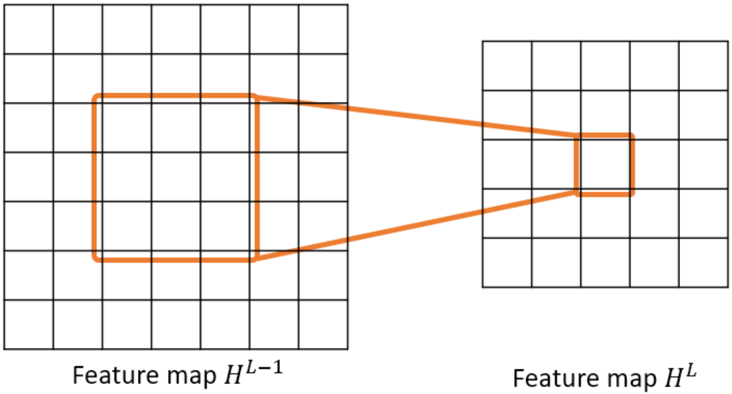
Convolutional Networks 2: Training, deep convolutional networks

Hakan Bilen

Machine Learning Practical — MLP Lecture 8
30 October / 6 November 2018

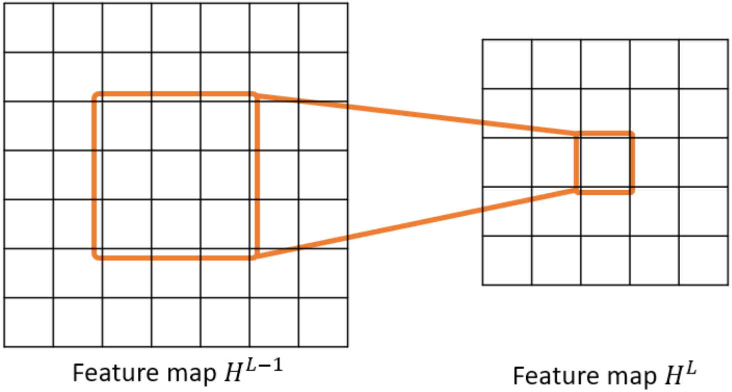
Question

Q1. How can we increase the receptive field area of a conv layer?



Question

Q1. How can we increase the receptive field area of a conv layer?
Q2. Can we do it without increasing kernel size?



Input arguments for convolution function

class `torch.nn.Conv2d`(*in_channels, out_channels, kernel_size, stride=1, padding=0, dilation=1, groups=1, bias=True*) [source]

Applies a 2D convolution over an input signal composed of several input planes.

- `in_channels`
- `out_channels`
- `kernel_size`
- `stride`
- `padding`
- `bias`

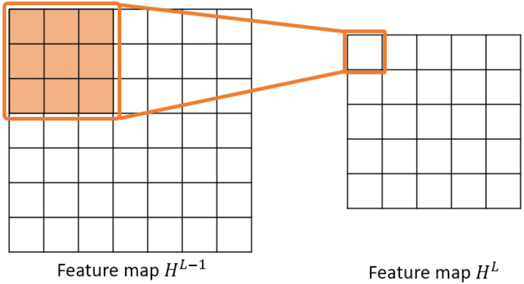
Input arguments for convolution function

class **torch.nn.Conv2d**(*in_channels, out_channels, kernel_size, stride=1, padding=0, dilation=1, groups=1, bias=True*) [source]

Applies a 2D convolution over an input signal composed of several input planes.

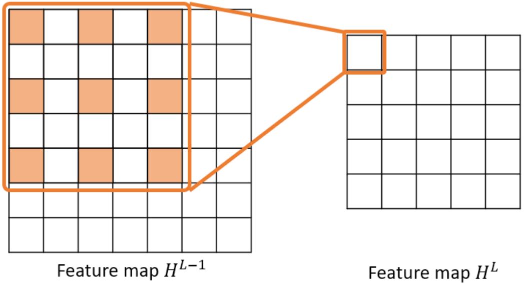
- `in_channels`
- `out_channels`
- `kernel_size`
- `stride`
- `padding`
- `bias`
- `dilation?`
- `groups?`

Dilated convolutions



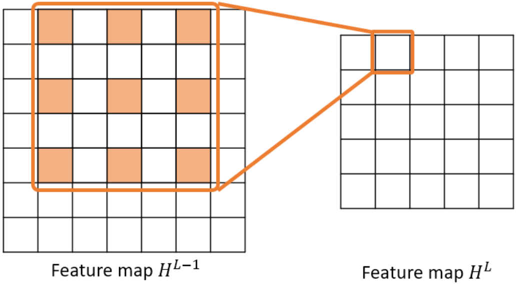
Dilated convolutions

Increased receptive field by **inflating** the kernel by inserting $D - 1$ spaces between the kernel elements



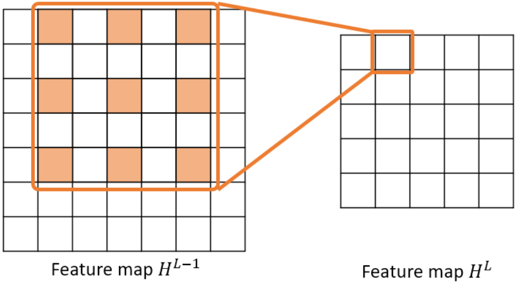
Dilated convolutions

Increased receptive field by **inflating** the kernel by inserting $D-1$ spaces between the kernel elements



Dilated convolutions

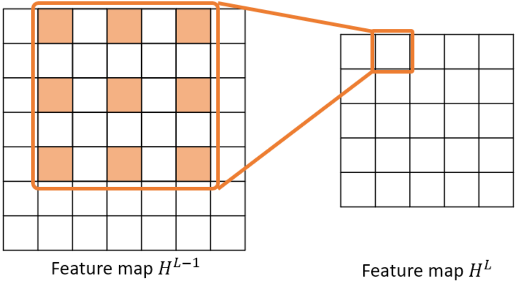
Increased receptive field by **inflating** the kernel by inserting $D-1$ spaces between the kernel elements



Why to increase receptive field size?

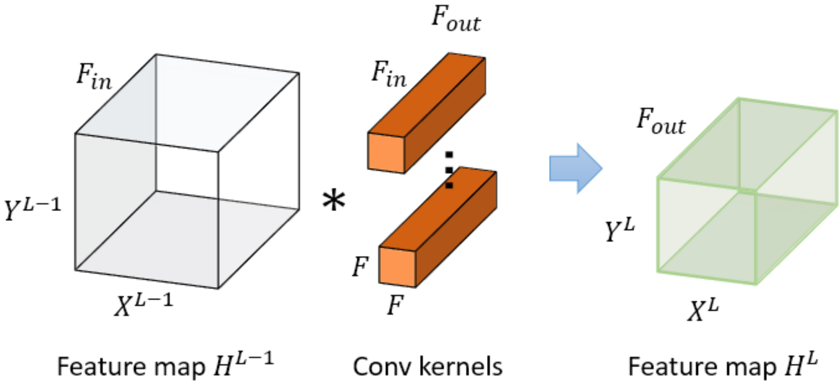
Dilated convolutions

Increased receptive field by **inflating** the kernel by inserting $D-1$ spaces between the kernel elements



