Deep Learning

Loo Tiang Kuan, Leonard

A0098795B

supervised by
Associate Professor Ji Hui

co-supervised by
Assistant Professor Alvina Goh Siew Wee

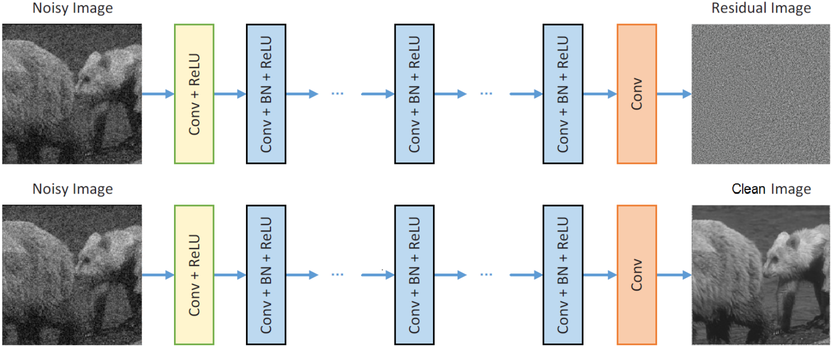
May 5, 2017

Today's Agenda

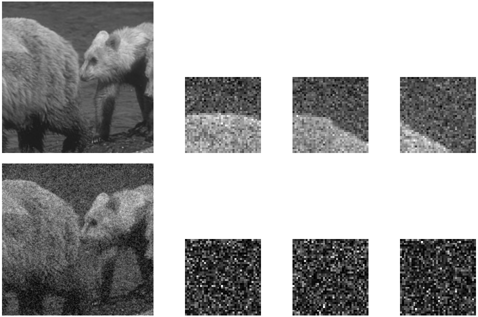
1. Denoising
2. Autoencoders
3. Generative Adversarial Networks
4. Improving Discriminator
5. Summary & Interesting Applications

Outline

Architecture



Training



- Motivation from ResNet (macro scale only)
- Can feed image of any size ('same' padding throughout).
- Heavy data augmentation

Experiments (Trained only with sigma=10)

