

Bigdata and Deep Learning: Using Python Keras Predict Patient Diabetes and Employee's Wages per Hour

C. Rajeev, A. Damodar

Abstract: In this paper we analyze big data analytic & Deep Learning is not supposing as two entire various concept. BigData mean extreme simple larger data into set in that may be analyzes as finding into pattern, trend. The first techniques in that may useful with data analyzed therefore in capable to helping to finding abstract pattern into Big Data is DeepLearning. It is applying into DeepLearning into Big Data, it can be find out nameless & useful pattern in that not possible up to now. This is technique as present into extra active areas into researches in the medical sciences. From increases sizes & complex into medical data's such as X-ray, deeplearning gain into small success to prediction as several diseases such as pneumonia, diabetes. The project is proposed into two deeplearning model used to Keras & too we can be building in a regression models in to predicted as employee pay per hour, & we are builds in a classifications models in predict when it is na patient have been diabetes.

Index terms: Deep Learning, Big data.

I. INTRODUCTION

In the recent terms as BigData mean collect, process & presents in the result of the small amount as data we comes on high speed into various of the format. The customary Machine Learning tool has short coming while if they face for Big Data & need to solving into BigData area problem. The BigData & deeplearning is two important words into data sciences nowadays. BigData Analytic & Deep Learning are two high-focus of data science. The larger volume into data collected among organization is utilized with several purposes like solved as problem into marketing, technology, medical science, national intelligence, fraud detection etc. The customary data processing system is not adequate into handles, analyzed & processing in the collect data is unlabelled, uncategorized & very complex.

Deeplearning as an increasing popular sub set as machinelearning. Deeplearning model is build into uses as neural network. At neural networks taking to input, which is the processing to out of sight layer using weight is adjusted throughout training. In the modeling as spitted into predictions. The weight is so adjust as finding into pattern as order as make good prediction. The user not required to specific value whatever it pattern is looked as the neural

Revised Manuscript Received on December 12, 2019

C. Rajeev, Department of CSE, Malla Reddy College Of Engineering For Women, Maisammaguda, Dhulapally, Kompally, Medchal (M), Secunderabad, Telangana 500100

A. Damodar, Department of CSE, Malla Reddy College Of Engineering For Women, Maisammaguda, Dhulapally, Kompally, Medchal (M), Secunderabad, Telangana 500100

networks learn as it isowning. Deeplearning is suitable form exploit larger volume of the data & form analyzes rare data from multiple source and into various style.

II. REVIEW OF LITERATURE

BIGDATA:

In the terms of “big data” as relatively latest, in the act into gather & stored into larger amount of the information with event analysis as age old. In the concept of gained moment into the near the beginning 2000 which is industry analysis Doug Laney articulated in the now main stream definite of the big data to the three Vs.

DEEP LEARNING:

Deeplearning as aspects into artificial intelligence (AI) (i.e), concerns into emulating to the learn approaches as that human being used to gain certain type into knowledge's. In the simple, deep learning may be during into ways as automatic predictive analytic shared .Deep Learning as sub-field to machine learning concentrate for algorithm inspire among the structure & function to the brain known as artificial neuralnetworks. YoshuaBengio is other leaders into deep learning though begin for strong interested as the automatic featurelearning as larger neural network is capable into achieved. It is described as deep learning into term of algorithm capability as discovered as & learn good representations using feature learning.“Deep learning algorithms seen to use unknown structures into the input distributions to order into discover good representation, often at multiple levels, with higher-level learned features defined in terms of the low level feature”.

KERAS:

Keras as powerful & easy to uses from free open source and Python libraries with developed into evaluate into deep learning model. It is efficient numeric computation library &Tensor Flows & it is allowed into defined & train neural networks model as just a few line as code. Uses if Keras you are needed into deeplearning library as a

1. Allow with easy & fast prototype into user friendliness, modularity, & extensibility
2. Supports convolution networks and recurrent networks, as known as combinations in the two.
3. Runs seamlessly on CPU & GPU. And run out Python 2.7 / 3.5 & it is seamless executed into GPU and CPU gives as the under as framework.

Build Deep Learning Models with Keras:

The focused into Keras as ideal model is the main types in the modeling as known to Sequence while it can linear stacks of the layer. It is created as a sequence & adding layer in that ordered as that your wishes with the computation is perform. Once define, your run the model which is made of uses as the under frameworks in to optimizes in the computational is performed through the model. It is specifically to the losses function & the optimizes is used. Once compile, in the models might be fit into data. It is finished as one batching as data on time that fire off entire models trained as region. Whatever is all the computes happened. Once trained, may be uses in your models as to making into prediction at new data.

It is summarizes in the constructions of the deeplearning model into Kera as following:

1. It is called as you models and it can create as sequences& add layer.
2. Compile you can model. Specific losses to function & optimizer.
3. Fits into your models. Executed to the models using data's.
4. Making as prediction. Uses in the model as generated prediction on new data's.

