TU München
Fakultät für Informatik
PD Dr. Rudolph Triebel
J. Chiotellis, P. Häusser

Machine Learning for Computer Vision
Winter term 2016

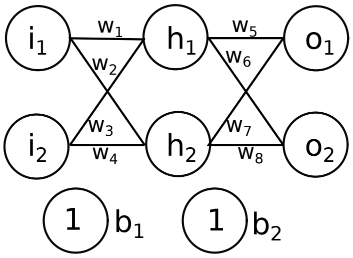
December 9, 2016
Deep Learning, Boosting

Exercise 1: Back Propagation

- Suppose we modify a single neuron in a feedforward network so that the output from the neuron is given by $f(\sum_j w_j x_j + b)$, where f is some function other than the sigmoid.
  How should we modify the backpropagation algorithm (from chapter 2 of the above reading) in this case?

  Derivative of the activation function $f(\cdot)$ at the corresponding neuron should be derived. During back propagation, the error should be back-propagated with f' for the corresponding neuron.

- Compute the gradient of the cost function C respect to w_5 $\left(\frac{\partial C}{\partial w_5}\right)$ given the following network:



$C = \sum_{i=1}^{2}(t_i - a_i)^2$ where t_i is the target value for the respective output neuron o_i and a_i is the output of the neuron o_i. Input/output of h_1, h_2, o_1, o_2 is computed as $z_i^l = \sum_j w_j \cdot a_j + b_l$, $a_i^l = \sigma(z_i^l)$ where $\sigma(\cdot)$ is any activation function.

$$\begin{aligned}
\frac{\partial C}{\partial w_5} &= \frac{\partial C}{\partial a_1^2} \cdot \frac{\partial a_1^2}{\partial z_1^2} \cdot \frac{\partial z_1^2}{\partial w_5} \\
\frac{\partial C}{\partial w_5} &= -2(t_1 - a_1^2) \cdot \sigma'(z_1^2) \cdot a_1^1 \\
\frac{\partial C}{\partial w_5} &= -2(t_1 - a_1^2) \cdot \sigma'\Bigg(w_5 a_1^1 + w_7 a_2^1 + b_2\Bigg) \cdot a_1^1
\end{aligned} \tag{1}$$

Exercise 2: Convolutional Layer Arithmetic

Consider a very simple convolutional neural network that just consists of one convolutional layer. It has the following parameters:

- number of kernels: $num = 64$
- size of kernels: $k = 3 \times 5$
- stride: $s = 2$
- padding: $p = 1$

Assume, the input to this layer is an a batch of RGB images. There are 10 images in one batch and the images have a dimension of 123×81.

In general, blobs have the canonical shape $N \times C \times H \times W$.

a) What is the shape of the input blob to the convolutional layer? Hint: it's a tensor with four axes.

The input blob will have the shape $10 \times 3 \times 81 \times 123$.

b) What is the shape of the output blob of the convolutional layer?

In order to find the output blob shape, one has to consider what happens during a convolution. If there is a non-zero padding, zeros will get virtually added at the boundary of the input. So the input size increases by $2 \times p$ where p is the padding size. Every time a kernel gets multiplied with pixels, one number for the activation map is computed. Thus, the number of times that you can "apply" a kernel along the height/width of the input will give you the dimensions of the output. Putting all this together will give you the following formula:

$$\tilde{x} = \frac{x + 2 \times p - k}{s} + 1$$

where $\tilde{x}$ is the output height or width, respectively. k is the kernel height or width and s is the stride. That means that the output blob will have the shape
$10 \times 64 \times 40 \times 62$

Exercise 3: Adaboost

See code.