



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
CORSO DI LAUREA IN INGEGNERIA INFORMATICA

**“Machine learning applicato alla modellistica dei circuiti:
stima del Remaining Useful Lifetime”**

Relatore: Prof. / Dott Enrico Zanoni

Laureando/a: Denise Marogna

ANNO ACCADEMICO 2021 – 2022

Data di laurea 22 Luglio 2022

Indice

| | |
|--|-----------|
| 1. Introduzione..... | 3 |
| 2. Machine learning e reti neurali..... | 5 |
| 2.1 Definizione di machine learning..... | 5 |
| 2.2 I modelli del machine learning..... | 6 |
| 2.3 Le reti neurali..... | 7 |
| 2.4 Tipologie di reti neurali..... | 9 |
| 3. La modellistica dei circuiti..... | 10 |
| 3.1 Definizione..... | 10 |
| 3.2 Machine learning applicato alla modellistica..... | 11 |
| 3.3 Machine learning e MOSFET..... | 14 |
| 4. Il Remaining Useful Lifetime..... | 16 |
| 4.1 Definizione..... | 16 |
| 4.2 Stima con i filtri di Kalman..... | 17 |
| 5. Stima del R.U.L. con reti neurali artificiali..... | 20 |
| 5.1 Implementazione della rete neurale..... | 20 |
| 5.2 Stima del Remaining Useful Lifetime..... | 22 |
| 5.3 Discussione dei risultati..... | 24 |
| 5.4 Che tipologia di rete neurale utilizzare?..... | 25 |
| 6. Conclusioni..... | 28 |
| 7. Bibliografia..... | 30 |

Capitolo 1:

Introduzione

Nel corso degli ultimi anni, l'apprendimento automatico (o machine learning), ha subito notevoli sviluppi, con la conseguenza di essere sempre più frequentemente utilizzato in una grande varietà di discipline. Questa branca dell'intelligenza artificiale, infatti, consente di ottenere un notevole miglioramento delle prestazioni rispetto ai metodi utilizzati in precedenza.

La definizione di machine learning comprende anche il modello delle reti neurali. Una rete neurale artificiale è un sistema che simula il funzionamento del cervello umano, basandosi sulla sua struttura costituita da neuroni collegati tra loro tramite delle sinapsi. Partendo da informazioni apprese in fase di addestramento, tale tipo di modello è in grado di compiere decisioni in autonomia e ricavare nuove informazioni da quelle che già possiede. Con il graduale approfondimento e sviluppo delle reti neurali, il machine learning si è sempre più affermato in diverse aree quali la biologia, la matematica, la fisica, l'elettronica e molte altre.

In particolare, sono state studiate diverse applicazioni dell'apprendimento automatico nel campo dei circuiti elettronici, soprattutto in ambiti specifici come design, modellistica e processo di produzione. Questi, sono infatti controllati da diversi parametri, la cui correttezza è molto importante, in quanto una loro leggera deviazione potrebbe compromettere il funzionamento dell'intero dispositivo. Molte aziende produttrici stanno progressivamente implementando tale tecnica, in quanto ha la capacità di velocizzare alcune parti del processo ma, cosa di maggiore importanza, non comporta nessun aumento dei costi.

Un parametro molto importante per studiare il corretto funzionamento dei circuiti elettronici è il Remaining Useful Lifetime, tempo di vita che viene utilizzato come indicatore per prevedere la degradazione dei dispositivi sottoponendoli a test di invecchiamento accelerato.

Solitamente, questo parametro viene calcolato attraverso diversi metodi analitici, ma, grazie al progressivo sviluppo del machine learning, è nata la possibilità di trarne una stima con maggiore rapidità ma, soprattutto, con migliore accuratezza. Tale procedimento avviene tramite l'impiego delle reti neurali artificiali che, una volta addestrate, riescono a stimare il RUL in diverse condizioni, evitando così simulazioni dipendenti dall'intervento umano che richiederebbero molto più tempo e capacità di calcolo, senza considerare i numerosi errori in cui ci si potrebbe imbattere.

Nei seguenti capitoli, dopo un'introduzione sull'apprendimento automatico e le reti neurali artificiali, verrà illustrata una particolare applicazione di quest'ultime per la stima del Remaining Useful Lifetime in un transistor MOSFET. Saranno anche esposte brevemente alcune metodologie utilizzate in precedenza per tale calcolo, quindi senza l'ausilio dell'intelligenza artificiale, per poter comparare entrambe le tecniche in termini di prestazioni e accuratezza. Seguirà una riflessione su eventuali problematiche relative a questa applicazione, con l'obiettivo di ipotizzare delle soluzioni ricorrendo a diverse topologie di reti neurali.

