Figure convolutional neural networks

The total error(will be refer to later on as E) is the sum of errors for each output neuron and can be calculated using log-loss (sometimes called cross entropy) function:

where N is the number of output neurons, is the expected output of neuron k and is the real output of neuron k.

We will look at how the networks adjust its weight using by looking at effect on each layer.

**Fully connected layer (Neural networks)**

Figure fully connected layer

The real output at fully connected layer can be expressed in terms of net output by function () in section

Therefore, we can calculate

Each neuron at the output layer has a net value defined by the following function

where is the net output of neuron k at layer l, is the real output of neuron j at layer (l-1) where j = {1,2,3,…,m}, is the weight of the connection between neuron j and k, is the bias value of neuron k.

The amount of error that we want to adjust at each neuron is equal to:

(as a result of chain rule)

where N is the number of output neurons, o stands for output layer, is the error rate at neuron N in the output layer, is the net output of neuron N in the input layer, , is the real output of neuron N in the input layer.

We will propagate this error forward. Accordingly, this rate will affect the weight at neuron N by an amount of

(as a result of chain rule)

where is the amount of weight to adjust between neuron j and neuron , is the real output of neuron at output layer, is the net output of neuron at output layer, is the current weight between neuron j and neuron .

From function () and (), function () can be rewritten as:

where is the amount of weight to adjust between neuron j and neuron , is the current weight between neuron j and neuron is is the error rate at neuron N in the output layer, is the real output of neuron at hidden layer h1 where h1 precedes o.

To decrease the error, we then subtract this value from the current weight

where denotes the learning rate, which is normally set to 0.5

For every neuron at each hidden and input layer of the neural networks, we can calculate and by applying the same technique. Thus our calculation can be generalized as follows:

where is the error rate at neuron j in layer l, is the net output of neuron j in layer l, is the real output of neuron j in layer l , represents the total net output of all neurons in layer (l+1) that are pairwise connected with neuron j, m is the total number of neurons in layer (l+1), is the error rate at neuron i’ in layer (l+1) and is the weight between neuron j and neuron i’ where i’ = {0,1,…,m}.

We will propagate this error forward

where is the amount of weight to adjust between neuron i and neuron j, is the error rate at neuron j in layer l, is the real output of neuron i in layer (l-1) .

**ReLU layer**

The output for each neuron can be written as

where is the real output value of neuron i of ReLU layer (l-1). Thus for each output at layer ReLU layer, we want to adjust an error amount equivalent to

We need not to adjust weight at this layer because it has no weight.

**Max pooling layer**

The output for each neuron can be represented by

where is the the real output value of neuron at position (x,y) of max pooling layer l, is the real output value of neuron at position (x+p,y+q) of convolutional layer (l-1) where (p,q) is the position in a kernel of size kxk.

For each output pixel at max pooling layer, we want to adjust an amount of

We need not to adjust weight at this layer because it also has no weight.

**Convolutional layer**

Figure convolutional layer fatten into neural networks

The net output of each neuron at convolutional layer can be represented by

as discussed in section

The amount of error that we want to adjust at each neuron is equal to:

(as discussed in fully connected layer)

function () is thus equivalent to

where is net output of pixel at position (x,y) at layer l, is real output of pixel at position (x,y) at layer l, represents the total net output of all pixels at layer (l+1) that are pairwise connected with pixel at position (x,y), is the error rate of pixel at position (x,y) at layer l, is the error rate of pixel at position (x,y) at layer (l+1), is the weight between (i’,j’) and (x,y), is the net output of pixel (i’,j’), is the real output of pixel (i’,j’).

This error is contributed into the current weight as follows:

where is the weight between pixel (a,b) and pixel (x,y), is net output of pixel (x,y) at layer l, is real output of pixel (x,y), is the error rate of pixel (x,y) at layer l.