sigma = 5	sigma = 10	sigma = 15
		

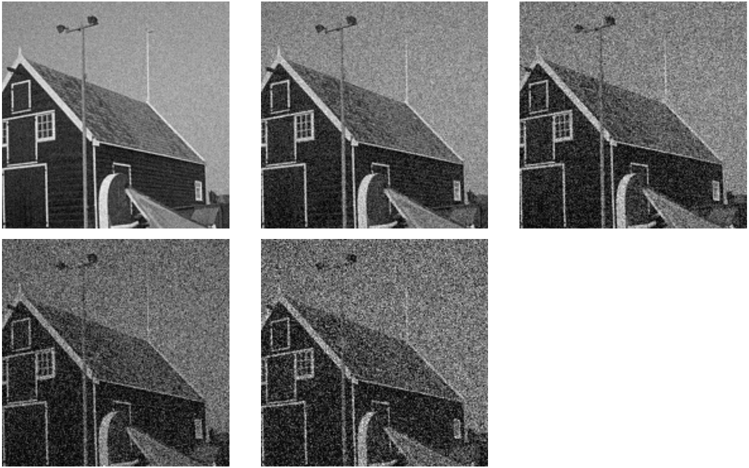
Top Row: Noisy

Bottom Row: Denoised

Outline

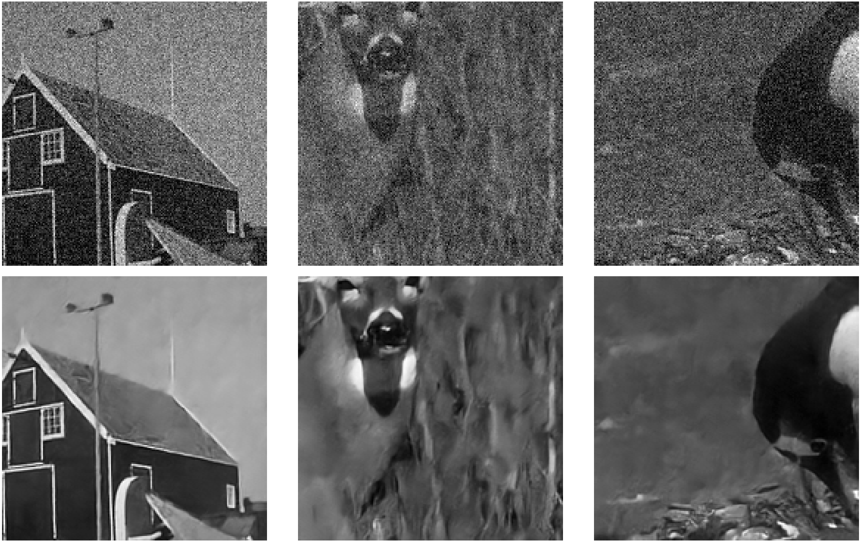
Blind Denoising



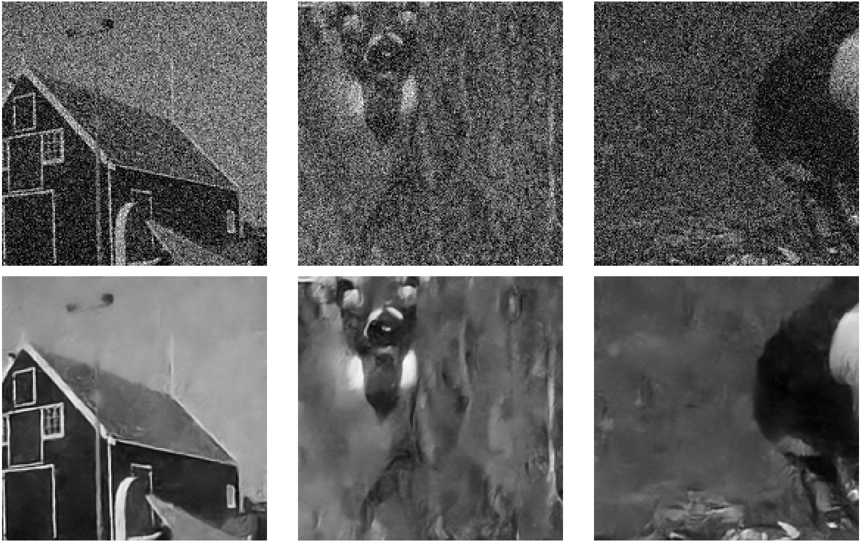
- Train with varying variance $\implies$ blind denoising on range of the variances trained. $\sigma \in [5, 50]$

