**Sample NeuralNetworks**

Design & Usage

Parameter Doc

Version V0.1

Table of contents

[1 Introduction 6](#_Toc11784960)

[1.1 Short overview 6](#_Toc11784961)

[1.2 Important information 6](#_Toc11784962)

[2 Use Cases 6](#_Toc11784963)

[2.1 Use Cases with Sub Use Cases 6](#_Toc11784964)

[2.2 Functional/Business Open Issues 7](#_Toc11784965)

[2.3 Technical Open Issues 8](#_Toc11784966)

[3 Architecture 8](#_Toc11784967)

[3.1 Business 8](#_Toc11784968)

[3.2 Software 8](#_Toc11784969)

[3.2.1 Project folder structure 8](#_Toc11784970)

[3.2.2 System folder structure 10](#_Toc11784971)

[3.3 Technology 10](#_Toc11784972)

[3.3.1 Front End 10](#_Toc11784973)

[3.3.2 Web Dashboard 10](#_Toc11784974)

[3.3.3 Excel Dashboard 10](#_Toc11784975)

[3.3.4 Visualization 10](#_Toc11784976)

[3.3.5 Back End - Deep Learning 10](#_Toc11784977)

[3.3.6 Back End - Other 11](#_Toc11784978)

[3.3.7 Development 11](#_Toc11784979)

[4 Application design 12](#_Toc11784980)

[4.1 General concepts 12](#_Toc11784981)

[4.2 Business components 12](#_Toc11784982)

[4.3 Workflow & Processes 12](#_Toc11784983)

[4.4 Interfaces to extern 12](#_Toc11784984)

[4.5 Data 12](#_Toc11784985)

[4.6 User interfaces 12](#_Toc11784986)

[4.6.1 Elements 12](#_Toc11784987)

[4.6.2 Controls 13](#_Toc11784988)

[5 Supplements 13](#_Toc11784989)

[5.1 Glossary 13](#_Toc11784990)

[5.2 References 13](#_Toc11784991)

[6 Vorlage Tabellen 13](#_Toc11784992)

[6.1 Struktur 13](#_Toc11784993)

[6.2 Positive tables 13](#_Toc11784994)

[6.3 Negative tables 13](#_Toc11784995)

[7 Level I 14](#_Toc11784996)

[7.1 Level II 14](#_Toc11784997)

[7.1.1 Level III 14](#_Toc11784998)

**Version control Template**

|  |  |
| --- | --- |
|  | |
| **Created by** | Dr. rer. nat. Christian Dury |
| **Maintained by** |  |

|  |  |
| --- | --- |
|  | |
| **Document owner** |  |
| **Author** |  |
| **Status** |  |
| **Confidentiality** | Only for internal usage |
| **Last review** |  |

**Version log Templates**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** |  |  |  |
| **0.1** | 10/2018 | Christian Dury | Initial version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Version control document**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Datum** |  |  |
| **V0.1** | 10/2018 | Christian Dury | Initial version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
|  | |
| **Date last touched** | 6/18/2019 |
| **Number of pages** | 14 |
| **Filename** | Sample\_NNs\_V0.1.docx |

# Introduction

**TODO:**

* **Switch to English language**
* **Set & actualize Document variables**
* **Arrangement of tables**
* **Numbering of tables header**

**DISCLAIMER:** This is a living document and is therefore not static.

This document gives an overview of the parameter definition and usages for the Sample Neural Networks (eventually resulting in an Template).

The technical documentation of the Neural Network Trading Advisor is done in the code itself with Github style annotations and will be generated with sphinx.

## Short overview

* The Sample Neural Networks is a collection of Neural Networks with different architectures.
* But even with different Architectures there should bes something which resembles a template for better usage of these and new networks.

# Use Cases

## Categorization Networks

All networks with categorization should be in one “frame work”

Hierachical structure bottom up:

* networks\_keras/Base\_Supervised\_Categorical
* network\_keras/<NettworkTyp>\_K\_<Structure>\_<Dataset>.py (i.e. SNN\_K\_nLayer\_HAR.py )
* K\_Categorical.py
* HyperOpt.py

### HyperOpt

Can be run stand alone. This is for hyperparameter search.

Imports:

* run\_and\_get\_error (from K\_Categorical)
* HyperParameter (from K\_Categorical)

Parameter:

Parameters in space (init\_space, build\_space, data\_space, learning\_space) take precedence over hyperparamters

* HYPER\_BASE: <optimization\_name>\_<timestamp>

Instantiate:

* Hyperparameter(model\_name=<optimization\_name>\_<trial\_id>, loglevel=3)

Runs:

* run\_and\_get\_error(hyperparameters)

### K\_Categorical

Can be run stand alone.

Imports:

* NeuralNetwork
* HyperParameter
* model\_name

Instantiate:

* Hyperparameter(model\_name=<imported model\_name from Network specifics>, loglevel=3)

Runs:

* NeuralNetwork
* Category\_evaluation if run from command line

### Network specifics

Can be run stand alone.

