FMECA - On the Edge

# Progetto n° 22

Design Requirement Specification Document

DIBRIS – Università di Genova. Scuola Politecnica, Corso di Ingegneria del Software 80154

**DATA - 18/07/2020**

**VERSION: 1.2**

**Autori**

Lavaggi Stefano

Cardano Matteo

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Versione | Data | Autori | Note |
| 1.0 | 06/05/2020 | Cardano Lavaggi | Prima stesura |
| 1.1 | 12/05/2020 | Cardano Lavaggi | Interf. e doc. |
| 1.2 | 19/05/2020 | Cardano Lavaggi | Dia e grafica |
| 1.3 | 29/05/2020 | Cardano Lavaggi | Correzione errori |
| 1.4 | 04/06/2020 | Cardano Lavaggi | Definizione Classi |
| 1.5 | 18/07/2020 | Cardano Lavaggi | Correzioni |
| 1.6 | 24/07/2020 | Cardano Lavaggi | Precisazioni |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Indice dei Contenuti**

1[1Introduzione 3](#_gjdgxs)

[1.1Riferimenti Bibliografici 3](#_30j0zll)

[1.2Definizioni e Acronimi 3](#_1fob9te)

[1.3Overview del documento 3](#_3znysh7)

2[2Descrizione del Progetto 3](#_tyjcwt)

[2.1Introduzione 3](#_3dy6vkm)

[2.2Architettura del Sistema 3](#_1t3h5sf)

[2.3Vincoli e Assunzioni 3](#_4d34og8)

3[3Interfacce 4](#_2s8eyo1)

4[4Data Stores 4](#_3rdcrjn)

5[5Structural Design 4](#_26in1rg)

[5.1Class Diagram 4](#_lnxbz9)

[5.1.1Descrizione della classe 4](#_35nkun2)

[5.2Object Diagram 5](#_1ksv4uv)

6[6Dynamic Models 5](#_44sinio)

[6.1Requisito 5](#_2jxsxqh)

# Introduzione

Lo scopo del documento è di comunicare i dettagli della realizzazione di una NN ai fini di garantire la corretta manutenzione, nel caso specifico di RINA, e fornire in modo più generale un software di analisi e simulazione di NN su HW come RB o AR. La stesura di un documento di questa tipologia permette di organizzare tutti i possibili pensieri e di considerare gli aspetti di design del progetto senza lasciarne fuori.

## Riferimenti Bibliografici

* <https://www.tensorflow.org/>
* <https://www.python.org/>
* <https://keras.io/>
* <https://www.raspberrypi.org/downloads/>
* <https://it.wikipedia.org/wiki/Rete_neurale_artificiale>

## Definizioni e Acronimi

|  |  |
| --- | --- |
| **Acronimo-Nome** | **Definizione** |
| SE | Software Engineering |
| AI | Artificial intelligence |
| NN | Neural network |
| RB | RaspBerry PI |
| AR | Arduino |
| TF | TensorFlow |
| KR | Keras |

## Overview del documento

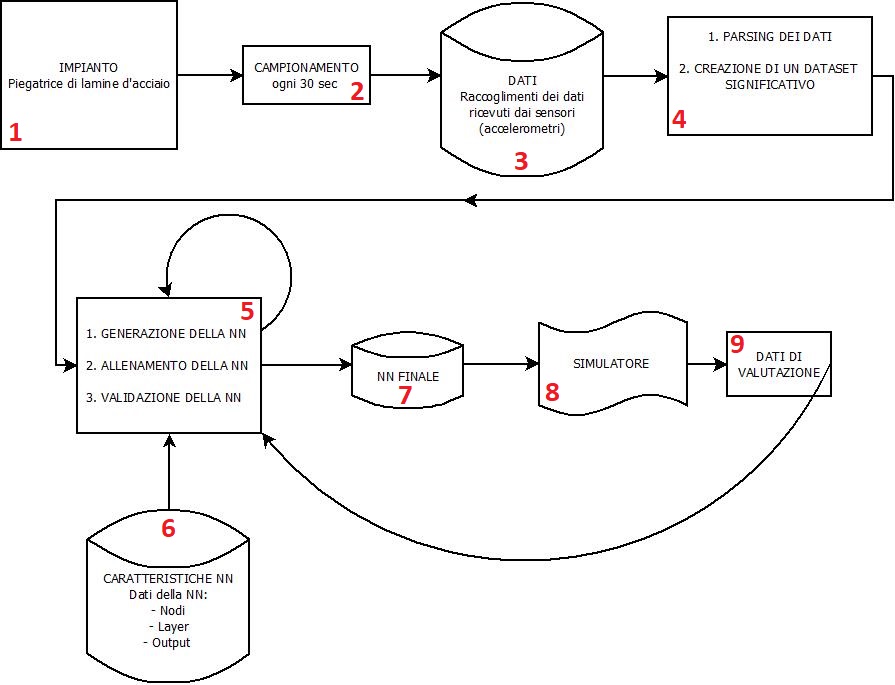
nelle sezioni sottostanti si trova una descrizione del sistema e una successiva rappresentazione grafica dei vari componenti .