Sigma=10

Sigma=30



Sigma=50



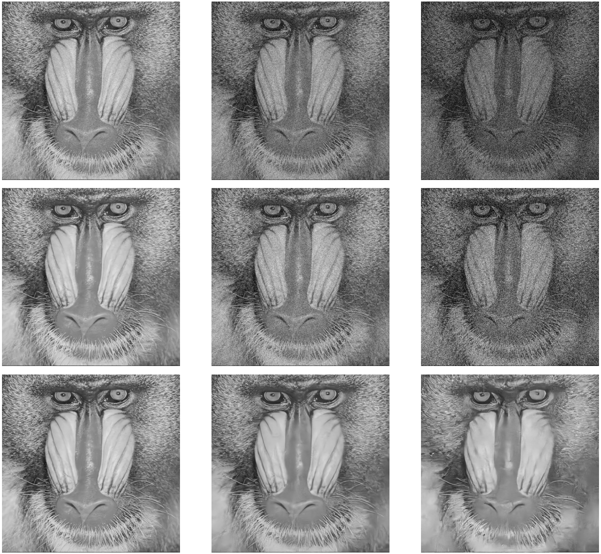
Comparison

Sigma=10 Sigma=25 Sigma=50

Noisy

Sigma 10 Denoising

Blind Denoising



Comparison

Sigma=10 Sigma=25 Sigma=50

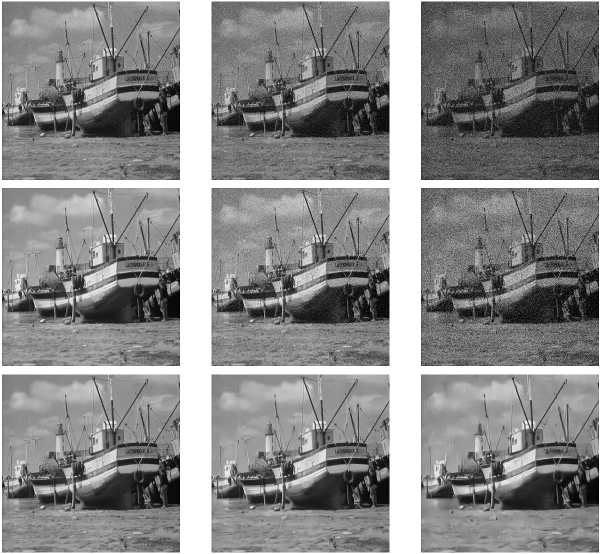
Noisy

Sigma 10 Denoising

Blind Denoising

Comparison

Sigma=10 | Sigma=25 | Sigma=50



Noisy

Sigma 10 Denoising

Blind Denoising

Comparison

Sigma=10 Sigma=25 Sigma=50

Noisy

Sigma 10 Denoising

Blind Denoising

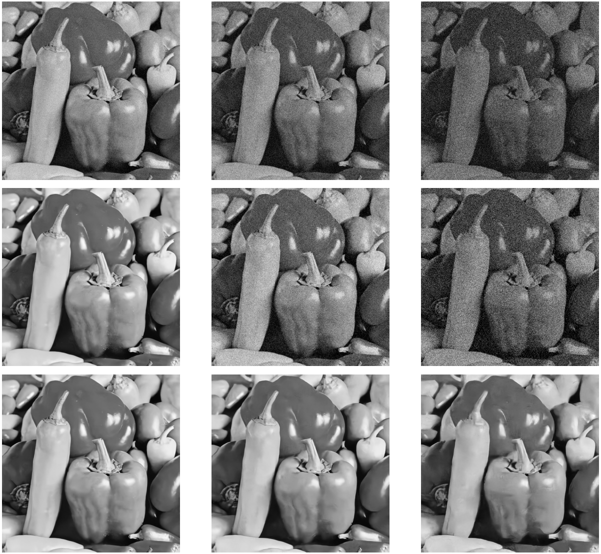
Comparison

Sigma=10 | Sigma=25 | Sigma=50

Noisy

Sigma 10 Denoising

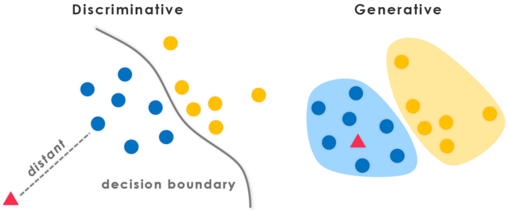
Blind Denoising



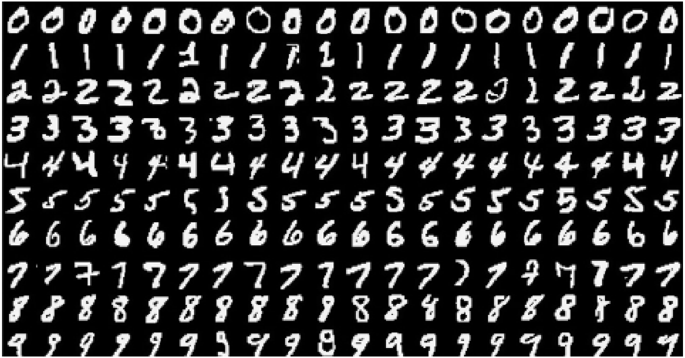
References

Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2016). *"Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising."* arXiv preprint arXiv:1608.03981.

