

## **Indice**

# 1 Analisi dei dati di probabilità

Durante il periodo 20/05 - 24/05 mi sono occupata di analizzare la probabilità che ha un candidato di rispondere correttamente alle domande in fase di test; valutando le relazioni di dipendenza che possono esistere tra più domande e l'impatto che può assumere la fortuna.

## 1.1 Problema in esame

Test, sottoposto ad un candidato durante un colloquio, composto da *domande a tripla risposta multipla*.

Nel suddetta sezione vengono analizzate le relazioni che intercorrono tra due domande, denominate A e B, a seconda se il candidato risulta in grado di rispondervi correttamente o meno.

## 1.2 Caratteristiche degli eventi di coppia

Tipi di eventi trattati:

- **Eventi indipendenti;**
- **Eventi dipendenti:**
  - A e B sono strettamente dipendenti;
  - A implica B.
- **Evento conosciuto ed evento indovinato.**

Struttura usata per rappresentare la probabilità degli eventi di coppia:

$$\begin{array}{c} AB \\ / \backslash \\ A \ B \\ \backslash / \\ Z \end{array}$$

con:

- $AB$  rappresenta la probabilità complessiva dell'evento che si verifica sempre;
- $A$  rappresenta la probabilità che permette il verificarsi di A, ma non di B;

- $B$  rappresenta la probabilità che permette il verificarsi di  $B$ , ma non di  $A$ ;
- $Z$  rappresenta la probabilità a zero, l'impossibilità del verificarsi dell'evento.

### 1.2.1 Eventi indipendenti

$A$  e  $B$  sono due domande la quali risposte sono completamente scorrelate tra di loro.

$$\begin{array}{c}
 P(A)P(B) \\
 / \wedge \\
 P(A)(1 - P(B)) \ (1 - P(A))P(B) \\
 \backslash / \\
 (1 - P(A))(1 - P(B))
 \end{array}$$

**Considerazioni generali** La probabilità complessiva nel caso di domande indipendenti  $A$  e  $B$  viene data da  $P(A)$  per  $P(B)$ .

Se è conosciuta dal candidato la risposta alla domanda  $A$  ma non alla domanda  $B$  la probabilità di ottenere una risposta corretta è  $P(A)$ , mentre la probabilità di ottenere una risposta non corretta per  $B$  vale  $1 - P(B)$ . Il ragionamento duale è svolto nel calcolo della probabilità per la risposta corretta alla domanda  $B$  ma non ad  $A$ .

La probabilità di non ottenere alcuna risposta corretta alle due domande viene calcolata prendendo in considerazione gli eventi contrari a quelli coinvolti. Dunque per  $A$  la probabilità che il candidato non conosca la soluzione è  $1 - P(B)$ , dualmente per  $B$  la probabilità è  $1 - P(A)$ .

### 1.2.2 Eventi dipendenti

**A e B sono due domande fortemente correlate tra di loro** se si risponde correttamente ad una delle due domande si risponde correttamente anche all'altra.

$$\begin{array}{c}
 P(A)^2 \\
 / \wedge \\
 0 \ 0 \\
 \backslash / \\
 (1 - P(A))^2
 \end{array}$$

**Considerazioni generali** La probabilità complessiva nel caso di domande dipendenti A e B viene data da  $P(A)$  per  $P(B)$ ; ma  $P(A) = P(B)$  dunque  $P(A)^2 = P(B)^2$ .

Conseguentemente se il candidato non conosce la risposta alla domanda A non può conoscere la risposta alla domanda B perciò la probabilità di conoscere uno dei due eventi è pari a 0.

In questo caso la probabilità a 0 è  $(1 - P(A))(1 - P(B)) = (1 - P(A))^2$  essendo che A=B.

**A implica B** Se si sa rispondere alla domanda A di conseguenza si è in grado di rispondere anche alla domanda B.

Tuttavia non vale il ragionamento opposto, se si sa rispondere alla domanda B non significa che si è in grado di rispondere alla domanda A.

$$\begin{array}{c} P(A) \\ /\backslash \\ 0 \quad P(B) - P(A) \\ \backslash / \\ 1 - P(B) \end{array}$$

**Considerazioni generali** La probabilità complessiva nel caso di domande dipendenti A e B viene data esclusivamente da  $P(A)$  in quanto la conoscenza di sia di A che di B è possibile solo se si ha piena conoscenza di A.

Dunque la probabilità che si conosca la risposta alla domanda A ma non a B è impossibile (pari a 0); mentre se si ha conoscenza della domanda B ma non di A la probabilità si stanzia a  $P(B) - P(A)$ .

La probabilità a zero è  $1 - P(B)$  indicatore dell'impossibilità di avere la risposta corretta per A.

### 1.3 Evento conosciuto ed evento indovinato

Durante un test il candidato deve saper scegliere la risposta, corretta o meno, alla domanda posta. Le variabili che entrano in gioco durante l'esecuzione dell'atto non riguardano esclusivamente la conoscenza personale del singolo. La probabilità di un evento A è data dalla formula:

$$P(A) = P(A_C) + P(A_I)$$

Le variabili in uso sono:

- $P(A_C)$ : probabilità che il candidato sappia rispondere alla domanda A correttamente per sua conoscenza;
- $P(A_I)$ : probabilità che il candidato sappia rispondere alla domanda A correttamente indovinando.

Per quanto appena definito sopra valgono le seguenti proprietà:

1.  $P(B_C|A_C) = 1$
2.  $P(B_C|A_I) = P(B_C)$
3.  $P(B_I|A_C) = 0$
4.  $P(B_I|A_I) = P(B_I)$

### 1.3.1 Probabilità di rispondere correttamente ad una domanda

Variabili coinvolte:

- $P(A)$ : probabilità necessaria perchè si verifichi, per la domanda A, che il candidato dia la risposta corretta. Per la legge dei grandi numeri la frequenza porta alla probabilità.
- $S_0$ : insieme dei casi in cui in una domanda non viene scartata alcuna risposta dal dominio delle risposte possibili;
- $S_1$ : insieme dei casi in cui in una domanda viene scartata una risposta dal dominio delle risposte possibili;
- $S_2$ : insieme dei casi in cui in una domanda vengono scartate due risposte dal dominio delle risposte possibili.
- $P(I)$ : probabilità di dare la risposta corretta alla domanda A indovinando;
- $P(C)$ : probabilità di dare la risposta corretta alla domanda A per conoscenza.

Sapendo che  $P(I) = P(A) - P(C)$  logicamente vale anche  $P(A) = P(I) + P(C)$ .

Se un candidato non è in grado scartare alcuna risposta dalla domanda ha 1 possibilità su 3 di, indovinando, dare la risposta corretta. Se un candidato invece risulta in grado di scartare una risposta sbagliata alla domanda, rimane con 1 possibilità su 2 di poter dare la risposta corretta. Se invece, caso ottimo, il candidato ha piena conoscenza della domanda posta risulta in grado di scartare due risposte sbagliate lasciando un'unica risposta possibile, quella esatta. Il ragionamento sopra espresso può venire espresso con la seguente espressione:

$$P(A) = P(S_0)\frac{1}{3} + P(S_1)\frac{1}{2} + P(S_2)$$

Ora individuiamo quale è la probabilità effettiva per un candidato di dare la risposta corretta ad una domanda A.

$$\begin{aligned} 1 &= S_0 + S_1 + S_2 \\ S_0 &= 1 - S_1 - S_2 \end{aligned}$$

Sostituendo:

$$\begin{aligned} P(A) &= (1 - P(S_1) - P(S_2))\frac{1}{3} + P(S_1)\frac{1}{2} + P(S_2) \\ &= \frac{1}{3} - \frac{1}{3}P(S_1) - \frac{1}{3}P(S_2) + \frac{1}{2}P(S_1) + P(S_2) \\ &= \frac{1}{3} + \frac{1}{6}P(S_1) + \frac{2}{3}P(S_2) \end{aligned}$$

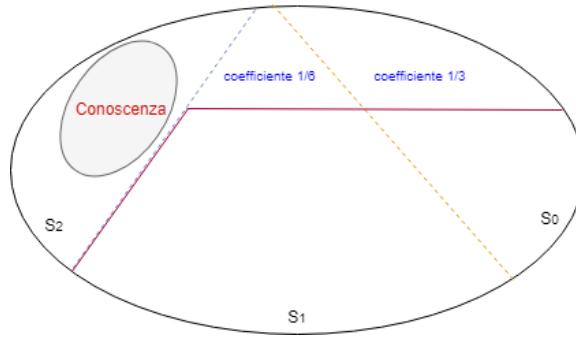


Figura 1: Rappresentazione insiemistica della probabilità di rispondere correttamente ad una domanda:  $P(A)$

### Considerazioni importanti

In conclusione  $P(A) = \frac{1}{3} + \frac{1}{6}P(S_1) + \frac{2}{3}P(S_2)$ . Ovvero la probabilità per un candidato di dare in una domanda A la risposta corretta dipende dai seguenti fattori:

- $\frac{1}{3}$ : coefficiente che rappresenta la probabilità effettiva per chi non conosce la risposta alla domanda di dare la risposta corretta;
- $\frac{1}{2}P(S_1)$ : coefficiente che rappresenta la probabilità effettiva di dare la risposta corretta quando il candidato è in grado di scartare una risposta sbagliata alla domanda;
- $\frac{2}{3}P(S_2)$ : coefficiente che rappresenta la probabilità effettiva di dare la risposta corretta quando il candidato è in grado di scartare due risposte sbagliate alla domanda.

Dall'analisi della tipologia di eventi di coppia e dal calcolo della probabilità necessaria per poter rispondere correttamente ad una domanda, si giunge alla valenza dei seguenti assiomi:

1. Le coppie di domande A e B devono essere fra loro disgiunte, altrimenti si genererebbero situazioni di invalidità dei risultati;
2. Per rispondere correttamente ad una domanda non è necessario che il candidato abbia piena conoscenza di tutti gli argomenti richiesti dall'esame, ma bensì ne risultano sufficienti  $n - 1$ ;
3. La probabilità di conoscere è contenuta all'interno di  $S_2$ , in quanto se un candidato conosce è conseguentemente in grado, da una domanda, di scartare due risposte sbagliate.

### 1.3.2 Il piano

La probabilità  $P(A)$  che un candidato ha in gioco nel momento in cui si appoggia a rispondere ad una domanda può venire rappresentata in un piano.

Di seguito viene mostrata l'immagine di un modellino, rappresentativo di  $P(A)$ , realizzato durante l'analisi.

TODO: foto modello

Ognuno dei tre assi cartesiani rappresenta l'insieme dei casi di scarto ( $S_0$ ,  $S_1$ ,  $S_2$ ). L'intersezione tra i punti del piano indica la regione accettabile contenente il range di valori assumibili da  $P(A)$ . Tale punto proiettato su ognuno dei tre assi permette l'individuazione esatta dei coefficienti delle variabili  $S_0$ ,  $S_1$ ,  $S_2$ .

Ogni porzione della regione del piano viene individuata con la seguente tecnica:

1. Per individuare ogni retta passante per  $S_0$ ,  $S_1$  e  $S_2$  è necessario assumere che  $S_0 + S_1 + S_2 = 1$ ;
2. La retta passante per  $S_0$  è rappresentabile per mezzo delle seguenti equazioni:

$$S_0 = 0 \text{ e } S_1 + S_2 = 1$$

In questo modo l'asse  $S_0$  è fissato a 0 e estrapolando  $S_1$  e  $S_2$  da  $S_1 = -S_2 + 1$  assumono valori tra (0,1).

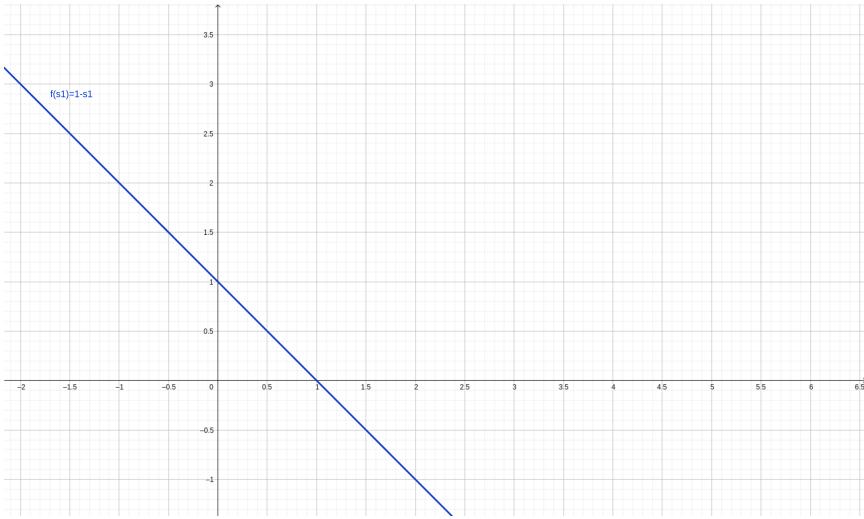


Figura 2: Rappresentazione della retta passante per  $S_0 = 0$

3. Il medesimo ragionamento vale per le rette passanti per  $S_1$  e  $S_2$ .

$$S_1 = 0 \text{ e } S_0 + S_2 = 1$$

l'asse  $S_1$  è fissato a 0 e  $S_0$  e  $S_2$  assumono valori tra (0,1).

$$S_2 = 0 \text{ e } S_1 + S_0 = 1$$

l'asse  $S_2$  è fissato a 0 e  $S_0$  e  $S_2$  assumono valori tra (0,1).

4. In questo modo l'unione di tutte le rette passanti per gli assi creano la regione accettabile dei valori di  $P(A)$ .

Avendo rappresentato il piano si ottiene nei punti di intersezioni fra le tre rette la regione accettabile per  $P(A)$ . Inoltre è possibile, ora, individuare il fascio di rette che tangenti il piano permettono di affermare se una specifica domanda è, in base alla sua frequenza, ha difficoltà bassa, media, alta per un candidato.

- Se una domanda ha una difficoltà bassa la retta si situa passante per i punti  $0 < S_2 \leq 1$  (molto vicino a 1) e  $(S_0, S_1) < 0$  (tendenti a 0);
- Se una domanda ha una difficoltà alta la retta si situa passante per i punti  $S_2 \leq 0$  (molto vicino a 0),  $S_1 < 1$  e  $S_0 \leq 1$  (tendente a non scartare alcuna risposta);

- Se una domanda ha una difficoltà media la retta si situa nella parte centrale della regione accettabile, passante per i punti  $0 \leq (S_0, S_1, S_2) \leq 1$ .

### Rappresentazione di $P(A)$

Vediamo alcuni casi di come le domande possono venire rappresentate sul piano:

La funzione di partenza è:

$$F = \frac{1}{3} + \frac{1}{6}S_1 + \frac{2}{3}S_2$$

Va esplicitato  $S_1$ , i passaggi utili da fare sono i seguenti:

$$\frac{-1}{6}S_1 = \frac{1}{3} + \frac{2}{3}S_2 - F \rightarrow S_1 = -4S_2 - 2 + 6F$$

Essendo che  $0 \leq S_2 \leq 1$  usando  $S_1 = 1$  e  $S_2 = 0$  si ottiene che  $F = \frac{1}{2} = 0.5$

Quanto appena calcolato può venire rappresentato graficamente impiegando la retta  $S_1 = 1 - S_2$  (responsabile di definire una porzione del piano in base alle variaibili coinvolte) e mediante la retta  $S_1 = -4S_2 - 2 + 6F$  (che permette di calcolare il fascio di rette tangenti alla prima retta).



Figura 3: Rappresentazione di  $P(A)$  per una frequenza 0.5 proiettata su assi  $S_0 = 0$ ,  $S_1$  e  $S_2$ .

Nella figura sopra sono rappresentati i seguenti significati:

- La linea azzurra rappresenta  $S_2 = 1 - S_1$ ;
- La linea rosa rappresenta la retta tangente  $S_1 = -4S_2 + 1$ ;
- Punto A (blu):

$$\begin{aligned} S_1 &= 0.5 = \frac{1}{2} \\ \frac{1}{2} &= -4S_2 + 1 \rightarrow 4S_2 = 1 - \frac{1}{2} \rightarrow S_2 = \frac{1}{8} \\ S_0 &= 1 - \frac{1}{2} - \frac{1}{8} = \frac{3}{8} \end{aligned}$$

Ovvero metà dei candidati sottoposti alla domanda sa scartare una delle risposte, lo 0.16% sa dare la risposta corretta e lo 0.36% non sa scartare alcune delle risposte possibili.

- Punto B (verde):

$$\begin{aligned} S_1 &= 0 \\ S_2 &= \frac{1}{4} \\ S_0 &= 1 - \frac{1}{4} = \frac{3}{4} \end{aligned}$$

Ovvero nessun dei candidati sottoposti alla domanda sa scartare una delle risposte, lo 0.25% sa dare la risposta corretta e lo 0.75% non sa scartare alcune delle risposte possibili.

- Punto C (fucsia):

$$\begin{aligned} S_1 &= \frac{1}{4} \\ S_2 &= \frac{3}{16} \\ S_0 &= 1 - \frac{1}{4} - \frac{3}{16} = \frac{9}{16} \end{aligned}$$

Ovvero lo 0.25% dei candidati sottoposti alla domanda sa scartare una delle risposte, lo 0.19% sa dare la risposta corretta e lo 0.56% non sa scartare alcune delle risposte possibili.

- Punto D (arancione):

$$\begin{aligned} S_1 &= \frac{1}{3} \\ S_2 &= \frac{1}{3} \\ S_0 &= 1 - \frac{1}{3} - \frac{1}{3} = \frac{1}{3} \end{aligned}$$

Osserviamo che il punto in esame fuoriesce dalla regione delimitata dalla retta tangente di frequenza 0.5 ( $S_1 = -4S_2 + 1$ ). Conseguenza diretta data dall'impossibilità di ottenere una probabilità del 50% sulla domanda con  $\frac{1}{3}$  di candidati che sa scartare 2 risposte,  $\frac{1}{3}$  che ne sa scartare 1 e  $\frac{1}{3}$  nessuna.

Vediamo ulteriori due esempi che permettono di valutare cosa accade nel piano nel caso di una frequenza:

1. Quasi in prossimità di 1;

2. Pari alla soglia minima dell'indovinato.

Il grafico è il seguente:



Figura 4: Rappresentazione di  $P(A)$  per una frequenza 0.33 e 0.8 proiettate su assi  $S_0 = 0$ ,  $S_1$  e  $S_2$ .

- La linea azzurra mostra la retta tangente con frequenza 0.80%. In questa abbiamo calcolato il punto E (giallo):

$$\begin{aligned} S_1 &= 0 \\ S_2 &= \frac{14}{20} \\ S_0 &= 0 \end{aligned}$$

Quasi la totalità dei candidati ha la conoscenza per poter scartare tutte le risposte sbagliate e dare la risposta giusta alla domanda.

- La linea viola mostra la retta tangente con frequenza 0.33%. In questa abbiamo calcolato il punto F (rosa):

$$\begin{aligned} S_1 &= 0 \\ S_2 &= 0 \\ S_0 &= 1 \end{aligned}$$

Ovvero nessuno dei candidati ha la conoscenza per poter scartare né una nè due risposte, percui l'unica possibilità per un candidato di rispondere alla domanda è indovinare. È evidente come se un candidato non sa la risposta ad una domanda ha una probabilità dello 0.33% di poter indovinare la risposta corretta.

## 2 Rete neurale

La libreria utilizzata per sviluppare la Rete neurale è stata *ConvNetJS*. L'aspetto positivo di tale scelta è la semplicità nell'utilizzo del linguaggio javascript e la possibilità di implementare funzioni complesse dotate di interfaccia grafica. L'aspetto negativo riguarda la totale mancanza di mantenibilità della libreria, con scarsa presenza di esempi applicativi oltre alla documentazione ufficiale, che costringono lo sviluppatore ad una ricerca approfondita personale e a continui test per validare i risultati prodotti.

### Questions test - *Prevision Neural Net*

The screenshot displays the 'Prevision Neural Net' application interface. It includes three main sections: 'CONFIGURAZIONE RETE NEURALE' (Configuration), 'OUTPUT DELLA RETE' (Network Output), and 'INSERIMENTO DATI DI ADDESTRAMENTO' (Training Data Insertion).

- CONFIGURAZIONE RETE NEURALE:** Shows the code for defining the neural network structure and training it with SGD:

```
layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:6});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type: 'regression', num_neurons:6});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01, momentum:0.1, batch_size:10, l2_decay:0.001});
```

- OUTPUT DELLA RETE:** Shows the results of a prediction request. It displays the input vector [0,1,0,0,0,0] and the calculated prediction [-0.14178025279470646, -0.01463923787831693, -0.08131295482446743, 0.11790470015357472, 0.14410273493990078, 0.1329311955165462]. It also shows the status of the neural network and the results of a test run.
- INSERIMENTO DATI DI ADDESTRAMENTO:** A section for inserting training data. It includes fields for 'Number of fields (max 9999):' (set to 6), 'Input1' through 'Input6' dropdowns (all set to 0), a 'save' button, and a 'start trainer' button. Below these are dropdowns for 'Vettore di previsione' (set to 0, 1, 0, 0, 0, 0) and a 'prevision trainer' button.

Figura 5: Primo prototipo di interfaccia utente della Rete neurale di prova.

Durante il periodo 24/05 - 31/05 mi sono occupata dello sviluppo di una Rete neurale in grado di ricevere in input un training set di dimensione 6 e di restituire una previsione sui dati di apprendimento ricevuti. Il problema che la rete mira ad analizzare è quello discusso nel precedente capitolo *Analisi dei dati di probabilità*.

Per agevolare l'apprendimento della rete, ed ottenere delle previsioni stabili, ho implementato due metodi di generazione randomica di dati in modo da far apprendere massicciamente la stessa. Il dato prodotto consiste in un vettore di 6 elementi, composto da -1, 0 e 1 con il seguente criterio:

- **-1**: la domanda x è stata posta al candidato che ha risposto in maniera errata;
- **0**: la domanda x non è stata posta al candidato;
- **1**: la domanda x è stata posta al candidato che ha saputo rispondere correttamente.

Il primo metodo sviluppato si occupa di generare un vettore di dati di apprendimento basandosi esclusivamente su come le domande sono interconnesse tra di loro (grazie all'uso di un grafo della conoscenza costruito ad hoc). Il secondo metodo ripropone quanto perseguito dal primo metodo, con il valore aggiunto, di generazione di un profilo randomico di un candidato, che tiene conto della probabilità di risposta ad una domande seguendo la formula  $P(A) = \frac{1}{3} + \frac{1}{6}P(S_1) + \frac{2}{3}P(S_2)$ .

## 2.1 Test effettuati

Quando si parla di domande nel database, si fa riferimento se nel periodo citato alle 89 domande di logica.

Alcune decisioni che ho preso per individuare l'architettura adeguata di rete possono essere così riassunte:

1. Una rete neurale non deve, per fornire dei dati attendibili, possedere un numero di neuroni troppo elevato rispetto al trainset effettuato; altrimenti la previsione ritornerebbe l'identità del vettore di input della stessa, come conseguenza diretta della capacità troppo elevata di immagazzinare dati.
2. I layers, ho deciso, di allenarli mediante tecnica di regressione, che permette l'inserimento in input di una funzione obiettivo e l'ottenimento di un risultato. L'output, anche in virgola mobile è composto di tanti elementi quanti sono i neuroni di regressione dichiarati. Per la mia rete di prova è sufficiente dichiarare 6 neuroni in regressione perché l'output, appunto, che ci si aspetta dal sistema è di 6 elementi.
3. Per costruire un dataset di dati consistente, che consenta alla rete di imparare qualcosa, ho costruito un Grafo della conoscenza con lo scopo di mettere in relazione degli argomenti d'interesse di più domande.

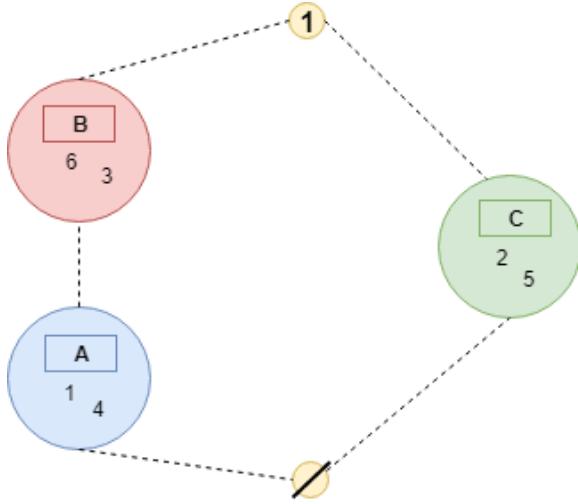


Figura 6: Grafo rappresentante le relazioni esistenti tra il set di domande di prova.

Per svolgere l'apprendimento ogni vettore, facente parte del dataset, viene dato in pasto alla rete, che a sua volta provvede alla sua assimilazione, conseguenza diretta della tecnica dell'autoencoder, ovvero la rete impara il vettore riducendone lo spazio occupato.

4. Per creare il dataset ho ritenuto sufficiente generare *2000* vettori di risposta in modo da compiere in maniera esaustivo l'apprendimento della rete.

La tematica della validità dell'architettura viene approfondita nel dettaglio della sezione ??.

Il vettore passato in input per svolgere le previsioni è  $[0,0,0,0,0,0]$ ,  $[0,0,1,0,1,0]$  e  $[0,0,-1,0,0,0]$

Le aspettative riguardano la previsione di risposta di un candidato . Di seguito riporto quanto è stato rilevato in fase di test.

### 2.1.1 Architettura della rete: 4 neuroni per ciascuno dei 2 layers

Architettura della rete utilizzata:

```
layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:6});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:6});
```

```

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
  momentum:0.1, batch_size:10, l2_decay:0.001});

```

I layers utilizzati sono 2 e composti da 4 neuroni.

### Training set standard a 4 neuroni per ciascuno dei 2 layers

- Il vettore [0,0,0,0,0,0] ha previsione calcolata di [-0.021598804903572744, -0.1372509042342871, 0.06611969158456255, 0.018121335417653706, -0.11264571886853292, 0.17520370837747462]

Appaiono in relazione le domande 1, 2, 5 e 3, 4, 6.

