Backpropagation learning for convolutional networks

Sebastian Seung

Generalize layered structure to directed acyclic graph

Forward pass

$$
\begin{aligned}
u_i^\alpha &= \sum_{\beta j} W_{ij}^{\alpha\beta} x_j^\beta + b_i^\alpha \\
x_i^\alpha &= f(u_i^\alpha)
\end{aligned}
$$

Convolutional constraint

$$W_{ij}^{\alpha\beta} = w_{i-j}^{\alpha\beta} \qquad\qquad b_i^{\alpha} = b^{\alpha}$$

$$\begin{aligned}
\mathbf{u}^{\alpha} &= \sum_{\beta} \mathbf{w}^{\alpha\beta} * \mathbf{x}^{\beta} + b^{\alpha}\mathbf{1} \\
\mathbf{x}^{\alpha} &= \mathbf{f}(\mathbf{u}^{\alpha})
\end{aligned}$$

Backward pass

$$\hat{x}_j^\beta = \sum_{i\alpha} \hat{u}_i^\alpha W_{ij}^{\alpha\beta}$$

$$\hat{u}_j^\beta = f'(u_j^\beta)\hat{x}_j^\beta$$

Backward pass

- Use the spatial inversion of the filter
  - i.e., flip the filter about each axis
- Convolution is multiplication by a matrix of the form $W_{ij} = w_{i-j}$
- The matrix transpose is

$$W_{ji} = w_{j-i}$$

Gradient “sharing”

$$x(t) = t \qquad y(t) = t$$

$$\begin{aligned} \frac{d}{dt} f(x(t), y(t)) &= \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} \\ &= \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \end{aligned}$$

Gradient learning with parameter sharing

- Find the derivatives with respect to each of the individual parameters, treating them as independent. Then sum all of these to find the derivative with respect to the shared parameter.

For Whoever Shares, to Him More Gradient Will Be Given

corruption of Mark 4:25

Weight update from cross-correlation

$$\Delta W_{ij}^{\alpha\beta} \propto \hat{u}_i^\alpha x_j^\beta$$

$$\begin{aligned}
\Delta w_k^{\alpha\beta} \quad &\propto \quad \sum_{\substack{i,j \\ i-j=k}} \hat{u}_i^\alpha x_j^\beta \\
&= \quad \sum_j \hat{u}_{j+k}^\alpha x_j^\beta
\end{aligned}$$