Generative vs Discriminative



- Discriminative Approach
  - Directly finds properties in data.
  - Conditional probabilistic approach $\sim$ P(class|data)
- Generative Approach
  - Cares about how the data is generated.
  - Find properties based on the generation.
  - Joint probabilistic approach $\sim$ P(class,data)

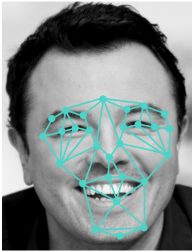


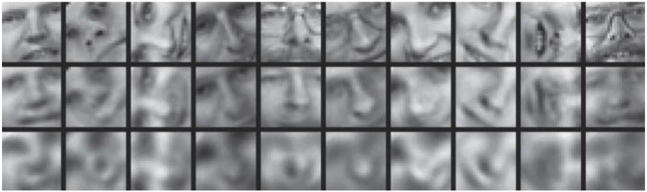
(a) Actual MNIST

(b) Real or Fake?

Outline

Dimensionality Reduction



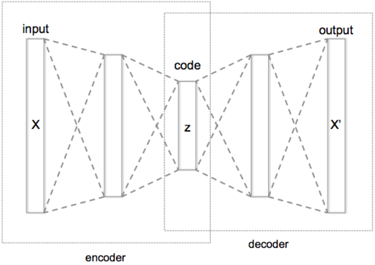


- Extracting emotions from a face.
  - Can reduce to problem of finding relationship with facial features (distance, curvature, etc).
  - Manual selection is difficult.
- PCA is linear $\implies$ limitations in kinds of features extracted
  - top: Original
  - middle: Autoencoder
  - bottom: PCA

Traditional Autoencoders

Aim: $I(X) \approx \psi(\phi(X))$.

- Unsupervised Learning Model
  - Let the network learn by itself.
  - Encoder: $\phi(X)$
  - Decoder: $\psi(z)$
    - Learn to generate X by input a smaller dimension z.



Traditional Autoencoders

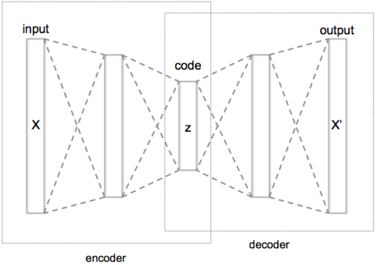
Aim: $I(x) \approx F(x) = \psi(\phi(x))$.

Cost: $C = C(x, F(x))$

Problem:
With a strong encoding and decoding neural network, F(x) may overfit, without capturing useful properties.

Regularised Autoencoders

Aim: $I(x) \approx F(x) = \psi(\phi(x))$.
Cost: $C = C(x, F(x))$



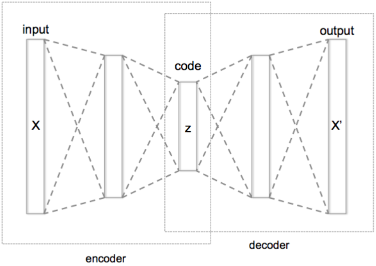
- Regularised Autoencoder
  - Allow model to learn useful features.
  - Impose penalties. (Sparse/Contractive autoencoder)
  - Augmentation. (Denoising autoencoder)

Outline

Outline

Sparse Autoencoders

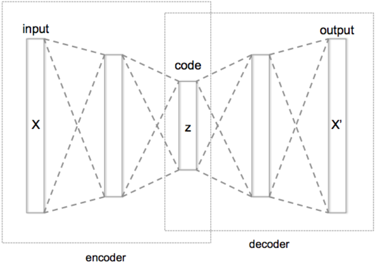
$F(x) = \psi(\phi(x))$



- Impose penalty to encourage sparseness on code $z = \phi(x)$
- $C(x, F(x) + \lambda \|\phi(x)\|)$

Outline

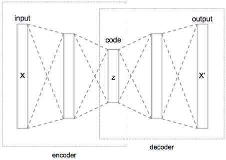
Contractive Autoencoders



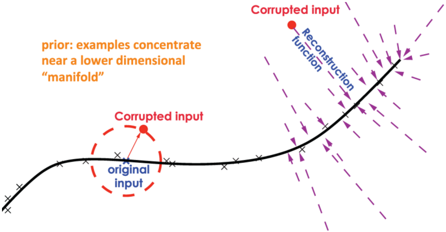
- Impose penalty instead, on derivative of z w.r.t. x.
- $C(x, F(x) + \lambda \left\| \nabla \phi(x) \right\|)$
- Force model to learn encoder that changes less when x deviates slightly.
    - Locally smooth about the training examples.

