



Consiglio Nazionale delle Ricerche  
Istituto di Calcolo e Reti ad Alte Prestazioni

## Soft Sensors: Metodologie di progettazione e ottimizzazione

**Rapporto Tecnico N.: 6**  
**RT-ICAR-PA-15-06**

**Dicembre 2015**



Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)  
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: [www.icar.cnr.it](http://www.icar.cnr.it)  
– Sede di Napoli, Via P. Castellino 111, 80131 Napoli, URL: [www.na.icar.cnr.it](http://www.na.icar.cnr.it)  
– Sede di Palermo, Viale delle Scienze, 90128 Palermo, URL: [www.pa.icar.cnr.it](http://www.pa.icar.cnr.it)



Consiglio Nazionale delle Ricerche  
Istituto di Calcolo e Reti ad Alte Prestazioni

## **Soft Sensors: Metodologie di progettazione e ottimizzazione**

Umberto Maniscalco

**Rapporto Tecnico N.: 6  
RT-ICAR-PA-15-06**

**Data:  
Dicembre 2015**

---

*I rapporti tecnici dell'ICAR-CNR sono pubblicati dall'Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche. Tali rapporti, approntati sotto l'esclusiva responsabilità scientifica degli autori, descrivono attività di ricerca del personale e dei collaboratori dell'ICAR, in alcuni casi in un formato preliminare prima della pubblicazione definitiva in altra sede.*

## 1 Introduzione.

I sensori software, meglio noti in letteratura come soft sensor o virtual sensor o inferential model sono enti matematici generalmente implementati come software in grado di processare misure allo scopo di ottenere altre misure.

I Soft Sensors sono usati al posto dei sensori reali quando il misurando “non può essere misurato” da sensori reali o banalmente quando è più “conveniente” usare sensori software al posto di quelli reali.

L'impossibilità di misurare il misurando può ovviamente discendere da diverse motivazioni. Ad esempio, il misurando potrebbe essere una grandezza non fisica e quindi potrebbe non esistere una sensore reale in grado di misurarlo. Si pensi ad esempio alla misurazione (o meglio alla stima) degli indici che rappresentano la fiducia in un leader politico o le intenzioni di voto di una popolazione. In altri casi il misurando, sebbene sia una grandezza fisica, potrebbe non essere misurabile direttamente perché inaccessibile, come la distanza di una stella dalla terra o perché non esistono strumenti reali in grado di misurarne la grandezza come nel caso delle temperature di certe reazioni nucleari.

L'aspetto legato alla “convenienza” dell'uso di soft sensors al posto di sensori reali non può essere affrontato se non si definisce una funzione di costo rispetto alla quale misurare la convenienza. In molti casi l'impiego dei sensori reali può risultare molto costoso sia in termini di danaro sia in termini di mancato guadagno. Nel primo caso è molto semplice comprendere la convenienza e allo stesso modo dovrebbe risultare semplice stabilire se sia più conveniente l'uso di sensori software piuttosto che l'uso di sensori reali.

Nel secondo caso le cose potrebbero essere meno evidenti. Si pensi ad esempio alle lunghe campagne di monitoraggio di alcuni beni culturali architettonici.

In questo caso, oltre i costi per l'acquisto o noleggio e l'installazione dell'insieme dei sensori, devono essere considerati altri costi. Il costo della non fruibilità (o fruibilità ridotta) del bene architettonico sia in termini economici, nel caso in cui il bene era visitabile a pagamento sia in termini culturali. Infatti, spesso il monitoraggio condotto con sensori reali è fortemente invasivo nei confronti del bene e ne deturpa l'aspetto in modo sostanziale.

Un altro aspetto da tenere in grande considerazione è che i soft sensors, una volta “addestrati” (sarà più chiaro più avanti) possono continuare a misurare per lunghissimo tempo senza alcun costo.

In molti casi i soft sensors non sostituiscono i sensori reali ma piuttosto lavorano in parallelo ad essi. Questo è il caso in cui essi costituiscono un back up per i sensori reali. Si pensi ad esempio al caso di un wireless sensor network distribuita su una vasta zona geografica, magari in condizioni territoriali e climatiche critiche. In queste circostanze ci possono essere molti motivi per cui uno o più sensori possono fallire una misura. Infatti, la misura può fallire per cause direttamente imputabili al sensore, ad esempio per un mal funzionamento, staratura, mancanza di alimentazione etc. Oppure, la misura può non essere disponibile per un problema legato all'infrastruttura di comunicazione. In ogni caso ci si ritroverebbe in una condizione di dato mancante o affetto da errore. In queste circostanze appare chiaro come una rete di sensori software, che sia lo specchio della rete reale, può costituire un valido back up per continuare le misure, senza perdita di informazione.

Da un punto di vista più formale i soft sensors possono essere definiti come un modello matematico che permette di ricostruire/stimare variabili di interesse (non misurabili) sulla base di altre variabili misurabili.

Nel suo aspetto più semplice, questo modello matematico può essere rappresentato come nella seguente figura, in cui  $Y_m$  rappresenta l'insieme delle variabili misurabili,  $\bar{Y}_n$  rappresenta

l'insieme delle variabili da stimare,  $U$  indica un comando, generalmente coincidente con un timing e infine  $MSS_n$  è il modello matematico da individuare.



**Figura 1: Schema generale di un soft sensor a catena aperta**

Nella progettazione dei soft sensors il problema principale è quello di trovare il modello matematico “giusto” o “adatto” a stimare con un “ragionevole” grado di errore le variabili di interesse.

Gli aggettivi giusto, adatto o ragionevole indicano la circostanza che non esiste ovviamente un modello matematico unico per ogni problema e che, anche se definiamo il problema, restano sempre aperte diverse scelte progettuali che fanno sì che non resti univocamente determinato il modello matematico migliore.

Nella progettazione dei soft sensors sono spesso due le linee guida che vanno seguite. La prima riguarda i “dati”, l'altra invece “l'applicazione”. La frase precedente va interpretata nel senso che sono spesso i dati a disposizione che determinano le scelte progettuali di base. Ovvero la loro moltitudine, il periodo temporale coperto, la loro correttezza etc.

L'applicazione, invece stabilisce il grado di precisione che devono avere i soft sensors. In altre parole, la possibilità di sostituire un sensore reale (ove possibile), con un sensore software, dipende dall'errore ammissibile per la specifica applicazione. Ad esempio prendendo in considerazione un parametro fisico come la temperatura, se l'applicazione può ammettere un errore intorno ad un grado allora probabilmente la possibilità di riuscire a progettare e realizzare un soft sensor con queste caratteristiche è concreta. Altrimenti se l'applicazione necessita di precisioni al decimo di grado allora probabilmente non si riuscirà a progettare e a realizzare soft sensors con tale precisione.

Quindi ricapitolando il modello matematico con cui realizzare il particolare soft sensor sarà individuato in funzione dei dati a disposizione e in base all'applicazione specifica in cui esso sarà utilizzato.

### **1.1 Strumenti statistici.**

Gli strumenti a disposizione per la progettazione dei soft sensors, e quindi per l'implementazione del modello matematico che li rappresenta, sono numerosi. Infatti, come detto precedentemente i soft sensors possono essere delle semplici formule fisiche, per cui dalla misura di alcune grandezze si possono ricavare, in forma chiusa, delle altre grandezze. In questo caso, il più semplice, il modello matematico è costituito da un software procedurale che opera la trasduzione delle grandezze. L'errore in questo caso è computabile anch'esso in forma chiusa ed è funzione diretta degli errori dei sensori reali da cui si ricavano le misure.

Un caso un po' più complesso è quello in cui il soft sensor è uno stimatore. In questo caso il soft sensor opera una "previsione" sulla misura, la previsione può essere sia di tipo temporale, ovvero si stima cosa accadrà ad una certa grandezza in un futuro prossimo, sia di tipo spaziale, ovvero si stima una grandezza in un certo luogo partendo da misure prese in altri luoghi. Ovviamente esistono casi ancora più complessi in cui la stima della grandezza da misurare è in realtà una previsione spazio temporale. Tutti questi casi però sono accomunati dal medesimo principio di base, ovvero l'ipotesi che esista un modello associativo tra il pattern delle grandezze di ingresso, quelle che precedentemente sono state indicate con  $Y_m$  e il pattern delle grandezze di uscita indicate con  $\bar{Y}_n$ . Quindi il modello  $MSS_n$  è in realtà un modello associativo tra i due pattern di grandezze: quello d'ingresso e quello di uscita.

In questa logica, uno degli strumenti più semplici che si hanno a disposizione per la realizzazione di questo modello associativo sono le regressioni sia lineari che multivariate.

Le regressioni consentono di risolvere il problema di una relazione funzionale tra variabili misurate sulla base di dati campionari ottenuti da popolazione infinita.

$$\bar{Y}_i = \beta_0 + \beta_1 * X_i + u_i$$

dove  $\bar{Y}_i$  è la variabile da stimare,  $\beta_0$  è l'intercetta,  $\beta_1$  è il coefficiente angolare ed  $X_i$  è la variabile indipendente e  $u_i$  rappresenta l'errore statistico. L'indice  $i$  varia tra 1 ed  $n$  e rappresenta le osservazioni della serie campionaria.

