Aim: Comparative analysis of Datasets 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.

Algorithm: DNN

Step 1: Input at the input layer

Step 2: Calculate net o/p and output at the next hidden layer

Step 3: Repeat step 2 for all hidden layers

Step 4: Calculate net output for output layer

Step 5: Calculate error at output layer by using *desired output*.

Step 6: Calculate error at hidden layers.

Step 7: Update weights and biases for all layers.

Step 8: Repeat Step 2 to Step 7 until min. threshold < threshold.

Algorithm Rprop

1: $\eta+ = 1.2, \eta- = 0.5, \Delta max = 50, \Delta min = 10-6$

2: pick $\Delta ij\ (0)$

3: $\Delta wij\ (0) = -\ sgn\ \partial E(0)\ \partial wij \cdot \Delta ij\ (0)$

4: for all $t \in [1..T\ ]$ do

5: if $\partial E(t)\ \partial wij \cdot \partial E(t-1)\ \partial wij > 0$ then

6: $\Delta ij\ (t) = \min\{\Delta ij\ (t-1) \cdot \eta+, \Delta max\}$

7: $\Delta wij\ (t) = -\ sgn\ \partial E(t)\ \partial wij \cdot \Delta ij\ (t)$

8: $wij\ (t+1) = wij\ (t) + \Delta wij\ (t)$

9: $\partial E(t-1)\ \partial wij = \partial E(t)\ \partial wij$

10: else if $\partial E(t)\ \partial wij \cdot \partial E(t-1)\ \partial wij < 0$ then

11: $\Delta ij\ (t) = \max\{\Delta ij\ (t-1) \cdot \eta-, \Delta min\}$

12: $\partial E(t-1)\ \partial wij = 0$

13: else

14: $\Delta wij\ (t) = -\ sgn\ \partial E(t)\ \partial wij \cdot \Delta ij\ (t)$

15: $wij\ (t+1) = wij\ (t) + \Delta wij\ (t)$

16: $\partial E(t-1)\ \partial wij = \partial E(t)\ \partial wij$

17: end if

18: end for

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.

Datasets:

1. Mushroom Dataset:

Description:

1. Title: Mushroom Database
2. Sources:

   Mushroom records drawn from The Audubon Society Field Guide to NorthAmerican Mushrooms (1981). G. H. Lincoff (Pres.), New York: Alfred Knopf

   Donor: Jeff Schlimmer (Jeffrey.Schlimmer@a.gp.cs.cmu.edu)

3. 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.
5. Number of Instances: 8124
6. Number of Attributes: 22 (all nominally valued)

7. Missing Attribute Values: 2480 of them (denoted by "?"), all for Attribute.
8. Class Distribution:
    i. edible: 4208 (51.8%)
    ii. poisonous: 3916 (48.2%)
    iii. total: 8124 instances
9. 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

2. DDSM dataset:

Description:

1. Title: Breast cancer
2. Creator:

1. Dr. William H. Wolberg, General Surgery Dept. University of Wisconsin, Clinical Sciences Center
Madison, WI 53792
wolberg '@' eagle.surgery.wisc.edu
Relevant Information:
2.W. Nick Street, Computer Sciences Dept. University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
street '@' cs.wisc.edu 608-262-6619

3. Olvi L. Mangasarian, Computer Sciences Dept. University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
olvi '@' cs.wisc.edu
Donor: Nick Street

3. **Relevant Information:**

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.
Separating plane described above was obtained using Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree Construction Via Linear Programming." Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society, pp. 97-101, 1992], a classification method which uses linear programming to construct a decision tree. Relevant features were selected using an exhaustive search in the space of 1-4 features and 1-3 separating planes.

The actual linear program used to obtain the separating plane in the 3-dimensional space is that described in: [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming

Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1,1992,23-34].