# Descrizione del Progetto

## Introduzione

Si vuole realizzare una NN utilizzando il linguaggio di programmazione Python, l’object oriented e la sua adattabilità allo sviluppo di applicazioni di calcolo numerico e testing; sfruttando soprattutto la presenza di librerie “open source” dedicate alla gestione di file contenenti dati, allo sviluppo di NN e al testing di quest’ultime. L’obiettivo è quello di rilevare stati di ‘Danger’ durante il funzionamento di un impianto produttivo, in modo da garantire la corretta manutenzione . Il software permette tramite la piattaforma sviluppata, il training , il testing e la validazione tramite il simulatore , il prodotto fornito è quindi in grado di segnalare opportunamente i vari casi e le problematiche in gioco. Il tool fornito permette all’utente di configurare la rete con grandi libertà, le interfacce ad alto livello permettono una facile interazione e selezione delle caratteristiche.

## Architettura del Sistema



1. Impianto: case study(piegatrice di lamine d’acciaio). Su di essa sono presenti vari sensori, nel test di esempio si parla di 4 sensori montati su punti strategici e caratteristici per il corretto funzionamento del macchinario
2. Acquisizione dati :

* sensore : accelerometro
* unità di misura: g
* frequenza : 0.033 Hertz

1. I dati degli accelerometri sono salvati in un file dati( es. csv, excel )
2. Viste le informazioni ridotte fornite dal sistema al momento , per un corretto test viene generato un dataset significativo ( per una simulazione plausibile si può sfruttare ad esempio Iris-dataset),le caratteristiche generali della rete non pongono alcun limite a cambiamenti dei dataset futuri.
3. Tramite il dataset creato in precedenza e i parametri configurati dall‘utente si andrà a creare la NN . Successivamente una parte del dataset (training set) viene utilizzata per allenare la NN, mentre parte restante del dataset viene utilizzata per validarla (test set). Così da ottenere dei parametri qualitativi della NN creata, se essi non soddisfano le richieste è possibile riallenarla oppure crearne una nuova.
4. L’utente in fase di progettazione può definire le caratteristiche della NN che si vuole generare questo permette un’ottima personalizzazione e adattamento al caso d’uso.
5. Il sistema è in grado di fornire come output la rete NN allenata e opportunamente pesata, mantenendo anche le caratteristiche e i dettagli inseriti dall’utente.
6. Tramite un simulatore si valuterà il funzionamento e le prestazione sull’ HW .
7. Valutazione dei risultati.

## Vincoli e Assunzioni

il contesto applicativo richiede un’implementazione specifica e ottimizzata , in particolare occorre tener conto delle caratteristiche HW . Questo ha vincolato alcune scelte progettuali come ad esempio l’allenamento sul calcolatore e non embedded, il che permette di ridurre i tempi e di sfruttare le differenti capacità di calcolo . allo stesso modo si è deciso di orientarsi verso dispositivi facilmente configurabili e adattabile sia per il “linguaggio che per le caratteristiche HW”, come nel caso del Raspberry (emulatore) . Le scelte sono legate alle difficoltà implementative legate ad una corretta realizzazione di una rete neurale altamente configurabile .