Gli scostamenti tra le coppie 2, 5 e 3, 6 sono consistenti con quelle che sono le relazioni di dipendenza, invece 1, 4 ha una differenza di 0.016 circa che parte da qualche millesimo fino 0.5 Le domande 3 e 6 si dovrebbero presentare con una positività inferiore rispetto a 1 e 4; nel test in analisi questo non viene rispettato da nessuna delle coppie in analisi per differenze che vanno da qualche millesimo fino a 0.018 circa.

### Osservazioni

L'architettura testata si compone di 4 neuroni a layer su una base di 2000 test correndo il rischio di avere una rete che apprende troppo e come effetto negativo "veda" addirittura cose che non esistono. A prova di ciò sono i risultati non conformi alle attese. Dunque mi fermo qui con il test di tale architettura e riducendone il numero di neuroni presenti in ciascun layers e/o il numero di layers presenti.

Le nuove architetture su cui ho effettuato i test sono esposte nei paragrafi seguenti.

#### 2.1.2 Architettura della rete a 2 neuroni per ciascuno dei 2 layers

Architettura della rete utilizzata:

```

layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:6});

```

```

layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:6});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

```

I layers utilizzati sono 2 composti da 2 neuroni.

### **Training set standard su rete a 2 neuroni per ciascuno dei 2 layers**

- Il vettore [0,0,0,0,0,0] ha previsione calcolata di [0.31232372051574936, 0.7253754889487585, -0.5051208979797573, 0.32075742158673093, 0.7324947496336937, -0.4348299972940168] Appaiono in relazione le domande 1, 2, 4, 5 e 3, 6. Gli scostamenti tra le coppie 1, 4 e 3, 6 e 2, 5 sono consistenti con quelle che sono le relazioni di dipendenza fra le domande. Le domande 3 e 6 si dovrebbero presentare con una positività inferiore rispetto a 1 e 4; in questo test la regola viene rispettata pienamente. Dai dati della previsione si nota come il candidato ha una buona probabilità di saper rispondere alla coppia 1 e 4, e ancora più elevata di saper rispondere correttamente alla coppia 2 e 5; molto bassa di saper rispondere correttamente alle 3 e 6 che sono, appunto, di una difficoltà maggiore rispetto alla coppia 1 e 4.
- Il vettore [0,0,1,0,1,0] ha previsione calcolata di [0.5123144717131076, 0.9123354449531641, 0.2837937822420923, 0.46449868699771607, 0.9029832167165894, 0.3227303792035435] Appaiono in relazione le domande 1, 2, 3 4, 5, 6. Gli scostamenti tra le coppie 1, 4 e 3, 6 e 2, 5 sono consistenti con quelle che sono le relazioni di dipendenza fra le domande. Le domande 3 e 6 si dovrebbero presentare con una positività inferiore rispetto a 1 e 4; in questo test la regola viene rispettata pienamente. Dai dati della previsione si nota come il candidato ha un'ottima probabilità di saper rispondere alla coppia 2 e 5 (come imposto dal vettore previsione), buona di saper rispondere alla coppie 3 e 6 (come imposto dal vettore previsione) e più che buona di

saper rispondere alle 1 e 4, che sono di una semplicità più elevata rispetto alla 3 e 4.

- Il vettore  $[0,0,-1,0,0,0]$  ha previsione calcolata di  $[0.3698539826215957, 0.288907514487717, -0.8504159455662308, 0.3663192502433841, 0.2937448801761998, -0.7845589473185985]$

Appaiono in relazione le domande 1, 2, 4, 5 e 3, 6.

Gli scostamenti tra le coppie 1, 4 e 3, 6 e 2, 5 sono consistenti con quelle che sono le relazioni di dipendenza fra le domande. Le domande 3 e 6 si dovrebbero presentare con una positività inferiore rispetto a 1 e 4; in questo test la regola viene rispettata pienamente.

Dai dati della previsione si nota come il candidato ha una discreta probabilità di saper rispondere alla coppia 2 e 5, un pò meglio di saper rispondere alla coppia 1 e 4 e più di non saper saper rispondere alle 3 e 6 (come imposto dal vettore previsione).

#### **Training set con generazione del profilo di un candidato e calcolo delle probabilità di risposta a 2 neuroni per ciascuno dei 2 layers**

- Il vettore  $[0,0,0,0,0,0]$  ha previsione calcolata di  $[0.057781303506280995, 0.0513731100126314, -0.06600467867066256, 0.029940883111932555, -0.019564515397168573, -0.09570617900597932]$

Appaiono in relazione le domande 1, 2, 4 e 3, 5, 6.

Gli scostamenti tra la coppia 1, 4 e 3, 6 sono consistenti con quelle che sono le relazioni di dipendenza fra le domande; invece per la coppia 2 e 5 i segni sono opposti con una differenza di 0.024. Le domande 3 e 6 si dovrebbero presentare con una positività inferiore rispetto a 1 e 4, la regola viene rispettata pienamente. Le anomalie riscontrate sono da ricondurre alla natura del vettore di training che si basa sul calcolo della probabilità di una risposta che sul grafo della conoscenza.

- Il vettore  $[0,0,1,0,1,0]$  ha previsione calcolata di  $[0.19494624113789977, 0.1712744021266377, 0.577963304906936, 0.781098215373483, 0.3774535909060714, 0.03617314870307162]$

Appaiono in relazione le domande 1, 2, 3, 4, 5, 6.

Gli scostamenti tra le coppie 1 e 4, 2 e 5, 3 e 6 sono consistenti con quelle che sono le relazioni di dipendenza fra le domande. Le domande 3 e 6 si dovrebbero presentare con una positività inferiore rispetto a 1 e 4, la regola non viene rispettata dalla domanda 1 in rapporto con la domanda per una differenza di 0.37 circa. Le anomalie riscontrate sono da ricondurre alla natura del vettore di training che si basa sul calcolo della probabilità di una risposta che sul grafo della conoscenza.

- Il vettore [0,0,-1,0,0,0] ha previsione calcolata di [0.09845785763965222,0.015421380649956663,-0.5138068038427066,-0.4853190165287735,-0.22629262719814794,0.0008152164571250502]

Appaiono in relazione le domande 1, 2, 6 e 3, 4, 5.

Gli scostamenti tra le coppie 1, 4 e 2, 5 e 3, 6 per una differenza tuttavia trascurabile che oscilla dallo 0.2 allo 0.5. Le domande 3 e 6 si dovrebbero presentare con una positività inferiore rispetto a 1 e 4, la regola non vale per la coppia 6 e 4. Le anomalie riscontrate sono da ricondurre alla natura del vettore di training che si basa sul calcolo della probabilità di una risposta che sul grafo della conoscenza.

## Osservazioni

Confrontando i risultati ottenuti dalla rete con i layers impostati a 4 neuroni con quanto emerso dai dati risultanti dalla rete a 2 neuroni; emerge come l'architettura a 2 neuroni a layers è sicuramente quella che da i risultati attesi.

Quanto emerso di discordate dal secondo training set è, come da aspettative, da associare alla natura stessa della creazione del set di dati.

### 2.1.3 Architettura della rete a 4 neuroni per 1 layer

Architettura della rete utilizzata:

```
layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:6});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:6});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});
```

Viene utilizzato un unico layer da 4 neuroni.

### Training set standard su rete a 4 neuroni per 1 layer

- Il vettore [0,0,0,0,0,0] ha previsione calcolata di [0.12202628618565468,0.08221724740100582,0.02233631914718809,0.09586625658118901,0.05558075220027264,0.13443779128784109]

Appaiono in relazione le domande 1, 2, 3, 4, 5, 6.

Gli scostamenti tra le coppie 1, 4 e 3, 6 e 2, 5 sono consistenti con quelle che sono le relazioni di dipendenza fra le domande. Le domande 3 e 6 si dovrebbero presentare con una positività inferiore rispetto a 1 e 4; in questo test la regola non viene rispettata dalla domanda 6.

Dai dati della previsione si nota come il candidato non ha una buona probabilità di saper rispondere alle domande e la domanda 6 non si presenta conforme alle aspettative.

## Osservazioni

*Rispetto a quanto osservato nei casi precedenti, ancora l'architettura che rispetta le attese è quella con 2 neuroni per 2 layers.*

Tale conclusione ha perfettamente senso in quanto il Grafo della Conoscenza, che ho usato come base per costruire i vettori di apprendimento, è composto da 3 nodi (A, B, C) indicanti 3 neuroni. Il quarto può venire valutato come un nodo della rete utile per parametri in entrata e in uscita.

Per estendere maggiormente la mia conoscenza della rete, ho provveduto ad aumentare progressivamente il numero di neuroni a layers e osservarne le interazioni. Svolgendo ciò mi sono accorta che il risultato ottenuto dalla previsione era il più possibile vicino al vettore previsione; causa di un numero eccessivo di neuroni dati alla rete per l'apprendimento rispetto al training set svolto. La situazione è generatrice di overfitting e non attendibilità dei dati raccolti. L'architettura a 1 e 2 neuroni invece presenta una buona capacità di previsione in quasi tutti i casi, però tende ad andare in underfitting, come riporto di seguito:

```
Il vettore [0,0,0,0,0,1] ha previsione calcolata di  
[0.5613347853884025, 0.8310670629630683, -1.03049430206139,  
0.5492731069379962, 0.5679700877862532, -0.8637707232817535]
```

Il numero di neuroni non è sufficiente per memorizzare che la domanda 6 deve essere positiva, e comporta a cascata la correttezza anche delle domande 3, 4 e 1. La situazione si presenta simile se il layer con 1 neurone è posto al di sotto.

## 2.2 Sviluppo della rete delle domande nel database

### 2.2.1 Montaggio e configurazione della rete per le domande di logica

Durante la settimana dal 03/06 al 07/06 la mia attività principale è stata il montaggio e la configurazione della Rete neurale, inerente il database aziendale con dataset i colloqui ai candidati per le domande di logica. Inoltre ho

rivolto parte delle ore a modificare e ottimizzare quanto già implementato nella Rete di prova, in modo che ogni cosa implementata sulla rete del database è presente anche in versione ridotta.

Per rendere più comprensibile le previsioni di probabilità ottenute, a seguito dell'addestramento della rete e della data in pasto del vettore previsione, ho realizzato un'immagine canvas in cui ogni domanda viene raffigurata con un quadrattino colorato, in base alla previsione risultante (verde se a 1, bianco a 0, rosso a -1, gradazioni di bianco - verde e bianco - rosso per i valori intermedi).

The screenshot shows a user interface for a neural network. At the top, there is a code editor window titled "CONFIGURAZIONE RETE NEURALE" containing Python code for defining layers and training a neural network:

```

CONFIGURAZIONE RETE NEURALE
layer_defs = []
layer_defs.push({type:'input', out_sz:1, out_sy:1, out_depth: 0});
layer_defs.push({type:'fc', num_neurons:2, activation:'tanh'});
layer_defs.push({type:'fc', num_neurons:2, activation:'tanh'});
layer_defs.push({type:'regression', num_neurons:6});
net = new convnetjs.Net();
net.makeLayers(layer_defs);
trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01, momentum:0.1, batch_size:10, l2_decay:0.001});

```

Below this is another code editor window titled "OUTPUT DELLA RETE" containing JSON data for input vectors and their corresponding outputs:

```

OUTPUT DELLA RETE
create JSON
Inserito vettore di risposta: [1,-1,1,1,-1,1]
Inserito vettore di risposta: [1,-1,1,1,0,1]
Inserito vettore di risposta: [1,1,-1,1,0,-1]
Inserito vettore di risposta: [1,1,-1,1,1,-1]
Inserito vettore di risposta: [1,1,0,1,0,1]
Inserito vettore di risposta: [1,1,0,1,1,1]
Inserito vettore di risposta: [-1,1,-1,-1,-1,-1]
Inserito vettore di risposta: [0,1,1,0,1,-1]
Inserito vettore di risposta: [1,0,1,1,0,1]
Inserito vettore di risposta: [1,0,1,1,1,1]
Inserito vettore di risposta: [0,1,-1,-1,1,-1]
Inserito vettore di risposta: [1,-1,1,1,1,-1]
Inserito vettore di risposta: [1,-1,1,1,1,1]
Inserito vettore di risposta: [1,0,1,1,-1,1]
Richiesta di previsione inoltrata alla rete ...
Il vettore [0,0,0,0,0,0] ha previsione calcolata di
[0.44336712317213167, -0.333002413740274]
Rete neurale in attesa ...

```

At the bottom of the interface, there is a "INSERIMENTO DATI DI PREVISIONE" section with a dropdown menu for selecting a "Vettore di previsione" and a button labeled "prevision trainer". Below the dropdown are several colored squares (green, red, white) representing the output categories.

Figura 7: Rete di prova dopo lo sviluppo del canvas per le previsioni.

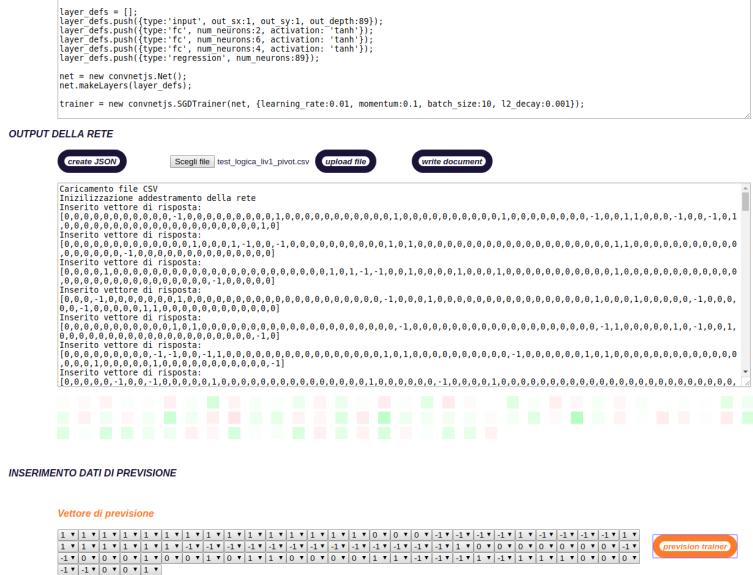


Figura 8: Rete neurale del database aziendale.

Una prima architettura su cui ho deciso di analizzare i risultati della rete del database a 89 domande di logica, basandomi anche sul quanto appreso dalla Rete neurale di prova e dal numero di vettori di test utilizzati (1245 vettori x 89), è stata la seguente:

```

layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:6, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

```

Ho aggiunto un layer e messo un numero di neuroni per layer in modo da formare un romboide.

### 2.2.2 Natura delle domande di logica contenute nel database aziendale

Analizzando il training set dei vettori ho riscontrato tali correlazioni:

- Solo una piccola parte delle domande di logica, presenti nel database, vengono svolte durante un colloquio con un candidato. In media una decina su 89 possibili.
- Dalla rete sembra che le domande abbiano qualche correlazione, tuttavia la configurazione attuale ne rende difficoltosa l'individuazione.

### 2.2.3 Test e Documentazione

Durante la settimana dal 10/06 al 18/06 ho effettuato quanto definito all'interno del Piano di Lavoro come "Test e Documentazione".

#### Test nella Rete di prova

Architettura della rete utilizzata:

```
layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:6});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:6});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});
```

- Il vettore [1,1,1,1,1,1] ha previsione calcolata di  
 $[0.8521066399598267, 0.898137375081856, 0.9993098151218291, 0.792190337086403,$   
 $0.811145866789799, 0.9514731560722426]$

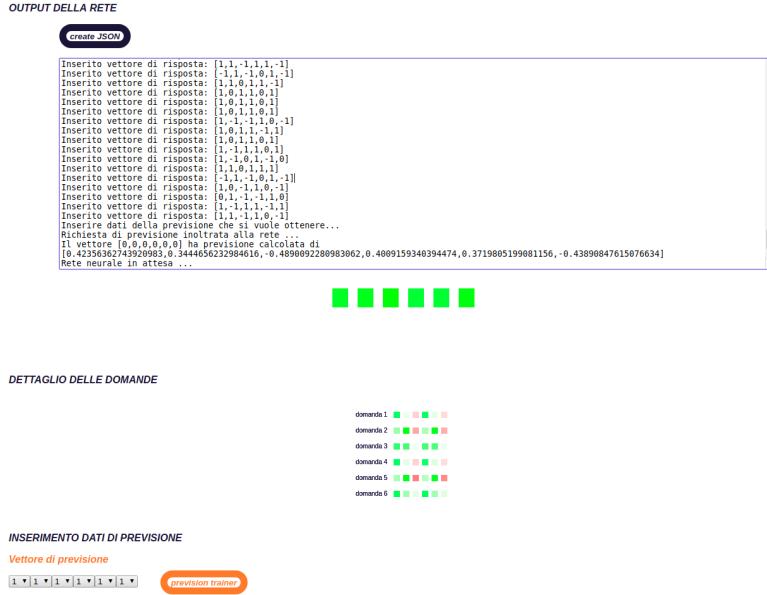


Figura 9: Risultato della rete di prova a seguito di un vettore di previsione [1, 1, 1, 1, 1, 1] in input.

Quanto mostrato dal dettaglio delle domande ha il seguente significato per un candidato:

- se la domanda 1 è settata a 1 (corretta): la rete prevede che la domanda 1 e 4 abbiano una probabilità alta di essere risposte in modo corretto (verde); la 3 e 6 una probabilità non eccessiva di venire risposte in modo sbagliato (rosa attenuato), la 2 e la 5 di non venire nemmeno poste (bianco con qualche minima sfumatura di verde).
- se la domanda 2 è settata a 1 (corretta): la rete prevede che la domanda 1 e 4 abbiano una probabilità non molto alta di essere risposte in modo corretto (bianco con qualche sfumatura di verde); la 3 e 6 una buona probabilità di venire risposte in modo sbagliato (rosa), la 2 e la 5 di venire date in modo corretto (verde).
- se la domanda 3 è settata a 1 (corretta): la rete prevede che la domanda 3 e 6 abbiano una probabilità comunque bassa di essere risposte in modo corretto (bianco con qualche sfumatura di verde); la 1 e 4 con probabilità di venire risposte in modo corretto (verde) perchè più semplici delle domande 3 e 6, la 2 e la 5 di venire risposte correttamente (verde).
- se la domanda 4 è settata a 1 (corretta): la rete prevede un risultato identico a quanto ottenuto dalla domanda 1.

- se la domanda 5 è settata a 1 (corretta): la rete prevede un risultato similare a quanto ottenuto dalla domanda 2. Cambia solo quanto previsto dalle domande 3 e 6 che si presentano con un rosa un pò più intenso, in quanto correlate alla coppia di domande 2 e 5.
- se la domanda 6 è settata a 1 (corretta): la rete prevede un risultato simile a quanto ottenuto dalla domanda 3. La coppia 2 e 5 hanno una probabilità minore di essere date correttamente (bianco con sfumature di verde); ma perchè non correlate alle domande 3 e 6.
- Il vettore  $[-1, -1, -1, -1, -1, -1]$  ha previsione calcolata di  $[0.3440856175367477, -0.5026946644729329, -1.284368009920025, 0.35883842020377565, -0.37844446052773495, -1.1717763012412878]$



Figura 10: Risultato della rete di prova a seguito di un vettore di previsione  $[-1, -1, -1, -1, -1, -1]$  in input.

- se la domanda 1 è settata a -1 (non corretta): la rete prevede che la domanda 1 e 4 non abbiano una probabilità alta di essere risposte in modo corretto (bianco con sfumature di verde); la 3 e 6 una probabilità molto alta di venire risposte in modo sbagliato (rosso), la 2 e la 5 di non venire nemmeno poste (verde con qualche sfumatura di bianco).

- *se la domanda 2 è settata a -1 (non corretta)*: la rete prevede che la domanda 1 e 4 abbiano una probabilità non molto alta di essere risposte in modo non corretto (verde con qualche sfumatura di bianco); la 3 e 6 una buona probabilità di venire risposte in modo sbagliato (rosa), la 2 e la 5 di venire date in modo non corretto (rosa molto attenuato).
- *se la domanda 3 è settata a -1 (non corretta)*: la rete prevede che la domanda 3 e 6 abbiano una probabilità comunque alta di essere risposte in modo non corretto (rosso); la 1 e 4 con bassa probabilità di venire risposte in modo corretto (bianco con qualche sfumatura di verde) perchè più semplici delle domande 3 e 6, la 2 e la 5 di non venire nemmeno poste o comunque basso di venire risposto correttamente(bianco con sfumature di verde).
- *se la domanda 4 è settata a -1 (non corretta)*: la rete prevede un risultato identico a quanto ottenuto dalla domanda 1.
- *se la domanda 5 è settata a -1 (non corretta)*: la rete prevede un risultato similare a quanto ottenuto dalla domanda 2. Cambia solo quanto previsto dalle domande 3 e 6 che si presentano con un rosa un pò meno intenso, in quanto non correlate alla coppia di domande 2 e 5.
- *se la domanda 6 è settata a -1 (non corretta)*: la rete prevede un risultato simile a quanto ottenuto dalla domanda 3. La coppia 2 e 5 hanno una probabilità maggiore di essere date correttamente (bianco con sfumature di verde) perchè non correlate alle domande 3 e 6.

## Test nella Rete del database

### Architetture testate

Durante tutto il periodo ho effettuato una serie di test su molteplici architettura della rete, con gradi di correlazione tra le domande pari al 100% fino ad un differenziale massimo di 5 punti colore, rispetto al canvas risultante per ogni domanda <sup>1</sup>.

Tuttavia non sono, durante questo periodo, riuscita ad individuare un'architettura sufficientemente stabile per prevedere risultati attendibili.

Fattori dominanti, che impediscono la scelta di una configurazione nel caso trattato, sono:

- La molteplicità di dati che hanno aumentato esponenzialmente la complessità di analisi, rispetto alla Rete di prova;

---

<sup>1</sup>valutando i cluster generati

- la creazione stessa del set di input, parzialmente randomica.

I fattori sopra citati vengono trattati approfonditamente nella sezione ??.

Architettura della rete utilizzata:

```
● layer_defs = [];  
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});  
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});  
layer_defs.push({type:'fc', num_neurons:6, activation: 'tanh'});  
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});  
layer_defs.push({type:'regression', num_neurons:89});  
  
net = new convnetjs.Net();  
net.makeLayers(layer_defs);  
  
trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,  
momentum:0.1, batch_size:10, l2_decay:0.001});
```



Figura 11: Risultato della rete del database a seguito di un vettore di previsione [1, 1, 1, 1, 1, 1] in input.

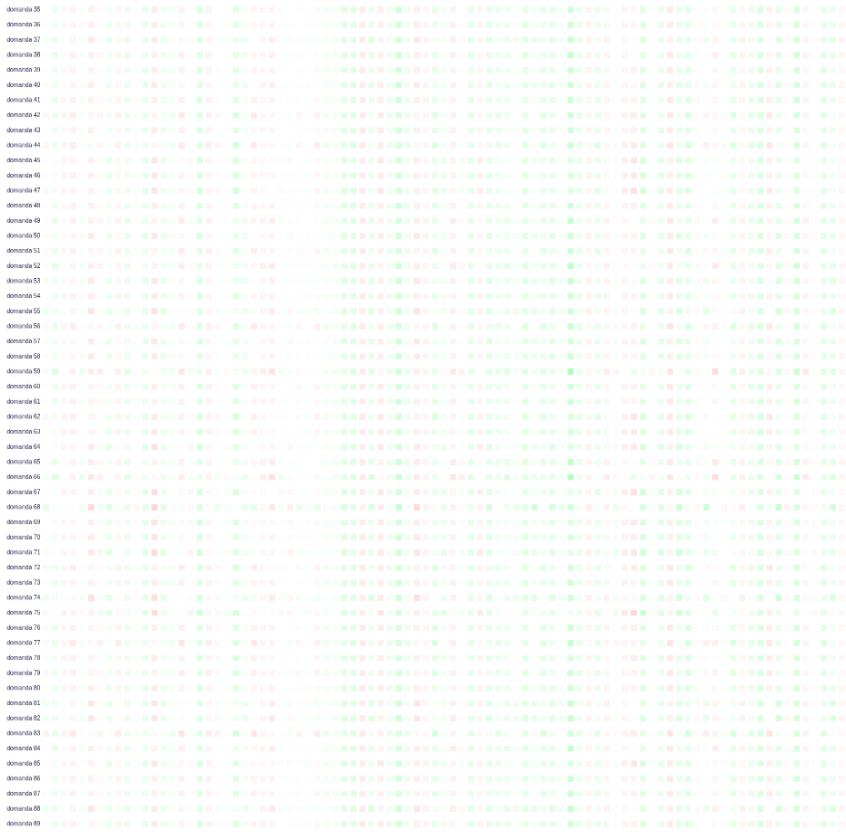


Figura 12: Risultato della rete del database a seguito di un vettore di previsione  $[1, 1, 1, 1, 1, 1]$  in input.

Dagli screen della rete riportati sopra appare come "sembrano" domande:

1. Analisi verticale:

- *semplici* la 18, 22, 34, 35, 37, 39, 41, 44, 48, 50, 51, 52, 53, 54, 55, 56, 57, 59, 60, 67, 69, 71, 72, 77, 79, 80, 82, 84, 87 e 88. Inoltre di queste sembrano in relazione ancora più stretta le domande 18, 22, 40 e 59.
- *difficili* la 3, 4, 6, 13, 16, 19, 24, 25, 26, 36, 38, 42, 43, 46, 49, 61, 63, 65, 66, 70, 78, 81, 85 e 89. Inoltre di queste sembrano in relazione ancora più stretta le domande 6, 13, 19, 36, 38, 42, 46 e 70.

2. Analisi orizzontale: Appaiono in accoppiate le seguenti domande:

- 2, 3, 4, 5;
- 7, 8, 9;
- 14, 16;

- 20, 21;
- 26, 32;
- 29, 32
- 33, 34, 35, 36, 38;
- 39, 41, 43;
- 46, 48;
- 49, 52;
- 50, 53;
- 72, 79;
- 81, 82;
- 86, 87, 88.



Figura 13: Risultato della rete del database a seguito di un vettore di previsione  $[-1, -1, -1, -1, -1, -1]$  in input.

