ECE544NA: Convolution Neural Network Tutorial

Raymond Yeh

University of Illinois at Urbana Champaign

yeh17@illinois.edu

September 26, 2016

Neural Networks in a slide!

Given a set $\{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ of training examples.
We define the loss of a model as

$$\mathcal{L}(W) = \sum_{i=1}^{N} l(\Phi_W(x_i), y_i) \tag{1}$$

Training a neural network

$$W = \arg\min_W \mathcal{L}(W) \tag{2}$$

Making a prediction for a neural network

$$\hat{y}_i = \Phi_W(x_i) \tag{3}$$

Fully Connected Layer

$z_j^{(l-1)}$ = the j^{th} hidden node's activation at $(l-1)^{th}$ layer. $z_j^{(l-1)} \in \mathbb{R}$.

$W_{ij}^{(l-l)}$ = the weight in the $(l-1)^{th}$ layer going from input i to hidden unit j. $W_{ij}^{(l-l)} \in \mathbb{R}$

$b_j^{(l-1)}$ = the bias for j^{th} node at $(l-1)^{th}$ layer. $b_j^{(l-1)} \in \mathbb{R}$

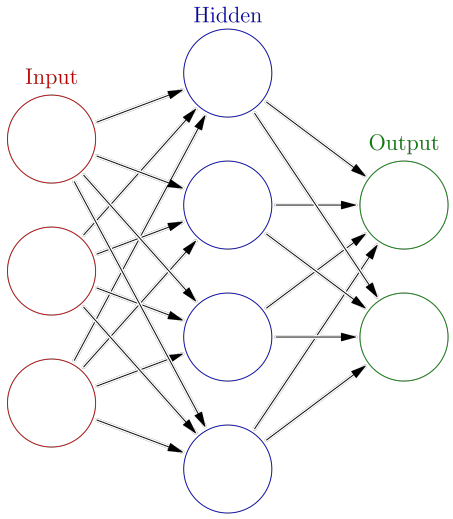
$f(\cdot)$ = a element-wise non-linear function.

The forward Convolution Operation

$$a_j^{(l)} = (\sum_i z_i^{(l-1)} \cdot W_{ij}^{(l-1)}) + b_j^{(l-1)}$$

$$z_j^{(l)} = f(a_j^{(l)})$$

Fully Connected Layer for Image?

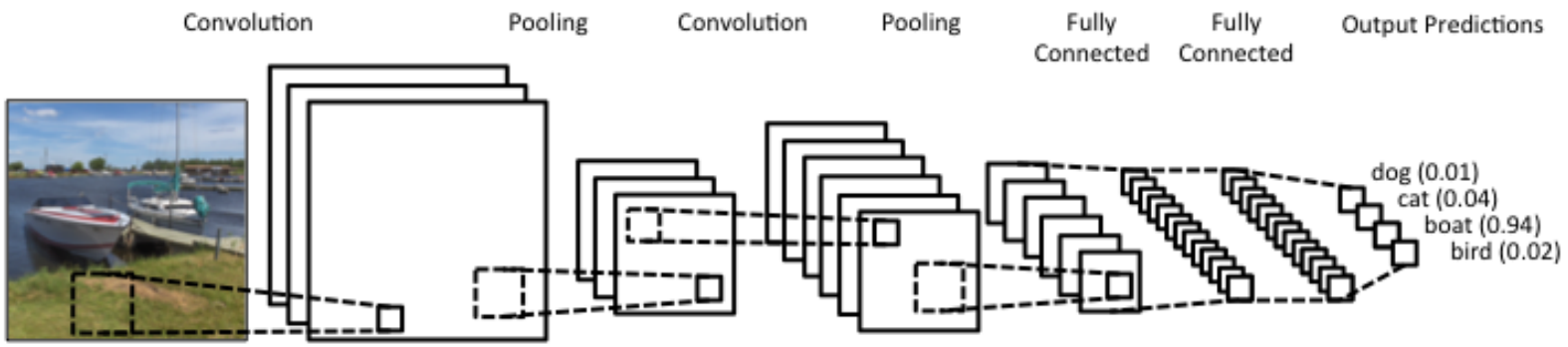


Neural network illustration

Consider an image of size 250×250.
For each hidden node, it requires $250 \times 250 = 62,500$ weights.
If we choose the same number of hidden node as the input, then we need $62,500 \times 62,500 = 3,906,250,000$ weights per layer.
Not the best idea! (1) Expensive to compute, (2) Very complicated model.

Convolutional Neural Network Motivation

Assume stationary image statistics, this motivates the weight sharing between the inputs, and results in the convolution layer.



CNN illustration from clarifai.com.

Advantage: (1) Reduces the number of weights, (2) Generalizes better in practice, (3) Respects the image structure, and enable operators that are easier to interpret.

Convolution Layer

$z_j^{(l)}$ = the j^{th} channel of the activation map at l^{th} layer. $z_j^{(l)}$ is a matrix.
$W_{ij}^{(l)}$ = the i^{th} channel of the j^{th} filter at l^{th} layer. $W_{ij}^{(l)}$ is a matrix.
$b_j^{(l-1)}$ = the bias for j^{th} filter at l^{th} layer. $b_j^{(l-1)}$ is a scalar.
$f(\cdot)$ = a element-wise non-linear function.
$\star$ = convolution

The forward Convolution Operation

$$a_j^{(l)} = (\sum_i z_i^{(l-1)} \star W_{ij}^{(l-1)}) + b_j^{(l-1)} \cdot \mathbf{1}$$

$$z_j^{(l)} = f(a_j^{(l)})$$

Note: $\mathbf{1}$ denotes matrix of all ones.

Convolution Layer Gradient

$\mathcal{L}$ = the cost function
$\delta_j^{(l)}(u,v) = \frac{\partial \mathcal{L}}{\partial a_j^{(l)}(u,v)}$ = the backprop error.

