Improvised Nature Inspired Deep Belief Network & its application in heart disease prediction

May-September,2016

Under the guidance of
Prof. Anupam Shukla & Dr. Ritu Tiwari

Presented by :

Kanika Singhal (2013IPG-056)
Ritu Jha (2013IPG-092)
Sarthak Joshi (2013IPG-096)

Introduction

- **Earlier**, the use of computer was to build knowledge from medical experts.
- This **time consuming** process depends on medical experts' **subjective** opinions.
- **Machine learning** techniques have been developed to gain knowledge automatically from raw data.
- Here, an **improvised deep belief network** is presented for the diagnosis of heart disease.

DEEP LEARNING

- **Training** data through **multiple** neuron layers.
- It allows to solve **complex** problems.
- This creates **multiple layers of abstraction**.
- Deep learning achieves desired results in many application-
  Example: Recognition of complex patterns by using many layers-first layer can recognize simple edges , second layer can recognize the figures like rectangle and triangle and higher layers can be used to recognize complex shapes .
- Here, **deep belief network**, a part of deep learning is used

Deep Belief Network

- **DBN = RBM + NN**
- Multi-layer belief network
- **Multiple RBM** layers stacked together to form DBN
- **Topmost layer** is a perceptron layer which classifies the data which have come through RBM layers
- **Greedy layer-wise** training in which most optimized trained value of one hidden layer is passed on to next layer

RBM

- **Boltzmann machine** are bidirectional graph used to model the **probability distribution**
- Graph is **bipartite**
- Consists of hidden layer and visible layer with weight distribution between them
- extract features and reconstruct input

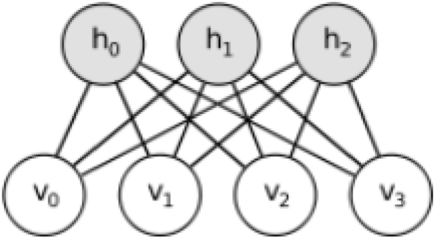

Figure: Structure of RBM[1]

Probability Distribution by RBM

STEP 1:

Energy function E(v,h)[2] defining energy distribution for the network is:

$$\mathbf{E(v, h) = h'Wv + b'v + c'h} \tag{1}$$

h,v : activations of hidden and visible layer

b,c : biases of hidden and visible layer

W : weight matrix of RBM

STEP 2:

Joint probability distribution of RBM:

$$\mathbf{p(v, h) = \frac{1}{N} e^{E(v,h)}} \tag{2}$$

[2]Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H. et al.: 2007,Greedy layer-wise training of deep networks,Advances in neuralinformation processing systems19, 153

STEP 3:

Conditional probability distribution of RBM

The conditional probabilities associated with the joint probability distribution are denoted by Q(h|v) and P(v|h).

$$P(v_k = 1|h) = sigm(-b_k - \sum_j (W_{jk} * h_j)) \tag{3}$$

$$Q(h_j = 1|v) = sigm(-c_j - \sum_k (W_{jk} * v_k)) \tag{4}$$

Using above, P and Q **sampling** on RBM unit is carried out.

STEP 4 AND 5:

STEP 4: We calculate the log likelihood of the input observations

$$log(P(v_0)) = log(\sum_h P(v_0, h)) = log(\sum_h (e^- E(v_0, h))) - log(\sum_{h,v} (e^- E(v, h))) \tag{5}$$

Here L is the learning rate

STEP 5: By applying gradient descent we minimize the negative log likelihood of v_0

Training of DBN

1. X be a feature **input matrix**
2. Training of RBM on X to obtain is done to have its **weight matrix W**
3. X is **transformed** by RBM and a new data X' is formed.
4. Repeating this procedure with **X** $\leftarrow$ **X'** for the next pair of layers till the two top layers of network are reached.
5. **Fine-tuning** of all the parameters of this deep architecture with **supervised training.**

