Deep Neural Network with Keras for Regression

- In this notebook we will be using DNN on one of Kaggle's dataset Predict Hourly Wage (https://www.kaggle.com/c/predict-hourly-wage/).
- After getting the prediction of our neural net we will submit the prediction to kaggle competition and check our score.

NN basics

- Deep Learning is an increasingly popular part of Machine Learning.
- A neural network (NN) with more than one hidden layer is called **Deep Neural Network (DNN)**.
- A NN takes input, process it in hidden layers with weights and spits out a output/prediction.
- With NN we dont have to worry about feature selection.
- NN adjusts the weights (during forward and back propagation) to meet the target value and in turn provide a prediction.

- In this notebook we will build a DNN for regression problem.
- We will predict hourly wage for an employee.
- Dataset can be downloaded from here (https://www.kaggle.com/c/predict-hourly-wage/data)

Content:

1. Load data
2. Data pre-processing
3. Prepare feature and target variable
4. Build DNN model
5. Compile the model
6. Train the model
7. Predict on new data
8. Submit the prediction to Kaggle for evaluation
9. Additional: Increasing model capacity with additional layers and nodes

1. Load data using Pandas

In [41]:

```
# Import pandas and numpy for file handling and numeric algebra respectively
import pandas as pd
import numpy as np
```

In [42]:

```
# Load the data
data=pd.read_csv('Income_training.csv')
```

In [43]:

```
# Check numerical statictics of data using pandas 'info' method
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3197 entries, 0 to 3196
Data columns (total 4 columns):
compositeHourlyWages    3197 non-null float64
age                     3197 non-null int64
yearsEducation          3197 non-null int64
sex1M0F                 3197 non-null int64
dtypes: float64(1), int64(3)
memory usage: 100.0 KB
```

In [44]:

```
# Check 1st two rows using pandas 'head' method
data.head(2)
```

Out[44]:

	compositeHourlyWages	age	yearsEducation	sex1M0F
0	21.38	58	10	1
1	25.15	42	16	1

2. Data pre-processing

- This dataset is already clean [i.e. no missing value, no categorical values], hence we don't have to perform any pre-processing.
- However we can standardize the data as wage and age/year are in different units. [try it]

3. Prepare feature and target variable

In [45]:

```
# Prepare feature variable i.e. all features except target i.e. compositeHourlyWages
X_train=data.drop(columns=['compositeHourlyWages'])
# Prepare target variable i.e. compositeHourlyWages
y_train=data['compositeHourlyWages']
```

In [46]:

```
# Check feature data
X_train.head(2)
```

Out[46]:

	age	yearsEducation	sex1M0F
0	58	10	1
1	42	16	1

In [47]:

```
# Check target data
y_train.head()
```

Out[47]:

```
0    21.38
1    25.15
2     8.57
3    12.07
4    10.97
Name: compositeHourlyWages, dtype: float64
```

4. Build DNN model

- For basic info on keras model kindly refer KERAS API notebook [here (https://github.com/jay-pm/Deep-Learning/tree/master/The%20Keras%20API)](https://github.com/jay-pm/Deep-Learning/tree/master/The%20Keras%20API)

In [48]:

```
# Import Sequential and Dense from Keras

from keras.models import Sequential
from keras.layers import Dense
```

In [49]:

```
# create the model
model=Sequential()
# get the number of columns of training data
n_cols=X_train.shape[1] # X_train.shape >> (3197, 3)
```

- We will use the ‘add()’ function to add layers to our model. We will add two hidden layers and an output layer.
- Dense is a standard layer. In a dense layer, all nodes in the previous layer connect to the nodes in the current layer.
- We have kept 6 nodes in each of our hidden layer.
- number of nodes in hidden layer can be hundreds or thousands.
- increasing the number of nodes in a layer increases model capacity but at the cost of computational costs.
- activation function adds non-linearity to data.
- we will use RELU activation function which is proven to work well with NN.
- The input shape specifies the number of rows and columns in the input. The number of columns in our input is stored in ‘n_cols’. There is nothing after the comma which indicates that there can be any amount of rows.
- The last layer is the output layer. It only has one node, which is for our prediction.

