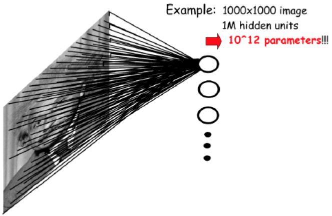
Neural Networks and Deep Learning: Convolution Properties

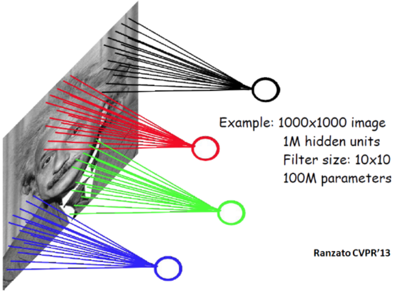
Nicolas Thome

Conservatoire National des Arts et Métiers (Cnam)
Département Informatique

<u>**Convolution Layers:**</u> **overcome important limitations of fully connected layers**

1. **Local connection, shared weights ⇒ drastic reduction in the number of parameters**
   a) Sparse connectivity: hidden unit only connected to a local patch

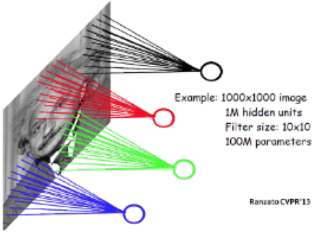


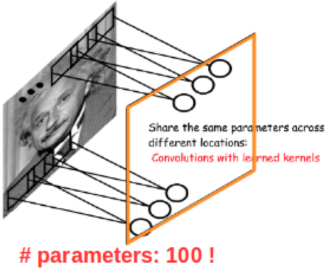


Convolution *vs* Fully Connected Layers

Convolution Layers: **overcome important limitations of fully connected layers**

1. **Local connection, shared weights $\Rightarrow$ drastic reduction in the number of parameters**
   b) Weight sharing: same feature detected across all image positions

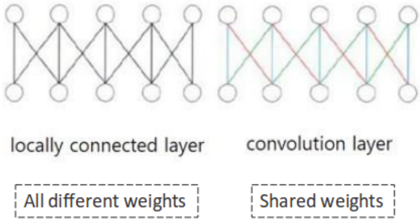




- **Convolution:** number of parameters independent of input image size
  ! $\neq$ fully connected layers

Translation-Invariant Feature Detection

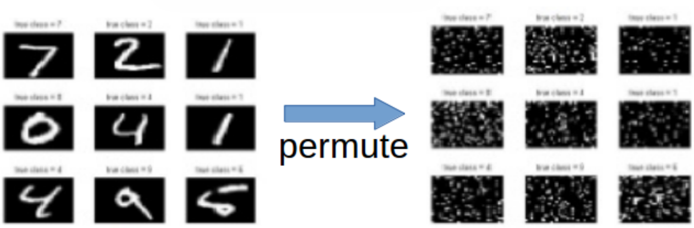
- Convolution, weight sharing: same feature detected across all image positions
- Very relevant *prior* for object classification / scene recognition
- Locally connected: useful in some specific contexts, *e.g.* face recognition ⇒ following!



Convolution *vs* Fully Connected Layers

Convolution Layers: overcome important limitations of fully connected layers

2. **Convolution: local spatial structure**
   - Analyses shape/appearance in a local neighborhood
   - Permutation to input images $\Rightarrow$ very different local info $\Rightarrow$ very different convolution maps $\Rightarrow$ Different classification performances



Convolution *vs* Fully Connected Layers

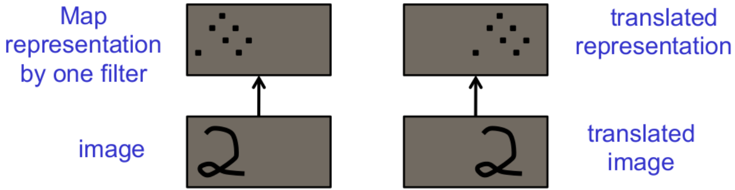
Convolution Layers: overcome important limitations of fully connected layers

3. **Convolution: equivariance property**
   - **Equivariance:** function $f(x)$ equivariant $g \Leftrightarrow f[g(x)] = g[f(x)]$
   - Convolution equivariant to translation:
     $T[x(t-\tau)] = x(t-\tau) \star h(t) = (x \star h)(t-\tau) = y(t-\tau)$

$$
\begin{tikzcd}
X \arrow[r, "g"] \arrow[d, "f"'] & X \arrow[d, "f"] \\
Y \arrow[r, "g"'] & Y
\end{tikzcd}
$$

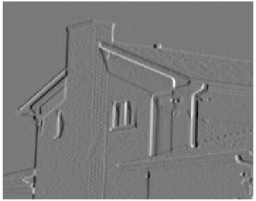
Convolution Layers: overcome important limitations of fully connected layers

3. **Convolution: translation equivariance**
    - Ensure that deformation, *i.e.* translation, encoded in maps
    - Local translation invariance: local pooling $\Rightarrow$ next !

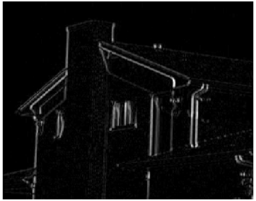


Convolution and Non-Linearity

source image I



$I \star M_x$



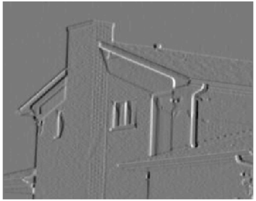
$|I \star M_x|$

$(I \star M_x)^2$

- Convolution, linear operation for each feature map
  - Ex: Gradient: $I_x = \frac{\partial I}{\partial x} \approx I \star M_x$, $M_x = \frac{1}{4} \cdot \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
- Followed by point-wise non-linearity
  $\sim$ fully connected networks
  - **Detector: large value $\Rightarrow$ presence of feature**
  - Ex: $\sigma(z) = z^2$, $\sigma(z) = |z|$
    $\Rightarrow$ activate for large > 0 & < 0 I_x values

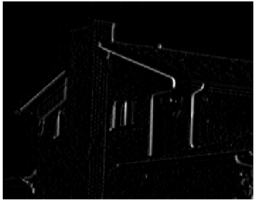
Convolution and Non-Linearity

source image I



$I \star M_x$

Sigmoid



ReLU

- Other non-linearities: only activate for $I_x > 0$
  - Sigmoid (with bias) $\sigma(z) = (1 + e^{-a(z-b)})^{-1}$, $a = 8 \cdot 10^{-}2$, $b = 50$
  - ReLU (see later) $\sigma(z) = max(0, z)$

Conclusion

- **Convolution: efficiency, locality, equivariance**
- **Non-linearity: feature detection**
- Limit number of parameters? Invariance?
  Pooling ⇒ following!