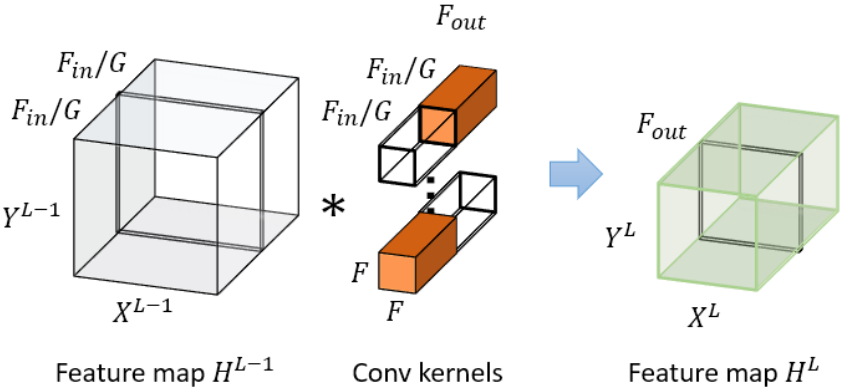
Yu & Koltun, "Multi-scale context aggregation by dilated convolutions", *ICLR*, 2016.
https://arxiv.org/pdf/1511.07122.pdf

(Convolutional) filter groups



(Convolutional) filter groups

- G is number of groups
- Reduces number of convolutional filters (or parameters)
- Regularisation effect



Convolution and cross-correlation

- We can write the feature map hidden unit equation (Index-0):

$$h_{i,j} = \sum_{m=0} \sum_{n=0} I(m+i, n+j) W(m,n)$$

$$h = W \otimes I$$

$\otimes$ is a cross-correlation

Convolution and cross-correlation

- We can write the feature map hidden unit equation (Index-0):

$$h_{i,j} = \sum_{m=0} \sum_{n=0} I(m+i, n+j) W(m,n)$$
$$h = W \otimes I$$

$\otimes$ is a cross-correlation

- In signal processing a 2D convolution is written as

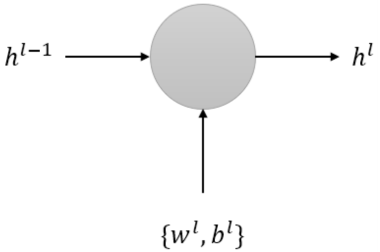
$$h_{i,j} = (V * I) = \sum_{m=0} \sum_{n=0} I(m,n) V(i-m, j-n)$$
$$h_{i,j} = (I * V) = \sum_{m=0} \sum_{n=0} I(i-m, j-n) V(m,n)$$

- If we "flip" (reflect horizontally and vertically) $\boldsymbol{W}$ (cross-correlation) then we obtain $\boldsymbol{V}$ (convolution)

Training Convolutional Networks

Forward pass

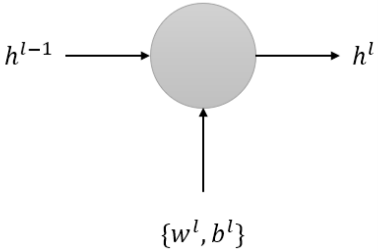
$$h^l = f^l(h^{l-1}, \{w^l, b^l\})$$



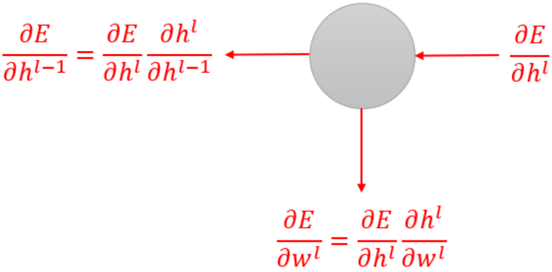
Training Convolutional Networks

Forward pass

$$h^l = f^l(h^{l-1}, \{w^l, b^l\})$$



Backward pass



Example

$$\underbrace{\begin{bmatrix} h^{l-1}_{11} & h^{l-1}_{12} & h^{l-1}_{13} \\ h^{l-1}_{21} & h^{l-1}_{22} & h^{l-1}_{23} \\ h^{l-1}_{31} & h^{l-1}_{32} & h^{l-1}_{33} \end{bmatrix}}_{H^{l-1}} \quad \underbrace{\begin{bmatrix} w^{l}_{11} & w^{l}_{12} \\ w^{l}_{21} & w^{l}_{22} \end{bmatrix}}_{W^{l}} = \underbrace{\begin{bmatrix} h^{l}_{11} & h^{l}_{12} \\ h^{l}_{21} & h^{l}_{22} \end{bmatrix}}_{H^{l}}$$

$$h^l_{11} = w^l_{11} h^{l-1}_{11} + w^l_{12} h^{l-1}_{12} + w^l_{21} h^{l-1}_{21} + w^l_{22} h^{l-1}_{22} + b$$
$$h^l_{12} = w^l_{11} h^{l-1}_{12} + w^l_{12} h^{l-1}_{13} + w^l_{21} h^{l-1}_{22} + w^l_{22} h^{l-1}_{23} + b$$
$$h^l_{21} = w^l_{11} h^{l-1}_{21} + w^l_{12} h^{l-1}_{22} + w^l_{21} h^{l-1}_{31} + w^l_{22} h^{l-1}_{32} + b$$
$$h^l_{22} = w^l_{11} h^{l-1}_{22} + w^l_{12} h^{l-1}_{23} + w^l_{21} h^{l-1}_{32} + w^l_{22} h^{l-1}_{33} + b$$

Gradients of E w.r.t W^l

$$\begin{array}{ccc}
\begin{array}{|c|c|c|}\hline h^{l-1}_{11} & h^{l-1}_{12} & h^{l-1}_{13} \\ \hline h^{l-1}_{21} & h^{l-1}_{22} & h^{l-1}_{23} \\ \hline h^{l-1}_{31} & h^{l-1}_{32} & h^{l-1}_{33} \\ \hline \end{array}
&
\begin{array}{|c|c|}\hline \dfrac{\partial E}{\partial h^l_{11}} & \dfrac{\partial E}{\partial h^l_{12}} \\ \hline \dfrac{\partial E}{\partial h^l_{21}} & \dfrac{\partial E}{\partial h^l_{22}} \\ \hline \end{array}
\; = &
\begin{array}{|c|c|}\hline \dfrac{\partial E}{\partial w^l_{11}} & \dfrac{\partial E}{\partial w^l_{12}} \\ \hline \dfrac{\partial E}{\partial w^l_{21}} & \dfrac{\partial E}{\partial w^l_{22}} \\ \hline \end{array}
\\
H^{l-1} & \partial E/\partial H^l & \partial E/\partial W^l
\end{array}$$

Example

$$
\begin{aligned}
h^l_{11} &= w^l_{11} h^{l-1}_{11} + w^l_{12} h^{l-1}_{12} + w^l_{21} h^{l-1}_{21} + w^l_{22} h^{l-1}_{22} + b \\
h^l_{12} &= w^l_{11} h^{l-1}_{12} + w^l_{12} h^{l-1}_{13} + w^l_{21} h^{l-1}_{22} + w^l_{22} h^{l-1}_{23} + b \\
h^l_{21} &= w^l_{11} h^{l-1}_{21} + w^l_{12} h^{l-1}_{22} + w^l_{21} h^{l-1}_{31} + w^l_{22} h^{l-1}_{32} + b \\
h^l_{22} &= w^l_{11} h^{l-1}_{22} + w^l_{12} h^{l-1}_{23} + w^l_{21} h^{l-1}_{32} + w^l_{22} h^{l-1}_{33} + b
\end{aligned}
$$

