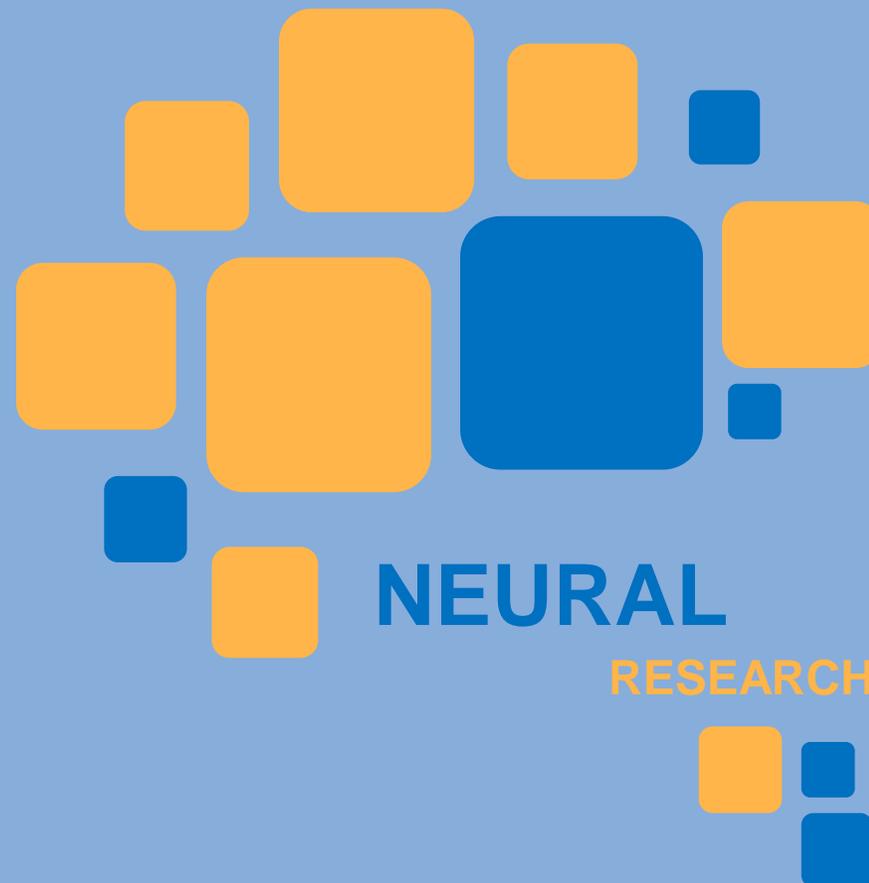


La Neural Research:
una realtà
innovativa in
uno scenario in
cerca di
innovazione



La Neural Research srl è una “start-up innovativa” specializzata nella simulazione software neurale, nel Machine Learning, nel Deep Learning e negli algoritmi di **Intelligenza Artificiale** in generale.

Stato attuale della tecnologia neurale



Attualmente, lo sviluppo della tecnologia di simulazione neurale si trova nelle stesse condizioni degli inizi dell'aviazione.

Stato attuale della tecnologia neurale

Si sapeva che un oggetto più pesante dell'aria poteva volare. Lo si sapeva perché si vedeva che gli uccelli sapevano volare. Quindi i primi tentativi vennero fatti ad imitazione del volo degli uccelli.



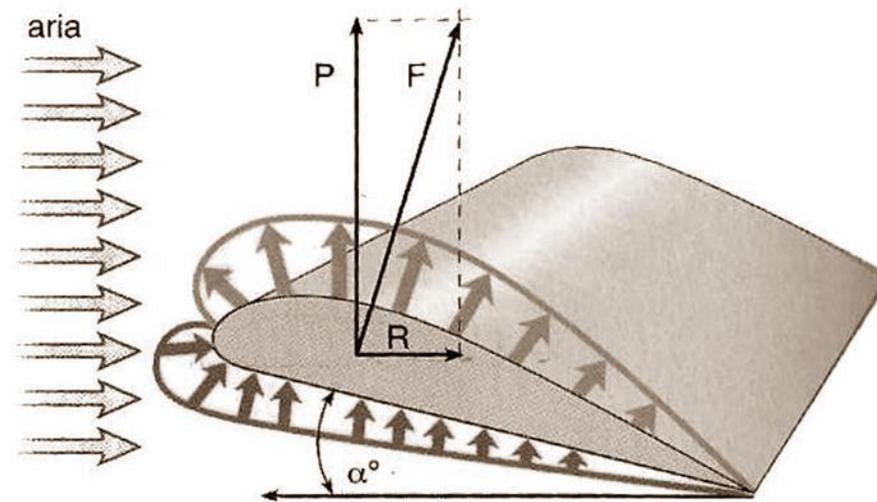
Stato attuale della tecnologia neurale

Non conoscendo la fisica alla base del volo si pensava che il principio fisico a fondamento del volo fosse lo sbattere le ali



Stato attuale della tecnologia neurale

Poi si comprese che il principio fisico che permette il volo non è lo sbattere le ali ma la «portanza».

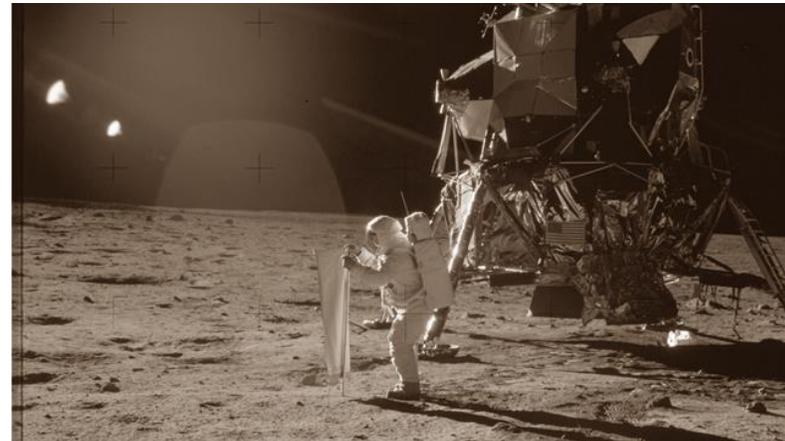


Stato attuale della tecnologia neurale

Lo sbattere le ali è uno dei possibili modi per creare portanza. E' semplicemente il modo seguito dalla biologia, ma ora sappiamo che è poco efficiente e difficile da ingegnerizzare.

