L'analisi effettuata punta a determinare il modello da utilizzare per l'identificazione parametrica di una singola fuel cell. L'obiettivo è determinare il numero di circuiti RC, ovvero il numero di cerchi, necessari per rappresentare l'impedenza analizzata. Per svolgere questo compito è stata utilizzata una rete neurale, ovvero uno strumento orientato allo svolgimento di compiti che possono risultare semplici per una persona, ma difficili da definire come algoritmo.

## Definizione problema

Come scritto prima, il problema da risolvere è la determinazione del numero di circuiti RC necessari per rappresentare l'impedenza.
Sono stati considerate impedenze con al più 3 circuiti RC.

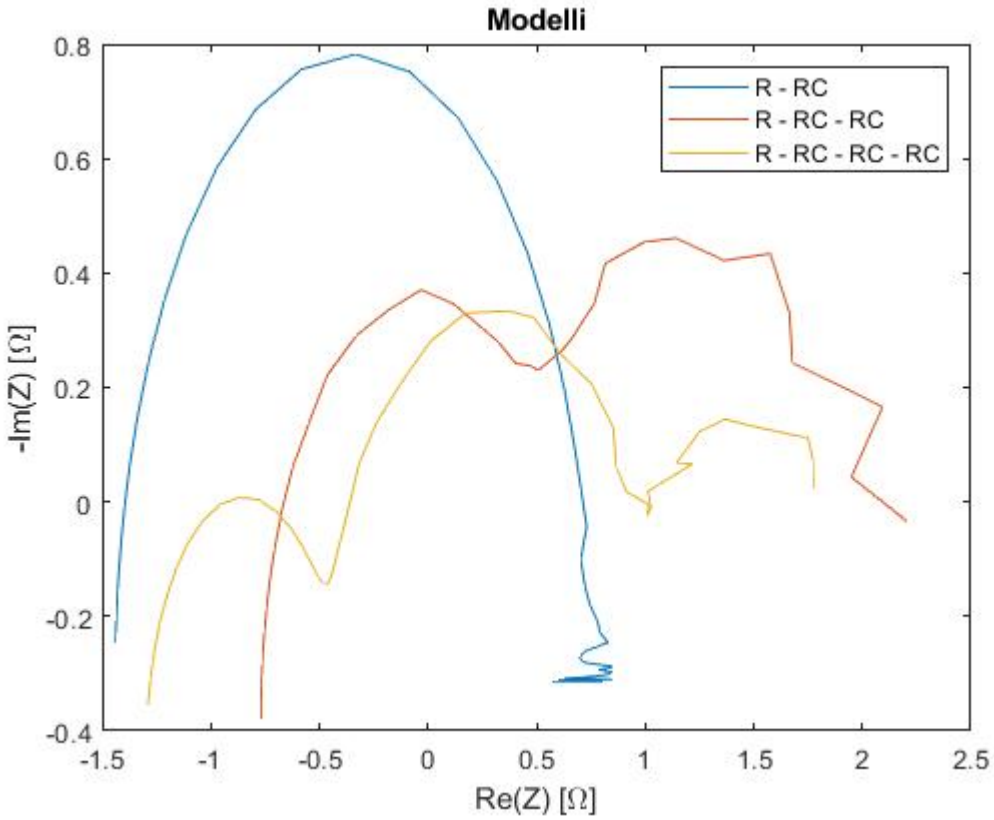


*Figura 1: Confronto modelli utilizzati*

Dalla figura 1, si possono osservare i tre modelli scelti. Le tre curve sono state generate casualmente ed in seguito è stato aggiunto rumore (maggiore a bassa frequenza).
Ciò che si vuole quindi ottenere è riconoscere il numero di "gobbe" che presenta la curva esaminata.
Si vuole quindi risolvere un problema di classificazione.

## Preparazione dati

L'impedenza considerata subisce diverse trasformazioni prima di essere inserita nella rete neurale.

