

THE UNIVERSITY of EDINBURGH
informatics

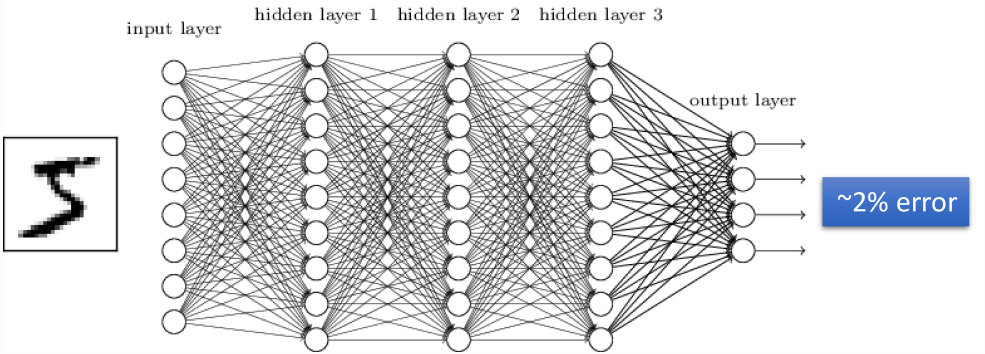
Machine Learning Practical (MLP)

Convolutional Networks

Changjian Li | Pavlos Andreadis

Recap: Fully-connected network for MNIST

On MNIST, we can get about 2% error (or even better) using these kind of networks



(image from: Michael Nielsen, *Neural Networks and Deep Learning*,
`http://neuralnetworksanddeeplearning.com/chap6.html`)

How about more complex images?

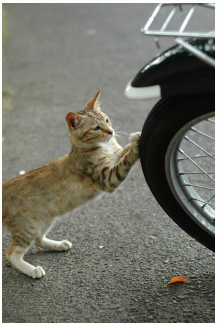




image credit: COCO dataset

Generic object classification



- High resolution images
- Large variations in appearance and pose
- Objects are not always centered
- Objects are in different scales
- Background and clutter
- Occlusions

Today's lecture

Convolutional neural networks (CNNs)

- Convolution operation
- Convolutional layer
- Subsampling (pooling)
- CNN architectures

Fully-connected network in high dimension

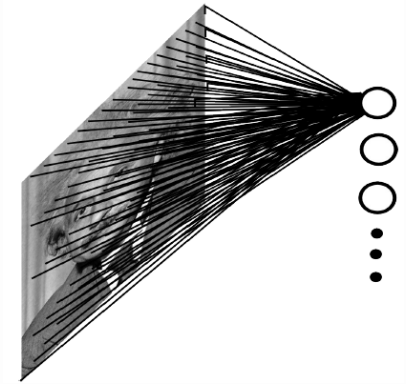


image credit: Lecun & Ranzato

For an input size of 200×200, a deep network with one hidden layer (1000 hidden units) has

- # input units is 40,000
- # connections is 40,000,000
- # parameters is 40,000,000

Fully-connected network in high dimension

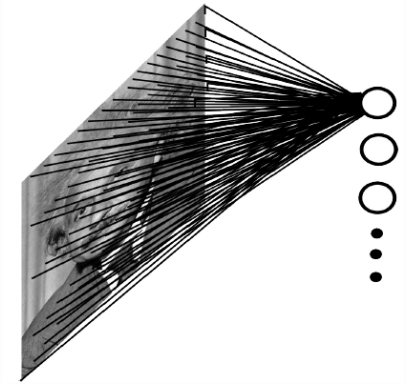


image credit: Lecun & Ranzato

For an input size of 200×200, a deep network with one hidden layer (1000 hidden units) has

- # input units is 40,000
- # connections is 40,000,000
- # parameters is 40,000,000

Observations:

- Too many parameters to learn!
- Spatial (2-D) structure of input image is ignored
- No easy way of learning the same feature at different coordinates

Convolution operation

- Convolution is an operation on two functions of a real- valued argument

$$s(t) = \int x(a)w(t-a)da$$

- Denoted with asterisk ($*$)

$$s(t) = (x * w)(t)$$

- Discrete convolution

$$s(t) = \sum_{a=-\infty}^{a=\infty} x(a)w(t-a)da$$

- Convolution in multiple dimensions

$$s(i,j) = (\boldsymbol{x} * \boldsymbol{w})(i,j) = \sum_{k=0}^{m-1}\sum_{l=0}^{n-1} x_{k,l}w_{i-k,j-l}$$

Convolution operation

- Multiple dimensions

$$s(i,j) = (\boldsymbol{x} * \boldsymbol{w})(i,j) = \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} x_{k,l} w_{i-k,j-l}$$

- Convolution is commutative

$$s(i,j) = (\boldsymbol{w} * \boldsymbol{x})(i,j) = \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} x_{i-k,j-l} w_{k,l}$$

Convolution operation

- Multiple dimensions

$$s(i,j) = (\boldsymbol{x} * \boldsymbol{w})(i,j) = \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} x_{k,l} w_{i-k,j-l}$$

- Convolution is commutative

$$s(i,j) = (\boldsymbol{w} * \boldsymbol{x})(i,j) = \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} x_{i-k,j-l} w_{k,l}$$

- Convolutions in deep learning libraries are actually **cross-correlation** ($\otimes$)

$$s(i,j) = (\boldsymbol{w} \otimes \boldsymbol{x})(i,j) = \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} x_{i+k,j+l} w_{k,l}$$

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

Convolution vs Cross-correlation

- In machine learning the network learns the kernel appropriate to its orientation – so if convolution is implemented with a flipped kernel, it will learn that it is a flipped implementation
- So it is OK to use an efficient (flipped) implementation of convolution for convolutional layers
- In convolutional networks, we also use a bias term:

$$s(i,j) = (\boldsymbol{w} \otimes \boldsymbol{x})(i,j) + b = \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} x_{i+k,j+l} w_{k,l} + b$$

Example: convolution operation

- $\boldsymbol{h} = f(\underbrace{\boldsymbol{x} \otimes \boldsymbol{w}}_{\boldsymbol{z}})$ where f is an activation function
- $\boldsymbol{h}$ is called **feature map**

$$\underbrace{\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} & z_{15} \\ z_{21} & z_{22} & z_{23} & z_{24} & z_{25} \\ z_{31} & z_{32} & z_{33} & z_{34} & z_{35} \\ z_{41} & z_{42} & z_{43} & z_{44} & z_{45} \\ z_{51} & z_{52} & z_{53} & z_{54} & z_{55} \end{pmatrix}}_{\boldsymbol{z}} = \underbrace{\begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & x_{17} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} & x_{27} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} & x_{37} \\ x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} & x_{47} \\ x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} & x_{57} \\ x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} & x_{67} \\ x_{71} & x_{72} & x_{73} & x_{74} & x_{75} & x_{76} & x_{77} \end{pmatrix}}_{\boldsymbol{x}\text{: Input}} \otimes \underbrace{\begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix}}_{\boldsymbol{w}\text{: Kernel}}$$

Example: convolution operation

$$\begin{pmatrix} z_{00} & z_{01} & z_{02} & z_{03} & z_{04} \\ z_{10} & z_{11} & z_{12} & z_{13} & z_{14} \\ z_{20} & z_{21} & z_{22} & z_{23} & z_{24} \\ z_{30} & z_{31} & z_{32} & z_{33} & z_{34} \\ z_{40} & z_{41} & z_{42} & z_{43} & z_{44} \end{pmatrix} \boxed{=} \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix} \boxed{\otimes} \begin{pmatrix} w_{00} & w_{01} & w_{02} \\ w_{10} & w_{11} & w_{12} \\ w_{20} & w_{21} & w_{22} \end{pmatrix}$$