Let's calculate the parameter updates ($\frac{\partial E}{\partial W^l}$)

$$
\frac{\partial E}{\partial w^l_{11}} = \frac{\partial E}{\partial h^l_{11}} \frac{\partial h^l_{11}}{\partial w^l_{11}} + \frac{\partial E}{\partial h^l_{12}} \frac{\partial h^l_{12}}{\partial w^l_{11}} + \frac{\partial E}{\partial h^l_{21}} \frac{\partial h^l_{21}}{\partial w^l_{11}} + \frac{\partial E}{\partial h^l_{22}} \frac{\partial h^l_{22}}{\partial w^l_{11}}
$$

Example

$$
\begin{aligned}
h^l_{11} &= w^l_{11}\boxed{h^{l-1}_{11}} + w^l_{12}h^{l-1}_{12} + w^l_{21}h^{l-1}_{21} + w^l_{22}h^{l-1}_{22} + b \\
h^l_{12} &= w^l_{11}\boxed{h^{l-1}_{12}} + w^l_{12}h^{l-1}_{13} + w^l_{21}h^{l-1}_{22} + w^l_{22}h^{l-1}_{23} + b \\
h^l_{21} &= w^l_{11}\boxed{h^{l-1}_{21}} + w^l_{12}h^{l-1}_{22} + w^l_{21}h^{l-1}_{31} + w^l_{22}h^{l-1}_{32} + b \\
h^l_{22} &= w^l_{11}\boxed{h^{l-1}_{22}} + w^l_{12}h^{l-1}_{23} + w^l_{21}h^{l-1}_{32} + w^l_{22}h^{l-1}_{33} + b
\end{aligned}
$$

Let's calculate the parameter updates ($\frac{\partial E}{\partial W^l}$)

$$
\frac{\partial E}{\partial w^l_{11}} = \frac{\partial E}{\partial h^l_{11}} h^{l-1}_{11} + \frac{\partial E}{\partial h^l_{12}} h^{l-1}_{12} + \frac{\partial E}{\partial h^l_{21}} h^{l-1}_{21} + \frac{\partial E}{\partial h^l_{22}} h^{l-1}_{22}
$$

Example

$$
\begin{aligned}
h^l_{11} &= w^l_{11} h^{l-1}_{11} + w^l_{12} h^{l-1}_{12} + w^l_{21} h^{l-1}_{21} + w^l_{22} h^{l-1}_{22} + b \\
h^l_{12} &= w^l_{11} h^{l-1}_{12} + w^l_{12} h^{l-1}_{13} + w^l_{21} h^{l-1}_{22} + w^l_{22} h^{l-1}_{23} + b \\
h^l_{21} &= w^l_{11} h^{l-1}_{21} + w^l_{12} h^{l-1}_{22} + w^l_{21} h^{l-1}_{31} + w^l_{22} h^{l-1}_{32} + b \\
h^l_{22} &= w^l_{11} h^{l-1}_{22} + w^l_{12} h^{l-1}_{23} + w^l_{21} h^{l-1}_{32} + w^l_{22} h^{l-1}_{33} + b
\end{aligned}
$$

Let's calculate the parameter updates ($\frac{\partial E}{\partial W^l}$)

$$
\frac{\partial E}{\partial w^l_{12}} = \frac{\partial E}{\partial h^l_{11}} h^{l-1}_{12} + \frac{\partial E}{\partial h^l_{1,2}} h^{l-1}_{13} + \frac{\partial E}{\partial h^l_{21}} h^{l-1}_{22} + \frac{\partial E}{\partial h^l_{22}} h^{l-1}_{23}
$$

Example

$$
\begin{aligned}
h^l_{11} &= w^l_{11} h^{l-1}_{11} + w^l_{12} h^{l-1}_{12} + w^l_{21} \boxed{h^{l-1}_{21}} + w^l_{22} h^{l-1}_{22} + b \\
h^l_{12} &= w^l_{11} h^{l-1}_{12} + w^l_{12} h^{l-1}_{13} + w^l_{21} \boxed{h^{l-1}_{22}} + w^l_{22} h^{l-1}_{23} + b \\
h^l_{21} &= w^l_{11} h^{l-1}_{21} + w^l_{12} h^{l-1}_{22} + w^l_{21} \boxed{h^{l-1}_{31}} + w^l_{22} h^{l-1}_{32} + b \\
h^l_{22} &= w^l_{11} h^{l-1}_{22} + w^l_{12} h^{l-1}_{23} + w^l_{21} \boxed{h^{l-1}_{32}} + w^l_{22} h^{l-1}_{33} + b
\end{aligned}
$$

Let's calculate the parameter updates ($\frac{\partial E}{\partial W^l}$)

$$
\frac{\partial E}{\partial w^l_{2,1}} = \frac{\partial E}{\partial h^l_{11}} h^{l-1}_{21} + \frac{\partial E}{\partial h^l_{12}} h^{l-1}_{22} + \frac{\partial E}{\partial h^l_{21}} h^{l-1}_{31} + \frac{\partial E}{\partial h^l_{22}} h^{l-1}_{32}
$$

Example

$$
\begin{aligned}
h^l_{11} &= w^l_{11} h^{l-1}_{11} + w^l_{12} h^{l-1}_{12} + w^l_{21} h^{l-1}_{21} + w^l_{22} h^{l-1}_{22} + b \\
h^l_{12} &= w^l_{11} h^{l-1}_{12} + w^l_{12} h^{l-1}_{13} + w^l_{21} h^{l-1}_{22} + w^l_{22} h^{l-1}_{23} + b \\
h^l_{21} &= w^l_{11} h^{l-1}_{21} + w^l_{12} h^{l-1}_{22} + w^l_{21} h^{l-1}_{31} + w^l_{22} h^{l-1}_{32} + b \\
h^l_{22} &= w^l_{11} h^{l-1}_{22} + w^l_{12} h^{l-1}_{23} + w^l_{21} h^{l-1}_{32} + w^l_{22} h^{l-1}_{33} + b
\end{aligned}
$$

Let's calculate the parameter updates $(\frac{\partial E}{\partial W^l})$

$$
\frac{\partial E}{\partial w^l_{2,2}} = \frac{\partial E}{\partial h^l_{11}} h^{l-1}_{22} + \frac{\partial E}{\partial h^l_{12}} h^{l-1}_{33} + \frac{\partial E}{\partial h^l_{21}} h^{l-1}_{32} + \frac{\partial E}{\partial h^l_{22}} h^{l-1}_{33}
$$

Gradients of E w.r.t W^l

$$
\begin{aligned}
\frac{\partial E}{\partial w^l_{1,1}} &= \frac{\partial E}{\partial h^l_{11}} h^{l-1}_{11} + \frac{\partial E}{\partial h^l_{12}} h^{l-1}_{12} + \frac{\partial E}{\partial h^l_{21}} h^{l-1}_{21} + \frac{\partial E}{\partial h^l_{22}} h^{l-1}_{22} \\
\frac{\partial E}{\partial w^l_{12}} &= \frac{\partial E}{\partial h^l_{11}} h^{l-1}_{12} + \frac{\partial E}{\partial h^l_{12}} h^{l-1}_{13} + \frac{\partial E}{\partial h^l_{21}} h^{l-1}_{22} + \frac{\partial E}{\partial h^l_{22}} h^{l-1}_{23} \\
\frac{\partial E}{\partial w^l_{21}} &= \frac{\partial E}{\partial h^l_{11}} h^{l-1}_{21} + \frac{\partial E}{\partial h^l_{12}} h^{l-1}_{22} + \frac{\partial E}{\partial h^l_{21}} h^{l-1}_{31} + \frac{\partial E}{\partial h^l_{22}} h^{l-1}_{32} \\
\frac{\partial E}{\partial w^l_{22}} &= \frac{\partial E}{\partial h^l_{11}} h^{l-1}_{22} + \frac{\partial E}{\partial h^l_{12}} h^{l-1}_{33} + \frac{\partial E}{\partial h^l_{21}} h^{l-1}_{32} + \frac{\partial E}{\partial h^l_{22}} h^{l-1}_{33}
\end{aligned}
$$