Stato attuale della tecnologia neurale

Compresa la fisica del volo, nasce il volo umano e in pochi decenni l'uomo raggiunge la Luna.



Stato attuale della tecnologia neurale

Attualmente, allo stesso modo, sapendo che gli uomini sono «intelligenti», si cerca di replicare l'intelligenza umana simulando le connessioni neurali. Ovvero ingegnerizzando reti neurali artificiali.



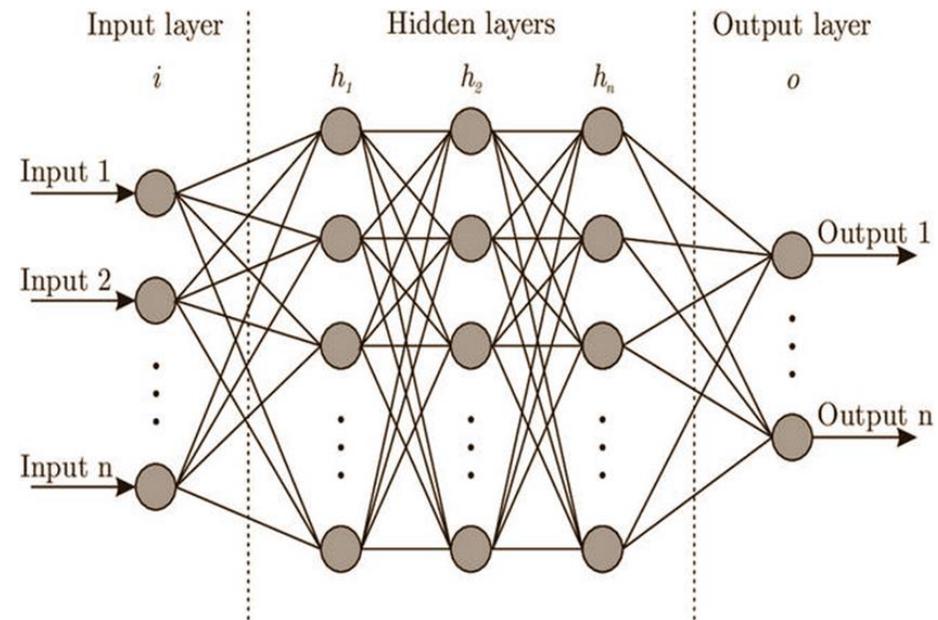
Stato attuale della tecnologia neurale

In altre parole,
l'intelligenza
artificiale sta ancora
all'ornitottero.



A questo punto prendono avvio gli studi e le soluzioni di Neural Research, per cercare di dimostrare che i neuroni e le connessioni sinaptiche non sono il principio dell'intelligenza. Il connettoma neurale è il modo attraverso il quale la biologia ha realizzato l'intelligenza, ma poco efficiente e difficile da ingegnerizzare.

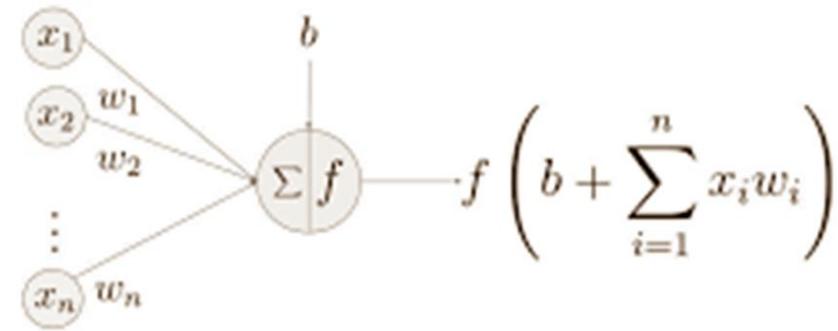
Che cos'è una rete neurale artificiale?



Una rete neurale artificiale è un sistema fortemente interconnesso dove l'informazione, proveniente dagli input, viene distribuita nelle unità interne attraverso il principio della somma pesata. Principio alla base del neurone matematico elaborato da McCulloch e Pitts nel 1943.

Le innovazioni della Neural Research srl

Dalla seguente formulazione del neurone matematico si nota facilmente che la somma pesata altro non è che una equazione della retta. Le reti neurali creano rette, piani, iperpiani che tagliano lo spazio matematico degli input . Tagliando questi spazi, separano e discriminano patterns.



An example of a neuron showing the input $(x_1 - x_n)$, their corresponding weights $(w_1 - w_n)$, a bias (b) and the activation function f applied to the weighted sum of the inputs.

Le innovazioni della Neural Research srl

In fase di apprendimento i pesi della rete vengono modificati e di conseguenza vengono modificate le inclinazioni di queste rette, piani, iperpiani, sino a renderle idonee ad ottenere l'output desiderato.

A nostro avviso il principio della somma pesata è il modo attraverso il quale la biologia struttura la propria intelligenza, ma cercare di simulare ed ingegnerizzare questo principio è come tentare di far sbattere le ali ad un ornitottero per farlo volare. Non è il miglior modo per volare.