Volendo essere più formali, la regressione lineare è un metodo per stimare del valore atteso condizionato di una variabile dipendente, dati i valori di altre variabili indipendenti. Quindi esattamente quello che ci interessa realizzare. Nel caso in cui la nostra variabile dipendente, dipendesse da più variabili di "ingresso" allora si ricadrebbe nel caso di una regressione semplice multivariata rappresentata dalla seguente equazione:

$$\bar{Y}_i = \beta_0 + \beta_1 * X_{1i} + \beta_2 * X_{2i} + \dots + \beta_n * X_{ni} + u_i$$

I regressori sia lineari che multivariati, sono tra gli strumenti statistici più semplici con cui realizzare il modello  $MSS_n$  che rappresenta il nostro soft sensor.

Altri strumenti della statistica possono essere impiegati con efficacia per la costruzione di soft sensors elementari. Essendo però, questi strumenti tra i più semplici spesso non consentono di ottenere risultati accettabili per l'applicazione che si intende realizzare.

## 1.2 Strumenti neurali.

In molti casi occorre utilizzare strumenti decisamente più complessi. Ovvero strumenti che si basano sul paradigma del machine learning. Strumenti che costruiscono il modello associativo tra il pattern delle variabili di ingresso e il pattern delle variabili di uscita sulla base di esempi.

Tra questi strumenti, probabilmente, i più versatili e quindi anche quelli maggiormente impiegati nelle diverse applicazioni che prevedono l'utilizzo di soft sensors sono le reti neurali.

Riportando una definizione semplificata che è possibile ritrovare su wikipedia, sulle reti neurali si può dire che: "Le reti neurali artificiali sono modelli matematici che rappresentano l'interconnessione tra elementi definiti neuroni artificiali, ossia costrutti matematici che in qualche misura imitano le proprietà dei neuroni viventi. Questi modelli matematici possono essere utilizzati sia per ottenere una comprensione delle reti neurali biologiche, ma ancor di più per risolvere problemi ingegneristici di intelligenza artificiale come quelli che si pongono in diversi ambiti tecnologici (in elettronica, informatica, simulazione, e altre discipline).

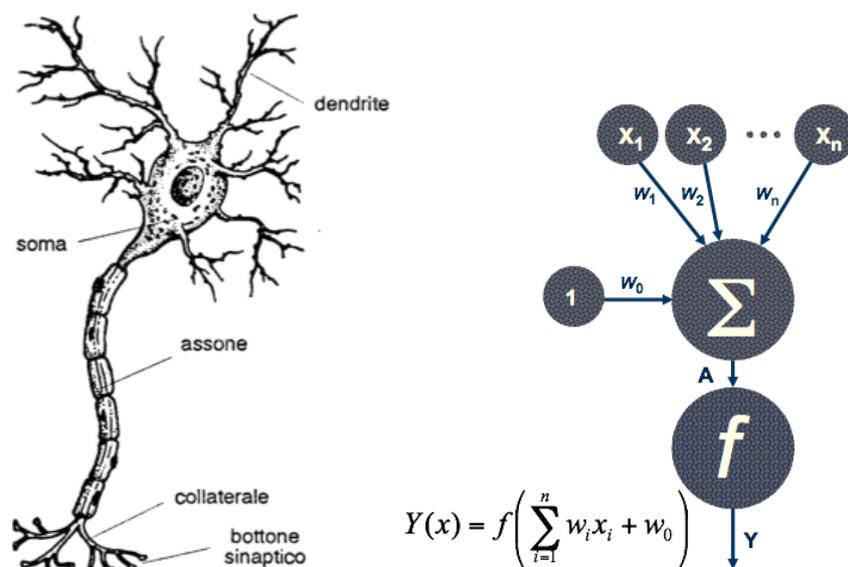
Una rete neurale artificiale può essere realizzata sia da programmi software che da hardware dedicato (DSP, Digital Signal Processing)”.  
 Nella successiva figura sono rappresentati a confronto le strutture del neurone biologico e del neurone artificiale.

Nel neurone biologico possono distinguersi 3 parti fondamentali:

- il soma: corpo cellulare
- l'assone: linea di uscita del neurone unica ma che si dirama in migliaia di rami
- il dendrite: linea di entrata del neurone che riceve segnali in ingresso da altri assoni tramite le sinapsi

Il corpo cellulare esegue una “somma pesata” (integrazione) dei segnali in ingresso. Se il risultato supera un certo valore di soglia allora il neurone si attiva ed è prodotto un “potenziale di azione” che è trasportato all'assone. Se il risultato non supera il valore di soglia, il neurone rimane in uno stato di riposo.

Il neurone artificiale in pratica replica, in modo più o meno fedele, il comportamento del neurone biologico. Ovvero, la somma pesata dei segnali d'ingresso costituisce la variabile indipendente di una funzione, detta appunto funzione di attivazione. In definitiva l'uscita del neurone è una funzione della somma pesata dei segnali d'ingresso.



**Figura 2: Confronto tra la struttura di un neurone biologico a sinistra e un neurone artificiale a destra.**

Tra le più utilizzate funzioni di attivazione nelle reti neurali di tipo feedforward si possono distinguere: il gradino o step ovvero il neurone ha un'uscita a livello alto (attivo) se la somma pesata degli ingressi supera un certo valore, altrimenti resta a livello basso (inibito). La retta o pureline, in questo caso l'uscita è direttamente proporzionale (o inversamente proporzionale dipende dal coefficiente angolare della retta). La sigmoide, che costituisce un compromesso tra le due funzioni precedenti, infatti, nel tratto contrale ha un comportamento proporzionale simile alla retta mentre essendo asintotica per alti (in valore assoluto) valori dell'ingresso ha un comportamento simile al gradino.

Stabilito il funzionamento generale di un singolo neurone, così come nel caso delle reti biologiche, anche nel caso delle reti artificiali è di estrema importanza la topologia della rete. Ovvero, il modo in cui le singole unità computazionali sono interconnesse tra loro.

Tra le prime topologie sperimentate è sicuramente da tenere in considerazione il perceptrone multistrato (Multi Layer Perceptron), questo tipo di rete, come mostra la figura, interconnette tutti i neuroni di uno strato con ogni singolo neurone dello strato successivo. L'uscita pertanto è ottenuta come la propagazione degli ingressi attraverso gli strati funzionali che compongono la rete. Il Multi Layer Perceptron è già uno strumento decisamente più adatto e versatile per la realizzazione dei soft sensor rispetto ai "semplici" strumenti statistici descritti precedentemente. Infatti è in grado di "imparare" il modello di associazione tra gli ingressi e le uscite grazie agli algoritmi di addestramento che determinano la configurazione dei pesi.

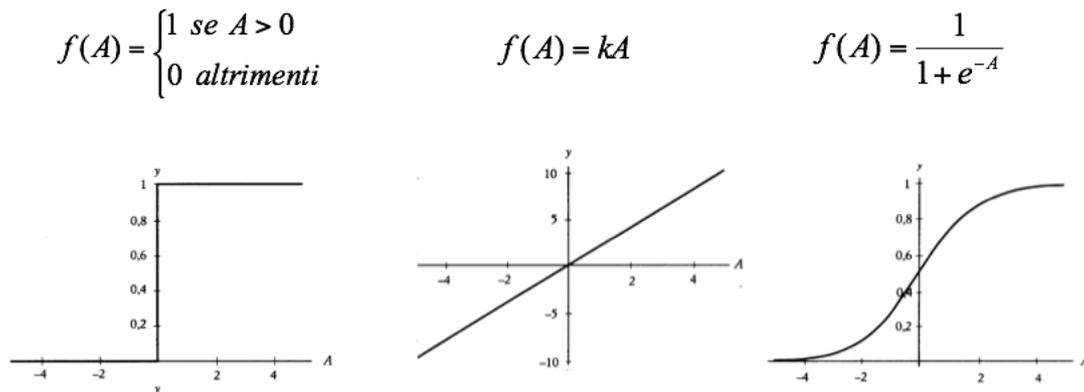


Figura 3: Le più comuni funzioni di attivazione utilizzate nelle reti feedforward.

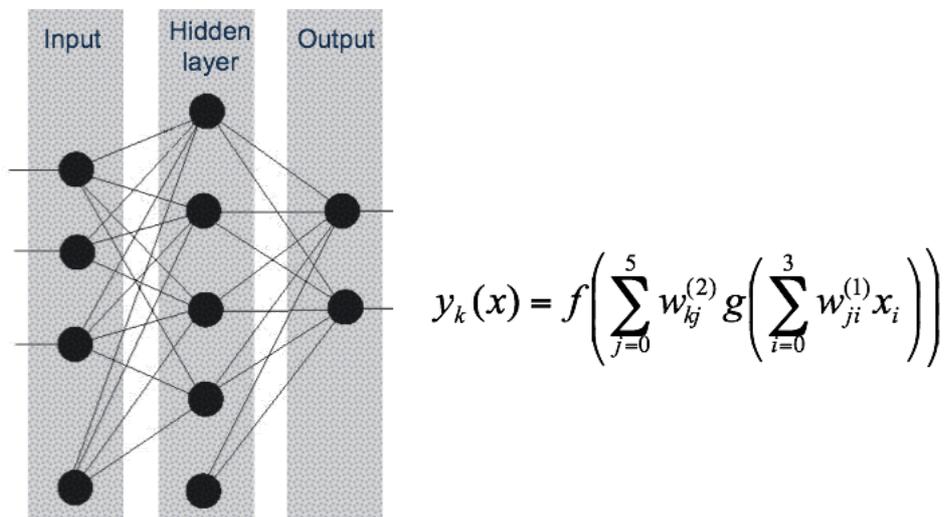


Figura 4: La topologia del Multi Layer Perceptron una delle topologie più semplici tra reti feedforward.