La prima trasformazione è la traslazione della curva nell'origine dell'asse reale. Lo spostamento orizzontale infatti non è legato al numero di circuiti RC, ma dipende solamente dalla resistenza R presente in serie ai circuiti. Rappresenta quindi una caratteristica non rilevante per la rete neurale e per la classificazione dell'impedenza.

Dopo la traslazione, la parte reale e la parte immaginaria vengono normalizzate rispetto al valore massimo della parte reale e il valore massimo della parte immaginaria. Questo re-scaling dei dati aiuta la rete neurale a dare lo stesso peso a tutte le variabili di input.

Infine, i dati vengono sistemati in un vettore [ real(Z), imag(Z) ] per poter essere forniti in ingresso alla rete neurale.

La trasformazione dei dati prende il nome di pre-processing ed è fondamentale per migliorare le prestazioni della rete [1, p. 295].

## Rete neurale

Le reti neurali sono uno strumento progettato per il riconoscimento di pattern. Possono essere utilizzate per raggruppare dati in base a somiglianze nei parametri di input, oppure sono in grado di classificare dati se si ha a disposizione un set su cui addestrare la rete.

La rete è composta da un numero variabile di layer, dove ogni layer è composto da un certo numero di nodi. Ogni nodo riceve in ingresso l'ouput dei nodi del layer precedente, li amplifica (o riduce) per un certo fattore, li somma e li trasforma per fornirli ai nodi del livello successivo. La figura 2 mostra la struttura di un nodo

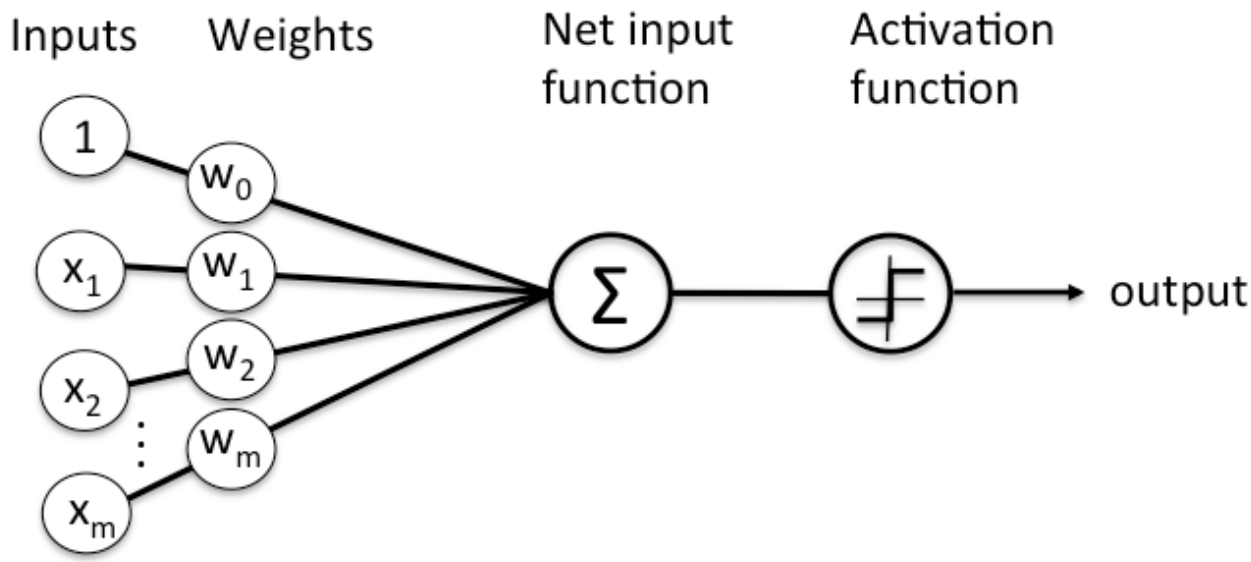


*Figura 2: Struttura di un nodo*

Ogni layer è composto da un numero variabile di nodi. Una rete presenta un layer di ingresso, un layer di uscita e un numero variabile di layer nascosti (in assenza di layer nascosto, il sistema applica una regressione lineare ai dati di ingresso).