Capitolo 2:

Machine learning e reti neurali

2.1 – Definizione di machine learning

Il machine learning può essere definito [1] come un processo di costruzione di sistemi informatici che automaticamente migliorano con l'esperienza e che implementano un processo di apprendimento. Tale modello addestrato può procurare nuove conoscenze su come le variabili in input vengono mappate nell'output e può essere utilizzato per fare previsioni per valori che non fanno parte di quelli utilizzati per l'addestramento.

Le quantità osservabili che sono fornite in ingresso ad un algoritmo di apprendimento automatico vengono chiamate "features" [2]. L'algoritmo costruisce una mappatura a partire da queste features per ottenere le variabili desiderate in output, regolando i parametri del modello. Per questo motivo, è importante che le features siano rilevanti per la previsione dell'uscita, in quanto potrebbero peggiorare le performance in termini di tempo e memoria.

Nel machine learning, vi sono tre tipi di apprendimento:

- *Apprendimento supervisionato*: è una tecnica per apprendere una funzione a partire dai dati inseriti per l'addestramento. Questi consistono in coppie di oggetti in input e output desiderati. Le uscite possono essere un valore continuo (regressione) o possono prevedere l'etichetta della classe di un oggetto posto in ingresso (classificazione). L'obiettivo è di prevedere il valore della funzione per ogni oggetto in input valido dopo aver visto un certo numero di esempi in fase di addestramento.
- *Apprendimento non-supervisionato*: è una tecnica dove le etichette manuali degli ingressi non sono utilizzate. Vengono dati solo un training set di vettori senza i valori della funzione. Un set di dati in input viene raggruppato, e poi questo metodo li

gestisce come un set di variabili casuali. Infine, raggruppa gli oggetti simili e crea cluster o associazioni tra di essi.

- *Apprendimento semi-supervisionato*: mentre addestrano un modello predittivo, questi algoritmi possono sfruttare sia i valori di output supervisionati che la distribuzione dei dati in quelli non supervisionati. A causa della parte di dati non etichettata, vengono fatte ipotesi sul valore di questi, quindi non sempre questo metodo si adatta ad ogni tipo di modello.
- *Apprendimento per rinforzo*: i compiti sono divisi in azioni elementari e quando una di queste viene completata con successo, viene data una ricompensa all'agente. Questo meccanismo si ripete fino a che la macchina non completa l'intero compito.

L'abilità di un modello di prevedere l'output corretto dati nuovi campioni dopo l'apprendimento, viene definita generalizzazione. La capacità di generalizzazione di un modello può essere valutata solo utilizzando campioni esterni al training set. Per rispettare questo vincolo, di solito si

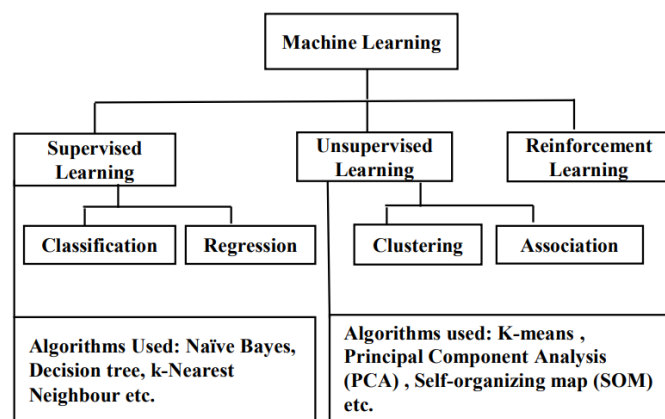


Figura 1 - Classificazione dei processi del machine learning

dividono i dati per l'addestramento in due parti. La prima parte viene chiamata training set e solitamente rappresenta la porzione maggiore dei dati (circa il 70%). La seconda parte viene chiamata validation set, e viene utilizzata per calcolare l'errore per quel modello con lo scopo di testare la sua generalizzazione. Il modello che restituisce l'errore minore nel validation set viene considerato il migliore. Inoltre, mentre riportiamo la performance attesa del nostro modello addestrato, utilizziamo un terzo set, il test set, il quale contiene esempi non utilizzati negli altri due. Un modello può risultare inadeguato (underfit) se l'errore riscontrato nel training set non è abbastanza basso. Al contrario, un modello può essere in condizione di overfit se l'errore dell'addestramento e quello del test set hanno troppa differenza tra loro.