Import:

* data : Function for loading data
* LABELS: Labels of the different categories

#### Hyperparameters

Provides:

* model\_name : Name of the specific network
* Class Hyperparameters(model\_name,loglevel)

|  |  |  |  |
| --- | --- | --- | --- |
| **Typ** | **C** | **Name** | **Description/Usage** |
| **Technical & HyperOpr** |  |  |  |
|  | B | model\_name | <model\_name> |
|  | B | parent\_name | “”: <HYPER\_BASE> in HyperOpt |
|  | B | tid | None: <trial id> in HyperOpt |
|  | B | tb\_suffix | “”: <HYPERBASE>\_tid\_<tid> in HyperOpt |
|  | B | timestamp | GMT timestamp |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Filesystem**  **&Paths** |  |  |  |
|  | B | model\_path | <Home\_path>/models/ |
|  | B | model\_dir | <model\_path>/<model\_name>\_<timestamp>/ Callback: ModelCheckpoint & Plot: KerasModel |
|  | B | tensorboard\_path | <Home\_path>/tensorboard/ |
|  | B | tensorboard\_dir | < tensorboard\_path >/<model\_name>\_<timestamp>/  Callback: Tensorboard |
|  | B | data\_path | <Home\_path>/data |
|  | H | data\_dir | <data\_path> |
|  | B | log\_path | <Home\_path>/log/ |
| **Logging** |  |  |  |
|  | B | loglevel | 0: All Msg 1: No INFO 2: No INFO & WARNING 3: No INFO, WARNING & ERROR  loglevel->os.environ[“TF\_CPP\_MIN\_LOG\_LEVEL”]  0: model.summary() |
|  | B | show\_graphs | True: Plot <model\_dir>/<model\_name>.png of netowrk |
|  | B | fitting\_verbosity | 0: verbosity in model.fit |
|  | B | eval\_verbosity | 0: verbosity of model.evaluate |
| **Data (Hyper)** |  |  |  |
|  | H | columns\_to\_use | Used in load\_data() |
|  | H | column\_names |  |
|  | H | LABELS |  |
| **Model (Calc)** |  |  |  |
|  | B | n\_input |  |
|  | B | n\_classes |  |
| **Model (Hyper)** |  |  |  |
|  | H | n\_hidden\_1 | Number of neurons in Layer 1 |
|  | H | n\_hidden\_2 | Number of neurons in Layer 2 |
|  | H | n\_hidden\_3 | Number of neurons in Layer 3 |
|  | H | n\_hidden\_4 | Number of neurons in Layer 4 |
|  | H | init\_kernel | “random\_normal” “he\_normal”: Init of weights |
|  | H | activation\_hidden | “relu”: activation function in hidden layer |
|  | H | activation\_output | “softmax”: activation function in output layer |
| **Optimizer (Hyper)** |  |  |  |
|  | H | learning\_rate | Learning rate for Adam optimizer |
|  | H | lambda\_loss\_amount | For L2-regularizer in hidden layers |
|  | H | metric | “accuracy”: used metric |
| **Callbacks** |  |  |  |
|  | B | tb\_update\_freq | Callback: Tensorboard |
|  | B | tb\_write\_graph | Callback: Tensorboard |
|  | B | tb\_write\_images | Callback: Tensorboard |
|  | B | save\_every\_epochs | Callback: ModelCheckpoint |
|  | B | callback\_verbosity | 0: Callback: ModelCheckpoint & EarlyStopping |
|  |  |  |  |
| **Training (Hyper)** |  |  |  |
|  | B | monitor | “val\_loss”: Callback: ModelCheckpoint & EarlyStopping |
|  | B | mode | “auto”, “min” & “max”:  Callback: ModelCheckpoint & EarlyStopping |
|  | B | patience | 5: Callback: EarlyStopping |
|  | B | batch\_size | 0 , >0: Batch size to be trained in model.fit |
|  | B | Epochs | Epochs to train in model.fit |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* …

#### <NetworkClass>(BaseNN)

Provides:

* Members
  + parameter: Hyperparameter object
  + inherited members of BaseNN
* Methods:
  + load\_data: loads data, categories have to be one hot encoded
  + define\_model: Model definition

### Base\_Supervised\_Categorical

Provides:

* Class BaseParameters
  + Initialize
    - model\_name
    - loglevel
  + Members  
    see Class Hyperparameters
  + Methods

|  |  |  |
| --- | --- | --- |
| **Ebene** | **Ebene** | **Object** |
| **Obj1** |  |  |
|  | Subclass | Description |

* Class BaseNN
  + Initialize
  + Members

|  |  |  |
| --- | --- | --- |
| **Ebene** | **Ebene** | **Object** |
| **Obj1** |  |  |
|  | parameter | Hyperparameter-Object |
|  | test\_data | Test data set |
|  | validation\_data | Validation data set |

* + Methods

|  |  |  |
| --- | --- | --- |
| **Ebene** | **Ebene** | **Object** |
| **Interfaces** |  |  |
|  | load\_data |  |
|  | define\_model |  |
|  | train\_model |  |
| **Implemented** |  |  |
|  | calc\_categorical\_accuracy | Loss / Accuracy of Dataset |
|  | is\_vs\_should\_categorical | predicition category# & is category# |

# Vorlage Tabellen

## Struktur

|  |  |  |
| --- | --- | --- |
| **Ebene** | **Ebene** | **Object** |
| **Obj1** |  |  |
|  | Subclass | Description |

## Positive tables

|  |  |
| --- | --- |
| **Tabelle 1** |  |
| Entry 1 | Comment 1 |
| Entry2 | Comment 2 |

cx

## Negative tables

|  |  |
| --- | --- |
| **Tabelle 2** |  |
| Entry 1 | Comment1 |
| Entry 2 | Comment 2 |

# Level I

## Level II

< fehlt noch >

### Level III

<fehlt noch>

#### Last Level IV

<fehlt noch>

##### Last Level IV

<fehlt noch>