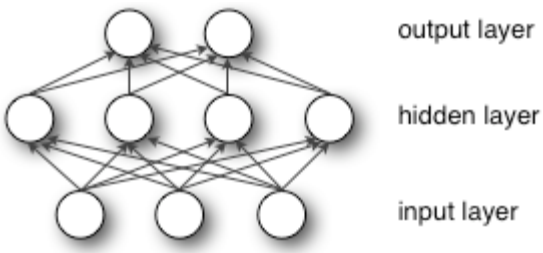


*Figura 3: Struttura rete*

Una rete con più di un livello nascosto viene definita come "deep neural network", altrimenti la si può definire come "shallow neural network".

L'output della rete rappresenterà la probabilità dei dati di input di appartenere alle possibili classi. [2]

Durante il processo di training, il valore dei nodi del livello di output è noto e i pesi di ogni nodo sono inizializzati casualmente. È possibile confrontare gli output calcolati dalla rete con gli output reali, in modo da determinare un errore per ogni nodo e aggiustare il valore dei pesi all'esecuzione successiva. Il processo consiste quindi nell'inserire l'insieme di dati (uno alla volta) nella rete e aggiustare i valori dei pesi; una volta esaurito il dataset, il processo può essere ripetuto per migliorare le prestazioni.

La scelta del numero di livelli e i nodi per livello non è deterministica, ma si basa sull'esperienza del designer della rete. Una regola spesso citata è che una rete con un layer nascosto può approssimare qualsiasi funzione [3]. Esistono, però, contro-prove di funzioni che non sono correttamente rappresentate tramite un singolo livello nascosto (da approfondire). Un numero di livello nascosti pari a due aiuterebbe a migliorare le prestazioni delle reti in cui un singolo livello risolve il problema proposto, ma in maniera inefficiente [4].

Per quanto riguarda la scelta dei nodi, esistono diverse indicazioni che consigliano di utilizzare un numero di nodi compreso tra il numero di input e il numero di output, o al più pari al doppio del numero di input. Queste non sono formule o regole dimostrate, ma solamente consigli; il modo migliore per determinare il numero di nodi è quindi provare più reti fino a determinare quale combinazione di nodi si comporta meglio.

## Data augmentation

Per addestrare correttamente una rete neurale è necessario un adeguato numero di dati classificati.
Non avendo molte impedenze a disposizione, si può utilizzare un processo chiamato data augmentation per ottenere nuove curve per l'addestramento della rete.
Il processo mira ad apportare piccole modifiche ad una curva, in modo da crearne altre simili dello stesso modello, secondo le seguenti regole:

1. Pesi: ad ogni punto viene assegnato un peso di 0.35 per i primi 12 punti, e con valori crescenti da 0.05 a 1 per i restanti 36, secondo la seguente istruzione
   ```
   weights = [0.35*rand(1,12) (0.5+0.5*rand(1,36)).*linspace(0.1,1,36)];
   ```
2. Rumore: ogni punto viene moltiplicato per il rispettivo peso e per un numero random compreso tra 1.035 e 0.965.
3. Scaling: tutti i punti vengono moltiplicati per un numero compreso tra 0.75 e 1.25.
4. Rotazione: la curva viene ruotata per un angolo casuale compreso -5° e 5°
5. La curva ha un 50% di probabilità di essere ribaltata

Nota: è consigliato effettuare anche una traslazione dei dati; in questo caso non è stata apportata in quanto, come illustrato prima, i dati vengono traslati nell'origine prima di essere inseriti nella rete neurale.