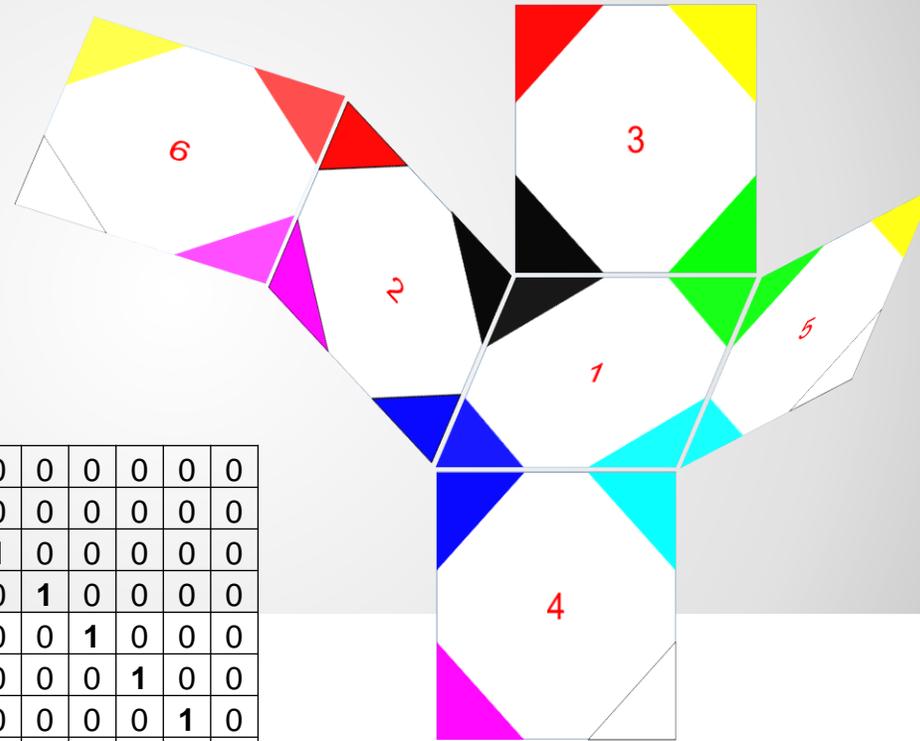
Le innovazioni della Neural Research srl

Per descrivere meglio questo principio delle reti neurali artificiali riporto, nelle due slide seguenti, l'esempio di una rete strutturata, su input RGB, per discriminare i colori fondamentali.

Si nota che nello spazio matematico RGB la rete impara a discriminare i colori fondamentali tagliando, con otto piani, gli otto spigoli del cubo RGB corrispondenti agli otto colori fondamentali.

Naturalmente utilizzando reti più complesse, in deep learning si possono ottenere più tagli e quindi applicazioni più complesse.

APPRENDIMENTO NEURALE DEI COLORI FONDAMENTALI

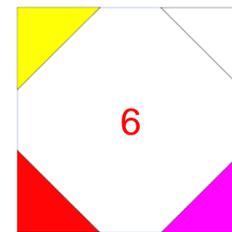
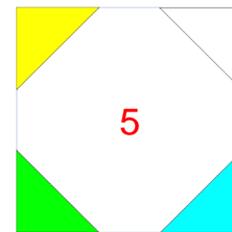
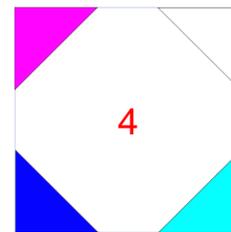
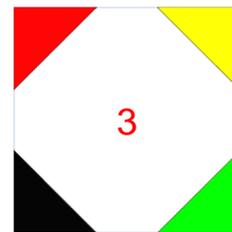
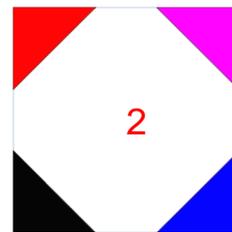
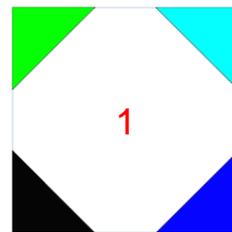


R G B

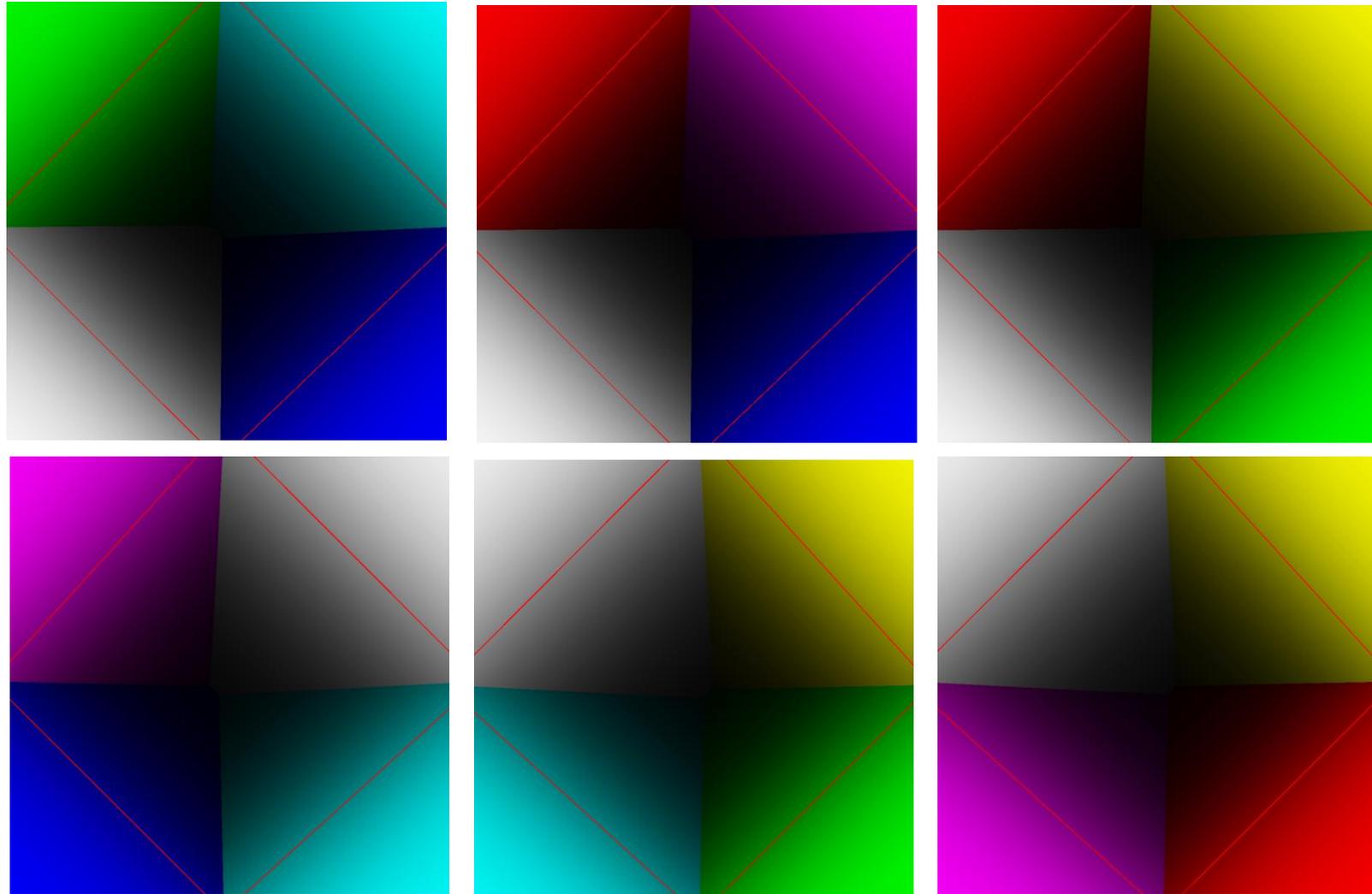
BLACK
WHITE
RED
GREEN
BLU
YELLOW
MAGENTA
CYAN