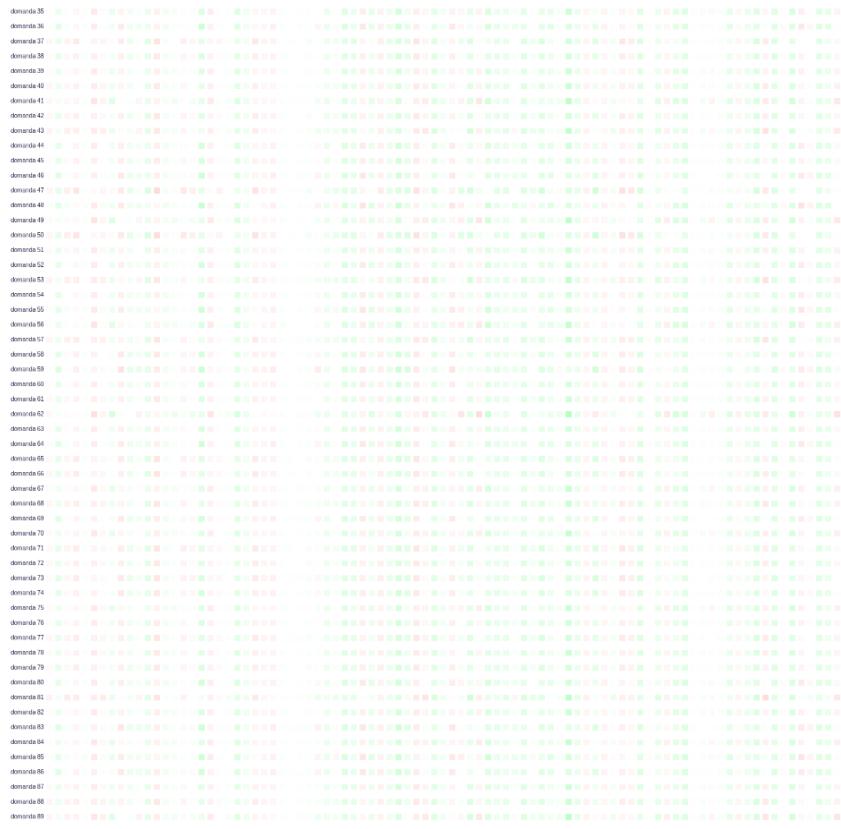


Figura 14: Risultato della rete del database a seguito di un vettore di previsione  $[-1, -1, -1, -1, -1, -1]$  in input.

Dagli screen della rete riportati sopra appare come "sembrano" domande:

1. Analisi verticale:

- *semplici* la 18, 22, 34, 35, 37, 38, 39, 41, 44, 48, 50, 51, 52, 54, 55, 56, 67, 69, 71, 72, 77, 79, 80, 82, 84, 87 e 88. Inoltre di queste sembrano in relazione ancora più stretta le domande 18, 22, 38, 52 e 57.
- *difficili* la 3, 4, 6, 13, 19, 27, 36, 38, 42, 43, 46, 61, 62, 65, 70. Inoltre di queste sembrano in relazione ancora più stretta le domande 6, 13, 19, 36, 38, 42, 46, 61 e 62.

2. Analisi orizzontale: Appaiono accoppiate le seguenti domande:

- rimaste consistenti con il vettore  $[1, 1, 1, 1, 1, 1]$ .

```
• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
```

```

layer_defs.push({type:'fc', num_neurons:12, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01, momentum:0.1
, batch_size:10, l2_decay:0.001});

```

Noto che aumentando il numero di neuroni sull'unico layer esistente, il valore della domanda corrispondente al vettore della previsione sembra sempre più marcato, segno che la rete "impara troppo" e ricade nel restituire l'immagine stessa del vettore previsione.



Figura 15: Risultato della rete del database a seguito di un vettore di previsione [1, 1, 1, 1, 1, 1] in input.

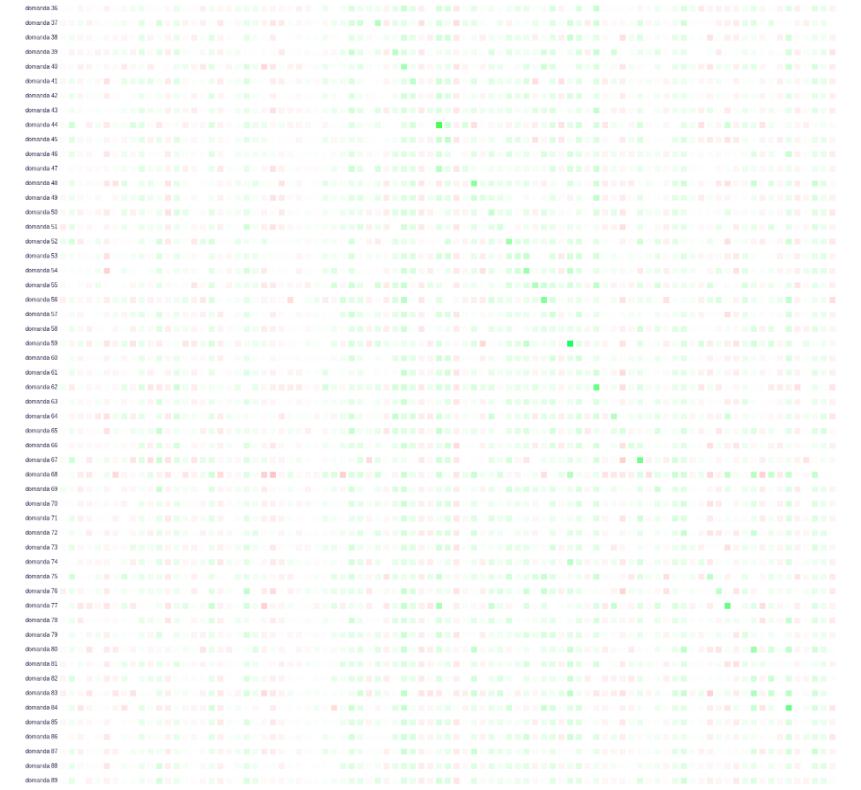


Figura 16: Risultato della rete del database a seguito di un vettore di previsione  $[1, 1, 1, 1, 1, 1]$  in input.

Dagli screen della rete riportati sopra appare come la situazione appare meno lineare del caso analizzato precedentemente. Le domande non vengono separate per linee rete; ma per aree di relazione. Appare come "sembrano" domande:

1. Analisi verticale:

- *semplici* la 18, 22, 34, 35, 37, 39, 41, 44, 48, 50, 51, 52, 53, 54, 55, 56, 57, 59, 60, 67, 69, 71, 72, 77, 79, 80, 82, 84, 87 e 88. Inoltre di queste sembrano in relazione ancora più stretta le domande 18, 22, 40 e 59.
- *difficili* la 3, 4, 6, 13, 16, 19, 24, 25, 26, 36, 38, 42, 43, 46, 49, 61, 63, 65, 66, 70, 78, 81, 85 e 89. Inoltre di queste sembrano in relazione ancora più stretta le domande 6, 13, 19, 36, 38, 42, 46 e 70.

2. Analisi orizzontale: Appaiono in relazione stretta le seguenti domande:

- Vengono meno le relazioni individuate precedentemente.

Architetture sottoposte ad analisi:

```

• trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:12, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:12, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:8, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:6, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:6, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:6, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001})

```

- ```

layer_defs = [];
layer_defs.push
({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:6, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

```
- ```

layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:6, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

```
- ```

layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

```
- ```

layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});

```

```

layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:3, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:3, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:3, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];

```

```

layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:3, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:1, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:3, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:1, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:5, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:1, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

```

```

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:1, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:1, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:3, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:1, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:6, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:6, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:6, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:6, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

```

- layer\_defs = [];
 

```
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:18, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:18, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});
```
- net = new convnetjs.Net();
 net.makeLayers(layer\_defs);

 trainer = new convnetjs.SGDTrainer(net, {learning\_rate:0.01,
momentum:0.1, batch\_size:10, l2\_decay:0.001});
- layer\_defs = [];
 

```
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:18, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:18, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:18, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});
```
- net = new convnetjs.Net();
 net.makeLayers(layer\_defs);

 trainer = new convnetjs.SGDTrainer(net, {learning\_rate:0.01,
momentum:0.1, batch\_size:10, l2\_decay:0.001});
- layer\_defs = [];
 

```
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:3, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:1, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});
```
- net = new convnetjs.Net();
 net.makeLayers(layer\_defs);

 trainer = new convnetjs.SGDTrainer(net, {learning\_rate:0.01,
momentum:0.1, batch\_size:10, l2\_decay:0.001});
- layer\_defs = [];
 

```
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:3, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:3, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:1, activation: 'tanh'});
```

```

layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:6, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:3, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:1, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:6, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:4, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:1, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

• layer_defs = [];
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth:89});
layer_defs.push({type:'fc', num_neurons:8, activation: 'tanh'});
layer_defs.push({type:'fc', num_neurons:8, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:89});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

```

```

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01,
momentum:0.1, batch_size:10, l2_decay:0.001});

```

L’obiettivo dei miei test è stato individuare una configurazione abbastanza ”forte” da permettere un riscontro di correlazioni e dei possibili cluster esistenti con il solo ausilio della Rete. Tuttavia per proseguire il mio studio è indispensabile che mi avvalga di anche altri strumenti di supporto. Il prossimo passo è costruire un modello di dati mediante la tecnica di Principal Component Analysis. Avvalendomi dello studio approfondito dello studio del testo delle 89 domande di logica e dai test sopra effettuati ho tratto le seguenti considerazioni:

- Un layer non sembra sufficiente per trattare il problema in esame. Mi aspetto che i dati si aggregino per argomento: due argomenti principali (serie e eulero-venn) e la parte restante, domande senza un vero e proprio tema, concorrente a creare il pavimento della Rete;
- Un numero di neuroni superiore a 11 per hidden layer genera situazioni riconducibili all’overfitting.

### **2.3 Funzionalità offerte dall’interfaccia della Rete neurale costruita per le domande di logica**

La Rete neurale sviluppata è stata pensata per una persona che deve svolgere analisi dei dati mirata alla previsione dei risultati sulle domande del database aziendale. L’applicativo realizzato offre le seguenti funzionalità:

- Caricamento dei dati con l’utilizzo di formati *CSV* o *TXT*;
- Possibilità di visualizzare i dati caricati direttamente su pagina web. Ogni risposta viene presentata con codice di test e codice della domanda;
- Possibilità di visualizzare i dati caricati nella Rete mediante formato *JSON*. In questo modo la rete post apprendimento viene salvata ed è possibile visualizzare come ogni nodo pesa ogni variabile. Inoltre l’ultima rete salvata può venire caricata in ogni momento e riutilizzata;
- La Rete neurale è fornita di due textarea:
  - La prima che permette la visualizzazione e la modifica dell’architettura in uso, con indicazione della tipologia di trainer utilizzato. Questa sezione permette la modifica delle variabili di configurazione,Se si desidera modificare il numero di domande analizzate dalla rete si deve modificare il numero di neuroni espressi nella regressione, il file di previsione, il file configurazione e index della rete.;

- La seconda che consente la visualizzazione dei risultati ottenuti dalla Rete.
- È possibile settare la previsione che si vuole ottenere, mediante un’area di inserimento con box a radio. Il valore di default imposto è 0.
- Per ogni previsione svolta viene visualizzato non solo il risultato della stessa numericamente espresso all’interno della seconda textarea; ma viene anche rappresentato visivamente per mezzo di canvas;
- È offerta la funzionalità di inserimento dei parametri di differenziale con filtro sulla colorazione, che permette l’indicazione dei cluster esistenti;
- È possibile visualizzare per ogni elemento soggetto alla previsione il dettaglio della previsione stessa. Questa viene presentata visivamente per mezzo di canvas;
- È possibile eliminare lo storico dei dati stampati nella seconda textarea.

### 2.3.1 Diagrammi dei casi d’uso

#### Attori

Gli *attori* esistenti all’interno del sistema e interagiscono con esso sono:

- **Utente osservatore:** utente che agisce direttamente sulla Rete neurale, in modo da effettuare osservazioni ed analisi dei dati di interesse.

#### Casi d’uso

**UC0\_g: Operazioni utente - Visione generale interfacciamento con la Rete neurale**

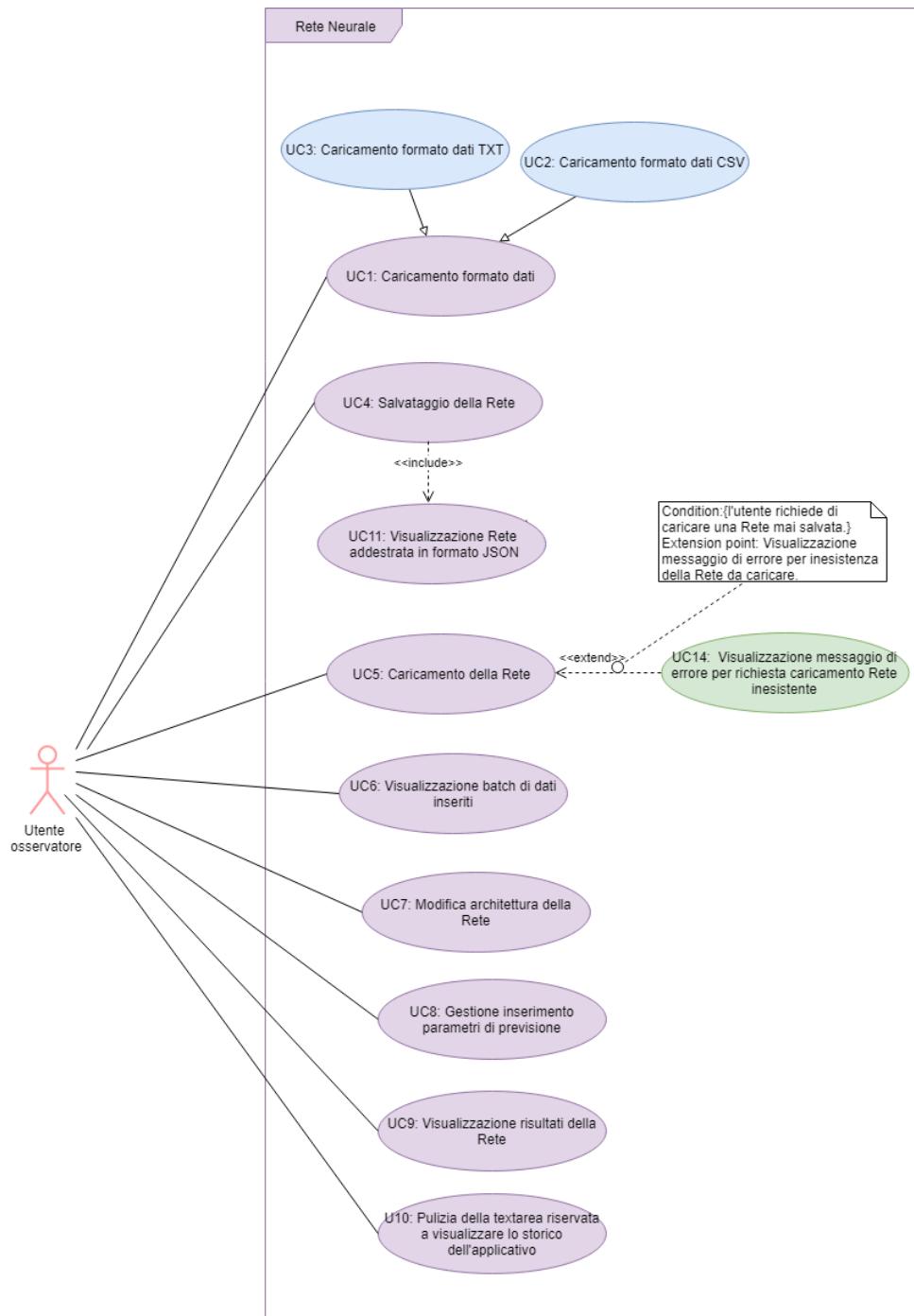


Figura 17: UC0\_g: Operazioni utente - Visione generale interfacciamento con la Rete neurale

- **Descrizione:** Il sistema permette all'utente di interfacciarsi con la

Rete neurale;

- **Attori:** Utente osservatore;
- **Precondizione:** La Rete neurale è in attesa che l'utente effettui almeno un'operazione;
- **Postcondizione:** L'utente si è interfacciato con le funzionalità offerte della Rete neurale.
- **Scenario principale:**
  1. L'utente può effettuare il caricamento dei dati di analisi nella Rete in formato txt o csv (UC1);
  2. L'utente può effettuare il salvataggio di una Rete addestrata (UC4);
  3. L'utente può caricare nel sistema l'ultima Rete salvata (UC5);
  4. L'utente può visualizzare il batch di dati immessi nella Rete (UC6);
  5. L'utente può modificare la configurazione dell'architettura di Rete (UC7);
  6. L'utente può effettuare l'inserimento dei parametri di previsione (UC8);
  7. L'utente può visualizzare i risultati della Rete, ottenuti sui dati caricati e addestrati, a seguito delle operazioni di previsione (UC9);
  8. L'utente può procedere all'eliminazione dello storico della Rete contenuto all'interno della textarea dedicata (UC10).

#### **UC1: Caricamento formato dati**

- **Descrizione:** Il sistema permette all'utente di poter caricare i dati su cui effettuare l'analisi usando formato csv o txt;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete neurale è in attesa che l'utente effettui l'operazione di caricamento del file di dati;
- **Postcondizione:** L'utente ha potuto caricare il file di dati di suo interesse;
- **Scenario principale:**
  1. L'utente effettua il caricamento di un file dati;

2. L'utente conferma l'operazione di caricamento dati (UC0\_g.1).

#### **Generalizzazioni:**

1. L'utente può effettuare il caricamento di un formato di dati csv (UC2);
2. L'utente può effettuare il caricamento di un formato di dati txt (UC3).

#### **UC2: Caricamento formato dati CSV**

- **Descrizione:** Il sistema permette all'utente di poter caricare i dati su cui effettuare l'analisi usando formato csv;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete neurale è in attesa che l'utente effettui l'operazione di caricamento del file di dati;
- **Postcondizione:** L'utente ha potuto caricare il file di dati di suo interesse nel formato csv;
- **Scenario principale:** L'utente sceglie il file di dati che vuole caricare nella Rete.

#### **UC3: Caricamento formato dati TXT**

- **Descrizione:** Il sistema permette all'utente di poter caricare i dati su cui effettuare l'analisi usando formato txt;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete neurale è in attesa che l'utente effettui l'operazione di caricamento del file di dati;
- **Postcondizione:** L'utente ha potuto caricare il file di dati di suo interesse nel formato txt;
- **Scenario principale:** L'utente sceglie il file di dati che vuole caricare nella Rete.

#### **UC0\_g.1: Conferma operazione di caricamento**

- **Descrizione:** L'utente conferma il file dati scelto con cui addestrare la Rete;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha scelto il file dati da caricare;

- **Postcondizione:** L'utente ha potuto caricare il file di dati di suo interesse;
- **Scenario principale:**
  1. L'utente ha confermato il file dati scelto con l'apposto pulsante.
- **Estensioni:**
  1. Il browser in uso dall'utente non supporta lo standard HTML5. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC\_g.2);
  2. L'utente ha effettuato il caricamento di un formato di dati non conforme al formato csv o txt. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC\_g.3).

**UC\_g.2: Visualizzazione messaggio di errore di mancata idoneità del browser in uso**

- **Descrizione:** Il file nel formato corretto viene caricato nel sistema, che non è in grado di processare alcuna informazione in quanto il browser in uso dall'utente non supporta lo standard di markup HTML5;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente non ha ancora caricato il file nella Rete;
- **Postcondizione:** Il sistema produce un messaggio d'errore per l'utente; con il quale quest'ultimo viene informato nella non conformità del browser in uso con gli standard minimi richiesti per portare a termine l'operazione di caricamento del file nella Rete con successo;
- **Scenario principale:** Il file scelto dall'utente viene caricato all'interno della Rete.

**UC\_g.3: Visualizzazione messaggio di errore per formato di dati non coerente con le aspettative**

- **Descrizione:** Il file nel che viene caricato nella rete non è del formato richiesto corretto;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente non ha ancora caricato il file nella Rete;
- **Postcondizione:** Il sistema produce un messaggio d'errore per l'utente; con il quale quest'ultimo viene informato nella non conformità del del formato di dati scelto;
- **Scenario principale:** Il file viene caricato all'interno della Rete.

#### **UC4: Salvataggio della Rete**

- **Descrizione:** L'utente richiede che i dati caricati all'interno della Rete vengano salvati nello stato in cui si trovano;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha caricato un file dati nella Rete e quest'ultima ne ha provveduto all'addestramento;
- **Postcondizione:** L'utente ha gestito i dati caricati all'interno della Rete in modo da salvare la Rete;
- **Scenario principale:** L'utente effettua l'operazione di salvataggio della Rete mediante l'apposito pulsante.
- **Inclusioni:**
  1. La Rete salvata viene visualizzata sulla textarea riservata i dati della rete addestrata in formato JSON (UC1).

#### **UC11: Visualizzazione Rete addestrata in formato JSON**

- **Descrizione:** L'utente visualizza in formato JSON i dati della Rete sottoposti ad addestramento;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha effettuato il salvataggio della Rete;
- **Postcondizione:** L'utente ha visualizzato la Rete salvata, tradotta in formato JSON, all'interno della textarea riservata dell'applicativo.
- **Scenario principale:** L'utente visualizza i dati addestrati in formato JSON.

#### **UC5: Caricamento della Rete**

- **Descrizione:** L'utente richiede che la Rete salvata precedentemente venga ricaricata e preparata all'uso nella Rete;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha salvato una Rete precedentemente;
- **Postcondizione:** L'utente ha gestito i dati caricati all'interno della Rete in modo da ricaricare una Rete precedentemente salvata;
- **Scenario principale:** L'utente effettua il caricamento della Rete precedente salvata mediante l'apposito pulsante.

- **Estensioni:**

1. L'utente ha richiesto il caricamento di una Rete mai precedentemente salvata. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC14).

**UC14: Visualizzazione messaggio di errore per richiesta caricamento Rete inesistente**

- **Descrizione:** L'utente visualizza un messaggio di errore nel caso in cui abbia richiesto il caricamento di una Rete quando non ne è ancora stata salvata nessuna;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha richiesto il caricamento dei dati della Rete precedente;
- **Postcondizione:** Il sistema ha fornito all'utente la visualizzazione di un messaggio di errore che indica l'inesistenza di una Rete da caricare.
- **Scenario principale:** All'utente viene indicato l'errore.

**UC6: Visualizzazione batch di dati inseriti**

- **Descrizione:** L'utente visualizza i dati caricati nella Rete all'interno di una textarea riservata, all'interno della pagina web dell'applicativo; e ne può ottenere un maggior dettaglio visualizzando gli stessi su una pagina web riservata, in cui vengono evidenziati i significati dei dati mostrati;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha effettuato il caricamento dei dati nella Rete in uno dei formati idonei;
- **Postcondizione:** L'utente visualizza il batch di dati caricati nella Rete;
- **Scenario principale:**
  1. L'utente visualizza i dati caricati nella Rete all'interno della pagina web dell'applicativo stesso (UC6.1);
  2. L'utente visualizza i dati caricati nella Rete su una pagina web a parte (UC6.2).

**UC6.1: Visualizzazione dei dati caricati all'interno della pagina web riservata all'applicativo**

- **Descrizione:** L'utente visualizza i dati caricati nella Rete all'interno di una textarea riservata, all'interno della pagina web dell'applicativo, che ha il compito di tenere traccia di tutte le operazioni svolte nella Rete;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha effettuato il caricamento dei dati nella Rete, in uno dei formati idonei;
- **Postcondizione:** L'utente ha visualizzato all'interno della textarea i vettori di dati caricati nella Rete;
- **Scenario principale:** L'utente visualizza i dati caricati nella Rete all'interno della textarea dell'applicativo della Rete prima che il sistema proceda con l'addestramento.

#### **UC6.2: Visualizzazione dei dati caricati all'interno della Rete su pagina web a parte**

- **Descrizione:** L'utente visualizza i dati caricati nella Rete su una pagina web riservata ad adempire tale mansione. I dati caricati vengono presentati con indicazione del codice di test e di domanda a cui ognuno fa capo. La funzionalità ha lo scopo di rendere più agevole lo svolgimento di osservazioni sui dati;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha effettuato il caricamento dei dati nella Rete in uno dei formati idonei;
- **Postcondizione:** L'utente ha visualizzato all'interno di una pagina web riservata i vettori di dati caricati nella Rete;
- **Scenario principale:** L'utente visualizza i dati caricati nella Rete all'interno di una pagina web reindirizzata dall'applicativo della Rete.

#### **UC7: Modifica architettura della Rete**

- **Descrizione:** L'utente ha la possibilità di riconfigurare l'architettura della Rete neurale modificando i layer presenti e il loro contenuto.
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete desiderata dall'utente non esiste, e di conseguenza l'utente non ha ancora effettuato alcuna operazione sulla essa;

- **Postcondizione:** L'utente ha provveduto a configurare la Rete in base alle proprie necessità;
- **Scenario principale:**
  1. L'utente effettua sulla Rete operazioni di modifica dell'architettura concesse (UC7.1);
  2. L'utente effettua sulla Rete operazioni di modifica dell'architettura non concesse (U7.2);
  3. L'utente conferma l'operazione di modifica dell'architettura di Rete (UC7.3).

#### **UC7.1: Gestione operazioni di configurazione dell'architettura concesse**

- **Descrizione:** L'utente ha la possibilità di riconfigurare l'architettura della Rete neurale. Esso può procedere a modificare il numero di neuroni per layer, impostare dei nuovi layers, eliminare i layers intermedi esistenti, modificare/eliminare la funzione attivazione e ridefinire i parametri del trainer della Rete;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete desiderata dall'utente non esiste, e di conseguenza l'utente non ha ancora effettuato alcuna operazione sulla essa;
- **Postcondizione:** L'utente ha provveduto a configurare la Rete in base alle proprie necessità;
- **Scenario principale:**
  1. L'utente ha aggiunto un nuovo layer intermedio nella Rete (UC7.1.1);
  2. L'utente ha eliminato un layer intermedio esistente nella Rete (UC7.1.2);
  3. L'utente ha modificato un layer intermedio (UC7.1.3);
  4. L'utente ha modificato i parametri del trainer della Rete (UC7.1.4).