Example: convolution operation

$$z_{00} = \sum_{k=0}^{2} \sum_{l=0}^{2} x_{k,l} w_{kl} + b$$

$$\begin{pmatrix} z_{00} & z_{01} & z_{02} & z_{03} & z_{04} \\ z_{10} & z_{11} & z_{12} & z_{13} & z_{14} \\ z_{20} & z_{21} & z_{22} & z_{23} & z_{24} \\ z_{30} & z_{31} & z_{32} & z_{33} & z_{34} \\ z_{40} & z_{41} & z_{42} & z_{43} & z_{44} \end{pmatrix} = \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix} \otimes \begin{pmatrix} w_{00} & w_{01} & w_{02} \\ w_{10} & w_{11} & w_{12} \\ w_{20} & w_{21} & w_{22} \end{pmatrix}$$

Example: convolution operation

$$z_{01} = \sum_{k=0}^{2} \sum_{l=0}^{2} x_{k,1+l} w_{kl} + b$$

$$\begin{pmatrix} z_{00} & z_{01} & z_{02} & z_{03} & z_{04} \\ z_{10} & z_{11} & z_{12} & z_{13} & z_{14} \\ z_{20} & z_{21} & z_{22} & z_{23} & z_{24} \\ z_{30} & z_{31} & z_{32} & z_{33} & z_{34} \\ z_{40} & z_{41} & z_{42} & z_{43} & z_{44} \end{pmatrix} = \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix} \otimes \begin{pmatrix} w_{00} & w_{01} & w_{02} \\ w_{10} & w_{11} & w_{12} \\ w_{20} & w_{21} & w_{22} \end{pmatrix}$$

Example: convolution operation

$$z_{04} = \sum_{k=0}^{2} \sum_{l=0}^{2} x_{k,4+l} w_{kl} + b$$

$$\begin{pmatrix} z_{00} & z_{01} & z_{02} & z_{03} & z_{04} \\ z_{10} & z_{11} & z_{12} & z_{13} & z_{14} \\ z_{20} & z_{21} & z_{22} & z_{23} & z_{24} \\ z_{30} & z_{31} & z_{32} & z_{33} & z_{34} \\ z_{40} & z_{41} & z_{42} & z_{43} & z_{44} \end{pmatrix} = \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix} \otimes \begin{pmatrix} w_{00} & w_{01} & w_{02} \\ w_{10} & w_{11} & w_{12} \\ w_{20} & w_{21} & w_{22} \end{pmatrix}$$

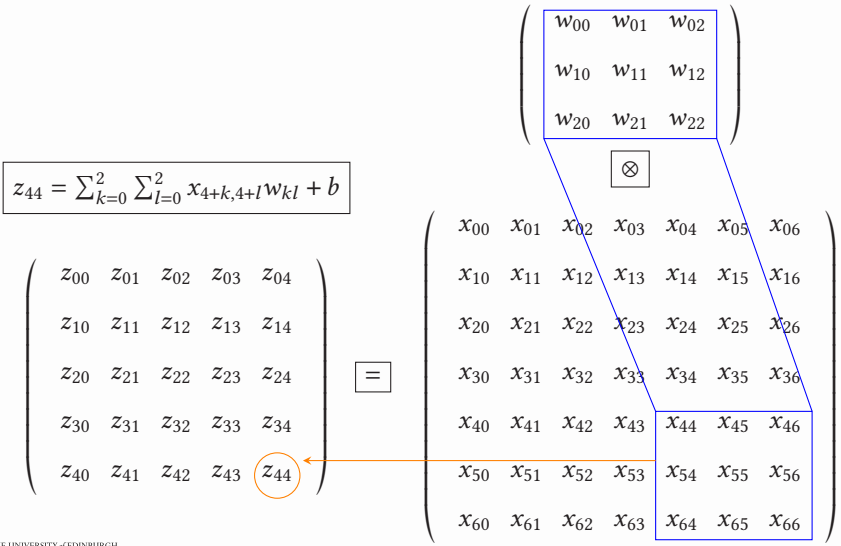
Example: convolution operation

$$z_{10} = \sum_{k=0}^{2} \sum_{l=0}^{2} x_{1+k,l} w_{kl} + b$$

$$\begin{pmatrix} z_{00} & z_{01} & z_{02} & z_{03} & z_{04} \\ z_{10} & z_{11} & z_{12} & z_{13} & z_{14} \\ z_{20} & z_{21} & z_{22} & z_{23} & z_{24} \\ z_{30} & z_{31} & z_{32} & z_{33} & z_{34} \\ z_{40} & z_{41} & z_{42} & z_{43} & z_{44} \end{pmatrix} = \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix} \otimes \begin{pmatrix} w_{00} & w_{01} & w_{02} \\ w_{10} & w_{11} & w_{12} \\ w_{20} & w_{21} & w_{22} \end{pmatrix}$$

Example: convolution operation

$$z_{44} = \sum_{k=0}^{2} \sum_{l=0}^{2} x_{4+k,4+l} w_{kl} + b$$



Example: convolution operation

Number of ...

- input units are 7×7
- weights is $3 \times 3 + 1$ (9 for kernel + 1 for bias)
- hidden units is 5×5
- connections is $5 \times 5 \times 3 \times 3$

Example: convolution operation

Number of ...

- input units are 7×7
- weights is $3 \times 3 + 1$ (9 for kernel + 1 for bias)
- hidden units is 5×5
- connections is $5 \times 5 \times 3 \times 3$

Properties

- Weights (kernel) are shared across all hidden units
- Spatial correspondence between pixels and hidden units
  ("2D matrix of hidden units"="feature map")
- Translation invariance: extract same features irrespective of where an image patch is located in the input

Question

If $\boldsymbol{x}$ is 7×7 and $\boldsymbol{w}$ is 3×3, then the feature map $\boldsymbol{h}$ is 5×5.

If $\boldsymbol{x}$ is 7×7 and $\boldsymbol{w}$ is 5×5, what will be the dimensions of $\boldsymbol{h}$?

(a) 3×3, (b) 5×5, (c) 7×7

Question

If $\boldsymbol{x}$ is 7×7 and $\boldsymbol{w}$ is 5×5, what will be the dimensions of $\boldsymbol{h}$?

$$\begin{pmatrix} w_{00} & w_{01} & w_{02} & w_{03} & w_{04} \\ w_{10} & w_{11} & w_{12} & w_{13} & w_{14} \\ w_{20} & w_{21} & w_{22} & w_{23} & w_{24} \\ w_{30} & w_{31} & w_{32} & w_{33} & w_{34} \\ w_{40} & w_{41} & w_{42} & w_{43} & w_{44} \end{pmatrix}$$

$$\boxed{\otimes}$$

$$\begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix}$$

$$\boxed{=}$$

$$\begin{pmatrix} z_{00} & z_{01} & z_{02} \\ z_{10} & z_{11} & z_{12} \\ z_{20} & z_{21} & z_{22} \end{pmatrix}$$