0	0	0
1	1	1
1	0	0
0	1	0
0	0	1
1	1	0
1	0	1
0	1	1

1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1



Rete neurale sviluppata dalla Neural Research nei propri sistemi di visione artificiale. Si nota come la rete attua i suoi tagli in corrispondenza degli spigoli



Le innovazioni della Neural Research srl

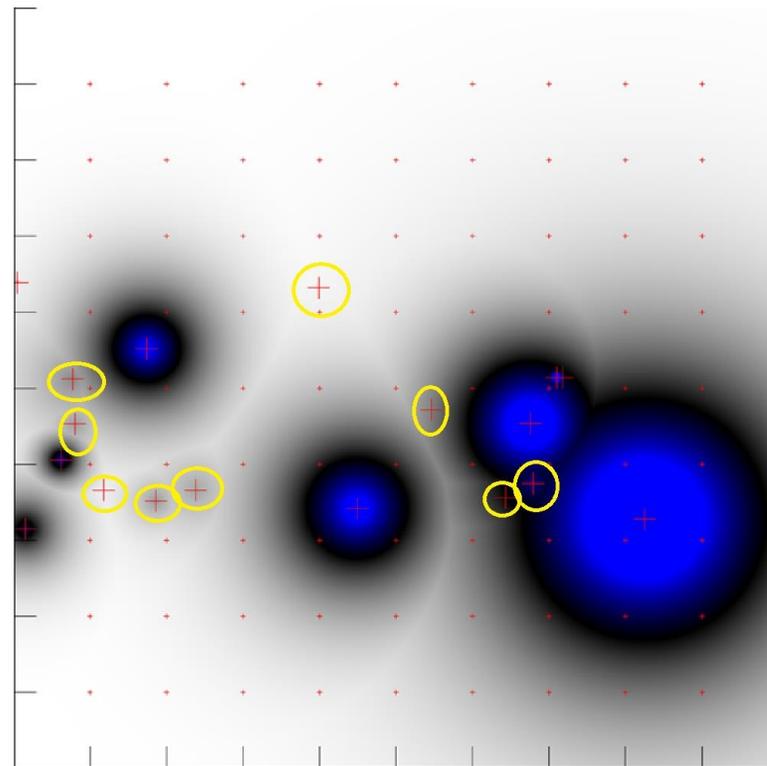
Accanto alle applicazioni dei classici algoritmi di simulazione neurale, la Neural Research ha sviluppato dei propri paradigmi d'intelligenza artificiale.

Mutando prospettiva, prima viene definito uno spazio matematico adeguato a contenere il «concetto» analizzato e poi attraverso l'utilizzo di «fattori di generalizzazione» (non spiegati in queste slide) viene attuato il machine learning.

I risultati di questi nuovi paradigmi d'Intelligenza Artificiale sono incredibilmente potenti.

Le innovazioni della Neural Research srl

A fianco viene riportato un esempio applicativo di questi nuovi algoritmi.
Sistema predittivo utilizzato nell'analisi di librerie molecolari per individuare molecole farmacologicamente interessanti.



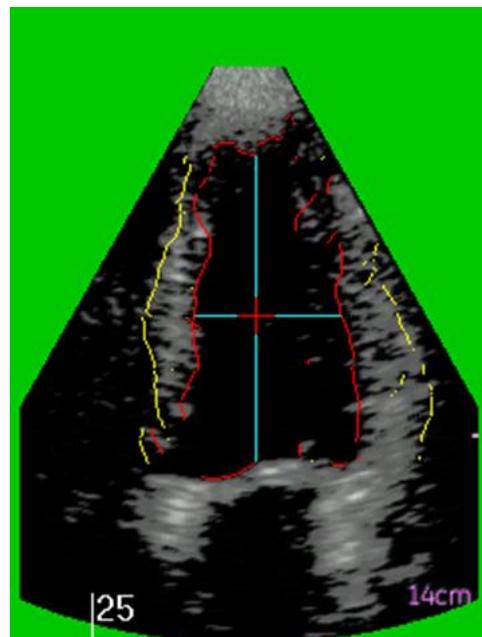
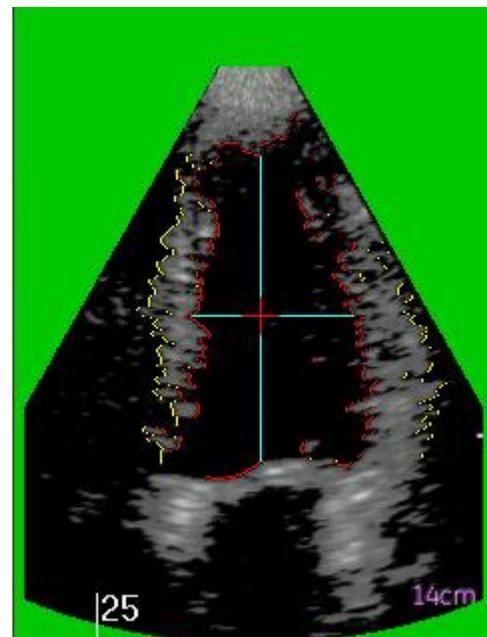
Le innovazioni della Neural Research srl

La Neural Research ha fissato questi nuovi algoritmi d'Intelligenza Artificiale in una libreria c/c++ chiamata Nlib, condivisibile con gruppi di lavoro estesi ed altre società.