In [50]:

```
# 1st hidden layer
model.add(Dense(6,activation='relu', input_shape=(n_cols,)))

# 2nd hidden later
model.add(Dense(6, activation='relu'))

# output layer
model.add(Dense(1))
```

5. Compile the model

- Compiling the model takes two parameters: optimizer and loss.
- The optimizer controls the learning rate.
- 'adam' is a good default optimizer to use, and most of the time works well.
- The adam optimizer adjusts the learning rate throughout training.
- The learning rate determines how fast the optimal weights for the model are calculated. A smaller learning rate may lead to more accurate weights (up to a certain point), but the time it takes to compute the weights will be longer.
- Loss function depends on the problem at hand. Mean squared error is a common loss function and will optimize for predicting the mean, as is done in least squares regression. For classification use binary_crossentropy [2 class] or categorical_crossentropy [multiclass]

In [51]:

```
# compile the model
model.compile(optimizer='adam', loss='mean_squared_error')
```

6. Train the model

- For training the model, we will use the ‘fit()’ function on our model with the following five parameters: training data (X), target data (y), validation split, the number of epochs and callbacks.
- The validation split will randomly split the data into use for training and testing. We will set the validation split at 0.2, which means that 20% of the training data we provide in the model will be set aside for testing model performance.
- The number of epochs is the number of times the model will cycle through the data. The more epochs we run, the more the model will improve, up to a certain point. After that point, the model will stop improving during each epoch. In addition, the more epochs, the longer the model will take to run. To monitor this, we will use ‘early stopping’.
- Early stopping will stop the model from training before the number of epochs is reached if the model stops improving. We will set our early stopping monitor to 3. This means that after 3 epochs in a row in which the model doesn’t improve, training will stop.

In [52]:

```
from keras.callbacks import EarlyStopping

# train the model
model.fit(X_train,y_train,validation_split=.2,epochs=40,callbacks=[EarlyStopping(patien
ce=3)])
```

```
Train on 2557 samples, validate on 640 samples
Epoch 1/40
2557/2557 [==============================] - 1s 289us/step - loss: 555.342
3 - val_loss: 344.5116
Epoch 2/40
2557/2557 [==============================] - 0s 80us/step - loss: 176.0397
- val_loss: 73.1601
Epoch 3/40
2557/2557 [==============================] - 0s 69us/step - loss: 55.5342
- val_loss: 43.9945
Epoch 4/40
2557/2557 [==============================] - 0s 65us/step - loss: 47.7153
- val_loss: 43.5987
Epoch 5/40
2557/2557 [==============================] - 0s 69us/step - loss: 47.4134
- val_loss: 43.3194
Epoch 6/40
2557/2557 [==============================] - 0s 70us/step - loss: 47.2314
- val_loss: 43.1674
Epoch 7/40
2557/2557 [==============================] - 0s 65us/step - loss: 47.0663
- val_loss: 43.0678
Epoch 8/40
2557/2557 [==============================] - 0s 65us/step - loss: 46.9442
- val_loss: 42.9762
Epoch 9/40
2557/2557 [==============================] - 0s 77us/step - loss: 46.9709
- val_loss: 42.9202