Uno strumento ancora un po' più sofisticato del Multy Layer Perceptron è la Radial Basis Funcion (RBF). Le reti neurali RBF sono fondamentalmente dei Multy Layer Perceptron ad un singolo strato nascosto e i neuroni di questo strato hanno funzioni di attivazione a simmetria radiale. Quindi le funzioni (di qualsiasi tipo) che si vogliono approssimare con una rete neurale di questo tipo possono essere ottenute come combinazioni lineari delle formule di attivazione dei singoli neuroni. Il nome basis deriva del fatto che la formula del perceptrone è una sommatoria di funzioni, ovvero una loro combinazione lineare. In algebra lineare una base è

proprio l'insieme di vettori indipendenti che generano uno spazio vettoriale attraverso una combinazione lineare.

Particolarmente interessanti, rispetto alla possibilità di essere impiegati come soft sensors, sono i modelli ricorsivi. Ovvero quelle topologie di rete neurale in cui l'uscita non dipende solo dalla propagazione degli ingressi attraverso i successivi layer della rete, ma dipende anche dallo "stato della rete".

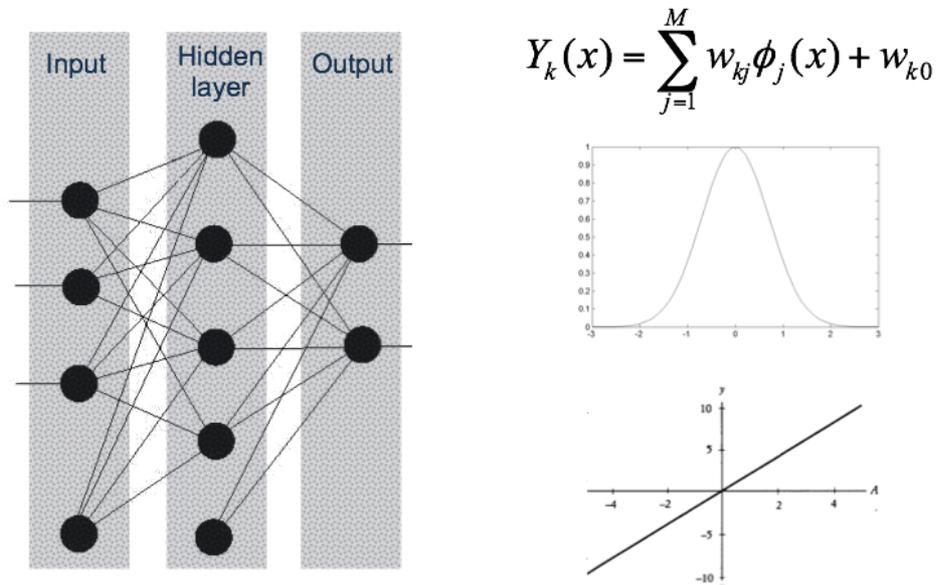


Figura 5: La topologia delle radial basis function una delle topologie più utilizzate come approssimatore funzionale.

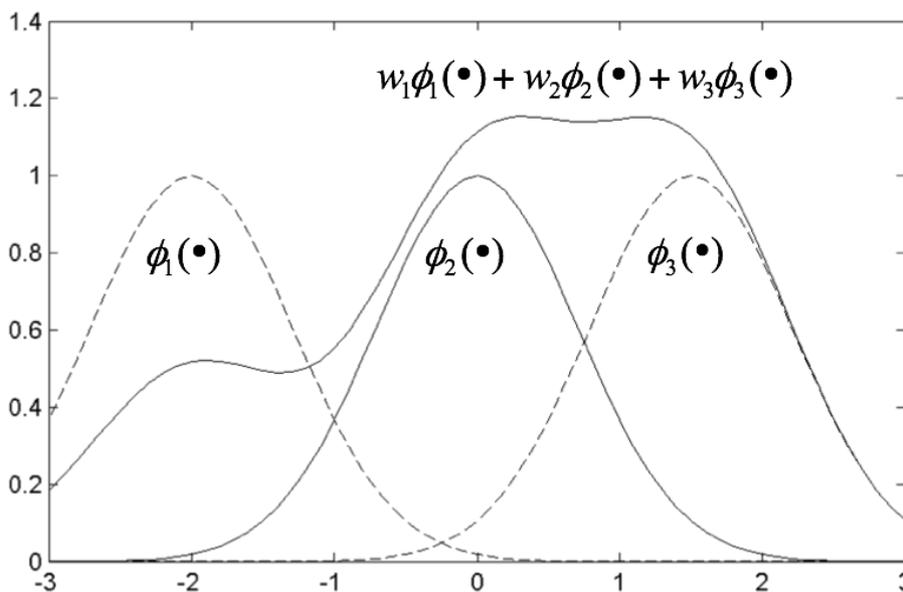
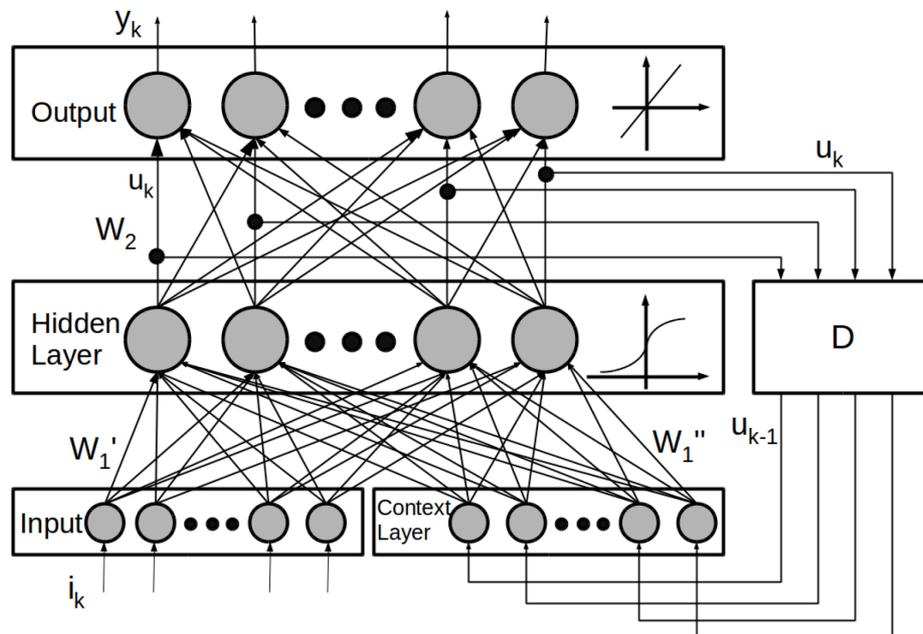


Figura 6: Le più comuni funzioni di attivazione utilizzate nelle reti feedforward.

Oppure, esprimendo il concetto in altri termini, l'uscita della rete dipende sia dagli ingressi al tempo al  $t_1$  sia dall'uscita della rete al tempo  $t_0$ , la quale viene reimpressa nella rete attraverso un strato di neuroni chiamato "context layer" come mostrato in figura. La caratteristica della ricorsione in questo tipo di reti le rende particolarmente utili ed adatte per l'approssimazione di fenomeni fisici con inerzia temporale. Le uscite della rete sono condizionate dall'intera serie

storica dell'evento che si intende approssimare. In questo modo, le reti ricorrenti tengono conto del loro stato precedente, cosa che le consente di svolgere compiti di previsione di sequenze temporali, spesso difficilmente alla portata dei perceptron multistrato convenzionali.



$$u_k = \text{tansig}(W_1' * i_k + W_1'' * u_{k-1} + b_1)$$

$$y_k = \text{purelin}(W_2 * u_k + b_2)$$

Figura 7: Topologia di una rete ricorsiva di tipo Elman.

Oltre gli aspetti topologici che definiscono la tipologia della rete va tenuto in grande considerazione l'aspetto dell'apprendimento.

Anzi si può sicuramente affermare che la parte dell'apprendimento è sicuramente quella più importante nella realizzazione di un soft sensor. Infatti, è durante l'addestramento della rete neurale (o delle reti neurali che lo implementano) che si "instilla" la conoscenza all'interno del modello neurale.