Given $H^l \in \mathcal{R}^{M^l \times N^l}$

$$
\boxed{\frac{\partial E}{\partial w^l_{r,s}} = \sum_{m=1}^{M^l} \sum_{n=1}^{N^l} \frac{\partial E}{\partial h^l_{m,n}} h^{l-1}_{r+m-1,s+n-1}}
$$

Gradients of E w.r.t W^l

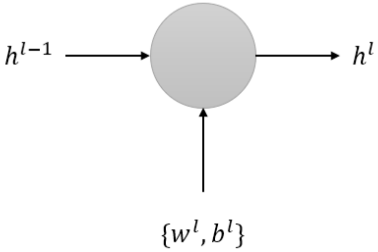
$$\frac{\partial E}{\partial w^l_{r,s}} = \sum_{m=1}^{M^l} \sum_{n=1}^{N^l} \frac{\partial E}{\partial h^l_{m,n}} h^{l-1}_{r+m-1,s+n-1}$$

$$\underbrace{\begin{array}{|c|c|c|} \hline h^{l-1}_{11} & h^{l-1}_{12} & h^{l-1}_{13} \\ \hline h^{l-1}_{21} & h^{l-1}_{22} & h^{l-1}_{23} \\ \hline h^{l-1}_{31} & h^{l-1}_{32} & h^{l-1}_{33} \\ \hline \end{array}}_{H^{l-1}} \qquad \underbrace{\begin{array}{|c|c|} \hline \frac{\partial E}{\partial h^l_{11}} & \frac{\partial E}{\partial h^l_{12}} \\ \hline \frac{\partial E}{\partial h^l_{21}} & \frac{\partial E}{\partial h^l_{22}} \\ \hline \end{array}}_{\partial E/\partial H^l} = \underbrace{\begin{array}{|c|c|} \hline \frac{\partial E}{\partial w^l_{11}} & \frac{\partial E}{\partial w^l_{12}} \\ \hline \frac{\partial E}{\partial w^l_{21}} & \frac{\partial E}{\partial w^l_{22}} \\ \hline \end{array}}_{\partial E/\partial W^l}$$

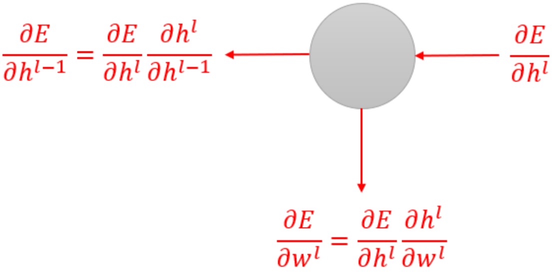
Gradients of E w.r.t H^{l-1}

Forward pass

$$h^l = f^l(h^{l-1}, \{w^l, b^l\})$$



Backward pass



Gradients of E w.r.t H^{l-1}

$$\underbrace{\begin{bmatrix} h_{11}^{l-1} & h_{12}^{l-1} & h_{13}^{l-1} \\ h_{21}^{l-1} & h_{22}^{l-1} & h_{23}^{l-1} \\ h_{31}^{l-1} & h_{32}^{l-1} & h_{33}^{l-1} \end{bmatrix}}_{H^{l-1}} \quad \underbrace{\begin{bmatrix} w_{11}^{l} & w_{12}^{l} \\ w_{21}^{l} & w_{22}^{l} \end{bmatrix}}_{W^{l}} = \underbrace{\begin{bmatrix} h_{11}^{l} & h_{12}^{l} \\ h_{21}^{l} & h_{22}^{l} \end{bmatrix}}_{H^{l}}$$

Gradients of E w.r.t H^{l-1}

$$
\begin{array}{|c|c|c|c|}
\hline
0 & 0 & 0 & 0 \\
\hline
0 & \frac{\partial E}{\partial h^l_{11}} & \frac{\partial E}{\partial h^l_{12}} & 0 \\
\hline
0 & \frac{\partial E}{\partial h^l_{21}} & \frac{\partial E}{\partial h^l_{22}} & 0 \\
\hline
0 & 0 & 0 & 0 \\
\hline
\end{array}
\quad
\begin{array}{|c|c|}
\hline
w^l_{22} & w^l_{21} \\
\hline
w^l_{12} & w^l_{11} \\
\hline
\end{array}
\quad = \quad
\begin{array}{|c|c|c|}
\hline
\frac{\partial E}{\partial h^{l-1}_{11}} & \frac{\partial E}{\partial h^{l-1}_{12}} & \frac{\partial E}{\partial h^{l-1}_{13}} \\
\hline
\frac{\partial E}{\partial h^{l-1}_{21}} & \frac{\partial E}{\partial h^{l-1}_{22}} & \frac{\partial E}{\partial h^{l-1}_{23}} \\
\hline
\frac{\partial E}{\partial h^{l-1}_{31}} & \frac{\partial E}{\partial h^{l-1}_{32}} & \frac{\partial E}{\partial h^{l-1}_{33}} \\
\hline
\end{array}
$$

Padded $\partial E / \partial H^l$ Rotated W^l $\partial E / \partial H^{l-1}$

Imagine inverting the receptive field!

Gradients of E w.r.t h^{l-1}

$$
\begin{aligned}
h^l_{11} &= w^l_{11} h^{l-1}_{11} + w^l_{12} h^{l-1}_{12} + w^l_{21} h^{l-1}_{21} + w^l_{22} h^{l-1}_{22} + b \\
h^l_{12} &= w^l_{11} h^{l-1}_{12} + w^l_{12} h^{l-1}_{13} + w^l_{21} h^{l-1}_{22} + w^l_{22} h^{l-1}_{23} + b \\
h^l_{21} &= w^l_{11} h^{l-1}_{21} + w^l_{12} h^{l-1}_{22} + w^l_{21} h^{l-1}_{31} + w^l_{22} h^{l-1}_{32} + b \\
h^l_{22} &= w^l_{11} h^{l-1}_{22} + w^l_{12} h^{l-1}_{23} + w^l_{21} h^{l-1}_{32} + w^l_{22} h^{l-1}_{33} + b
\end{aligned}
$$

Let's calculate the gradients of loss function (E) with respect to previous layer (H^{l-1})

$$
\frac{\partial E}{\partial h^{l-1}_{11}} = \frac{\partial E}{\partial h^l_{11}} \frac{\partial h^l_{11}}{\partial h^{l-1}_{11}} + \frac{\partial E}{\partial h^l_{12}} \frac{\partial h^l_{12}}{\partial h^{l-1}_{11}} + \frac{\partial E}{\partial h^l_{21}} \frac{\partial h^l_{21}}{\partial h^{l-1}_{11}} + \frac{\partial E}{\partial h^l_{22}} \frac{\partial h^l_{22}}{\partial h^{l-1}_{11}}
$$