Epoch 10/40
2557/2557 [==============================] - 0s 66us/step - loss: 46.8610
- val_loss: 42.8603
Epoch 11/40
2557/2557 [==============================] - 0s 64us/step - loss: 46.7992
- val_loss: 42.7997
Epoch 12/40
2557/2557 [==============================] - 0s 69us/step - loss: 46.7369
- val_loss: 42.7392
Epoch 13/40
2557/2557 [==============================] - 0s 63us/step - loss: 46.6458
- val_loss: 42.8804
Epoch 14/40
2557/2557 [==============================] - 0s 76us/step - loss: 46.6715
- val_loss: 42.7034
Epoch 15/40
2557/2557 [==============================] - 0s 66us/step - loss: 46.6371
- val_loss: 42.7083
Epoch 16/40
2557/2557 [==============================] - 0s 81us/step - loss: 46.6728
- val_loss: 42.5380
Epoch 17/40
2557/2557 [==============================] - 0s 73us/step - loss: 46.4713
- val_loss: 42.5304
Epoch 18/40
2557/2557 [==============================] - 0s 65us/step - loss: 46.4186
- val_loss: 42.4496
Epoch 19/40
2557/2557 [==============================] - 0s 66us/step - loss: 46.4102
- val_loss: 42.3821
Epoch 20/40
2557/2557 [==============================] - 0s 124us/step - loss: 46.3003
- val_loss: 42.4336
```

```
Epoch 21/40
2557/2557 [==============================] - 0s 110us/step - loss: 46.2956
- val_loss: 42.2729
Epoch 22/40
2557/2557 [==============================] - 0s 74us/step - loss: 46.2587
- val_loss: 42.2828
Epoch 23/40
2557/2557 [==============================] - 0s 66us/step - loss: 46.2077
- val_loss: 42.2463
Epoch 24/40
2557/2557 [==============================] - 0s 65us/step - loss: 46.1183
- val_loss: 42.0924
Epoch 25/40
2557/2557 [==============================] - 0s 70us/step - loss: 46.1329
- val_loss: 42.0814
Epoch 26/40
2557/2557 [==============================] - 0s 69us/step - loss: 46.1367
- val_loss: 42.0413
Epoch 27/40
2557/2557 [==============================] - 0s 66us/step - loss: 45.9999
- val_loss: 41.9407
Epoch 28/40
2557/2557 [==============================] - 0s 72us/step - loss: 45.9199
- val_loss: 41.8529
Epoch 29/40
2557/2557 [==============================] - 0s 65us/step - loss: 45.9618
- val_loss: 41.8465
Epoch 30/40
2557/2557 [==============================] - 0s 76us/step - loss: 45.8014
- val_loss: 41.7530
Epoch 31/40
2557/2557 [==============================] - 0s 67us/step - loss: 45.9501
- val_loss: 41.7626
Epoch 32/40
2557/2557 [==============================] - 0s 68us/step - loss: 45.6753
- val_loss: 41.7222
Epoch 33/40
2557/2557 [==============================] - 0s 70us/step - loss: 45.6605
- val_loss: 41.6070
Epoch 34/40
2557/2557 [==============================] - 0s 70us/step - loss: 45.5739
- val_loss: 41.8983
Epoch 35/40
2557/2557 [==============================] - 0s 65us/step - loss: 45.5856
- val_loss: 41.7266
Epoch 36/40
2557/2557 [==============================] - 0s 69us/step - loss: 45.5822
- val_loss: 41.3999
Epoch 37/40
2557/2557 [==============================] - 0s 79us/step - loss: 45.4459
- val_loss: 41.3713
Epoch 38/40
2557/2557 [==============================] - 0s 64us/step - loss: 45.4687
- val_loss: 41.5781
Epoch 39/40
2557/2557 [==============================] - 0s 70us/step - loss: 45.4191
- val_loss: 41.2016
Epoch 40/40
2557/2557 [==============================] - 0s 66us/step - loss: 45.2640
- val_loss: 41.1305
```