2.2 – I modelli del machine learning

L'apprendimento automatico presenta diversi tipi di modelli [3], che si differenziano tra loro sia per il tipo di apprendimento che utilizzano, sia per gli scopi per cui vengono impiegati:

- *Macchine a vettori di supporto*: insieme di metodi di apprendimento supervisionato utilizzati per la classificazione e la regressione dei pattern. Costruiscono un iperpiano N-dimensionale che separa i dati in due categorie in modo ottimo, per poi riuscire a prevedere a quale delle due appartiene un nuovo esempio di input.
- *Alberi di decisione*: modello predittivo basato sull'apprendimento supervisionato, che ha lo scopo di approssimare una funzione obiettivo discreta. Ogni nodo interno rappresenta una variabile, un arco rappresenta un possibile valore per essa, e una foglia il valore previsto per la variabile obiettivo.
- *Programmazione genetica*: algoritmo euristico utilizzato per la risoluzione di problemi di ottimizzazione. Esso è basato sul principio dell'evoluzione delle specie e su quello della selezione naturale.
- *Reti bayesiane*: modello grafico che si basa sulla considerazione delle variabili in esame come quantità probabilistiche. In questo modo, prende decisioni ottime basandosi sull'analisi delle probabilità di queste.
- *Clustering*: metodo di apprendimento non supervisionato, che analizza i dati con lo scopo di raggruppare elementi simili in un insieme di dati.
- *Reti neurali artificiali*: sistema adattativo che evolve la sua struttura basandosi su informazioni esterne assunte durante la fase di addestramento.

2.3 – Le reti neurali

Una rete neurale artificiale è un modello matematico che cerca di simulare il comportamento dei neuroni interconnessi che formano il cervello umano, i quali sono in grado di imparare e prendere decisioni. Questo particolare sistema gode di diverse caratteristiche, tra cui un elevato parallelismo, una conservazione dei dati distribuita e, di conseguenza, una consistente resistenza ai guasti.

L'elemento basilare di questo modello è il perceptrone, un tipo di classificatore binario che mappa i suoi ingressi in un valore in uscita dato da:

$$f(x) = \varphi(\langle w, x \rangle + b)$$

dove w è un vettore di pesi, l'operatore $\langle \cdot, \cdot \rangle$ è l'operazione di prodotto scalare e b è il bias, un ulteriore peso che si considera collegato a un input fittizio con valore sempre uguale a 1.

Quest'ultimo è utile per tarare il punto ottimale di lavoro del neurone. Il simbolo φ

rappresenta la funzione di attivazione, cioè il modo in cui le informazioni in ingresso vengono processate dal neurone. Esistono diversi tipi di funzione di attivazione, come il gradino, la rampa, la sigmoide, la tangente parabolica e molti altri.

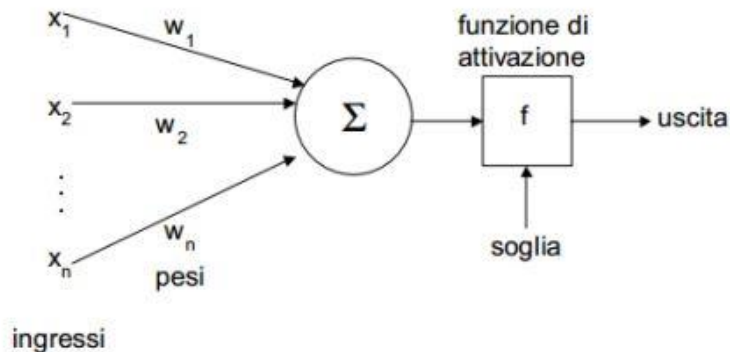


Figura 2 - Struttura del perceptrone

Una rete è costituita da diversi livelli, o layers:

- *Input layer*: costituito da numerosi neuroni che portano i dati in input verso altri livelli interni processandoli.
- *Hidden layer*: i neuroni ricevono in ingresso una serie di input pesati e producono un output attraverso la funzione di attivazione.
- *Output layer*: produce i risultati finali del sistema.

Questi modelli possono essere costituiti da uno o più livelli nascosti.

Il metodo più comunemente utilizzato per l'addestramento di una rete è l'algoritmo di retropropagazione dell'errore, o backpropagation. Esso sfrutta l'errore che si ottiene comparando gli output effettivi con quelli desiderati, utilizzandolo poi per modificare i pesi delle connessioni in ingresso e in uscita e, quindi, migliorare il risultato. Uno dei suoi problemi principali è la scomparsa del gradiente, fenomeno causato dall'impiego di funzioni di attivazione non lineari che, all'aumentare della profondità della rete neurale, fanno diminuire esponenzialmente il valore del gradiente.

