Subject: CSP251 Project-Based Learning (PBL) -1
Professor: Dharm Raj; Sharda University, School of Engineering, Assistant Professor
Team members: Tuiba Ashraf
Priyal Rathore
Rachakonda Hrithik Sagar

Heart Disease Prediction System Using Supervised Learning Classifier

Abstract:

Cardiovascular disease remains the biggest cause of deaths worldwide and the Heart Disease Prediction at the early stage is important. In this paper Supervised Learning Algorithm is adopted for heart disease prediction at the early stage using the patient's medical record is proposed and the results are compared with the known supervised classifier Support Vector Machine (SVM). The information in the patient record is classified using a Cascaded Neural Network (CNN) classifier. In the classification stage, 14 attributes are given as input to the CNN classifier to determine the risk of heart disease. The proposed system will provide aid for the physicians to diagnose the disease in a more efficient way. The efficiency of the classifier is tested using the records collected from 304 patients. The results show the CNN classifier can predict the likelihood of patients with heart disease in a more efficient way

Keywords:

Cascaded Neural Network, Heart Disease Prediction, Support Vector Machine, Supervised Learning Algorithm

Software requirements:

- Processors:Intel® Core™ i5 processor 4300M at 2.60 GHz or 2.59 GHz (1 socket, 2 cores, 2 threads per core), 8 GB of DRAMIntel® Xeon® processor E5-2698 v3 at 2.30 GHz (2 sockets, 16 cores each, 1 thread per core), 64 GB of DRAMIntel® Xeon Phi™ processor 7210 at 1.30 GHz (1 socket, 64 cores, 4 threads per core), 32 GB of DRAM, 16 GB of MCDRAM (flat mode enabled)
- Processors: Intel Atom® processor or Intel® Core™ i3 processor
- Disk space: 1 GB
- Operating systems: Windows* 7 or later, macOS, and Linux
- Python* versions: 2.7.X, 3.6.X

Problem statement:

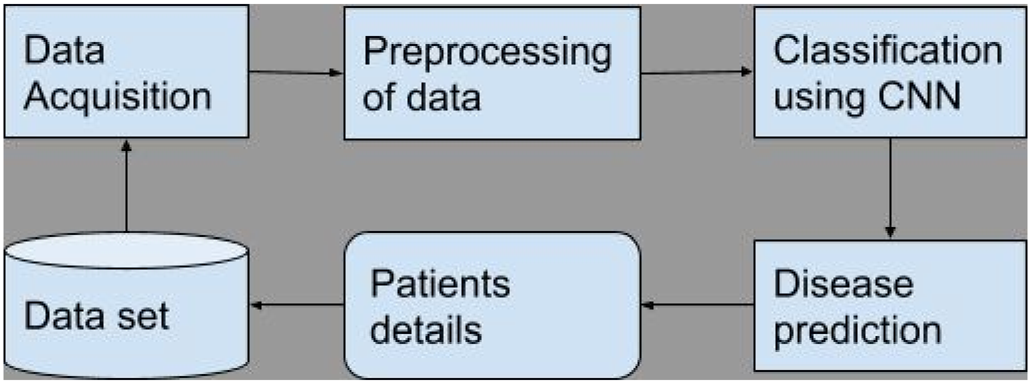
The analysis of disease is a vital job in medicine. The health care industry collects huge amount of healthcare data and then they are mined to discover hidden information for effective decision making. Cardiovascular disease is a kind of serious health imperiling and frequent happening disease. Cardiovascular diseases refer to any disease that affects the cardiovascular system. Medical diagnosis is considered a significant task that needs to be carried out precisely and efficiently. The automation of the same would be highly beneficial.

It's estimated that around 17.9 million people died (world) due to Cardiovascular disease(CVD) in the year 2016 out of this 85 % are due to heart disease and stroke i.e. 4 out of 5 CVD deaths are due to heart disease. These can be easily measured in primary care facilities, identifying those at highest risk of CVD's and ensuring they receive appropriate treatment can prevent premature deaths. by this way, we can save at least 25% of lives i.e. 4.4 million people. So this is our try towards saving the world from CVD attacks
Which would take the data from the user (around 14 attributes) and using artificial intelligence it predicts whether the user is having the risk of heart disease or not

Data set :

Data used for the proposed system is obtained from kaggle.com. The data have been collected from 304 patients are used for proposed work. This database contains 14 attributes. Cleaning and filtering of the data set are done to remove duplicate records, normalize the values, accounting for missing data and removing irrelevant data items.

