CS 7643: Deep Learning

Topics:
- Announcements
- Transposed convolutions
- Presentations on Visualizations / Explanations

Dhruv Batra
Georgia Tech
Administrativia

• HW2 + PS2 out

• No class on Tuesday 10/03

• Guest Lecture by Dr. Stefan Lee on 10/05
  – No papers to read. No student presentations.
  – Use the time wisely to work (hint: HW+PS!)

• No class on 10/10
  – Fall break

• Next paper reading / student presentations: 10/12

• Note on reviews

(C) Dhruv Batra
Note on reviews

• Public
  – Good and bad

• Common Problem #1: Vague negatives
  – “x could have been done better”

• Common Problem #2: Virtue Signaling
  – “I have higher standards than this”

• Positive suggestion: Assume good intent
  – There’s a grad student just like you behind that paper

• Snobbery has to be earned
Learnable Upsampling: Transpose Convolution

Recall: Typical 3 x 3 convolution, stride 1 pad 1

Input: 4 x 4

Output: 4 x 4

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Learnable Upsampling: Transpose Convolution

**Recall:** Normal 3 x 3 convolution, stride 1 pad 1

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Learnable Upsampling: Transpose Convolution

**Recall:** Normal 3 x 3 convolution, stride 1 pad 1

Dot product between filter and input

Input: 4 x 4

Output: 4 x 4
Learnable Upsampling: Transpose Convolution

**Recall:** Normal 3 x 3 convolution, stride 2, pad 1

**Input:** 4 x 4  
**Output:** 2 x 2

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Learnable Upsampling: Transpose Convolution

**Recall:** Normal 3 x 3 convolution, stride 2 pad 1

- **Input:** 4 x 4
- **Output:** 2 x 2

[Diagram showing the process of transpose convolution with a 3x3 filter sliding over a 4x4 input to produce a 2x2 output through dot product calculations.]
Learnable Upsampling: Transpose Convolution

Recall: Normal 3 x 3 convolution, stride 2 pad 1

Filter moves 2 pixels in the input for every one pixel in the output.

Stride gives ratio between movement in input and output.

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Learnable Upsampling: Transpose Convolution

3 x 3 transpose convolution, stride 2 pad 1

Input: 2 x 2

Output: 4 x 4

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Learnable Upsampling: Transpose Convolution

3 x 3 transpose convolution, stride 2 pad 1

Input: 2 x 2

Output: 4 x 4

Input gives weight for filter

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Learnable Upsampling: Transpose Convolution

$3 \times 3$ transpose convolution, stride 2 pad 1

Input gives weight for filter

Input: $2 \times 2$

Output: $4 \times 4$

Sum where output overlaps

Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Learnable Upsampling: Transpose Convolution

Other names:
- Deconvolution (bad)
- Upconvolution
- Fractionally strided convolution
- Backward strided convolution

3 x 3 transpose convolution, stride 2 pad 1

Input gives weight for filter

Input: 2 x 2

Output: 4 x 4

Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

Sum where output overlaps

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Transpose Convolution: 1D Example

Input

Filter

Output

Output contains copies of the filter weighted by the input, summing at where at overlaps in the output.

Need to crop one pixel from output to make output exactly 2x input.

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Transposed Convolution

- https://distill.pub/2016/deconv-checkerboard/
Transposed Convolution

• In 3D
Semantic Segmentation Idea: Fully Convolutional

**Downsampling:**
Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

- **Input:** $3 \times H \times W$
- **High-res:** $D_1 \times H/2 \times W/2$
- **Med-res:** $D_2 \times H/4 \times W/4$
- **Low-res:** $D_3 \times H/4 \times W/4$
- **Predictions:** $H \times W$

**Upsampling:**
Unpooling or strided transpose convolution


Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
What is deconvolution?

• (Non-blind) Deconvolution

\[ y = w * x \]

\[ x \ast w = y \]

\[ w = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \]

\[ x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \]

\[ y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} \]
What does “deconvolution” have to do with “transposed convolution”
“transposed convolution” is a convolution!