Out[52]:

```
<keras.callbacks.History at 0xf14d908>
```

7. Predict on new data

- we will use the ‘predict()’ function, passing in our new data. The output would be ‘wage per hour’ predictions.

In [53]:

```
# load the new data
data_new=pd.read_csv('Income_testing.csv') # this data dont have target i.e. compositeH
ourlyWages which we have to predict
```

In [54]:

```
# check the test data
data_new.head()
```

Out[54]:

	ID	age	yearsEducation	sex1M0F
0	1	36	20	0
1	2	38	17	0
2	3	24	10	0
3	4	39	12	1
4	5	50	12	0

In [55]:

```
data_test=data_new.drop(columns='ID') # drop ID column as it is not in our training dat
a
data_test.head(2)
```

Out[55]:

	age	yearsEducation	sex1M0F
0	36	20	0
1	38	17	0

In [56]:

```
# predict on new data with already created model
y_predict=model.predict(data_test)
```

In [57]:

```
y_predict
```

Out[57]:

```
array([[ 19.2833252 ],
       [ 17.7055378 ],
       [ 10.51393127],
       [ 15.98699665],
       [ 16.70557022],
       [ 16.53741455],
       [ 16.95337296],
       [ 17.46391296],
       [ 11.87750721],
       [ 17.45021057],
       [  9.72189045],
       [ 11.71563816],
       [ 13.06551266],
       [ 13.50887966],
       [ 14.07300091],
       [ 20.64690018],
       [ 18.8474865 ],
       [ 15.28630447],
       [ 13.45529842],
       [ 19.34431839],
       [ 15.46275043],
       [ 14.37625408],
       [ 15.4963398 ],
       [ 20.54603195],
       [ 14.82516575],
       [ 19.59964752],
       [ 18.15631676],
       [ 13.06551266],
       [  9.00960541],
       [ 14.33450603],
       [ 12.17888927],
       [ 18.31227875],
       [ 17.51749039],
       [ 13.7305069 ],
       [ 11.93108749],
       [ 15.04679012],
       [ 16.89360809],
       [ 14.7441721 ],
       [ 18.24348259],
       [ 16.71175194],
       [ 15.50386333],
       [ 18.24348259],
       [ 18.20236969],
       [ 11.25238991],
       [ 17.92098618],
       [ 15.8251276 ],
       [ 16.56358528],
       [ 19.43148804],
       [ 17.55108261],
       [ 12.87747288],
       [ 12.87747288],
       [ 19.25715446],
       [ 13.97212887],
       [ 10.35947704],
       [ 11.06434917],
       [ 14.07300091],
       [ 22.7290535 ],
       [ 17.0542469 ],
       [ 14.8724556 ],
```

```
       [ 16.8799057 ],
       [ 21.76144218],
       [ 16.73915863],
       [ 15.72425556],
       [ 14.44908428],
       [ 15.38928795],
       [ 12.0381422 ],
       [ 19.25715446],
       [ 16.37678146],
       [ 14.54995346],
       [ 13.14650059],
       [ 13.69692039],
       [ 16.37678146],
       [ 14.2946291 ],
       [ 11.71563816],
       [ 16.4502449 ],
       [ 22.99055862],
       [ 17.30327988],
       [ 12.38063526],
       [ 18.0218544 ],
       [ 10.54751873],
       [ 23.65555573],
       [ 17.49132156],
       [ 14.48266888],
       [ 16.75285912],
       [ 15.46275043],
       [ 13.95842457],
       [ 18.26965904],
       [ 16.16133118],
       [  9.86264038],
       [ 15.52374554],
       [ 13.18008804],
       [ 17.50502777],
       [ 19.24345016],
       [ 14.49637413],
       [ 19.42530823],
       [ 16.10033607],
       [ 11.44043064],
       [ 14.85875034],
       [ 12.56867504],
       [ 21.82872772],
       [ 19.05540848],
       [ 15.49015713],
       [  4.63861036],
       [ 16.66445923],
       [ 11.01076794],
       [ 11.3868494 ],
       [ 20.37169075],
       [ 11.3395586 ],
       [ 16.38692856],
       [ 15.71178341],
       [ 16.06046104],
       [ 21.94948387],
       [ 17.54490471],
       [ 16.41665649],
       [ 11.74304867],
       [ 18.96823883],
       [ 14.2809267 ],
       [ 13.50887966],
       [ 13.66951084],
       [ 16.34936905],
```

```
       [ 13.49517536],
       [ 15.56362247],
       [ 12.32705307],
       [ 16.79891968],
       [ 18.21607399],
       [ 15.28841686],
       [ 17.30327988],
       [ 14.12029171],
       [ 17.06794739],
       [ 17.47761536],
       [ 22.71535301],
       [ 18.20236969],
       [ 10.68826771],
       [ 14.58353901],
       [ 16.99448204],
       [ 15.20124626],
       [ 15.01320457],
       [ 15.32199955],
       [ 10.35947704],
       [ 12.00455379],
       [ 17.08165359],
