|  |  |
| --- | --- |
| 2021/03/01 | Artificial Neural Network (ANN) |
| **Perceptron** | The perceptron consists a single neuron with adjustable synaptic weights and a hard limiter.  w1  xd  x1  activation  wi  xi  z  wd  y  Notation:     Some activation function Step function:    Sigmod function   Perceptron learning rule Notation:  : learning rate  : Threshold/Bias  Xn x d: train data set with n row, each row is vector d- dimensional  Each row of dataset we will do 3 steps as below:  Step 1: initialization  Initial weights  Initial bias  Initial  Step 2: Activation    Step 3: Weight update    Go to next row Example: Traing AND gate   Bias = 0.2  Learning rate = 0.1   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | x1 | x2 | ydesired | w1 | w2 | z | y | w1-new | w2-new | | 0 | 0 | 0 | 0.3 | -0.1 | = 0 \* 0.3 + 0 \* -0.1 – 0.2  = -0.2 | = step(-0.2) = 0 | = 0.3 + 0.1 \* (0 - 0) \* 0  = 0.3 | = -0.1 + 0.1 \* (0 - 0) \* 0  = -0.1 | | 0 | 1 | 0 | 0.3 | -0.1 | = 0 \* 0.3 + 1 \* -0.1 – 0.2  = -0.3 | = step(-0.2) = 0 | = 0.3 + 0.1 \* (0 - 0) \* 0  = 0.3 | = -0.1 + 0.1 \* (0 - 0) \* 1  = -0.1 | | 1 | 0 | 0 | 0.3 | -0.1 | = 1 \* 0.3 + 0 \* -0.1 – 0.2  = 0.1 | = step(0.1)  = 1 | = 0.3 + 0.1 \* (0 - 1) \* 1  = 0.2 | = -0.1 + 0.1 \* (0 - 1) \* 0  = -0.1 | | 1 | 1 | 1 | 0.2 | -0.1 | = 0.2 \* 1 – 0.1 \* 1 – 0.2  = -0.1 | = step(-0.1)  = 0 | = 0.2 + 0.1 \* (1 - 0) \* 1  = 0.3 | = -0.1 + 0.1 \* (1 – 0) \* 1  = 0 | |
| **ANN** | Insted of having input layer and outpyt layer, The ANN includes mutiple hidder layers  x1  x2  x3  Input layer. Hidden Layer 1. Hidden Layer 2. Output Layer  y1  y2  y3  W1  W2  W3  Notation      …    …      …  …      …    : sum of input \* weight of particular layer of an particular unit  : the actual output after applied activation function of particular layer of a particular unit  : the amount of error of particular layer of particular unit  : desired output with particular index Feed forward neural networkBackpropagation neural network Output Layer ErrorHidden layer error |
| **Activation function** | Sigmoid or Logistic Activation function   Range: [0, 1]  It is especially used for models where we have to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, sigmoid is right choice. Tanh   Range: [-1, 1] ReLU Range: [0, infinity]  refs: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 |