

Modules

| | | |
|----------------------|-----------------------------------|----------------------|
| math | matplotlib.pyplot | time |
| os | tensorflow | |

Classes

[builtins.object](#)

[NNcreators](#)

class **NNcreators**([builtins.object](#))

[NNcreators](#)(numeroLayer, numeroNodi, percentualeTs, funzione, ModelName, datasetPath)

Classe di operazioni di Rete Neurale

Attributi

reteNeurale: NeuralNetwork

Oggetto Rete Neurale

column_names: string[]

Nomi di tutte le colonne del dataset

feature_names: string[]

Nomi delle colonne riferite alla caratteristiche

label_name: string

Nome della colonna riferito alla tipologia

class_names: string[]

Tipologie di errori

Methods defined here:

__init__(self, numeroLayer, numeroNodi, percentualeTs, funzione, ModelName, datasetPath)

Inizializzazione degli attributi della reteNeurale

Parametri

numeroLayer: int

Numero di hidden-layer della Rete Neurale

numeroNodi: int

Numero di nodi per layer

percentualeTs: int

Percentuale del dataset dedicata al training

funzione: string

Funzione di decisione

ModelName: string

Nome della rete

datasetPath: string

Percorso del dataset

generazioneDatasets(self)

Generazione dei dataset che vengono utilizzati per training e testing della rete

generazioneFileDatasets(self)

Genera i due file di dataset:

- uno dedicato per il training della rete

- l' altro dedicato al testing della rete

generazioneModello(self)

Generazione della Rete Neurale con le caratteristiche scelte

previsione(self, predictionPath)

Effettua la prediction sul file scelto

Parametri

predictionPath: string

Percorso del file contenente il dataset su cui si deve effettuare la 'prediction'

Ritorna

```
ris: string[]
Risultati della 'prediction' nella forma:
"Riga: " + i +
"Previsione: " + nomeFiorePredetto + percentualediAccuratezza
```

risultati(self)

Salvataggio della Rete Neurale generata

Ritorna

testAccuracyResult: string[]

Percentuale di 'Accuracy' del test effettuato

trainAccuracyResult: string[]

Percentuale di 'Accuracy' del training effettuato

salvataggioRisultati(self)

Salva l' andamento della Loss e della Accuracy in una immagine (.png)

setModello(self, percorso)

Caricamento di una Rete Neurale già configurata

Parametri

percorso: string

Percorso in cui si trova la Rete Neurale scelta

switch_fun(self, argument)

Funzione di switch per le varie funzioni possibili

Parametri

argument: string

Stringa contenente il nome della funzione di decisione scelta

Ritorna

switcher.get(argument): tf.nn.*

Funzione di decisione che viene utilizzata nella Rete Neurale

testing(self)

Effettua il testing della Rete Neurale

trainingModello(self)

Training della Rete Neurale

Data

```
__email__ = 'ste.lavaggi@gmail.com matteo.cardano@gmail.com'
__maintainer__ = 'Matteo Cardano, Stefano Lavaggi'
__warningregistry__ = {'version': 12}
```

Author

Matteo Cardano, Stefano Lavaggi