

Package ‘Buddle’

December 21, 2018

Type Package

Title A Deep Learning Package for Statistical Classification Analysis

Version 1.0

Date 2018-12-11

Author Jiwoong Kim <ji_woong_kim@hotmail.com>

Maintainer Jiwoong Kim <jwboys26@gmail.com>

Description Statistical classification has been popular among various fields and stayed in the limelight of scientists of those fields. Examples of the fields include clinical trials where the statistical classification of patients is indispensable to predict the clinical courses of diseases. Considering the negative impact of diseases on performing daily tasks, correctly classifying patients based on the clinical information is vital in that we need to identify patients of the high-risk group to develop a severe state and arrange medical treatment for them at an opportune moment. Deep learning - a part of artificial intelligence - has gained much attention, and research on it burgeons during past decades. It is a veritable technique which was originally designed for the classification, and hence, the EzDL package can provide sublime solutions to various challenging classification problems encountered in the clinical trials. The EzDL package is based on the back-propagation algorithm which performs a multi-layer feed-forward neural network. This package contains two functions: `Buddle_Main()` and `Buddle_Predict()`. `Buddle_Main()` builds a feed-forward neural network model and trains the model. `Buddle_Predict()` provokes the trained model which is the output of `Buddle_Main()`, classifies given data, and make a final prediction for the data.

Encoding UTF-8

Depends R (>= 3.2.2)

License GPL-2

LazyData TRUE

Imports Rcpp (>= 0.12.17)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 6.1.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-12-21 15:50:03 UTC

R topics documented:

Buddle_Main	2
Buddle_Predict	3

Index	5
--------------	----------

Buddle_Main	<i>Building a multi-layer feed-forward neural network model for statistical classification</i>
-------------	--

Description

Building a multi-layer feed-forward neural network model for statistical classification

Usage

```
Buddle_Main(Data, Label, Train_Size, Batch_Size, Optimization = "SGD",
  Learning_Rate = 0.05, Iteration = 100, Layer = 3, Neuron = 20,
  Activation = "Sigmoid")
```

Arguments

Data	- Input matrix.
Label	- Vector of training labels.
Train_Size	- Size of data which is used for training .
Batch_Size	- Batch size.
Optimization	- Method used to minimize loss function. It can take one of "SGD", "Moment", or "AdaGrad."
Learning_Rate	- Default is 0.05.
Iteration	- Number of iterations. Default is 100.
Layer	- Number of layers. Default is 3.
Neuron	- Number of neurons. Default is 20.
Activation	- The name of activation function. It takes either "Relu" or "Sigmoid."

Value

Loss - Vector of values of the loss function.
W - Matrix of weights in the first layer
b - Vector of weights in the first layer
ZList - List of matrices of weights in the middle layers
cList - List of vectors of weights in the middle layers
ZFinal - Matrix of weights in the final layer
cFinal - Vector of weights in the final layer

Train_acc - Accuracy of the classifier when applied to the train data
 Test_acc - Accuracy of the classifier when applied to the test data
 Epoch - Number of epoch.

References

- [1] Geron, A. Hand-On Machine Learning with Scikit-Learn and TensorFlow. Sebastopol: O'Reilly, 2017. Print.
 [2] Han, J., Pei, J, Kamber, M. Data Mining: Concepts and Techniques. New York: Elsevier, 2011. Print.

See Also

Buddle_Predict

Examples

```
#####
n <- 50
p <- 3
Data <- matrix(runif(n*p, 0,50), nrow=n, ncol=p) ##### Generate n-by-p input matrix for data
Label = sample.int(n, n, replace=TRUE) ##### Generate n-by-1 vector for the label
Layer = 6 ##### Number of layers
Neuron = 20 ##### Number of neurons
lr = 0.005 ##### Learning rate
Iter = 100 ##### Iteration
Opt = "SGD" ##### Method to optimize the loss function
Act = "Sigmoid" ##### Activation function
TrSize = 20 ##### Train_Size
BatSize = 5 ##### Batch_Size
DLResult = Buddle_Main(Data, Label, TrSize, BatSize, Opt, lr, Iter, Layer, Neuron, Act)
Loss_Vector = DLResult$Loss
Train_Accuracy = DLResult$Train_acc
Test_Accuracy = DLResult$Test_acc
```

Buddle_Predict	<i>Making a prediction based on the trained network model obtained from Buddle_Main()</i>
----------------	---

Description

Making a prediction based on the trained network model obtained from Buddle_Main()

Usage

```
Buddle_Predict(1st, xVec)
```

Arguments

lst - Feed-forward neural network model
xVec - Vector of data

Value

Predicted classification for given data. #####

References

- [1] Geron, A. Hand-On Machine Learning with Scikit-Learn and TensorFlow. Sebastopol: O'Reilly, 2017. Print.
- [2] Han, J., Pei, J, Kamber, M. Data Mining: Concepts and Techniques. New York: Elsevier, 2011. Print.

See Also

Buddle_Main

Examples

```
n = 50
p = 3
Data = matrix(runif(n*p, 0,50), nrow=n, ncol=p) ##### Generate n-by-p input matrix for data
Label = sample.int(n, n, replace=TRUE) ##### Generate n-by-1 vector for the label
DLResult = Buddle_Main(Data, Label, 20, 5, "SGD", 0.01, 100, 6, 20, "Sigmoid")

xVec=rep(0, times=p)
Index = Buddle_Predict(DLResult, xVec) ##### Predict for given xVec
```

Index

Buddle_Main, [2](#)

Buddle_Predict, [3](#)