III. PROPOSED WORK

Keras as user friendly neuralnetworks libraries write into Python's. It is proposed into two deeplearning model uses as Keras: one form regressions & one with classifications. It is built at a regression models as to predicted into employee run per hour, & it is build on classifications models as to predicts which is a patients have been diabetes. The dataset can be uses as relative cleans, therefore it can performing as many data pre-processor into ordering as to getting our data ready with models. Dataset in used to future project cannot be therefore cleaned in this examples, We may missed into value as still can require as to used as data pre-process technique as altered you dataset into getting more accurate result.

Reading in the training data

With the regression deeplearning models, in the first steps as to reading out data they will be used as inputs. In this examples, they have been used to the 'hour wage' data set

```

import pandas as pd

#read in data using pandas
train_df = pd.read_csv('data/hourly_wages_data.csv')

#check data has been read in properly
train_df.head()
    
```

	wage_per_hour	union	education_yrs	experience_yrs	age	female	marr	south	manufacturing	construction
0	5.10	0	8	21	35	1	1	0	1	0
1	4.95	0	9	42	57	1	1	0	1	0
2	6.67	0	12	1	19	0	0	0	1	0
3	4.00	0	12	4	22	0	0	0	0	0
4	7.50	0	12	17	35	0	1	0	0	0

Split up the dataset into inputs and targets

Then, it is required as splitted as up you data set to input (train X) & our targets (train y). Our inputs shall every columns never 'wage per hour' therefore 'wage per hour' is what we will be attempting to predict. Therefore, 'wage per hour' can be our targets. It is used as pandas 'drops' functions as drops in the columns 'wage per hour' with our data's framework & stored into the variables 'train X'. and input.

	union	education_yrs	experience_yrs	age	female	marr	south	manufacturing	construction
0	0	8	21	35	1	1	0	1	0
1	0	9	42	57	1	1	0	1	0
2	0	12	1	19	0	0	0	1	0
3	0	12	4	22	0	0	0	0	0
4	0	12	17	35	0	1	0	0	0

```

#create a dataframe with all training data except the target column
train_X = train_df.drop(columns=['wage_per_hour'])

#check that the target variable has been removed
train_X.head()
    
```

We will insert the column 'wage_per_hour' into our target variable (train y).

```

#create a data frame with only the target column
train_y = train_df[['wage_per_hour']]
#view data frame
train_y.head()
    
```

Building the model

Next, we have to build the model. Here is the code:

```

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

#create model
model = Sequential()

#get number of columns in training data
n_cols = train_X.shape[1]

#add model layers
model.add(Dense(10, activation='relu', input_shape=(n_cols,)))
model.add(Dense(10, activation='relu'))
model.add(Dense(1))
    
```

Compiling the model

Then, it is needed as to compile our models. Compile the model taken in two parameter: optimizes as a losses. These are optimizes controls are the learning rate.

```

#compile model using mse as a measure of model performance
model.compile(optimizer='adam', loss='mean_squared_error')
    
```

E. Training the model

Then it will training as our models. at train, it is uses in the 'fit()' functions into models at the follows as five parameter: training data (train X), target data (train y), validations into split, in the number as epochs & callback.



```
from keras.callbacks import EarlyStopping

#set early stopping monitor so the model stops training when it won't
improve anymore
early_stopping_monitor = EarlyStopping(patience=3)

#train model
model.fit(train_X, train_y, validation_split=0.2, epochs=30,
callbacks=[early_stopping_monitor])
```

Making predictions on new data

If it is models into make prediction as a new data, we will be uses in the 'predict()' functions, passed into our new data. The output will be 'wage per hour' prediction.

```
#example on how to use our newly trained model on how to make
predictions on unseen data (we will pretend our new data is saved in
a dataframe called 'test_X').

test_y_predictions = model.predict(test_X)
```

Classification model

Recently as let's moves into onto builds as our model with classifications. while several step it will be repeated for the before models, as only going to over new concept. In these are next model, they are go to predicts since patient has been diabetes or not.

```
#read in training data
train_df_2 = pd.read_csv('documents/data/diabetes_data.csv')

#view data structure
train_df_2.head()
```

	pregnancies	glucose	diastolic	triceps	insulin	bmi	dpf	age	diabetes
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
#create a dataframe with all training data except the target column
train_X_2 = train_df_2.drop(columns=['diabetes'])

#check that the target variable has been removed
train_X_2.head()
```

	pregnancies	glucose	diastolic	triceps	insulin	bmi	dpf	age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33

Which it separates as the targets as columns, it is required to call 'to categorical ()' functions and it can be column can be 'one hot encode'. Current, as patients as no diabetes as represents within as 0 into the diabetes columns & a patient diabetes as represent for on 1. From one-hot encodes, the integer can be removes & an binary variable as input within each categories. In our cases, it is two categories: no diabetes & diabetes. At patient for no diabetes has represent through [1 0] & patient for diabetes as represented among [0 1].

