Object Detection

Object localization

deeplearning.ai
What are localization and detection?

Image classification

Classification with localization

Detection

Car

Car

1 object

multiple objects
Classification with localization

1 - pedestrian
2 - car
3 - motorcycle
4 - background

$\begin{align*}
    b_x &= 0.5 \\
    b_y &= 0.7 \\
    bh &= 0.3 \\
    bw &= 0.4
\end{align*}$
Defining the target label $y$

1 - pedestrian
2 - car
3 - motorcycle
4 - background

Need to output $b_x, b_y, b_h, b_w, \text{class label (1-4)}$
Landmark detection

\[ b_x, b_y, b_h, b_w \]

\[ \{\ell_{1x}, \ell_{1y}, \ell_{2x}, \ell_{2y}, \ell_{3x}, \ell_{3y}, \ldots, \ell_{64}, \ell_{64y}\} \]

\[ x, y \]
Object Detection

Object detection
Car detection example

Training set:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
Sliding windows detection

\[ \text{ConvNet} \rightarrow O \]

\[ \text{Compute loss} \]

Andrew Ng
Object Detection

Convolutional implementation of sliding windows
Turning FC layer into convolutional layers

14 × 14 × 3 → 10 × 10 × 16 → 5 × 5 × 16 → FC → FC → softmax (4)

14 × 14 × 3 → 10 × 10 × 16 → 5 × 5 × 16 → FC → FC → 1 × 1 × 4
Convolution implementation of sliding windows

[Sermanet et al., 2014, OverFeat: Integrated recognition, localization and detection using convolutional networks]
Convolution implementation of sliding windows

Andrew Ng
Object Detection

Bounding box predictions
Output accurate bounding boxes
YOLO algorithm

Labels for training
For each grid cell:

Target output: $3 \times 3 \times 8$

[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Specify the bounding boxes

[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Object Detection

Intersection over union
Evaluating object localization

More generally, IoU is a measure of the overlap between two bounding boxes.

Andrew Ng
Object Detection

Non-max suppression
Non-max suppression example
Non-max suppression example

19x19
Non-max suppression example
Non-max suppression algorithm

Each output prediction is:

Discard all boxes with $p_c \leq 0.6$

While there are any remaining boxes:

• Pick the box with the largest $p_c$
  Output that as a prediction.

• Discard any remaining box with $\text{IoU} \geq 0.5$ with the box output in the previous step
Object Detection

Anchor boxes
Overlapping objects:

\[ y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_n \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \]

[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Anchor box algorithm

Previously:

Each object in training image is assigned to grid cell that contains that object’s midpoint.

Output $y$:

$3 \times 2 \times 8$

With two anchor boxes:

Each object in training image is assigned to grid cell that contains object’s midpoint and anchor box for the grid cell with highest IoU.

Output $y$:

$3 \times 3 \times 16$

$3 \times 3 \times 2 \times 8$

(grid cell, anchor box)
Anchor box example

Anchor box 1:  Anchor box 2:

\[ y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \]
Object Detection

Putting it together: YOLO algorithm
Training

1 - pedestrian
2 - car
3 - motorcycle

\[ y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \]

\[ \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} = \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 1 \end{bmatrix} \]

\[ \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 1 \end{bmatrix} \]

\[ \frac{b_x}{1} - \frac{b_y}{1} - \frac{b_h}{2} - \frac{b_w}{2} \]

[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Making predictions

\[
\begin{align*}
  y &= \begin{bmatrix}
  p_c \\
  b_x \\
  b_y \\
  b_h \\
  b_w \\
  c_1 \\
  c_2 \\
  c_3
\end{bmatrix} \\
  &= \begin{bmatrix}
  b_x \\
  b_y \\
  b_h \\
  b_w \\
  c_1 \\
  c_2 \\
  c_3
\end{bmatrix}
\end{align*}
\]
Outputting the non-max supressed outputs

- For each grid call, get 2 predicted bounding boxes.
- Get rid of low probability predictions.
- For each class (pedestrian, car, motorcycle) use non-max suppression to generate final predictions.
Object Detection

Region proposals (Optional)
Region proposal: R-CNN

[Girshik et. al, 2013, Rich feature hierarchies for accurate object detection and semantic segmentation] Andrew Ng
Faster algorithms

→ R-CNN: Propose regions. Classify proposed regions one at a time. Output label + bounding box.

Fast R-CNN: Propose regions. Use convolution implementation of sliding windows to classify all the proposed regions.

Faster R-CNN: Use convolutional network to propose regions.

[Ren et. al, 2016. Faster R-CNN: Towards real-time object detection with region proposal networks]