##### **UC7.1.1: Aggiunta layer intermedio**

- **Descrizione:** L'utente ha la possibilità di riconfigurare l'architettura della Rete neurale. Esso può procedere ad aggiungere un layer intermedio nella Rete;
- **Attori:** Utente osservatore;

- **Precondizione:** La Rete desiderata dall'utente non esiste, e di conseguenza l'utente non ha ancora effettuato alcuna operazione sulla essa;
- **Postcondizione:** L'utente ha provveduto a configurare la Rete in base alle proprie necessità aggiungendo un nuovo layer intermedio;
- **Scenario principale:**

1. L'utente ha aggiunto un layer intermedio alla Rete:
  - L'utente effettua l'inserimento del parametro di tipo;
  - L'utente effettua l'inserimento del numero di neuroni;
  - L'utente può indicare o meno la funzione attivazione;

#### **UC7.1.2: Eliminazione layer intermedio esistente**

- **Descrizione:** L'utente ha la possibilità di riconfigurare l'architettura della Rete neurale. Esso può procedere ad eliminare un layer intermedio presente nella Rete;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete desiderata dall'utente non esiste, e di conseguenza l'utente non ha ancora effettuato alcuna operazione sulla essa;
- **Postcondizione:** L'utente ha provveduto a configurare la Rete in base alle proprie necessità eliminando un layer intermedio;
- **Scenario principale:**

1. L'utente ha eliminato un layer intermedio esistente nella Rete:
  - L'utente elimina il parametro di tipo;
  - L'utente elimina il numero di numero di neuroni;
  - L'utente elimina se presente la funzione attivazione.

#### **UC7.1.3: Modifica layer intermedio esistente**

- **Descrizione:** L'utente ha la possibilità di modificare un layer intermedio presente nella Rete
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete desiderata dall'utente non esiste, e di conseguenza l'utente non ha ancora effettuato alcuna operazione sulla essa;

- **Postcondizione:** L'utente ha provveduto a configurare la Rete in base alle proprie necessità provvedendo a ridefinire il contenuto di un layer intermedio;
- **Scenario principale:**

1. L'utente ha ridefinito un layer intermedio presente nella Rete:
  - L'utente può modificare il numero di neuroni presenti nella Rete;
  - L'utente può modificare/aggiungere/eliminare la funzione attivazione.

#### **UC7.1.4: Modifica di trainer**

- **Descrizione:** L'utente ha la possibilità di configurare la dichiarazione del trainer della Rete;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente non ha ancora effettuato alcuna operazione e la Rete desiderata ancora non esiste;
- **Postcondizione:** L'utente ha provveduto a configurare la Rete in base alle proprie necessità definendo il trainer;
- **Scenario principale:**

1. L'utente ha configurato il trainer della Rete:
  - L'utente può modificare il parametro di learning\_rate;
  - L'utente può modificare il parametro momentum;
  - L'utente può modificare il parametro batch\_size;
  - L'utente può modificare il parametro l2\_decay;
  - L'utente conferma l'operazione di modifica della Rete.

#### **UC7.2: Gestione operazioni di configurazione dell'architettura non concesse**

- **Descrizione:** L'utente ha la possibilità di riconfigurare l'architettura della Rete neurale in base alle proprie esigenze e ai valori delle variabili di cui necessità per compiere le osservazioni sulla Rete.
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete desiderata dall'utente non esiste, e di conseguenza l'utente non ha ancora effettuato alcuna operazione sulla essa;

- **Postcondizione:** L'utente ha provveduto a configurare la Rete in base alle proprie necessità;
- **Scenario principale:**
  1. Le operazioni non concesse all'utente riguardano le seguenti operazioni:
    - (a) Modifica del layer di input (UC7.2.1);
    - (b) Cancellazione del layer di input (UC7.2.2)
    - (c) Modifica del layer di output (UC7.2.3);
    - (d) Cancellazione del layer di output (UC7.2.4);
    - (e) Cancellazione parziale di un layer intermedio (UC7.2.5).

#### **UC7.2.1: Modifica layer di input**

- **Descrizione:** L'utente ha la possibilità di riconfigurare l'architettura della Rete neurale in base alle proprie esigenze, ed è interessato ad effettuando la modifica del layer di input;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete desiderata dall'utente non esiste, e di conseguenza l'utente non ha ancora effettuato alcuna operazione sulla essa;
- **Postcondizione:** L'utente ha provveduto a configurare la Rete in base alle proprie necessità;
- **Scenario principale:**
  1. Le operazioni non concesse all'utente riguardano la modifica nel layer di input:
    - (a) Modifica del tipo di input;
    - (b) Modifica della profondità della Rete;
    - (c) Modifica della larghezza della Rete;
    - (d) Modifica dell'altezza della Rete.

#### **UC7.2.2: Cancellazione del layer di input**

- **Descrizione:** L'utente ha la possibilità di riconfigurare l'architettura della Rete neurale in base alle proprie esigenze, ed è interessato ad effettuando la cancellazione del layer di input;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete desiderata dall'utente non esiste, e di conseguenza l'utente non ha ancora effettuato alcuna operazione sulla essa;

- **Postcondizione:** L'utente ha provveduto a configurare la Rete in base alle proprie necessità;
- **Scenario principale:**
  1. Le operazioni non concesse all'utente riguardano la cancellazione del layer di input:
    - (a) Cancellazione del tipo di input;
    - (b) Cancellazione della profondità della Rete;
    - (c) Cancellazione della larghezza della Rete;
    - (d) Cancellazione dell'altezza della Rete.

#### **UC7.2.3: Modifica layer di output**

- **Descrizione:** L'utente ha la possibilità di riconfigurare l'architettura della Rete neurale in base alle proprie esigenze, ed è interessato ad effettuando la modifica del layer di output;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete desiderata dall'utente non esiste, e di conseguenza l'utente non ha ancora effettuato alcuna operazione sulla essa;
- **Postcondizione:** L'utente ha provveduto a configurare la Rete in base alle proprie necessità;
- **Scenario principale:**
  1. Le operazioni non concesse all'utente riguardano la modifica nel layer di output:
    - (a) Modifica del tipo di regressione;
    - (b) Modifica del numero di neuroni.

#### **UC7.2.4: Cancellazione del layer di output**

- **Descrizione:** L'utente ha la possibilità di riconfigurare l'architettura della Rete neurale in base alle proprie esigenze, ed è interessato ad effettuando la cancellazione del layer di output;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete desiderata dall'utente non esiste, e di conseguenza l'utente non ha ancora effettuato alcuna operazione sulla essa;
- **Postcondizione:** L'utente ha provveduto a configurare la Rete in base alle proprie necessità;

- **Scenario principale:**

1. Le operazioni non concesse all'utente riguardano la cancellazione del layer di output:
  - (a) Cancellazione del tipo di regressione;
  - (b) Cancellazione del numero di neuroni.

**UC7.2.5: Cancellazione parziale di un layer intermedio esistente**

- **Descrizione:** L'utente ha la possibilità di riconfigurare l'architettura della Rete neurale in base alle proprie esigenze, ed è interessato ad effettuando la cancellazione di alcuni campi di un layer intermedio esistente;

- **Attori:** Utente osservatore;

- **Precondizione:** La Rete desiderata dall'utente non esiste, e di conseguenza l'utente non ha ancora effettuato alcuna operazione sulla essa;

- **Postcondizione:** L'utente ha provveduto a configurare la Rete in base alle proprie necessità;

- **Scenario principale:**

1. Le operazioni non concesse all'utente riguardano la cancellazione del layer di intermedio:
  - (a) Cancellazione del tipo del layer intermedio;
  - (b) Cancellazione del numero di neuroni;
  - (c) Cancellazione della funzione attivazione, se presente.

**UC7.3: Conferma dell'operazione di modifica della Rete**

- **Descrizione:** L'utente conferma la configurazione della Rete presentata nella textarea dedicata;

- **Attori:** Utente osservatore;

- **Precondizione:** La Rete desiderata dall'utente non esiste, e di conseguenza l'utente non ha ancora effettuato alcuna operazione sulla essa;

- **Postcondizione:** L'utente ha provveduto a configurare la Rete in base alle proprie necessità;

- **Scenario principale:**

1. L'utente ha confermato la configurazione di Rete.

- **Estensioni:**

1. L'utente ha tentato di modificare il type input sul layer di input della Rete. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC7.4);
2. L'utente ha tentato di modificare la larghezza dichiarata (out\_sx) sul layer di input della Rete. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC7.5);
3. L'utente ha tentato di modificare l'altezza dichiarata (out\_sy) sul layer di input della Rete. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC7.6);
4. L'utente ha tentato di modificare la profondità dichiarata (out\_depth) sul layer di input della Rete. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC7.7);
5. L'utente ha tentato di modificare il tipo di connessione fulled connected su un layer intermedio della Rete. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC7.8);
6. L'utente ha tentato di cancellare il numero di neuroni presenti su un layer intermedio della Rete. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC7.9);
7. L'utente ha tentato di inserire un numero di neuroni presenti su un layer intermedio della Rete minore o uguale a 0. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC7.10);
8. L'utente ha tentato di modificare la funzione di attivazione su un layer intermedio della Rete, con una dichiarazione di funzione non valida. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC7.11);
9. L'utente ha tentato di modificare il tipo regressione presente sul layer di output della Rete. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC7.12);
10. L'utente ha tentato di modificare il numero di neuroni presente sul layer di output della Rete. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC7.13).

**UC7.6: Visualizzazione messaggio di errore di type input sul layer di input**

- **Descrizione:** L'utente visualizza un messaggio di errore nel caso in cui abbia modificato nel layer di input il tipo dello stesso;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha fornito un tipo nel layer di input
- **Postcondizione:** Il sistema ha fornito all'utente la visualizzazione di un messaggio di errore che indica la incompatibilità del tipo richiesto con la natura della Rete;
- **Scenario principale:** All'utente viene indicato l'errore con l'indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

**UC7.7: Visualizzazione messaggio di errore per la larghezza dichiarata (out\_sx) sul layer di input**

- **Descrizione:** L'utente visualizza un messaggio di errore nel caso in cui abbia modificato nel layer di input la larghezza dichiarata;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha fornito la dimensione della larghezza della componente base della Rete Vol;
- **Postcondizione:** Il sistema ha fornito all'utente la visualizzazione di un messaggio di errore che indica la incompatibilità del della dimensione richiesta con la natura della Rete;
- **Scenario principale:** All'utente viene indicato l'errore con l'indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

**UC7.8: Visualizzazione messaggio di errore per altezza dichiarata (out\_sy) sul layer di input**

- **Descrizione:** L'utente visualizza un messaggio di errore nel caso in cui abbia modificato nel layer di input l'altezza dichiarata;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha fornito la dimensione dell'altezza della componente base della Rete Vol;
- **Postcondizione:** Il sistema ha fornito all'utente la visualizzazione di un messaggio di errore che indica la incompatibilità del della dimensione richiesta con la natura della Rete;

- **Scenario principale:** All’utente viene indicato l’errore con l’indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

**UC7.9: Visualizzazione messaggio di errore per la profondità (out\_depth) dichiarata input sul layer di input**

- **Descrizione:** L’utente visualizza un messaggio di errore nel caso in cui abbia modificato nel layer di input la profondità dichiarata;
- **Attori:** Utente osservatore;
- **Precondizione:** L’utente ha fornito la dimensione della profondità della componente base della Rete Vol;
- **Postcondizione:** Il sistema ha fornito all’utente la visualizzazione di un messaggio di errore che indica la incompatibilità della dimensione richiesta con la natura della Rete;
- **Scenario principale:** All’utente viene indicato l’errore con l’indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

**UC7.10: Visualizzazione messaggio di errore per type fulled connected sui layers intermedi**

- **Descrizione:** L’utente visualizza un messaggio di errore nel caso in cui abbia inserito in uno dei layers intermedi un tipo diverso da fulled connected;
- **Attori:** Utente osservatore;
- **Precondizione:** L’utente ha fornito il tipo in un layer intermedio;
- **Postcondizione:** Il sistema ha fornito all’utente la visualizzazione di un messaggio di errore che indica la incompatibilità del tipo richiesto con la natura della Rete;
- **Scenario principale:** All’utente viene indicato l’errore con l’indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

**UC7.11: Visualizzazione messaggio di errore per cancellazione numero di neuroni sui layers intermedi**

- **Descrizione:** L’utente visualizza un messaggio di errore nel caso in cui effettuato la cancellazione in uno dei layers intermedi del numero di neuroni presenti;

- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha eliminato il numero di neuroni da un layer intermedio;
- **Postcondizione:** Il sistema ha fornito all'utente la visualizzazione di un messaggio di errore che indica la incompatibilità dell'operazione richiesta con la natura della Rete;
- **Scenario principale:** All'utente viene indicato l'errore con l'indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

**UC7.12: Visualizzazione messaggio di errore per inserimento numero di neuroni sui layers intermedi minore o uguale a 0**

- **Descrizione:** L'utente visualizza un messaggio di errore nel caso abbia inserito/modificato il numero di neuroni presenti in un layer intermedio con un valore minore o uguale a 0;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha inserito il numero di neuroni in un layer intermedio;
- **Postcondizione:** Il sistema ha fornito all'utente la visualizzazione di un messaggio di errore che indica la incompatibilità del valore definito con la natura della Rete;
- **Scenario principale:** All'utente viene indicato l'errore con l'indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

**UC7.13: Visualizzazione messaggio di errore per la funzione attivazione presente sui layers intermedi**

- **Descrizione:** L'utente visualizza un messaggio di errore nel caso abbia inserito/modificato la funzione di attivazione su un layer intermedio con parametro diverso da uno dei desiderati.
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha inserito una funzione attivazione in un layer intermedio;
- **Postcondizione:** Il sistema ha fornito all'utente la visualizzazione di un messaggio di errore che indica l'inesistenza della funzione attivazione richiesta per la Rete;

- **Scenario principale:** All’utente viene indicato l’errore con l’indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

**UC7.14: Visualizzazione messaggio di errore per type regression sul layer di output**

- **Descrizione:** L’utente visualizza un messaggio di errore nel caso abbia modificato il tipo dichiarato nel layer di output
- **Attori:** Utente osservatore;
- **Precondizione:** L’utente ha modificato il tipo presente nel layer di output;
- **Postcondizione:** Il sistema ha fornito all’utente la visualizzazione di un messaggio di errore che indica l’incompatibilità del valore del tipo richiesto con la natura della Rete;
- **Scenario principale:** All’utente viene indicato l’errore con l’indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

**UC7.15: Visualizzazione messaggio di errore per numero di neuroni sul livello di output**

- **Descrizione:** L’utente visualizza un messaggio di errore nel caso abbia modificato il numero di neuroni dichiarati nel layer di output
- **Attori:** Utente osservatore;
- **Precondizione:** L’utente ha modificato il numero di neuroni presenti nel layer di output;
- **Postcondizione:** Il sistema ha fornito all’utente la visualizzazione di un messaggio di errore che indica l’incompatibilità del tipo di neuroni richiesti con la natura della Rete;
- **Scenario principale:** All’utente viene indicato l’errore con l’indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

**UC8: Gestione inserimento parametri di previsione**

- **Descrizione:** L’utente inserisce i parametri necessari per effettuare previsione sui dati addestrati della Rete;
- **Attori:** Utente osservatore;

- **Precondizione:** La Rete è addestrata con i dati caricati dall'utente;
- **Postcondizione:** L'utente ha ottenuto la previsione sui dati addestrati in base ai parametri passati in input alla Rete;
- **Scenario principale:**
  1. L'utente ha inserito i parametri necessari per effettuare una clusterizzazione sui dati, sui dati predetti dall'applicativo (UC8.1);
  2. L'utente ha selezionato per ogni domanda una delle risposte, tra quelle concesse, in modo da generare un vettore previsione con il quale si effettua la previsione sui dati addestrati dalla Rete (UC8.2);
  3. L'utente ha confermato le operazioni dando il via al calcolo dei risultati di previsione (UC8.3).

#### **UC8.1: Inserimento parametri di differenziale di accoppiamento per effettuare clusterizzazione della previsione**

- **Descrizione:** L'utente inserisce i parametri necessari per effettuare clusterizzazione una volta ottenuti dalla Rete i dati di previsione;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete è addestrata con i dati caricati dall'utente;
- **Postcondizione:** L'utente ha inserito i parametri di differenziale di accoppiamento necessari;
- **Scenario principale:**
  1. Vengono inseriti i tre parametri di clusterizzazione:
    - (a) L'utente può inserire il primo parametro differenziale;
    - (b) L'utente può inserire il secondo parametro differenziale;
    - (c) L'utente può inserire il terzo parametro differenziale.

#### **UC8.2: Inserimento dei parametri del vettore previsione**

- **Descrizione:** L'utente inserisce il valore delle domande per effettuare previsione sui dati addestrati della Rete;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete è addestrata con i dati caricati dall'utente;
- **Postcondizione:** L'utente ha inserito le risposte alle domande nel vettore previsione;
- **Scenario principale:**

1. L'utente sceglie il contenuto del vettore previsione:
  - (a) L'utente sceglie la risposta alla domanda  $n = 1, \dots, 89$  con uno dei seguenti valori: 0 risposta non data, -1 risposta sbagliata, 1 risposta corretta.

#### **UC8.3: Conferma operazioni di previsione inseriti**

- **Descrizione:** L'utente conferma i dati inseriti come differenziali di accoppiamento e i valori delle risposte delle domande nel vettore previsione;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha inserito i parametri su cui effettuare previsione su una Rete addestrata;
- **Postcondizione:** L'utente ha confermato i dati inseriti e innescato il calcolo della previsione;
- **Scenario principale:** L'utente conferma i differenziali di accoppiamento e il vettore previsione inseriti nell'applicativo. **Estensioni:**
  1. L'utente ha inserito per uno o più dei parametri di clusterizzazione un valore stringa. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC8.4);
  2. L'utente ha inserito per uno o più dei parametri di clusterizzazione un valore vuoto. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC8.5);
  3. L'utente ha inserito per uno o più dei parametri di clusterizzazione un numero negativo. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC8.6);
  4. L'utente ha inserito per uno o più dei parametri di clusterizzazione un numero superiore a 255. Viene ritornato un messaggio esplicativo che indica l'impossibilità di terminare l'operazione (UC8.7).

#### **UC8.4: Visualizzazione messaggio di errore inserimento valore stringa**

- **Descrizione:** L'utente visualizza un messaggio di errore nel caso abbia inserito come parametro di clusterizzazione un valore stringa;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha inserito come differenziale di accoppiamento un valore stringa;

- **Postcondizione:** Il sistema ha fornito all'utente la visualizzazione di un messaggio di errore che indica l'incompatibilità del tipo di valore inserito;
- **Scenario principale:** All'utente viene indicato l'esistenza di un'errore con l'indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

**UC8.5: Visualizzazione messaggio di errore inserimento valore vuoto**

- **Descrizione:** L'utente visualizza un messaggio di errore nel caso abbia inserito come parametro di clusterizzazione un valore vuoto;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha inserito come differenziale di accoppiamento un valore vuoto;
- **Postcondizione:** Il sistema ha fornito all'utente la visualizzazione di un messaggio di errore che indica l'incompatibilità del tipo di valore inserito;
- **Scenario principale:** All'utente viene indicato l'esistenza di un'errore con l'indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

**UC8.6: Visualizzazione messaggio di errore inserimento numero negativo**

- **Descrizione:** L'utente visualizza un messaggio di errore nel caso abbia inserito come parametro di clusterizzazione un numero negativo;
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha inserito come differenziale di accoppiamento un numero negativo;
- **Postcondizione:** Il sistema ha fornito all'utente la visualizzazione di un messaggio di errore che indica l'incompatibilità del tipo di valore inserito;
- **Scenario principale:** All'utente viene indicato l'esistenza di un'errore con l'indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

**UC8.7: Visualizzazione messaggio di errore inserimento numero positivo massimo**

- **Descrizione:** L'utente visualizza un messaggio di errore nel caso abbia inserito come parametro di clusterizzazione un numero positivo superiore a 255 (massima parametro rgb possibile);
- **Attori:** Utente osservatore;
- **Precondizione:** L'utente ha inserito come differenziale di accoppiamento un numero positivo superiore a 255;
- **Postcondizione:** Il sistema ha fornito all'utente la visualizzazione di un messaggio di errore che indica l'incompatibilità del tipo di valore inserito;
- **Scenario principale:** All'utente viene indicato l'esistenza di un'errore con l'indicazione della correzione necessaria per il corretto funzionamento dalla Rete.

#### **UC9: Visualizzazione risultati della Rete**

- **Descrizione:** L'utente visualizza i risultati ottenuti dalla Rete a seguito della richiesta, andata a buon fine, di previsione sui dati addestrati della Rete;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete è addestrata e con richiesta di previsione pendente;
- **Postcondizione:** L'utente ha visualizzato i risultati di previsione ottenuti dalla Rete;
- **Scenario principale:**
  1. Visualizzazione nella textarea riservata dell'applicativo dei risultati di previsione ottenuti, espressi in termini numerici su ogni domanda (UC9.1);
  2. Visualizzazione nella textarea riservata dell'applicativo dei risultati di previsione ottenuti, espressi in codice rgb su ogni domanda (UC9.2);
  3. Visualizzazione nella textarea riservata dell'applicativo dei cluster delle domande generati sulla base dei parametri di differenziale di accoppiamento, passati in input e applicati alle previsioni espresse in codice rgb (UC9.3);
  4. Visualizzazione generale, mediante tecnologia canvas, dei risultati delle previsioni, espressi in termini rgb su ogni domanda (UC9.4).

### **UC9.1: Visualizzazione risultati di previsione in termini numerici**

- **Descrizione:** L'utente visualizza i risultati di previsione in termini numerici (calcolati sul vettore previsione della Rete addestrata) all'interno della textarea riservata;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete è addestrata e ha svolto la funzione di previsione;
- **Postcondizione:** L'utente ha visualizzato i risultati di previsione numerici, ottenuti dalla Rete, all'interno della textarea riservata;
- **Scenario principale:** Visualizzazione dei valori di previsione, ottenuti dal vettore di previsione passato in input, stampati all'interno della textarea riservata a contenere lo storico delle osservazione di Rete.

### **UC9.2: Visualizzazione risultati di previsione in termini rgb**

- **Descrizione:** L'utente visualizza i risultati di previsione convertiti in rgb dai valori numerici (calcolati sul vettore previsione della Rete addestrata) all'interno della textarea riservata;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete è addestrata e ha svolto la funzione di previsione;
- **Postcondizione:** L'utente ha visualizzato i risultati di previsione in codice rgb, ottenuti dalla Rete, all'interno della textarea riservata;
- **Scenario principale:** Visualizzazione dei valori di previsione, ottenuti dal vettore di previsione passato in input ed espressi in codice rgb, stampati all'interno della textarea riservata a contenere lo storico delle osservazione di Rete.

### **UC9.3: Visualizzazione risultati di clusterizzazione**

- **Descrizione:** L'utente visualizza, all'interno della textarea dedicata, la clusterizzazione dei dati calcolata sulla previsione espressa in codice rgb;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete è addestrata e ha svolto la funzione di previsione;

- **Postcondizione:** L'utente ha visualizzato la clusterizzazione ottenuta dalla Rete, all'interno della textarea riservata;
- **Scenario principale:** Visualizzazione della clusterizzazione delle domande generata dalla previsione espressa in codice rgb.

**UC9.4: Visualizzazione dei risultati di previsione mediante canvas**

- **Descrizione:** L'utente visualizza i risultati ottenuti dalla Rete, mediante canvas applicati alle previsioni espresse in termini rgb, a seguito della richiesta di previsione sui dati addestrati della Rete;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete è addestrata e con richiesta di previsione pendente;
- **Postcondizione:** L'utente ha visualizzato i risultati di previsione ottenuti dalla Rete rappresentato per mezzo di canvas;
- **Scenario principale:**
  1. Visualizzazione di un quadrato per ogni domanda in analisi con gradazione verde, bianco, rosso; sulla base del risultato di previsione ottenuto a seguito dell'addestramento della Rete e del vettore previsione inoltrato in input;
  2. Visualizzazione di dettaglio, mediante tecnologia canvas, della previsione ottenuta (UC9.4.1).

**UC9.4.1: Visualizzazione dei risultati di previsione in dettaglio mediante canvas**

- **Descrizione:** L'utente visualizza i risultati ottenuti dalla Rete, mediante canvas applicata alle previsioni espresse in termini rgb su un vettore previsione che pone come domanda corretta solo la domanda in esame;
- **Attori:** Utente osservatore;
- **Precondizione:** La Rete ha effettuato previsione sui dati;
- **Postcondizione:** L'utente ha visualizzato i risultati di previsione di dettaglio ottenuti dalla Rete rappresentato per mezzo di canvas;
- **Scenario principale:**

- Visualizzazione di un quadrato, tanti quante sono le domande per ogni domanda in analisi, con gradazione verde, bianco, rosso. La gradazione è calcolata sul risultato di previsione ottenuto a seguito dell'addestramento della Rete e del vettore previsione che pone corretta la domanda in esame e sbagliate tutte le rimanenti.

### UC10: Pulizia della textarea riservata a visualizzare lo storico dell'applicativo

- Descrizione:** L'utente effettua la cancellazione dello storico in visualizzazione nella textarea riservata;
- Attori:** Utente osservatore;
- Precondizione:** L'utente ha effettuato nella Rete alcune o anche nessuna operazione;
- Postcondizione:** L'utente visualizza la textarea pulita dallo storico delle operazioni effettuate dalla Rete;
- Scenario principale:** L'utente visualizza l'area di lavoro priva di informazioni sui dati elaborati fino a quel momento.