Question

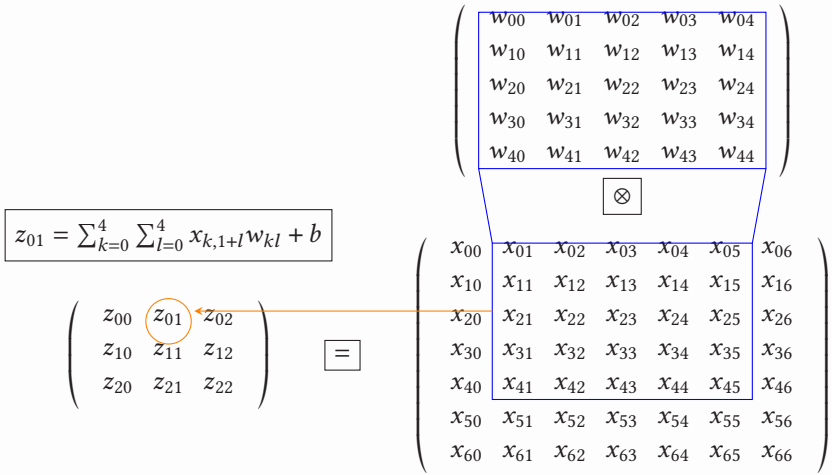
If $\boldsymbol{x}$ is 7×7 and $\boldsymbol{w}$ is 5×5, what will be the dimensions of $\boldsymbol{h}$?

$$z_{00} = \sum_{k=0}^{4} \sum_{l=0}^{4} x_{k,l} w_{kl} + b$$

$$\begin{pmatrix} z_{00} & z_{01} & z_{02} \\ z_{10} & z_{11} & z_{12} \\ z_{20} & z_{21} & z_{22} \end{pmatrix} = \begin{pmatrix} w_{00} & w_{01} & w_{02} & w_{03} & w_{04} \\ w_{10} & w_{11} & w_{12} & w_{13} & w_{14} \\ w_{20} & w_{21} & w_{22} & w_{23} & w_{24} \\ w_{30} & w_{31} & w_{32} & w_{33} & w_{34} \\ w_{40} & w_{41} & w_{42} & w_{43} & w_{44} \end{pmatrix} \otimes \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix}$$

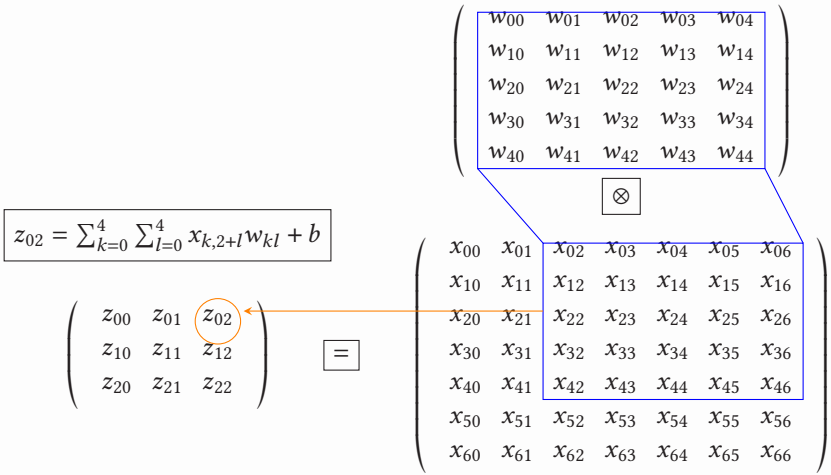
Question

If $\boldsymbol{x}$ is 7×7 and $\boldsymbol{w}$ is 5×5, what will be the dimensions of $\boldsymbol{h}$?



Question

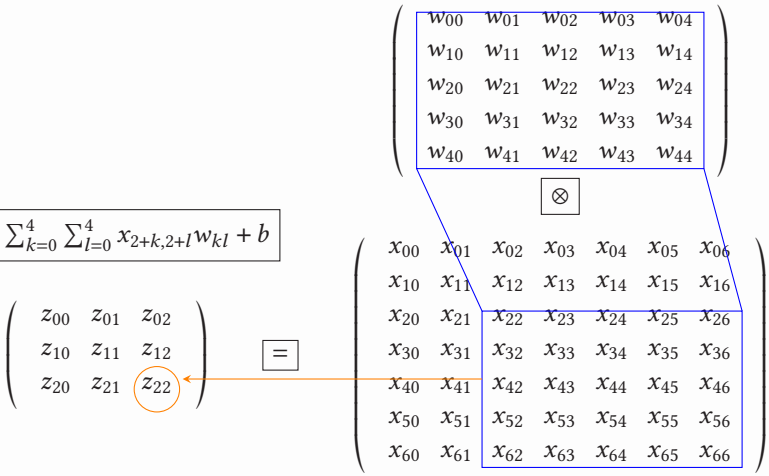
If $\boldsymbol{x}$ is 7×7 and $\boldsymbol{w}$ is 5×5, what will be the dimensions of $\boldsymbol{h}$?



Question

If $\boldsymbol{x}$ is 7×7 and $\boldsymbol{w}$ is 5×5, what will be the dimensions of $\boldsymbol{h}$?

$$z_{22} = \sum_{k=0}^{4} \sum_{l=0}^{4} x_{2+k,2+l} w_{kl} + b$$



Calculating the feature map size

Q. If $\boldsymbol{x}$ is $q \times r$ and $\boldsymbol{w}$ is $m \times n$ dimensional, what is the output dimensionality?

A. $(q - m + 1) \times (r - n + 1)$

Padding

Q. What can we do to get 7×7 feature map size?

$$\begin{pmatrix} w_{00} & w_{01} & w_{02} \\ w_{10} & w_{11} & w_{12} \\ w_{20} & w_{21} & w_{22} \end{pmatrix}$$

$$\boxed{\otimes}$$

$$\begin{pmatrix} z_{00} & z_{01} & z_{02} & z_{03} & z_{04} & z_{05} & z_{06} \\ z_{10} & z_{11} & z_{12} & z_{13} & z_{14} & z_{15} & z_{16} \\ z_{20} & z_{21} & z_{22} & z_{23} & z_{24} & z_{25} & z_{26} \\ z_{30} & z_{31} & z_{32} & z_{33} & z_{34} & z_{35} & z_{36} \\ z_{40} & z_{41} & z_{42} & z_{43} & z_{44} & z_{45} & z_{46} \\ z_{50} & z_{51} & z_{52} & z_{53} & z_{54} & z_{55} & z_{56} \\ z_{60} & z_{61} & z_{62} & z_{63} & z_{64} & z_{65} & z_{66} \end{pmatrix}$$

$$\boxed{=}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} & 0 \\ 0 & x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & 0 \\ 0 & x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} & 0 \\ 0 & x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} & 0 \\ 0 & x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} & 0 \\ 0 & x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} & 0 \\ 0 & x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Padding