Block diagram :



Characteristics of the patients like the number of times of chest pain and age in years were recorded. In Table , there are 14 attributes used in this system, including 8 symbolic and 6 numeric attributes Some other important parameters are variable parameters and that need to be checked for every 2 hours the latch (maximum heart rate achieved), blood pressure (mm Hg), serum cholesterol in (mg/dl), and electrocardiographic result. In real-world, data is not always complete and in the case of the medical data, it is always true. To remove the number of inconsistencies which are associated with data we use Data preprocessing. Table 1 describes the input attributes used for prediction.

1	Age in year
2	Sex (value 1: Male; value 0 :Female)
3	Chest pain type (value 1:typical type 1 angina; value 2 : typical type angina; value 3: non-angina pain; value 4 : asymptomatic)
4	Resting blood pressure (mm Hg on admission to the hospital)
5	Serum Cholesterol in mg/dl
6	Fasting blood sugar (value 1: > 120 mg/dl; value 0 :< 120 mg/dl)
7	Resting Electrocardiographic results (values 0:normal;value1: 1 having ST-T wave abnormality; value 2:showing probable or definite left ventricular hypertrophy)
8	Maximum heart rate achieved
9	Exercise induced angina (value 1:yes; value 0 : no)
10	Old peak = ST depression induced by exercise relative to rest
11	The slope of the peak exercise ST segment (value 1: unsloping; value 2 : flat; value 3 :down sloping)
12	Number of major vessels colored by fluoroscopy (value 0-3)
13	Thal(value3 = normal; value 6 = fixed defect; value 7 = reversible defect)

Complete attribution documentation:

Complete attribute documentation:
1 id: patient identification number
2 ccf: social security number (I replaced this with a dummy value of 0)
3 age: age in years
4 sex: sex (1 = male; 0 = female)
5 painloc: chest pain location (1 = substernal; 0 = otherwise)
6 painexer (1 = provoked by exertion; 0 = otherwise)
7 relrest (1 = relieved after rest; 0 = otherwise)
8 pncaden (sum of 5, 6, and 7)
9 cp: chest pain type

-- Value 1: typical angina
-- Value 2: atypical angina
-- Value 3: non-anginal pain
-- Value 4: asymptomatic
10 trestbps: resting blood pressure (in mm Hg on admission to the hospital)
11 htn
12 chol: serum cholestoral in mg/dl
13 smoke: I believe this is 1 = yes; 0 = no (is or is not a smoker)
14 cigs (cigarettes per day)
15 years (number of years as a smoker)
16 fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
17 dm (1 = history of diabetes; 0 = no such history)
18 famhist: family history of coronary artery disease (1 = yes; 0 = no)
19 restecg: resting electrocardiographic results
-- Value 0: normal
-- Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
-- Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
20 ekgmo (month of exercise ECG reading)
21 ekgday(day of exercise ECG reading)
22 ekgyr (year of exercise ECG reading)
23 dig (digitalis used furing exercise ECG: 1 = yes; 0 = no)
24 prop (Beta blocker used during exercise ECG: 1 = yes; 0 = no)
25 nitr (nitrates used during exercise ECG: 1 = yes; 0 = no)
26 pro (calcium channel blocker used during exercise ECG: 1 = yes; 0 = no)
27 diuretic (diuretic used used during exercise ECG: 1 = yes; 0 = no)
28 proto: exercise protocol
1 = Bruce
2 = Kottus
3 = McHenry
4 = fast Balke
5 = Balke
6 = Noughton
7 = bike 150 kpa min/min (Not sure if "kpa min/min" is what was written!)
8 = bike 125 kpa min/min
9 = bike 100 kpa min/min
10 = bike 75 kpa min/min
11 = bike 50 kpa min/min
12 = arm ergometer
29 thaldur: duration of exercise test in minutes
30 thaltime: time when ST measure depression was noted
31 met: mets achieved
32 thalach: maximum heart rate achieved
33 thalrest: resting heart rate
34 tpeakbps: peak exercise blood pressure (first of 2 parts)
35 tpeakbpd: peak exercise blood pressure (second of 2 parts)
36 dummy