Immagine che contiene screenshot

Descrizione generata automaticamente Immagine che contiene screenshot

Descrizione generata automaticamente

# Interfacce

Sono presenti diverse interfacce in modo da gestire l’esecuzione e lo sviluppo della rete neurale nei vari step. Una di esse è dedicata alla prima fase, precisamente dove viene richiesto allo sviluppatore di inserire i dati della RN che verrà generata. Dopo l’avvenuta generazione avrà luogo il cosiddetto “training” della RN, al termine dell’allenamento tramite opportuni indici è possibile valutare le prestazioni ottenute. Infine è possibile valutare il funzionamento su un set di funzionamento (senza status) per valutare l’effettiva previsione del sistema appena realizzato.

## Interfaccia di Configurazione

In essa troviamo :

* Pulsanti per la selezione del numero di ‘layer’
* Pulsanti per la definizione numero di nodi
* Pulsante per poter avviare il training
* Pulsante per la convalida delle rete
* Pulsante per il salvataggio della rete
* Pulsante per il training della rete
* Pulsante per caricare una rete precedentemente salvata in modo riallenarla o valutarne le prestazioni
* Pulsante per l’upload Rete per simulazione su dispositivo virtuale
* Menù a tendina per la scelta della funzione di decisione (relu,sigmoidal,..)
* Possibilità di scelta della frammentazione del dataset per realizzare TS e VS
* Visualizzazione della scelta di suddivisione
* Text per la visualizzazione di eventuali errori di configurazione
* Campo di testo per l’inserimento del nome della RN

(nella selezione della rete o del dataset si apre il file manager di windows per la selezione )

Immagine che contiene screenshot

Descrizione generata automaticamente

(il salvataggio prevede la realizzazione di una cartella avente come nome = “nome\_Rete” con al suo interno la rete generata e il file contente anch’esso i dati di configurazione selezionati , anche per questi si sfrutta il nome precedentemente inserito )

## Interfaccia di Visualizzazione

Nella parte relativa alla visualizzazione dei dati sono disponibili:

* gli indici prestazionali della RN
  + precisione del training (%)
  + precisione del test(%)
* plot delle funzione di :
  + loss
  + accuracy
* La possibilità di scartare il modello e riconfigurarlo attraverso la scelta di nuovi parametri
* Text per la visualizzazione di errori
* Pulsante per la previsione di risultati dato in input seq di parametri (senza tipologia di errore)
* Tempi di esecuzione all’interno dell’ambiente di simulazione ,in particolare il tempo effettivo di training.

Immagine che contiene screenshot

Descrizione generata automaticamente

(per riallenare o scartare la rete basta “tornare” all’interfaccia precedente di configurazione )

## Interfaccia di Previsione

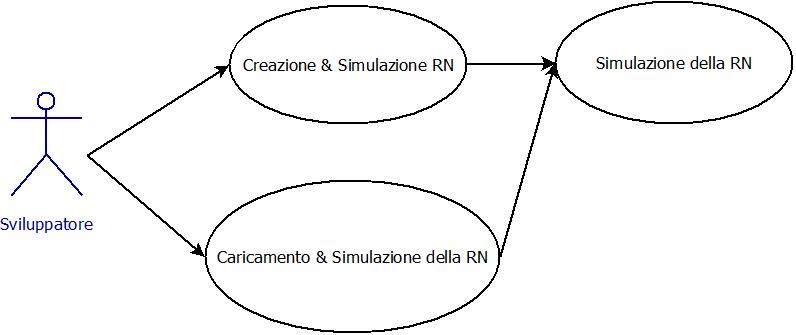
* ‘ListView’ contenente i risultati della previsione ovvero sulla sequenza di input fornita es: il primo campione (riga 0) risulta una tipologia con una determinata probabilità

Immagine che contiene screenshot

Descrizione generata automaticamente

(per tornare alla configurazione o al test di un nuovo campione è sufficiente chiudere questa interfaccia, le caratteristiche e il modello vengono salvate in automatico)

