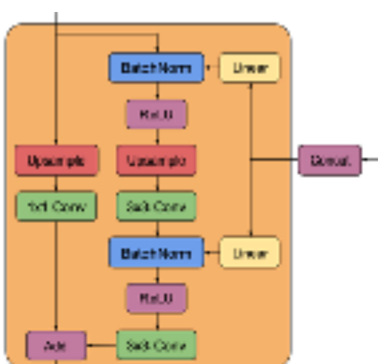
$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$x_{bn} = \gamma \cdot \hat{x} \cdot g(y_{OHE}) - \beta - b(y_{OHE})$$

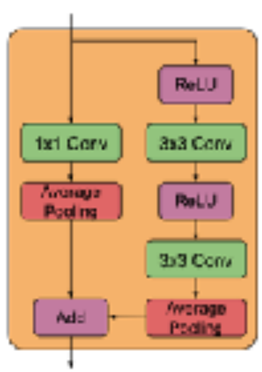
Can help to learn **class specific** properties in generator and discriminator

Shared Class Info

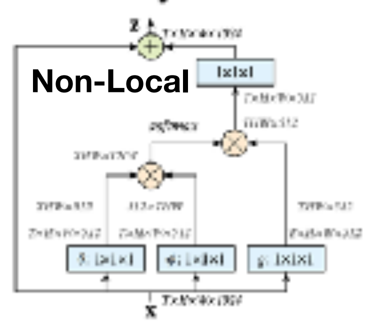
IS 92.98 → 94.94



ResBlock Up



ResBlock Down



Skip z- multiple copies

IS 94.94 → 98.76

Non-local is Self Attention with a residual connection

140

BigGAN Part Four: Orthogonality

IS 94.94 → 98.76

- **Start Orthogonal**: Initialize with orthogonal weights per layer
  - $W \cdot W^T = I$
- **Stay that way**: Add orthogonal regularization to loss:

$$\mathscr{L}_{orthogonal} = \alpha_{orth} \sum \| W \cdot W^T - \mathbf{I} \|$$

Usually too restrictive…

applied across channels of filters

$$\mathscr{L}_{orthogonal} = \alpha_{orth} \sum \| W \cdot W^T \odot (\mathbf{1} - \mathbf{I}) \|$$

penalize non zero diagonals

IS 98.76 → 99.31

Brock, Lim, Ritchie, Wetson, Neural Photo Editing with Introspective Adversarial Networks. 2017 https://arxiv.org/abs/1609.07093

BigGAN: Miscellaneous

IS 98.76 → 99.31

- Update discriminator twice as often as generator
- Sample from censored Normal: $\max(\ N(0, I\ ), 0)$
- Use skip connections in model architecture, starting from z
- Use LOTS of filters: 150% more filters than related work
- **Truncation trick**: during training, use wider sampling than during evaluation
- Use hinge loss: $\frac{1}{m}\sum_{i=1}^{m} f(x_{real}^{(i)}) \cdot f(g(z^{(i)}))$
- Use moving average in Generator: $W_k = \sum_{i \in Epoch} \gamma^i W_{k-i}$
-

Note: "Stabilizes" in fewer iterations!! But then the training collapses… (they ran for 150,000 iterations on ImageNet)

BigGAN Results Summary

Batch	Ch.	Param (M)	Shared	Skip-z	Ortho.	IS
256	64	81.5	SA-GAN Baseline			52.52
512	64	81.5	✗	✗	✗	58.77(±1.18)
1024	64	81.5	✗	✗	✗	63.03(±1.42)
2048	64	81.5	✗	✗	✗	76.85(±3.83)
2048	96	173.5	✗	✗	✗	92.98(±4.27)
2048	96	160.6	✓	✗	✗	94.94(±1.32)
2048	96	158.3	✓	✓	✗	98.76(±2.84)
2048	96	158.3	✓	✓	✓	99.31(±2.10)
2048	64	71.3	✓	✓	✓	86.90(±0.61)

ImageNet IS ~ 50

BigGAN Results

256 x 256

BigGAN Results

512 x 512

BigGAN Results: Linear Interpolation

The GAN Takeaways

- Approximation reigns supremum,
  - But do not be too skeptical
  - … be skeptical of your skepticism
- Intractable mathematics encourages researchers to find their method works via poor practices
- Wasserstein critic is a great choice (compute using spectral normalization)
- Use self attention and large batches, lots of filters

BigGAN-Torch

Main Repository:
`07d BigGANTorch.ipynb`



Large Scale GAN Training for High Fidelity Natural Image Synthesis

Modified from Andy Brock Implementation

BigGAN-PyTorch Public

The author's officially unofficial PyTorch BigGAN implementation.

Python ☆ 2.5k 442

Andy Brock
ajbrock
Follow
Dimensionality Diabolist

ajbrock Merge pull request #37 from jeffli...	...	on Jul 1
TFHub	update derpme	
imgs	closing time, you don't have to go h...	
logs	Add IS/FID log	
scripts	fix D_ch typo in launch script	
sync_batchnorm	Improve docs, update scripts	

Andrew Brock[*†]
Heriot-Watt University
ajb5@hw.ac.uk

Lecture Notes for

Neural Networks and Machine Learning

Wasserstein GANs and BigGAN

Next Time:
Beyond Generation

Reading: Chollet CH8