Nella maggior parte dei casi, o almeno nei casi delle reti neurali utilizzate per l'implementazione dei soft sensors, l'addestramento si basa sul paradigma "supervisionato". Come si può leggere su wikipedia "un apprendimento è supervisionato (supervised learning), qualora si disponga di un insieme di dati per l'addestramento (o training set) comprendente esempi tipici d'ingressi con le relative uscite loro corrispondenti: in tal modo la rete può imparare ad inferire la relazione che li lega. Successivamente, la rete è addestrata mediante un opportuno algoritmo il quale usa tali dati allo scopo di modificare i pesi ed altri parametri della rete stessa in modo tale da minimizzare l'errore di previsione relativo all'insieme d'addestramento. Se l'addestramento ha successo, la rete impara a riconoscere la relazione incognita che lega le variabili d'ingresso a quelle d'uscita, ed è quindi in grado di fare previsioni anche laddove l'uscita non è nota a priori; in altri termini, l'obiettivo finale dell'apprendimento supervisionato è la previsione del valore dell'uscita per ogni valore valido dell'ingresso, basandosi soltanto su un numero limitato di esempi di corrispondenza (vale a dire, coppie di valori input-output). Per fare ciò, la rete deve

essere infine dotata di un'adeguata capacità di generalizzazione, con riferimento a casi ad essa ignoti. Ciò consente di risolvere problemi di regressione o classificazione”.

Da quanto appena detto appare chiaro come il modello neurale sia spesso il miglior candidato per implementare il modello  $MSS_n$  di cui si è discusso precedentemente.

## 2 Progettazione.

I soft sensors sono sempre una parte di un sistema ingegneristico poiché non hanno ragion di esistere avulsi da esso. Pur essendo perfettamente coscienti che a seconda dell'applicazione e del sistema di cui essi fanno parte, la loro progettazione può differire, tuttavia si può pensare ad un modello sufficientemente generale che possa servire da riferimento per la loro progettazione. In altri termini, possiamo adottare un livello di astrazione progettuale per garantire la generalità necessaria senza per questo perdere di utilità. La successiva trattazione non ha la pretesa di rispondere alle esigenze progettuali di qualsiasi applicazione che preveda l'uso dei soft sensors. Ma di certo riesce a rappresentare una grande categoria di applicazioni basati sull'uso dei sensori software e in particolare tutte quelle applicazioni in cui i soft sensors lavorano in sostituzione o come back-up di sensori reali.

In questa ultima ipotesi alcune macro fasi progettuali, strettamente inerenti la progettazione di applicazioni basate sui sensori software neurali possono essere nettamente distinte.

### 2.1 Acquisizione e validazione dei dati.

Questa fase oltre al collazionamento dei dati che serviranno per l'addestramento (e successivamente per il testing) è di fondamentale importanza per la comprensione della natura del problema.

Sono infatti la tipologia dei dati a dettare i primi vincoli progettuali. O per meglio dire l'analisi delle funzioni (discrete) da approssimare. Spesso, sulla base dei dati si sceglie lo strumento con cui realizzare il modello  $MSS_n$ , e nel caso questa scelta coinvolga un modello neurale l'analisi dei dati serve ad individuare la tipologia di rete più adatta.

Pertanto, una buona analisi dei dati ricca di statistiche è di fondamentale importanza per compiere le prime scelte progettuali.

La fase di acquisizione dei dati non si dovrebbe tuttavia limitare solo a questo. Infatti, nella pratica, quando si acquisisce una grande mole di dati sensoriali per lunghi periodi è indispensabile operare una procedura di verifica dei dati.

E' comune infatti aspettarsi una certa percentuale di dati non utilizzabili perché affetti da errori. Errori che possono derivare sia dal fallimento del sensore nella sua misura sia da un errore di comunicazione del dato. Questi ultimi si verificano quando sensori reali fanno parte di una "rete" di sensori interconnessa sia in modo wireless sia cablata. In ogni caso è sempre necessario operare una verifica (spesso utilizzando gli strumenti della statistica) per comprendere quali dati sono da considerare validi e quali invece sono da considerare inutilizzabili.

Questo passo nasconde delle insidie intrinseche che potrebbero compromettere in modo sostanziale le prestazioni dei sensori software. Infatti, può risultare estremamente complesso distinguere un dato non valido da un outlier valido. Ovvero un dato che presenta un valore non accettabile da un valore accettabile sebbene improbabile. Un errore di questo tipo potrebbe avere come effetto quello di comprimere il range di valori che il soft sensor è in grado di stimare e quindi di non replicare con esattezza il comportamento del sensore reale. Questo è

solo un esempio delle difficoltà che la fase di validazione dei dati comporta e se ne possono presentare molte altre. Sempre a titolo di esempio se in una serie temporale si presentano numerose occorrenze dello stesso valore (magari con approssimazione alla massima precisione del sensore) questo potrebbe essere indice del fatto che siamo in presenza di un dato valido da un punto di vista numerico ma non valido da un punto di vista logico. Ovvero un valore che per un qualche motivo viene generato nell'ambito dei valori validi ma non consistente con la realtà, un sorta di valore di default per una certo tipo di errore.

## 2.2 Composizione degli insiemi di addestramento.

Anche questa fase è di vitale importanza per raggiungere le performances desiderate per i soft sensors che si stanno progettando. Infatti, considerando che alla base di questa tipologia di soft sensors vi è il paradigma del machine learning è evidente che stabilire quali e quanti sono gli esempi che dovranno formare il set di addestramento è basilare. Lacune, omissioni, ridondanze, errori, avranno ripercussioni sul funzionamento dei soft sensors.

Ovviamente non esistono formule né regole fisse da seguire ma soltanto l'esperienza e il rispetto di alcuni principi di base potranno garantire la formazione di set di addestramento e di test adeguati.

Si consideri l'ipotesi di voler virtualizzare un sensore reale di temperatura. Si supponga che questo sensore ha misurato ogni ora per la durata di circa un anno la temperatura. Allo stesso modo altri sensori nelle vicinanze hanno misurato ogni ora delle temperature per circa un anno anche loro. Riassumendo si hanno quindi a disposizione delle serie storiche di temperatura che coprono circa un anno. Di questi dati è necessario fare una partizione per ottenere almeno due set: il set di addestramento e il set di test. In molti casi si rende necessario avere a disposizione un ulteriore insieme di dati per la fase di validazione.

E' abbastanza facile intuire che il nostro sensore acquisirà una certa conoscenza del problema in funzione degli esempi che ad esso si forniranno in fase di addestramento. In questo specifico caso, se si decidesse di partizionare l'intera serie storica in due parti, prendendo la prima (temporalmente parlando) per costituire il training set, e la seconda parte per costituire il test set, ci si troverebbe nella condizione di addestrare il soft sensor con dati relativi a due stagionalità (autunno, inverno per esempio) e di testarne le proprietà su le altre due stagionalità (primavera, estate).

Questo ovviamente è un errore (concettuale) che ha gravissime ripercussioni sul funzionamento del soft sensor ed è facile da riconoscere. In molti altri casi errori altrettanto gravi, come conseguenze sulle prestazioni finali del soft sensor, possono essere molto meno evidenti.

## 2.3 Individuazione della tipologia, della topologia e strategie di addestramento.

Per ciò che concerne l'individuazione della tipologia di strumento da utilizzare per l'implementazione del modello  $MSS_n$ , si può affermare che solo l'analisi approfondita dell'applicazione da realizzare, e come già accennato, lo studio dei dati a disposizione possono dettare le linee guida. Sono tuttavia rari i casi in cui si può prescindere dal paradigma del machine learning a vantaggio degli strumenti della statistica.

In molti casi ci si può avvantaggiare dall'uso di strumenti ibridi, ovvero modelli che mettono insieme enti matematici differenti e li integrano tra loro per un miglioramento delle prestazioni dei soft sensors.

Il modello  $MSS_n$ , di cui si è discusso precedentemente, evidenzia che in generale si è nelle condizioni in cui un certo insieme di variabili  $\bar{Y}_n$  devono essere stimate a partire da un altro

insieme di variabili  $Y_m$ . Come mostra la figura seguente, le scelte progettuali da compiere sono molte e iniziano ancor prima di identificare lo strumento (o gli strumenti) matematico da utilizzare. La figura rappresenta, a sinistra, una topologia in cui tutte le variabili di ingresso contribuiscono contemporaneamente a stimare tutte le variabili di uscita. Invece, a destra, le variabili di uscita (o anche una parte di esse, in ordine alla dipendenza funzionale) contribuiscono alla stima di una sola variabile di uscita.

L'esempio riportato non è ovviamente esaustivo delle possibili scelte progettuali da un punto di vista topologico, ma vuole essere un modo per spiegare la complessità di tali scelte.

Allo stesso modo la scelta del modello  $MSS_n$  deve essere sempre un prodotto dell'attenta analisi degli elementi che si hanno a disposizione e delle performances minime che si intendono raggiungere. Nel caso in cui ci si voglia basare su modelli neurali la scelta in generale può variare dagli approssimatori funzionali ai classificatori. Spesso, nella pratica progettuale sono da favorire modelli ibridi che possano mettere insieme i vantaggi dei singoli modelli neurali.

Le topologie ibride, come sarà chiaro nel seguito, consentono anche di ridurre l'errore finale dei soft sensors sfruttando metodologie mutuata dalla teoria degli errori classica opportunamente riadattate al caso di strumenti stocastici e quindi con precisione infinita.