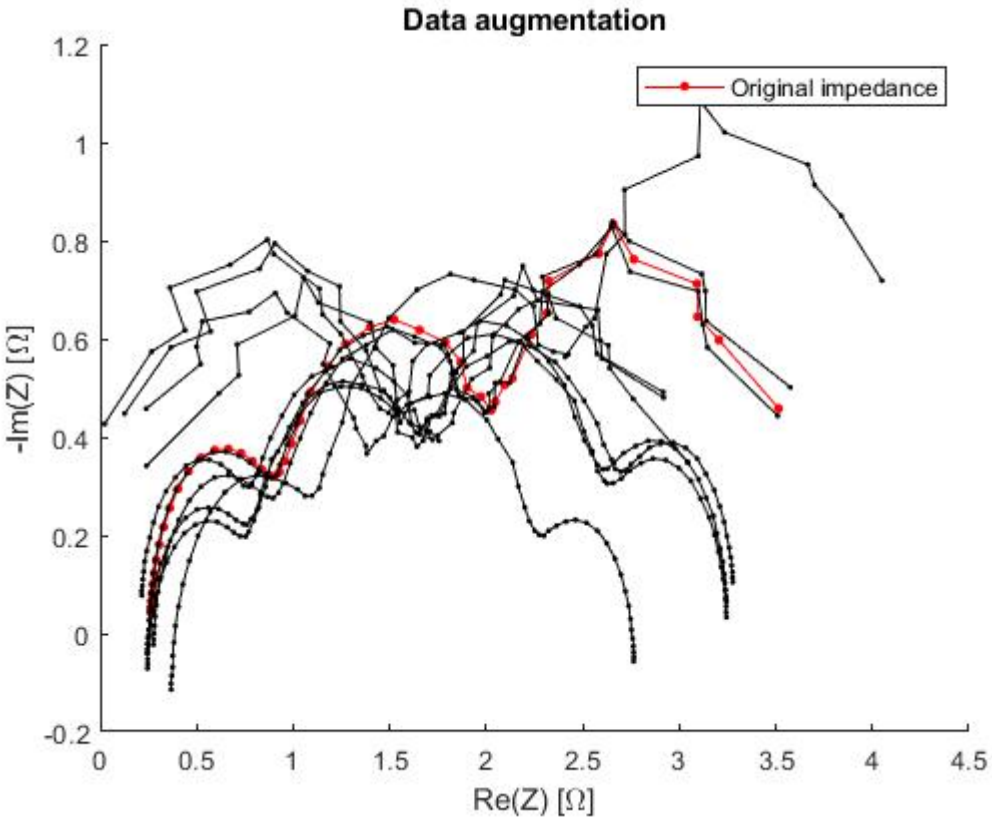


*Figura 4: Data augmentation per una impedenza con tre circuiti RC*

Nella figura 4 è riportato un esempio di data augmentation effettuato su una curva a tre cerchi. La curva rossa è l'impedenza originale, mentre le curve nere sono le impedenze ottenute tramite il processo di data augmentation.

## Risultati con curve simulate

Prima di utilizzare curve reali per l'addestramento delle rete neurale, sono stati effettuati dei test con curve generate tramite Matlab:

1. 1500 curve per il training della rete, 135000 curve per il test della rete.

Per ogni test sono state addestrate le reti con le seguenti caratteristiche:

1. Layer 1: 24 nodi.
2. Layer 1: 48 nodi.
3. Layer 1: 96 nodi.
4. Layer 1: 24 nodi. Layer 2: 12 nodi.
5. Layer 1: 48 nodi. Layer 2: 24 nodi.
6. Layer 1: 96 nodi. Layer 2: 48 nodi.
7. Layer 1: 96 nodi. Layer 2: 48 nodi. Layer 3: 24 nodi.

### Risultati test 1

Per ogni modello sono state generate 50000 curve, di cui 500 utilizzate per il training della rete.

Tutte le reti hanno classificato correttamente tutte le curve di tipo 1 (R-RC) e di tipo 2 (R-RC-RC). Di seguito sono riportati i dati riguardo la classificazione delle curve di tipo 3. Le curve di tipo 3 erroneamente classificate sono state sempre assegnate al modello di tipo 2.