Gradients of E w.r.t h^{l-1}

$$
\begin{aligned}
h^l_{11} &= \boxed{w^l_{11} h^{l-1}_{11}} + w^l_{12} h^{l-1}_{12} + w^l_{21} h^{l-1}_{21} + w^l_{22} h^{l-1}_{22} + b \\
h^l_{12} &= w^l_{11} h^{l-1}_{12} + w^l_{12} h^{l-1}_{13} + w^l_{21} h^{l-1}_{22} + w^l_{22} h^{l-1}_{23} + b \\
h^l_{21} &= w^l_{11} h^{l-1}_{21} + w^l_{12} h^{l-1}_{22} + w^l_{21} h^{l-1}_{31} + w^l_{22} h^{l-1}_{32} + b \\
h^l_{22} &= w^l_{11} h^{l-1}_{22} + w^l_{12} h^{l-1}_{23} + w^l_{21} h^{l-1}_{32} + w^l_{22} h^{l-1}_{33} + b
\end{aligned}
$$

Let's calculate the gradients of loss function (E) with respect to previous layer (H^{l-1})

$$
\frac{\partial E}{\partial h^{l-1}_{11}} = \frac{\partial E}{\partial h^l_{11}} \frac{\partial h^l_{11}}{\partial h^{l-1}_{11}} + \frac{\partial E}{\partial h^l_{12}} \frac{\partial h^l_{12}}{\partial h^{l-1}_{11}} + \frac{\partial E}{\partial h^l_{21}} \frac{\partial h^l_{21}}{\partial h^{l-1}_{11}} + \frac{\partial E}{\partial h^l_{22}} \frac{\partial h^l_{22}}{\partial h^{l-1}_{11}}
$$

$$
\frac{\partial E}{\partial h^{l-1}_{11}} = \frac{\partial E}{\partial h^l_{11}} w^l_{11} + \frac{\partial E}{\partial h^l_{12}} 0 + \frac{\partial E}{\partial h^l_{21}} 0 + \frac{\partial E}{\partial h^l_{22}} 0
$$

Gradients of E w.r.t h^{l-1}

$$
\begin{aligned}
h^l_{11} &= w^l_{11} h^{l-1}_{11} + w^l_{12} h^{l-1}_{12} + w^l_{21} h^{l-1}_{21} + w^l_{22} h^{l-1}_{22} + b \\
h^l_{12} &= w^l_{11} h^{l-1}_{12} + w^l_{12} h^{l-1}_{13} + w^l_{21} h^{l-1}_{22} + w^l_{22} h^{l-1}_{23} + b \\
h^l_{21} &= w^l_{11} h^{l-1}_{21} + w^l_{12} h^{l-1}_{22} + w^l_{21} h^{l-1}_{31} + w^l_{22} h^{l-1}_{32} + b \\
h^l_{22} &= w^l_{11} h^{l-1}_{22} + w^l_{12} h^{l-1}_{23} + w^l_{21} h^{l-1}_{32} + w^l_{22} h^{l-1}_{33} + b
\end{aligned}
$$

Let's calculate the gradients of loss function (E) with respect to previous layer (H^{l-1})

$$
\frac{\partial E}{\partial h^{l-1}_{22}} = \frac{\partial E}{\partial h^l_{11}} \frac{\partial h^l_{11}}{\partial h^{l-1}_{22}} + \frac{\partial E}{\partial h^l_{12}} \frac{\partial h^l_{12}}{\partial h^{l-1}_{22}} + \frac{\partial E}{\partial h^l_{21}} \frac{\partial h^l_{21}}{\partial h^{l-1}_{22}} + \frac{\partial E}{\partial h^l_{22}} \frac{\partial h^l_{22}}{\partial h^{l-1}_{22}}
$$

Gradients of E w.r.t h^{l-1}

$$
\begin{aligned}
h^l_{11} &= w^l_{11} h^{l-1}_{11} + w^l_{12} h^{l-1}_{12} + w^l_{21} h^{l-1}_{21} + \boxed{w^l_{22} h^{l-1}_{22}} + b \\
h^l_{12} &= w^l_{11} h^{l-1}_{12} + w^l_{12} h^{l-1}_{13} + \boxed{w^l_{21} h^{l-1}_{22}} + w^l_{22} h^{l-1}_{23} + b \\
h^l_{21} &= w^l_{11} h^{l-1}_{21} + \boxed{w^l_{12} h^{l-1}_{22}} + w^l_{21} h^{l-1}_{31} + w^l_{22} h^{l-1}_{32} + b \\
h^l_{22} &= \boxed{w^l_{11} h^{l-1}_{22}} + w^l_{12} h^{l-1}_{23} + w^l_{21} h^{l-1}_{32} + w^l_{22} h^{l-1}_{33} + b
\end{aligned}
$$

Let's calculate the gradients of loss function (E) with respect to previous layer (H^{l-1})

$$
\frac{\partial E}{\partial h^{l-1}_{22}} = \frac{\partial E}{\partial h^l_{11}} \frac{\partial h^l_{11}}{\partial h^{l-1}_{22}} + \frac{\partial E}{\partial h^l_{12}} \frac{\partial h^l_{12}}{\partial h^{l-1}_{22}} + \frac{\partial E}{\partial h^l_{21}} \frac{\partial h^l_{21}}{\partial h^{l-1}_{22}} + \frac{\partial E}{\partial h^l_{22}} \frac{\partial h^l_{22}}{\partial h^{l-1}_{22}}
$$

$$
\frac{\partial E}{\partial h^{l-1}_{22}} = \frac{\partial E}{\partial h^l_{11}} w^l_{22} + \frac{\partial E}{\partial h^l_{12}} w^l_{21} + \frac{\partial E}{\partial h^l_{21}} w^l_{12} + \frac{\partial E}{\partial h^l_{22}} w^l_{11}
$$

Gradients of E w.r.t H^{l-1}

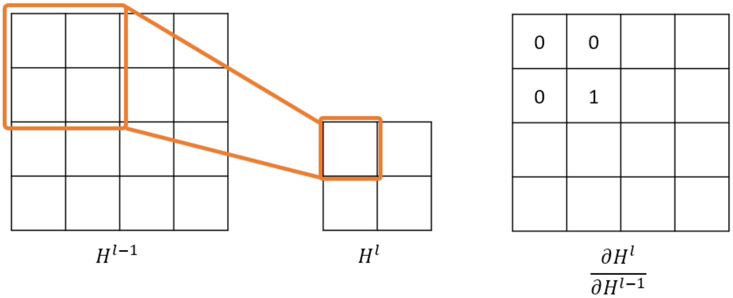
$$
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & \frac{\partial E}{\partial h^l_{11}} & \frac{\partial E}{\partial h^l_{12}} & 0 \\
0 & \frac{\partial E}{\partial h^l_{21}} & \frac{\partial E}{\partial h^l_{22}} & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\quad
\begin{bmatrix}
w^l_{22} & w^l_{21} \\
w^l_{12} & w^l_{11}
\end{bmatrix}
=
\begin{bmatrix}
\frac{\partial E}{\partial h^{l-1}_{11}} & \frac{\partial E}{\partial h^{l-1}_{12}} & \frac{\partial E}{\partial h^{l-1}_{13}} \\
\frac{\partial E}{\partial h^{l-1}_{21}} & \frac{\partial E}{\partial h^{l-1}_{22}} & \frac{\partial E}{\partial h^{l-1}_{23}} \\
\frac{\partial E}{\partial h^{l-1}_{31}} & \frac{\partial E}{\partial h^{l-1}_{32}} & \frac{\partial E}{\partial h^{l-1}_{33}}
\end{bmatrix}
$$

Padded $\partial E/\partial H^l$ Rotated W^l $\partial E/\partial H^{l-1}$

Imagine inverting the receptive field!