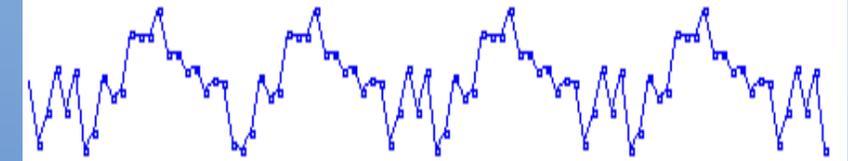
Le Nlib sono state progettate per sviluppare progetti innovativi e a basso livello.



A seguire verranno riportati alcuni esempi di successo nei quali sono state applicate le librerie Nlib.



LVD 36 mm	LVEDV 137 mL/m ²
LVEDD 42 mm	LVESV 47 mL/m ²
LVESD 25 mm	LVEF 65.6566 %



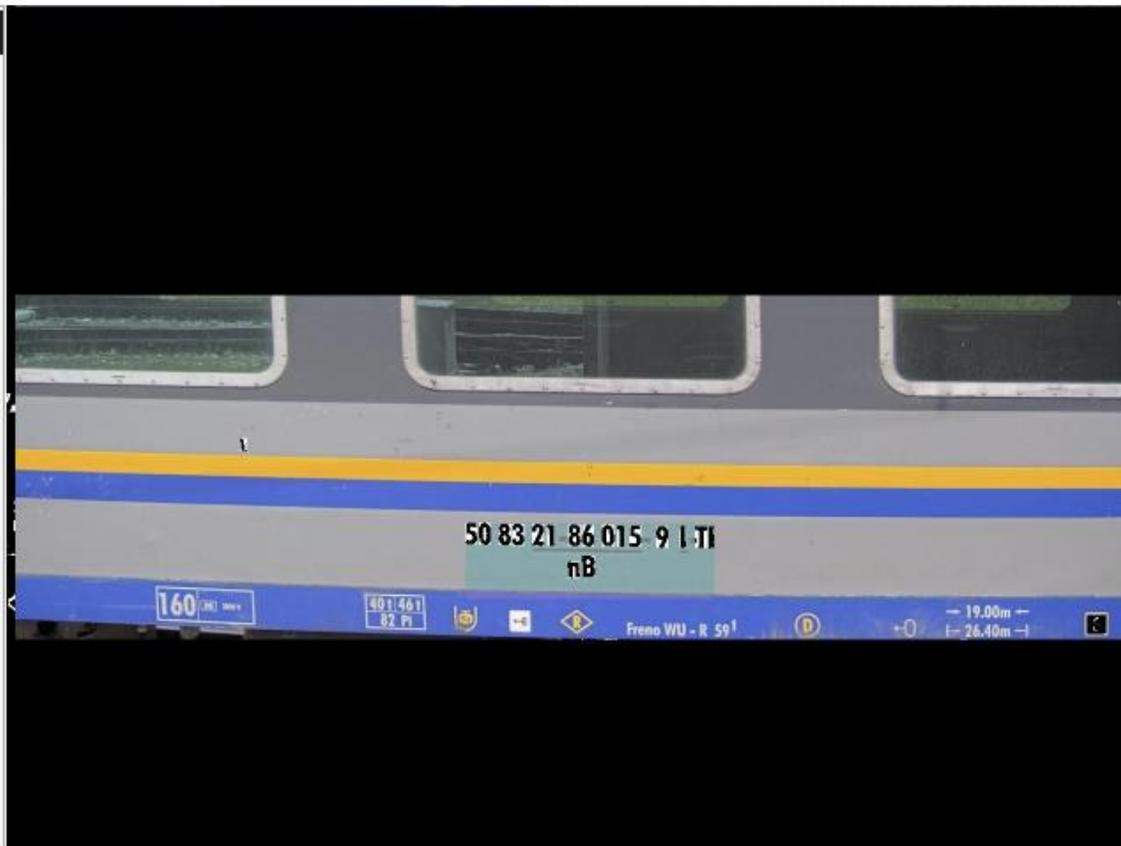
LVD 36 mm	LVEDV 143 mL/m ²
LVEDD 43 mm	LVESV 49 mL/m ²
LVESD 22 mm	LVEF 65.6566 %



**Project**

Create
Open/Close →
Develop...

Monitoring
Learning
Tools
Viewer
Database
Statistics
Home automation
Traffic index