## 2.4 Interfaccia utente Rete neurale

**Questions test - Prevision Neural Net**

---

**CONFIGURAZIONE RETE NEURALE**

```
layer_defs = []
layer_defs.push({type:'input', out_sx:1, out_sy:1, out_depth: 6});
layer_defs.push({type:'fc', num_neurons:2, activation:'tanh'});
layer_defs.push({type:'fc', num_neurons:2, activation: 'tanh'});
layer_defs.push({type:'regression', num_neurons:6});

net = new convnetjs.Net();
net.makeLayers(layer_defs);

trainer = new convnetjs.SGDTrainer(net, {learning_rate:0.01, momentum:0.1, batch_size:10, lr_decay:0.001});
```

**OUTPUT DELLA RETE**

**create JSON** **Clean**

Richiesta di inserimento:  
 - Parametri di differenziale tra le domande (valore valido  $\geq 0$ ). Utili per creare dei cluster parametrizzati in base al codice del colore  
 - Dati del vettore previsione  
 Successivamente premere il pulsante di previsione

Richiesta di previsione inoltrata alla rete ...  
 Il vettore  $[0,0,0,0,0,0]$  ha previsione calcolata di  
 $[0.506284644831758, 0.3321313045918288, -0.32621589743208046, 0.49428328908691455, 0.3386207175509344, -0.2744086771129914]$

Analisi dei dati  
 domanda 1: 125-255-125 - domanda 2: 170-255-170 - domanda 3: 255-171-171 - domanda 4: 128-255-128 - domanda 5: 168-255-168 - domanda 6: 255-185-185  
 Parametri di differenziale dei gruppi, su cui si vuole valutare matematicamente il comportamento della Rete: 0 0 0  
 Relazione fra le domande con un parametri differenziazione configurati  
 cluster 1: 1  
 cluster 2: 2  
 cluster 3: 3  
 cluster 4: 4  
 cluster 5: 5  
 cluster 6: 6

Figura 18: Interfaccia finale utente della Rete neurale di prova - 1.

trainer = new convnetjs.SGDTraainer(net, {learning\_rate:0.01, momentum:0.1, batch\_size:10, l2\_decay:0.001});

**OUTPUT DELLA RETE**

[create JSON](#) [clean](#)

Richiesta di inserimento:  
 - Parametri di differenziale tra le domande (valore valido >= 0). Utili per creare dei cluster parametrizzati in base al codice del colore  
 - Dati del vettore previsione  
 Successivamente premere il pulsante di previsione

Richiesta di previsione inoltrata alla rete ...  
 Il vettore [0,0,0,0,0] ha previsione calcolata di  
 [0.5062846448317758,0.3321313045918288,-0.32621589743208046,0.49428328908691455,0.3386207175509344,-0.2744086771129914]

Analisi dei dati  
 domanda 1: 125-255-125 - domanda 2: 170-255-170 - domanda 3: 255-171-171 - domanda 4: 128-255-128 - domanda 5: 168-255-168 - domanda 6: 255-185-185  
 Relazione fra le domande con un parametri differenziazione configurati

cluster 1: 1  
 cluster 2: 2  
 cluster 3: 3  
 cluster 4: 4  
 cluster 5: 5  
 cluster 6: 6



**DETTAGLIO DELLE DOMANDE**

[back](#)

domanda	1	2	3	4	5	6
domanda 1	■	■	■	■	■	■
domanda 2	■	■	■	■	■	■
domanda 3	■	■	■	■	■	■
domanda 4	■	■	■	■	■	■
domanda 5	■	■	■	■	■	■
domanda 6	■	■	■	■	■	■

Figura 19: Interfaccia finale utente della Rete neurale di prova - 2.

**OUTPUT DELLA RETE**

[create JSON](#) [clean](#)

Parametri di differenziale dei gruppi, su cui si vuole valutare matematicamente il comportamento della Rete: 8 8 8  
 Relazione fra le domande con un parametri differenziazione configurati

cluster 1: 1, 4  
 cluster 2: 2, 5  
 cluster 3: 3  
 cluster 4: 6

Richiesta di previsione inoltrata alla rete ...  
 Il vettore [0,0,0,0,0] ha previsione calcolata di  
 [0.5062846448317758,0.3321313045918288,-0.32621589743208046,0.49428328908691455,0.3386207175509344,-0.2744086771129914]

Analisi dei dati  
 domanda 1: 125-255-125 - domanda 2: 170-255-170 - domanda 3: 255-171-171 - domanda 4: 128-255-128 - domanda 5: 168-255-168 - domanda 6: 255-185-185  
 Relazione fra le domande con un parametri differenziazione configurati

cluster 1: 1, 4  
 cluster 2: 2, 5  
 cluster 3: 3, 6



**go to detail**

**INSEGNAMENTO DATI DI PREVISIONE**

Differenziale di accoppiamento: 8 16 16

Vettore di previsione

Domanda 1 -1 0 0 1 0	Domanda 2 -1 0 0 1 0	Domanda 3 -1 0 0 1 0
Domanda 4 -1 0 0 1 0	Domanda 5 -1 0 0 1 0	Domanda 6 -1 0 0 1 0

[prevision trainer](#)

Figura 20: Interfaccia finale utente della Rete neurale di prova - 3.

## Questions test - Prevision Neural Net

### CONFIGURAZIONE RETE NEURALE

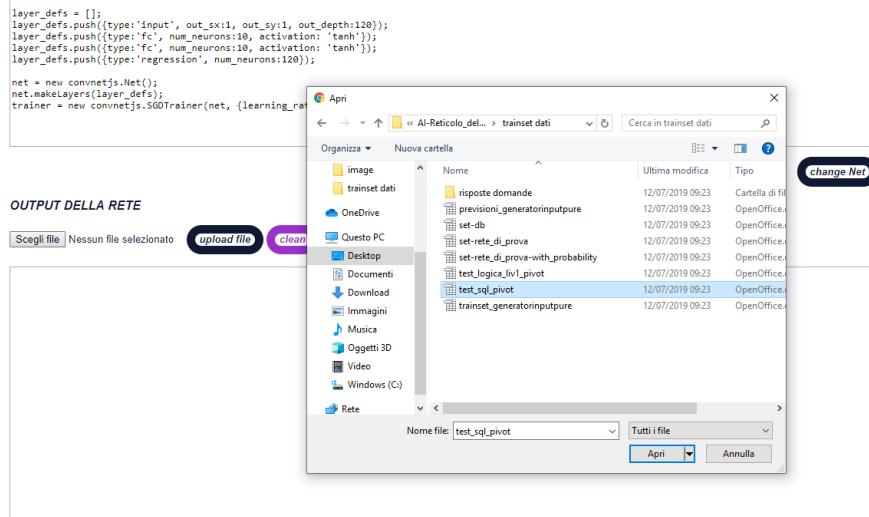


Figura 21: Interfaccia finale utente della Rete neurale del database -1.

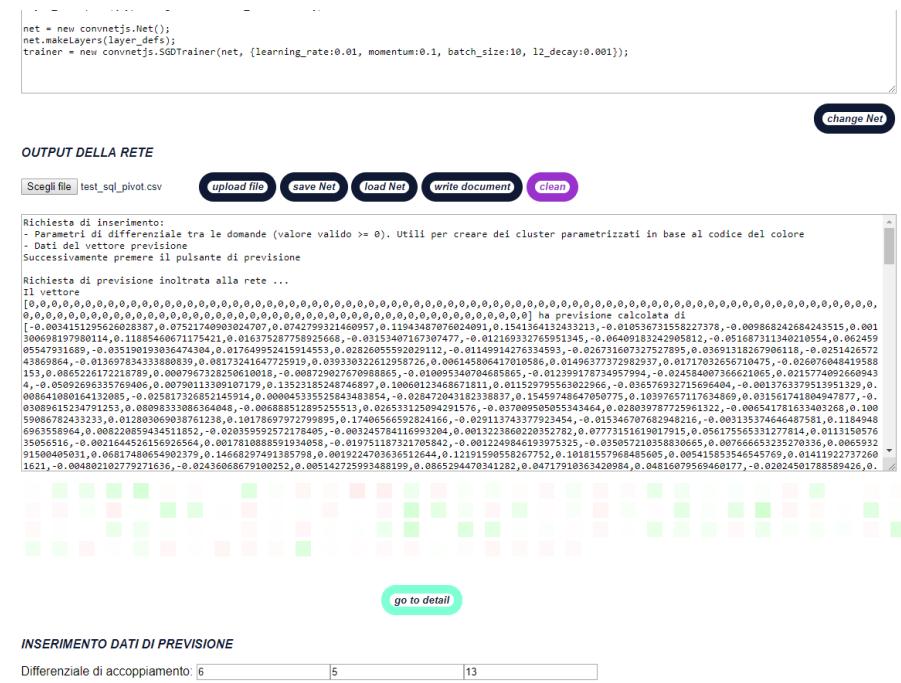


Figura 22: Interfaccia finale utente della Rete neurale del database -2.



## 3 Principal Component Analysis

La **Principal Component Analysis** (acronimo PCA) è una tecnica impiegata nell'ambito della statistica multivariata<sup>2</sup> usata per semplificare i dati d'origine.

Lo scopo che tale tecnica persegue è lo studio della relazione esistenti tra i campioni d'interesse, con la riduzione di un numero più o meno elevato di variabili. La riduzione dimensionale avviene tramite una trasformazione lineare delle variabili coinvolte con lo scopo di effettuare la proiezione di quelle originarie in un nuovo sistema cartesiano, in cui tutte le variabili vengono ordinate in maniera decrescente per ordine di varianza. Successivamente la variabile con maggiore varianza viene proiettata sul primo asse, la seconda sul secondo e via via sempre così per tutte le variabili sotto esame. La PCA è particolarmente utile quando la dimensionalità dello spazio delle misure è elevata (molte colonne); tuttavia i campioni si vengono a trovare in uno spazio di dimensioni significativamente ridotte.

Indispensabile per la buona riuscita della tecnica è, come già esposto sopra, la ricerca del numero di componenti principali significative, ovvero tutte le variabili coinvolte a meno di quelle legate al rumore, che concorrono a comporre la *dimensionalità intrinseca*. Il "rumore" è sempre concentrato nelle ultime variabili, non includerle nell'analisi dei dati porta a dati più puliti, con un rapporto segnale/rumore più alto.

Il calcolo di quali sono le componenti più significative si ottiene con la varianza. La prima componente principale spiega la massima percentuale della variabilità presente nei dati rappresentabili in una singola dimensione, in poche parole la direzione lungo cui si registra la massima dispersione dei dati (quanto il valore medio si discosta). La varianza porta il vantaggio di essere indipendente dal sistema di riferimento, difatti conseguentemente una rotazione degli assi mantiene immutata la varianza totale all'interno del sistema.

### 3.1 Metodologia applicata

L'analisi e l'implementazione della tecnica di **Principal Component Analysis** l'ho svolta nel periodo 17 - 21 luglio con la seguente metodologia:

- Studio e configurazione di R e R-Studio;
- Studio del significato della tecnica e sua correlazione con le Rete neurali;
- Studio della metodologia di implementazione;
- Scelta del package più adeguato per svolgere l'applicazione della PCA sui dati di trainset;

---

<sup>2</sup>parte della statistica in cui l'oggetto dell'analisi è almeno composta da due elementi

- Implementazione del metodo della PCA e dei metodi di supporto ad analisi dei dati;
- Analisi dei risultati ottenuti dalla PCA;
- Analisi delle domande presenti nel database aziendale;
- Confronto dei risultati della PCA con quelli ottenuti dalla Rete neurale e confronto della loro validità con le domande del database.

Ogni implementazione e osservazione l'ho svolta prima nei dati del trainset di prova e solo successivamente gli traslati all'interno del trainset delle domande nel database aziendale. In modo ho avuto sempre ben la correttezza e le possibili correlazioni vigenti nel metodo in uso.

### 3.2 Sviluppo

La tecnica di PCA effettua analisi dei dati. I dati che ho impiegato sono i medesimi che ho utilizzato per effettuare il trainset della Rete neurale. Il formato utilizzato è sempre CSV e i dati contenuti sono strutturati per riga con i risultati di ogni test e per colonna con il risultato di una specifica domanda k.

1. Caricamento del package **factoextra**: la mia scelta è ricaduta su tale package e non in altri con la medesima funzione, per via della sua visualizzazione dei dati elegante basata su ggplot2;
2. **Caricamento del file CSV** in memoria per mezzo della trasposizione in ata frame;
3. *Standardizzazione* dei dati del trainset. Tale compito mi è risultato indispensabile perchè anche se di norma, la standardizzazione, viene usata per evitare situazioni erronee (alcune variabili X presentate con una variabilità molto maggiore rispetto ad altre) e io nei casi in analisi ero in possesso di dati già presentati con la medesima scala; avendo la necessità di individuare gli autovettori e applicare il criterio di Keiser nel trainset ho dovuto procedere ugualmente a standardizzare.
4. Calcolo della **PCA** con la seguente formula:

```
prcomp(df_numeric, scale = FALSE)
```

R offre due metodi per calcolare la PCA:

- *prcomp(x, scale = FALSE)*: dove *x* rappresenta una matrice numerica o data frame e *scale* un valore logico che indica se le variabili devono essere ridimensionate/standardizzate;

- *princomp(x, cor = FALSE, scores = TRUE)*: dove *x* rappresenta una matrice numerica o data frame, *cor* valore logico che se a true ridimensiona e centra i dati prima di procedere all'analisi e *scores* valore logico che se a true calcola le coordinate su ciascun componente principale.

Metodologia che ho utilizzato per passi:

Ho deciso di far uso della funzione *prcomp* perchè usa la decomposizione del valore singolare (SGV) che offre una precisione leggermente migliore rispetto all'uso del metodo *princomp*. Il metodo *prcomp* include nei propri elementi di output:

- (a) *sdev*: deviazione standard delle componenti principali;
  - (b) *rotation*: la matrice dei carichi delle variabili ovvero le colonne degli autovettori;
  - (c) *center*: la media variabile, indica se le variabili devono essere spostate per essere centrate sullo zero;
  - (d) *scale*: deviazione standard delle variabili;
  - (e) *x*: coordinate degli individui sulle componenti principali.
5. Calcolo degli **autovalori della matrice di covarianza**. Mostra la percentuale di varianze di competenza di ciascun componente principale con

```
get_eig(res.pca)
```

6. **Riepilogo** mediante il metodo *summary* dei risultati ottenuti dal calcolo della PCA. Effettua la standardizzazione dei risultati ottenuti e ne calcola gli autovalori di covarianza con individuazione di deviazione standard, proporzione della varianza e proporzione cumulata.

```
summary(res.pca)
```

7. **Calcolo degli autovettori** con

```
loadings(res.pca)
```

8. **Individuazione della matrice correlazione** dei dati con

```
cor(df_numeric)
```

9. Creazione di un data frame contenete per ogni variabile principale quali sono le componenti ordinate in modo decrescente strettamente correlate.

### 3.3 Risultati ottenuti

I dati risultati dall'elaborazione dei dati inerenti il trainset delle domande nel database gli ho riportati, per motivi di spazio e comprensione, in modo parziale. Ogni metodo utilizzato viene di seguito descritto nel dettaglio.

#### 3.3.1 Calcolo della Principal Component Analysis

```
> prcomp(df_numeric, scale = FALSE)
Standard deviations (1, ..., p=6):
[1] 1.5905164 0.9270882 0.6549647 0.2877259 0.2567623 0.2267723

Rotation (n x k) = (6 x 6):
PC1          PC2          PC3          PC4          PC5          PC6
V1 -0.3136368 -0.06351095 0.6506652 -0.0008078537 0.660023581 0.196472324
V2  0.3888043 0.53581153 0.2479470 0.7072146111 -0.008846399 0.005359089
V3 -0.5186977 0.45305827 -0.1486278 0.00111520653 0.150101946 -0.693591199
V4 -0.3009414 -0.07357741 0.6112341 -0.0012775235 -0.719969817 -0.109796655
V5  0.3876316 0.53818077 0.2452635 -0.7069936980 -0.008478550 0.006089356
V6 -0.4905017 0.45668194 -0.2435273 0.0019475491 -0.152746837 0.684257823
```

Figura 24: Visualizzazione del calcolo della PCA tramite prcomop per il trainset di prova.

```
> prcomp(df_numeric, scale = FALSE)
Standard deviations (1, ..., p=9):
[1] 0.58043495 0.55848746 0.54971833 0.53126069 0.52059133 0.51577310 0.49349655 0.48596238 0.47999862 -0.47610358 0.446691672 0.45897619 0.44990142 0.44254212 0.43588986 0.43233385 0.42688299
[19] 0.42079249 0.41550349 0.41868652 0.40526268 0.40119893 0.39676359 0.38988115 0.38394838 0.38248451 0.38161779 0.37446799 0.37951118 0.36811364 0.36109314 0.3588208 0.3535258 0.34981222 0.34507283
[37] 0.34271823 0.33861509 0.33532453 0.33275471 0.32925297 0.32449963 0.31716223 0.31408345 0.30962208 0.30523877 0.30247259 0.29816970 0.29458133 0.29251356 0.29083447 0.28421416
[55] 0.28077829 0.28074860 0.27888496 0.27585038 0.27422473 0.27139698 0.27019256 0.26679640 0.26427239 0.26314990 0.25905281 0.25708924 0.25574716 0.25325341 0.24954843 0.2459152 0.24355894
[73] 0.24187129 0.23789164 0.23380141 0.22959808 0.22384654 0.21341210 0.19833321 0.18012521 0.17449777 0.15236083 0.14016744 0.13282731 0.10140180 0.08865511 0.02703589 0.02534545
[73] 0.24187129 0.23789164 0.23380141 0.22959808 0.22384654 0.21341210 0.19833321 0.18012521 0.17449777 0.15236083 0.14016744 0.13282731 0.10140180 0.08865511 0.02703589 0.02534545

Rotation (n x k) = (89 x 9):
PC1          PC2          PC3          PC4          PC5          PC6          PC7          PC8          PC9          PC10         PC11         PC12         PC13         PC14
V1 -2.636347e-03 0.005200166 -4.597895e-03 0.084770589 -6.267818e-04 0.001175541 -0.001552997 -1.338486e-03 0.000166171 -0.003520794 0.013571066 -0.012466570 0.01606655
V2  0.604902e-02 0.001175541 0.000166171 0.000166171 -0.000166171 0.000166171 0.000166171 0.000166171 0.000166171 0.000166171 0.000166171 0.000166171 0.000166171
V3 -2.415304e-03 -0.007382031 1.237214e-02 -0.0130707491 3.664980e-03 -0.017104971 0.813993561 -0.0076250979 -1.464139e-02 -0.026493810 0.013597338 0.0123222475 0.01310805902 -0.007941486
V4 -6.730258e-02 -0.015745591 0.2788848e-02 -0.0117814649 4.127785e-02 0.002278841 0.002278841 0.002278841 0.002278841 0.002278841 0.002278841 0.002278841 0.002278841
V5  1.514325e-02 -0.005739754 -6.121892e-03 0.0045128218 5.433210e-03 0.0005548622 -0.018518277 -0.008391506 5.902556e-03 0.002751825 0.009012403 0.0399523132 0.00801994258 -0.007935540
V6 -3.397826e-02 0.033586699 0.1107013e-02 -0.0933261778 2.604523e-05 0.0141769088 0.0067276127 -0.001570883 2.553817e-02 0.0384170573 0.0118988680 -0.0180807858 0.0477093378 0.022818524
V7  1.314959e-02 0.0067794227 -3.653460e-02 0.033219931 4.94064e-02 0.025328818 0.02921940 0.000166171 0.000166171 0.000166171 0.000166171 0.000166171 0.000166171
V8 -1.273746e-02 -0.001273746 0.000166171 0.000166171 0.000166171 0.000166171 0.000166171 0.000166171 0.000166171 0.000166171 0.000166171 0.000166171 0.000166171
V9  4.916774e-02 0.031362089 0.165863e-02 0.164462e-01 0.0159534515 0.0340172249 0.0340172249 0.0340172249 0.0340172249 0.0340172249 0.0340172249 0.0340172249
V10 1.203660e-02 0.006331039 -1.25139e-02 -0.024174814 -2.706789e-02 0.0023647659 -0.019339945 0.0222774247 0.0529256183 -0.0389252821 -0.029166157
V11 2.767335e-02 0.037633958 -7.111640e-02 0.1515741831 -0.052191e-02 0.0595648752 -0.0358436247 -0.1682752238 0.1520353897 -0.0680381778 0.0477093378 0.0262770930
V12 1.707129e-02 0.161049972 -0.1711640e-01 0.0471039798 0.0471039798 0.0471039798 0.0471039798 0.0471039798 0.0471039798 0.0471039798 0.0471039798 0.0471039798
V13 2.441796e-04 -0.0680153860 -0.168016e-02 -0.0395959227 -0.480856e-01 0.019669461 0.0252357671 0.0631595227 0.04869945 0.049566387
```

Figura 25: Visualizzazione del calcolo della PCA tramite prcomop per il trainset delle domande nel database.

**Osservazioni** Osservando esclusivamente la deviazione standard ottenuta dal trainset di prova, appare come i PC da prendere in considerazione per l'analisi sono i primi tre; inoltre nella rotazione appare chiaro che le valutazioni ottenute da V1 sono in relazione con V4, V3 con V6 e V2 con V5. Tuttavia queste sono solo mere osservazioni senza ancora alcuna prova matematica completa a supporto.

Per quanto concerne il trainset delle domande nel database è molto difficile fare qualunque tipo di assunzione sulla natura dei dati causa la loro numerosità.

### 3.3.2 Calcolo degli autovettori ed individuazione di Summary

```
> summary(res.pca)
Importance of components:
                                         Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6
Standard deviation     1.9297771 1.0581456 0.9099156 0.38288957 0.34464930 0.244856110
Proportion of Variance 0.6209837 0.1867054 0.1380601 0.02444629 0.01980709 0.009997418
Cumulative Proportion 0.6209837 0.8076891 0.9457492 0.97019549 0.99000258 1.000000000
> # fornisce gli autovalori della matrice di covarianza/varianza delle dimensioni principali
> get_eig(res.pca)
eigenvalue variance.percent cumulative.variance.percent
Dim.1 3.72403947   62.0983737          62.09837
Dim.2 1.11967213   18.6705374          80.76891
Dim.3 0.82794633 13.8060084          94.57492
Dim.4 0.14660442  2.4446293          97.01955
Dim.5 0.11878314   1.9807094          99.00026
Dim.6 0.05995451   0.9997418          100.00000
```

Figura 26: Visualizzazione del calcolo del metodo summary ed individuazione degli autovalori per il trainset di prova.

```
> summary(res.pca)
Importance of components:
                                         Corp.1    Corp.2    Corp.3    Corp.4    Corp.5    Corp.6    Corp.7    Corp.8    Corp.9    Corp.10   Corp.11   Corp.12   Corp.13   Corp.14   Corp.15   Corp.16
Standard deviation     1.43136144 1.30933969 1.27234891 1.25540770 1.24609579 1.22206570 1.2114257 1.20907948 1.19278200 1.1790083 1.17845300 1.16204099 1.15753299 1.15663032 1.14998663
Proportion of Variance 0.02103668 0.01972807 0.01820813 0.01772265 0.01746071 0.01698565 0.01679377 0.01660811 0.01643872 0.01598587 0.01540521 0.01534701 0.01506496 0.01504464 0.014847688
Cumulative Proportion 0.02393868 0.04231675 0.06055249 0.07824758 0.09578028 0.11269334 0.12948710 0.14617321 0.16261151 0.17861659 0.19424280 0.22499832 0.24006328 0.25510814 0.26995692
                                         Corp.17   Corp.18   Corp.19   Corp.20   Corp.21   Corp.22   Corp.23   Corp.24   Corp.25   Corp.26   Corp.27   Corp.28   Corp.29   Corp.30   Corp.31   Corp.32
Standard deviation     1.10319196 1.12880856 1.12050001 1.11484441 1.10424641 1.09340403 1.08344441 1.07344441 1.06344441 1.05344441 1.04344441 1.03344441 1.02344441 1.01344441 1.00344441
Proportion of Variance 0.018452855 0.018453055 0.018453147 0.018453231 0.018453315 0.018453395 0.018453479 0.018453561 0.018453645 0.018453729 0.018453813 0.018453898 0.018453973 0.018454048 0.018454123
Cumulative Proportion 0.02452855 0.02985910 0.031307331 0.032704943 0.034081138 0.035457041 0.036153229 0.0368152687 0.037473653 0.040783165 0.042687653 0.04376981 0.044647318 0.045093812 0.047558093 0.048393076
                                         Corp.33   Corp.34   Corp.35   Corp.36   Corp.37   Corp.38   Corp.39   Corp.40   Corp.41   Corp.42   Corp.43   Corp.44   Corp.45   Corp.46   Corp.47   Corp.48
Standard deviation     1.04333725 1.04116609 1.02871449 1.02479858 1.02307959 1.01723836 1.015598054 1.00733966 1.00662121 0.99742479 0.99650091 0.99570404 0.99376459 0.99285548 0.97349201 0.71725294
Proportion of Variance 0.01224076 0.01218987 0.01190085 0.01180962 0.01177086 0.01159812 0.01149862 0.01139439 0.01118753 0.0111643 0.01114863 0.01114826 0.010956272 0.01084597 0.010656977
Cumulative Proportion 0.014334887 0.018198857 0.021988857 0.025888857 0.029788857 0.033688857 0.037588857 0.041488857 0.045388857 0.049288857 0.053188857 0.056988857 0.060888857 0.064788857 0.068688857 0.072588857
                                         Corp.49   Corp.50   Corp.51   Corp.52   Corp.53   Corp.54   Corp.55   Corp.56   Corp.57   Corp.58   Corp.59   Corp.60   Corp.61   Corp.62   Corp.63
Standard deviation     0.96534887 0.95952979 0.95755899 0.95339737 0.941994693 0.93825598 0.932384719 0.931159065 0.92139027 0.918259971 0.91639097 0.907151736 0.901228296 0.894532247 0.893069712
Proportion of Variance 0.018047919 0.01035323 0.01813874 0.01821132 0.009976378 0.009899143 0.009770454 0.009749859 0.00954468 0.009481603 0.00944323 0.009253773 0.009133317 0.008998101 0.00896704
Cumulative Proportion 0.67685352 0.68720863 0.69751937 0.70774666 0.717171968 0.727616203 0.737390257 0.747146163 0.756681203 0.766166203 0.775699203 0.784834634 0.794967961 0.80499419 0.810963523
                                         Corp.75   Corp.76   Corp.77   Corp.78
Standard deviation     0.887804612 0.880084930 0.877852651 0.871862307 0.865454738 0.859708007 0.856741253 0.847836174 0.846578159 0.837088957 0.829652320 0.817133935 0.812411088 0.808770840
Proportion of Variance 0.009885229 0.008707979 0.008649981 0.008547944 0.0089421851 0.008311147 0.00823881 0.008093110 0.008079564 0.00794849 0.00782846 0.007512879 0.007421824 0.007355447
Cumulative Proportion 0.820828816 0.829536615 0.8318186499 0.846734442 0.8515156293 0.863467440 0.871721324 0.879804515 0.887867164 0.895746728 0.903487576 0.911116222 0.918628301 0.926050125 0.933405572
                                         Corp.79   Corp.80   Corp.81   Corp.82   Corp.83   Corp.84   Corp.85   Corp.86   Corp.87   Corp.88   Corp.89
Standard deviation     0.79836916 0.796146684 0.782386719 0.772551946 0.759737073 0.755173864 0.745553834 0.728101546 0.717568688 0.699421206 0.663402160 0.624248461
Proportion of Variance 0.007167148 0.0071272629 0.0066883383 0.0067111419 0.0064949611 0.00641863 0.006217062 0.005961326 0.005798069 0.005402106 0.002438461
Cumulative Proportion 0.94057305 0.947786081 0.954584963 0.961295482 0.967786093 0.974198956 0.984410018 0.983377344 0.992187413 0.997569319 1.000000000
```

Figura 27: Visualizzazione del calcolo del metodo summary per il trainset delle domande nel database.

```

> get_eig(res.pca)
   eigenvalue variance.percent cumulative.variance.percent
Dim.1    2.0487956      2.3038680          2.303868
Dim.2    1.7143702      1.9278071          4.231675
Dim.3    1.6192262      1.8208178          6.052493
Dim.4    1.5760487      1.7722647          7.824758
Dim.5    1.5527547      1.7460707          9.570828
Dim.6    1.5104555      1.6985053         11.269334
Dim.7    1.4934446      1.6793765         12.948710
Dim.8    1.4838713      1.6686114         14.617321
Dim.9    1.4618708      1.6438718         16.261193
Dim.10   1.4227291      1.5998571         17.861050
Dim.11   1.3902728      1.5633600         19.424410
Dim.12   1.3699621      1.5405206         20.964931
Dim.13   1.3647870      1.5347012         22.499632
Dim.14   1.3398826      1.5066963         24.006328
Dim.15   1.3379168      1.5044857         25.510814
Dim.16   1.3204001      1.4847882         26.995602
Dim.17   1.2959135      1.4572531         28.452855
Dim.18   1.2743942      1.4330546         29.885910
Dim.19   1.2640485      1.4214209         31.307331
Dim.20   1.2428757      1.3976122         32.704943
Dim.21   1.2240074      1.3763948         34.081338
Dim.22   1.2233920      1.3757027         35.457041
Dim.23   1.2060378      1.3561879         36.813229
Dim.24   1.1911601      1.3394580         38.152687
Dim.25   1.1747105      1.3209604         39.473647
Dim.26   1.1645347      1.3095178         40.783165
Dim.27   1.1600618      1.3044880         42.087653
Dim.28   1.1465801      1.2893278         43.376981
Dim.29   1.1296923      1.2703376         44.647318
Dim.30   1.1166696      1.2556935         45.903012
Dim.31   1.1093668      1.2474815         47.150493
Dim.32   1.1050109      1.2425833         48.393076
Dim.33   1.0885526      1.2240760         49.617152

```

Figura 28: Individuazione degli autovalori per il trainset delle domande nel database.