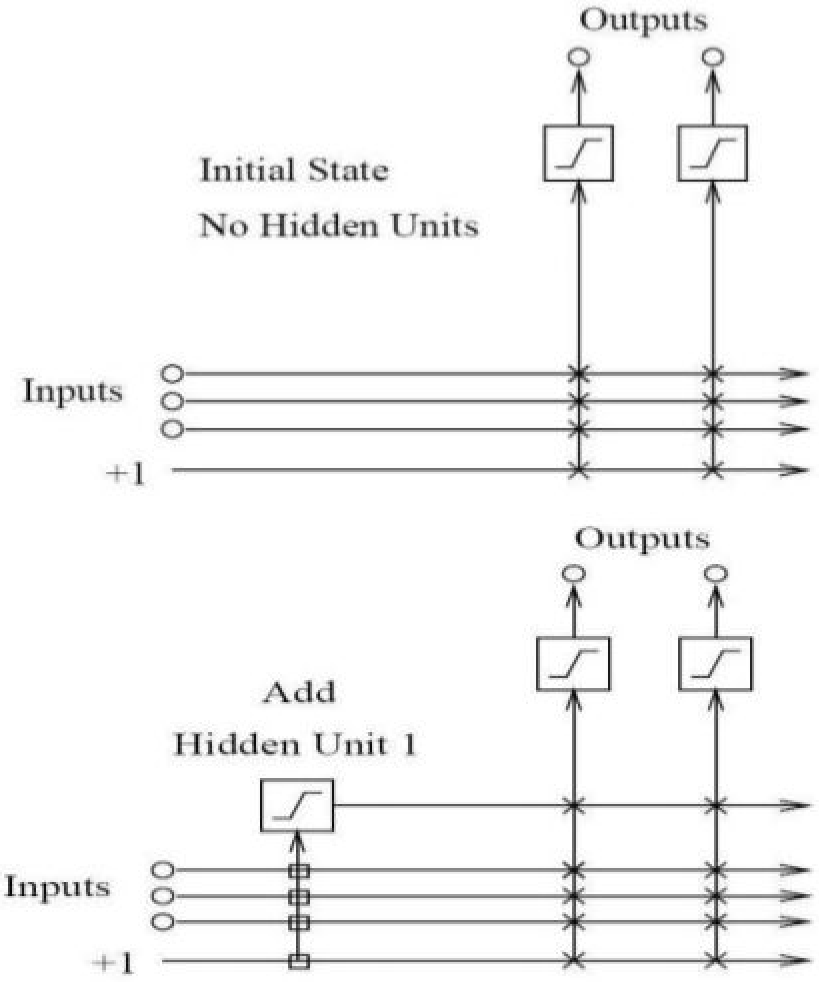
37 trestbpd: resting blood pressure
38 exang: exercise induced angina (1 = yes; 0 = no)
39 xhypo: (1 = yes; 0 = no)
40 oldpeak = ST depression induced by exercise relative to rest
41 slope: the slope of the peak exercise ST segment
-- Value 1: upsloping
-- Value 2: flat
-- Value 3: downsloping
42 rldv5: height at rest
43 rldv5e: height at peak exercise
44 ca: number of major vessels (0-3) colored by flourosopy
45 restckm: irrelevant
46 exerckm: irrelevant
47 restef: rest raidonuclid (sp?) ejection fraction
48 restwm: rest wall (sp?) motion abnormality
0 = none
1 = mild or moderate
2 = moderate or severe
3 = akinesis or dyskmem (sp?)
49 exeref: exercise radinalid (sp?) ejection fraction
50 exerwm: exercise wall (sp?) motion
51 thal: 3 = normal; 6 = fixed defect; 7 = reversable defect
52 thalsev: not used
53 thalpul: not used
54 earlobe: not used
55 cmo: month of cardiac cath (sp?) (perhaps "call")
56 cday: day of cardiac cath (sp?)
57 cyr: year of cardiac cath (sp?)
58 num: diagnosis of heart disease (angiographic disease status)
-- Value 0: < 50% diameter narrowing
-- Value 1: > 50% diameter narrowing
(in any major vessel: attributes 59 through 68 are vessels)
59 lmt
60 ladprox
61 laddist
62 diag
63 cxmain
64 ramus
65 om1
66 om2
67 rcaprox
68 rcadist
69 lvx1: not used
70 lvx2: not used
71 lvx3: not used
72 lvx4: not used
73 lvf: not used