We can express convolution in terms of a matrix multiplication

\[
\vec{x} \ast \vec{a} = X \vec{a}
\]

Example: 1D conv, kernel size=3, stride=1, padding=1

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
"transposed convolution" is a convolution!

We can express convolution in terms of a matrix multiplication:

\[ \vec{x} \ast \vec{a} = X \vec{a} \]

Convolution transpose multiplies by the transpose of the same matrix:

\[ (\vec{x} \ast_T \vec{a}) = X^T \vec{a} \]

Example: 1D conv, kernel size=3, stride=1, padding=1

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
“transposed convolution” is a convolution

We can express convolution in terms of a matrix multiplication

\[ \tilde{x} \ast \tilde{a} = X \tilde{a} \]

\[
\begin{bmatrix}
    x & y & z & 0 & 0 & 0 \\
    0 & x & y & z & 0 & 0 \\
    0 & 0 & x & y & z & 0 \\
    0 & 0 & 0 & x & y & z \\
\end{bmatrix}
\begin{bmatrix}
    0 \\
    a \\
    b \\
    c \\
    d \\
    0 \\
\end{bmatrix}
= \begin{bmatrix}
    ay + bz \\
    ax + by + cz \\
    bx + cy + dz \\
    cx + dy \\
\end{bmatrix}
\]

Example: 1D conv, kernel size=3, stride=1, padding=1

Convolution transpose multiplies by the transpose of the same matrix:

\[ \tilde{x} \ast^T \tilde{a} = X^T \tilde{a} \]

\[
\begin{bmatrix}
    x & 0 & 0 & 0 \\
    y & x & 0 & 0 \\
    z & y & x & 0 \\
    0 & z & y & x \\
    0 & 0 & z & y \\
    0 & 0 & 0 & z \\
\end{bmatrix}
\begin{bmatrix}
    a \\
    b \\
    c \\
    d \\
\end{bmatrix}
= \begin{bmatrix}
    ax \\
    ay + bx \\
    az + by + cx \\
    bz + cy + dx \\
    cz + dy \\
    dz \\
\end{bmatrix}
\]

When stride=1, convolution transpose is just a regular convolution (with different padding rules)

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
We can express convolution in terms of a matrix multiplication:

\[ \vec{x} \ast \vec{a} = X \vec{a} \]

Example: 1D conv, kernel size=3, \textit{stride}=2, \textit{padding}=1
But not always

We can express convolution in terms of a matrix multiplication

\[ \vec{x} \ast \vec{a} = X \vec{a} \]

Example: 1D conv, kernel size=3, \text{stride}=2, padding=1

\[
\begin{bmatrix}
x & y & z & 0 & 0 & 0 \\
0 & 0 & x & y & z & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
0 \\
a \\
b \\
c \\
d \\
0
\end{bmatrix}
= \begin{bmatrix}
ay + bz \\
by + cz + d \end{bmatrix}
\]

Convolution transpose multiplies by the transpose of the same matrix:

\[ \vec{x} \ast^T \vec{a} = X^T \vec{a} \]

\[
\begin{bmatrix}
x & 0 \\
y & 0 \\
z & x \\
0 & y \\
0 & z \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix}
= \begin{bmatrix}
ax \\
ay \\
z + bx \\
by \\
bz \\
0
\end{bmatrix}
\]

When \text{stride}>1, convolution transpose is no longer a normal convolution!
Student Presentations

• Presenters:
  – Aneeq Zia, Mikhail Isaev, Chris Donlan, Ayesha Khan, Deshraj Yadav, Ardavan Afshar

• Google Slides:
  – https://docs.google.com/presentation/d/1HadX--rN-KC7tJquC2mhDIsteouLYiJKqpnAgz54h5o/edit#slide=id.p

• Google Drive
  – https://drive.google.com/drive/folders/0B8zT-Fl5PDf_dlpBREQwZ1VHa0k?usp=sharing