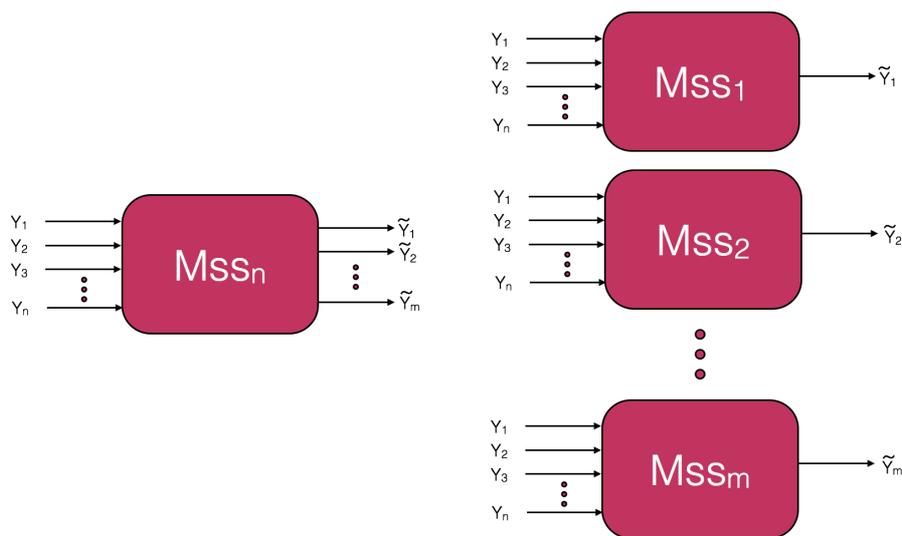


Figura 8 Differenti topologie per la stima del pattern di uscita .

## 2.4 Individuazione del miglior soft sensor.

Se si considera che nella progettazione dei soft sensors è necessario compiere numerose scelte progettuali, risulta evidente che la verifica della bontà della scelta impone una verifica sperimentale. Ovvero, occorre verificare che la scelta operata abbia prodotto dei risultati migliori dell'opzione scartata. Inoltre, se si considerano i soft sensor basati su i modelli neurali, va ricordato che anche avendo scelto la tipologia di rete neurale ed avendone definito la topologia in termini di pattern di ingresso e di uscita, rimangono ancora da stabilire alcuni parametri interni della rete. La numerosità dei soft sensors possibili derivanti da questi parametri è pari al cartesiano dei loro range di variabilità. A titolo di esempio se si pensa ad un perceptrone multistrato possiamo pensare ipotizzare di avere il dubbio sulla scelta di 3 possibili funzioni di attivazione per lo strato hidden e non sapere quanti neuroni assegnare allo stesso

strato ma immaginare che un numero compreso tra 20 e 40 possa essere un valore adeguato. In questo caso saremmo di fronte a  $3 \times 20 = 60$  possibili diversi soft sensors.

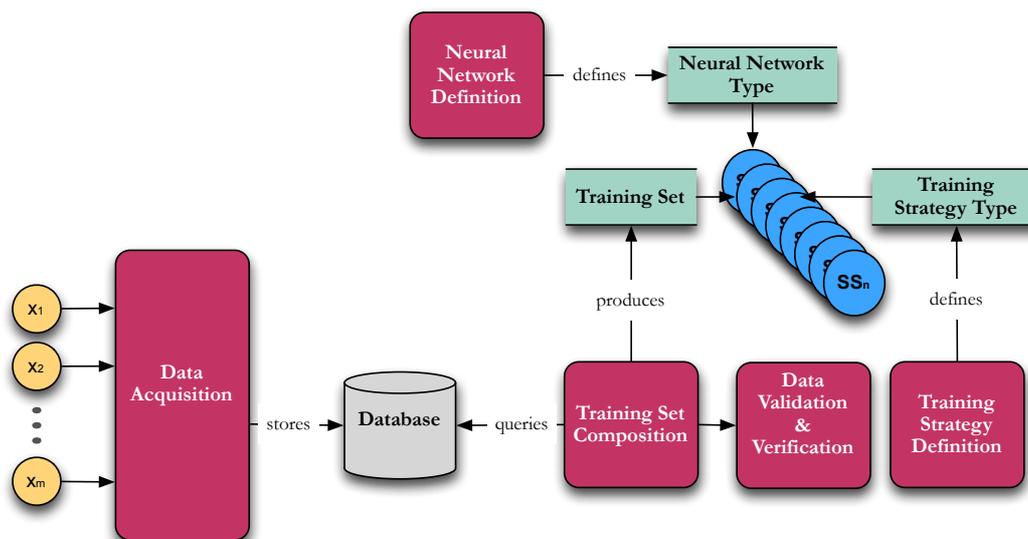
La realtà è ovviamente notevolmente più complessa di questo semplice esempio e le variabili in gioco sono sempre più numerose di quelle prese in considerazione per il perceptrone.

D'altra parte, non esiste un teorema o semplicemente una teoria o metodologia per definire il set di parametri a priori. Pertanto serve una strategia d'indagine per l'identificazione del miglior set di parametri interni ed esterni al modello neurale per ottenere il soft sensor con le migliori prestazioni.

L'utilizzo di metodi di calcolo numerico come il metodo delle bisezioni o delle tangenti possono essere proficuamente utilizzati per coprire il range dei valori dei parametri evitando la "forza bruta" e risparmiando molto tempo e risorse di calcolo.

Per la scelta del miglior soft sensor tra quelli prodotti dalla combinazione di tutti i possibili parametri serve la definizione di un funzionale d'errore. In questo caso, potrebbe essere sufficiente un parametro overall come la deviazione standard della distribuzione d'errore del test set.

Le tre figure successive rappresentano graficamente quanto espresso nei paragrafi precedenti e costituiscono un framework di riferimento per la progettazione dei sensori software.



**Figura 9 Fase di training: I dati acquisiti sono immagazzinati in una base di dati e mediante delle apposite query vengono selezionati e validati per la composizione del training set. Le strategie di definizione dello strumento neurale insieme alle strategie di addestramento producono un insieme di soft sensors da cui dovrà essere selezionato il migliore durante la fase di testing.**

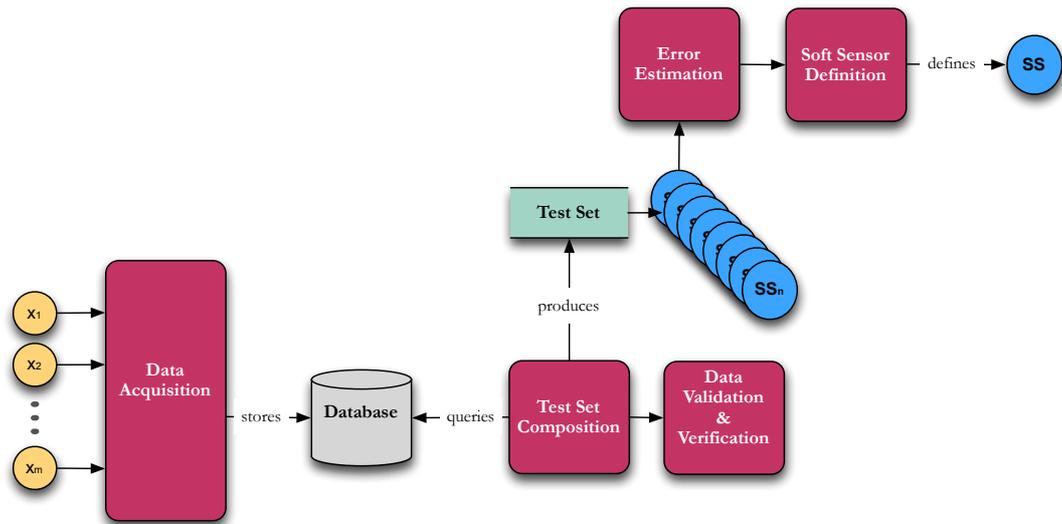


Figura 10 Fase di testing: I dati acquisiti sono immagazzinati in una base di dati e mediante delle apposite query vengono selezionati e validati per la composizione del test set. La misura dell'errore su ogni soft sensor consente di selezionare quello con le migliori performance

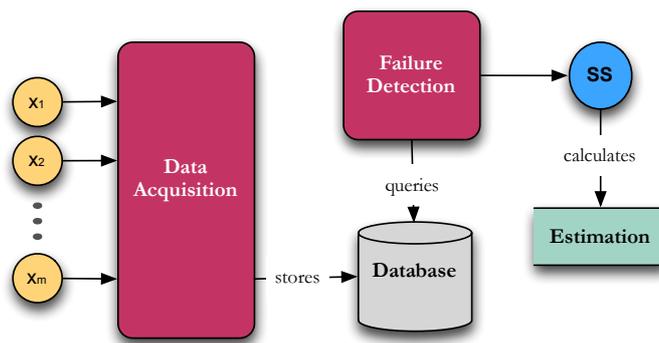


Figura 11 Fase di esercizio: I dati acquisiti sono immagazzinati in una base di dati e mediante delle apposite query vengono selezionati e validati per la fase di esercizio. I dati in ingresso al soft sensor consentono la stima della misura desiderata.

### 3 Misura dell'errore e caratterizzazione statistica dei soft sensors.

Le procedure classiche per la valutazione della performance degli strumenti per la stima della misura, e quindi anche delle reti neurali, consistono nella validazione statistica dei risultati su un test set. Ovvero, dopo aver “definito” è “istruito” il modello  $MSS_n$ , utilizzando un insieme di dati per l'addestramento, si verificano le prestazioni utilizzando un insieme di dati per il test differente dal precedente. Questa procedura produce tipicamente come risultato un vettore di errore che deve essere valutato.