Le reti neurali artificiali presentano due principali topologie:

- *Feedforward*: il flusso di informazioni scorre unicamente in avanti. Un'unità invia informazioni ad un'altra ma non ne riceve di ritorno. Hanno input e output fissati.
- *Feedback*: i segnali possono viaggiare in entrambe le direzioni. Rimangono in un punto di equilibrio fino a che non viene inserito un nuovo ingresso, quindi se ne cerca

uno nuovo. Questa topologia viene denominata anche ricorrente, in quanto si ha un ritorno di informazione.

2.4 – Tipologie di reti neurali

Esistono diverse tipologie di reti neurali, ognuna delle quali trova applicazioni specifiche grazie alla peculiarità della sua struttura.

- *Percettrone multistrato*: si tratta di una rete neurale artificiale costituita da diversi livelli di molteplici percettroni. Ogni neurone utilizza una funzione di attivazione lineare, tranne per i nodi di ingresso. La sua struttura presenta strati multipli di nodi all'interno di un grafo diretto, con ogni strato completamente connesso al successivo. Il vantaggio di questa rete è che può distinguere i dati che non sono linearmente separabili.
- *Reti ricorrenti*: particolare classe di rete neurale in cui i neuroni sono collegati tra loro in un loop. Solitamente, i valori di output di un livello superiore vengono utilizzati come input di un livello inferiore. Questa particolare conformazione consente alla rete di utilizzare uno degli strati come memoria di stato. Ciò fornisce a questa tipologia di rete l'abilità di gestire in maniera corretta situazioni in cui serve mantenere un contesto per elaborare correttamente le informazioni.
- *Reti convoluzionali*: tipologia di rete neurale feedforward costituita da più stadi e, similmente a quanto succede nella corteccia visiva, ogni stadio è specializzato in mansioni diverse. Si differenzia dalle altre per l'implementazione di livelli di convoluzione, i quali estraggono attraverso l'ausilio di filtri le caratteristiche delle immagini di cui si vuole analizzare il contenuto.
- *Reti profonde*: funziona esattamente come una rete neurale artificiale, ma necessita un numero molto elevato di livelli nascosti di neuroni intermedi. Grazie al maggior numero di livelli, si ha una potenza computazionale notevolmente maggiore, quindi risulta vantaggiosa nella gestione di reti più grandi.

Capitolo 3

La modellistica dei circuiti

3.1 – Definizione

La modellistica dei circuiti elettrici consiste nell'utilizzo di modelli matematici per replicare il comportamento di un dispositivo reale. Un simulatore circuitale consente, invece, di prevedere il funzionamento di un circuito, le sue prestazioni ed i suoi limiti e pertanto di apportare modifiche al progetto stesso in una fase preliminare, prima ancora che esso sia fisicamente realizzato. Questo può consentire di limitare le fasi di prototipazione con enormi vantaggi in termini di costi e tempi di sviluppo.

Un ulteriore vantaggio che questo strumento conferisce, è la possibilità di scomporre un progetto in più moduli circuitali e di simulare ciascuno di essi separatamente dagli altri. Questo rende i simulatori molto adatti a fasi della progettazione in cui la modularità semplifica notevolmente la costruzione dei dispositivi.

Uno dei simulatori più utilizzati nel mondo dell'elettronica è Spice. Storicamente il concetto di modello Spice nasce da un'iniziativa dell'Electronics Research Laboratory dell'Università di Berkeley in California che mirava a realizzare degli strumenti CAD per la simulazione dei circuiti elettronici. I software e i modelli di componenti elettronici che rispettano questo standard sono da allora definiti rispettivamente software e modelli Spice.

Il modello di un componente altro non è che una descrizione matematico-circuitale dello stesso che consente di simularne il comportamento elettrico ai morsetti. Esso può essere di diversa natura e più o meno complesso ed elaborato, a seconda dei fenomeni fisici di cui tiene conto. Se il modello è orientato a descrivere determinati fenomeni o effetti, questi saranno contemplati nella simulazione e risulteranno visibili e indagabili nei risultati, altrimenti il

componente si comporterà come ideale da quel punto di vista e determinati fenomeni non appariranno evidenti, non perché non presenti nel dispositivo reale ma semplicemente perché non di interesse nel caso particolare studiato. Per questo motivo è spesso necessario comprendere i limiti dei modelli che si stanno utilizzando al fine di come questi influenzeranno i risultati ottenuti attraverso le simulazioni ed evitare di ritenere corretti risultati che in realtà non possono esserlo.

