Convolutional Network for Image Synthesis

Jun-Yan Zhu

16-726 Learning-based Image Synthesis, Spring 2022

many slides from Alyosha Efros, Phillip Isola, Richard Zhang, James Hays, and Andrea Vedaldi, Jitendra Malik.

© Zhang et al., ECCV 2016
Review (data-driven graphics)
Review (data-driven graphics)

Nearest neighbor methods:
1. Stored examples
2. Calculate distance between two examples
3. Voting (label transfer): image blending/averaging
Visual similarity via labels

“Penguin”

? ==

“Penguin”
Machine Learning as data association

image $X$ \leftrightarrow \text{black box classifier} \leftrightarrow \text{“Penguin”} \leftrightarrow \text{label } Y
At test time...

image $X$
Examples from MNIST dataset [LeCun. 1998]
Warm-up Example: Binary Digit Classification

7 vs. 1
Learning Approach to Digit Recognition

• Collect Training Images
  • Positive: 2 2 7 7 2 2 2 7 7 7 7 7 7 7
  • Negative: /

• Training Time
  • Compute feature vectors for positive and negative example images
  • Train a classifier

• Test Time
  • Compute feature vector on new test image:
  • Evaluate classifier
Let us take an example…
Let us take an example...

Note that there are several ways to construct a feature vector. This is one example.
In feature space, positive and negative examples are just points...
How do we classify a new point?
Nearest neighbor rule
“transfer label of nearest example”
Linear classifier rule

\[ \langle w, x \rangle + b = 0 \] is the decision boundary

Learned at training time

Negative examples

Positive examples of digit 7
Basic idea

Brain/Machine → “clown fish”
Object recognition

Feature extractors: Edges, Texture, Colors

Classifier: Segments, Parts

"clown fish"
Neural network

Learned

“clown fish”
Neural network

Learned

“clown fish”
Deep neural network

Learned

“clown fish”
Computation in a neural net

Input representation

Output representation
Computation in a neural net

\[ y_j = \sum_i w_{ij} x_i \]

i: the \( i^{th} \) dimension of \( x \), j: the \( j^{th} \) dimension of \( y \)
Computation in a neural net

Rectified linear unit (ReLU)

\[ g(y) = \max(0, y) \]
Computation in a neural net

Rectified linear unit (ReLU)

\[ g(x) = \max(0, x) \]
Computation in a neural net

\[ y_j = \sum_i w_{ij} x_i \]

\[ z_k = \max_{j \in \mathcal{N}(j)} g(y_j) \]

i: the \( i^{th} \) dimension of \( x \), j; the \( j^{th} \) dimension of \( y \)
Computation in a neural net

Single depth slice

max pool with 2x2 filters and stride 2

224x224x64
pool

112x112x64

downsampling

224

112
Computation in a neural net

\[ f(x) = f_L(\ldots f_2(f_1(x))) \]
Convolutional Neural Nets
Convolutional Neural Nets

Convolution

Filter
[Hubel and Wiesel 59]

Slide from Andrea Vedaldi
Computation in a neural net

Last layer

- dolphin
- cat
- grizzly bear
- angel fish
- chameleon
- clown fish
- iguana
- elephant

\[
\text{argmax}
\]

“clown fish”
Learning with deep nets

Learned

“clown fish”
Learning with deep nets

- “clown fish”
- “grizzly bear”
- “chameleon”

Train network to associate the right label with each image
Learning with deep nets

"clown fish"

\[
\begin{align*}
\argmin_w L(w_1, \ldots, w_6) \\
\rightarrow \text{Loss} \rightarrow L() \\
\end{align*}
\]
Network output

- dolphin
- cat
- grizzly bear
- angel fish
- chameleon
- **clown fish**
- iguana
- elephant

Ground truth label

- "clown fish"

Loss $\rightarrow$ error
Loss function

Network output

- dolphin
- cat
- grizzly bear
- angel fish
- chameleon
- **clown fish**
- iguana
- elephant
- ...

Ground truth label

“clown fish”

Loss $\rightarrow$ **small**
Loss function

Network output

- dolphin
- cat
- grizzly bear
- angel fish
- chameleon
- **clown fish**
- iguana
- elephant

Ground truth label

- “grizzly bear”

Loss → **large**
Loss function for classification

Network output | Ground truth label

\[ H(\hat{z}, z) = - \sum_c z_c \log \hat{z}_c \]

Cross-entropy loss

c is the \( c^{th} \) class in the output
Learning with deep nets

\[ x^{(1)}, z^{(1)} \text{ is the input and label of the 1st training image} \]
Learning with deep nets

$x^{(2)}$ is the input and label of the 2nd training image

$z^{(2)}$, $z'$ is the input and label of the 2nd training image

$\ell(z^{(2)}, f(x^{(2)}, w))$
Learning with deep nets

\[ \text{Loss} \left( z^{(3)}, f(x^{(3)}, w) \right) \]

\[ \arg \min_w \sum_i \ell(z^{(i)}, f(x^{(i)}, w)) \]
Gradient descent

\[
\arg\min_w \sum_i \ell(z^{(i)}, f(x^{(i)}, w)) = \arg\min_w L(w)
\]

