Imperial College London

60006 Computer Vision (Term 2)

Tutorial 4: Image Classification

1. In neural networks, the activation function defines the output of a neuron given an input.

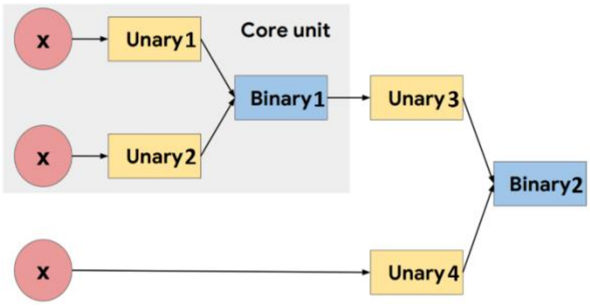
   1.1 The sigmoid function is a type of activation function. It is defined as,

   $$f(x) = \frac{1}{1+e^{-x}}$$

   Calculate the derivative of $f(x)$ with respect to x.

   1.2 Describe a problem with the sigmoid function when we train a neural network using the gradient descent algorithm.

2. A method was proposed to automatically search and discover new activation functions (P. Ramachandran et al, ICLR 2018 workshop). It assumes that the activation function has a structure shown in this figure,



   which consists of input x, unary functions and binary functions. The unary function takes a single scalar input and returns a single scalar output, such as $u(x) = x$. The binary function takes two scalar inputs and returns a single scalar output, such as $b(x_1, x_2) = x_1 \cdot x_2$.

   2.1 Suppose in the figure, the unary functions are respectively $u_1(x) = x$, $u_2(x) = x$, $u_3(x) = \sigma(x)$, $u_4(x) = \sigma(x)$, where $\sigma(x)$ denotes the sigmoid function and u_i denotes “Unary i” in the figure. The binary functions are $b_1(x_1, x_2) = \max(x_1, x_2)$ and $b_2(x_1, x_2) = \max(x_1, x_2)$, where b_i denotes “Binary i” in the figure.

   Write down the activation function.

   2.2 In the search space of activation functions, there are in total M possible unary functions (e.g. x, $-x$, $|x|$, x^2, x^3, $\sqrt{x}$, βx, …) and N possible binary functions (e.g. x_1+x_2, $x_1 \cdot x_2$, x_1-x_2, $\frac{x_1}{x_2+\epsilon}$, $\max(x_1, x_2)$, $\min(x_1, x_2)$, …). Calculate the number of possible combinations using the big O notation.

   2.3 The method utilises reinforcement learning to search for activation functions. Finally, it finds a novel activation function, $f(x) = x \cdot \sigma(\beta x)$, which performs well. What does this new activation function look like if $\beta = 0$ and if $\beta \to \infty$?

3. A data analyst has developed a neural network model to predict the chances of the three scenarios (win, draw, lose). However, the current model outputs a vector of three integers, $c = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$, instead of a probability vector. The three integers are not necessarily positive. But the larger the integer, the higher the chance is for that scenario.

3.1 He/she decides to apply the softmax function to convert the vector $c = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$ into a probability vector $p = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix}$. The softmax function is defined as,

$$p_i = \frac{e^{c_i}}{\sum_k e^{c_k}} \text{ for } i = 1,2,3.$$

Check whether p fulfils the properties of a probability vector, i.e. it is non-negative and its elements sum to 1.

3.2 Gradient descent is used to train this model. To calculate the gradient of the loss function, one step is to work out the derivative $\frac{\partial p_i}{\partial c_j}$. Please help him/her derive this. (Hint: consider two scenarios, $i = j$ and $i \neq j$.)

4. A convolutional neural network (CNN) takes a 28x28 image as input and produces an output of 10-dimensional probability vector and cross-entropy loss. It mainly consists of convolutional layers, max pooling layers and a loss layer. The network architecture is specified in the following table.

layer	0	1	2	3	4	5	6	7
type	input	conv	pool	conv	pool	conv	conv	loss
filter shape	-	5x5x1	2x2	5x5x20	2x2	4x4x50	1x1x500	-
#filters	-	20	-	50	-	500	10	-
stride	-	1	2	1	2	1	1	-
pad	-	0	0	0	0	0	0	-
data shape	1x28x28x1							
data size	3.06KB							
receptive field	1x1	5x5						

4.1 The input data x_0 is of shape 1x28x28x1, which represents BWHC (B: batch size; W: width; H: height; C: channel). If we use single precision floating-point data (4 bytes), the data size is 1x28x28x1x4 $\approx$ 3KB. Calculate the data shape and size for each following layer in the table. Data means input image at Layer 0 and feature map at subsequent layers.

4.2 The receptive field of a neuron represents the size of the region in input image that can affect this neuron. For example, Layer 1 uses a 5x5 convolution filter. Therefore, a neuron at Layer 1 has a receptive field of 5x5, since each neuron is affected by a 5x5 region in the input image. Calculate the receptive fields for neurons in the following layers and fill in the table.