Quasi tutti i simulatori circuitali consentono, inoltre, di condurre simulazioni parametriche che permettono di studiare il funzionamento di un circuito al variare di uno specifico parametro elettrico di un determinato componente. A seconda degli algoritmi e dei modelli di dispositivi integrati, nel simulatore può inoltre essere possibile o meno simulare effetti di rumore e fenomeni di deriva termica. Il modello di componenti, come condensatori e resistori, può risultare relativamente semplice, componenti come transistor possono, invece, essere modellizzati attraverso resistori, condensatori e generatori pilotati, mentre componenti più complessi, come operazionali e circuiti integrati specifici, possono essere rappresentati da veri e propri macromodelli. In ogni caso, non va mai dimenticato che una simulazione circuitale altro non è che un'approssimazione più o meno accurata della realtà.

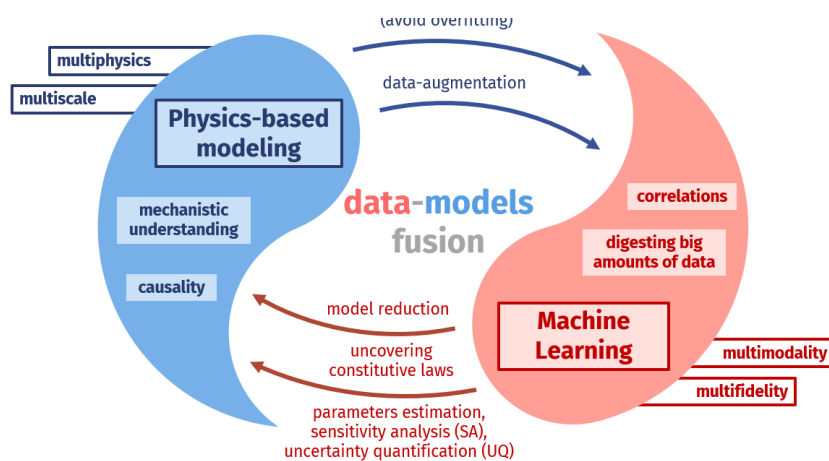
3.2 – Machine learning applicato alla modellistica

I processi di design, modellistica e manifattura dei dispositivi elettronici richiedono diversi parametri di controllo di grande importanza. Una leggera deviazione di questi parametri chiave potrebbe cambiare le performance del dispositivo stesso.

Analizzare la correlazione tra i parametri di processo e i parametri del dispositivo sta diventando difficile utilizzando i metodi convenzionali. Diventa ancora più difficile se i parametri in output variano non linearmente con i parametri di processo. Inoltre, sviluppare i modelli dei dispositivi richiede grande impegno e tempo dopo un iniziale sviluppo della tecnologia. Per risolvere questo problema, sono stati costruiti dai ricercatori dei modelli veloci e accurati basati sui dati utilizzando il machine learning per analizzare le caratteristiche del dispositivo e ottimizzarle.

Usando tecniche di automatizzazione è possibile rendere il processo di produzione più veloce e accurato. Più importante, questo strumento non incrementa il costo del processo di manifattura. Per questo motivo, il machine learning ha acquisito notevole importanza in diversi ambiti riguardanti lo sviluppo di dispositivi.

Figura 3 - Schema che evidenzia i vantaggi della modellistica con l'impiego di reti neurali



In particolare, vengono utilizzate le reti neurali. Queste, una volta addestrate, possono essere utilizzate nella simulazione e nella progettazione di alto livello, fornendo risposte rapide al compito che hanno appreso. Le reti neurali

sono alternative efficienti ai metodi convenzionali come i metodi di modellazione numerica, che potrebbero essere computazionalmente costosi, o metodi analitici, che potrebbero essere difficili da ottenere per nuovi dispositivi, o modelli empirici, la cui portata e accuratezza potrebbero essere limitati.

