Deep Learning in Comp. Photo

With slides from Phil Isola, James Hays, and Andrea Vedaldi

CS194: Image Manipulation & Computational Photography
Alexei Efros, UC Berkeley, Fall 2018
Visual similarity via labels

“Penguin”

==

“Penguin”
Machine Learning as data association

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

image $X$
Quick Background on Supervised Learning

Jitendra Malik
Fig. 4. Size-normalized examples from the MNIST database.
Warm-up Example: Binary Digit Classification

7 vs. 1
Learning Approach to Object Recognition

- Collect Training Images
  - Positive: 
  - 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...
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
Basic idea

Brain/Machine

“clown fish”
Object recognition

Feature extractors

Classifier

Edges
Texture
Colors

Segments
Parts

“clown fish”
Object recognition

Feature extractors

Edges
Texture
Colors

Classifier

Segments
Parts

“clown fish”

Learned
Neural network

"clown fish"
Neural network

[Image of a clown fish]

Learned

→  

→  

→ “clown fish”
Deep neural network

Learned

“clown fish”
Computation in a neural net

Input representation

Output representation
Computation in a neural net

$y_j = \sum_i w_{ij} x_i$
Computation in a neural net

Rectified linear unit (ReLU)

\[ g(y) = \max(0, y) \]
Convolutional Neural Nets

input features → a bank of 2 filters → 2-dimensional output features

Slide from Andrea Vedaldi
Convolutional Neural Nets

Convolution

filter
Computation in a neural net

\[ f(x) = f_L(...f_2(f_1(x))) \]
Computation in a neural net

Last layer

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

argmax

“clown fish”
Learning with deep nets
Learning with deep nets

Train network to associate the right label with each image:

- "clown fish"
- "grizzly bear"
- "chameleon"
Learning with deep nets

“clown fish”

\[
\begin{align*}
\text{argmin}_w & \quad L(w_1, \ldots, w_6)\\
\text{Loss} & \quad \rightarrow \quad L() 
\end{align*}
\]
Loss function

Network output

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

Ground truth label

- “clown fish”

Loss → error
Loss function

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

Ground truth label:
- "clown fish"

Loss → 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
\[ \hat{z} \]

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

Ground truth label
\[ z \]

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

Probability of the observed data under the model
\[ H(\hat{z}, z) = - \sum_c \hat{z}_c \log z_c \]

Results in learning a probability model \( p(c|x) \)!
Learning with deep nets

\[ \ell(z_1, f(x_1; w)) \]
Learning with deep nets

\[ z_2 \]

"grizzly bear"

\[ x_2 \]

\[ w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5 \quad w_6 \]

\[ \ell(z_2, f(x_2; w)) \]

\[ \text{Loss} \]

Learned
Learning with deep nets

\[ \text{argmin}_w \sum_i \ell(z_i, f(x_i; w)) \]

\[ \ell(z_i, f(x_i; w)) \]

Learning mechanism for deep nets:
- Input: \( x_i \)
- Output: \( z_i \)
- Weights: \( w_1, w_2, w_3, w_4, w_5, w_6 \)
- Loss function: \( \ell(z_i, f(x_i; w)) \)
Gradient descent

\[ \text{argmin}_w \sum_i \ell(z_i, f(x_i; w)) = L(w) \]

One iteration of gradient descent:

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

learning rate
Texture synthesis by non-parametric sampling

Synthesizing a pixel

Models \( P(p|N(p)) \)

[Efros & Leung 1999]
Texture synthesis with a deep net

[van der Oord et al. 2016]
Texture synthesis with a deep net

[van der Oord et al. 2016]
Sampling

Network output

... →

- turquoise
- blue
- green
- red
- orange
- white
- gray
- black

\[ P(p_i|p_1, \cdots, p_{i-1}) \]
Sampling

Network output

turquoise
blue
green
red
orange
white
gray
black

\[ P(p_i | p_1, \cdots, p_{i-1}) \]
Sampling

Network output

<table>
<thead>
<tr>
<th>Turquoise</th>
<th>Blue</th>
<th>Green</th>
<th>Red</th>
<th>Orange</th>
<th>White</th>
<th>Gray</th>
<th>Black</th>
</tr>
</thead>
</table>

\[ P(p_i | p_1, \cdots, p_{i-1}) \]
Sampling

Network output

\[ P(p_i | p_1, \cdots, p_{i-1}) \]

...
Sampling

Network output

\[ P(p_i|p_1, \cdots, p_{i-1}) \]
Sampling

Network output

\[ P(p_i | p_1, \cdots, p_{i-1}) \]
Sampling full images