Outline

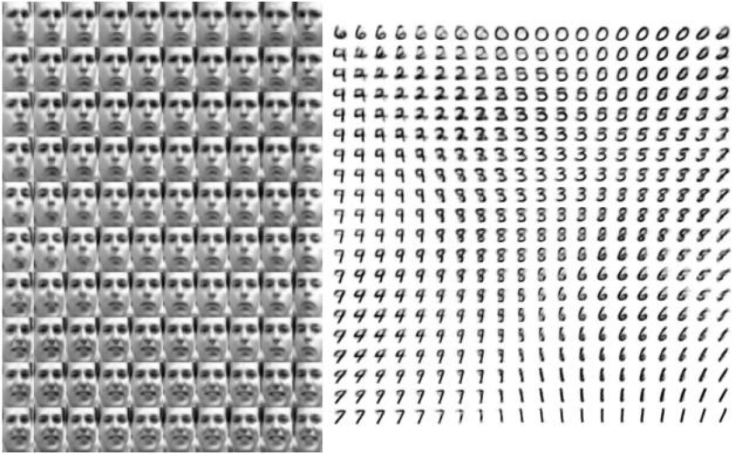
Denoising Autoencoders



- Augment input x by some noise ϵ.
- $C(x, F(x+\epsilon))$



2D Manifold of MNIST Digits



2D-Manifold

y-axis: Emotions   Digits

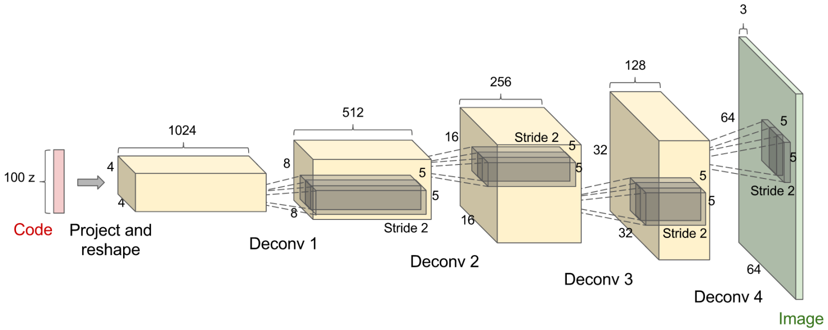
x-axis: Rotation

References

- Buduma, N. *"The curse of dimensionality and the autoencoder."* http://nikhilbuduma.com/2015/03/10/the-curse-of-dimensionality
- Bengio, Y., Goodfellow, I. J., & Courville, A. (2015). *"Deep learning."*
- Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. *"Extracting and composing robust features with denoising autoencoders."* In ICML, 2008.
- Diederik P Kingma and Max Welling. *"Auto-Encoding Variational Bayes"* ICLR, 2014.
- Doersch, C. (2016). *"Tutorial on variational autoencoders. "* arXiv preprint arXiv:1606.05908.

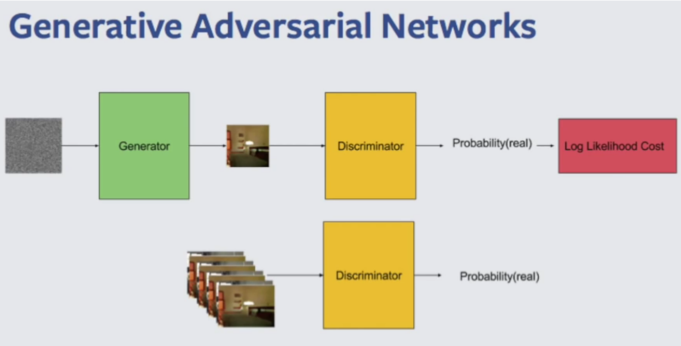
Outline

Constructing Loss



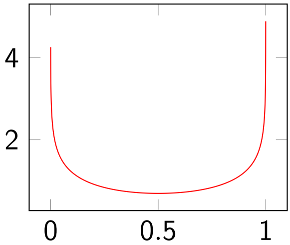
- Want generated images to look real.
- How to construct loss function?