Q. What can we do to get 7×7 feature map size?

$$\begin{pmatrix} w_{00} & w_{01} & w_{02} \\ w_{10} & w_{11} & w_{12} \\ w_{20} & w_{21} & w_{22} \end{pmatrix}$$

$$\boxed{\otimes}$$

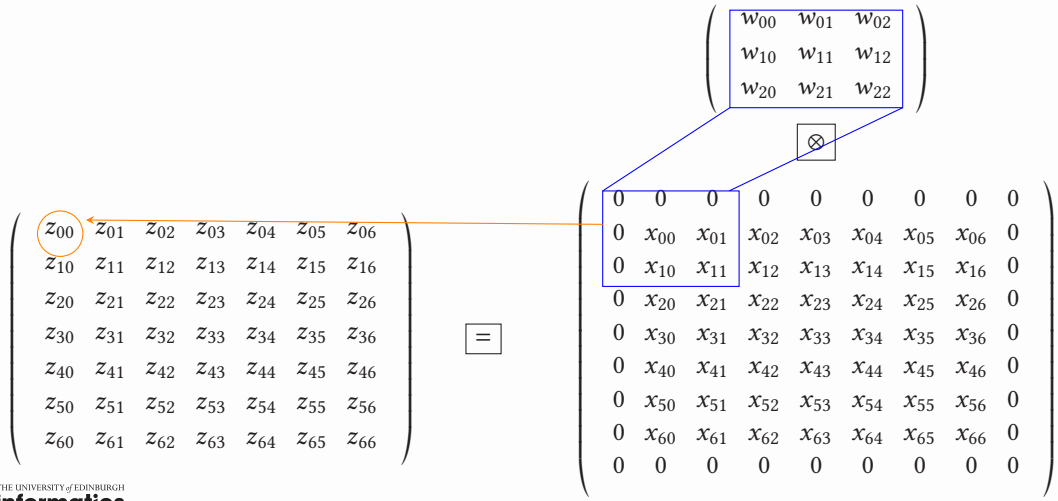
$$\begin{pmatrix} z_{00} & z_{01} & z_{02} & z_{03} & z_{04} & z_{05} & z_{06} \\ z_{10} & z_{11} & z_{12} & z_{13} & z_{14} & z_{15} & z_{16} \\ z_{20} & z_{21} & z_{22} & z_{23} & z_{24} & z_{25} & z_{26} \\ z_{30} & z_{31} & z_{32} & z_{33} & z_{34} & z_{35} & z_{36} \\ z_{40} & z_{41} & z_{42} & z_{43} & z_{44} & z_{45} & z_{46} \\ z_{50} & z_{51} & z_{52} & z_{53} & z_{54} & z_{55} & z_{56} \\ z_{60} & z_{61} & z_{62} & z_{63} & z_{64} & z_{65} & z_{66} \end{pmatrix}$$

$$\boxed{=}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} & 0 \\ 0 & x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & 0 \\ 0 & x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} & 0 \\ 0 & x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} & 0 \\ 0 & x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} & 0 \\ 0 & x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} & 0 \\ 0 & x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Padding

Q. What can we do to get 7×7 feature map size?



Stride $s = 2$

$$\begin{pmatrix} w_{00} & w_{01} & w_{02} \\ w_{10} & w_{11} & w_{12} \\ w_{20} & w_{21} & w_{22} \end{pmatrix}$$

$$\boxed{\otimes}$$

$$\begin{pmatrix} z_{00} & z_{01} & z_{02} \\ z_{10} & z_{11} & z_{12} \\ z_{20} & z_{21} & z_{22} \end{pmatrix} \quad \boxed{=} \quad \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix}$$

Stride $s = 2$

$$z_{00} = \sum_{k=0}^{2} \sum_{l=0}^{2} x_{k,l} w_{kl} + b$$

$$\begin{pmatrix} z_{00} & z_{01} & z_{02} \\ z_{10} & z_{11} & z_{12} \\ z_{20} & z_{21} & z_{22} \end{pmatrix} = \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix} \otimes \begin{pmatrix} w_{00} & w_{01} & w_{02} \\ w_{10} & w_{11} & w_{12} \\ w_{20} & w_{21} & w_{22} \end{pmatrix}$$

Stride $s = 2$

$$z_{01} = \sum_{k=0}^{2} \sum_{l=0}^{2} x_{k,s+l} w_{kl} + b$$

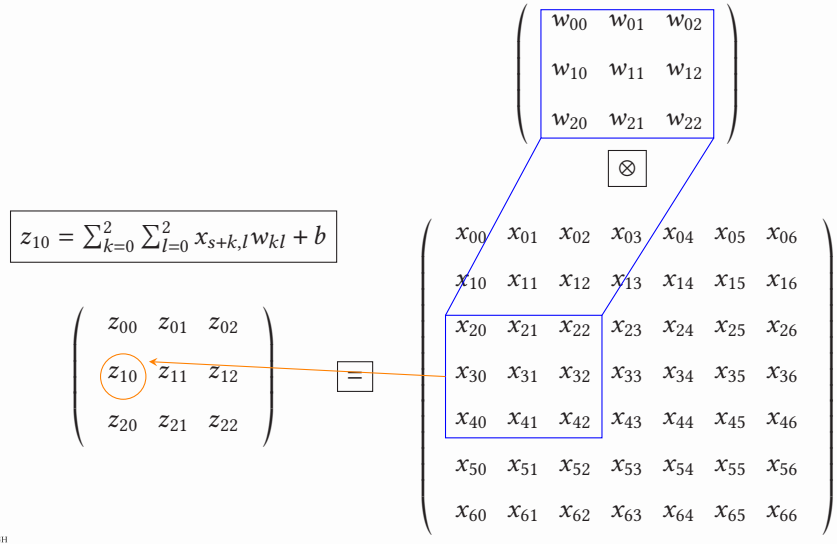
$$\begin{pmatrix} z_{00} & z_{01} & z_{02} \\ z_{10} & z_{11} & z_{12} \\ z_{20} & z_{21} & z_{22} \end{pmatrix} = \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix} \otimes \begin{pmatrix} w_{00} & w_{01} & w_{02} \\ w_{10} & w_{11} & w_{12} \\ w_{20} & w_{21} & w_{22} \end{pmatrix}$$

Stride $s = 2$

$$z_{02} = \sum_{k=0}^{2} \sum_{l=0}^{2} x_{k,2s+l} w_{kl} + b$$

$$\begin{pmatrix} z_{00} & z_{01} & z_{02} \\ z_{10} & z_{11} & z_{12} \\ z_{20} & z_{21} & z_{22} \end{pmatrix} = \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix} \otimes \begin{pmatrix} w_{00} & w_{01} & w_{02} \\ w_{10} & w_{11} & w_{12} \\ w_{20} & w_{21} & w_{22} \end{pmatrix}$$

Stride $s = 2$


$z_{10} = \sum_{k=0}^{2} \sum_{l=0}^{2} x_{s+k,l} w_{kl} + b$
w_{00} w_{01} w_{02}
w_{10} w_{11} w_{12}
w_{20} w_{21} w_{22}
$\otimes$
z_{00} z_{01} z_{02}
z_{10} z_{11} z_{12}
z_{20} z_{21} z_{22}
$=$
x_{00} x_{01} x_{02} x_{03} x_{04} x_{05} x_{06}
x_{10} x_{11} x_{12} x_{13} x_{14} x_{15} x_{16}
x_{20} x_{21} x_{22} x_{23} x_{24} x_{25} x_{26}
x_{30} x_{31} x_{32} x_{33} x_{34} x_{35} x_{36}
x_{40} x_{41} x_{42} x_{43} x_{44} x_{45} x_{46}
x_{50} x_{51} x_{52} x_{53} x_{54} x_{55} x_{56}
x_{60} x_{61} x_{62} x_{63} x_{64} x_{65} x_{66}

Take-home exercise

What is the feature map size when $\boldsymbol{x}$ is $q \times r$ and $\boldsymbol{w}$ is $m \times n$ dimensional, for padding p and stride s?