Per ciò che concerne gli strumenti di misura reali, l'errore è sempre definito come:

$$E = |X_m - X_t|$$

dove con  $X_m$  si intende il valore misurato mentre con  $X_t$  si intende il valore vero. In analogia a quanto si definisce per gli strumenti reali di misura, si può definire un errore per gli strumenti software di misura:

$$E_s = |X_e - X_m|$$

dove con  $X_e$  si intende il valore stimato mentre con  $X_m$  si intende il valore misurato. In questo caso il valore  $X_m$  costituisce il valore target che il soft sensor deve raggiungere. In altre parole, se per gli strumenti di misura reali l'errore è misurato rispetto ad un valore teorico corretto, per gli strumenti software di misura questo valore teorico corretto è costituito dal valore misurato dal sensore reale. In questa logica l'errore attribuito al soft sensor è da considerarsi come un errore di sostituzione. Resta inteso che l'errore complessivo è comunque la somma algebrica di  $E + E_s$ .

Nel seguito per misurare le performance dei soft sensor ci si riferirà all'errore di sostituzione. Questo per seguire la logica secondo la quale la performance del soft sensor è tanto migliore quanto più replica il comportamento del sensore reale. Inoltre, il soft sensor non "conosce" l'errore che ha commesso il sensore reale nella misura ma viene "addestrato" sui valori del training set come se questi fossero i valori di riferimento assoluto.

Definito l'errore di sostituzione, si può immediatamente misurare la deviazione standard della distribuzione di errore prodotta dal test set e la rispettiva media. Spesso queste due misure sono le uniche che vengono utilizzate per caratterizzare il comportamento di uno strumento per la stima della misura. Queste misure, tuttavia, costituiscono degli stimatori over all che mal si configurano per dare una descrizione completa del comportamento di uno strumento di misura se pur virtuale.

A questo proposito, l'analisi di quanto è ormai consolidato nelle procedure di caratterizzazione degli strumenti di misura reali può dare delle indicazioni anche per la caratterizzazione metrologica degli strumenti software di misura.

Partendo dalla definizione dell'errore di sostituzione  $E_s$  si possono individuare immediatamente alcuni valutatori statistici overall che caratterizzano la distribuzione di errore:

- $\min(E_s) = \min(|X_e - X_m|)$  ovvero l'errore minimo nella distribuzione;
- $\max(E_s) = \max(|X_e - X_m|)$  ovvero l'errore massimo nella distribuzione;
- $\text{mean}(E_s) = \text{mean}(|X_e - X_m|)$  ovvero la media della distribuzione d'errore;
- $\text{std}(E_s) = \text{std}(|X_e - X_m|)$  ovvero la deviazione standard della distribuzione d'errore;
- $CC(X_e, X_m)$  ovvero il coefficiente di correlazione tra i valori stimati e i valori misurati.

Inoltre, da un punto di vista grafico anche l'istogramma della distribuzione di errore, unitamente ad un test di gaussianità può dare un'informazione aggiuntiva sul comportamento del soft sensor.

Sempre sulla base dell'errore di sostituzione, un altro stimatore overall che può essere individuato è il rapporto  $RR$  calcolato come il rapporto tra i range dei vettori dei valori stimati e quelli misurati nel test set:

$$RR = \frac{|\max(X_e) - \min(X_e)|}{|\max(X_m) - \min(X_m)|}$$

Come mostrato in figura, questo rapporto stabilisce se il soft sensor riproduce (a meno di una traslazione che va interpretata come un bias e quindi eliminabile) fedelmente la scala dei valori oppure la amplifica o la comprime.

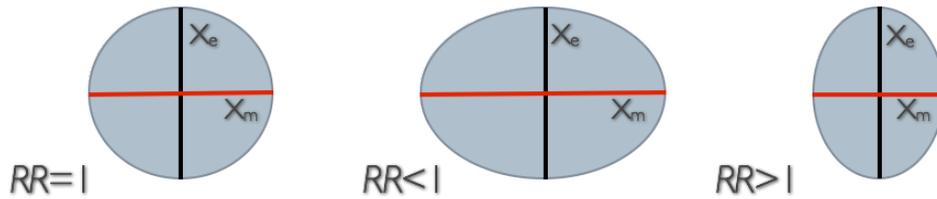


Figura 12 rappresentazione grafica del valutatore RR.

Altri importanti valutatori statistici che possono contribuire alla conoscenza approfondita del comportamento del soft sensor come strumento di misura, sono  $R_{\epsilon}$  e  $R_E$  che misurano rispettivamente il rapporto tra le misure stimate del soft sensor con errore minore di  $\epsilon$  o maggiore di  $E$  e il numero di elementi del test set.

In altre parole rappresentano un indice normalizzato ad 1 rispetto alle soglie  $\epsilon$  e  $E$  di valori accettabili o non accettabili per l'applicazione.

Sempre in analogia a quanto si fa per gli strumenti di misura reali, anche per gli strumenti di misura software si può definire la "classe di precisione" dello strumento. In particolare la classe per gli strumenti software di misura è definita come:

$$C_{sa} = \frac{\text{mean}(|X_e - X_m|)}{|\max(X_e) - \min(X_e)|} * 100$$

Un ulteriore stimatore, questa volta più analitico, si può realizzare analizzando l'andamento dell'errore di sostituzione in funzione delle parametri di ingresso del soft sensor.

Se si considera il modello  $MSS_n$ , in cui  $Y_m$  rappresenta il vettore delle di ingresso e  $\bar{Y}_n$  rappresenta il vettore delle variabili stimate, se si considera anche l'insieme delle coppie ingresso uscita che formano il test set, suddividendo il range del parametro  $Y_i$  (uno dei possibili  $Y_m$ ) in  $C$  sottointervalli  $I_c$  di ugual misura, conseguentemente il test set è suddiviso in  $C$  sottoinsiemi e per ogni  $I_c$  è possibile misurare il corrispettivo vettore rappresentativo dell'errore di sostituzione  $E_s^c$ . Quindi per ogni  $E_s^c$  è possibile calcolare tutti gli stimatori precedentemente definiti.

L'uso di un simile stimatore consente di definire l'intervallo di confidenza dell'errore in funzione di un dato parametro d'ingresso.

A titolo di esempio si consideri un soft sensor addestrato per misurare la temperatura di contatto  $T_c$  in un certo punto di un monumento, si supponga che questa temperatura venga stimata sulla base della temperatura dell'aria  $T_a$  misurata su media oraria  $h$  da un sensore posto ad una certa distanza.

In questo particolare caso il modello  $MSS_n$  ha come variabili di ingresso  $Y_m$  le grandezze  $h$  e  $T_a$  mentre l'uscita  $\bar{Y}_n$  è costituita dalla sola grandezza  $T_c$ . Ciò posto, suddividendo il range del parametro  $h$ , ad esempio, in  $C$  sotto intervalli  $I_c$  di ugual misura, conseguentemente, come già detto il test set risulta automaticamente suddiviso in  $C$  sottoinsiemi e per ogni  $I_c$ . Quindi dividendo il range di  $h$  in 23 sottointervalli, uno per ogni ora del giorno si possono ottenere 23  $E_s^c$  e quindi misurare ad esempio la deviazione standard e la media di ogni  $E_s^c$  e diagrammarla in funzione delle ore.

Nella figura successiva sono riportati, con la linea continua, gli andamenti di  $\text{mean}(E_s^c)$  e  $\text{std}(E_s^c)$ , con la linea tratteggiata sono riportati anche gli andamenti della media e della

deviazione standard di  $E_s^c$  relativi a quello che può definirsi il sensore nullo, soft sensors il cui la funzione di trasferimento è una costante.

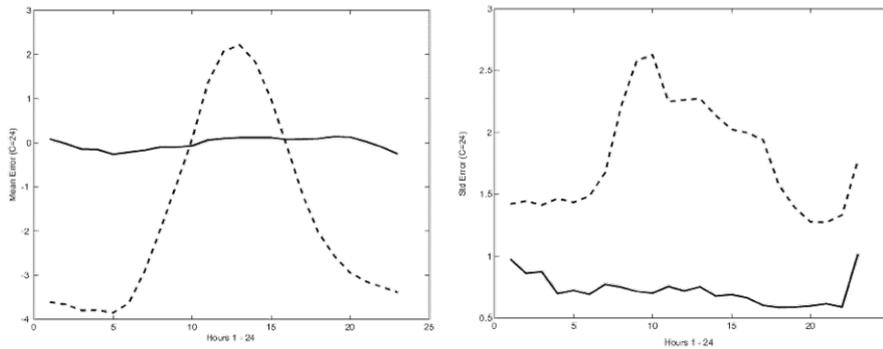


Figura 13 rappresentazione grafica degli adamanti di  $\text{mean}(E_s^c)$  e  $\text{std}(E_s^c)$ .

L'insieme di tutti questi valutatori statistici sia di tipo overall sia di tipo locale, insieme a i test di gaussianità della distribuzione di errore e l'istogramma della stessa, costituiscono una caratterizzazione statistica per i soft sensor che da un punto di vista metrologico e li rendo simili agli strumenti reali di misura.