Generative Adversarial Network (GAN)



Estimating generative models via an adversarial process

- Generative model G
  - Aims to have D classify wrongly.
  - Learns data distribution.
- Discriminative model D
  - Learn input is real or fake data → binary problem.
  - Unable to distinguish real or fake at the end.



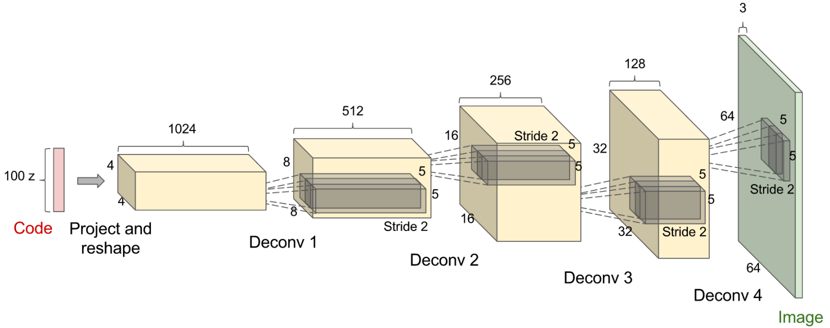
$$C(\mathbf{x}, \mathbf{w}, \mathbf{b}) = -\left[\left(y \log a^L\right) + (1-y)\log\left(1-a^L\right)\right]$$

Generative Adversarial Network (GAN)

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_X(x)}[\log D(x)] + \mathbb{E}_{z \sim P_Z(z)}[\log(1 - D(G(z)))]$$

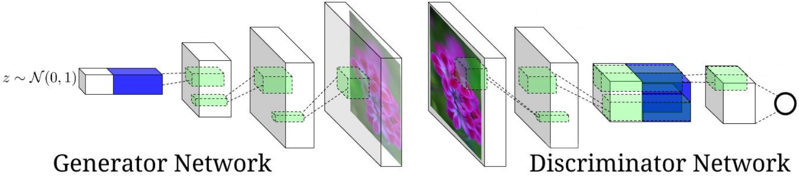
- $D(x)$
  - $D(x) \approx 1 \implies x$ is likely to be data.
  - $D(x) \approx 0 \implies x$ is likely to be artificial.
- $G(z)$
  - With code z, generates image $G(z)$ that attempts to look like X.
- $V(D, G)$
  - If quadratic:
    $V(D, G) = \mathbb{E}_{x \sim P_X(x)}\left[-\|D(x) - 1\|_2^2\right] + \mathbb{E}_{z \sim P_Z(z)}\left[-\|D(G(z))\|_2^2\right]$
    - Experimentally, output looks blurry

Generator



100 z
Code
Project and reshape
1024
4
4
512
8
8
5
5
Stride 2
Deconv 1
256
16
16
Stride 2
5
5
Deconv 2
128
32
32
5
5
Stride 2
Deconv 3
3
64
64
5
5
Stride 2
Deconv 4
Image

Generator & Discriminator

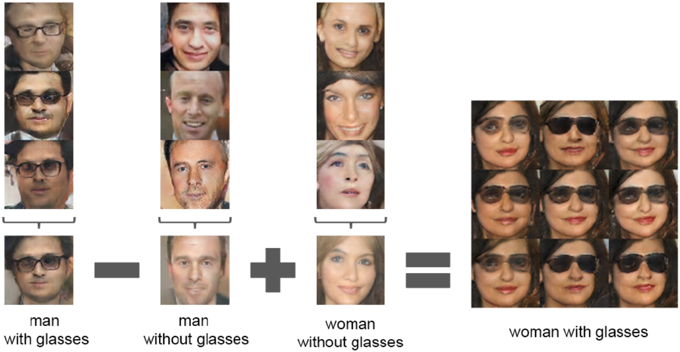


- Hybrid of Dense and Convolutional layers

Autoencoder vs GAN

- Autoencoder: Aim to get pixel-perfect reconstruction
- GAN: Aim to construct believable images that fit the distribution

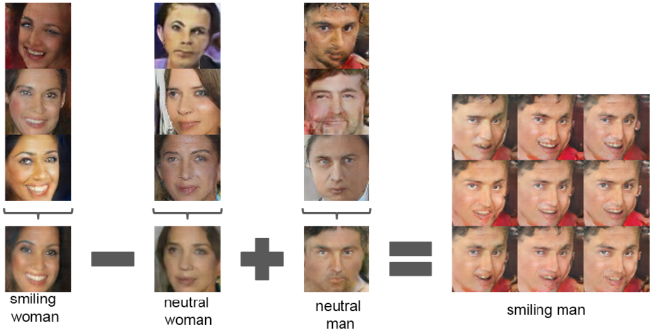
Vector arithmetic for visual concepts.