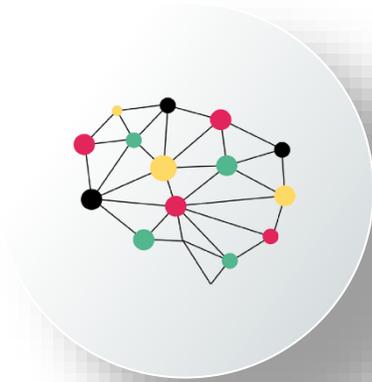


```

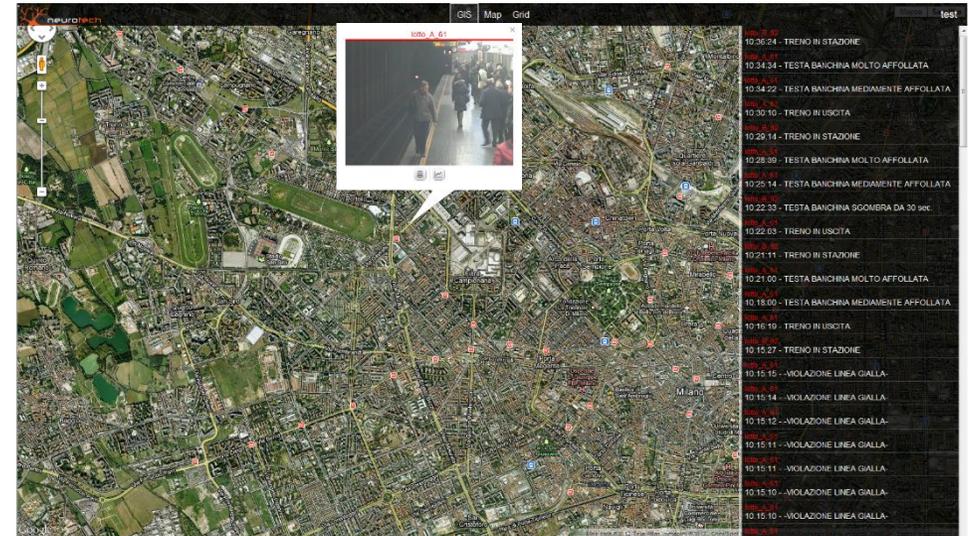
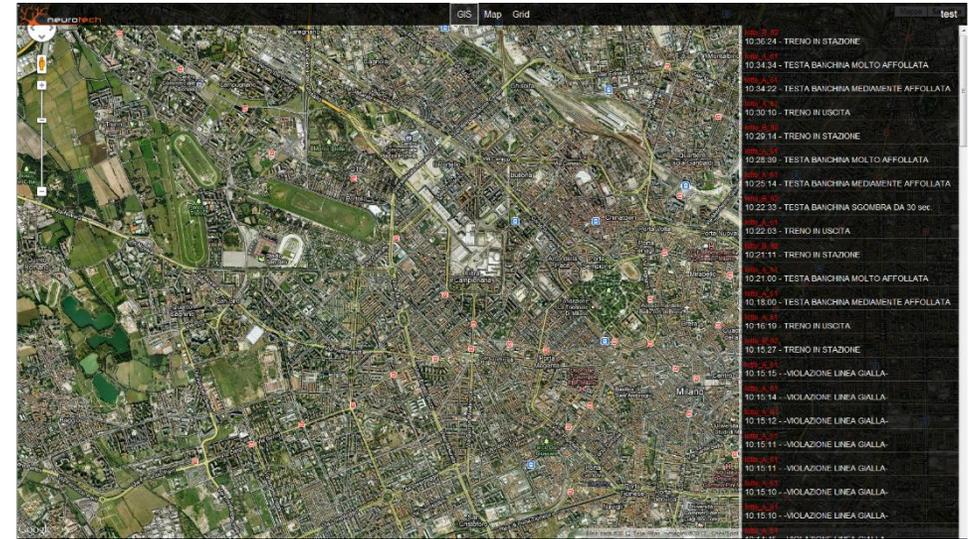
10:52:32 MODE2_READ: 508321860159ITInB
10:52:32 MODE2_READ: 508321860159ITInB (i:0-d: 2)
10:52:32 MODE2_START: 0, 17, 1
10:52:32 MODE2_START: 0, 17, 1
10:52:27 MODE2_READ: 508321860159ITInB (i:0-d: 2)
10:52:27 MODE2_START: 1, 17, 0
10:52:27 MODE2_START: 1, 17, 0
10:52:27 MODE2_READ: 50832186-159ITIBB (i:0-d: 2)
10:52:27 MODE2_START: 1, 17, 0
10:52:27 MODE2_START: 1, 17, 0
10:52:26 MODE2_READ: 80832186-159ITInB (i:0-d: 2)
10:52:26 MODE2_START: 1, 17, 0
10:52:26 MODE2_START: 1, 17, 0
10:52:26 MODE2_READ: 508321-60159-TIn- (i:0-d: 2)
10:52:26 MODE2_START: 1, 17, 0

```

Integrated Territorial Analysis Neural

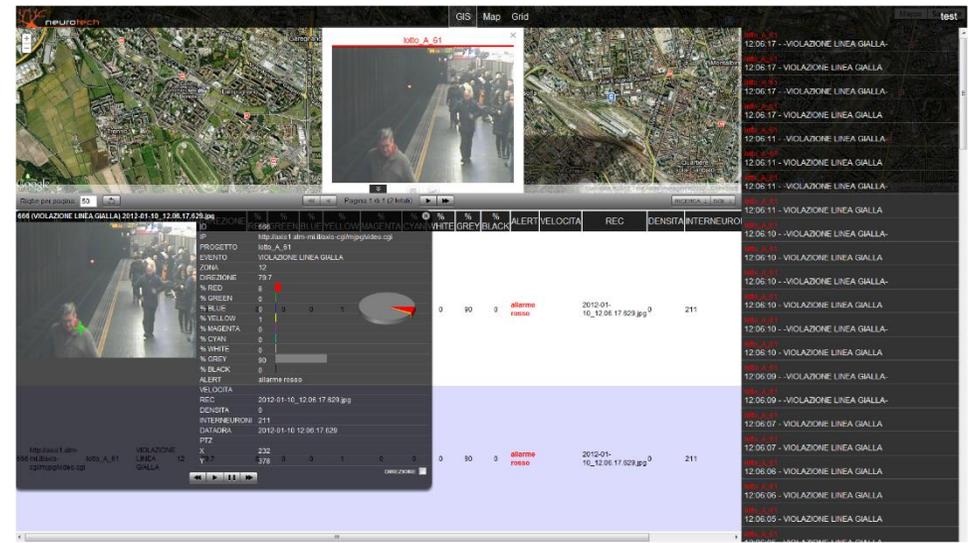
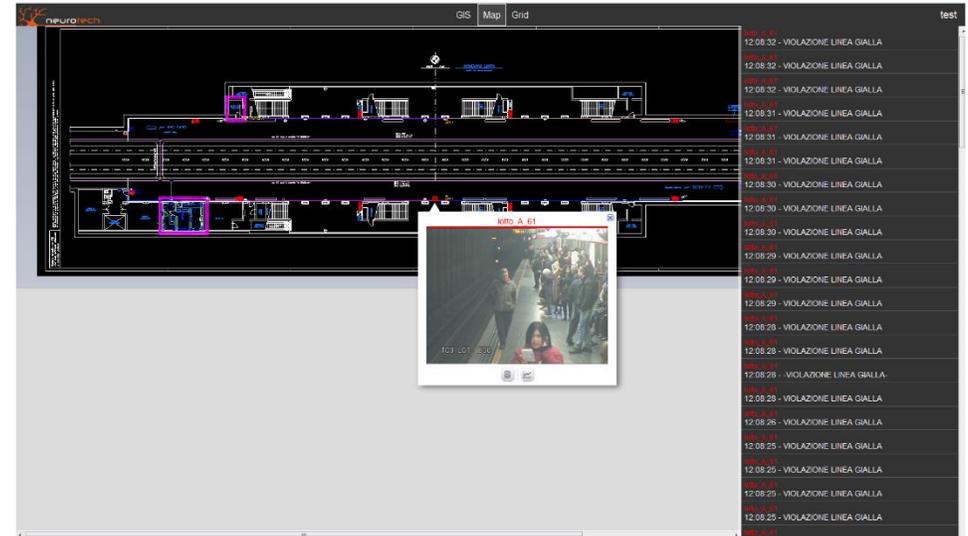


