Synthesizing Images with Generative Adversarial Networks

Phillip Isola
OpenAI/Berkeley
10/26/17

CS194: Image Manipulation & Computational Photography
Alexei Efros, UC Berkeley, Fall 2017
Image classification

image X → Learner → “Fish”

label Y
Image classification

- image X
- Learner
- "Fish"
- label Y
Image classification

image X \rightarrow \text{Learner} \rightarrow \text{“Fish”}

label Y
Image classification

image X \rightarrow \text{Learner} \rightarrow \text{“Fish”}

label Y
Image prediction ("structured prediction")

Object labeling

[Long et al. 2015, …]

Edge Detection

[Xie et al. 2015, …]

Text-to-photo

“this small bird has a pink breast and crown…”

[Reed et al. 2014, …]

Style transfer

[Gatys et al. 2016, …]
Challenges

1. Output is high-dimensional, structured object

2. Uncertainty in mapping; many plausible outputs

3. Lack of supervised training data

“this small bird has a pink breast and crown…”
Training data

\[
\begin{align*}
&\{x, \text{"Fish"}, \} \\
&\{\text{"Grizzly"}, \} \\
&\{\text{"Chameleon"}, \} \\
&\vdots
\end{align*}
\]

\[
\arg\min_{F} \mathbb{E}_{x,y}[L(F(x), y)]
\]

Objective function (loss)

Neural Network
arg min \mathbb{E}_{x,y} [L(\mathcal{F}(x), y)]

“What should I do”  “How should I do it?”
Basic loss functions

Prediction: $\hat{y} = \mathcal{F}(x)$

Truth: $y$

Classification (cross-entropy):

$\mathcal{L}(\hat{y}, y) = -\sum_i \hat{y}_i \log y_i$

How many extra bits it takes to correct the predictions

Least-squares regression:

$\mathcal{L}(\hat{y}, y) = \|\hat{y} - y\|_2$

How far off we are in Euclidean distance
Training data:

\[
\begin{align*}
\mathcal{F} : & \mathbb{R}^n \rightarrow \mathbb{R}^m \\
\text{arg min} & \mathbb{E}_{x,y} \left[ L(\mathcal{F}(x), y) \right] \\
\text{Objective function (loss)} & \\
\text{Neural Network} & \\
\text{Color information: ab channels} & \\
\text{L channel} & \\
\end{align*}
\]
“rockfish”
Designing loss functions

\[ L_2(\hat{Y}, Y) = \frac{1}{2} \sum_{h,w} || Y_{h,w} - \hat{Y}_{h,w} ||^2 \]
\[ L_2(\hat{Y}, Y) = \frac{1}{2} \sum_{h,w} \| Y_{h,w} - \hat{Y}_{h,w} \|_2^2 \]
Designing loss functions

Color distribution cross-entropy loss with colorfulness enhancing term.

[Zhang, Isola, Efros, ECCV 2016]
Designing loss functions

Be careful what you wish for!
Designing loss functions

Image colorization

[Johnson, Alahi, Li, ECCV 2016]

L2 regression

Super-resolution

[Zhang, Isola, Efros, ECCV 2016]

L2 regression

[Johnson, Alahi, Li, ECCV 2016]
Designing loss functions

Image colorization

Cross entropy objective, with colorfulness term

[Zhang, Isola, Efros, ECCV 2016]

Super-resolution

Deep feature covariance matching objective

[Johnson, Alahi, Li, ECCV 2016]
Universal loss?
“Generative Adversarial Network” (GANs)

Generated vs Real (classifier)

[Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, Bengio 2014]
Conditional GANs

[Goodfellow et al., 2014]
[Isola et al., 2017]
Generator $G$ and $G'(x)$ according to Goodfellow et al., 2014.
\( \mathbf{G} \) tries to synthesize fake images that fool \( \mathbf{D} \)

\( \mathbf{D} \) tries to identify the fakes

[Goodfellow et al., 2014]
\[
\text{arg max}_D \mathbb{E}_{x,y} \left[ \log D(G(x)) + \log(1 - D(y)) \right]
\]

[Goodfellow et al., 2014]
\( G \) tries to synthesize fake images that fool \( D \):

\[
\arg\min_{G} \mathbb{E}_{x,y} \left[ \log D(G(x)) + \log(1 - D(y)) \right]
\]

[Goodfellow et al., 2014]
G tries to synthesize fake images that fool the best D:

$$\arg\min_G \arg\max_D \mathbb{E}_{x,y} \left[ \log D(G(x)) + \log(1 - D(y)) \right]$$

[Goodfellow et al., 2014]
G’s perspective: $D$ is a loss function.