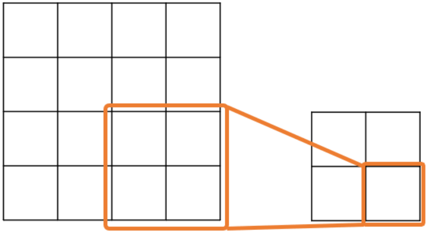
Backpropagation for pooling

- Max function: $m = \max(a, b)$
- $\frac{\partial m}{\partial a} = \begin{cases} 1 \text{ if } a > b, \\ 0 \text{ else.} \end{cases}$ $\quad \frac{\partial m}{\partial b} = \begin{cases} 1 \text{ if } b > a, \\ 0 \text{ else.} \end{cases}$



Backpropagation for pooling

- Max function: $m = \max(a, b)$
- $\frac{\partial m}{\partial a} = \begin{cases} 1 \text{ if } a > b, \\ 0 \text{ else.} \end{cases}$ $\quad \frac{\partial m}{\partial b} = \begin{cases} 1 \text{ if } b > a, \\ 0 \text{ else.} \end{cases}$



H^{l-1} $\quad$ H^l

$$\begin{array}{|c|c|c|c|} \hline 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \hline \end{array}$$

$$\frac{\partial H^l}{\partial H^{l-1}}$$

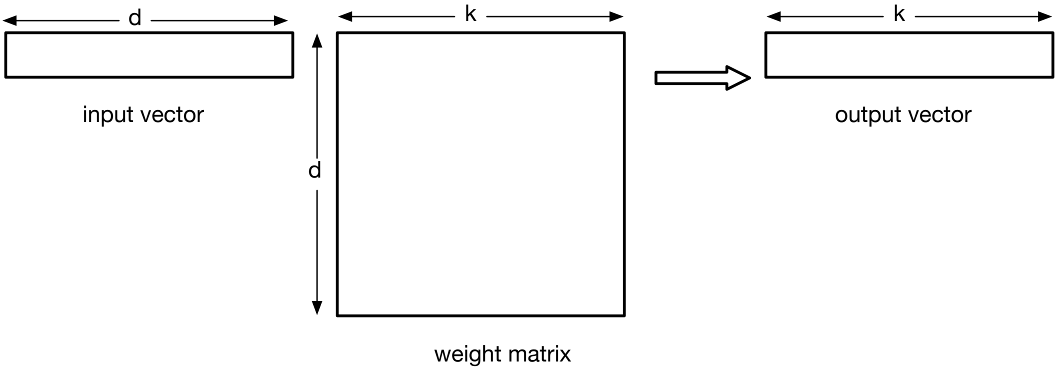
Backpropagation for pooling

- Max function: $m = \max(a, b)$
- $\frac{\partial m}{\partial a} = \begin{cases} 1 \text{ if } a > b, \\ 0 \text{ else.} \end{cases} \qquad \frac{\partial m}{\partial b} = \begin{cases} 1 \text{ if } b > a, \\ 0 \text{ else.} \end{cases}$

$$\underset{\frac{\partial E}{\partial H^l}}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}} \times \underset{\frac{\partial H^l}{\partial H^{l-1}}}{\begin{array}{|c|c|c|c|}\hline 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \hline \end{array}} = \underset{\frac{\partial E}{\partial H^{l-1}}}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}}$$

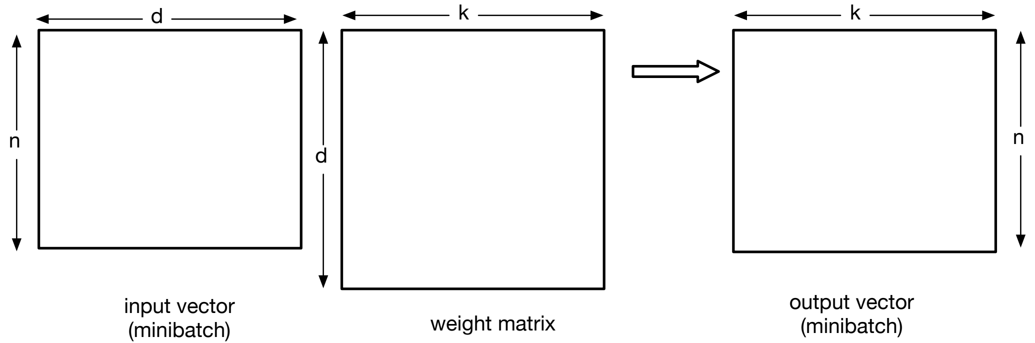
Implementing fully-connected networks

Example at a time:

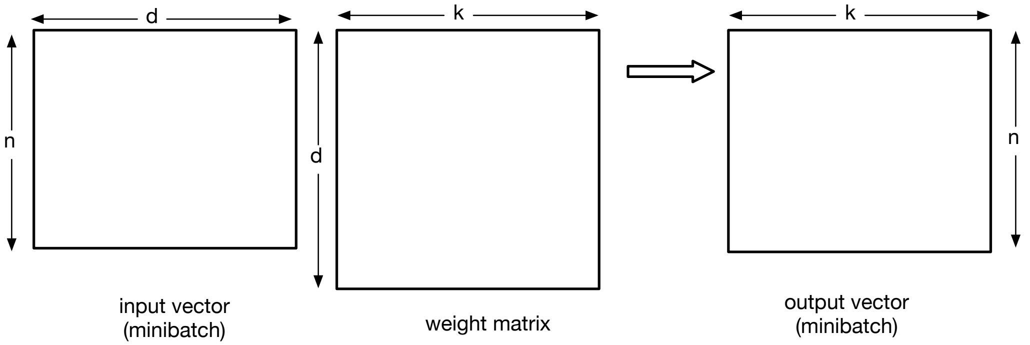


Implementing fully-connected networks

Minibatch:



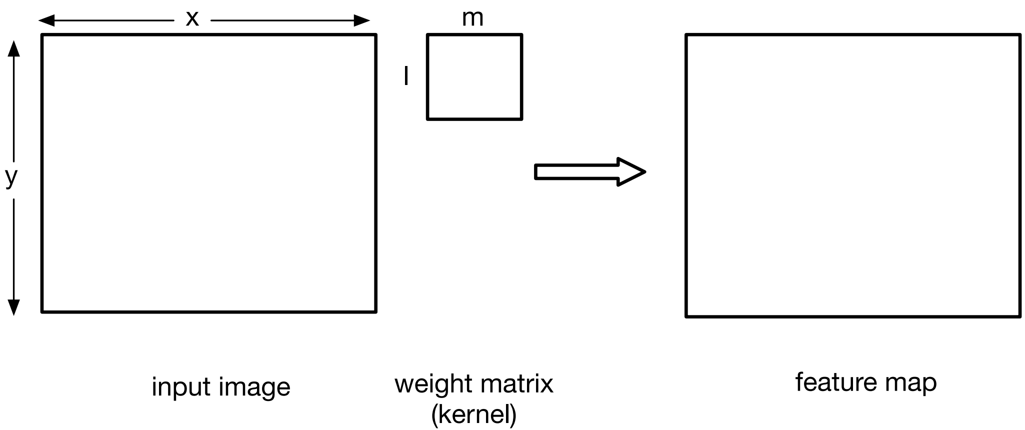
Minibatch:



input dimension x minibatch: Represent each layer as a 2-dimension matrix, where each row corresponds to a training example, and the number of minibatch examples is the number of rows