Receptive field

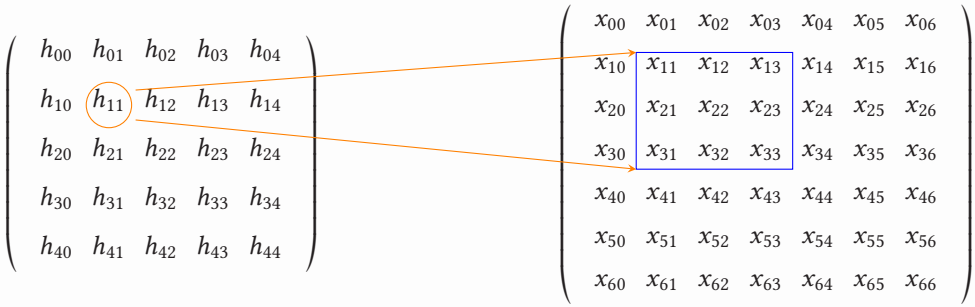
- Biology: The **receptive field** of an individual sensory neuron is the particular region of the sensory space,
- Convolutional networks: The region in the input space that a hidden unit is looking at.

$$\begin{pmatrix} h_{00} & h_{01} & h_{02} & h_{03} & h_{04} \\ h_{10} & h_{11} & h_{12} & h_{13} & h_{14} \\ h_{20} & h_{21} & h_{22} & h_{23} & h_{24} \\ h_{30} & h_{31} & h_{32} & h_{33} & h_{34} \\ h_{40} & h_{41} & h_{42} & h_{43} & h_{44} \end{pmatrix}$$

$$\begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix}$$

Receptive field

- Biology: The **receptive field** of an individual sensory neuron is the particular region of the sensory space,
- Convolutional networks: The region in the input space that a hidden unit is looking at.



Receptive field

$$\boldsymbol{h}^{(2)} = f(\boldsymbol{w}^2 \otimes f(\boldsymbol{w}^1 \otimes \boldsymbol{x}))$$

Assume that $\boldsymbol{w}^1$ and $\boldsymbol{w}^2$ are 3×3 kernels and $\boldsymbol{x}$ is 7×7.

Q. What size is the receptive field of a hidden unit in second convolutional layer?

A. 3, 5, 6, 7?

$$\begin{pmatrix} h^{(2)}_{00} & h^{(2)}_{01} & h^{(2)}_{02} \\ h^{(2)}_{10} & h^{(2)}_{11} & h^{(2)}_{12} \\ h^{(2)}_{20} & h^{(2)}_{21} & h^{(2)}_{22} \end{pmatrix} \quad \begin{pmatrix} h^{(1)}_{00} & h^{(1)}_{01} & h^{(1)}_{02} & h^{(1)}_{03} & h^{(1)}_{04} \\ h^{(1)}_{10} & h^{(1)}_{11} & h^{(1)}_{12} & h^{(1)}_{13} & h^{(1)}_{14} \\ h^{(1)}_{20} & h^{(1)}_{21} & h^{(1)}_{22} & h^{(1)}_{23} & h^{(1)}_{24} \\ h^{(1)}_{30} & h^{(1)}_{31} & h^{(1)}_{32} & h^{(1)}_{33} & h^{(1)}_{34} \\ h^{(1)}_{40} & h^{(1)}_{41} & h^{(1)}_{42} & h^{(1)}_{43} & h^{(1)}_{44} \end{pmatrix} \quad \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix}$$

Receptive field

$$\boldsymbol{h}^{(2)} = f(\boldsymbol{w}^2 \otimes f(\boldsymbol{w}^1 \otimes \boldsymbol{x}))$$

Assume that $\boldsymbol{w}^1$ and $\boldsymbol{w}^2$ are 3×3 kernels and $\boldsymbol{x}$ is 7×7.

Q. What size is the receptive field of a hidden unit in second convolutional layer?

A. 3, 5, 6, 7?

$$\begin{pmatrix} h_{00}^{(2)} & h_{01}^{(2)} & h_{02}^{(2)} \\ h_{10}^{(2)} & h_{11}^{(2)} & h_{12}^{(2)} \\ h_{20}^{(2)} & h_{21}^{(2)} & h_{22}^{(2)} \end{pmatrix} \quad \begin{pmatrix} h_{00}^{(1)} & h_{01}^{(1)} & h_{02}^{(1)} & h_{03}^{(1)} & h_{04}^{(1)} \\ h_{10}^{(1)} & h_{11}^{(1)} & h_{12}^{(1)} & h_{13}^{(1)} & h_{14}^{(1)} \\ h_{20}^{(1)} & h_{21}^{(1)} & h_{22}^{(1)} & h_{23}^{(1)} & h_{24}^{(1)} \\ h_{30}^{(1)} & h_{31}^{(1)} & h_{32}^{(1)} & h_{33}^{(1)} & h_{34}^{(1)} \\ h_{40}^{(1)} & h_{41}^{(1)} & h_{42}^{(1)} & h_{43}^{(1)} & h_{44}^{(1)} \end{pmatrix} \quad \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{pmatrix}$$

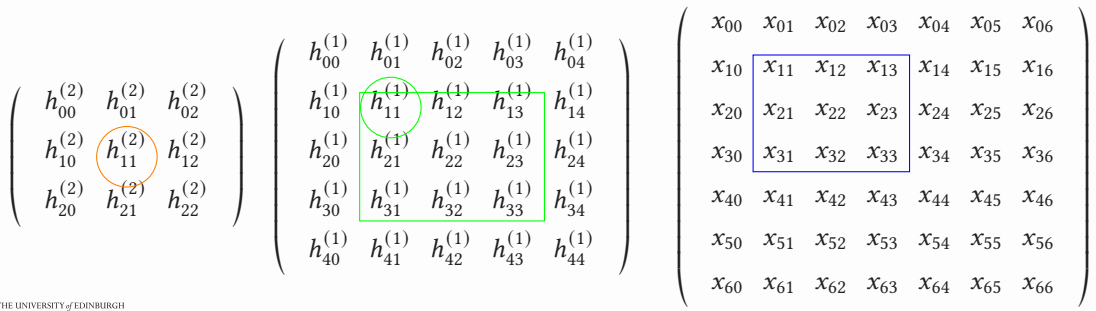
Receptive field

$$\boldsymbol{h}^{(2)} = f(\boldsymbol{w}^2 \otimes f(\boldsymbol{w}^1 \otimes \boldsymbol{x}))$$

Assume that $\boldsymbol{w}^1$ and $\boldsymbol{w}^2$ are 3×3 kernels and $\boldsymbol{x}$ is 7×7.

Q. What size is the receptive field of a hidden unit in second convolutional layer?

A. 3, 5, 6, 7?



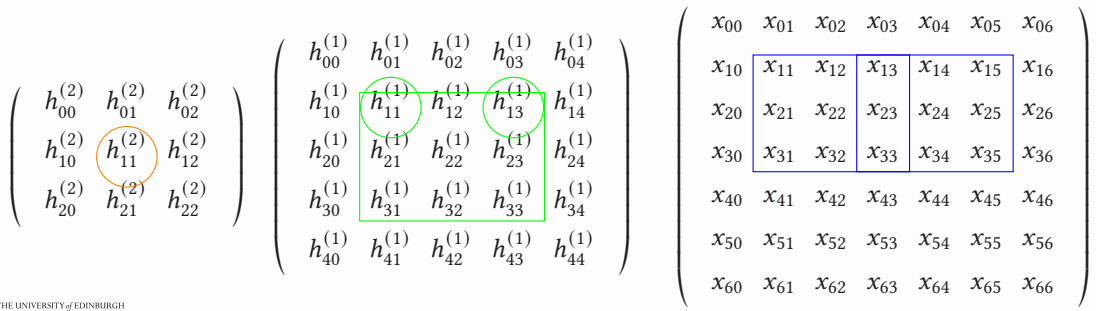
Receptive field

$$\boldsymbol{h}^{(2)} = f(\boldsymbol{w}^2 \otimes f(\boldsymbol{w}^1 \otimes \boldsymbol{x}))$$

Assume that $\boldsymbol{w}^1$ and $\boldsymbol{w}^2$ are 3×3 kernels and $\boldsymbol{x}$ is 7×7.