One iteration of gradient descent:

\[
w^{t+1} = w^t - \eta_t \frac{\partial L(w^t)}{\partial w}
\]

learning rate
Gradient descent

$L(w)$

$w_1$  $w_2$
$$p(c|x)$$

<table>
<thead>
<tr>
<th>mite</th>
<th>container ship</th>
<th>motor scooter</th>
<th>leopard</th>
</tr>
</thead>
<tbody>
<tr>
<td>mite</td>
<td>container ship</td>
<td>motor scooter</td>
<td>leopard</td>
</tr>
<tr>
<td>black widow</td>
<td>lifeboat</td>
<td>go-kart</td>
<td>jaguar</td>
</tr>
<tr>
<td>cockroach</td>
<td>amphibian</td>
<td>moped</td>
<td>cheetah</td>
</tr>
<tr>
<td>tick</td>
<td>fireboat</td>
<td>bumper car</td>
<td>snow leopard</td>
</tr>
<tr>
<td>starfish</td>
<td>drilling platform</td>
<td>golfcart</td>
<td>Egyptian cat</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>grille</th>
<th>mushroom</th>
<th>cherry</th>
<th>Madagascar cat</th>
</tr>
</thead>
<tbody>
<tr>
<td>convertible</td>
<td>agaric</td>
<td>dalmatian</td>
<td>squirrel monkey</td>
</tr>
<tr>
<td>grille</td>
<td>mushroom</td>
<td>grape</td>
<td>spider monkey</td>
</tr>
<tr>
<td>pickup</td>
<td>jelly fungus</td>
<td>elderberry</td>
<td>titi</td>
</tr>
<tr>
<td>beach wagon</td>
<td>gill fungus</td>
<td>currant</td>
<td>indri</td>
</tr>
<tr>
<td>fire engine</td>
<td>dead-man's-fingers</td>
<td>offordshire</td>
<td>howler monkey</td>
</tr>
</tbody>
</table>

[Krizhevsky et al. NIPS 2012]
Computer Vision before 2012

Features → Clustering → Pooling → Classification → Cat
Computer Vision Now

Deep Learning for Computer Vision

Top 5 accuracy on ImageNet benchmark

Object detection

Human understanding

Autonomous driving
Can Deep Learning Help Graphics?

Cat → Modeling → Texturing → Lighting → Rendering
Can Deep Learning Help Graphics?

Cat ➔ Modeling ➔ Texturing ➔ Lighting ➔ Rendering ➔ Cat

Deep Net

Did not work

Can Deep Learning Help Graphics?
Generating images is hard!

Cat ➔ Modeling ➔ Texturing ➔ Lighting ➔ Rendering

8 ➔ Deep Net ➔ 28x28 pixels
from Classification to Generation
Predicting the color value of an output pixel given a patch
Discriminative Deep Networks

“Rockfish”
Discriminative Deep Networks

Raw, Unlabeled Pixels
Generative Deep Networks

Raw, Unlabeled Pixels
Better Architectures
Fractionally-strided Convolution

Regular conv  Fractiaionally-strided conv
Generating chairs conditional on chair ID, viewpoint, and transformation parameters

Dosovitskiy et al. Learning to Generate Chairs, Tables and Cars with Convolutional Networks
PAMI 2017 (CVPR 2015)
With Varying Viewpoints

Dosovitskiy et al. Learning to Generate Chairs, Tables and Cars with Convolutional Networks
PAMI 2017 (CVPR 2015)
With Varying Transformation Parameters

Dosovitskiy et al. Learning to Generate Chairs, Tables and Cars with Convolutional Networks
PAMI 2017, (CVPR 2015)
Interpolation between Two Chairs

Dosovitskiy et al. Learning to Generate Chairs, Tables and Cars with Convolutional Networks
PAMI 2017 (CVPR 2015)
Better Loss Functions
Ansel Adams. *Yosemite Valley Bridge.*
Grayscale image: $L$ channel

$X \in \mathbb{R}^{H \times W \times 1}$

$F$

Color information: $ab$ channels

$\hat{Y} \in \mathbb{R}^{H \times W \times 2}$

Concatenate $(L, ab)$ channels

Grayscale image: $L$ channel

$X \in \mathbb{R}^{H \times W \times 1}$

$\mathcal{F}$

Concatenate $(L, ab)$ channels

$(X, \hat{Y})$

Simple L2 regression doesn’t work 😞

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

\[ \theta^* = \arg \min_{\theta} \ell(\mathcal{F}_\theta(X), Y) \]

- Regression with L2 loss inadequate
  \[ L_2(\hat{Y}, Y) = \frac{1}{2} \sum_{h,w} \| Y_{h,w} - \hat{Y}_{h,w} \|^2 \]

- Use per-pixel multinomial classification
  \[ L(\hat{Z}, Z) = -\frac{1}{HW} \sum_{h,w} \sum_{q} Z_{h,w,q} \log(\hat{Z}_{h,w,q}) \]
Designing loss functions

Input

Zhang et al. 2016

Ground truth

Color distribution cross-entropy loss with colorfulness enhancing term.

[Zhang, Isola, Efros, ECCV 2016]
Thank You!

16-726, Spring 2022
https://learning-image-synthesis.github.io/sp22/