This database is also available through the UW CS ftp server:
ftp ftp.cs.wisc.edu
cd math-prog/cpo-dataset/machine-learn/WDBC/

4. Number of Instances: 321
5. Number of Attributes: 10
6. Missing Attribute Values: N/A
7. Class Distribution:
    i. Benign: 357 (62.75%)
    ii. Malignant: 212 (37.25%)
    iii. total: 569 instances
8. Attribute Information:

1) ID number
2) Diagnosis (M = malignant, B = benign)
Ten real-valued features are computed for each cell nucleus:
a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)

3. MIAS dataset:

Description:

1. Title: Breast Cancer
2. Relevant Information:
3. Number of Instances: 569
4. Number of Attributes: 31
5. Missing Attribute Values: -
6. Class Distribution:
    i. Benign: 357 (62.74%)
    ii. Malignant: 212 (37.25%)
    iii. total: 569 instances

7. Attribute Information: (class: Benign=B, Malignant=M)

4. Eye dataset:

Description:

1. Title: EEGEye Detection Database
2. Sources:

    i. Baden-Wuerttemberg Cooperative State University (DHBW), Stuttgart, Germany
    ii. Donor: Oliver Roesler, it12148 '@' lehre.dhbw-stuttgart.de
    iii. Date: 6th October 2013.

3. Relevant Information: All data is from one continuous EEG measurement with the Emotiv EEG Neuroheadset. The duration of the measurement was 117 seconds. The eye state was detected via a camera during the EEG measurement and added later manually to the file after analyzing the video frames. '1' indicates the eye-closed and '0' the eye-open state. All values are in chronological order with the first measured value at the top of the data.

4. Number of Instances: 14980
5. Number of Attributes: 15
6. Missing Attribute Values: N/A
7. Class Distribution:
    i. Open state: 10,486 (70%)
    ii. Close state: 4494(30%)
    iii. total: 14980 instances

8. Attribute Information: (class: Open State=1, Close state=0)

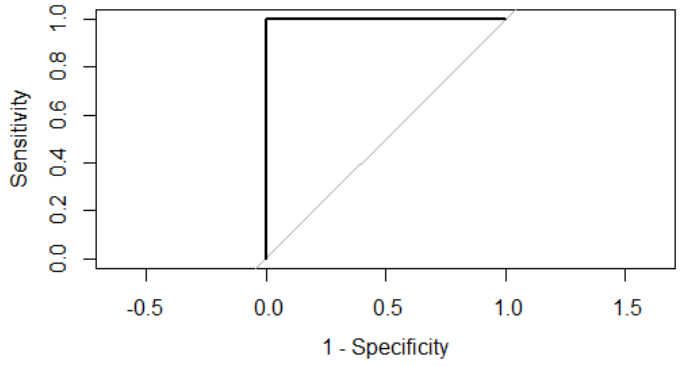
Observations:

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:



Hyper-parameter adjustments:

Algorithm	Error function	Activation function	Accuracy (%)	Hidden layers						
rprop+	sse	logistic	100	10	5					
rprop+	sse	logistic	100	10	10	10	10			
rprop+	sse	logistic	100	10	10	10	10	10		
rprop+	sse	logistic	100	10	10	10				
rprop+	sse	logistic	100	10	10					
rprop+	sse	logistic	99.95	8	8	8				
rprop+	sse	logistic	99.5	3		quickly				
rprop+	sse	logistic	99.5	10						
rprop+	sse	logistic	99.5	100						
rprop+	sse	logistic	99.3	9	9	9				
rprop+	sse	logistic	99.3	11	11					
rprop+	sse	logistic	99.26	5	5	5				
rprop+	sse	logistic	98.81	4	4	4				
rprop+	sse	logistic	98.6	7	7	7				
rprop+	sse	logistic	98.195	5						
rprop+	sse	logistic	97.74	2	2	2				
rprop+	sse	tanh	97.74	3		long delay				
rprop+	sse	logistic	97.74	10	10	10	10	10	1	
rprop+	sse	logistic	97.66	2						
rprop+	sse	tanh	97.66	2						
rprop+	sse	logistic	97.66	3	3	3				
rprop+	sse	logistic	97.046	1	1	1				
rprop+	sse	logistic	97.046	1	1					
rprop+	sse	logistic	97.046	1						
rprop+	sse	logistic	97.046	2	1	1				
rprop+	sse	logistic	94.91	1	1	1	1			
rprop+	sse	logistic	92.73	1	1	1	1	1		
rprop+	sse	tanh	0	1						
backprop	sse	logistic	0	10	10	10				
backprop	sse	tanh	0	10	10	10				