**Osservazioni** È essenziale per individuare il numero di componenti (PC) necessarie per effettuare un’analisi corretta basarsi sul calcolo della *variance.percent* o *Proportion of Variance*. A tale scopo è prendere le componenti principali che catturano la maggior parte di variabilità dei dati.

Nel caso del trainset di prova basta le variabili PC1, PC2 e PC3 sono sufficienti a catturare il 93% della variabilità presente.

Quanto appena descritto si può riscontrare anche graficamente, come presentato di seguito.

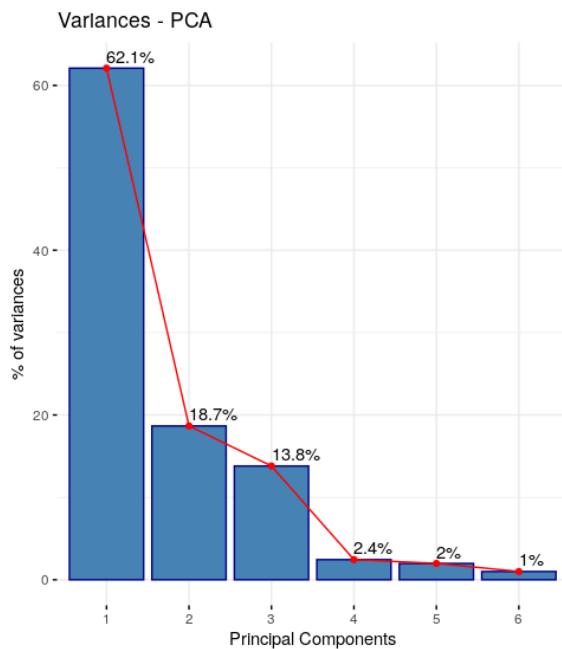


Figura 29: Plot della rappresentazione grafica della varianza sui PC del trainset di prova.

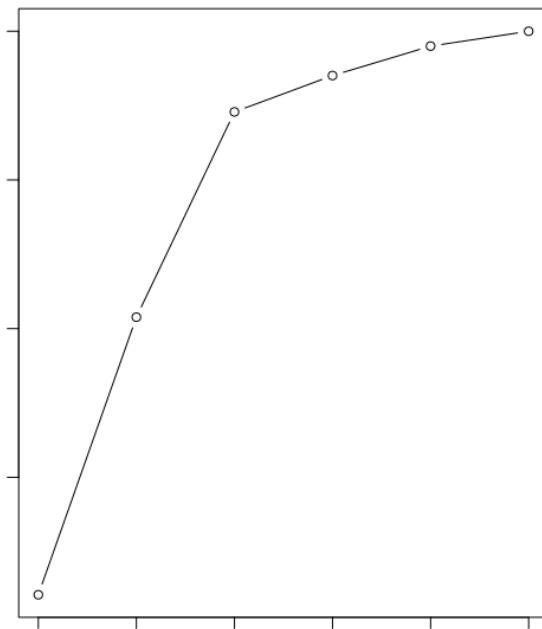


Figura 30: Plot della rappresentazione grafica della varianza sui PC del trainset di prova.

Le ultime PC4, PC5 e PC6 hanno una variabilità molto bassa, trascurabile.

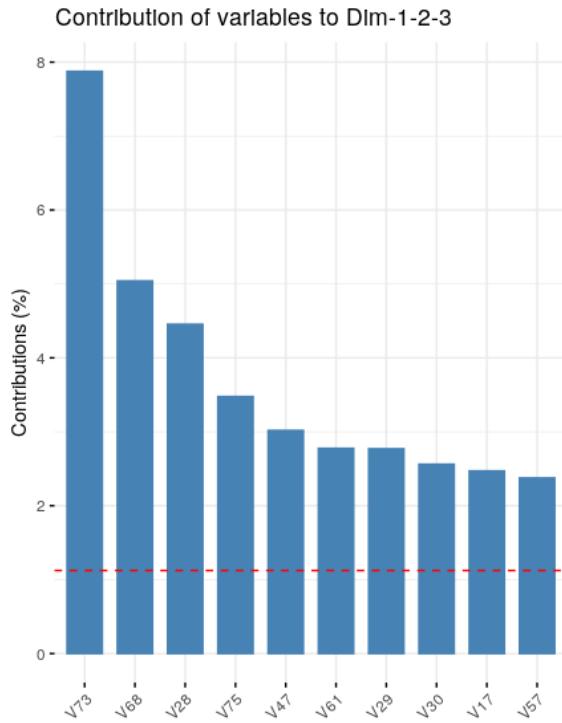


Figura 31: Rappresentazione grafica di come la varianza si distribuisce sulle PC individuate dal modello sul trainset di prova.

Tuttavia per quanto riguarda le domande nel database ogni conclusione "a occhio" risulta impossibile da effettuare sempre a causa della numerosità dei dati di trainset. L'utilizzo di un analisi dei risultati per mezzo di plot è l'unica via percorribile.

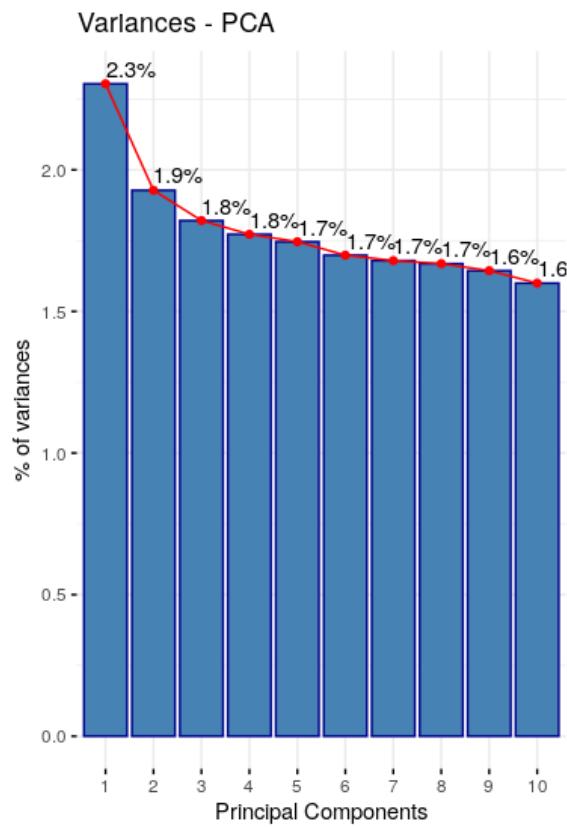


Figura 32: Plot della rappresentazione grafica della varianza dei primi dieci PC del trainset delle domande nel database.

Il plot mostra esclusivamente le prime dieci componenti; tutte si presentano con una varianza molto bassa. A causa di ciò per poter affermare quante PC sono indispensabili per una valutazione oggettiva dei dati è indispensabile avere una visione totalitaria di tutte variabili coinvolte nel modello.

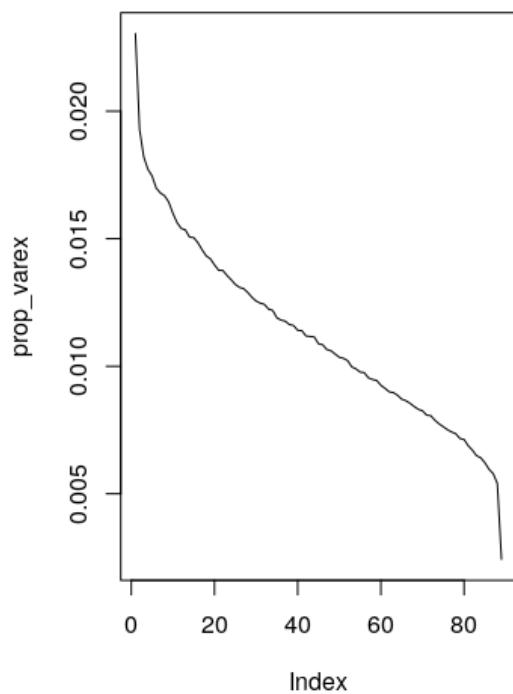


Figura 33: Plot della rappresentazione grafica della varianza di tutti i PC del trainset delle domande nel database.

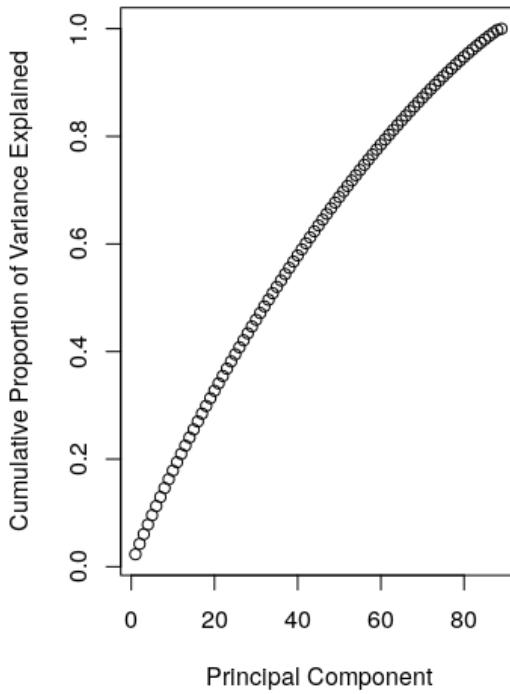


Figura 34: Plot della rappresentazione grafica della varianza di tutti i PC del trainset delle domande nel database.

Non risulta sufficiente l'andamento dei plot per riuscire a definire il numero adeguato di componenti principali da utilizzare. In questi casi, è buona norma, fare riferimento a tre criteri:

- *Quota della varianza totale*: si deve considerare un numero di CP tale che tenga conto di una percentuale sufficientemente elevata di varianza totale proporzionale al numero di variabili originarie (ovvero più è alto il numero di componenti del modello e più è accettata una percentuale minore di varianza spiegata).
- *Screen-graph*: fa uso dei plot degli autovalori in funzione al numero di CP. Gli autovalori sono decrescenti, per cui il grafico ha una buona possibilità, di assumerà la forma di una spezzata con pendenza negativa.
- *Eigenvalue one o Regola di Kaiser*: afferma di considerare tutte ed esclusivamente le CP con autovalore maggiore di 1.

Per soddisfare il *primo criterio* è sufficiente fare riferimento a *variance.percent* in *get\_eigen* o *Proportion of Variance* in *summary*. Da questa asserzione ne consegue un quesito: quale è il numero di varianza accettabile avendo un numero di variabili molto elevato?. Per effettuare una delle vie utilizzate è procedere al soddisfacimento del *terzo criterio*. Gli autovalori di tutte le componenti coinvolte si possono vedere nella *Standard deviation* risultante dalla *summary*. Le PC del modello, con autovalore superiore a 1, sono le PC contenute nell'intervallo 1-41. Il *secondo criterio*, invece, in questo caso specifico non è stato in grado di dirmi molto, in quanto la diminuzione degli autovalori è graduale, senza salti evidenti.

In conclusione, ho considerato come una percentuale di copertura adeguata quella fornita dalla che 41 prime componenti principali.

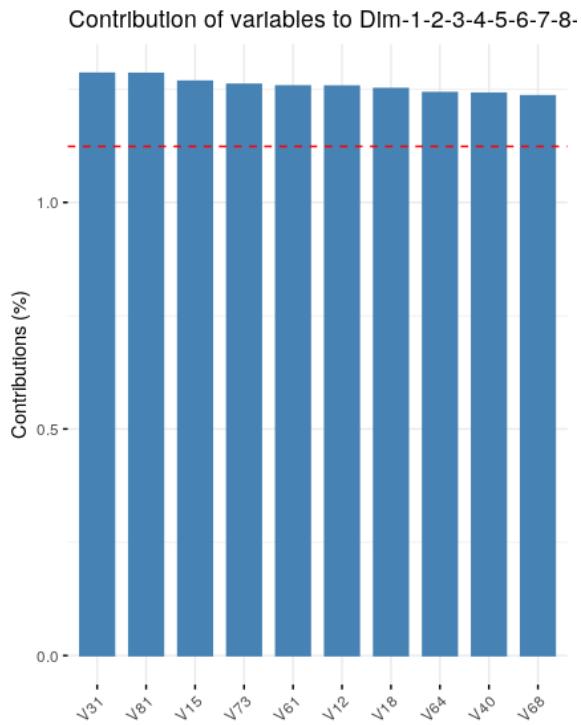


Figura 35: Rappresentazione grafica di come la varianza si distribuisce sulle PC individuate dal modello sul trainset delle domande nel database.

### 3.3.3 Calcolo degli autovettori

```
> loadings(res.pca)

Loadings:
  Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
V1  0.420    0.569  0.702
V2 -0.394   0.515  0.282      -0.707
V3  0.418   0.477 -0.278      -0.721
V4  0.421    0.565 -0.709
V5 -0.393   0.518  0.279      0.707
V6  0.402   0.489 -0.350      0.689

  Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
SS loadings   1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var 0.167  0.167  0.167  0.167  0.167  0.167
Cumulative Var 0.167  0.333  0.500  0.667  0.833  1.000
```

Figura 36: Autovettori del trainset di test.

**Osservazioni** L’analisi dei loadings sulle componenti principali permette di determinare il contributo delle variabili originarie al modello PC.

Per quanto riguarda l’analisi dei dati del trainset la prima variabile ha valori 0.420, 0.569 e 0.702, rispettivamente nelle componenti 1, 3 e 4 e sembra legarsi ai risultati nella variabile 4 (che riempie le medesime componenti oscillando di poco nei valori presentati), il medesimo match coinvolge le variabili 2 con 5 e 3 con 6.

Tali assunzioni sono molto più complesse da effettuare per il trainset dei dati delle domande nel databas e per fare ciò ho provveduto a calcolare la matrice correlazione.

### 3.3.4 Calcolo della matrice di correlazione

```
> cor.pca
          V1         V2         V3         V4         V5         V6
V1  1.0000000 -0.4725415  0.5354458  0.8534648 -0.4724833  0.4749618
V2 -0.4725415  1.0000000 -0.4025775 -0.4839185  0.8811501 -0.3892455
V3  0.5354458 -0.4025775  1.0000000  0.5237420 -0.3996431  0.9371832
V4  0.8534648 -0.4839185  0.5237420  1.0000000 -0.4838589  0.4736120
V5 -0.4724833  0.8811501 -0.3996431 -0.4838589  1.0000000 -0.3862259
V6  0.4749618 -0.3892455  0.9371832  0.4736120 -0.3862259  1.0000000
```

Figura 37: Autovettori del trainset di test.



con la domanda 1). Tale ragionamento vale per tutte le altre 5 componenti, i cui risultati rimangono coerenti con le aspettative e con ciò che viene dichiarato nel Grafo della Conoscenza, rappresentato nella figura ?? nella sezione § ??.

Pensavo di poter effettuare il medesimo ragionamento anche per analisi dei dati del trainset del database, tuttavia questo non è stato possibile, a causa delle seguenti motivazioni:

- In primo luogo, se una variabile nella matrice correlazione è dichiarata in correlazione con una componente, questa poi non risulta in correlazione con la variabile;
- Analizzando il testo delle domande, contenute nel database aziendale ho riscontrato come il modello prodotto crea relazioni strette fra domande appartenenti trivialmente a categorie diverse, (ad esempio domande che trattano relazioni con altre di serie numeriche) e meno con domande che parlano dello stesso tema.
- Dal pot delle variabili viene mostrato come le stesse indicate come correlate risultano, invece, sparse impedendo la formazione dei cluster. Su un numero n grande di variabili (come le 89 domande nel database) tale fenomeno verrebbe espanso impedendo a chi interpreta il modello di individuare una correlazione tra il grafico delle variabili e la matrice di correlazione.

A questo punto ho iniziato a farmi delle domande. Escludendo un mio errore di codifica (dopo aver opportunamente effettuato un accurato controllo del codice da me prodotto) e trovandomi di fronte ad una situazione dove i risultati inerenti ai dati del trainset di prova soddisfano appieno le attese (infatti non solo le correlazioni matchano perfettamente con quanto viene indicato dal grafo della conoscenza; ma anche il plot che viene generato dalla PCA sulle variabili presenta una coerenza stretta con tali assunzioni) ho iniziato a pensare che centrasse la possibilità che ha un candidato di indovinare correttamente una risposta ad una domanda.

Ho rifatto, perciò, tutta l'analisi su un nuovo modello, adoperando il trainset di prova generato da n input sottoposti alla probabilità di indovinare. Come viene illustrato dai plots nella sezione seguente, i dati individuati dal trainset di prova puro vengono totalmente falsati quando si tiene conto di tale eventualità rispetto alle aspettative imposte dal Grafo della conoscenza.

### 3.4 Conclusione dell'analisi

In conclusione, un test con domande a triplice risposta multipla concede ad un candidato l'elevata possibilità di indovinare le domande che non sa; e il

modello creato dalla Principal Component Analysis con i dati delle risposte alle domande nel database ne fornisce la prova.

Infatti se la possibilità di indovinare fosse bassa i risultati del modello in esame ne verrebbero appena "sporcati", e invece invalidano completamente la possibilità di avere un risultato attendibile e coerente con la realtà. Le figure seguenti mostrano proprio tale fenomeno: nel caso dell'indovinato le domande con una maggiore correlazione non creano alcun cluster.

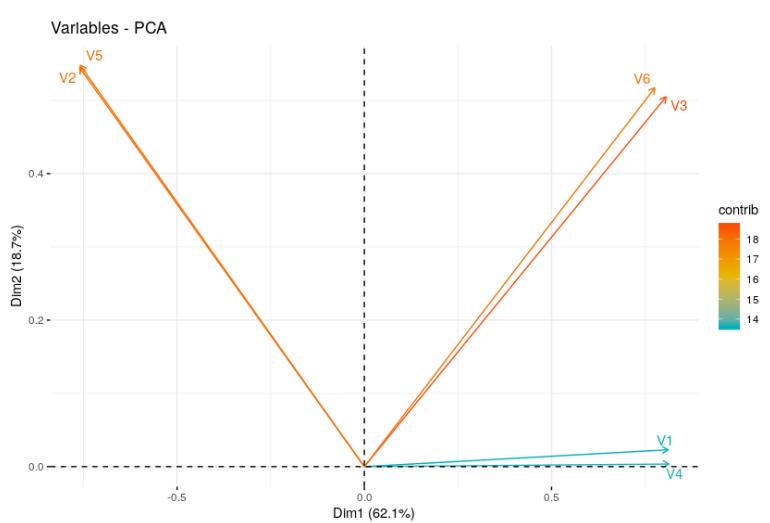


Figura 41: Rappresentazione per mezzo di plot di come le variabili si presentano nelle due componenti principali con il calcolo della PCA - utilizzo di trainset di prova puro sul grafo della conoscenza.

```
> correlazione
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    4    3    6    5    2
[2,]    2    5    6    3    1    4
[3,]    3    6    1    4    5    2
[4,]    4    1    3    6    5    2
[5,]    5    2    6    3    1    4
[6,]
```

Figura 42: Le domande in correlazione con le variabili principali in ordine decrescente sui dati di prova.

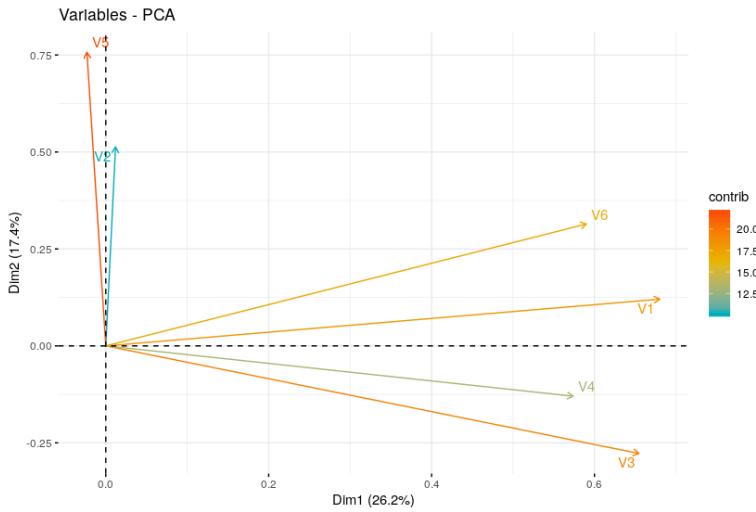


Figura 43: Rappresentazione per mezzo di plot di come le variabili si presentano nelle due componenti principali con il calcolo della PCA - utilizzo di trainset di prova spurio con la probabilità di indovinare.

```
> correlazione
[,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    6    3    4    5    2
[2,]    2    6    5    4    1    3
[3,]    3    1    4    6    2    5
[4,]    4    3    6    1    2    5
[5,]    5    6    2    1    4    3
[6,]    6    1    4    3    5    2
```

Figura 44: Le domande in correlazione con le variabili principali in ordine decrescente sui dati di prova soggetti alla probabilità di indovinare.

L'analisi del modello ha evidenziato come il Reticolo della Conoscenza costruito dai dati delle domande nel database, una volta costruito, risentirà inevitabilmente del fattore di rischio "domande indovinate". Mi aspetto, per cui, che in rapporto al contenuto delle stesse gli argomenti e la difficoltà delle domande appartenenti al medesimo cluster possano non essere coincidenti.

L'unica possibilità per continuare la costruzione del reticolo è rivolgere nuovamente l'attenzione alla Rete neurale e studiare i cluster possibili per mezzo delle previsioni assunte da ogni variabile sulla base della previsione standard<sup>3</sup>.

---

<sup>3</sup>Vettore previsione tutto a zero

## 4 Costruzione del Reticolo della Conoscenza

Dal 26/07 al 28/07 ho iniziato la Costruzione del Reticolo della Conoscenza. Per realizzarlo ho utilizzato un'applicativo sviluppato durante un precedente stage dell'azienda, che fa largo uso della libreria d3.js; effettuando elaborazione di dati presentandoli in *Cluster Based* o *Force Based*, a discrezione delle esigenze dell'utente.

L'attività di creazione, documentazione e test del Reticolo è durata fino al termine dello stage.

### 4.1 Descrizione del sistema

L'applicazione accetta in import file di estensione CSV. Ogni colonna di quest'ultimo viene interpretata dal sistema come un parametro da elaborare; per questo la prima riga del file deve essere o preceduta da una dichiarazione di variabili, tante quante sono le colonne da parametrizzare, oppure è questa prima riga che viene interpretata come una dichiarazione e conseguentemente non rappresentata all'interno del Reticolo.