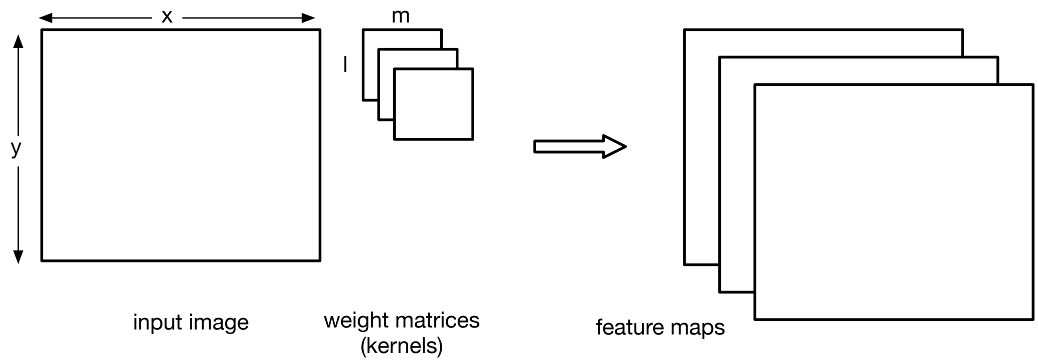
Implementing Convolutional Networks

Example at a time, single input image, single feature map:

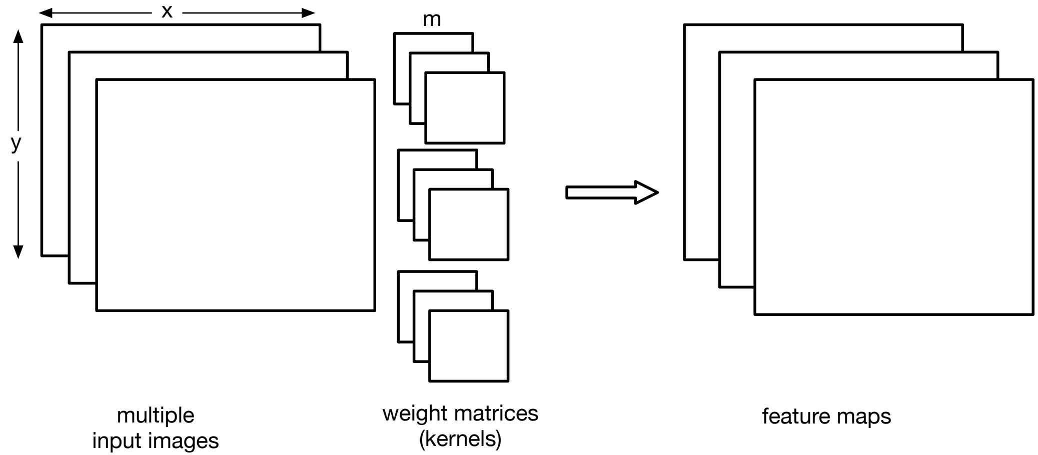


Implementing Convolutional Networks

Example at a time, single input image, multiple feature map:

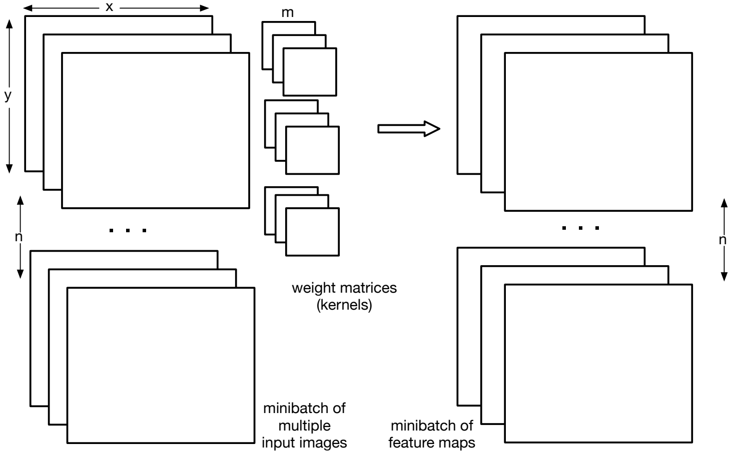


Example at a time, multiple input images, multiple feature map:



Implementing Convolutional Networks

Minibatch, multiple input images, multiple feature map:



Implementing Convolutional Networks

- Inputs / layer values:
  - Each input image (and convlutional and pooling layer) is 2-dimensions (x,y)
  - If we have multiple feature maps, then that is a third dimension
  - And the minibatch adds a fourth dimension
  - Thus we represent each input (layer values) using a 4-dimension *tensor* (array): (minibatch-size, num-fmaps, x, y)
- Weight matrices (kernels)
  - Each weight matrix used to scan across an image has 2 spatial dimensions (x,y)
  - If there are multiple feature maps to be computed, then that is a third dimension
  - Multiple input feature maps adds a fourth dimension
  - Thus the weight matrices are also represented using a 4-dimension tensor: (F_{in}, F_{out}, x, y)

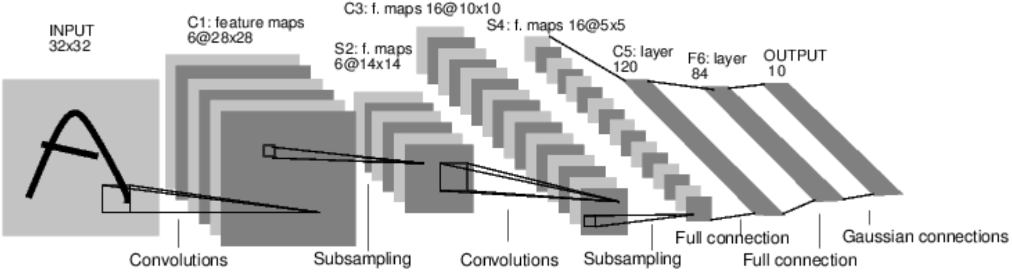
4D tensors in numpy

Both forward and back prop thus involves multiplying 4D tensors. There are various ways to do this:

- Explicitly loop over the dimensions: this results in simpler code, but can be inefficient. Although using cython to compile the loops as C can speed things up
- Serialisation: By replicating input patches and weight matrices, it is possible to convert the required 4D tensor multiplications into a large dot product. Requires careful manipulation of indices!
- Convolutions: use explicit convolution functions for forward and back prop, rotating for the backprop

Deep convolutional networks

LeNet5 (LeCun et al, 1997)



- 2 convolutional layers {C1, C3} + non-linearity
- 2 average pooling {S2, S4}
- 2 fully connected hidden layer (no weight sharing) {C5, F6}
- Softmax classifier layer

ImageNet Classification ("AlexNet")

Krizhevsky, Sutskever and Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS'12.

http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

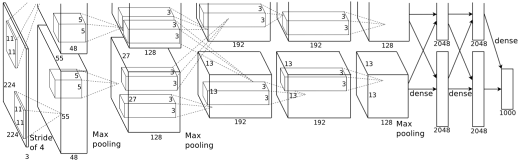


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

- 5 convolutional layers + non-linearity (ReLU)
- 3 max pooling layers
- 2 fully connected hidden layer
- Softmax classifier layer

ImageNet Classification (“VGGNet”)