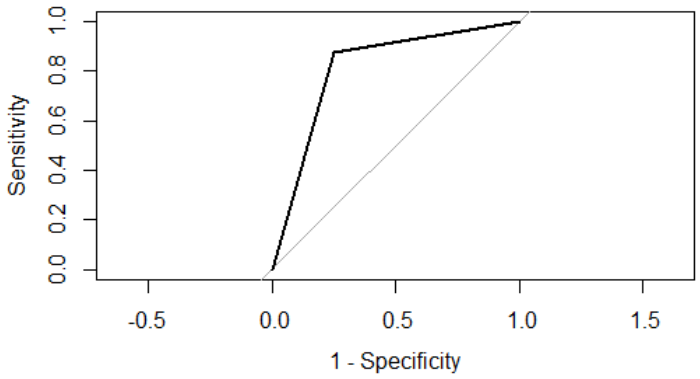
Observations for DDSM dataset:

Confusion Matrix and Statistics:

Prediction\Reference	B	M
B	12	4
M	2	14

1. Accuracy: 0.8125
2. Area Under Curve of ROC: 0.8125
3. Sensitivity: 0.875
4. Specificity: 0.75
5. Precision: 0.75
6. Recall: 0.8571

7. ROC:



Hyper-parameter Analysis:

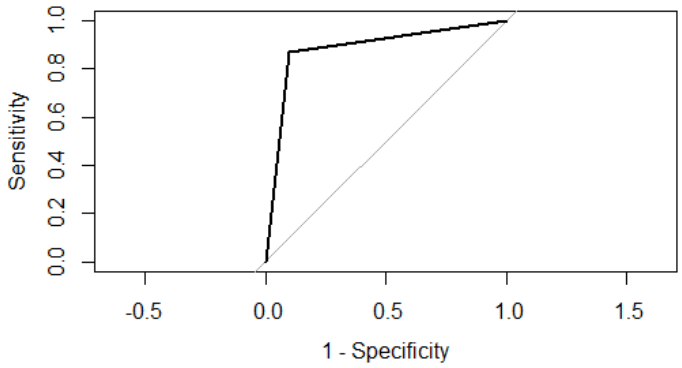
Algorithm	Error function	Activation function	Accuracy (%)	Hidden layers						
rprop+	sse	logistic	81.25	1	1	1	1			
rprop+	sse	logistic	81.25	1	1	1				
rprop+	sse	logistic	81.25	1	1					
rprop+	sse	logistic	81.25	1	2					
rprop+	sse	logistic	81.25	1	5					
rprop+	sse	logistic	81.25	1	10	10	1			
rprop+	sse	logistic	81.25	2	1					
rprop+	sse	logistic	81.25	2	2	2				
rprop+	sse	logistic	81.25	2	2	3				
rprop+	sse	logistic	81.25	2	2	5				
rprop+	sse	logistic	78.12	100	1	10				
rprop+	sse	logistic	75	1	10	1	1			
rprop+	sse	logistic	75	1	10					
rprop+	sse	logistic	75	1						
rprop+	sse	logistic	75	2	2	1				
rprop+	sse	logistic	75	101	1	10				
rprop+	sse	logistic	71.88	1	1	1	10			
rprop+	sse	logistic	71.88	1	1	10	1			
rprop+	sse	logistic	71.88	1	10	100	1			
rprop+	sse	logistic	71.88	2	2	2	2			
rprop+	sse	logistic	62.5	1	2	1				
rprop+	sse	logistic	59.68	2						
rprop+	sse	logistic	56.25	2	2					
rprop+	sse	logistic	did not converge	3						
rprop+	sse	logistic	did not converge	5						
rprop+	sse	logistic	did not converge	10						

Observations for MIAS dataset:

Confusion Matrix and Statistics:

Prediction\Reference	0	1
0	27	3
1	4	26

1. Accuracy: 0.8833
2. Area Under Curve of ROC: 0.8833
3. Sensitivity: 0.8667
4. Specificity: 0.9
5. Precision: 0.8709
6. Recall: 0.9
7. ROC:



Hyper-parameter Analysis:

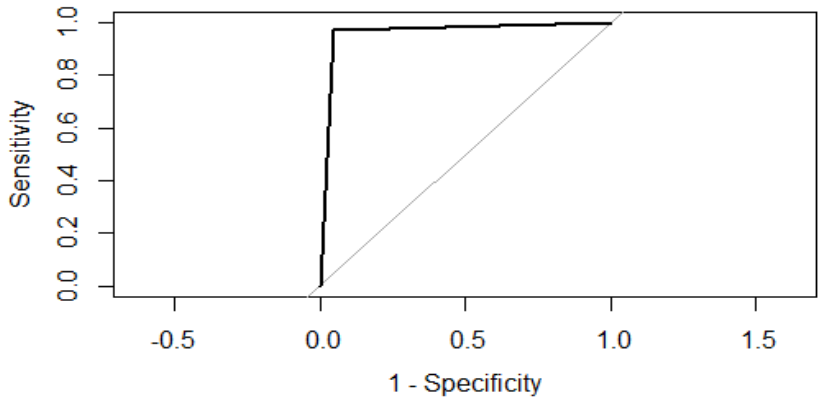
Algorithm	Error function	Activation function	Accuracy (%)	Hidden layers						Rep
rprop+	sse	logistic	88.33	1000	10	5	1			1
rprop+	sse	logistic	85	500	10	5	1			1
rprop+	sse	logistic	85	1000	100	10	5	1		1
rprop+	sse	logistic	80	1000	10	5				1
rprop+	sse	logistic	78.33	1000	10	5	2	1		1
rprop+	sse	logistic	78.33	1000	10	10				1
rprop+	sse	logistic	76.667	1000	100	10				1
rprop+	sse	logistic	63.33	1000	5					1
rprop+	sse	logistic	61.667	10	10	5				1
rprop+	sse	logistic	61.667	10	10	5				2
rprop+	sse	logistic	61.667	1000	1000	5				1
rprop+	sse	logistic	56.667	500	5	1				1
rprop+	sse	logistic	56.667	1000	10					1
rprop+	sse	logistic	50	10	10	10				1
rprop+	sse	logistic	50	1000	1					1
rprop+	sse	logistic	50	1000	5	1				1
rprop+	sse	logistic	50	1000	64	5				1
rprop+	sse	logistic	41.667	10	5					1

Observations for Eye Dataset:

Confusion Matrix and Statistics: For 4001 rows starting from 3000 in dataset

Prediction\Reference	0	1
0	561	27
1	17	595

1. Accuracy: 0.9633
2. Area Under Curve of ROC: 0.9632
3. Sensitivity: 0.9706
4. Specificity: 0.9722
5. Precision: 0.9541
6. Recall: 0.9706
7. ROC:



Hyper-parameter Analysis:

Dataset	Hidden Layer		stepmax	rep	Result	Accuracy
3000 to 7000	10	5	100000	1	Yes	0.9633
1 to 4000	10	5	500000	1	Yes	0.96
1 to 4000	10	5	100000	1	No	-
Whole	10		100000	1	No	-
Whole	10	5	100000	1	No	-
Whole	10	5	500000	1	No	-
Whole	10	5	500000	2	No	-

Comparison:

Dataset	**Accuracy(%)**	**Area under Curve**
Mushroom	100	1
DDSM	81.25	0.8125
MIAS	88.33	0.8833
Eye	96.33	0.9632

DNN showed these characteristics on different datasets:

Mushroom Dataset: Required default number of steps to converge.

MIAS Dataset: Required default number of steps to converge. However, the alterations of split ratio affected the accuracy. Max. accuracy for split ratio 0.79 (99.167).

DDSM Dataset: Required default number of steps to converge. However, DNN converged for only a limited set of hidden layer combinations.

EYE Dataset: Did not converge for any combination on taking whole dataset. DNN worked when we reduced dataset to 25%.

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 empirical method to find out this proper structure.)
- The accuracy of any given dataset fundamentally depends on the Number of entries in the dataset and the relation between the target attribute and the attribute chosen for classification.
- DNN takes very long time to train on larger datasets.
- The time required to train the neural network using “logistic” activation function is far less than that by “tanh” activation function.