Particle Swarm Optimisation

- PSO[3] was introduced by James Kennedy and Russel Eberhart in **1995**
- Used for optimising a **non-linear function**
- Based on **information sharing** between the agents in the search space.
- Velocity and position of particle:

$$U_i(k+1) = U_i(k) + c_1 r_1 (P_i - X_i(k)) + c_2 r_2 (P_g - X_i(k)) \tag{6}$$

$$X_i(k+1) = X_i(k) + U_i(k+1) \tag{7}$$

[3] Eberhart, R. C., Kennedy, J. et al.: 1995, A new optimizer usingparticle swarm theory,Proceedings of the sixth internationalsymposium on micro machine and human science, Vol. 1, New York,NY, pp. 3943

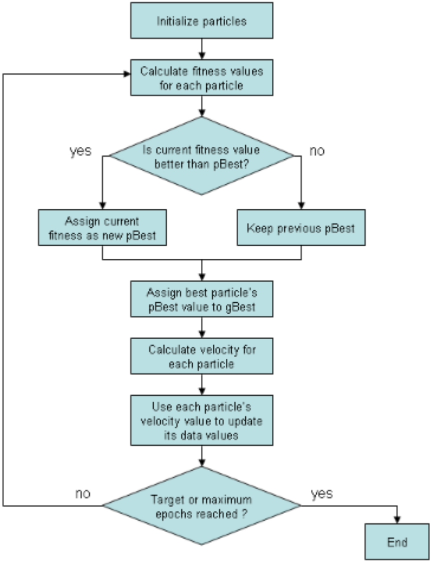

Figure: Flowchart of PSO

Motivation

- Clinical decisions are often taken on the basis of doctors perception and experience rather than on the knowledge rich data masked in the database. Every doctor is not evenly expert in every sub specialty and they are in several places a scarce resource.
- Therefore , automatic , precise , accurate and fast classification of medical heart disease data is today's need.

Background

- Data mining UCI repository dataset[4]
- Paper[5] describes deep belief nets as probabilistic generative models that are composed of multiple layers of stochastic latent variables (also called "feature detectors" or "hidden units").
  (MNIST dataset , CD , SVM , Limitations , large amount of time)

[4] eart data set, U.: uci, Uci heart data set

[5] Hinton, G.: 2011, Deep belief nets,Encyclopedia of Machine Learning, Springer,pp. 267269

- In this paper[6]. PSO is compared with the other algorithms and it was found that it needs fewer parameters .The paper concludes with the fact that more research can be made on PSO by improving the topology of particle swarm, by blending with other intelligent optimization algorithm.
- The best accuracy achieved on:
  Cleveland : 90% using SVM+GA [9] (Bhatia,2008)
  Hungarian : 92% using k-NN [12] (Setiawan,2009)

[6] Bai, Q.: 2010, Analysis of particle swarm optimization algorithm,Computer andinformation science3(1), 180

Table 1 PSO publications and citations by year

Year	Google Scholar Hits	IEEE Xplore Publications
1995	(28)	(2)
1996	(14)	(0)
1997	(24)	(3)
1998	(49)	(4)
1999	(62)	(7)
2000	(78)	(8)
2001	(118)	(11)
2002	(256)	(46)
2003	(373)	(82)
2004	(850)	(181)
2005	(1210)	(279)
2006	(1230)	(462)

[7] Kennedy, J.: 2011, Particle swarm optimization, Encyclopedia of machine learning,

Research Gap

- There are many hyper parameters to tune like number of layers, number of hidden units, learning rate and many more.The eciency of function is determined by these parameters therefore it is necessary to optimize all these parameters to get accurate results
- To optimize all these input parameters its takes a considerable amount of time . Therefore there is a need of some algorithm which can optimize these parameters in limited time giving better performance than simple DBN.

Problem Statement

- On time diagnosis of heart disease is necessary for patient from a cardiologist in an ideal situation. In case of emergencies we need some fast and efficient alternatives.
- The **large number of parameters** required by deep belief network like **regularization and dimensions of neuron and RBM layer** have a **non-linear dependency** which causes difficulty in training it.

Aim

- To construct a **classifier** using **deep belief network** and further optimize it by **nature inspired particle swarm optimization**

Objective

- To predict heart disease risk using **deep belief network optimized using PSO**
- **Comparison** of results with existing works done on the chosen heart disease dataset

Data set description

UCI repository dataset[8]

- **Cleveland clinical foundation Database**
  303 records of patients
  Each record has 13 attributes
  The output is either 0,1,2,3 or 4
  1,2,3 or 4: patient has heart disease(based on severity 1-4)
  0 : normal person
- **Hungarian Database**
  It has 294 records. Each record has 13 attributes. The output to each record is either 0 or 1. 0 indicates normal person and 1 indicates abnormal person.