Rather than being hand-designed, it is learned.
$$\arg\min_G \max_D \mathbb{E}_{x,y} \left[ \log D(G(x)) + \log(1 - D(y)) \right]$$

[Goodfellow et al., 2014]
$\arg \min_G \max_D \mathbb{E}_{x,y} \left[ \log D(G(x)) + \log(1 - D(y)) \right]$
\[ \arg \min_G \max_D \mathbb{E}_{x,y} \left[ \log D(G(x)) + \log(1 - D(y)) \right] \]

[Goodfellow et al., 2014]
[Isola et al., 2017]
\[
\arg \min_G \max_D \mathbb{E}_{x, y} \left[ \log D(x, G(x)) + \log(1 - D(x, y)) \right]
\]

[Goodfellow et al., 2014]
[Isola et al., 2017]
\[
\arg \min_G \max_D \mathbb{E}_{x,y} \left[ \log D(x, G(x)) + \log(1 - D(x, y)) \right]
\]

[Goodfellow et al., 2014]
[Isola et al., 2017]
\[
\begin{align*}
\arg \min_G \max_D & \quad \mathbb{E}_{x,y} \left[ \log D(x, G(x)) + \log(1 - D(x, y)) \right] \\
\text{[Goodfellow et al., 2014]} & \quad \text{[Isola et al., 2017]}
\end{align*}
\]
arg \min_G \max_D \mathbb{E}_{x,y} \left[ \log D(x, G(x)) + \log(1 - D(x, y)) \right]

[Goodfellow et al., 2014]
[Isola et al., 2017]
Data from [Russakovsky et al. 2015]
Edges → Images

Edges from [Xie & Tu, 2015]
Sketches → Images

Input → Output
Input → Output
Input → Output
Input → Output

Trained on Edges → Images

Data from [Eitz, Hays, Alexa, 2012]
#edges2cats [Chris Hesse]
Ivy Tasi @ivymyt

Vitaly Vidmirov @vvid
Hallucinations

Input

Output

Input

Output

Input

Output

Input

Output
Challenges —> Solutions

1. Output is high-dimensional, structured object
   —> Use a deep net, D, to analyze output!

2. Uncertainty in mapping; many plausible outputs
   —> D only cares about “plausibility”, doesn’t hedge

3. Lack of supervised training data
Challenges —> Solutions

1. Output is high-dimensional, structured object
   —> Use a deep net, D, to analyze output!

2. Uncertainty in mapping; many plausible outputs
   —> D only cares about “plausibility”, doesn’t hedge

3. Lack of supervised training data
Paired data

\[ x_i, y_i \]

\{ 
\begin{align*}
\text{[image of cat]} & , \\
\text{[image of cat]} & , \\
\text{[image of cat]} & , \\
\text{[image of cat]} & , \\
\cdots & .
\end{align*}
\}
Paired data

\[ x_i, y_i, \ldots \]

Unpaired data

\[ X, Y, \ldots \]
\[
\arg\min_G \max_D \mathbb{E}_{x,y} \left[ \log D(x, G(x)) + \log(1 - D(x, y)) \right]
\]
No input-output pairs!
x \rightarrow G \rightarrow G(x) \rightarrow D \rightarrow \text{real or fake?}

$$\arg \min_G \max_D \mathbb{E}_{x,y} \left[ \log D(G(x)) + \log(1 - D(y)) \right]$$

Usually loss functions check if output matches a target instance

GAN loss checks if output is part of an admissible set
\[ x \rightarrow G \rightarrow G(x) \rightarrow D \rightarrow \text{Real!} \]
Nothing to force output to correspond to input
Cycle-Consistent Adversarial Networks

... [Zhu et al. 2017], [Yi et al. 2017], [Kim et al. 2017]...
Cycle-Consistent Adversarial Networks
Cycle Consistency Loss

\[\left\| F(G(x)) - x \right\|_1\]
Cycle Consistency Loss

\[
\begin{align*}
\|F(G(x)) - x\|_1 & \quad \text{reconstruction error} \\
\|G(F(y)) - y\|_1 & \quad \text{reconstruction error}
\end{align*}
\]
Artistic style transfer

[Gatys et al. 2016, …]
Failure case
Applications of CycleGAN

\[ \text{MR} \rightarrow \text{CT} \quad \text{[Wolterink et al.]} \quad \text{arxiv: 1708.01155} \]

- **MRI reconstruction** [Quan et al.] arxiv:1709.00753
- **Cardiac MR images from CT** [Chartsias et al. 2017]
Latest from #CycleGAN

CycleGAN with architectural modifications, by itok_msi
https://qiita.com/itok_msi/items/b6b615bc28b1a720afd7
Challenges —> Solutions

1. Output is high-dimensional, structured object
   —> Use a deep net, D, to analyze output!

2. Uncertainty in mapping; many plausible outputs
   —> D only cares about “plausibility”, doesn’t hedge
   “this small bird has a pink breast and crown…”

3. Lack of supervised training data
   —> D doesn’t require paired training instances
Can we generate images from scratch?

$z \sim \mathcal{N}(0, 1)$

Gaussian noise

Synthesized image
\textbf{G} tries to synthesize fake images that fool \textbf{D}

\textbf{D} tries to identify the fakes

[Goodfellow et al., 2014]
GANs are implicit generative models

\[ p(x) \leftarrow \text{“generative model” of the data } x \]

Noise distribution \( z \sim \mathcal{N}(0, 1) \) \quad \text{GAN} \quad \text{Data distribution} \quad \text{GAN} \quad \text{Data distribution}

\( G(z) \sim p(x) \leftrightarrow \) Samples from GAN, at equilibrium, are samples from the data distribution
Randomly generated faces

[BEGAN: Berthelot et al., 2014]
Interpolation in z space

Input A  -----------  Interpolation from A to B  -----------  Input B
Pix2pix: 144 lines
CycleGAN: 220 lines