CS4501: Introduction to Computer Vision

Introduction to Machine Learning

Various slides from previous courses by:
D.A. Forsyth (Berkeley / UIUC), I. Kokkinos (Ecole Centrale / UCL), S. Lazebnik (UNC / UIUC), S. Seitz (MSR / Facebook), J. Hays (Brown / Georgia Tech), A. Berg (Stony Brook / UNC), D. Samaras (Stony Brook), J. M. Frahm (UNC), V. Ordonez (UVA), Steve Seitz (UW).
So far

• Image Formation: Cameras, Projections.
• Image Processing: Filtering, Convolutions.
• Finding Edges, Corners, Interest Points, SIFT Features.
• Geometry: Camera Calibration, Stereo Matching, 3D Reconstruction.
• Video: Optical Flow.
Today’s Class

• Introduction to Machine Learning
  • Unsupervised Learning: Clustering (e.g. k-means clustering)
  • Supervised Learning: Classification (e.g. k-nearest neighbors, softmax classifier)
Machine Learning

• **Machine learning** is the subfield of computer science that gives "computers the ability to learn without being explicitly programmed."
  - term coined by Arthur Samuel 1959 while at IBM

• The study of algorithms that can learn from data.

• In contrast to previous Artificial Intelligence systems based on Logic, e.g. ”Expert Systems”
Supervised Learning vs Unsupervised Learning

\[ x \rightarrow y \]

\[ x \]
Supervised Learning vs Unsupervised Learning

- Supervised Learning: $x \rightarrow y$
  - Cat
  - Dog
  - Bear

- Unsupervised Learning: $x$
  - Cat
  - Dog
  - Bear
Supervised Learning vs Unsupervised Learning

**Classification**

\[ x \rightarrow y \]

- Cat
- Dog
- Bear

**Clustering**

\[ x \]

- Cat
- Dog
- Bear
Supervised Learning Examples

Classification → cat

Face Detection

The screen was a sea of red → Language Parsing

Structured Prediction
Supervised Learning Examples

\[ \text{cat} = f( ) \]

\[ = f( ) \]

\[ = f( \text{The screen was a sea of red} ) \]
Supervised Learning – k-Nearest Neighbors

k=3
Supervised Learning – k-Nearest Neighbors

k=3

cat, dog, dog

bear, dog, dog
Supervised Learning – k-Nearest Neighbors

• How do we choose the right K?
• How do we choose the right features?
• How do we choose the right distance metric?
Supervised Learning – k-Nearest Neighbors

• How do we choose the right K?
• How do we choose the right features?
• How do we choose the right distance metric?

**Answer:** Just choose the one combination that works best! **BUT** not on the test data.

Instead split the training data into a ”Training set” and a ”Validation set” (also called ”Development set”)

Unsupervised Learning – k-means clustering

1. Initially assign all images to a random cluster

$k = 3$
Unsupervised Learning – k-means clustering

k = 3
2. Compute the mean image (in feature space) for each cluster
Unsupervised Learning – k-means clustering

$k = 3$

3. Reassign images to clusters based on similarity to cluster means
Unsupervised Learning – k-means clustering

4. Keep repeating this process until convergence
Unsupervised Learning – k-means clustering

• How do we choose the right K?
• How do we choose the right features?
• How do we choose the right distance metric?
• How sensitive is this method with respect to the random assignment of clusters?

**Answer:** Just choose the one combination that works best! **BUT** not on the test data.

Instead split the training data into a ”Training set” and a ”Validation set” (also called ”Development set”)

Supervised Learning - Classification

Training Data

Test Data
Supervised Learning - Classification

Training Data

- cat
- dog
- cat
- bear

Test Data

- 
- 
- 
- }
Supervised Learning - Classification

Training Data

\[
\begin{align*}
  x_1 & = [ \text{cat} ] & y_1 & = [ \text{cat} ] \\
  x_2 & = [ \text{dog} ] & y_2 & = [ \text{dog} ] \\
  x_3 & = [ \text{cat} ] & y_3 & = [ \text{cat} ] \\
  \quad \cdots & \quad \cdots & \quad \cdots & \quad \cdots \\
  x_n & = [ \text{bear} ] & y_n & = [ \text{bear} ]
\end{align*}
\]
Supervised Learning - Classification

We need to find a function that maps $x$ and $y$ for any of them.

$$\hat{y}_i = f(x_i; \theta)$$

How do we “learn” the parameters of this function?