[8] Lichman, M.: 2013, UCI machine learning repository.http:archive.ics.uci.edu/ml

Attribute	Description	Value Description
age	Age	Numerical
sex	Sex	1 if male, 0 if female
cp	Chest pain type	1 to 4
trestbps	Resting systolic blood pressure	Numerical
chol	Serum cholestrol(mg/dl)	Numerical
fbs	Fasting blood sugar	1 if yes and 0 if no
restecg	Resting cardigraphic results	0 to 2
thalach	Maximum heart rate achieved	Numerical
exang	Exercise induced angina	1 if yes and 0 if no
oldpeak	ST depression induced by excercise relative to rest	Numerical
slope	The slope of the peak exercise ST segment	1 upsloping and 2 flat
ca	Number of major vessels	Numerical
thal		3,6 or 7
num	Output Attribute	0 or 1

Figure: Summary of disease attributes[9]

Pseudo code of DBN

RBMupdate(v[0], epsilon, W, b, c)

```
for all hidden units i
    compute Q(h[0][i]=1|v[0])# sigmoid(b[i]+sum_j(W[i][j]*v[0][j]))
    sample h[0][i] from Q(h[0][i]=1|v[0])
for all visible units j:
    compute P(v[1][j] = 1|h[0]) #sigmoid(c[j] + sum_i(W[i][j]*h[0][i]))
    sample v[1][j] from P(v[1][j] = 1|h[0])
for all hidden units i:
    compute Q(h[1][i] =1|v[1])#sigmoid(b[i]+sum_j(W[i][j]*v[1][j]))
W += epsilon * (h[0] * v[0]' - Q(h[1][.] = 1 | v[1]) * v[1]')
a += epsilon * (h[0] - Q(h[1][.] = 1| v[1]))
b += epsilon * (v[0] - v[1])
```

PreTrainSupervisedDBN(Z, epsilon, L, n, W, b, V, c)

```
let X the marginal over the input part of Z
initialize b[0]=0
for l=1 to L-1
    initialize W[i]=0, b[i]=0
    while not stopping criterion:
        sample g[0]=x from X
        for i=1 to l-1:
            sample g[i] from Q(g[i]|g[i-1])
    RBMupdate(g[l-1], epsilon, W[l], b[l], b[l-1])
initialize W[L]=0, b[L]=0, V=0, c=0
while not stopping criterion:
    sample (g[0]=x, y) from Z
    for i=1 to L-1:
        sample g[i] from Q(g[i]|g[i-1])
    RBMupdate((g[L-1],y), epsilon, (W[L],V'), b[L], (b[L-1],c))
```

DBNSupervisedFineTuning(Z, C, epsilon_C, L, n, W, b, V,c)

*Define the network output function f(x) = V * x + c where x is the data transformed through rbm layers*

Iteratively minimize the expected value of C(f(x),y)

for pairs (x,y) sampled from Z by tuning parameters W, b, V, c

This can be done by stochastic gradient descent with learning rate epsilon_C,

using appropriate stopping criterion such as early stopping on validation set

TrainSupervisedDBN(Z, C, epsilon_CD, epsilon_C, L, n, W, b, V):

1. *let X the marginal over the input part of Z*
2. *PreTrainSupervisedDBN(X, epsilon_CD, L, n, W, b)*
3. *DBNSupervisedFineTuning(Z, C, epsilon_C, L, n, W, b, V, c*