       [ 17.67936134],
       [ 11.90367985],
       [ 19.70669937],
       [ 19.8075695 ],
       [ 20.28452301],
       [ 19.12269211],
       [ 16.02326584],
       [ 14.2946291 ],
       [ 16.80644226],
       [ 21.35177231],
       [ 20.55973244],
       [ 18.21607399],
       [ 17.61836815],
       [ 16.4502449 ],
       [ 16.01317024],
       [ 19.12269211],
       [ 15.92600155],
       [ 15.18754005],
       [ 18.06914902],
       [ 20.83494186],
       [ 16.59099579],
       [ 14.28845215],
       [ 18.2571888 ],
       [ 11.19880867],
       [ 24.13250732],
       [ 16.24850082],
       [ 17.24228668],
       [ 16.16133118],
       [ 13.88495731],
       [ 13.71062374],
       [ 14.93221569],
       [ 16.51753235],
       [ 16.47641754],
       [ 17.89357567],
       [ 14.65700626],
       [ 15.97329426],
       [ 15.61091423],
       [ 18.77401924],
       [ 13.89866257],
       [ 21.96318626],
```

```
       [ 18.90724373],
       [ 14.42167377],
       [ 16.4639473 ],
       [ 17.69306755],
       [ 20.4725666 ],
       [ 13.77038479],
       [ 15.28841686],
       [ 10.41305828],
       [ 14.44908428],
       [ 15.3755827 ],
       [ 12.94475651],
       [ 16.21401978],
       [ 20.57344055],
       [ 17.78023148],
       [ 15.05993271],
       [ 17.93078041],
       [ 17.63825035],
       [ 19.14257812],
       [ 13.50887966],
       [ 15.4228754 ],
       [ 14.81146145],
       [ 15.66449642],
       [ 13.67375183],
       [ 18.92094803],
       [ 16.4365387 ],
       [ 14.2946291 ],
       [ 17.22858047],
       [ 19.18368912],
       [ 22.51360893],
       [ 14.7441721 ],
       [ 13.03192616],
       [ 16.91349602],
       [ 16.8799057 ],
       [ 15.54991817],
       [ 14.3817997 ],
       [ 14.03909588],
       [ 18.59215927],
       [ 17.73294449],
       [ 13.39430237],
       [ 16.02326584],
       [ 18.1961956 ],
       [ 16.73915863],
       [ 16.10033607],
       [ 15.3357048 ],
       [ 14.72429371],
       [ 12.10542488],
       [ 18.26965904],
       [ 15.07296467],
       [ 16.8799057 ],
       [ 16.18873978],
       [ 15.04679012],
       [ 15.47645473],
       [ 15.07296467],
       [ 18.75278854],
       [ 10.54751873],
       [ 16.75285912],
       [ 21.09026718],
       [ 16.84003067],
       [ 19.68681335],
       [ 13.7305069 ],
       [ 17.74002457],
```

```
       [ 16.8799057 ],
       [ 15.56362247],
       [ 14.30833149],
       [ 16.32948875],
       [ 14.48266888],
       [ 12.40804195],
       [ 14.65700626],
       [ 20.11018753],
       [ 18.1824894 ],
       [ 16.89979172],
       [ 15.52374554],
       [ 14.91233158],
       [ 20.16994858],
       [ 12.99204731],
       [ 21.68797684],
       [ 16.22861862],
       [ 18.41781807],
       [ 12.80400944],
       [ 15.32199955],
       [  9.84893513],
       [ 17.45773315],
       [ 23.60826302],
       [ 16.32948875],
       [ 16.51753235],
       [ 15.07296467],
       [ 16.4502449 ],
       [ 13.50887966],
       [ 14.2946291 ],
       [ 11.91738415],
       [ 12.97834492],
       [ 19.45889664],
       [ 18.02803612],
       [ 13.52258301],
       [ 11.71563816],
       [ 21.29201508],
       [ 25.14617538],
       [ 18.83377838],
       [ 13.95842457],
       [ 18.34435654],
       [ 13.71680355],
       [ 15.61091423],
       [ 18.05544281],
       [ 17.63207054],
       [ 22.62818146],
       [ 20.15006828],
       [ 15.83251381],
       [ 18.39041138],
       [ 20.18365288],
       [ 18.67932701],
       [ 15.64176655],
       [ 18.77401924],
       [  9.84893513],
       [ 18.2297821 ],
       [ 17.60466576],
       [ 14.99949932],
       [ 14.8861599 ],
       [ 19.96202278],
       [ 13.45529842],
       [ 11.06434917],
       [ 16.17503738],
       [ 17.83999634],
```

```
       [ 12.68325138],
       [ 18.53239822],
       [ 20.4862709 ],
       [ 14.93221569],
       [ 12.60226154],
       [ 15.13395977],
       [ 20.11018753],
       [ 19.58594131],
       [ 20.35181046],
       [ 14.21624374],
       [ 10.50022793],
       [ 13.16638374],
       [ 16.61087799],
       [ 14.18985653],
       [ 17.1015358 ],
