Dense Neural Network

A deep neural network (DNN) is an ANN (Artificial Neural Network) with multiple hidden layers between the input and output layers. Similar to shallow ANNs, DNNs can model complex non-linear relationships.

The main purpose of a neural network is to receive a set of inputs, perform progressively complex calculations on them, and give output to solve real world problems like classification. We restrict ourselves to feed forward neural networks.

We have an input, an output, and a flow of sequential data in a deep network.

In deep learning, the number of hidden layers, mostly non-linear, can be large; say about 1000 layers.

DL models produce much better results than normal ML networks.

We mostly use the gradient descent method for optimising the network and minimising the loss function.

Gradient Descent:

Gradient descent is an optimization algorithm which is commonly-used to train machine learning models and neural networks. Training data helps these models learn over time, and the cost function within gradient descent specifically acts as a barometer, gauging its accuracy with each iteration of parameter updates. Until the function is close to or equal to zero, the model will continue to adjust its parameters to yield the smallest possible error.

The Loss Function:

The cost (or loss) function measures the difference, or error, between actual y and predicted y at its current position. This improves the machine learning model's efficacy by providing feedback to the model so that it can adjust the parameters to minimise the error and find the local or global minimum.

DNN models are, in general, very extensive, especially for larger types of real datasets. For their higher resource consumption they are run on GPU rather than CPUs.

The visualisation of DNN:

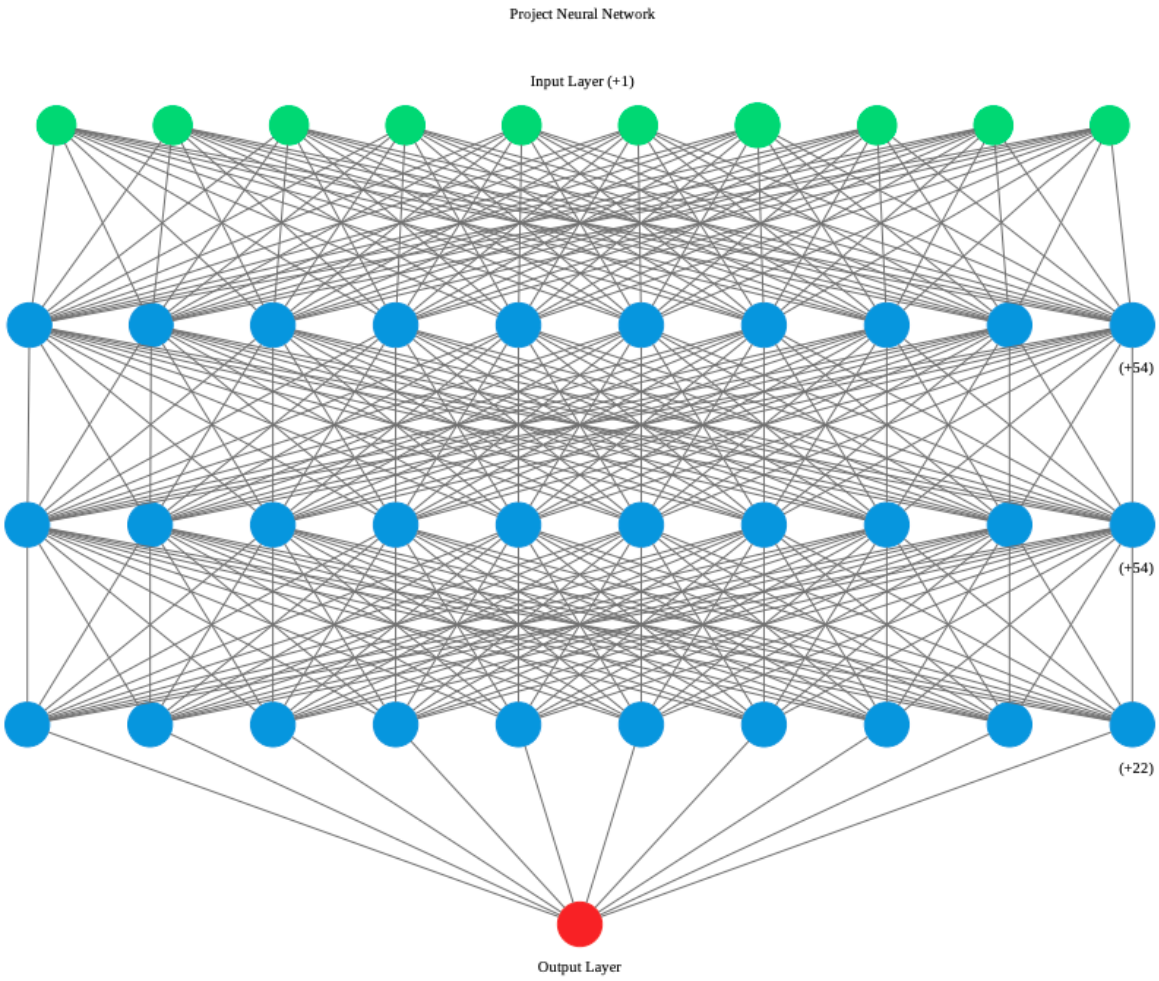


Figure: The fully connected DNN

The Sequential model of DNN provided by Kera's API is simple to configure and tune to get the best results.