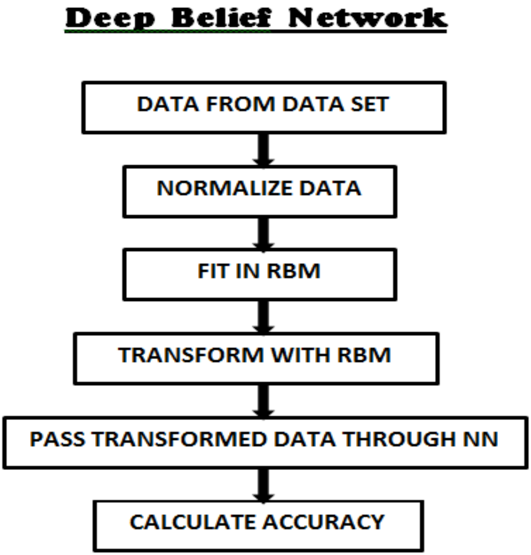

Figure: Deep belief network structure

Pseudo code for PSO

```
For each particle
    Initialize particle
Do until maximum iterations or minimum error criteria
    For each particle
        Calculate Data fitness value
        If the fitness value is better than pBest
            Set pBest = current fitness value
        if pbest is better than gbest
            Set gBest = pBest
     For each particle
           Calculate particle velocity
           Use gbest and Velocity to update particle data
```

Pseudo code for proposed algorithm

ModifiedTrainSupervisedDBN (Z, C, epsilon_CD, epsilon_C, L, n, W, b, V)

1. *let X the marginal over the input part of Z*
2. *PreTrainSupervisedDBN (X, epsilon_CD, L, n, W, b)*
3. *DBNSupervisedFineTuning (Z, C, epsilon_C, L, n, W, b, V, c)*
4. *Calculate accuracy of DBN with a testing data using W*
5. *return accuracy*

Pseudo code for proposed algorithm

ModifiedTrainSupervisedDBNwithPSO(Z,C,epsilonCD,epsilon C,L,W,b,v,range_n, swarmsize,iter)

1. *let vector s denote the swarm position of the particles*
2. *vector v denotes the velocity vector of the particles*
3. *initialize global_best = -infinity*
4. *initialize personal_best = -infinity*
5. *for i<swarmsize*
6. *initialize s[i] in range_n*
7. *increment i*
8. *while (iterations < iter)*
9. *for i<swarmsize*
10. *fitness_val=modifiedTrainSupervisedDBN(Z,C, epsilonCD, epsilonC,L,s[i],W,b,V)*

Pseudo code for proposed algorithm

```
    if fitness_val>particle_best
        set particle_best = fitness_val
     if particle_best > global_best
        set global_best=particle_best
    for i<swarmsize
        calculate the particle velocity for the particle i
        update s[i] according to v[i]
        use global best to update particle position
 return global_best that is the best dimension for the deep belief
network
```

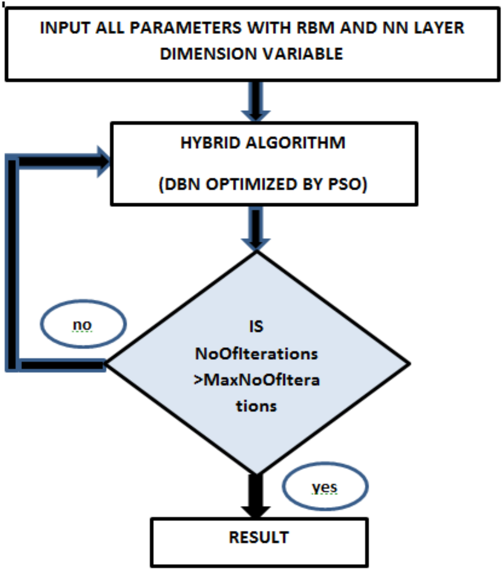

Figure: Proposed hybrid algorithm

Results

Table: **Comparison between existing & proposed algo-Cleveland**

Year	Algorithm	Accuracy
2000[13]	Active SVM	87%
2008[9]	SVM+GA	90%
2009[12]	FDSS	87%
Proposed	**DBN**	**88%**
Proposed	**DBN+PSO**	**91.18%**

