

Package ‘deepLearnR’

March 4, 2016

Title Interface to TensorFlow Deeplearning Framework

Version 0.0.0.9000

Date 2016-03-13

Description Package provides Classifier with different architectures - simple Linear Neural Network, Deep Neural Network and Recursive Neural Network (rnn,gru & lstm). The package interfaces with Tensorflow via the skflow python package.

Depends R (>= 3.2.3)

SystemRequirements Python pckages TensorFlow, skflow and Pandas

License GPL-3

LazyData true

Imports rPython

RoxygenNote 5.0.1

Repository CRAN

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Krishna Sankar [aut, cre],
Billy Vreeland [aut],
Tej Azad [aut]

Maintainer Krishna Sankar <ksankar42@gmail.com>

R topics documented:

deepLearnR	2
TensorFlow.Classifier	2
TensorFlow.predict	3
titanic.data	4
Index	5

deepLearnR	<i>Deep Learning interface to TensorFlow from R Leverage the distributed multicore capabilities of TensorFlow</i>
------------	---

Description

Functions to create deepLearning architectures and associated datasets. See examples for functions [TensorFlow.Classifier](#) and [TensorFlow.predict](#).

References

- [1] rPython and data in and out of pandas <https://statcompute.wordpress.com/2013/10/13/rpython-r-interface-to-python/>
- [2] some python code refactored from skflow examples in Tutorials (1,2 & 3) by Illia Polosukhin
<https://medium.com/@ilblackdragon/tensorflow-tutorial-part-1-c559c63c0cb1#.njjgnw8yh>
<https://medium.com/@ilblackdragon/tensorflow-tutorial-part-2-9ffe47049c92#.xxksiy8gg>
<https://medium.com/@ilblackdragon/tensorflow-tutorial-part-3-c5fc0662bc08#.md7qum553>
- [3] python code from skflow examples <https://github.com/tensorflow/skflow/tree/master/examples>

See Also

[TensorFlow.Classifier](#)
[TensorFlow.predict](#)

TensorFlow.Classifier	<i>Create Classifier model based on the parameters</i>
-----------------------	--

Description

Create Classifier model based on the parameters

Usage

```
TensorFlow.Classifier(modelTag, XTrain, YTrain, nClasses = 2,
    miniBatchSize = 128, steps = 500, optimizer = "SGD",
    learningRate = 0.05, hiddenUnits = NA, nnType = "linear",
    netType = "ReLU", cellType = "lstm")
```

Arguments

modelTag	Tag for this model - can be referenced in other calls like prediction
XTrain	The X Matrix for training
YTrain	The Y Matrix for training
nClasses	The number of classes
miniBatchSize	Batch Size for the mini batch for optimization algorithms like SGD

steps	Number of epochs for training
optimizer	The Optimizer algorithm = "SGD", "Adam", "Adagrad"
learningRate	The learning rate for optimize algorithm
hiddenUnits	The number and architecture of hidden unit layers e.g. [10,20,10]
nnType	The network type = "linear", "rnn", "dnn", "skit"
netType	The network type for the final round = "ReLU", "tanh"
cellType	The cell type for rnn network = "rnn", "gru", "lstm"

See Also

[TensorFlow.predict](#)

Examples

```
{
  Y <- titanic.data$Survived
  X <- deepLearnR::titanic.data[,c("Age", "SibSp", "Fare", "Pclass")]
  X$Age[is.na(X$Age)] <- mean(X$Age, na.rm=TRUE)
  set.seed(512)
  inTrain <- sample(1:nrow(X), trunc(nrow(X)*0.8))
  X.Train <- X[inTrain,]
  Y.Train <- Y[inTrain]
  X.Test <- X[-inTrain,]
  Y.Test <- Y[-inTrain]
  deepLearnR::TensorFlow.Classifier(modelTag="tflr-03", X=X.Train, Y=Y.Train, steps=5000)
  pred <- deepLearnR::TensorFlow.predict(modelTag="tflr-03", X=X.Test, Y=Y.Test)
  accuracy <- sum(pred == Y.Test)/length(Y.Test)
  print(accuracy) # Should be ~ 0.6312849
  pred <- deepLearnR::TensorFlow.predict(modelTag="tflr-03", X=X, Y=Y)
  accuracy <- sum(pred == Y)/length(Y)
  print(accuracy) # Should be ~ 0.6397306
}
```

TensorFlow.predict	<i>Predict using a model(modelTag) the Yvalues for the X Matrix</i>
--------------------	---

Description

Predict using a model(modelTag) the Yvalues for the X Matrix

Usage

```
TensorFlow.predict(modelTag, XTest, YTest)
```

Arguments

modelTag	Tag for this model - referenced in the model ceate calls like TensorFlow.Classifier
XTest	The X Matrix for test or prediction
YTest	The Y Matrix for test, to calculate the accuracy

See Also[TensorFlow.Classifier](#)

titanic.data*The titanic Dataset*

Description

A test data set of 12 variables

- PassengerId. Id of the passenger
- Survived. Whether they survived or not
- Pclass. Class (1,2,3)
- Name. Name of the passenger
- Sex.
- Age.
- Sibsp. Number of Siblings/Spouses Aboard
- Parch. Number of Parents/Children Aboard
- Ticket. Ticket Number
- Fare.
- Cabin.
- Embarked. Port of Embarkation

Index

*Topic **data**

titanic.data, [4](#)

deepLearnR, [2](#)

deepLearnR-package (deepLearnR), [2](#)

TensorFlow.Classifier, [2](#), [2](#), [4](#)

TensorFlow.predict, [2](#), [3](#), [3](#)

titanic.data, [4](#)