**Aim**: Analysis of Mushroom Dataset in R using Deep Neural Network

**Theory**:

**Deep** **Neural Networks:**

A deep neural network (DNN) is an artificial neural network (ANN) with multiple hidden layers between the input and output layers. Each such layer has multiple neurons. Each neuron has multiple inputs, an activation function and one output. This neuron is a place where computation happens, loosely patterned on a neuron in the human brain, which fires when it encounters sufficient stimuli.

DNNs can model complex non-linear relationships. The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network.

Deep learning maps inputs to outputs. It finds correlations. It is known as a “universal approximator”, because it can learn to approximate an unknown function f(x) = y between any input x and any output y, assuming they are related at all (by correlation or causation, for example). In the process of learning, a neural network finds the right f, or the correct manner of transforming x into y, whether that be f(x) = 3x + 12 or f(x) = 9x – 0.1.

**Classification:**

All classification tasks depend upon labeled datasets; that is, humans must transfer their knowledge to the dataset in order for a neural network to learn the correlation between labels and data. This is known as supervised learning.

**Working:**

DNN uses multiple layer feedforward network architecture. The first layer is the input layer in which user input data is fed as input. The output of the activation function of this layer is fed to the next layer. This goes on until final layer is reached. In DNN, the number of rounds, the learning rate and momentum are the parameters that decide the accuracy of the training.

Once the network is trained, it is tested with separate test cases. The accuracy can be calculated at this step.

**Strengths:**

* Has best-in-class performance on problems that significantly outperforms other solutions in multiple domains. This includes speech, language, vision, playing games like Go etc.
* Reduces the need for feature engineering, one of the most time-consuming parts of machine learning practice.
* Is an architecture that can be adapted to new problems relatively easily.

**Limitations:**

* Training is time consuming and its time complexity increases with increase in network architecture complexity and dataset size.

**Dataset:**

1. Title: Mushroom Database
2. Sources:
   * 1. Mushroom records drawn from The Audubon Society Field Guide to North
3. American Mushrooms (1981). G. H. Lincoff (Pres.), New York: Alfred Knopf
   * 1. Donor: Jeff Schlimmer (Jeffrey.Schlimmer@a.gp.cs.cmu.edu)
     2. Date: 27 April 1987
4. Relevant Information:

This data set includes descriptions of hypothetical sample corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500-525). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like ``leaflets three, let it be'' for Poisonous Oak and Ivy.

1. Number of Instances: 8124
2. Number of Attributes: 22 (all nominally valued)
3. Missing Attribute Values: 2480 of them (denoted by "?"), all for Attribute.
4. Class Distribution:

* edible: 4208 (51.8%)
* poisonous: 3916 (48.2%)
* total: 8124 instances

1. Attribute Information: (class: edible=e, poisonous=p)

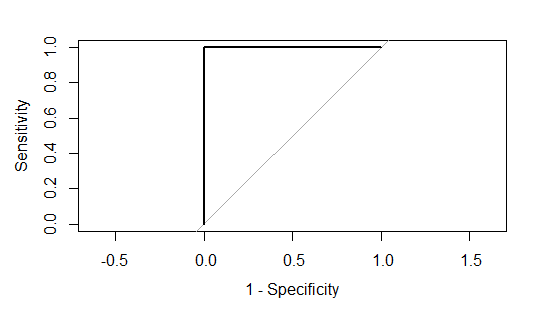
|  |  |  |
| --- | --- | --- |
| **Sr.** | **Feature name** | **Attributes** |
| 1 | cap-shape | bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s |
| 2 | cap-surface | fibrous=f, grooves=g, scaly=y, smooth=s |
| 3 | cap-colour | brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y |
| 4 | Bruises | bruises=t, no=f |
| 5 | Odor | almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s |
| 6 | gill-attachment | attached=a, descending=d, free=f, notched=n |
| 7 | gill-spacing | close=c, crowded=w, distant=d |
| 8 | gill-size | broad=b, narrow=n |
| 9 | gill-color | black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y |
| 10 | stalk-shape | enlarging=e, tapering=t |
| 11 | stalk-root | bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=? |
| 12 | stalk-surface-above-ring | fibrous=f, scaly=y, silky=k, smooth=s |
| 13 | stalk-surface-below-ring | fibrous=f, scaly=y, silky=k, smooth=s |
| 14 | stalk-color-above-ring | brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y |
| 15 | stalk-color-below-ring | brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y |
| 16 | veil-type | partial=p, universal=u |
| 17 | veil-color | brown=n, orange=o, white=w, yellow=y |
| 18 | ring-number | none=n, one=o, two=t |
| 19 | ring-type | cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z |
| 20 | spore-print-color | black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y |
| 21 | Population | abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y |
| 22 | Habitat | grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d |

**Observations for mushroom dataset**

**Confusion Matrix and Statistics:**

|  |  |  |
| --- | --- | --- |
| Prediction\Reference | e | P |
| e | 1262 | 0 |
| p | 0 | 1175 |

1. Accuracy: 1
2. Area Under Curve of ROC: 1
3. Sensitivity: 1
4. Specificity: 1
5. Precision: 1
6. Recall: 1
7. ROC:

****

**Conclusion:**

 In this project we learned –

1. Basic commands of R language and its usage for data mining.
2. Analysis of datasets based on their size, attributes, class and correlation between them.
3. Deep Neural Network implementation concepts.

We profess the following inferences –

Accuracy of classification can be maximized in DNN by evaluating a proper structure of hidden layers. (We employed trail and error method to find out this proper structure.)