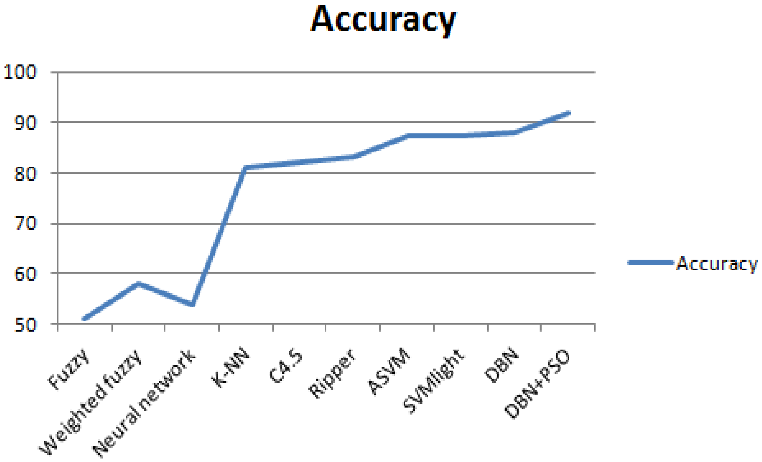

Figure: Accuracy of different algorithms on cleveland dataset

Results

Table: **Comparison between existing & proposed algo-HUNGARIAN**

Algorithm	Accuracy
Fuzzy[11]	36
NN[13]	46.417
Weighted Fuzzy [13]	50.83
MLP-ANN [12]	54
C4.5 [12]	66
Ripper [12]	68
K-NN [12]	92
Deep belief network	91.33
DBN+PSO(proposed Stategy)	**94.88**

Results

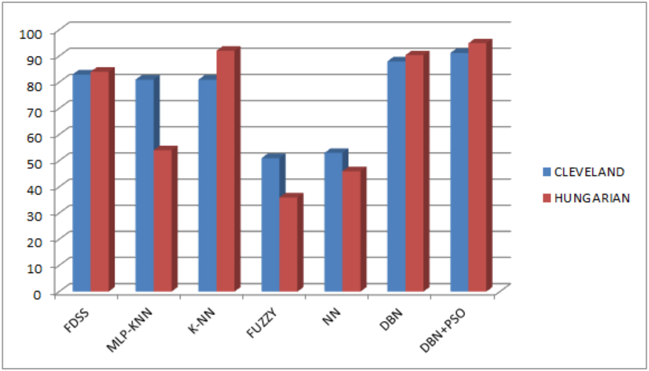

Figure: Comparison between algorithms

Discussions

- Comparison of proposed strategy with DBN
- Comparison of proposed strategy with existing algorithms applied on the same data set
- The algorithm used i.e. PSO for optimizing deep belief network is giving better results than simple deep belief network algorithm.
- The proposed algorithm gave better results than work done on the existing algorithm
- This is because we have taken into account the optimization of variables in our hybrid algorithm .

Conclusions

- Machine learning algorithms like DBN can be used to make the diagnosis fast and less costly.
- The proposed strategy of hybrid DBN algorithm with PSO successfully classified the heart disease type from the input data set provided.
- Mathematically simple[10] PSO can optimize multiple parameters of DBN at a time giving the most efficient results.
- It has been predicted in a very accurate manner whether a person has heart disease or not reducing the dependency on traditional techniques.

Future Scope

(i) This work can be extended by using other nature inspired algorithms like firefly , Ant colony Optimization(ACO),Flower pollination.

(ii) Also the system can be made for real time data.This will then take data in real time as well as predict or classify in real time whether a person has the disease or not.

(iii) Larger datasets can be managed using big data concepts.

References

9. Bhatia, Sumit, Praveen Prakash, and G. N. Pillai. "SVM based decision support system for heart disease classification with integer-coded genetic algorithm to select critical features." Proceedings of the World Congress on Engineering and Computer Science, WCECS. 2008.
10. Bai, Q.: 2010, Analysis of particle swarm optimization algorithm,Computerandinformation science3(1), 180
11. Anooj, P.: 2012, Clinical decision support system: risk level prediction of heartdisease using decision tree fuzzy rules,Int J Res Rev Cout Sci3(3).
12. Setiawan, N. A., Venkatachalam, P. and Hani, A. F. M.: 2009, Diagnosis of coro-nary artery disease using artificial intelligence based decision support system,proceedings of the international conference on man-machine systems (ICoMMS),Batu Ferringhi, Penang.
13. Mangasarian, O. L. and Musicant, D. R.: 2000, Active support vector machineclassification,NIPS, pp. 577583