       [ 12.22618198],
       [ 16.34936905],
       [ 12.75671673],
       [ 16.61087799],
       [ 14.08670521],
       [ 18.44399452],
       [ 20.19735909],
       [ 11.37792492],
       [ 18.12891006],
       [ 15.11407375],
       [ 14.88492203],
       [ 20.67431068],
       [ 15.69808292],
       [ 13.46900082],
       [ 19.54606628],
       [ 17.37674332],
       [ 17.31575012],
       [ 24.77009392],
       [ 14.63712406],
       [ 12.49520969],
       [ 21.5871067 ],
       [ 18.66562271],
       [ 14.09288502],
       [ 19.38703918],
       [ 21.51364136],
       [ 17.24228668],
       [ 17.01436996],
       [ 15.83883381],
       [ 14.2946291 ],
       [ 25.33421707],
       [ 12.22618198],
       [ 12.32705307],
       [ 12.75671673],
       [ 14.2809267 ],
       [ 19.63323402],
       [ 15.10037136],
       [ 14.2473402 ],
       [ 17.75282669],
       [ 10.68826771],
       [ 14.233634  ],
       [ 15.95341015],
       [ 16.4029541 ],
       [ 17.1015358 ],
       [ 13.52876282],
       [ 14.82516575],
       [ 18.14261436],
```

```
       [ 13.52258301],
       [ 18.24348259],
       [ 13.21996784],
       [ 12.0381422 ],
       [ 22.42643738],
       [ 15.91229916],
       [ 18.02803612],
       [ 13.03192616],
       [ 15.56362247],
       [ 12.49520969],
       [ 20.29822922],
       [ 13.71062374],
       [ 15.3357048 ],
       [ 15.63708973],
       [ 17.96827507],
       [ 14.91233158],
       [ 10.87630844],
       [ 15.45101261],
       [ 21.86231613],
       [ 13.07921505],
       [ 15.21494865],
       [ 15.14766407],
       [ 20.66060638],
       [ 12.38063526],
       [ 13.39430237],
       [ 14.7441721 ],
       [ 17.85370064],
       [ 15.17383862],
       [ 17.63825035],
       [ 10.9236002 ],
       [ 17.75282669],
       [ 19.23727036],
       [ 10.68826771],
       [ 13.54246807],
       [ 12.42792606],
       [ 16.72545433],
       [ 14.57072258],
       [ 15.71178341],
       [ 14.18757725],
       [ 24.77009392],
       [ 14.67071056],
       [ 12.65584373],
       [ 12.68943119],
       [ 17.36304092],
       [ 15.61091423],
       [ 20.08277893],
       [ 11.57489109],
       [ 15.30829906],
       [ 13.50887966],
       [ 15.36187935],
       [ 20.4588604 ],
       [ 17.47761536],
       [ 13.23367023],
       [ 14.46896839],
       [ 13.20626163],
       [ 13.79779148],
       [ 14.92603493],
       [ 18.09532166],
       [ 16.24232101],
       [ 15.47645473],
       [ 15.51004314],
```

```
       [ 21.41276932],
       [ 10.18514156],
       [ 15.22112942],
       [ 17.44402695],
       [ 11.85009956],
       [ 18.64574051],
       [ 17.36304092],
       [ 12.38063526],
       [ 18.37052917],
       [ 18.50499153],
       [ 17.21611404],
       [ 14.79775715],
       [ 14.05929661],
       [ 18.76649475],
       [  9.19764423],
       [ 12.75671673],
       [ 15.61091423],
       [ 17.09535599],
       [ 13.04099369],
       [ 18.59215927],
       [ 14.8724556 ],
       [ 22.25209808],
       [ 21.39906311],
       [ 21.48623276],
       [ 18.77401924],
       [ 13.40800762],
       [ 18.50499153],
       [ 15.79895496],
       [ 16.73915863],
       [ 17.14141083],
       [ 16.79273605],
       [ 18.82130814],
       [ 13.7305069 ],
       [ 11.65588093],
       [ 15.01320457],
       [ 17.21611404],
       [ 14.89862919],
       [ 17.67936134],
       [ 17.00065994],
       [ 15.38928795],
       [ 12.65584373],
       [ 12.79030228],
       [ 12.15271759],
       [ 18.31694984],
       [ 14.44908428],
       [ 10.87630844],
       [ 15.54991817],
       [ 15.17383862],
       [ 11.54130268],
       [ 20.86235046],
       [ 15.23483181],
       [ 17.01436996],
       [ 15.10037136],
       [ 14.26104259],
       [ 14.68441296],
       [ 14.48266888],
       [ 16.01317024],
       [ 16.02326584],
       [ 15.83883381],
       [ 14.7441721 ],
       [ 13.49517536],
```

```
       [ 13.33454227],
       [ 16.51753235],
       [ 13.44159603],
       [ 13.32083607],
       [ 20.19735909],
       [ 12.79030228],
       [ 20.63319969],
       [ 19.45889664],
       [ 13.44159603],