Gradient for bias term

$$\frac{\partial \mathcal{L}}{\partial b_j^{(l-1)}} = \sum_{j'} \sum_{u'} \sum_{v'} \frac{\partial \mathcal{L}}{\partial a_{j'}^{(l)}(u',v')} \cdot \frac{\partial a_{j'}^{(l)}(u',v')}{\partial b_j^{(l-1)}} = \sum_{u'} \sum_{v'} \frac{\partial \mathcal{L}}{\partial a_j^{(l)}(u',v')} \cdot \frac{\partial a_j^{(l)}(u',v')}{\partial b_j^{(l-1)}} =$$

$$\sum_{u'} \sum_{v'} \frac{\partial \mathcal{L}}{\partial a_j^{(l)}(u',v')} = \sum_{u'} \sum_{v'} \delta_j^{(l)}(u',v')$$

Recall $a_j^{(l)}(u,v) = (\sum_i z_i^{(l-1)} \star W_{ij}^{(l-1)})(u,v) + b_j^{(l-1)}$

Thus, $\frac{\partial a_j^{(l)}(u',v')}{\partial b_j^{(l-1)}} = 1$

Convolution Layer Gradient

Gradient for Weight term

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(l-1)}(u,v)} = \sum_{j'} \sum_{u'} \sum_{v'} \frac{\partial \mathcal{L}}{\partial a_{j'}^{(l)}(u',v')} \cdot \frac{\partial a_{j'}^{(l)}(u',v')}{\partial W_{ij}^{(l-1)}(u,v)} =$$

$$\sum_{u'} \sum_{v'} \frac{\partial \mathcal{L}}{\partial a_{j}^{(l)}(u',v')} \cdot \frac{\partial a_{j}^{(l)}(u',v')}{\partial W_{ij}^{(l-1)}(u,v)}$$

Recall, $a_j^{(l)}(u', v') = (\sum_{\hat{i}} \sum_{\hat{u}} \sum_{\hat{v}} z_{\hat{i}}^{(l-1)}(u' - \hat{u}, v' - \hat{v}) W_{\hat{i}j}^{(l-1)}(\hat{u}, \hat{v})) + b_j^{(l-1)}$.

(Expanded using def. of Conv).

Then, $\frac{\partial a_j^{(l)}(u',v')}{\partial W_{ij}^{(l-1)}(u,v)} = z_i^{(l-1)}(u' - u, v' - v)$ as the derivative is non-zero when $\hat{u} = u$, and $\hat{v} = v$.

Convolution Layer Gradient

Putting everything together.

Gradient for Weight term

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(l-1)}(u,v)} = \sum_{u'}\sum_{v'} \frac{\partial \mathcal{L}}{\partial a_j^{(l)}(u',v')} \cdot \frac{\partial a_j^{(l)}(u',v')}{\partial W_{ij}^{(l-1)}(u,v)} = \sum_{u'}\sum_{v'} \delta_j^{(l)}(u',v') z_i^{(l-1)}(u' - u, v' - v) = \sum_{u'}\sum_{v'} \delta_j^{(l)}(u',v') \widetilde{z}_i^{(l-1)}(u - u', v - v') = \delta_j^{(l)} \star \widetilde{z}_i^{(l-1)}(u,v)$$

Note: $\widetilde{z}_i^{(l-1)}(u,v) = z_i^{(l-1)}(-u,-v)$

Backprop Error

Backprop Error $\delta_j^{(l-1)}$

$$\delta_j^{(l-1)}(u,v) = \frac{\partial \mathcal{L}}{\partial a_j^{(l-1)}(u,v)} = \sum_{j'} \sum_{u'} \sum_{v'} \delta_{j'}^{(l)}(u',v') \cdot \frac{\partial a_{j'}^{(l)}(u',v')}{\partial a_j^{(l-1)}(u,v)}$$

Recall, $a_{j'}^{(l)}(u',v') = (\sum_{\hat{j}} \sum_{\hat{u}} \sum_{\hat{v}} z_{\hat{j}}^{(l-1)}(u'-\hat{u}, v'-\hat{v}) W_{\hat{j}j'}^{(l-1)}(\hat{u},\hat{v})) + b_{\hat{j}}^{(l-1)}$

Then, $\frac{\partial a_{j'}^{(l)}(u',v')}{\partial a_j^{(l-1)}(u,v)} = W_{jj'}^{(l-1)}(u'-u, v'-v) f'(a_j^{(l-1)}(u,v))$ as the derivative is non-zero only when $u'-\hat{u} = u$ and $v'-\hat{v} = v$.

Backprop Error

Putting everything together

Backprop Error $\delta_j^{(l-1)}$

$$\delta_j^{(l-1)}(u,v) = \frac{\partial \mathcal{L}}{\partial a_j^{(l-1)}(u,v)} =$$

$$\sum_{j'}\sum_{u'}\sum_{v'} \delta_{j'}^{(l)}(u',v') \cdot W_{jj'}^{(l-1)}(u'-u, v'-v) \cdot f'(a_j^{(l-1)}(u,v)) =$$

$$\sum_{j'}\sum_{u'}\sum_{v'} \delta_{j'}^{(l)}(u',v') \cdot \widetilde{W}_{jj'}^{(l-1)}(u-u', v-v') \cdot f'(a_j^{(l-1)}(u,v)) =$$

$$((\sum_{j'} \delta_{j'}^{(l)} \star \widetilde{W}_{jj'}^{(l-1)}) \cdot f'(a_j^{(l-1)}))(u,v)$$

Note: $\widetilde{W}_{jj'}^{(l-1)}(u,v) = W_{jj'}^{(l-1)}(-u,-v)$