Locally smooth generator property -
perturb mean by uniform noise $\sim U(-0.25, 0.25)$

Generative Adversarial Network (DCGAN)

Vector arithmetic for visual concepts.



Locally smooth generator property -
perturb mean by uniform noise $\sim U(-0.25, 0.25)$

Interpolation.

Figure 8: A "turn" vector was created from four averaged samples of faces looking left vs looking right. By adding interpolations along this axis to random samples we were able to reliably transform their pose.

References

[GAN] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). *"Generative adversarial nets."* In Advances in Neural Information Processing Systems (pp. 2672-2680).

[LAPGAN] Denton, Emily L., Soumith Chintala, and Rob Fergus. *"Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks."* Advances in neural information processing systems. 2015.

[DCGAN] Radford, A., Metz, L., & Chintala, S. (2015). *"Unsupervised representation learning with deep convolutional generative adversarial networks. "* arXiv preprint arXiv:1511.06434.

"From Facebook AI research - Soumith Chintala - Adversarial Networks" (Youtube Link)

Outline

Experiment on MNIST

Epoch 1

Epoch 2

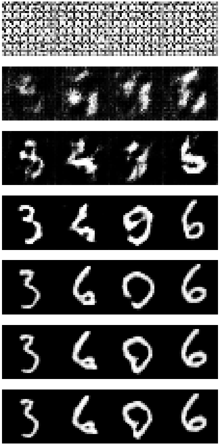
Epoch 3

Epoch 10

Epoch 50

Epoch 85

Epoch 100



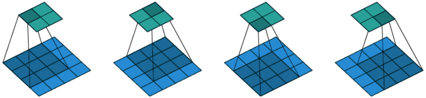
Outline

Outline

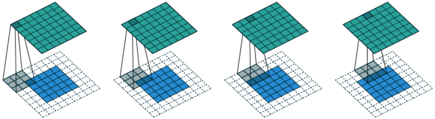
Typical Convolution: Padding

For a $m \times m$ filter f, consider $A * f$

- Valid
  - No padding.



- Full
  - Pads A by $m-1 \times m-1$.

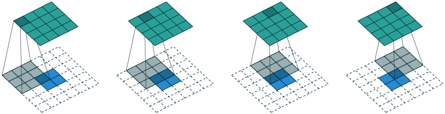


- Same
  - Pads A by $\frac{m-1}{2} \times \frac{m-1}{2}$ to get same output size.

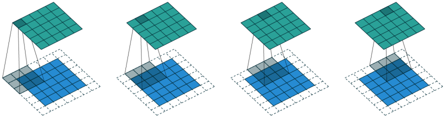
Transpose Convolution: Padding

Visualisation: Equivalent to its "dual" original convolution

- 'Valid' Tranposed Conv on 2×2 input
  - Use 'Full' with $*$ operation



- 'Same' Tranposed Conv with 3×3 filter
  - Use 'Same' with $*$ operation

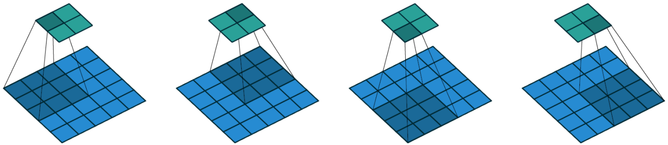


Outline

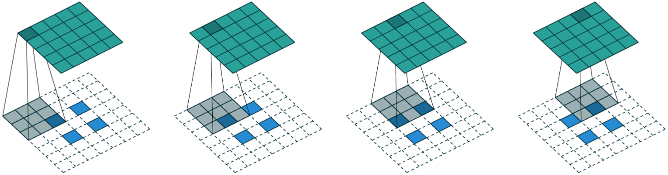
Transposed Convolution: Strides

Insert zeros in between pixels (Dilation)