1. **Uses Cases**

****

|  |  |  |
| --- | --- | --- |
| **ID** | **USE CASE** | **NOME DIAGRAM** |
| 5.2.1 | Creazione & Simulazione RN | Progettazione RN e Simulazione (Activity Diagram) |
| 5.2.2 | Caricamento & Simulazione della RN | Caricamento RN e Simulazione (Activity Diagram) |
| 5.3.1 | Simulazione della RN | Simulazione della RN (Sequence Diagram) |

Tabella 1: Collegamenti con i diagram successivi

# Structural Design

## Class Diagram

Immagine che contiene screenshot

Descrizione generata automaticamente

### Descrizione della classe

Dataset: file relativo al dataset contenente i dati rilevati dai vari sensori (nel nostro case-study 4 con i vari rilevamenti), tramite i comandi di configurazione della rete queste viene diviso opportunamente in due, in modo da generare i relativi dati per training e validazione della rete neurale.

**NNcreator**: classe relativa alla gestione e creazione della RN, troviamo vari metodi fra cui ad esempio il metodo per lo split del dataset in due file (training e validation), la generazione del modello secondo le specifiche fornite dall’utente, la possibilità di sfruttare il modello per prevedere i risultati dato un file contenente i campioni (senza lo status) , il salvataggio della rete neurale , l’allenamento per migliorarne le prestazioni.

**Configurazione**: classe relativa alla gestione dell’interfaccia utente per il set delle caratteristiche, questa permette la comunicazione uomo-macchina; in particolare questa classe gestisce l’inserimento delle caratteristiche utente (validando l’input) o il caricamento di una rete precedentemente allenata in modo da effettuare nuovamente il training.

**Configurazione**\_**ui**: crea la visualizzazione della GUI (interfaccia configurazione)

**Visualizzazione**: classe relativa alla gestione dei risultati di training ottenuti dalla simulazione verso l’utente, permette al cliente di visualizzare le caratteristiche ottenute, in particolare vedremo (come nell’immagine precedentemente definita) la precisione del training, del test e un immagine riassuntiva dell’evoluzione delle prestazioni durante la fase di training (andamento Loss e accuratezza), inoltre indica i tempi impiegati nella fase di training.

**Visualizzazione**\_**ui**: crea la visualizzazione della GUI (interfaccia visualizzazione)

**Previsione**: classe relativa alla visualizzazione delle previsioni effettuate sulla serie di dati inserite in input, in particolare tramite listwiev vengono mostrate le previsioni sul dataset

**Previsione**\_**ui**: crea la visualizzazione della GUI ( interfaccia previsione)

**NeuralNetwork**: racchiude sotto un unico oggetto tutti gli attributi utili per caratterizzare e realizzare la rete neurale con la classe precedentemente definita.

**RBSimulator**: simulatore del raspBerry e relativo S.O. data una rete correttamente allenata calcola la previsione sulla base della struttura disponibile.

## Activity Diagram

*ID: 5.2.1* Activity Diagram riferito alla progettazione e alla successiva simulazione della RN.

Immagine che contiene elettronico, remoto, schermo, monitor

Descrizione generata automaticamente

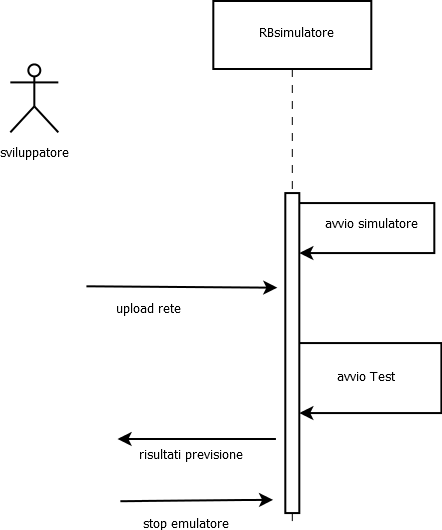
*ID: 5.2.2* Activity Diagram riferito al caricamento e alla successiva simulazione della RN.

Immagine che contiene sedendo, facciata, via, schermo

Descrizione generata automaticamente

## Sequence Diagram

*ID: 5.3.1* Sequence Diagram riferito alla sola simulazione della RN.



(al momento non sono stati trovati simulatori online con caratteristiche accettabili per l’implementazione corretta, si propone comunque un possibile SD in caso di sviluppi futuri)