#### 4 Ottimizzazione delle prestazioni dei soft sensor.

Nella teoria degli errori, la precisione è il grado di "convergenza" (o "dispersione") di dati rilevati individualmente (campione) rispetto al valore medio della serie cui appartengono. Nella stessa teoria degli errori, l'accuratezza è il grado di corrispondenza del dato teorico, desumibile da una serie di valori misurati (campione di dati), con il dato reale o di riferimento, ovvero la differenza tra valor medio campionario e valore vero o di riferimento.

Quanto detto si può riassumere da un punto di vista grafico nella figura successiva.

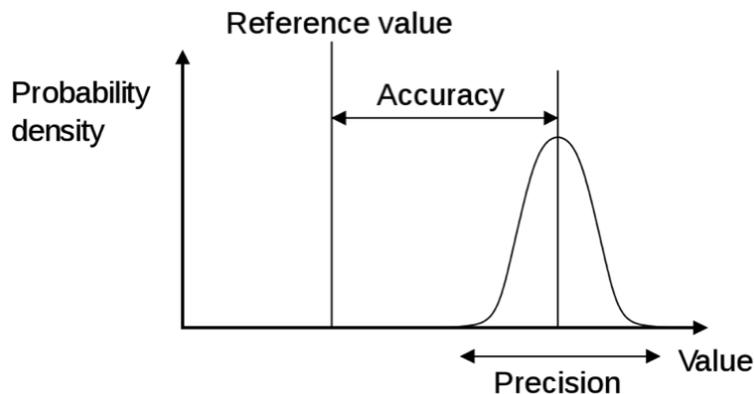


Figura 14 rappresentazione grafica del concetto di precisione e accuratezza

Quindi l'accuratezza rappresenta quanto mediamente (ripetendo  $n$  misure) lo strumento misura il valore di riferimento, mentre la precisione rappresenta la dispersione delle misure intorno al valor medio.

Resta sottinteso che per gli strumenti reali accuratezza e precisione si calcolano ripetendo delle misure di riferimento sempre nelle stesse condizioni ambientali.

Per quello che riguarda gli strumenti software di misura i concetti di precisione e accuratezza devono essere reinterpretati. Infatti, essendo sempre strumenti deterministici, ovvero strumenti che ad un pattern di ingresso fanno corrispondere sempre lo stesso pattern di uscita, il concetto di precisione perde automaticamente di senso. Oppure, forse più correttamente, va detto che sono strumenti a precisione infinita.

In altri termini questi strumenti sono come un arciere, che tirando su un bersaglio tira la freccia sempre nello stesso punto sbagliando ogni volta della stessa quantità.

Ciò posto, una tecnica molto semplice, adottata in caso di misure con strumenti reali, per ridurre l'errore di misura è quella di operare diverse misure con il medesimo strumento e nelle stesse condizioni ed operare la media di queste misure. Ad esempio supponendo di dover misurare una tensione con un voltmetro, ripetendo la misura di questa tensione alcune volte, la media di queste misure dovrebbe risultare più precisa delle singole misure.

Formalmente, quanto qualitativamente detto sopra si può esprimere come segue: l'effettuare  $n$  misure di una stessa grandezza, ovvero effettuare  $n$  estrazioni da una distribuzione di varianza  $\sigma^2$ , e costruire la media aritmetica del campione, equivale ad eseguire un'unica osservazione da una distribuzione che ha la stessa media teorica ( $m$ ) ed una varianza  $n$  volte più piccola  $\sigma^2/n$ .

Nel caso di strumenti software di misura questa tecnica non è ovviamente applicabile a causa della precisione infinita. Infatti, ripetendo la stima  $n$  volte otterremmo sempre la medesima misura, ovvero stimando  $n$  volte il pattern di uscita del modello  $MSS_n$  si otterrebbe sempre lo stesso pattern di uscita e pertanto questa operazione risulterebbe inutile al fine della riduzione dell'errore.

Tuttavia, esiste anche un altro modo per ridurre l'errore. Si consideri la situazione in cui 5 persone possiedono un orologio e questi 5 orologi misurano, come è plausibile pensare, 5 orari differenti. In queste circostanze la media dei 5 differenti orari è, ancora, più precisa delle singole misurazioni? La risposta è sì solo se i 5 orologi hanno caratteristiche simili. In termini più formali la risposta è sì se i 5 orologi appartengono alla medesima classe di precisione e pertanto presentano una deviazione standard della loro distribuzione d'errore, paragonabile.

Questo risultato, valido per gli strumenti reali di misura, si può applicare anche agli strumenti software di misura. Infatti, se anziché considerare un solo modello  $MSS_n$  per la stima delle grandezze desiderate si considerano più modelli  $MSS_n$ , e se si verifica che tali modelli siano linearmente indipendenti e al contempo producano risultati che presentano una distribuzione d'errore paragonabile, allora l'analogia con l'esempio dei 5 orologi è corretta.

In queste condizioni, una strategia per ridurre l'errore di sostituzione potrebbe essere quindi quella di stimare il pattern d'ingresso con  $n$  Soft Sensors linearmente indipendenti appartenenti alla medesima classe di precisione e mediare le uscite.

Nel caso di soft sensors basati su modelli neurali, come mostrato in figura, reti di tipo Elman, Radial basis function e reti di tipo Alpha-net, in determinati casi possono offrire questa opportunità poiché sono in grado di lavorare come approssimatori funzionali raggiungendo prestazioni analoghe.

Resta sottinteso che il test set deve essere il medesimo per verificare le condizioni di indipendenza lineare e di paragonabile deviazione standard della distribuzione di errore.

In linea teorica si potrebbe anche pensare di utilizzare un unico modello neurale per generare numerosi soft sensors che abbiano prestazioni simili e supporre la lineare indipendenza (o la quasi lineare indipendenza) sfruttando la circostanza che gli algoritmi di addestramento delle reti neurali, partono da una distribuzione dei pesi e dei bias random. Pertanto, il punto di partenza per l'addestramento è sempre differente. Inoltre, all'aumentare del numero dei neuroni aumenta anche la complessità del funzionale d'errore, ciò spesso implica il crescere di minimi locali. Tutto ciò può indurre a pensare che due modelli neurali uguali, sebbene addestrati sul medesimo training set, difficilmente sono uguali o solo linearmente dipendenti.

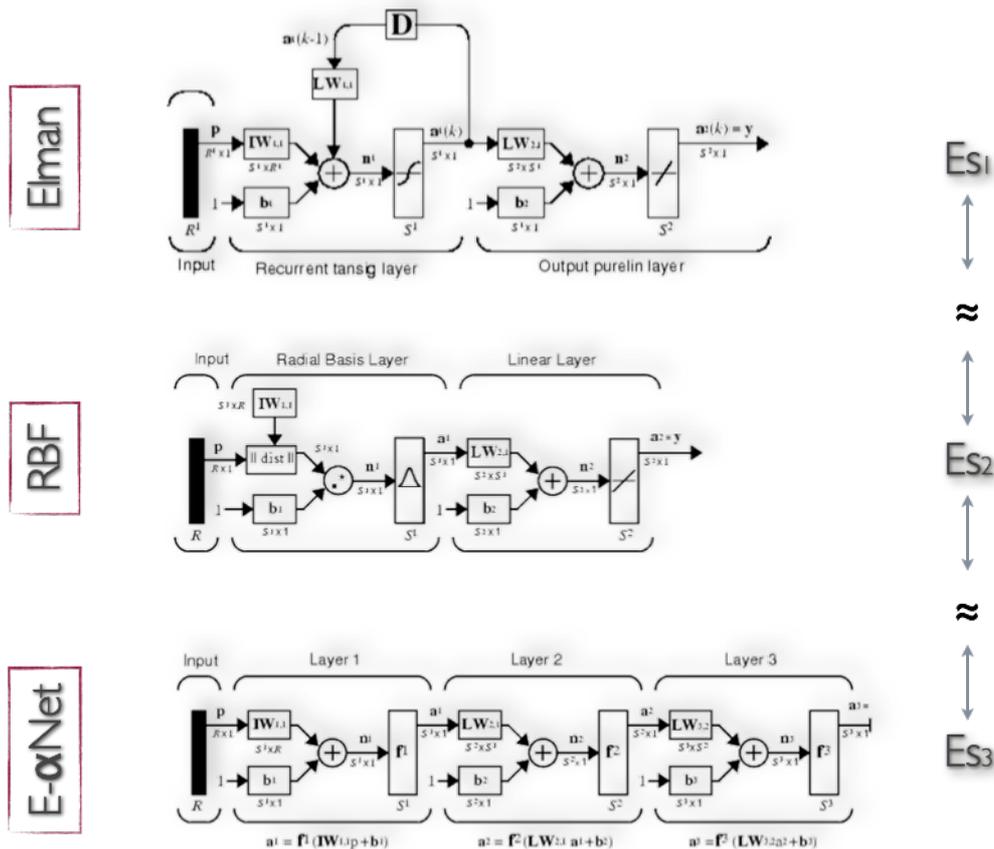


Figura 15 Esempio di tre modelli neurali Radial Basis Function, Elman Neural Network e E-AlphaNet che possono verificare le condizioni di indipendenza lineare e di paragonabile deviazione standard della distribuzione di errore.