- Strides s: 2×2



- Transposed Conv Strides s': 1×1
- Insert: 1×1



Summary

Convolution described by

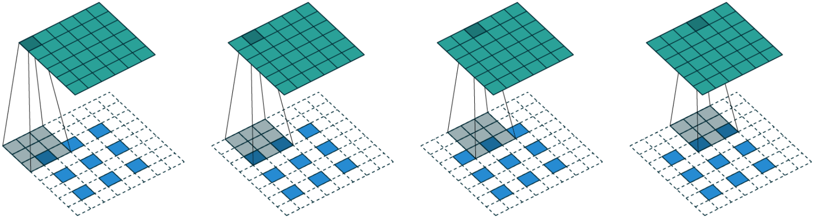
- filter size: m
- stride: s
- padding: p

has associated Transposed Convolution using

- filter size: $m' = m$
- stride: $s' = 1$
- padding: $p' = m - p - 1$
- insert/dilation: $i' = s - 1$
- diag: $a' = (i + 2p - m) \mod s$

with output size: $o' = s'(S - 1) + a' + m' - 2p'$,
where the 'stretched size' S is after adding i' zeros between each pixel

Summary: Example



Backward	Transposed Conv
Input: 6×6	Input: 3×3
Filter: 3×3	Filter: 3×3
Stride: 2×2	Stride: 1×1
Padding: 1×1	Padding: 1×1
	Insert: 1×1
	Diag: 1×1

Visualisation

References

Dumoulin, V. and Visin, F., 2016. *"A guide to convolution arithmetic for deep learning."* arXiv preprint arXiv:1603.07285.

Outline

Fooling ConvNets

Error in classification with minor perturbation.

(a) Correct Classification (b) Perturbation (c) Wrong Classification

- Perturb image by a small magnitude of $\frac{\partial C(x;\theta)}{\partial x}$
  - Gradient Ascent: $\tilde{x} = x + \epsilon \frac{\partial C(x;\theta)}{\partial x}$
- Note: Up until now, we are have been discussing $\frac{\partial C(x;\theta)}{\partial \theta}$ instead
  - Gradient Descent: $\theta_{k+1} = \theta_k - \eta \frac{\partial C(x;\theta_k)}{\partial \theta_k}$

Improving Discriminator

- Impose smooth local behaviour
  - Locally constant prior instead to perturb.
- Discriminator is more robust to perturbations
  - Assume different classes lies in disconnected manifolds.
  - Aim: $F(\hat{x}) = F(\hat{x} + \epsilon)$, for unlabelled $\hat{x}$ to lie locally in the same class.
  - Perturbations should not jump to other manifolds easily.
  - Better distinction of classes.
- To control $\left\|\tilde{x} - x\right\|_\infty$
  - Use $\operatorname{sign}\left(\frac{\partial C(x;\theta)}{\partial x}\right)$ instead to perturb.
- Redefine Cost function: $\widehat{C}(x;\theta) = \alpha C(x;\theta) + (1-\alpha) C(\tilde{x};\theta)$
- A form of data augmentation: increase training data set adaptively.

References

- Goodfellow IJ, Shlens J, Szegedy C. *"Explaining and harnessing adversarial examples."* arXiv preprint arXiv:1412.6572. 2014 Dec 20.
- Bengio, Y., Goodfellow, I.J. & Courville, A., 2015. *"Deep learning."* An MIT Press book in preparation.

Outline

Summary

- Neural Networks (Compositionality)
- Gradient Descent (Variants)
- Backpropagation
- Initialisation (Weights, Hyperparameters)
- Regularisation (Cost, Layers)
- Convolutional Neural Networks (Design, Architectures)
- Discriminators & Generators

Some Recent Progress

- Images
  - Image super-resolution
  - Object detection and classification
  - Detection and text-recognition
- Videos
  - Sharpening live videos
  - Fake Video
  - Modifying Trump's Hair from a video (26:00)
- Audio & Videos
  - Creating realistic-audio effects from silent video

Thank You