Fully Connect in Neural Network

Zhiliang Tian

2016.05.29

1 Introduction

Fully Connect is a layer in neural network. This layer has own parameter, which connect each input nodes and each output nodes, so it's called fully connect layer(FC layer). It is a Linear transformation from Input nodes to output nodes.

2 Feed Forward

2.1 Feed Forward

$$\mathbf{y} = W^T * \mathbf{x} + \mathbf{b} \tag{1}$$

$$y_k = \sum_i^N W_{ki} * x_i + b_k \tag{2}$$

$\mathbf{x}$ is a column vector with N dimension, x_i is the i-th dimension of input nodes $\mathbf{x}$.

$\mathbf{y}$ is a column vector with K dimension, y_k is the k-th dimension of output nodes $\mathbf{y}$.

W is a weight matrix with $N * K$ (N rows and K columns). W connect each nodes in $\mathbf{x}$ with each nodes in $\mathbf{y}$ fully, W_{ik} is the connection between x_i and y_k. Matrix W is a part of parameter in FC layer.

$\mathbf{b}$ is the bias in fully connect layer, a column vector with K dimension. b_k is the i-th dimension of $\mathbf{b}$, b_k will be add on y_k. Vector $\mathbf{b}$ is also a part of parameter in FC layer.

2.2 some discuss

- A linear transformation from Input nodes to output nodes.

- A feature select procedure. Just as linear regression, logistic regression, $\mathbf{x}$ is input feature, W is feature weight, $\mathbf{y}$ is output. FC layer will find a suitable feature weight.
- Considering FC is only a linear transformation, we usually add non-linear transformation on output $\mathbf{y}$ when applying FC in deep neural network. (sigmoid, tanh, ...)

3 Back propagation

3.1 Back propagation

$\mathbf{b}$'s gradient:

$$\frac{\alpha(\mathbf{y})}{\alpha(b_k)} = \frac{\alpha(y_k)}{\alpha(b_k)} = 1 \tag{3}$$

$$\frac{\alpha(\mathbf{y})}{\alpha(\mathbf{b})} = \mathbf{1} \tag{4}$$

W 's gradient:

$$\frac{\alpha(\mathbf{y})}{\alpha(W_{ik})} = \frac{\alpha(y_k)}{\alpha(W_{ik})} = x_i \tag{5}$$

$$\frac{\alpha(\mathbf{y})}{\alpha(W_k)} = \frac{\alpha(y_k)}{\alpha(W_k)} = \mathbf{x} \tag{6}$$

$$\frac{\alpha(\mathbf{y})}{\alpha(W)} = \{\frac{\alpha(\mathbf{y})}{\alpha(W_0)}, \frac{\alpha(\mathbf{y})}{\alpha(W_1)}, ..., \frac{\alpha(\mathbf{y})}{\alpha(W_K)}\} = \{\mathbf{x}, \mathbf{x}, ..., \mathbf{x}\} \tag{7}$$

In equation 6 and 7, W_i is the t-th column vector of W

$\mathbf{x}$'s gradient:

$$\frac{\alpha(\mathbf{y}_k)}{\alpha(\mathbf{x}_i)} = W_{ki} \tag{8}$$

$$\frac{\alpha(\mathbf{y})}{\alpha(\mathbf{x})} = W^T \tag{9}$$

$$\Delta(\mathbf{y}) = \frac{\alpha(\mathbf{y})}{\alpha(\mathbf{x})}\Delta(\mathbf{x}) = W^T\Delta(\mathbf{x}) \tag{10}$$

But $\frac{\alpha(\mathbf{y})}{\alpha(\mathbf{x})}$ is a matrix, how can we use it. The matrix will multiply with y's gradient. For example, in a model, J is loss from model, the gradient of J respect to $\mathbf{x}$ can be computed.

Using equation 10

$$W * \Delta(\mathbf{y}) = \Delta(\mathbf{x}) \tag{11}$$

$$\frac{\alpha(J)}{\alpha(\mathbf{x})} = \Delta(\mathbf{x}) = W * \Delta(\mathbf{y}) = W\frac{\alpha(J)}{\alpha(\mathbf{y})} \tag{12}$$

$$\frac{\alpha(J)}{\alpha(\mathbf{x})} = \frac{\alpha(J)}{\alpha(\mathbf{y})} * \frac{\alpha(\mathbf{y})}{\alpha(\mathbf{x})} = W\frac{\alpha(J)}{\alpha(\mathbf{x})} \tag{13}$$

3.2 some discuss

- Considering the initialization parameter usually small (such as (-0.1, 0.1)), bias' gradient is usually bigger than weight's and input's. Bias is easy to fit.
- Considering the initialization parameter usually small, when applying FC in DNN, bottom layer's gradient is usually smaller than top layer's.