La piattaforma **ITAN** monitora e gestisce il territorio utilizzando algoritmi di simulazione neurale per l'analisi video e dei dati sensoristici. L'approccio neurale di Machine Learning permette di ottenere algoritmi flessibili e robusti ai cambiamenti dell'ambiente analizzato.



Integrated Territorial Analysis Neural

Altro punto innovativo, che distingue nettamente la soluzione Neural Research da altri prodotti di monitoraggio, consiste nel fatto che i singoli “nodi” di analisi del territorio (corrispondenti a scene e regioni ristrette) interagiscono (scambiandosi informazioni) con nodi di monitoraggio contigui o disgiunti. Il territorio viene così analizzato in modalità integrata e il significato di ogni evento, geograficamente localizzato, è comunque determinato dall'analisi di tutto il territorio.



IDpz	ABETA	TAU	PTAU	FDG	AV45	Hippocam	Class
RID_4001	294,7	321,8	26,31	1,11475	1,5072	5532	AD
RID_4009	1340	238	20,15	1,16635	0,9097	5224	AD
RID_4024	934,2	616,4	67,97	1,13657	1,6764	6350	AD
RID_4039	593,6	542,7	57,04	0,759368	1,6386	6471	AD
RID_4029	1063	268,1	25,32	1,37591	1,0829	9311	MCI
RID_4034	486,2	470	46,46	0,987843	1,7252	4831	MCI
RID_4089	751,8	564,1	57,63	1,14074	1,5029	5536	AD
RID_4152	585,9	457,5	43,73	0,787513	1,3765	6696	AD
RID_4153	519,2	405,9	42,01	1,11946	1,2749	5649	AD
RID_4291	762	200,6	18,84	1,178726	1,360148	6691	CN
RID_4292	1635	209,9	19,56	1,536604	1,07255	7754	CN
RID_4308	1700	228,1	20,34	1,196816	1,046849	7888	CN
RID_4320	979,7	252,5	22	1,471985	1,189966	8697	CN
RID_4172	651,4	382	47,67	1,12559	1,0032	7182	AD
RID_4192	564,5	224,6	21,54	1,08677	1,5455	5766	AD
RID_4195	433,9	533,3	55,81	0,907822	1,3832	5948	AD
RID_4209	653,3	409,8	45,45	1,02473	1,5579	7108	AD
RID_4211	456	211,6	20,36	1,14126	1,3954	4981	AD
RID_4215	576,4	257,9	27,32	1,14265	1,6748	6098	AD

```

#include <stdlib.h>
#include <string.h>
#include <stdio.h>

// importo la libreria
#include "NLIB.h"

int main(){
    FILE *ds;

    /*definisco alcune variabili utili a configurare la rete:
    DATASET: path del file contenente il dataset
    ITERATION: numero di iterazioni
    SOURCE_NLIB_ONE: nome del file di apprendimento della rete
    SOURCE_NLIB_TWO: nome del secondo file di apprendimento della rete
    FG: valore che si riferisce al fattore di generalizzazione
    NUM_NEURONS: numero di parametri di input della rete
    */

    char DATASET[] = "csv/DBalzheimer.csv";
    int ITERATION = 519;
    char SOURCE_NLIB_ONE[] = "NET1";
    char SOURCE_NLIB_TWO[] = "NET2";
    int FG = 3;
    int NUM_NEURONS = 6;
    char* line;
    size_t len = 0;

    // carico in memoria l'apprendimento della rete dai due relativi file
    LOAD(SOURCE_NLIB_ONE, SOURCE_NLIB_TWO);

    //configuro il motore di analisi indicando il fattore di generalizzazione e valori massimi che un parametro per un certo indice può assumere
    SET(NUM_NEURONS, 1, 17000, FG);
    SET(NUM_NEURONS, 2, 8516, FG);
    SET(NUM_NEURONS, 3, 9208, FG);
    SET(NUM_NEURONS, 4, 170113, FG);
    SET(NUM_NEURONS, 5, 18570, FG);
    SET(NUM_NEURONS, 6, 10602, FG);

    //esegue in maniera autonoma l'apprendimento del contenuto del dataset
    LEARNING_DS(dataset);

    //salvo l'apprendimento eseguito nel file di rete
    SAVE(SOURCE_NLIB_ONE, SOURCE_NLIB_TWO);

    return 0;
}