Q. What size is the receptive field of a hidden unit in second convolutional layer?

A. 3, 5, 6, 7?



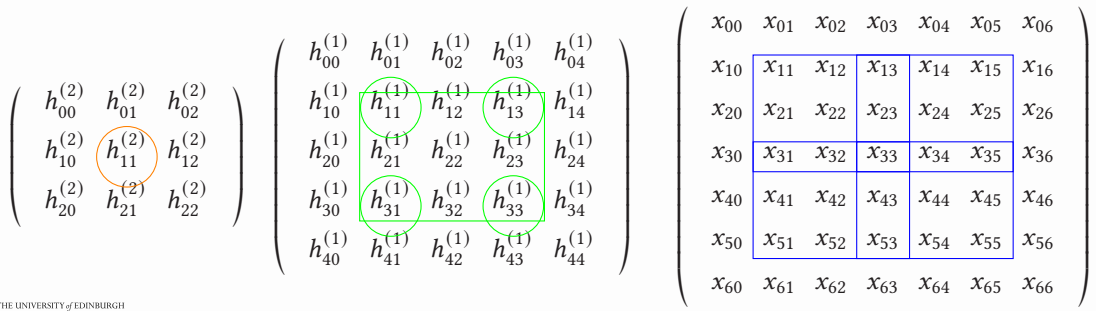
Receptive field

$$\boldsymbol{h}^{(2)} = f(\boldsymbol{w}^2 \otimes f(\boldsymbol{w}^1 \otimes \boldsymbol{x}))$$

Assume that $\boldsymbol{w}^1$ and $\boldsymbol{w}^2$ are 3×3 kernels and $\boldsymbol{x}$ is 7×7.

Q. What size is the receptive field of a hidden unit in second convolutional layer?

A. 3, 5, 6, 7?



Feature maps

- Constrain each hidden unit h_{ij} to extract the same feature by **sharing weights** across the receptive fields
- Local receptive fields with shared weights result in a **feature map** which is a map showing where the feature corresponding to the shared weight matrix (kernel) occurs in the image
- Feature map encodes **translation invariance**: extract the same features irrespective of where an image is located in the input

Multiple output feature maps

$$\begin{pmatrix} h_{000} & h_{001} & \dots & h_{004} \\ h_{010} & h_{011} & \dots & h_{014} \\ \vdots & \vdots & \ddots & \vdots \\ h_{040} & h_{041} & \dots & h_{044} \end{pmatrix} = \begin{pmatrix} x_{00} & x_{01} & \dots & x_{06} \\ x_{10} & x_{11} & \dots & x_{16} \\ \vdots & \vdots & \ddots & \vdots \\ x_{60} & x_{61} & \dots & x_{66} \end{pmatrix} \otimes \begin{pmatrix} w_{000} & w_{001} & w_{002} \\ w_{010} & w_{011} & w_{012} \\ w_{020} & w_{021} & w_{022} \end{pmatrix}$$

$$\begin{pmatrix} h_{100} & h_{101} & \dots & h_{104} \\ h_{110} & h_{111} & \dots & h_{114} \\ \vdots & \vdots & \ddots & \vdots \\ h_{140} & h_{141} & \dots & h_{144} \end{pmatrix} = \begin{pmatrix} x_{00} & x_{01} & \dots & x_{06} \\ x_{10} & x_{11} & \dots & x_{16} \\ \vdots & \vdots & \ddots & \vdots \\ x_{60} & x_{61} & \dots & x_{66} \end{pmatrix} \otimes \begin{pmatrix} w_{100} & w_{101} & w_{102} \\ w_{110} & w_{111} & w_{112} \\ w_{120} & w_{121} & w_{122} \end{pmatrix}$$

h_{ijk}: i-th feature-map, j-th row, k-th column

Multiple output feature maps

- # input units is 7×7
- # feature maps is F_{out}
- # hidden units is $F_{\text{out}} \times (5 \times 5)$
- # of parameters is $F_{\text{out}} \times (3 \times 3 + 1)$
- # of connections is $F_{\text{out}} \times (5 \times 5 \times 3 \times 3)$

Multiple input feature maps (or input channels)

$$\begin{pmatrix} h_{00} & h_{01} & \dots & h_{04} \\ h_{10} & h_{11} & \dots & h_{14} \\ \vdots & \vdots & \ddots & \vdots \\ h_{40} & h_{41} & \dots & h_{44} \end{pmatrix} = \begin{pmatrix} x_{000} & x_{001} & \dots & x_{006} \\ x_{010} & x_{011} & \dots & x_{016} \\ \vdots & \vdots & \ddots & \vdots \\ x_{060} & x_{061} & \dots & x_{066} \end{pmatrix} \otimes \begin{pmatrix} w_{000} & w_{001} & w_{002} \\ w_{010} & w_{011} & w_{012} \\ w_{020} & w_{021} & w_{022} \end{pmatrix}$$

$$+ \begin{pmatrix} x_{100} & x_{101} & \dots & x_{106} \\ x_{110} & x_{111} & \dots & x_{116} \\ \vdots & \vdots & \ddots & \vdots \\ x_{160} & x_{161} & \dots & x_{166} \end{pmatrix} \otimes \begin{pmatrix} w_{100} & w_{101} & w_{102} \\ w_{110} & w_{111} & w_{112} \\ w_{120} & w_{121} & w_{122} \end{pmatrix}$$

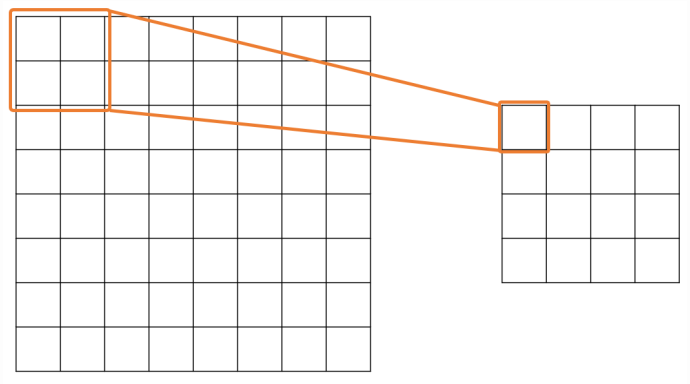
Multiple input feature maps (or input channels)

- # input channels is F_{in}
- # input units is $F_{\text{in}} \times 7 \times 7$
- # hidden units is 5×5
- # parameters is $F_{\text{in}} \times 3 \times 3 + 1$ for bias
- # connections is $F_{\text{in}} \times 5 \times 5 \times 3 \times 3$
- Typically we do not tie weights across feature maps
- Local receptive fields across multiple input images

