|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Input | | output | Comments |
| I1 | I2 |
| Training Data | 0 | 0 | 0 (Label Data) | Supervised Learning |
| 1 | 1 | 1 (Label Data) |
| Testing Data | 0 | 1 | ? | Open Test  (Intelligence OK ?) |
| 1 | 0 | ? |
| Testing Data | 0 | 0 | ? | Close Test  (Learning OK?) |
| 1 | 1 | ? |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Input | | Output | Comments |
| I1 | I2 |
| Training Data | 0 | 0 | No Label Data | Unsupervised Learning |
| 1 | 1 |
| 0 | 1 |
| Testing Data | 0 | 0 | Cluster-1 | Considering two clusters |
| 1 | 1 | Cluster-2 |
| 0 | 1 | Cluster-2 |

|  |
| --- |
| Supervised Machine Learning Algorithms |
| Artificial Neural Networks (ANN) |
| Support Vector Machine (SVM) |
| Decision Tree (DT) |
| Hidden Markov Model (HMM) |
| Nearest Neighbor (NN) |
| k-Nearest Neighbor (KNN) |
| Naïve Bayes (NB) Classifier |
| etc |

|  |
| --- |
| Unsupervised Machine Learning Algorithms |
| k-Mean Clustering |
| Hierarchical Clustering |
| Self-Organizing Map (SOM) |
| Gaussian Clustering |
| Fuzzy C-Mean Clustering |
| DBSCAN |
| etc |

Neuron Model:

W1

I2

I1

output

u=I1W1+I2W2

W2

Here, output=f(u) where f is activation function

Activation function:

1. Sigmoid
2. Softmax
3. Relu
4. Signum
5. Piece-wise Linear

Before Learning:

Synaptic Weights=W1, W2

After Learning:

Synaptic Weight will be updated

W1=W1+ΔW1

W2=W2+ΔW2

Problem:

Find output using the Neural Network for the Testing Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Input | | output | Comments |
| I1 | I2 |
| Training Data | 0 | 0 | 0=d1 | Pattern-I |
| 1 | 1 | 1=d2 | Pattern-II |
| Testing Data | 0 | 1 | ? |  |

Assume, Learning rate, η=0.1

Neural Network Architecture:

I1

W1

Output=f

δ=Error gradient

W2

I2

Initial value of W1=0.2 and W2=0.3

Activation Function is sigmoid where f=1/(1+e^(-u))

Complete one epoch of Neural Network

Solution:

Training the Neural Network by using the Pattern-I where I1=0, I2=0 and d1=0:

**Forward Pass:**

u=I1W1+I2W2=0\*0.2+0\*0.3=0,f =1/(1+e^(-u)) =0.5

**Backpropagation:**

***δ Calculation:***

δ=f1(d1-f) [f1 =first derivative of f= f (1-f) ]

=f(1-f)(d1-f)

=0.5\*(1-0.5)\*(0-0.5)

=-0.125

***Weight Update:***

W1=W1+ΔW1 W2=W2+ΔW2

=W1+ η\* δ\*I1 =W2+ η\* δ\*I2

=0.2+0.1\*(-0.125)\*0 =0.3+0.1\*(-0.125)\*0

=0.2 =0.3

Training the Neural Network by using the Pattern-II where I1=1, I2=1 and d2=1:

**Forward Pass:**

u=I1W1+I2W2=1\*0.2+1\*0.3=0.5,f =1/(1+e^(-u)) =0.621

**Backpropagation:**

***δ Calculation:***

δ=f1(d2-f) [f1 =first derivative of f= f (1-f) ]

=f(1-f)(d2-f)

=0.621\*(1-0.621)\*(1-0.621)

=0.0892

***Weight Update:***

W1=W1+ΔW1 W2=W2+ΔW2

=W1+ η\* δ\*I1 =W2+ η\* δ\*I2

=0.2+0.1\*0.0892\*1 =0.3+0.1\*0.0892\*1

=0.20892 =0.30892

***One epoch of Training is completed***

Testing the Neural Network by using the Testing Pattern, I1=0 and I2=1:

**Forward Pass:**

u=I1W1+I2W2=0\*0.20892+1\*0.30892=0.30892,f =1/(1+e^(-u)) =0.58 >0.4, output=1