Simonyan and Zisserman, “Very Deep Convolutional Networks for Large-Scale Visual Recognition”, ILSVRC-2014. http://www.robots.ox.ac.uk/~vgg/research/very_deep/

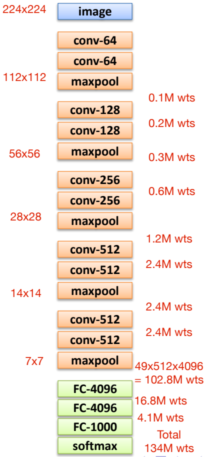
Network Design

Key design choices:

- 3x3 conv. kernels – very small
- conv. stride 1 – no loss of information

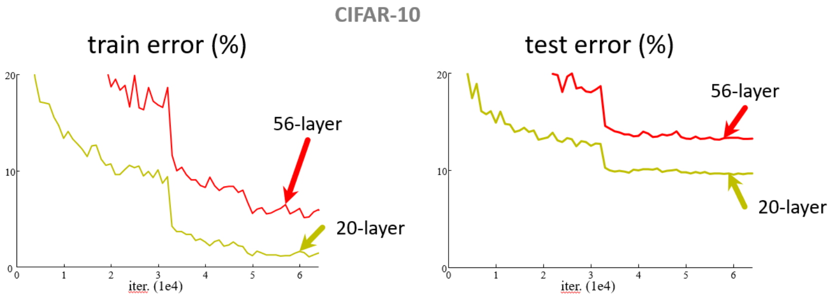
Other details:

- Rectification (ReLU) non-linearity
- 5 max-pool layers (x2 reduction)
- no normalisation
- 3 fully-connected (FC) layers



Simply stacking more layers?

He et al, "Deep Residual Learning for Image Recognition", CVPR-2016.
http://arxiv.org/abs/1512.03385

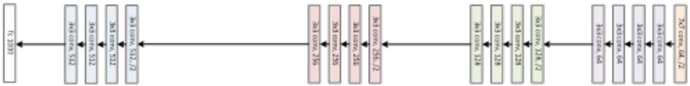


56-layer net has higher training error and test error than 20-layer net!

Deep Residual Learning ("ResNets")

He et al, "Deep Residual Learning for Image Recognition", CVPR-2016.

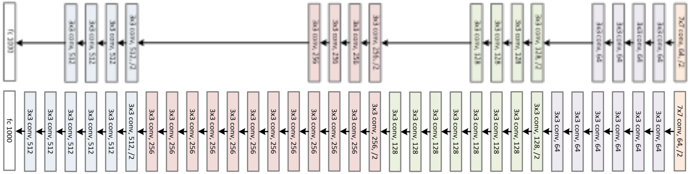
http://arxiv.org/abs/1512.03385



Deep Residual Learning ("ResNets")

He et al, "Deep Residual Learning for Image Recognition", CVPR-2016.

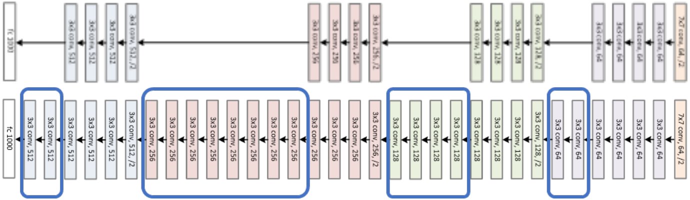
http://arxiv.org/abs/1512.03385



Deep Residual Learning ("ResNets")

He et al, "Deep Residual Learning for Image Recognition", CVPR-2016.

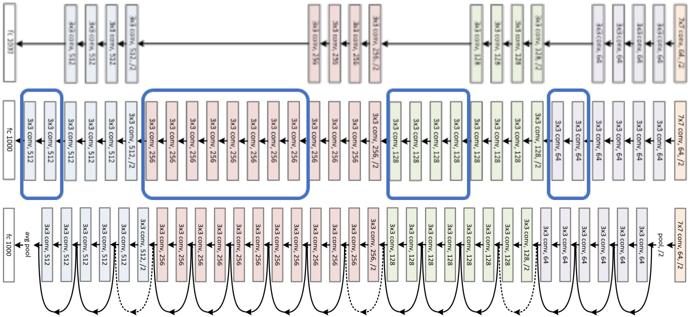
http://arxiv.org/abs/1512.03385



Deep Residual Learning ("ResNets")

He et al, "Deep Residual Learning for Image Recognition", CVPR-2016.

http://arxiv.org/abs/1512.03385



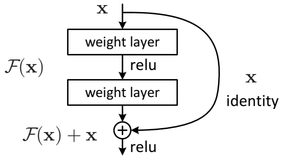


Figure 2. Residual learning: a building block.

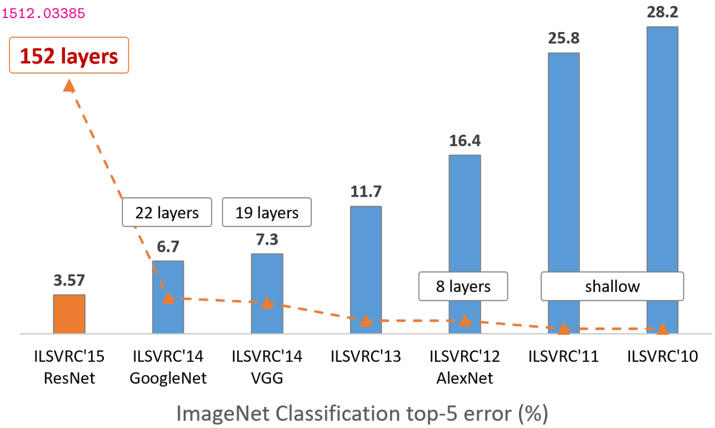
A solution by construction:

- original layers: copied from a learned shallower model
- extra layers: set as identity
- at least the same training error

Deep Residual Learning ("ResNets")

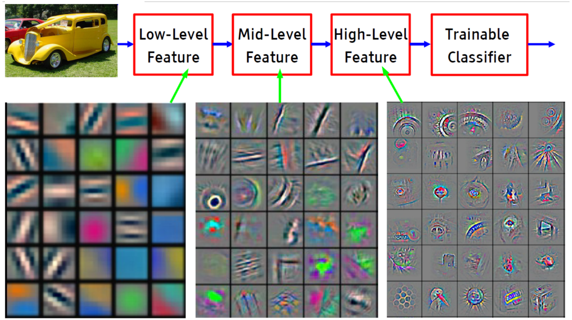
He et al, "Deep Residual Learning for Image Recognition", CVPR-2016.

http://arxiv.org/abs/1512.03385



Hierarchical Representations

Pixel $\rightarrow$ edge $\rightarrow$ texton $\rightarrow$ motif $\rightarrow$ part $\rightarrow$ object



Zeiler & Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV’14.

https://cs.nyu.edu/~fergus/papers/zeilerECCV2014.pdf

Slide credits: Lecun & Ranzato

Summary

- Convolutional networks include local receptive fields, weight sharing, and pooling leading
- Backprop training can also be implemented as a "reverse" convolutional layer (with the weight matrix rotated)
- Implement using 4D tensors:
  - Inputs / Layer values: minibatch-size, number-fmaps, x, y
  - Weights: F_{in}, F_{out}, x, y
  - Arguments: stride, kernel size, dilation, filter groups
- Reading:
  Goodfellow et al, *Deep Learning* (ch 9)
  http://www.deeplearningbook.org/contents/convnets.html