```

```

#include <stdlib.h>
#include <string.h>
#include <stdio.h>
// importo la libreria
#include "NLIB.h"

int main(){
FILE *ds;
/*definisco alcune variabili utili a configurare la rete:
  DATASET: path del file contenente il dataset
  ITERATION: numero di iterazioni
  SOURCE_NLIB_ONE: nome del file di apprendimento della rete
  SOURCE_NLIB_TWO: nome del secondo file di apprendimento della rete
  FG: valore che si riferisce al fattore di generalizzazione
  NUM_NEURONS: numero di parametri di input della rete
*/
char DATASET[] = "csv/DBalzheimer.csv";
int ITERATION = 519;
char SOURCE_NLIB_ONE[] = "NET1";
char SOURCE_NLIB_TWO[] = "NET2";
int FG = 3;
int NUM_NEURONS = 6;
char* line;
size_t len = 0;

// carico in memoria l'apprendimento della rete dai due relativi file
LOAD(SOURCE_NLIB_ONE, SOURCE_NLIB_TWO);
//configuro il motore di analisi indicando il fattore di generalizzazione e valori massimi che un parametro per un certo indice può assumere
SET(NUM_NEURONS, 1, 17000, FG);
SET(NUM_NEURONS, 2, 8516, FG);
SET(NUM_NEURONS, 3, 9208, FG);
SET(NUM_NEURONS, 4, 170113, FG);
SET(NUM_NEURONS, 5, 18570, FG);
SET(NUM_NEURONS, 6, 10602, FG);

ds = fopen(DATASET, "r");
if(ds == NULL){
  return -1;
}
char* line;
size_t len = 0;
//leggo dal file il contenuto del dataset
int input[NUM_NEURONS + 1];
for(int i = 0; i < ITERATION; i++){
  ssize_t read = getline(&line, &len, ds);
  if(read == -1) break;
  int index = 0;
  for(char* tok = strtok(line, ";"); tok != NULL; tok = strtok(NULL, ";"), index++){
    input[index] = atoi(tok);
    for(int j = 0; j < NUM_NEURONS; j++){
      //invio i valori di input nella rete
      INPUT(j + 1, input[j]);
    }
    //la rete apprende associando l'input inserito con il relativo evento con cui si vuole associarlo valorizzato con un numero intero
    int learn_output = LEARNING(input[NUM_NEURONS]);
  }
}
fclose(ds);
//salvo l'apprendimento eseguito nel file di rete
SAVE(SOURCE_NLIB_ONE, SOURCE_NLIB_TWO);
return 0;
}

```

```

#include <stdlib.h>
#include <string.h>
#include <stdio.h>

// importo la libreria
#include "NLIB.h"

int main(){
    FILE *ds;

    /*definisco alcune variabili utili a configurare la rete:
    DATASET: path del file contenente il dataset
    ITERATION: numero di iterazioni
    SOURCE_NLIB_ONE: nome del file di apprendimento della rete
    SOURCE_NLIB_TWO: nome del secondo file di apprendimento della rete
    FG: valore che si riferisce al fattore di generalizzazione
    NUM_NEURONS: numero di parametri di input della rete
    */

    char DATASET[] = "csv/DBalzheimer.csv";
    int ITERATION = 519;
    char SOURCE_NLIB_ONE[] = "NET1";
    char SOURCE_NLIB_TWO[] = "NET2";
    int FG = 3;
    int NUM_NEURONS = 6;
    char* line;
    size_t len = 0;

    // carico in memoria l'apprendimento della rete dai due relativi file
    LOAD(SOURCE_NLIB_ONE, SOURCE_NLIB_TWO);

    //configuro il motore di analisi indicando il fattore di generalizzazione e valori massimi che un parametro per un certo indice può assumere
    SET(NUM_NEURONS, 1, 17000, FG);
    SET(NUM_NEURONS, 2, 8516, FG);
    SET(NUM_NEURONS, 3, 9208, FG);
    SET(NUM_NEURONS, 4, 170113, FG);
    SET(NUM_NEURONS, 5, 18570, FG);
    SET(NUM_NEURONS, 6, 10602, FG);

    int output[ITERATION];

    //esegue in maniera autonoma l'analisi dei dati presenti nel dataset salvando l'output all'interno dell'array 'output'
    ANALYSIS_DS(dataset, output, ITERATION);

    return 0;
}

```

```

#include <stdlib.h>
#include <string.h>
#include <stdio.h>
// importo la libreria
#include "NLIB.h"

int main(){
    FILE *ds;

    /*definisco alcune variabili utili a configurare la rete:
    DATASET: path del file contenente il dataset
    ITERATION: numero di iterazioni
    SOURCE_NLIB_ONE: nome del file di apprendimento della rete
    SOURCE_NLIB_TWO: nome del secondo file di apprendimento della rete
    FG: valore che si riferisce al fattore di generalizzazione
    NUM_NEURONS: numero di parametri di input della rete
    */
    char DATASET[] = "csv/DBalzheimer.csv";
    int ITERATION = 519;
    char SOURCE_NLIB_ONE[] = "NET1";
    char SOURCE_NLIB_TWO[] = "NET2";
    int FG = 3;
    int NUM_NEURONS = 6;
    char* line;
    size_t len = 0;

    // carico in memoria l'apprendimento della rete dai due relativi file
    LOAD(SOURCE_NLIB_ONE, SOURCE_NLIB_TWO);
    //configuro il motore di analisi indicando il fattore di generalizzazione e valori massimi che un parametro per un certo indice può assumere
    SET(NUM_NEURONS, 1, 17000, FG);
    SET(NUM_NEURONS, 2, 8516, FG);
    SET(NUM_NEURONS, 3, 9208, FG);
    SET(NUM_NEURONS, 4, 170113, FG);
    SET(NUM_NEURONS, 5, 18570, FG);
    SET(NUM_NEURONS, 6, 10602, FG);

    ds = fopen(DATASET, "r");
    if(ds == NULL){
        return -1;
    }
    char* line;
    size_t len = 0;

    //leggo dal file il contenuto del dataset
    int input[NUM_NEURONS + 1];

    for(int i = 0; i < ITERATION; i++){
        ssize_t read = getline(&line, &len, ds);
        if(read == -1) break;
        int index = 0;
        for(char* tok = strtok(line, ";"); tok != NULL; tok = strtok(NULL, ";", index++) input[index] = atoi(tok);

        for(int j = 0; j < NUM_NEURONS; j++){
            INPUT(j + 1, input[j]);
            printf("%d, ", input[j]);
        }
        //eseguo l'analisi dei dati di input e salvo l'output
        int output_analysis = ANALYSIS();
        printf("Valore restituito dall'analisi=%d\n", output_analysis);
    }

    fclose(ds);
    return 0;
}

```

