Back-Propagation over CNN

Xenuts

October 5, 2014

Contents

1 Notations

- K: Number of classes. $k = 1 \to K$
- N: Number of training examples. $n = 1 \to N$
- L: Layer number. $l = 1 \to L$
- $\mathbf{z}^l$: The activation of layer l. $\mathbf{z}^1$ is exactly the input, and $\mathbf{z}^L$ is exactly the output $\mathbf{y}$.

- $\mathbf{y}$: Output. Shape $= K \times N$
- $\mathbf{t}$: Target (one-hot scheme). Shape $= K \times N$
- $f(\cdot)$: Activation function / non-linearity function
- W^l: Weights for fully-connected and output layers and filters for conv layers.
- $\mathbf{b}^l$: Bias.
- S_l: The size of fully-connected layers l or the number of feature maps of conv and pooling layers. For output layer, $K = S_L$
- $[r_l, c_l]$: The shape of feature maps of conv and pooling layers.

2 Shapes

2.1 Conv and Pooling layers

- $\mathbf{z}^l$: $[r_l \times c_l \times S_l \times N]$
- W^l: $[r_f \times c_f \times S_{l-1} \times S_l]$
- $\mathbf{b}^l$: $S_l \times 1$

2.2 Fully-connected and Output layers

- $\mathbf{z}^l$: $[S_l \times N]$
- W^l: $[S_{l-1} \times S_l]$
- $\mathbf{b}^l$: $S_l \times 1$

3 Output Layer L

- Input: $\mathbf{z}^{L-1}$
- Output: $\mathbf{y} = \mathbf{z}^L = f(\mathbf{u}^L), \mathbf{u}^L = W^L \mathbf{z}^{L-1} + \mathbf{b}^L$
- Target: $\mathbf{t}$ is one-hot fashion
- Gradients using delta rule:

– Weights:

$$\nabla W^L = \frac{1}{N} \cdot \mathbf{z}^{L-1}(\delta^L)^\top + \lambda W^L$$

– Bias:

$$\nabla \mathbf{b}^L = \frac{1}{N} \cdot \sum_{n=1}^{N} \delta^L_{\cdot,n}$$

3.1 Squared-Error Loss (i.e. sigmoid)

- Method: $f(x) = \text{sigmoid}(x)$
- Squared-Error Loss:

$$J = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} (\mathbf{t}_{k,n} - \mathbf{y}_{k,n})^2$$

- Error Sensitivity (see details):

$$\delta^L = f'(\mathbf{u}^L) \odot (\mathbf{y} - \mathbf{t})$$

3.2 Cross-Entropy Loss (Softmax)

- Method: $f(x) = \text{softmax}(x)$
- Cross-Entropy Loss:

$$J = -\sum_{n=1}^{N} \sum_{k=1}^{K} \mathbf{t} \odot \log \mathbf{y} + \frac{\lambda}{2} \sum_{l=1}^{L} ||W^l||_2^2$$

- Error Sensitivity (see details):

$$\delta^L = \mathbf{y} - \mathbf{t}$$

4 Fully-Connected Layer l

A fully-connected layer can only be followed by an output layer or another fully-connected layer.

- Input: $\mathbf{z}^{l-1}$
- Output: $\mathbf{z}^l = f(\mathbf{u}^l), \mathbf{u}^l = W^l \mathbf{z}^{l-1} + \mathbf{b}^l$

- Gradients using delta rule:
  - Weights:
$$\nabla W^l = \frac{1}{N} \cdot \mathbf{z}^{l-1}(\delta^l)^\top + \lambda W^l$$
  - Bias:
$$\nabla \mathbf{b}^l = \frac{1}{N} \cdot \sum_{n=1}^{N} \delta^l_{\cdot,n}$$
- Error Sensitivity (see details): shape is $S_l \times N$
$$\delta^l = W^{l+1}\delta^{l+1} \odot f'(\mathbf{u}^l)$$
- Derivative of Common Non-Linearity Function
  - Sigmoid:
$$f(x) = \frac{1}{1+\exp(-x)} \Rightarrow f'(x) = f(x)(1-f(x))$$
  - tanh:
$$f(x) = \tanh(x) \Rightarrow f'(x) = 1-(f(x))^2$$
  - ReLU:
$$f(x) = \max(x, 0) \Rightarrow f'(x) = (f(x) > 0)$$

5 Convolution Layer l

A convolution layer can be followed by layer 'p', 'c', 'f', 'o'.

- Gradients ($1 \leq i \leq S_{l-1}, 1 \leq j \leq S_l$):
  - Weights:
$$\nabla W^l_{i,j} = \frac{1}{N} \cdot (\mathbf{z}^{l-1}_{\cdot,\cdot,i,\cdot} \circledast_{valid} \mathrm{rot180}(\delta^l_{\cdot,\cdot,j,\cdot})) + \lambda W^l_{i,j}$$
  - Bias:
$$\nabla \mathbf{b}^l_j = \frac{1}{N} \cdot \sum_{n=1}^{N} \sum_{u,v} \delta^l_{u,v,j,n}$$

5.1 Followed by a Pooling Layer

- Error Sensitivity:

$$\delta^l = f'(\mathbf{u}^l) \odot \text{unpool}(\delta^{l+1})$$

5.2 Followed by a Convolution Layer

- Error Sensitivity:

$$\delta^l = f'(\mathbf{u}^l) \odot (\delta^{l+1} \circledast_{full} W^{l+1})$$

6 Pooling Layer l

A pooling layer can be followed by layer 'c', 'f', 'o'. The error sensitivity δ's computation is the same as above but here $f'(\mathbf{u}^l) = 1$.