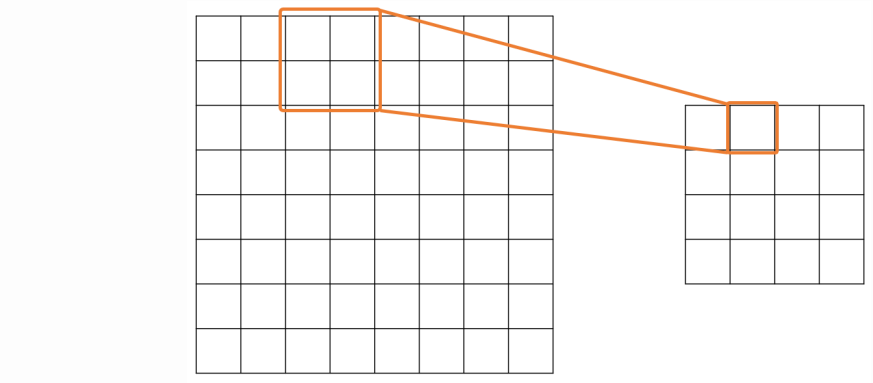
Multiple input and output feature maps

- # input channels is F_{in}
- # feature maps is F_{out}
- # input units is $F_{\text{in}} \times 7 \times 7$
- # hidden units is $F_{\text{out}} \times 5 \times 5$
- # parameters is $F_{\text{in}} \times F_{\text{out}} \times 3 \times 3 + F_{\text{out}}$ for bias
- # connections is $F_{\text{in}} \times F_{\text{out}} \times 5 \times 5 \times 3 \times 3$

Pooling (subsampling)



Pooling (subsampling)



- Similar to convolution, slides over input pixels but no learnable parameters
- Has local receptive field too
- Typical stride $s > 1$

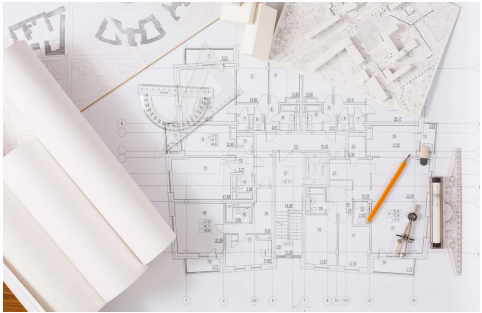
Pooling

- Pooling or subsampling takes a feature map and reduces it in size – e.g. by transforming a set of 2x2 regions to a single unit
- Reduces computation time and memory use
- Pooling functions
  - Max-pooling – takes the maximum value of the units in the region
  - L_p-pooling – take the L_p norm of the units in the region:

$$h' = \left(\sum_{i \in \text{region}} h_i^p \right)^{1/p}$$

  - Average- / Sum-pooling – takes the average / sum value of the pool
- Information reduction – pooling removes precise location information for a feature
- Apply pooling to each feature map separately

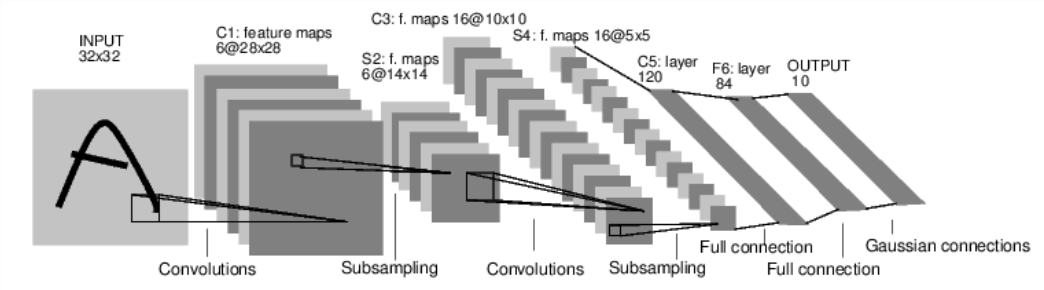
CNN architectures



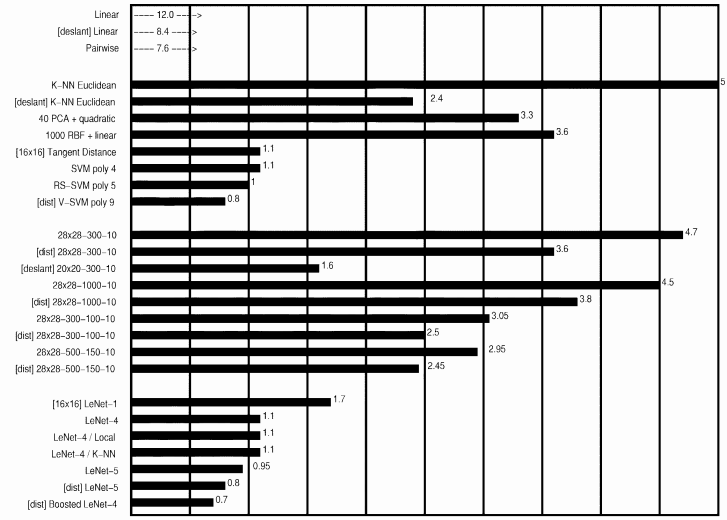
Designing more effective and efficient architectures is an active area of research

- Handcrafting architectures with heuristics and principles
- Automatic neural architecture search

Example: LeNet5 (LeCun et al, 1997)

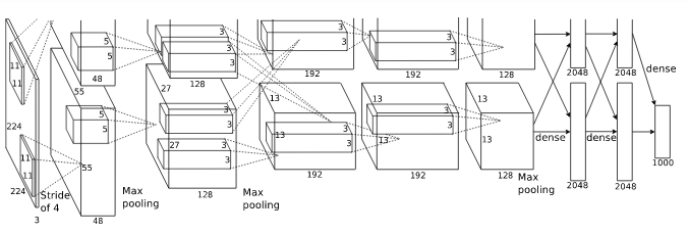


MNIST Results (1997)



THE UNIVERSITY of EDINBURGH
informatics

ImageNet Classification (“AlexNet”) 2012



Dual GPU architecture

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

Error rates in the ILSVRC-2012

- Outperformed the hand-crafted features in object classification
- 5 convolutional layers with ReLU + 3 fully connected layers
- 3 max pooling layers

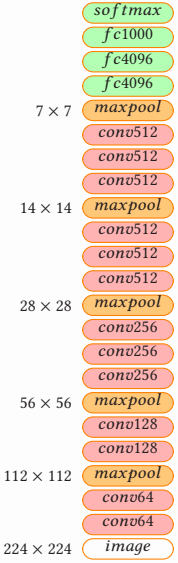
Krizhevsky, Sutskever and Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS’12.

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

ImageNet Classification (“VGGNet”) 2014

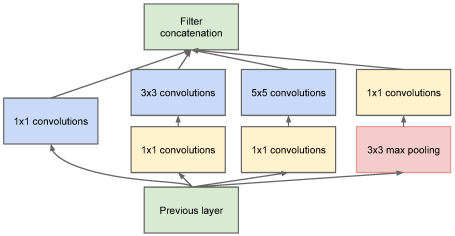
- Deeper than AlexNet, 13 conv. + 3 fc layers
- 3×3 conv. kernels – very small
- Conv. stride 1 – no loss of resolution
- ReLU non-linearity
- Vanishing/exploding gradients problem: train shallower networks and gradually add new layers

Simonyan and Zisserman, “Very Deep Convolutional Networks for Large-Scale Visual Recognition”, ICLR 2015.



