# Package 'DamiaNN'

January 20, 2025

**Type** Package

**Title** Neural Network Numerai

**Version** 1.0.0

**Author** Damian Siniakowicz

**Maintainer** Damian Siniakowicz <DamianSiniakowicz@gmail.com>

**Date** 2016-09-13

**Description** Interactively train neural networks on Numerai, <https://numer.ai/>, data. Generate tournament predictions and write them to a CSV.

**Imports** caret, methods, testthat

**License** GPL-3

**LazyData** FALSE

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-09-14 18:50:32

## Contents

## back_propogation,Neural_Network,numeric,numeric,numeric-method

*back prop*

### Description

updates connection strengths using results of last forward prop

### Usage

```
## S4 method for signature 'Neural_Network,numeric,numeric,numeric'
back_propogation(object,
  target, regularization_parameter, learning_rate)
```

### Arguments

| | |
|---|---|
| object | is a Neural_Network |
| target | is a numeric vector |
| regularization_parameter | is non-negative number punishes strong connections |
| learning_rate | is a positive number that controls the rate at which connections are adjusted |

### Value

Neural_Network

## forward_propogation,Neural_Network,matrix-method

*f_prop*

### Description

... part of the training program

### Usage

```
## S4 method for signature 'Neural_Network,matrix'
forward_propogation(object, dataset)
```

### Arguments

| | |
|---|---|
| object | is a Neural_Network |
| dataset | is a matrix not containing the target vector |

### Value

Neural_Network

## `Get_Cost,Neural_Network,numeric-method` *cost*

### Description

get the logarithmic loss for a set of predictions

### Usage

```
## S4 method for signature 'Neural_Network,numeric'
Get_Cost(object, target)
```

### Arguments

`object` ... a Neural_Network that has run forward_prop at least once

`target` ... a numeric vector ... the target ...

### Value

Numeric

## `Get_LogLoss` *log loss*

### Description

get log loss

### Usage

```
Get_LogLoss(predictions, target)
```

### Arguments

`predictions` is a numeric vector

`target` is a numeric vector

### Value

Numeric

## `Get_Number_Observations,Neural_Network-method` *num observs*

### Description

returns the number of observations that the network has processed

### Usage

```
## S4 method for signature 'Neural_Network'
Get_Number_Observations(object)
```

### Arguments

| | |
|---|---|
| `object` | ... a Neural Network that has called fprop. ie. that has called train/predict |

### Value

Numeric

## `initialize,Neural_Network-method` *init*

### Description

initalizes a neural network capable of studying datasets with ncol = to the ncol(sample_dataset) and making predictions on such datasets

### Usage

```
## S4 method for signature 'Neural_Network'
initialize(.Object, number_predictors,
  hidden_layer_lengths)
```

### Arguments

| | |
|---|---|
| `.Object` | ... a Neural_Network object |
| `number_predictors` | ... a numeric telling how many preditors there are |
| `hidden_layer_lengths` | ... a numeric telling the number of layers and the number of neurons in each layer |

### Details

NN is parametrized by its connection_strength matrices

### Value

Neural_Network

## `Neural_Network-class` *Neural Network implementation*

### Description

Neural Network implementation

## `Predict,Neural_Network,data.frame-method` *predict stuff*

### Description

returns predictions

### Usage

```
## S4 method for signature 'Neural_Network,data.frame'
Predict(object, dataset)
```

### Arguments

| | |
|---|---|
| `object` | : a neural network |
| `dataset` | : a dataframe of features and observations |

### Value

Numeric

`Start` *start script*

### Description

main function that runs the interactive script

### Usage

```
Start()
```

### Details

takes your numerai training data and trains a neural network to your architectural specifications. provides you with the out of sample error offers to retrain with a new architecture or predict on a numerai tournament dataset. Can then write the predictions to a CSV

`Train,Neural_Network,data.frame,numeric,numeric,numeric-method`
*train the NN*

### Description

gets NN parameters that minimize cost on dataset using optimization_method

### Usage

```
## S4 method for signature 'Neural_Network,data.frame,numeric,numeric,numeric'
Train(object,
  dataset, regularization_constant, learning_rate, tolerable_error)
```

### Arguments

`object` is a Neural Network

`dataset` is a data.frame, the original data frame that includes the target

`regularization_constant`
is a numeric

`learning_rate` is a numeric

`tolerable_error`
is a numeric, units : log loss

### Value

Neural_Network

# Index