       [ 13.74421024],
       [ 13.91854858],
       [ 17.28957939],
       [ 11.62847042],
       [ 14.07300091],
       [ 17.80640602],
       [ 18.86736488],
       [ 12.87747288],
       [ 20.58714104],
       [ 14.34821224],
       [ 13.52876282],
       [ 12.75671673],
       [ 15.76536942],
       [ 13.90484142],
       [ 12.94475651],
       [ 13.6097517 ],
       [ 12.84388351],
       [ 16.98077774],
       [ 13.88495731],
       [ 12.93105221],
       [ 18.95453644],
       [ 21.31189537],
       [ 16.53741455],
       [ 19.31690979],
       [ 15.30829906],
       [ 15.32199955],
       [ 12.65584373],
       [ 19.31690979],
       [ 16.14763069],
       [ 20.77518082],
       [ 21.1375618 ],
       [ 16.97702026],
       [ 17.87987137],
       [ 13.76072311],
       [ 10.60109901],
       [ 16.12774467],
       [ 15.44904518],
       [ 14.10658932],
       [ 13.85755157],
       [ 16.26220322],
       [ 13.35442448],
       [ 19.44519043],
       [ 13.71680355],
       [ 20.03548813],
       [ 11.26609516],
       [ 16.32948875],
       [ 22.52730942],
       [ 12.41422176],
       [ 19.54606628],
       [ 19.49877357],
       [ 14.02356339],
       [ 15.04679012],
```

```
       [ 13.88495731],
       [ 15.10037136],
       [ 17.5921936 ],
       [ 17.46391296],
       [ 14.43537807],
       [ 12.55497074],
       [ 21.12385368],
       [ 12.02163029],
       [ 10.2250185 ],
       [ 11.77663517],
       [ 15.27471066],
       [ 17.73294449],
       [ 16.41665649],
       [ 15.06049442],
       [ 13.69692039],
       [ 15.04679012],
       [ 17.74002457],
       [ 15.57732582],
       [ 16.98077774],
       [ 15.20124626],
       [ 15.77154827],
       [ 18.39041138],
       [ 21.68797684],
       [ 15.8861227 ],
       [ 13.93424702],
       [ 21.37918091],
       [ 20.57344055],
       [ 14.44908428],
       [ 18.31694984],
       [ 13.59604836],
       [ 15.3755827 ],
       [ 12.68943119],
       [ 15.23483181],
       [ 14.46278572],
       [ 14.82516575],
       [ 19.00811768],
       [ 18.39793777],
       [ 19.12269211],
       [ 18.93465042],
       [ 16.18425369],
       [ 16.12774467],
       [ 17.60466576],
       [ 15.27471066],
       [ 12.89117527],
       [ 20.18365288],
       [ 15.79895496],
       [ 12.75671673],
       [ 14.82516575],
       [ 16.15515137],
       [ 16.06798553],
       [ 15.36187935],
       [ 19.93585205],
       [ 15.57732582],
       [ 10.35947704],
       [ 16.61087799],
       [  9.84893513],
       [ 16.4029541 ],
       [ 15.13395977],
       [ 16.59552002],
       [ 12.17888927],
       [ 17.47761536],
```

```
       [ 14.16017056],
       [ 12.19259357],
       [ 14.2074604 ],
       [ 11.35326195],
       [ 14.48266888],
       [ 13.49517536],
       [ 20.43897629],
       [ 20.18365288],
       [ 13.6433382 ],
       [ 22.51360893],
       [ 12.09172249],
       [ 11.76293278],
       [ 13.39430237],
       [ 19.70051956],
       [ 15.04679012],
       [ 14.10041046],
       [ 19.92214775],
       [  9.66089535],
       [ 15.51004314],
       [ 18.05544281],
       [ 19.14257812],
       [ 17.06794739],
       [ 22.61447906],
       [ 18.31227875],
       [ 18.78019714],
       [ 12.36693096],
       [ 13.90484142],
       [ 17.06794739],
       [ 16.21401978],
       [ 18.63327408],
       [ 17.24228668],
       [ 16.37678146],
       [ 17.18252182],
       [ 14.42167377],
       [  8.54635525],
       [ 18.12891006],
       [ 15.79895496],
       [ 24.77009392],
       [ 14.34821224],
       [ 17.18252182],
       [ 16.32948875],
       [ 14.82516575],
       [ 14.98579693],
       [ 11.26609516],
       [ 14.22116375],
       [ 21.48623276],
       [ 16.51753235],
       [ 10.41305828],
       [ 17.71923828],
       [ 13.6296339 ],
       [ 16.24232101],
       [ 14.2809267 ],
       [ 12.94475651],
       [ 16.26220322],
       [ 15.58350849],
       [ 13.71680355],
       [ 13.06551266],
       [ 17.46391296],
       [ 14.67071056],
       [ 17.00065994],
       [ 16.67816544],
```

```
       [ 19.15627861],
       [ 19.74780846],
       [ 16.67816544],