Nel sistema possono venire settati i seguenti aspetti:

- La *tipologie* di Reticolo:
  - Cluster Based: raggruppa un insieme di oggetti in modo tale che gli tutti gli elementi contenuti nel medesimo cluster sono più simili l'uno all'altro rispetto a quelli contenuti in altri gruppi;
  - Force Based: in base alla forza di ogni nodo viene rappresentata come unica regione compatta le istanze appartenenti alla medesima classe in cui vengono visivamente identificati i percorsi di differenziazione. Nel layout le celle differenzianti sono poste in prossimità della classe più fortemente correlata.
- *Normalizzazione* dei dati in input:
  - No: non viene applicata alcuna tecnica di normalizzazione dei dati;
  - MinMax: i dati vengono ridimensionati su un intervallo specifico ( $\min$ ,  $\max$ ), tuttavia tale tecnica non è in grado di gestire i valori anomali;
  - Gaussian: o normale in cui i dati vengono normalizzati in una curva in cui i valori della stessa grandezza sono soggetti ad approssimazione;
  - Interquartile: si occupa di standardizzare i dati in modo da quantificare l'estensione del 50% della distribuzione del carattere che si trovano attorno alla mediana;

- Tipologia di *distanza* applicabili ai punti:
  - Euclidea: tiene conto della distanza tra i punti;
  - Camberra: tiene conto della distanza tra le coppie in uno spazio vettoriale;
  - Pearson: distanza di correlazione che misura il grado di correlazione tra due punti. Valuta la covarianza tra due variabili in rapporto al prodotto della deviazione standard. Non è vantaggiosa su dati semplici.
- *Metodo* di associazione dei punti:
  - Single: ”vicino al prossimo”, la distanza fra i gruppi è posta al pari della più piccola delle distanze calcolabili a due a due tra tutti gli elementi del gruppo. Accentua tutte le somiglianze tra i gruppi a discapito della loro differenziazione netta.
  - Complete: ”vicino più lontano”, viene considerata la maggiore tra le distanze calcolate a due a due tra gli elementi di due gruppi. Privilegia la differenziazione tra i gruppi a discapito dell’omogeneità degli elementi in essi contenuti. In questo caso i punti vengono rappresentati come meno compatti e diluiti.
  - Average: viene considerata come distanza fra due gruppi la media fra tutte le distanze calcolate a due a due tra gli elementi dei due gruppi. I risultati ottenibili sono i più attendibili (essendo basato sulla media delle distanze), i gruppi risultano più omogenei e differenziati tra di loro.

Un’ulteriore funzionalità permette all’utente di decidere se si desidera procedere con una rappresentazione del Reticolo manuale o automatica progressiva dei dati.

#### 4.1.1 Configurazione

Analizzando l’applicativo in base al carattere dei dati in ingresso e alle aspettative sull’output del modello, ho riscontrato che la configurazione necessaria per la formazione del Reticolo della Conoscenza è vincolata alla dischiarazione delle seguenti proprietà:

- *Redistance*: No;
- *Normalize*: No;
- *Distance-Type*: Euclidea;
- *Method*: Single.

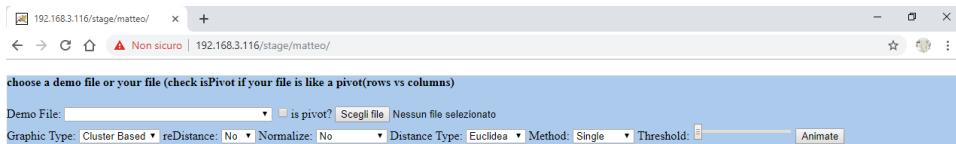


Figura 45: Configurazione usata nel sistema per generare il Reticolo della Conoscenza.

## 4.2 Creazione dei file CSV

Come già accentato all'interno della sezione §Descrizione del sistema prima di procedere alla creazione del Reticolo ho dovuto preparare i dati di previsione prima di poterli dare in pasto al sistema. Questo è stato reso più agevole grazie alla creazione, da parte mia, di un metodo che ha il compito, una volta messa in funzione la Rete neurale oggetto di studio (di prova o del database), di calcolare:

1. Le previsioni ottenibili da un vettore di previsione settato a 1 o -1 per ogni singolo elemento;
2. Sui dati del punto (1) un vettore delle differenze dove viene calcolato il delta in rapporto al vettore di standard <sup>4</sup>.

Ho fatto in modo che il vettore delle differenze venga stampato su console del browser <sup>5</sup>, in modo che ne basti prelevare il contenuto e inserirlo su un file CSV. Ogni elemento per poter funzionare all'interno dell'applicativo deve essere separato da un ; e ogni riga di previsione deve essere preceduta dal codice della domande in modo da rendere più agevole l'interpretazione del Reticolo. È a discrezione dell'utente l'inserimento di un'ulteriore riga di dichiarazione dei parametri.

## 4.3 Creazione del Reticolo della Conoscenza per sui dati di Prova

### 4.3.1 L'architettura della Rete

Durante la costruzione del Reticolo della Conoscenza è stato importante capire se effettivamente l'architettura scelta in precedenza fosse corretta, e il suo vero significato. La rete sviluppata per il progetto è una Rete neurale statica a multistrato, nella quale sono presenti i seguenti elementi:

---

<sup>4</sup>Vettore tutto a zero

<sup>5</sup>unica alternativa facendo uso di solo codice javascript

- input layer: dove viene dichiarata la dimensione dell'input: lunghezza, larghezzae profondità dei dati da analizzare;
- hidden layer ( o layer intermedio): composto da uno o pi' u strati nascosti (o intermedi) costituito da m neuroni;
- output layer: costituito da p neuroni pari al numero di output desiderati.

Una Rete neurale permette di formulare complessi modelli matematici (sistemi di equazioni differenziali ecc.) e di individuarne possibili soluzioni, inoltre le soluzioni si determinano a partire dai dati; tuttavia la scelta del modello è un'azione molto complessa.

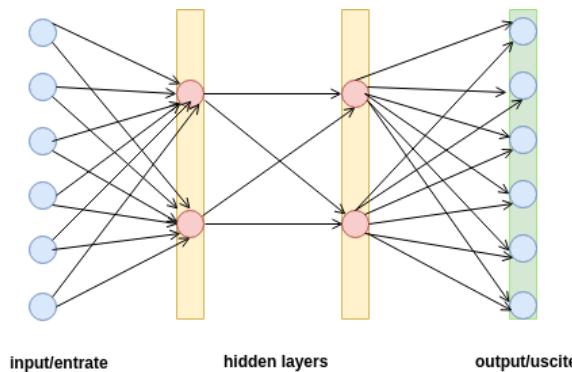


Figura 46: Archittettura delle Rete di prova: 6 ingressi in una rete a 3 strati con 2 stati nascosti e uscite.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0.1311737561160600	0.3285001848958143	0.0718524307568600	0.1380571123080200	0.3126662017956307	0.0670773376353959	0.125091758230576	0.2819274585022983	0.05645039030947080	0.1269235506446900	0.285727398302305	0.011564865009118674	
2	2.022344867572622770	0.4040430173839432	0.161244992748595	0.23005715403479	0.4861369876446212	0.1527138789739739	0.2683952413226600	0.545300456625983	0.1915968861708868	0.270968600451429	0.157248188579408	0.1804563900253014	
3	3.0159886030261298	0.111430322467592	0.6835251671568954	0.1511047420535179	0.12772266795852643	0.04567711969329	0.109517906872584	0.1305539493313490	0.5070337812452888	0.0936656242975840	0.140706413661771	0.479042657547656	
4	4.0159886030261298	0.111430322467592	0.6835251671568954	0.1511047420535179	0.12772266795852643	0.04567711969329	0.109517906872584	0.1305539493313490	0.5070337812452888	0.0936656242975840	0.140706413661771	0.479042657547656	
5	5.00676117890759713	0.4899828446991023	0.39515567526995	0.07622330787893	0.5045525781450591	0.3736855419971749	0.1267747481826263	0.5837897139900635	0.314141122919062	0.1352582210543210	0.5986903026496474	0.29722593020335	
6	6.0222225093915569	0.02844892865708170	0.8097741507867975	0.2187954013961559	0.0458081488764989	0.764843310309213	0.1518572481038853	0.0465793241271879	0.574045913935048249	0.05634067585657129	0.5422182071530703		
7													

Figura 47: file CSV generato per la creazione del Reticolo della Conoscenza sui dati di prova.

*Di seguito sono riportate le sequenze di creazione del Reticolo della Conoscenza per i dati di prova.*

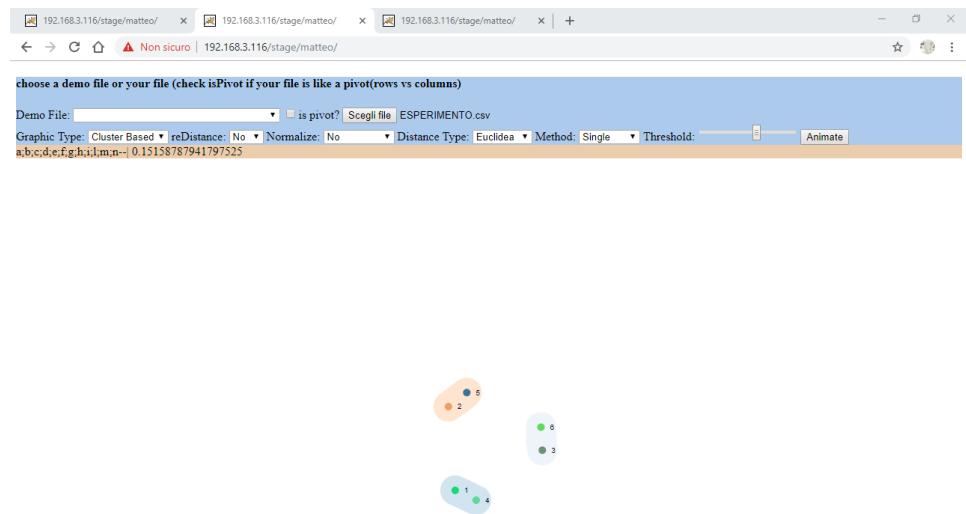


Figura 48: Reticolo della Conoscenza con rappresentazione di tipo Cluster Based - 1.

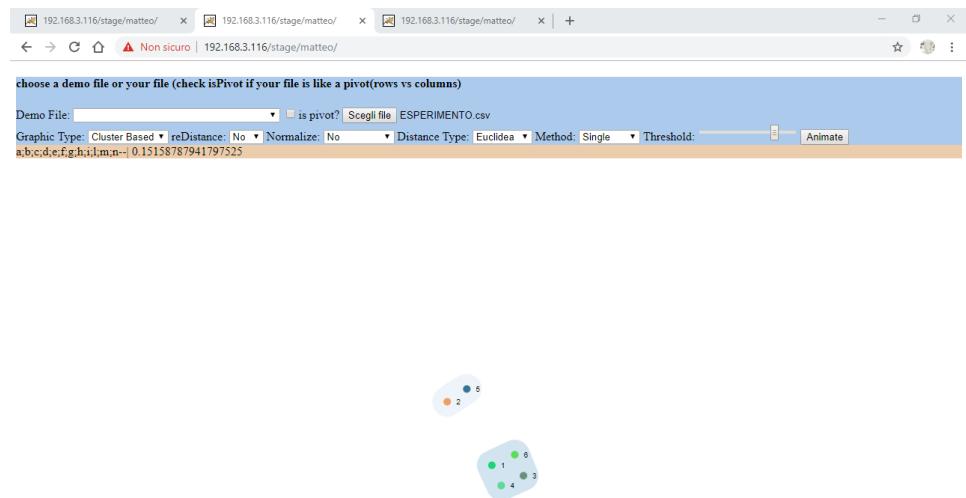


Figura 49: Reticolo della Conoscenza con rappresentazione di tipo Cluster Based - 2.

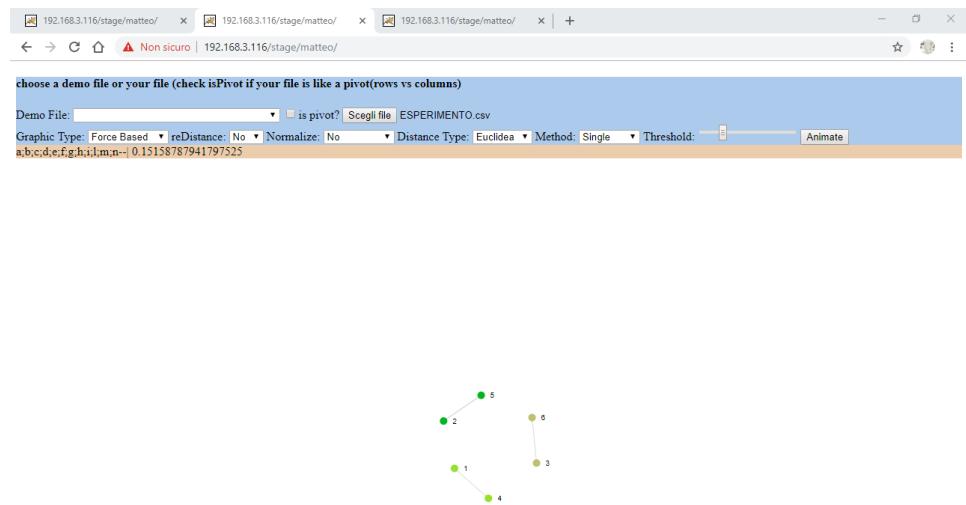


Figura 50: Reticolo della Conoscenza con rappresentazione di tipo Forced Based - 1.

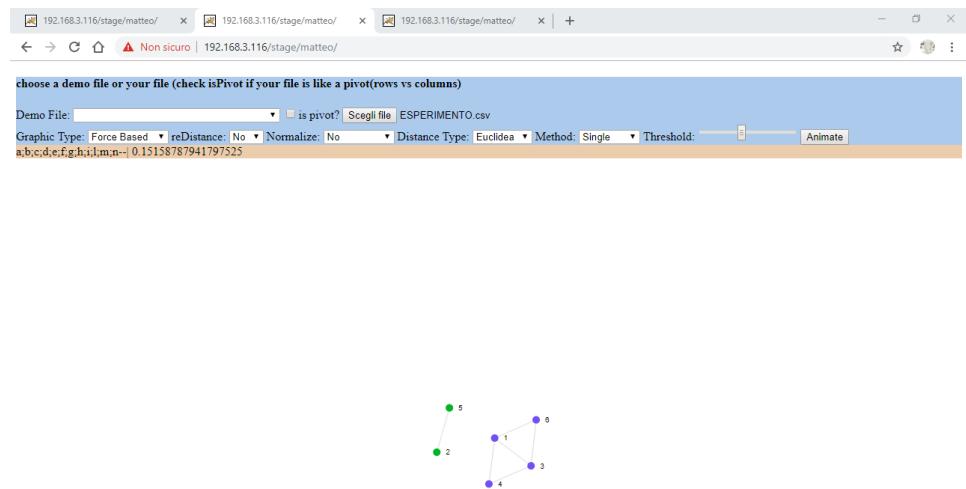


Figura 51: Reticolo della Conoscenza con rappresentazione di tipo Forced Based - 2.

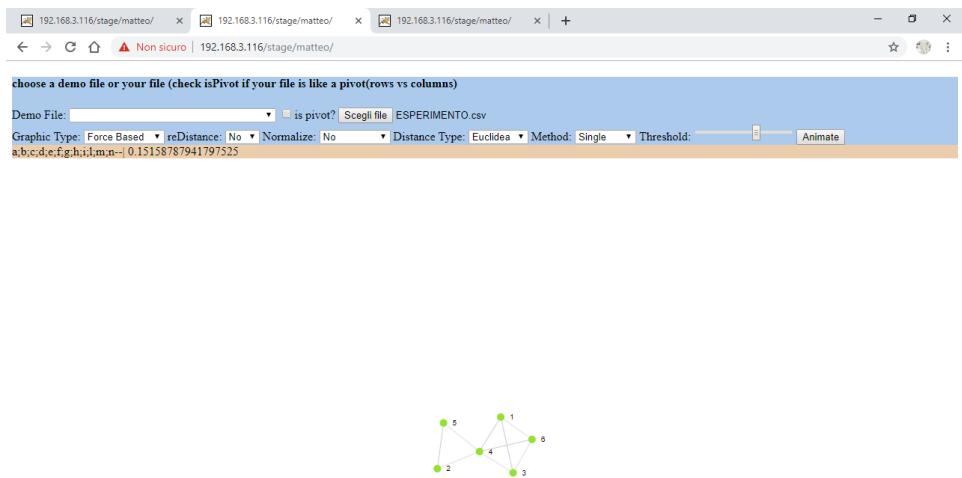


Figura 52: Reticolo della Conoscenza con rappresentazione di tipo Forced Based - 3.

*Appare evidente come il Reticolo generato rispetta quando definito dalla figura ??.*

*Tuttavia effettuando delle ulteriori prove con file dati differenti; ma provenienti dalla medesima Rete neurale ho riscontrato situazioni contrastanti. L'errore osservato nei casi non corretti riguarda le coppie di domande 1, 4 e 3, 6. Un esempio del fenomeno è illustrato di seguito.*

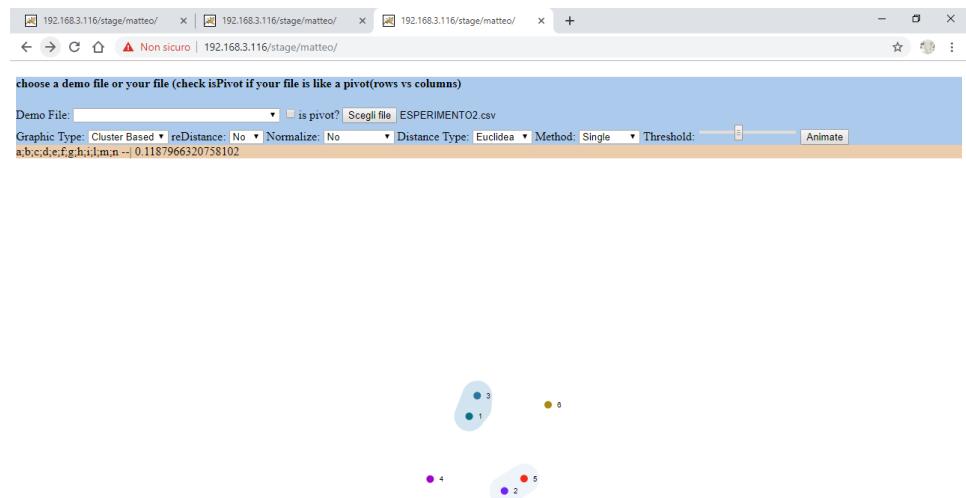


Figura 53: Reticolo della Conoscenza con rappresentazione di tipo Forced Based - 3.

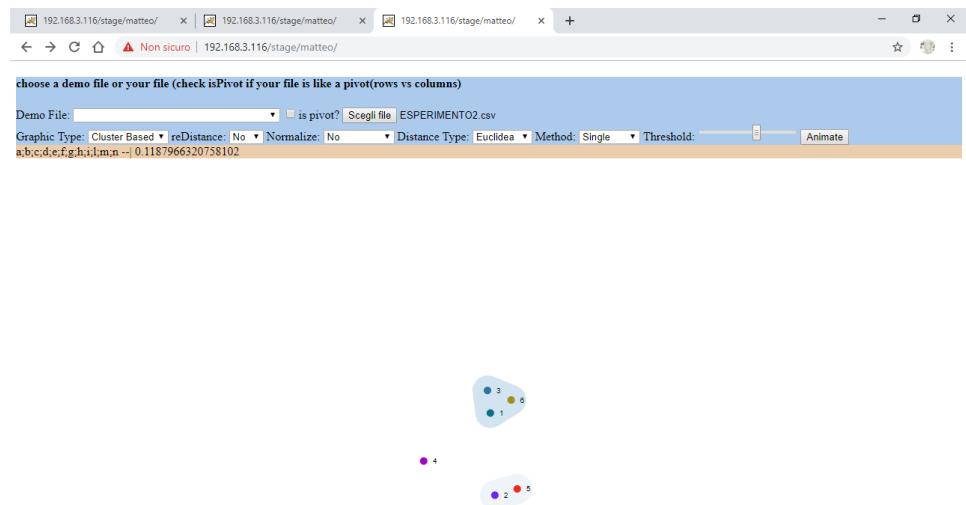


Figura 54: Reticolo della Conoscenza con rappresentazione di tipo Cluster Based - 1.

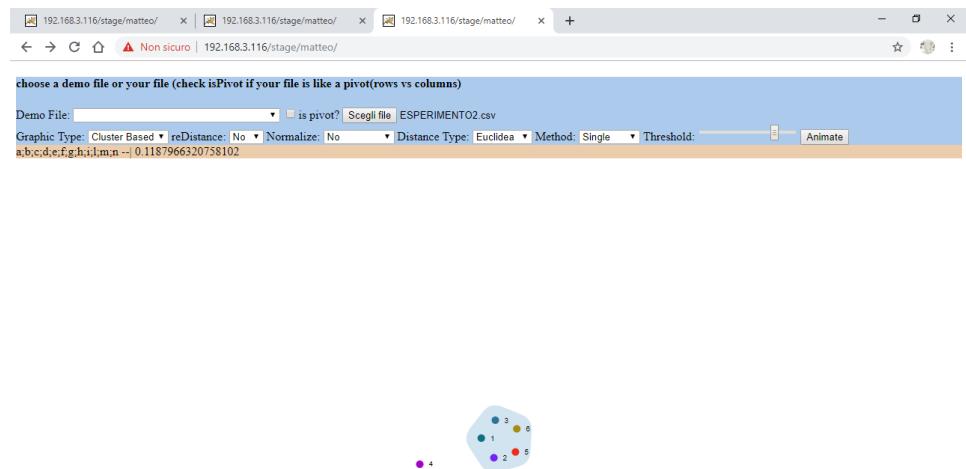


Figura 55: Reticolo della Conoscenza con rappresentazione di tipo Cluster Based - 2.

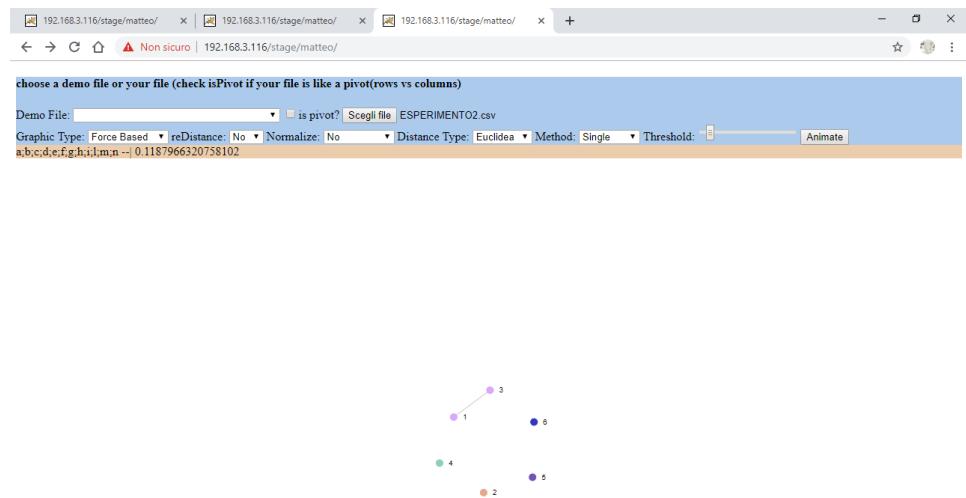


Figura 56: Reticolo della Conoscenza con rappresentazione di tipo Cluster Based - 3.

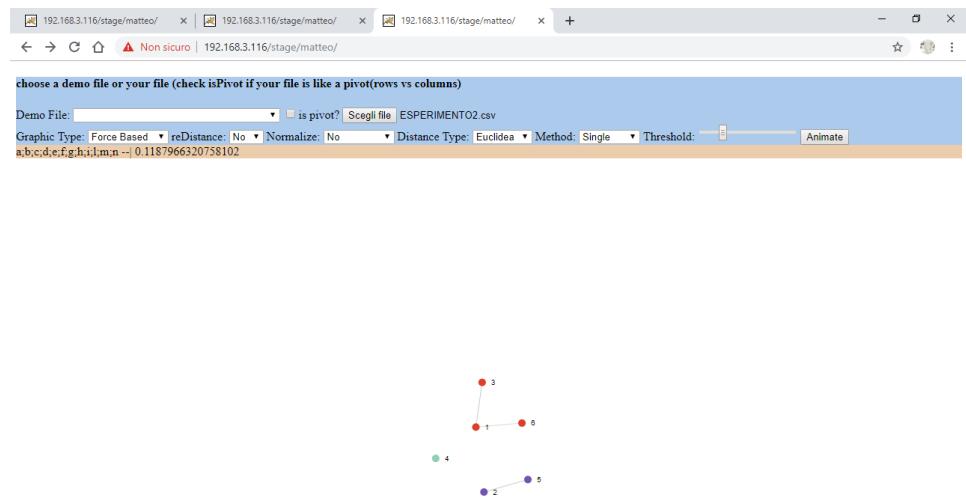


Figura 57: Reticolo della Conoscenza con rappresentazione di tipo Forced Based - 1.

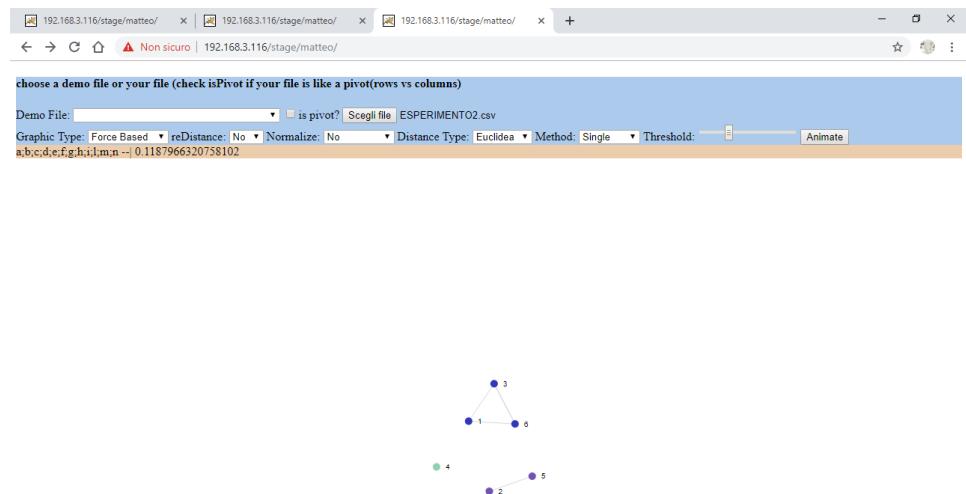


Figura 58: Reticolo della Conoscenza con rappresentazione di tipo Forced Based - 2.