We choose ones that makes the following quantity small:

$$\sum_{i=1}^{n} \text{Cost}(\hat{y}_i, y_i)$$

### Training Data

<table>
<thead>
<tr>
<th>inputs</th>
<th>targets / labels / ground truth</th>
<th>predictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1 = [x_{11} \ x_{12} \ x_{13} \ x_{14}]$</td>
<td>$y_1 = 1$</td>
<td>$\hat{y}_1 = 1$</td>
</tr>
<tr>
<td>$x_2 = [x_{21} \ x_{22} \ x_{23} \ x_{24}]$</td>
<td>$y_2 = 2$</td>
<td>$\hat{y}_2 = 2$</td>
</tr>
<tr>
<td>$x_3 = [x_{31} \ x_{32} \ x_{33} \ x_{34}]$</td>
<td>$y_3 = 1$</td>
<td>$\hat{y}_3 = 2$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_n = [x_{n1} \ x_{n2} \ x_{n3} \ x_{n4}]$</td>
<td>$y_n = 3$</td>
<td>$\hat{y}_n = 1$</td>
</tr>
</tbody>
</table>
## Supervised Learning – Linear Softmax

### Training Data

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Targets / Labels / Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1 = [x_{11} \ x_{12} \ x_{13} \ x_{14}]$</td>
<td>$y_1 = 1$</td>
</tr>
<tr>
<td>$x_2 = [x_{21} \ x_{22} \ x_{23} \ x_{24}]$</td>
<td>$y_2 = 2$</td>
</tr>
<tr>
<td>$x_3 = [x_{31} \ x_{32} \ x_{33} \ x_{34}]$</td>
<td>$y_3 = 1$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_n = [x_{n1} \ x_{n2} \ x_{n3} \ x_{n4}]$</td>
<td>$y_n = 3$</td>
</tr>
</tbody>
</table>
# Supervised Learning – Linear Softmax

## Training Data

<table>
<thead>
<tr>
<th>inputs</th>
<th>targets / labels / ground truth</th>
<th>predictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1 = [x_{11} \ x_{12} \ x_{13} \ x_{14}]$</td>
<td>$y_1 = [1 \ 0 \ 0]$</td>
<td>$\hat{y}_1 = [0.85 \ 0.10 \ 0.05]$</td>
</tr>
<tr>
<td>$x_2 = [x_{21} \ x_{22} \ x_{23} \ x_{24}]$</td>
<td>$y_2 = [0 \ 1 \ 0]$</td>
<td>$\hat{y}_2 = [0.20 \ 0.70 \ 0.10]$</td>
</tr>
<tr>
<td>$x_3 = [x_{31} \ x_{32} \ x_{33} \ x_{34}]$</td>
<td>$y_3 = [1 \ 0 \ 0]$</td>
<td>$\hat{y}_3 = [0.40 \ 0.45 \ 0.05]$</td>
</tr>
<tr>
<td>\cdots</td>
<td>\cdots</td>
<td>\cdots</td>
</tr>
<tr>
<td>$x_n = [x_{n1} \ x_{n2} \ x_{n3} \ x_{n4}]$</td>
<td>$y_n = [0 \ 0 \ 1]$</td>
<td>$\hat{y}_n = [0.40 \ 0.25 \ 0.35]$</td>
</tr>
</tbody>
</table>
Supervised Learning – Linear Softmax