       [ 15.67819786],
       [ 15.23483181],
       [ 17.83381271],
       [ 13.52876282],
       [ 11.62847042],
       [ 14.60971355],
       [ 16.18873978],
       [ 11.85009956],
       [ 18.90847778],
       [ 13.78408909],
       [ 16.63828659],
       [ 14.91233158],
       [ 19.61952972],
       [ 17.46391296],
       [ 17.98074341],
       [ 12.84388351],
       [ 15.98699665],
       [ 14.94592094],
       [ 18.37052917],
       [ 14.2747488 ],
       [ 15.4963398 ],
       [  9.48655891],
       [ 18.31076813],
       [ 12.09172249],
       [ 10.87630844],
       [ 15.32199955],
       [ 15.38928795],
       [ 13.71062374],
       [ 21.00310135],
       [ 10.87630844],
       [ 16.07416534],
       [ 17.7055378 ],
       [ 13.66951084],
       [ 13.78408909],
       [ 18.08161736],
       [ 12.22618198],
       [ 12.22618198],
       [ 13.93225193],
       [ 14.67071056],
       [ 14.89862919],
       [ 10.2106638 ],
       [ 11.54130268],
       [ 18.67932701],
       [ 13.91854858],
       [ 12.84388351],
       [ 14.94592094],
       [ 14.63712406],
       [ 16.51753235],
       [ 21.21102524],
       [ 19.8349762 ],
       [ 10.50022793],
       [ 13.06551266],
       [ 14.83886814],
       [ 18.43152618],
       [ 17.73294449],
       [ 17.61836815],
       [ 14.07300091],
       [ 12.65584373],
```

```
       [ 12.38063526],
       [ 12.97834492],
       [ 21.5871067 ],
       [ 16.55111694],
       [ 20.51244354],
       [ 18.2297821 ],
       [ 15.61091423],
       [ 21.17743683],
       [ 16.20121193],
       [ 11.99084759],
       [ 13.90484142],
       [ 11.85009956],
       [ 18.59215927],
       [ 12.41422176],
       [ 18.61956406],
       [ 12.56867504],
       [ 18.40411377],
       [ 20.74777222],
       [ 11.76293278],
       [ 13.52876282],
       [ 19.49877357],
       [ 12.0381422 ],
       [ 24.40771484],
       [ 19.8075695 ],
       [ 14.82516575],
       [ 13.04562664],
       [ 18.80760574],
       [ 14.44908428],
       [  9.00960541],
       [ 10.35947704],
       [ 15.34941006],
       [ 14.53625107],
       [ 16.24850082],
       [ 19.8075695 ],
       [ 19.2833252 ],
       [ 14.48266888],
       [ 14.85875034],
       [ 18.44523239],
       [ 13.59604836],
       [ 14.93221569],
       [ 13.93225193],
       [ 12.56867504],
       [ 13.32083607],
       [ 15.59103203],
       [ 16.14763069],
       [ 19.48507309],
       [ 20.66060638],
       [ 15.78524876],
       [ 13.49517536],
       [ 11.52760029],
       [ 15.59720898],
       [ 19.28456116],
       [ 17.44402695],
       [ 18.28336334],
       [ 15.24853706],
       [ 19.94832039],
       [ 14.63712406],
       [ 14.56983757],
       [ 14.233634  ],
       [ 20.66060638],
       [ 15.98699665],
```

```
       [ 16.63828659],
       [ 11.52760029],
       [ 17.9944458 ],
       [ 12.19259357],
       [ 11.64217567],
       [ 14.83886814],
       [ 16.79273605],
       [ 16.28726959],
       [ 15.8861227 ]], dtype=float32)
```

In [58]:

```
y_predict.shape
```

Out[58]:

```
(800, 1)
```

8. Submit the prediction to Kaggle for evaluation

In [59]:

```
# write prediction data to data_new
data_new['compositeHourlyWages']=y_predict
# preapre submission file only with ID and predicated wages in column 'compositeHourlyW
ages'
data_new[['ID', 'compositeHourlyWages']].to_csv('HourlyWage_DNN.csv', index=False)
```

In [60]:

```
# check the submission file
submission=pd.read_csv('HourlyWage_DNN.csv')
print(submission.head())
print(submission.shape)
```

```
   ID  compositeHourlyWages
0   1             19.283325
1   2             17.705538
2   3             10.513931
3   4             15.986997
4   5             16.705570
(800, 2)
```

- submit the 'HourlyWage_DNN.csv' to kaggle @ https://www.kaggle.com/c/predict-hourly-wage/submit (https://www.kaggle.com/c/predict-hourly-wage/submit)
- Our model score is 6.47462

- Top score for this competition is 6.33222 [https://www.kaggle.com/c/predict-hourly-wage/leaderboard (https://www.kaggle.com/c/predict-hourly-wage/leaderboard)]

In [61]:

```
# save the model
model.save('HourlyWagePred_DNN.h5')
```