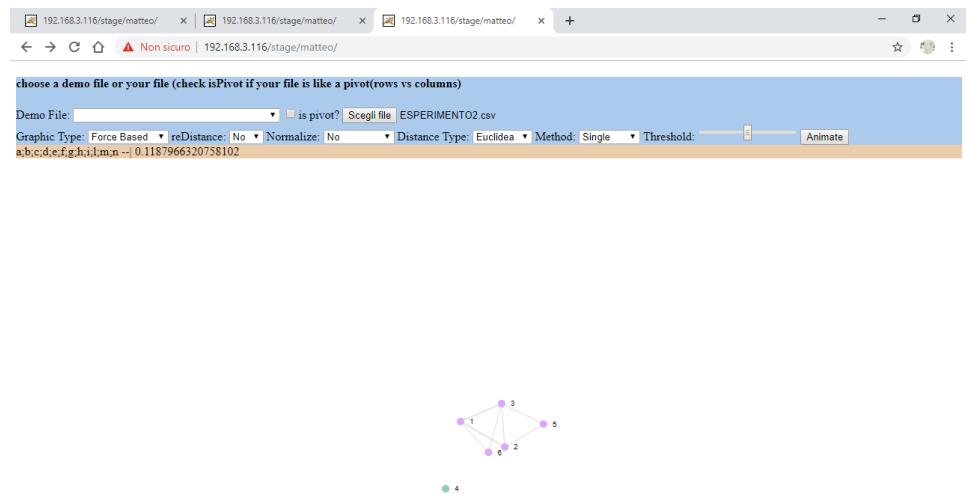


Figura 59: Reticolo della Conoscenza con rappresentazione di tipo Forced Based - 3.

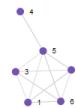
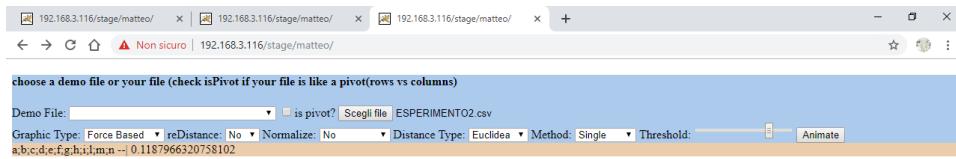


Figura 60: Reticolo della Conoscenza con rappresentazione di tipo Forced Based - 4.

La spiegazione del Reticolo malformato è da ricondurre all'applicativo usato per la sua rappresentazione. I punti vengono raggruppati, come spiegato nella sezione § ?? assieme usando la distanza euclidea, ciò implica che per effettuare i raggruppamenti viene calcolata la distanza che intercorre fra i punti dati in pasto al sistema. Fra due punti viene calcolata la distanza bidimensionale (TODO:da verificare) ovvero la radice quadrata delle differenze nei due assi, sommate tra di loro. Nel momento in cui il gruppo viene creato il sistema procede all'individuazione del prossimo gruppo, effettuata mediante l'uso del metodo Complete. Significa che vengono presi tutti i punti in relazione stretta con i punti del gruppo in esame e scelto il più vicino; escludendo perciò da qualsiasi altra considerazione presenta e futura gli ulteriori punti coinvolti, se non solo quando chiamati in causa da altri punti.

In fase di test mi sono anche occupata di capire come si comporta la Rete neurale, relativa ai dati di prova, considerando esclusivamente la possibilità che un candidato o risponda correttamente ad una domanda (1) o la sbagli (-1). I risultati ottenuti sono i seguenti:

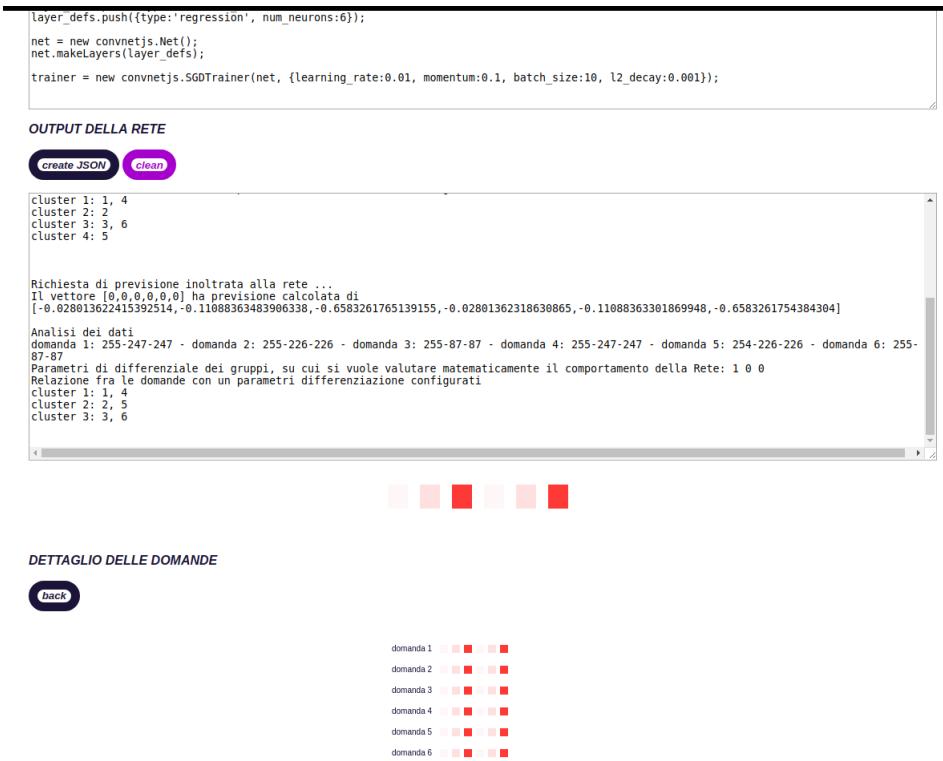


Figura 61: Risultati della Rete di test con un set di dati puro (esclusivamente 1 e -1).

Come si può vedere dalla figura ?? le previsioni ottenute per le coppie di domande (1;4), (3;6) e (2;5) sono identiche, ad eccezioni di alcune variazioni impercettibili da ricondurre ad oscillazioni di Rete e che non hanno alcun impatto sui risultati da raggiungere. Chiedendo una previsione sul valore 0 e avendo allenato la Rete esclusivamente con valori -1 e 1 mi aspetto che si venga effettuata la media, ed è quello che sembra venga fatto per tutte le domande coinvolte. Il fenomeno sorprendente, che mette in luce l'importanza di avere un valore 0 di risposta (non data) nel trainset, è mostrato nelle figure sotto.

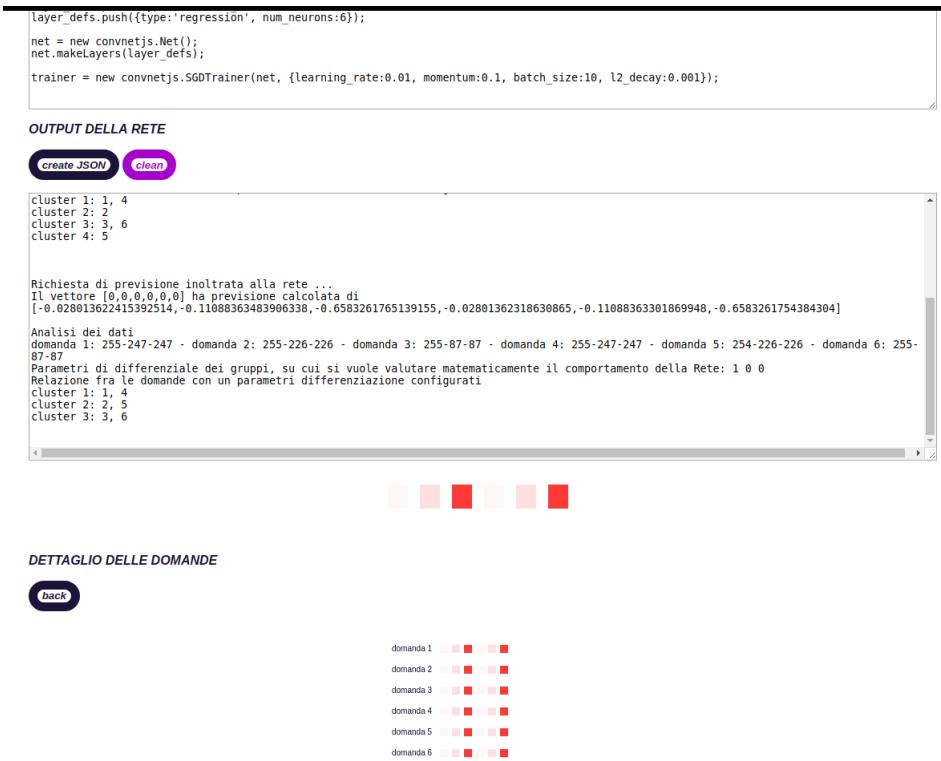


Figura 62: Risultati della Rete di test con un set di dati puro (esclusivamente 1 e -1), vettore previsione [-1, 0, 0, 0, 0, 0].

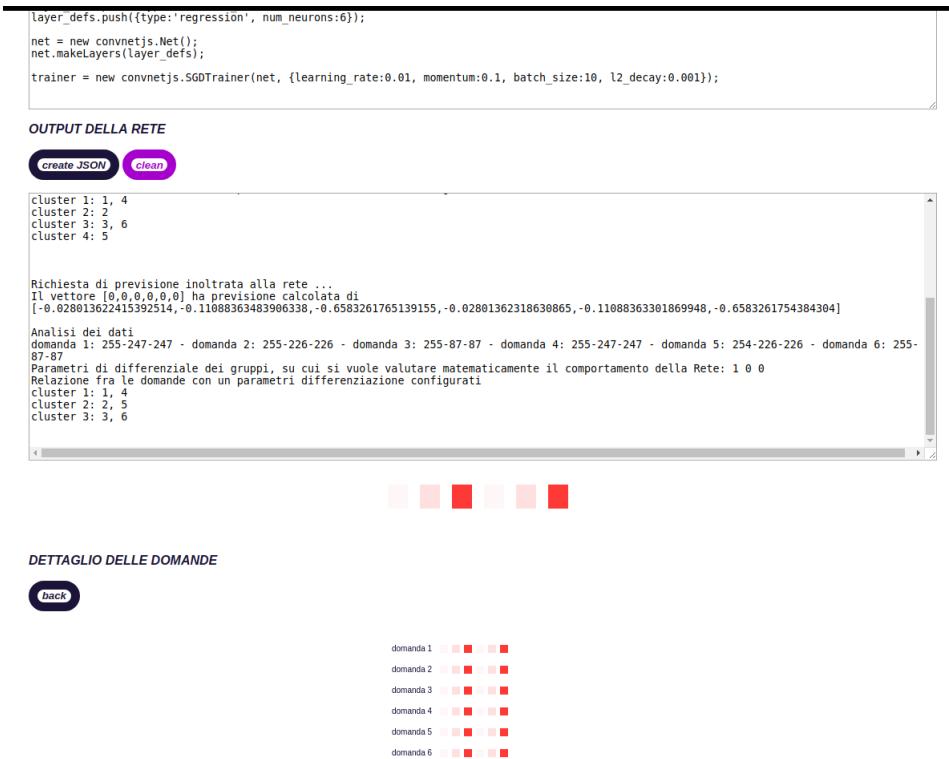


Figura 63: Risultati della Rete di test con un set di dati puro (esclusivamente 1 e -1), vettore previsione [0, 0, 0, -1, 0, 0].

Non avere un valore a 0 porta le le previsioni imposte alle singole domande, valutate in coppia, a mostrare come unica relazione stretta esistente la coppia coinvolta, scollegando ogni relazione con le altre coppie. Come si vede dall'esempio sopra, le domande 2 e 4 vengono colorate di verdino nella previsione della domanda 1 e di rosso nella domanda 4. Questo ha un impatto sulla previsione visiva della Rete, dal quale risulta impossibile la previsione di domande figlie e genitori.

La matrice correlazione ottenuta, tuttavia dal modello della PCA, risulta perfettamente allineata con le aspettative delle coppie di domande e delle previsioni, tuttavia questo è possibile solo perché i dati e le previsioni vengono valutati nella loro generalità e non tra coppia di domande. (TODO: manca SPIEGAZIONE 1 RETICOLO)

#### 4.3.2 Osservazioni

#### 4.3.3 Osservazioni Reticolo dati di prova

In conclusione il Reticolo della Conoscenza anche se non sempre perfettamente coerente con le aspettative; nella maggioranza dei casi ricalca fedel-



rete con 6, 8, 10, 12 neuroni ciascuno. Questo a causa della non conoscenza a priori del numero di cluster che comporranno il Reticolo, come invece accade per i dati di prova.

Tale differenziazione ha permesso durante la configurazione del sistema e la creazione del Reticolo che mi accorgersi dell'estrema variabilità dei cluster e correlazioni a seconda dell'oscillazione del numero di neuroni per layers, come viene evidenziato dalle immagini seguenti.

TODO: INSERIMENTO DEI SCREEN DEL RETICOLO SUI CASI A 6, 8, 10, 12 NEURONI A CLUSTER E A FORCE BASED.

## 4.5 Osservazioni

### 4.5.1 Osservazioni Reticolo dati del database

La variazione dei Reticoli ottenuti dalle diverse architetture della Rete neurale indica come sia necessario approfondire la tematica e individuare una strategia unica che permetta di individuare in modo univoco l'architettura necessaria per ottenere un Reticolo della Conoscenza valido.

## 4.6 Creazione del Reticolo prendendo in considerazione le frequenze alle domande

Ho ritenuto adeguato studiare il comportamento dei set di dati in base alla loro frequenza; in modo da poterne valutare le differenze con la Rete neurale e se vi fossero benefici o meno.

Come primo passo ho provveduto a calcolarmi per ogni domanda quale fosse la frequenza delle altre domande valutate in accordo e in disaccordo con la risposta data. Ho effettuato il procedimento sia nel caso di una domanda risposta correttamente che in modo sbagliato.

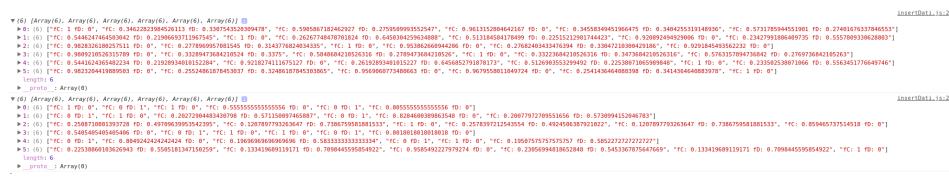


Figura 65: Frequenza delle domande dei dati di test.

L'immagine sopra mostra per ogni domanda valutata a 1 (primo set di dati) e a -1 (secondo set di dati) la frequenza che intercorre in tutte le domande con segno coerente o opposto a quello della domanda in esame. Il set di dati viene generato randomicamente con guida<sup>6</sup>. Analizzando i risultati di frequenza osservati ho riscontrato i seguenti fenomeni:

<sup>6</sup>con impiego del Grafo della Conoscenza (mostrato in figura ??)

- Considerando i valori per riga:
  - per le domande in cui le risposte si presentano corrette i valori di frequenza correlata e opposta si presentano con la seguente struttura:
    - \* la domanda 3 si presenta con una frequenza correlata che coinvolge le coppie (1,3) e (4,6);
    - \* la domanda 6 si presenta con una frequenza correlata che coinvolge le coppie (1,6) e (4,3).
  - per le domande in cui le risposte si presentano sbagliate i valori di frequenza correlata e opposta si presentano con la seguente struttura:
    - \* la domanda 1 si presenta con una frequenza correlata che coinvolge le coppie (1 FC:1, 3 FC:1) e (4 FC:0.55, 6 FC:0.80);
    - \* la domanda 4 si presenta con una frequenza correlata che coinvolge le coppie (4 FC:1, 3 FC:1) e (1 FC:0.54, 6 FC:0.80);
- Considerando i valori per colonna:
  - la domanda 1 si correla con la domanda 4;
  - la domanda 3 si correla con la domanda 6,
  - la domanda 2 con la domanda 5.

Tali considerazioni non sono esclusive per il singolo set di dati; ma rimangono costanti per ogni set di dati generato (anche se di natura parzialmente randomica). Ho riscontrato per ciascuna variabile al massimo un'oscillazione che si attesta mai superiore a 0.2. Perciò tali risultati risultano stabili e attendibili.

I valori di frequenza possono concorrere alla formazione del Reticolo della Conoscenza. Per renderlo possibile ho provveduto a creare un file csv che contenesse per ognuna delle domande tutte le domande coinvolte valutate con frequenze positive correlate e opposte e le frequenze negative correlate e opposte. La tecnica permette la generazione, per ogni domanda, di un vettore contenente un quantità di dati 4 volte la dimensione dell'input/output della Rete. Il divario permette, positivamente, di poter costruire un Reticolo ove i dati per ogni nodo hanno una stabilità concreta.

Tuttavia il Reticolo generato riscontra delle peculiarità:

- la frequenza non effettua alcuna riduzione dimensionale provocando uno utilizzo di spazio pari al numero di domande/dati da analizzare. Ciò comporta che per grandi moli da dati lo spazio necessario cresce elevato al numero di elementi di input;

- la frequenza funziona male sui valori anomali, come già spiegato nella sezione § ??

Il divario tra i valori per riga e colonna sono legittimati dalla natura del set di valori di input. In esso sono contenuti valori 0, e non esclusivamente -1 e 1, causa diretta dell'oscillazione delle correlazioni strette esistente tra le domande accoppiate<sup>7</sup>. La prova di quanto appena detto è evidente testando più volte la Rete. Per un n molto piccolo le previsioni, ottenute, risultano così vicine l'una all'altra che almeno 4 domande formano un unico cluster. Tale curvatura dei dati ha il suo impatto peggiore sulle domande figlie e genitori. Le domande estranee vengono associate prima dai figli esclusivamente perché molto semplici. Tuttavia l'oscillazione in esame è minima e il Reticolo ne risente esclusivamente quando i valori di previsione delle domande sono simili, motivo per cui in figura le colonne e le righe 2 e 5 risultano sempre conformi.



Figura 66: Configurazione usata nel sistema per generare il Reticolo della Conoscenza con i dati di frequenza (Cluster Based) - 1.

---

<sup>7</sup>per i dati di prova le domande accoppiate sono (1;4), (3;6) e (2;5)

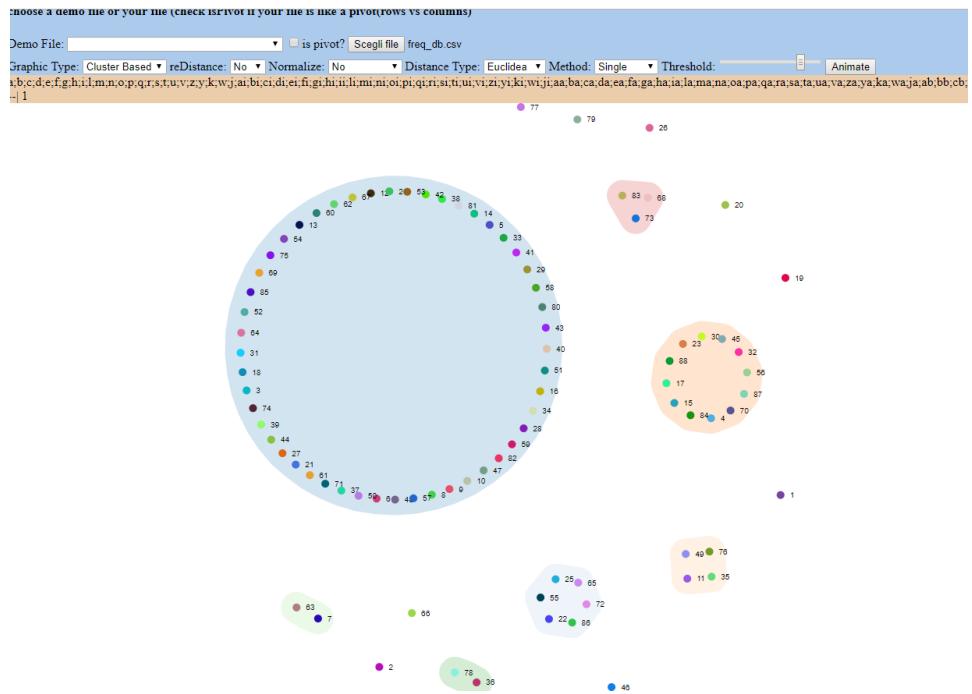


Figura 67: Configurazione usata nel sistema per generare il Reticolo della Conoscenza con i dati di frequenza (Cluster Based) - 2.

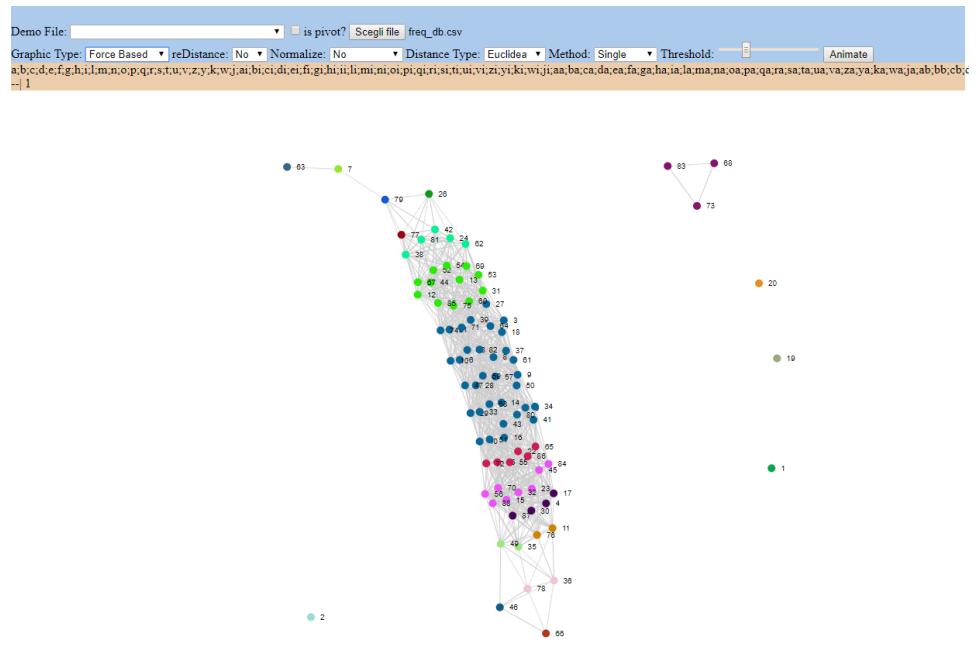


Figura 68: Configurazione usata nel sistema per generare il Reticolo della Conoscenza con i dati di frequenza (Forced Based).

## 5 Configurazione dell'applicativo progettato e strumenti di sviluppo

### 5.1 Test d'unità

Per eseguire, con successo, i test d'unità è necessario procedere all'installazione, se non già presenti, dei moduli node\_js e delle componenti aggiuntive necessarie ad una corretta esecuzione degli script. Di seguito riporto le istruzioni per poter effettuare con successo la build dei test implementati.

Prima di tutto l'utente deve aprire una *shell da terminale* e spostarsi all'interno delle sottocartelle rete\_neurale/rete neurale nella repository "AI-Reticolo\_della\_conoscenza". Per poter eseguire i test d'unità su qualsiasi sistema operativo, è necessarie procedere alle seguenti installazioni:

1. Installare node js da <https://nodejs.org/en/download/> per poter usare il comando npm;
2. Installare mocha: **npm install mocha**;
3. Installare il modulo babel core/register: **npm install --save-dev babel-core babel-preset-env**;
4. Installare presets per 2015: **npm install babel-cli babel-preset-es2015**;
5. Installare il modulo chai: **npm install chai**.

Per mandare in esecuzione i test digitare da terminale il comando **npm test**. Javascript è un linguaggio lato client, da browser. I test automatici sono stati sviluppati trasformando le singole unità in moduli node esportati. La sintassi exports poco si adatta all'utilizzo dei browser<sup>8</sup> per questo ho deciso di omettere le keywords exports come preambolo dei metodi soggetti a test; tale aggiunta è riservata all'utente nel momento precedente al test running.

---

<sup>8</sup>Il supporto viene offerto da Safari 10.1, Chrome 61, Firefox 54, Edge 16

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

#Test1 - generatore input
✓ generazione array mappato su grafo

#Test2 - generatore input pure
✓ generazione array mappato su grafo

#Test3 - generatore input probability
✓ generazione array spuro

#Test4
✓ array di colori uguali
✓ array di colori diversi

#Test5 - configurazione della rete andata a buon fine
✓ configurazione rete del database
✓ configurazione rete di prova

#Test6 - controllo valori inseriti nella form del differenziale di accoppiamento
✓ valore corretto entro i range stabiliti
✓ conversione a intero di un numero con virgola

#Test7 - stampa
✓ test del modulo di stampa

#Test8 - test matrice frequenza
✓ test frequenza con risposta positiva exists
✓ test frequenza con risposta positiva
✓ test frequenza con risposta negativa exists
✓ test frequenza con risposta negativa

32 passing (34ms)
eleonora@leonora-Lenovo-Z51-70:~/Scrivania/AI-Reticolo_della_conoscenza/rete_neurale/rete_neurale$ 

```

Figura 69: Resoconto unit test con successo.

## 5.2 Sistema operativo e browser

Il progetto è stato sviluppato su sistema operativo Linux e ripetutamente testato per tutta la durata dello stage sul sistema operativo Windows messo a disposizione in ufficio. L'applicativo della Rete neurale è sviluppato ad doc per browser Chrome in cui il caricamento di grandi moli di dati ha prestazioni migliori. Sconsiglio l'utilizzo della Rete neurale in uso per i dati del database su Microsoft Edge e Firefox , in quanto il caricamento del file di allenamento è molto lento.

### 5.2.1 Specifiche di sviluppo

- Sistemi operativi di sviluppo:
  - Versione sistema operativo di sviluppo principale: Ubuntu 4.15.0-52.56-generic 4.15.18;
  - Versione sistema operativo pc fisso aziendale: Windows 10 Enterprise, versione 1809;
- Browser di testing:
  - browser principale: Chrome versione 75.0.3770.100 a 64 bit e Versione 66.0.3359.170 a 64 bit;

- browser (sconsigliato per l'uso della Rete del database): Microsoft Edge 44.17763.1.0, Microsoft EdgeHTML 18.17763;
  - browser (sconsigliato per l'uso della Rete del database): Firefox Quantum 67.0.2 a 64 bit.
- Strumenti di codifica:
    - Visual Studio Code 1.35.1.
  - Strumenti di documentazione:
    - Texmaker 5.0.3 (compiled with Qt 5.9.5).