Essendo i risultati della rete dipendenti dall'inizializzazione (casuale) dei pesi dei ogni nodo, sono stati effettuati 200 test per ognuna delle configurazioni riportate sopra. Per ogni configurazione sono riportati il risultato migliore, il risultato peggiore e la media di curve errate. Come specificato sopra, le curve errate fanno riferimento a impedenze generate tramite il modello a tre circuiti RC e assegnate al modello a due circuiti RC.

| Configurazione | Caso migliore | Caso peggiore | Media curve errate |
|---|---|---|---|
| 1: 24 | 0 | 147 | 30.67 |
| 1: 48 | 0 | 150 | 19.59 |
| 1: 96 | 0 | 196 | 14.72 |
| 1: 24 – 2: 12 | 0 | 147 | 25.97 |
| 1: 48 – 2: 24 | 0 | 64 | 9.20 |
| 1: 96 – 2: 48 | 0 | 49 | 4.25 |
| 1: 96 – 2: 48 – 3: 24 | 0 | 107 | 5.84 |

## Risultati test 2

Per il secondo test, si sono utilizzate solamente le prime 100 curve per il training e le ultime 45000 per il testing, in modo da simulare una situazione con meno dati a disposizione. Come per il test precedente, si riportano i valori ottenuti rispetto a 200 test:

| Configurazione | Caso migliore | Caso peggiore | Media curve errate |
|---|---|---|---|
| 1: 24 | 264 | 1075 | 523.96 |
| 1: 48 | 85 | Tipo 1: 11<br>Tipo 3: 978 | 432.35 |
| 1: 96 | 103 | 1066 | 341.22 |
| 1: 24 – 2: 12 | 94 | Tipo 2: 1<br>Tipo 3: 1044 | 375.62 |
| 1: 48 – 2: 24 | 41 | 905 | 262.35 |
| 1: 96 – 2: 48 | 11 | 528 | 146.38 |
| 1: 96 – 2: 48 – 3: 24 | 1 | 859 | 124.38 |

Dove non esplicitamente indicato, le curve errate si riferiscono alle curve generate tramite il modello a tre circuiti RC.

## Risultati test 3

Come descritto precedentemente, tramite la tecnica del data augmentation è possibile ottenere più dati di training a partire da un insieme ridotto. In questo terzo test, a partire da 100 curve iniziali (per modello), sono state ottenute 500 curve (per modello) con cui addestrare la rete.

I risultati ottenuti sono i seguenti:

| Configurazione | Caso migliore | Caso peggiore | Media curve errate |
|---|---|---|---|
| 1: 24 | 14 | 506 | 140.14 |
| 1: 48 | 14 | 369 | 120.68 |
| 1: 96 | 16 | 321 | 78.01 |
| 1: 24 – 2: 12 | 4 | 594 | 108.46 |
| 1: 48 – 2: 24 | 5 | 240 | 77.77 |
| 1: 96 – 2: 48 | 7 | Tipo 2: 37<br>Tipo 3: 399 | 56.99 |
| 1: 96 – 2: 48 – 3: 24 | 1 | 156 | 40.94 |

Dove non esplicitamente indicato, le curve errate si riferiscono alle curve generate tramite il modello a tre circuiti RC.

### Conclusione

Da questi primi tre test, si nota come le ultime due configurazioni si comportano sempre meglio rispetto alle altre, se si considera la media di curve errate per esecuzione, indicando quindi un miglior risultato nel caso di una rete a più livelli e con un numero elevato di nodi.

Nel caso di questi test, i risultati sono stati ottenuti utilizzando solamente la parte immaginaria come input, in modo da alleggerire la rete. Nel caso di curve reali (e non generate) bisogna analizzare se questa semplificazione è possibile oppure porta a una perdita di informazione per la determinazione del modello.

Il prossimo passo è esaminare le impedenze a disposizione ed effettuare il training con queste curve e con la tecnica del data augmentation.

## Bibliografia


[1] C. Bishop, Neural Networks for Pattern Recognition, 1995.

[2] «https://skymind.ai/wiki/neural-network,» [Online].

[3] I. Goodfellow, Deep learning, 2016.

[4] R. Reed, Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks, 1999.