The following parameters were tested with different values that impact the accuracy of the DNN model.

The parameters can be added at different levels.

1. While creating an object of the model from Sequential() class.
number of layers = hidden layers added to the DNN

2. While adding the Dense hidden layer.
units = number of nodes or units added to the layer
activation = "relu" or "sigmoid" or "softmax"

3. While doing compilation of the model.
loss = 'binary_crossentropy' or 'mse'
optimizer = 'Adam',
metrics = ['accuracy']

4. While fitting the model.
epochs = 500 (how many times the full dataset is used for training the model in the runs)
batch size = 50 (size of the subset that is used while building

About the dataset:

Source: Kaggle

Size: The dataset is simply a CSV file containing 918 patients' labelled data. There are 11 different risk factors or features used before diagnosing cardiovascular disease.

This dataset has been created by combining the following 5 different datasets over 11 common features.

- Cleveland: 303 observations
- Hungarian: 294 observations

- Switzerland: 123 observations
- Long Beach VA: 200 observations
- Stalog (Heart) Data Set: 270 observations

Total: 1190 observations

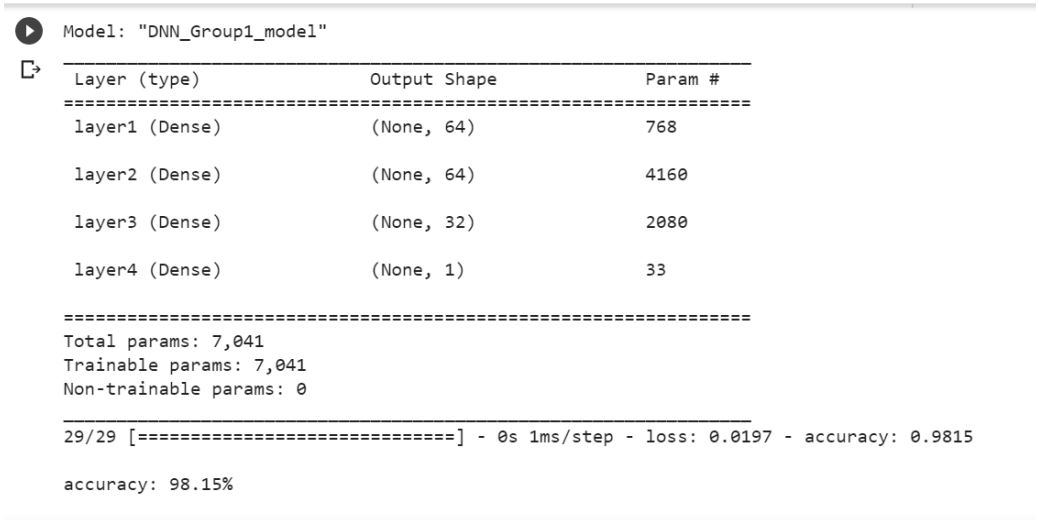
Duplicated: 272 observations (removed by Kaggle)

Final dataset: 918 observations

Features/ Risk factors (as given in Kaggle Metadata):

1. Age: age of the patient [years]
2. Sex: sex of the patient [M: Male, F: Female]
3. ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
4. RestingBP: resting blood pressure [mm Hg]
5. Cholesterol: serum cholesterol [mm/dl]
6. FastingBS: fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]
7. RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
8. MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]
9. ExerciseAngina: exercise-induced angina [Y: Yes, N: No]
10. Oldpeak: oldpeak = ST [Numeric value measured in depression]
11. ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
12. HeartDisease: output class [1: heart disease, 0: Normal]

Dense Neural Network Results:



```
Model: "DNN_Group1_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 layer1 (Dense)              (None, 64)                768

 layer2 (Dense)              (None, 64)                4160

 layer3 (Dense)              (None, 32)                2080

 layer4 (Dense)              (None, 1)                 33

=================================================================
Total params: 7,041
Trainable params: 7,041
Non-trainable params: 0
_________________________________________________________________
29/29 [==============================] - 0s 1ms/step - loss: 0.0197 - accuracy: 0.9815

accuracy: 98.15%
```

Note: When the code is run again the accuracies may change although closer. For this reason the screenshots of accuracies observed are attached in this document.

Code for the project along with the dataset is on github. Please refer to the link below.
https://github.com/saloniwalimbe/ML