\[ x_i = [x_{i1} \ x_{i2} \ x_{i3} \ x_{i4}] \quad y_i = [1 \ 0 \ 0] \quad \hat{y}_i = [f_c \ f_d \ f_b] \]

\[
g_c = w_{c1}x_{i1} + w_{c2}x_{i2} + w_{c3}x_{i3} + w_{c4}x_{i4} + b_c
\]

\[
g_d = w_{d1}x_{i1} + w_{d2}x_{i2} + w_{d3}x_{i3} + w_{d4}x_{i4} + b_d
\]

\[
g_b = w_{b1}x_{i1} + w_{b2}x_{i2} + w_{b3}x_{i3} + w_{b4}x_{i4} + b_b
\]

\[
f_c = e^{g_c}/(e^{g_c} + e^{g_d} + e^{g_b})
\]

\[
f_d = e^{g_d}/(e^{g_c} + e^{g_d} + e^{g_b})
\]

\[
f_b = e^{g_b}/(e^{g_c} + e^{g_d} + e^{g_b})
\]
How do we find a good \( w \) and \( b \)?

\[
x_i = [x_{i1} \ x_{i2} \ x_{i3} \ x_{i4}] \quad y_i = [1 \ 0 \ 0] \quad \hat{y}_i = [f_c(w, b) \ f_d(w, b) \ f_b(w, b)]
\]

We need to find \( w \), and \( b \) that minimize the following:

\[
L(w, b) = \sum_{i=1}^{n} \sum_{j=1}^{3} -y_{i,j} \log(\hat{y}_{i,j}) = \sum_{i=1}^{n} -\log(\hat{y}_{i,\text{label}}) = \sum_{i=1}^{n} -\log f_{i,\text{label}}(w, b)
\]

Why?
Gradient Descent (GD)

\[ \lambda = 0.01 \]

Initialize \( w \) and \( b \) randomly

\[ \text{for } e = 0, \text{num}_\text{epochs} \text{ do} \]

- Compute: \[ \frac{dL(w, b)}{dw} \text{ and } \frac{dL(w, b)}{db} \]
- Update \( w \): \[ w = w - \lambda \frac{dL(w, b)}{dw} \]
- Update \( b \): \[ b = b - \lambda \frac{dL(w, b)}{db} \]

Print: \[ L(w, b) \] // Useful to see if this is becoming smaller or not.

end

\[ L(w, b) = \sum_{i=1}^{n} -\log f_{i,\text{label}}(w, b) \]
Gradient Descent (GD) (idea)

1. Start with a random value of $w$ (e.g. $w = 12$)

2. Compute the gradient (derivative) of $L(w)$ at point $w = 12$. (e.g. $\frac{dL}{dw} = 6$)

3. Recompute $w$ as:

$$w = w - \lambda \cdot \frac{dL}{dw}$$
Gradient Descent (GD) (idea)

2. Compute the gradient (derivative) of $L(w)$ at point $w = 12$. (e.g. $dL/dw = 6$)

3. Recompute $w$ as:

$$w = w - \lambda \times (dL / dw)$$
(mini-batch) Stochastic Gradient Descent (SGD)

\[ l(w, b) = \sum_{i \in B} -\log f_{i, label}(w, b) \]

\[ \lambda = 0.01 \]

Initialize w and b randomly

\[ \text{for } e = 0, \text{ num\_epochs do} \]
\[ \text{for } b = 0, \text{ num\_batches do} \]

- Compute: \( dl(w, b)/dw \) and \( dl(w, b)/db \)
- Update w: \( w = w - \lambda \frac{dl(w, b)}{dw} \)
- Update b: \( b = b - \lambda \frac{dl(w, b)}{db} \)
- Print: \( l(w, b) \) // Useful to see if this is becoming smaller or not.

\[ \text{end} \]
\[ \text{end} \]
Three more things

• How to compute the gradient
• Regularization
• Momentum updates
Questions?