```
from keras.utils import to_categorical

#one-hot encode target column
train_y_2 = to_categorical(train_df_2.diabetes)

#vcheck that target column has been converted
train_y_2[0:5]
```

```
array([[ 0.,  1.],
       [ 1.,  0.],
       [ 0.,  1.],
       [ 1.,  0.],
       [ 0.,  1.]], dtype=float32)
```

```
#create model
model_2 = Sequential()

#get number of columns in training data
n_cols_2 = train_X_2.shape[1]

#add layers to model
model_2.add(Dense(250, activation='relu', input_shape=(n_cols_2,)))
model_2.add(Dense(250, activation='relu'))
model_2.add(Dense(250, activation='relu'))
model_2.add(Dense(2, activation='softmax'))
```

The last layer of our model has 2 nodes — one for each option: the patient has diabetes or they don't. The activation is 'softmax'. Softmax makes the output sum up to 1 so the output can be interpreted as probabilities. The model will then make its prediction based on which option has a higher probability. #compile model using accuracy to measure model performance model_2.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy']). We will use 'categorical_crossentropy' for our loss function. This is the most common choice for classification. A lower score indicates that the model is performing better. To make things even easier to interpret, we will use the 'accuracy' metric to see the accuracy score on the validation set at the end of each epoch. #train model model_2.fit(X_2, target, epochs=30, validation split=0.2, callbacks=[early stopping monitor])

IV. CONCLUSION AND FUTURE ENHANCEMENT

And this presented into significant challenge as deeplearning Deep learning methods as we used with data analyzed into therefore that capable into helps into finding as abstract pattern as Big Data. They will applys into Deep Learning as Big Data, they can finds into called as useful pattern in that have been not possible.



Deep Learning is the advantages as potential provides at a solutions of addressing to the data analysis & learning problem found out into huge volume in input data's. The proposed models , as two model is utilizes as trains and efficiently into deep machine learning base prediction model with predicting diabetes into patients & predicts as employee wage per hour . Deep learning make into this tasks as more effective on deep learning be efficiently in cases of the images data process. Form future works, optimizates as a result has been done form improves as the performances of the predictions. Furthermore, volumes as a images data is collected & data processing to done into top and Hadoop framework.

REFERENCES

1. Karan Jakhar, Nishtha Hooda, "Large Data Deep Learning Framework utilizing Keras: A Case Study of Pneumonia Prediction" IEEE 978-1-5386-6947-1/18,2018
2. Yann LeCun, Yoshua Bengio and Geoffrey Hinton, "Profound learning", Nature, Vol 521, May 2015.
3. Xue-Wen Chen and Xiaotong Lin, "Huge Data Deep Learning: Challenges and Perspectives", IEEE, Volume 2, May 2014.
4. Shui Yu, "Huge Privacy: Challenges and Opportunities of Privacy Study in the Age of Big Data", IEEE, Volume 4, June 2016.
5. P. Rajpurkar et al. "Chexnet: Radiologist-level pneumonia recognition on chest x-beams with profound learning."arXiv preprint arXiv:1711.0522,2017.
6. WHO. Pneumonia, 2016 [Online] Available: <http://www.who.int/news-room/truth-sheets/detail/pneumonia> [Accessed: May 24, 2018]
7. X. Chen. "Huge information profound learning: difficulties and perspectives."IEEE get to", pp. 514-525, 2014.
8. A. Gandomi et al. "Past the publicity: Big information ideas, techniques, and analytics."International Journal of Information Management vol 35(2), pp.137-144, 2014.
9. Building A Deep Learning Model utilizing Keras, 2018, [ONLINE] Available: <https://towardsdatascience.com/building-a-profound-learning-model-utilizing-keras-1548ca149d37>
10. Daniele Ravi, Charence Wong, Fani Deligianni, Melissa Berthelot, Javier Andreu-Perez, Benny Lo, and Guang-Zhong Yang. "Profound Learning for Health Informatics", IEEE Journal of Biomedical and Health Informatics, Vol. 21, No. 1, January 2017.
11. B M Wilamowski, Bo Wu, and Janusz Korniak, "Enormous information and Deep Learning", , twentieth Jubilee IEEE International Conference on Intelligent Engineering Systems, June 30-July 2, 2016.
12. X. Chen, and X. Lin, "Enormous Data Deep Learning: Challenges and Perspectives". In Access, IEEE, vol.2,pp.514,525,doi:10.1109/ACCESS.2014.2325029, 2014.
13. L. Deng, "An instructional exercise review of models, calculations, and applications for Deep Learning", In APSIPA Transactions on Signal and Information Processing, 2013.
14. H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring systems for preparing profound neural systems," J. Mach. Learn. Res., vol. 10, pp. 1–40, Jan. 2009.
15. K. Kavukcuoglu, M. A. Ranzato, R. Fergus, and Y. LeCun, "Learning invariant highlights through topographic channel maps," in Proc. Int. Conf. CVPR, 2009, pp. 1605–1612.