ImageNet Classification (“GoogLeNet”) 2014





- objects vary in scale so we should use multiple kernel sizes $(1, 3, 5)$ in parallel
- outputs of the parallel layers are concatenated
- the number of feature maps are reduced with $F_{in} \times F_{in}/2 \times 1 \times 1$ conv. to lower computational load

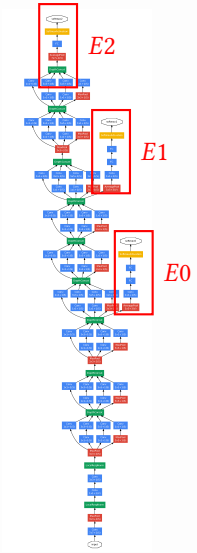
Szegedy et al., “Going deeper with convolutions”, CVPR 2015. Image credits

ImageNet Classification (“GoogLeNet”) 2014

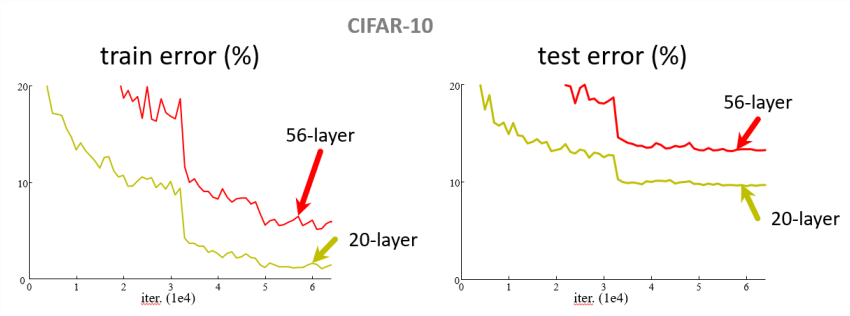
- 22 layer deep (27 including pooling layers)
- Use two “auxiliary classifiers” to middle layers to alleviate the vanishing gradient problem
- Optimizes a weighted sum of “auxiliary errors” ($\alpha_0 = 0.3$ and $\alpha_1 = 0.3$)

$$E = \alpha_0 E_0 + \alpha_1 E_1 + E_2$$

Szegedy et al., “Going deeper with convolutions”, CVPR 2015. Image credits



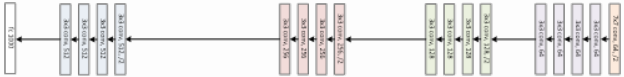
Simply stacking more layers?



Though it can be trained (with the help of Batch Norm),
56-layer net has higher training error and test error than 20-layer net!

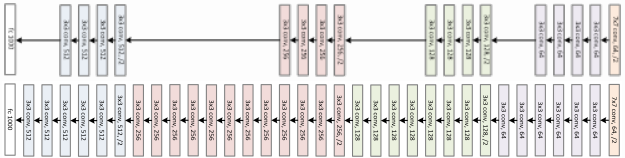
He et al, “Deep Residual Learning for Image Recognition”, CVPR-2016.

Deep Residual Learning (“ResNets”)



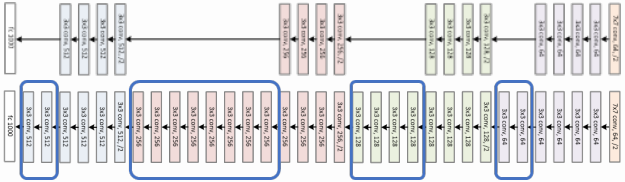
He et al, “Deep Residual Learning for Image Recognition”, CVPR-2016.

Deep Residual Learning (“ResNets”)



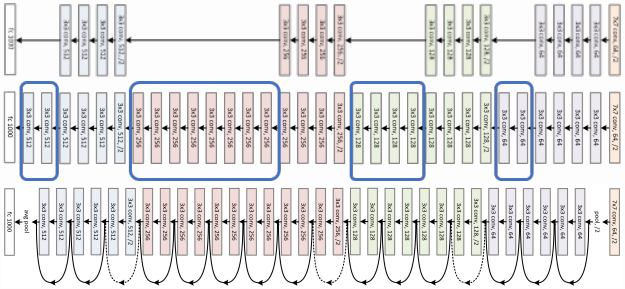
He et al. “Deep Residual Learning for Image Recognition”, CVPR-2016.

Deep Residual Learning (“ResNets”)



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

Deep Residual Learning (“ResNets”)



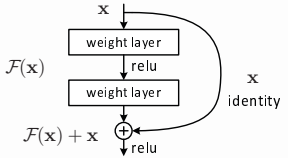


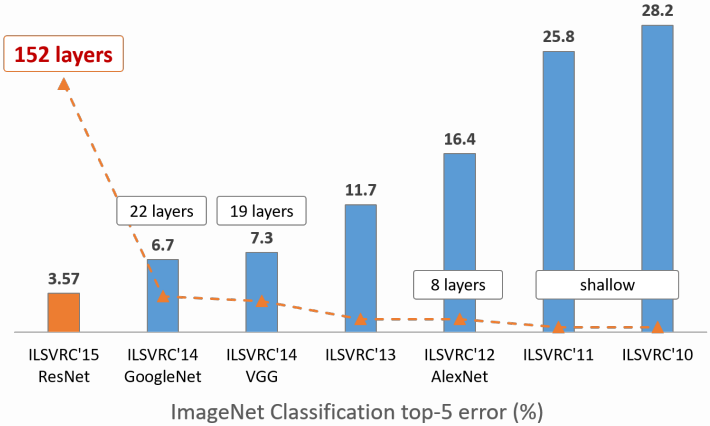
Figure 2. Residual learning: a building block.

A solution by construction:

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

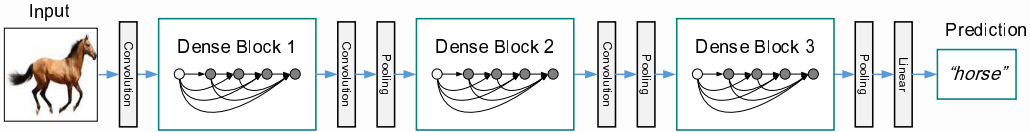
He et al. “Deep Residual Learning for Image Recognition”, CVPR-2016.

Deep Residual Learning (“ResNets”)



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

Densely Connected CNNs (“DenseNets”)

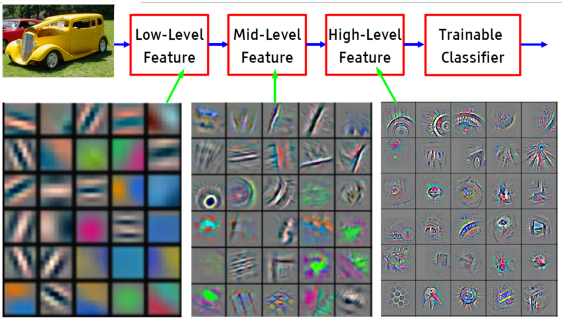


- Connect all layers with each other: each layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent layers
- Requires $L(L+1)/2$ connections in a L-layer network
- Instead uses dense connections within “dense blocks”
- Requires fewer parameters per layer than traditional CNNs, as there is no need to re-learn redundant feature-map

Huang et al, “Densely Connected Convolutional Networks”, CVPR-2017.

Hierarchical Representations

Pixel → edge → texton → motif → part → object



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

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

Slide credits: Lecun & Ranzato

Summary

CNNs include local receptive fields, weight sharing, and pooling leading to:

- Modelling the spatial structure
- Translation invariance
- Local feature detection

Not covered

- Dilated, depthwise separable, 1D, 3D convolutions
- Training CNNs

Reading: Goodfellow et al. *Deep Learning* (ch 9)



https://forms.cloud.microsoft/e/WdeYNv4A5c