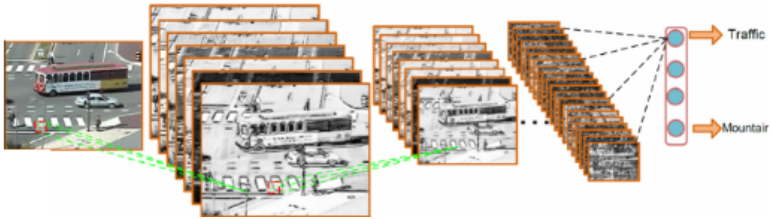
Neural Networks and Deep Learning: Representation Learning with ConvNets

Nicolas Thome

Conservatoire National des Arts et Métiers (Cnam)
Département Informatique

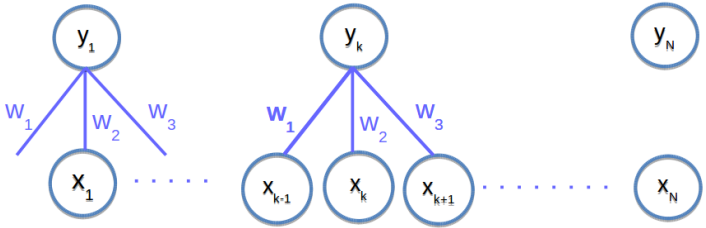
Deep ConvNets

- Deep models: hierarchy of sequential layers
- Layers: fully connected, $\overbrace{\text{convolution layer}}^{\text{convolution + non linearity}}$, pooling
- Supervised training for classification
  $\Rightarrow$ **Representation Learning**



Error Back-Propagation ConvNets

- **Convolution:** example for 1d scalar conv with mask $\mathbf{w} = [w_1 \; w_2 \; w_3]^T$



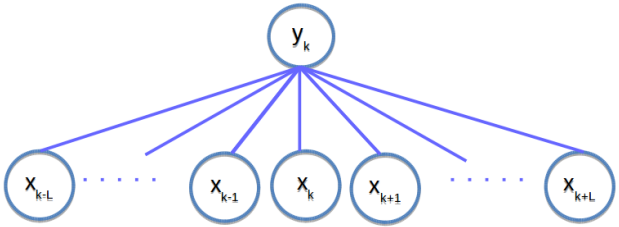
- **Shared weights: simple chain rule application**

$\Rightarrow$ Sum gradients for every region y_k!

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \sum_{k=1}^{N} \frac{\partial \mathcal{L}}{\partial y_k} \frac{\partial y_k}{\partial \mathbf{w}} \quad - \quad \frac{\partial y_k}{\partial \mathbf{w}} = [x_{k-1} \; x_k \; x_{k+1}]^T$$

Error Back-Propagation with ConvNets

- **Pooling:** example for pooling area of size $2L+1$:
- $y_k = f(x_{k-L}, ..., x_{k+L})$



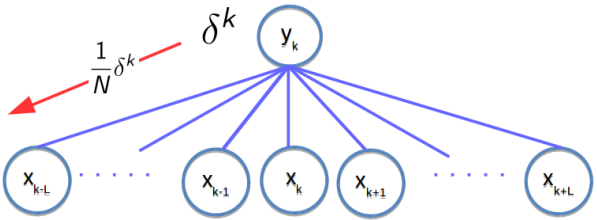
$$\frac{\partial \mathcal{L}}{\partial x_h} = \frac{\partial \mathcal{L}}{\partial y_k} \frac{\partial y_k}{\partial x_h} = \delta^k \frac{\partial y_k}{\partial x_h}$$

Error Back-Propagation for Average pooling

- Average pooling: $y_k = \frac{1}{N} \sum_{h=k-L}^{k+L} x_h$

$$\boxed{\frac{\partial \mathcal{L}}{\partial x_h} = \delta^k \frac{\partial y_k}{\partial x_h} = \frac{1}{N} \delta^k}$$

$\Rightarrow$ **Gradient propagated through each input node ($\frac{1}{N}$)**

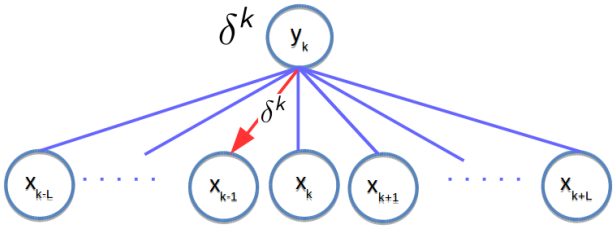


Error Back-Propagation for Max pooling

- Max pooling: $y_k = \max_{h'} x_h$

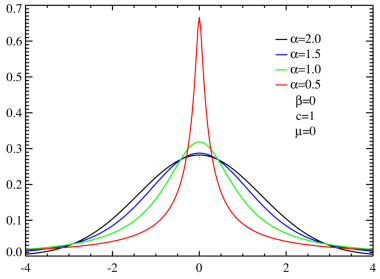
$$\frac{\partial \mathcal{L}}{\partial x_h} = \begin{cases} \delta^k \text{ if } x_h = \max_{h' \in \{k-L,\ldots,k+L\}} x_{h'} \\ 0 \text{ otherwise} \end{cases}$$

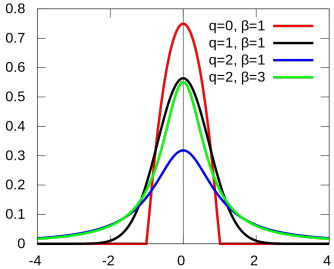
⇒ **Gradient propagated through arg max input node**