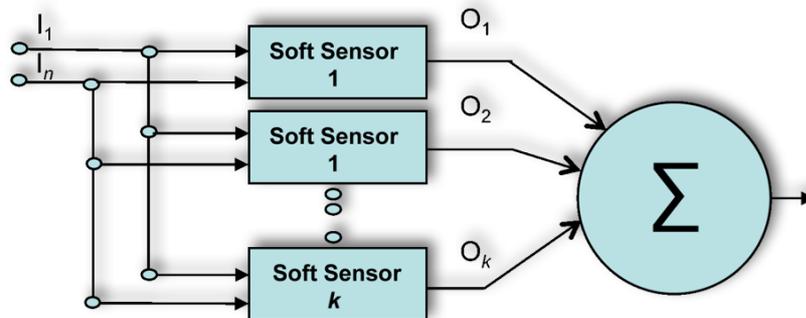


Figura 16 Schema topologico di un sistema MultiSoft Sensor che utilizza l'operazione di media per ottimizzare le prestazioni

Un ulteriore incremento delle prestazioni si può, in linea teorica, ottenere superando l'operazione di media e sostituendo la stessa con un'operazione di selezione.

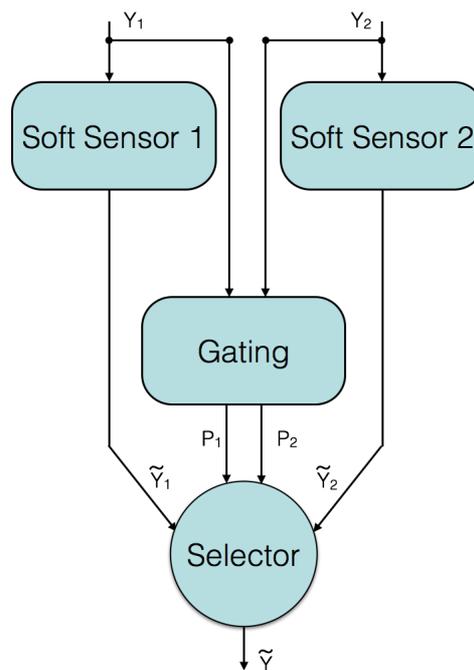
Se si considera la possibilità di stimare la medesima variabile con due soft sensors linearmente indipendenti ed appartenenti alla medesima classe di precisione, esisterà un subset del training set per cui uno dei due stima detta variabile in modo più accurato dell'altro.

Ovvero, per certi valori del pattern di ingresso risponde meglio un soft sensor mentre per altri valori risponde meglio l'altro.

Questa attitudine di un soft sensor a stimare meglio certi valori piuttosto che altri è ovviamente legata alla fase di training. Pertanto, si può pensare di addestrare una rete, che chiameremo Gating Network, in grado di generare la probabilità di successo di un soft sensor, rispetto all'altro, in funzione dei parametri di ingresso. Successivamente, un selettore stocastico sceglie quale soft sensor stima in modo più preciso la variabile desiderata.

Formalmente, con riferimento alla simbologia della figura:

$$P_1 = \frac{P(\tilde{Y}_1|Y_1, Y_2)}{P(\tilde{Y}_1|Y_1, Y_2) + P(\tilde{Y}_2|Y_1, Y_2)} \quad P_2 = \frac{P(\tilde{Y}_2|Y_1, Y_2)}{P(\tilde{Y}_1|Y_1, Y_2) + P(\tilde{Y}_2|Y_1, Y_2)}$$



**Figura 17** La Gating Network genera la probabilità di successo dei 2 soft sensors e il selettore stocastico a 2 ingressi sceglie il soft sensor che stima con maggiore precisione la misura.

In alternativa all'operazione di selezione si può anche pensare di utilizzare le probabilità condizionali generate dalla gating network come pesi di un'operazione di media. In questa operazione di data fusion si darebbe maggior importanza alla stima ottenuta dal soft sensor che ha maggiore probabilità di successo rispetto a quello con minore probabilità.

L'esempio condotto con due soft sensors è ovviamente estendibile al caso di un numero maggiore di soft sensors. Come riportato nello schema della figura seguente.

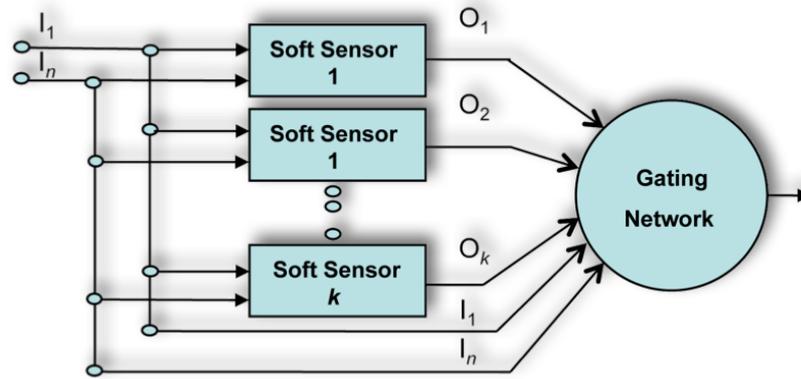


Figura 18 La Gating Network genera la probabilità di successo dei K soft sensors.

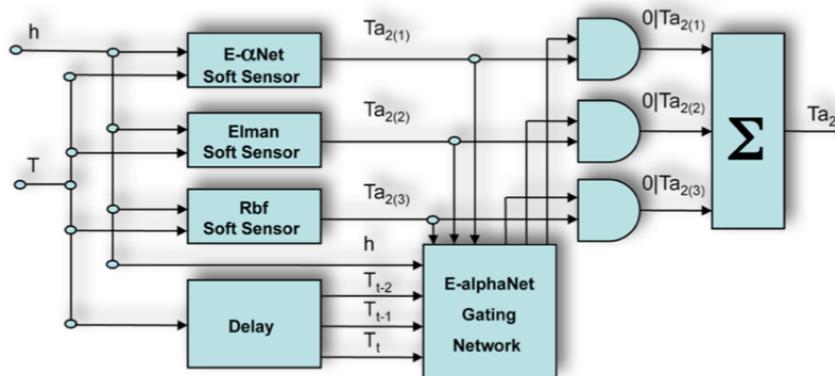


Figura 19 Esempio specifico di un multisensor per la stima della temperatura ambiente basato su tre modelli neurali e una gating network.

## Bibliografia

1. Maniscalco U. and Rizzo R., Adding a Virtual Layer in a Sensor Network to Improve Measurement Reliability, In Advanced Mathematical and Computational Tools in Metrology IX, Series on Advanced in Mathematics for Applied Sciences. Pavese et al., Editors. World Scientific, Singapore. 2014. ISSN : 1793-0901
2. Maniscalco U., Pilato G. Soft Sensors: The New Frontier in Measurement for Non-Invasive Monitoring of Cultural Heritage Sites. ERCIM News 2011, N. 86. ISSN 0926-498;
3. Maniscalco U., Pilato G. Multi soft-sensor data fusion in spatial forecasting of environmental parameters. In Advanced Mathematical and Computational Tools in Metrology IX, Series on Advanced in Mathematics for Applied Sciences. Pavese et al., Editors. Vol. 72, Page(s): 142–150. World Scientific, Singapore. 2011. ISSN : 1793-0901
4. Maniscalco U., Pilato G., Vassallo G., Soft Sensor based on E- $\alpha$ NETs, In: Frontiers in Artificial Intelligence and Applications. Vol. 226, 2010. pp. 172-179. Edited by Bruno Apolloni, Simone Bassis, Carlo F. Morabito. ISSN 0922-6389.

5. Ciarlini, P. and Maniscalco, U. 2008. Wavelets and Elman Neural Networks for monitoring environmental variables. *Journal of Computational and Applied Mathematics*. Vol. 221, N. 2 (Nov. 2008), Page(s): 302-309. Impact Factor: 1.048, 5-Year Impact Factor: 1.138
6. Ciarlini, P. and Maniscalco, U. 2008. Wavelet and Mixture of Soft Sensors to Improve the Monitoring of Environmental Parameters By Neural Network. 16th Symposium on Electrical Measurements and Instrumentation & 13th Workshop on ADC Modelling and Testing (TC4), Florence, ITALY, 2008
7. Chella, A., Ciarlini, P. and Maniscalco, U. 2006. Neural Networks as Soft Sensors: a Comparison in a Real World Application. *International Joint Conference on mNeural Networks, IJCNN '06*. 16-21, July 2006, Vancouver, BC, Canada. Page(s):2662 – 2668. ISBN: 0-7803-9490-9.
8. Maniscalco, U., Ciarlini, P. and Regoliosi, G. 2006. Validation of soft sensors in monitoring ambient parameters. In *Advanced Mathematical and Computational Tools in Metrology VII*, Series on Advanced in Mathematics for Applied Sciences. Ciarlini et al., Editors. Vol. 72, Page(s): 142–150. World Scientific, Singapore. 2006. ISBN: 978-981-256-674-4.
9. Maniscalco, U., 2005. A Connectionist System of Virtual Sensors to Support the Monitoring of Cultural Heritage Damage. In *Biological and Artificial Intelligence Environments*, ed. Bruno Apolloni, Maria Marinaro and Roberto Tagliaferri. pp 343-350. ISBN 978-1-4020-3431-2.