74 cathef: not used
75 junk: not used
76 name: last name of patient (I replaced this with the dummy string "name")

CNN training algorithm :

Step 1: Initialize the input and output units based on the problem defined. The input and output neurons are fully connected.
Step 2: Train the network with input and output neurons until the residual error no longer decreases.
Step 3: Select a temporary unit (Candidate unit) connected with the input unit and find the residual error.
Step 4: Train this network unit S (equ. 1) no longer improves.
Step 5: Connect the temporary unit with the output unit and freeze its weights.
Step 6: Train the Input, output and the hidden unit until the residual error is minimized.
Step 7: Repeat step 2 to step 6 until the net error falls below a given value.

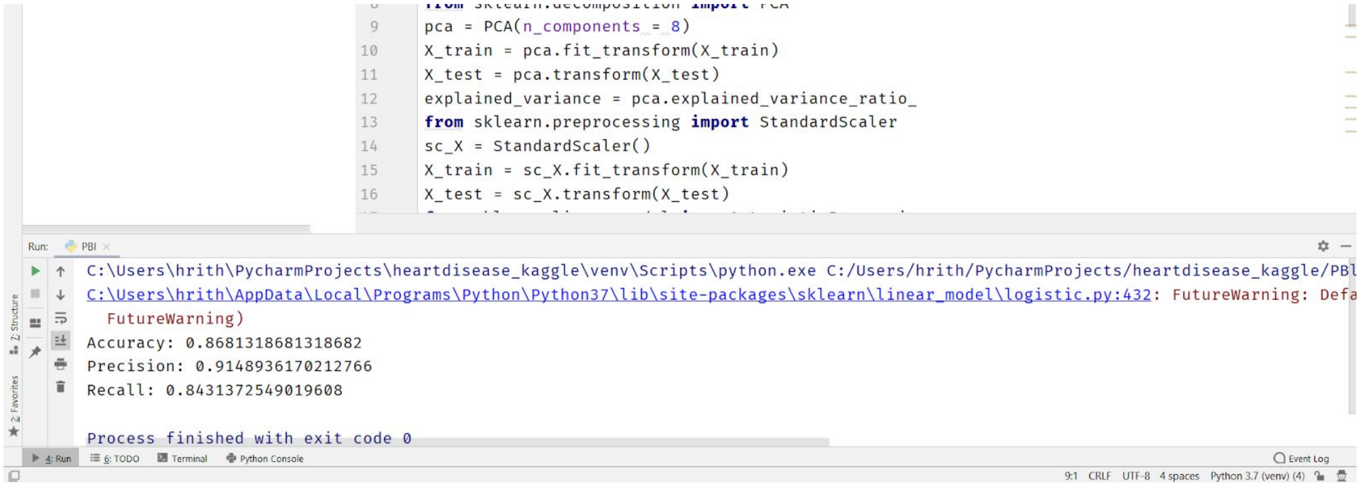
The initial state of the CNN Architecture and the architecture after adding one hidden unit is shown here:



Source code:

```
import pandas as pd
import numpy as np
dataset = pd.read_csv("heart.csv")
X = dataset.drop('target',axis=1)
y = dataset.iloc[:,-1].values
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3,random_state=109)
from sklearn.decomposition import PCA
pca = PCA(n_components = 8)
```

```
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))
```



```
from sklearn.decomposition import PCA
pca = PCA(n_components = 8)
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

```
C:\Users\hrith\PycharmProjects\heartdisease_kaggle\venv\Scripts\python.exe C:/Users/hrith/PycharmProjects/heartdisease_kaggle/PBl
C:\Users\hrith\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: Defa
  FutureWarning)
Accuracy: 0.8681318681318682
Precision: 0.9148936170212766
Recall: 0.8431372549019608

Process finished with exit code 0
```

Results :

Accuracy : 86.80%
Precision : 91.50%
Recall : 84.30%