ConvNets & Prior Distribution

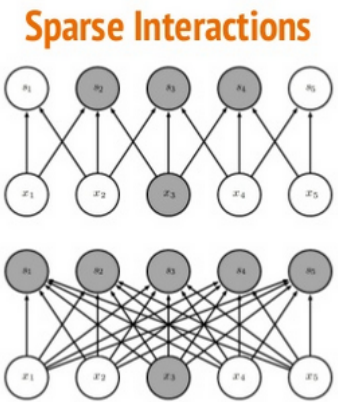
- **Prior:** imposing distribution on fully connected parameters
- Weak prior: high entropy (uncertainty), strong prior: low entropy
- **Infinitely strong prior:** zero probability on some parameters





ConvNet as Infinitely Strong Prior

- **ConvNet ~ Infinitely Strong Prior on Fully Connected net weights**
- Convolution: local interactions, shared weights ⇒ zero probability elsewhere

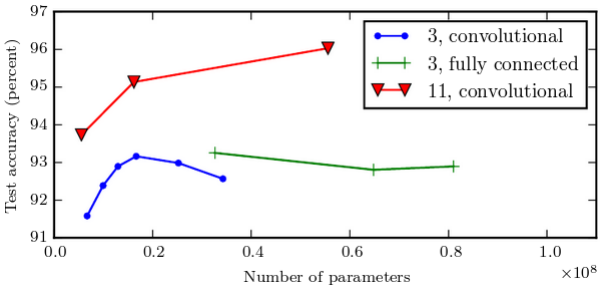


ConvNet for Learning Representation

- **ConvNet ~ Infinitely Strong Prior on Fully Connected net weights**
- **N.B.: weights adjusted for classification with back-prop**
- Convolution ⇒ support learning translation-equivariant features
- Pooling ⇒ support features invariant (stable) wrt local translations

ConvNet for Learning Representation

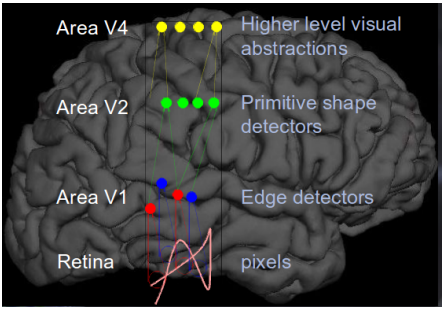
- ConvNet Infinitely Strong Prior: adapted to data with local interactions, *e.g.* image, speech, *etc*
- Very rich modeling capacities: local interactions ⇒ global with depth
- Significantly reduce # parameters ⇒ reducing over-fitting

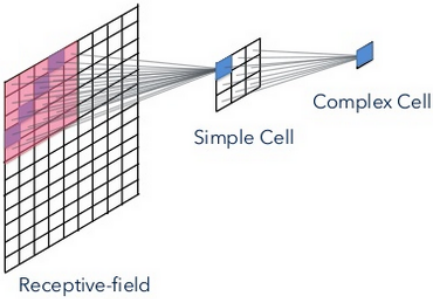


From [Goodfellow et al., 2016]

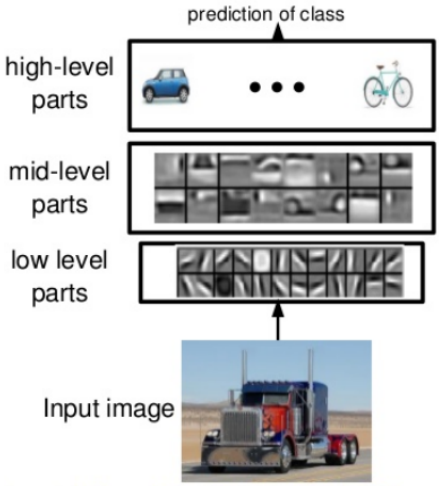
ConvNet for Learning Compositions

- **Conv/Pool hierarchies: feature composition**
  - Depth: gradual complexity, larger spatial extend
  - Intuitive processing for hierarchical information modeling
  - Biological foundations: simple cells, complex cells





ConvNet for Learning Compositions

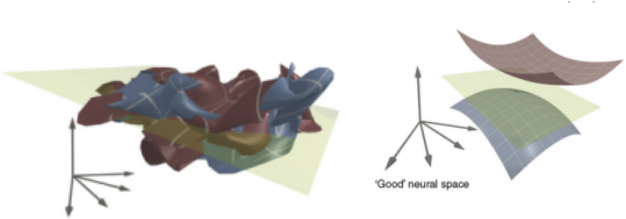


Credit: M.A Ranzato

- **Hierarchical Compositions**
  - Low-level: edges, color
  - Mid-level: corner, parts
  - Higher levels: objects, scene concepts
- **Distributed Representations: sharing**
  - Lower-level: shared by many classes
  - Higher-levels: more class specific

Representation Learning with Deep ConvNets: Conclusion

- Back-prop with ConvNets: **simple chain rule for Conv / Pool**
- Conv / Pool relevant *prior*: **learning good features for low-level signals**
- Conv / Pool hierarchies: **learning composition of parts**
- **Deep Learning & Manifold untangling?**
  ⇒ following!



References I


Goodfellow, I., Bengio, Y., and Courville, A. (2016).
Deep Learning.
MIT Press.
http://www.deeplearningbook.org.