In confronto ad altri strumenti tradizionali, le reti neurali presentano diversi vantaggi [4]:

- *Scalabilità*: per ottenere un certo livello di accuratezza, i modelli classici necessitano una grande quantità di dati e la complessità spaziale cresce esponenzialmente con il crescere della dimensione. Al contrario, il modello di rete neurale è leggero e scalabile.
- *Generalizzazione*: gli strumenti tradizionali mostrano scarse performance quando si tratta di generalizzazione, in quanto spesso presentano errori. Le reti neurali, invece, utilizzando i giusti algoritmi di apprendimento, possono essere ben generalizzate, il che le rende più robuste contro i rumori.
- *Fluidità*: un modello ideale deve necessariamente essere infinitamente differenziabile. I vecchi metodi potrebbero esserlo, ma a discapito della loro efficienza computazionale. Le reti neurali, al contrario, è garantito che lo siano.

La struttura di una rete neurale influenza fattori come la quantità di dati e tempo richiesta per l'addestramento, e l'accuratezza che si vuole raggiungere. Una struttura appropriata aiuta ad ottenere una maggiore accuratezza del modello con una quantità minore di dati necessari per l'addestramento, in un minore periodo di tempo.

Le tipologie di reti impiegate possono essere diverse [5]:

1. *Reti feed-forward:*

- Multilayer Perceptron (MLP): tipicamente, questo tipo di rete consiste in un livello di input e uno di output, combinato con uno o più livelli nascosti. Comunemente utilizza come funzioni di attivazione la sigmoide, l'arcotangente e la tangente iperbolica.
- Radial Basis Function (RBF): si tratta di una rete neurale a singolo livello, che utilizza la funzione radiale di base come funzione di attivazione.

2. *Reti con conoscenza a priori:*

Dato che nelle reti feed-forward l'intera mole di informazioni sul dispositivo proviene dal training set, è necessario che quest'ultimo sia di grandi dimensioni per assicurare che il modello sia accurato. Ciò le rende molto costose. Le reti neurali con conoscenza a priori risolvono questo problema.

3. *Reti gerarchiche:*

L'approccio gerarchico riduce il costo dello sviluppo delle librerie attraverso una riduzione della collezione di dati necessari e un minore tempo di addestramento.

4. *Altre tipologie di reti:*

Esistono numerose tipologie di reti neurali, ognuna con una specifica funzione che si può adattare al meglio ad uno specifico problema. Ad esempio, utilizzando le self-organizing maps (SOM) si possono raggruppare in cluster dei campioni con comportamento simile, con conseguente miglioramento di accuratezza e tempistiche.

Per ottenere una rete neurale che si presti alle esigenze della modellistica, è importante costruirla passo per passo nella maniera corretta [6]:

1. *Formulazione del problema ed elaborazione dei dati:*

Per prima cosa si identificano gli input x e gli output y , basandosi sullo scopo della rete neurale. In seguito si definisce il range di dati da utilizzare. Utilizzando un software per la simulazione, si generano le coppie x - y . Queste vengono poi suddivise in tre set separati: training set, validation set e test set.

2. *Addestramento:*

I pesi della rete vengono inizializzati con dei valori casuali compresi in un intervallo dato. Si formula poi il processo di addestramento, in particolare occorre definire l'errore della rete neurale. Vengono calcolate anche le derivate dell'errore. In seguito si controllano e si risolvono eventuali fenomeni di over-learning o under-learning. Come ultima cosa si valuta la qualità dei risultati.

3.3 – Machine learning e MOSFET

Durante l'ultimo decennio, l'impiego di reti neurali artificiali (ANN) ha giocato un ruolo di rilievo sempre maggiore nella modellistica di dispositivi non lineari, compresi i transistor. Questo approccio ha permesso una simulazione molto più accurata sia nelle regioni attive che in quelle passive rispetto a quanto era possibile fare in precedenza.

I modelli di transistor compatti sono diventati molto più complicati nel corso degli anni. È diventato fondamentale studiare la dettagliata dipendenza dalla temperatura ambientale e gli effetti dinamici del auto-riscaldamento sulle caratteristiche I-V e Q-V, dal momento che la dimensione e la potenza dei transistor raggiungono livelli critici con l'incremento di quest'ultimo [7]. In termini di reti neurali artificiali, ciò significa che le relazioni costitutive sono addestrate su un'altra variabile, la temperatura, aggiungendo un'ulteriore dimensione. È sia più facile che più preciso aggiungere una nuova variabile alla funzione di una rete neurale artificiale che implementare, a mano, una relazione termica esplicita in una funzione che descrive un modello elettrico complicato e non lineare.

Un esempio di modelli di transistor generati con le reti neurali è quello del MOSFET. Un modello di rete neurale che riproduce la caratteristica a corrente continua di tale dispositivo viene sviluppato basandosi su dati fisici ottenuti utilizzando algoritmi di generazione automatica.

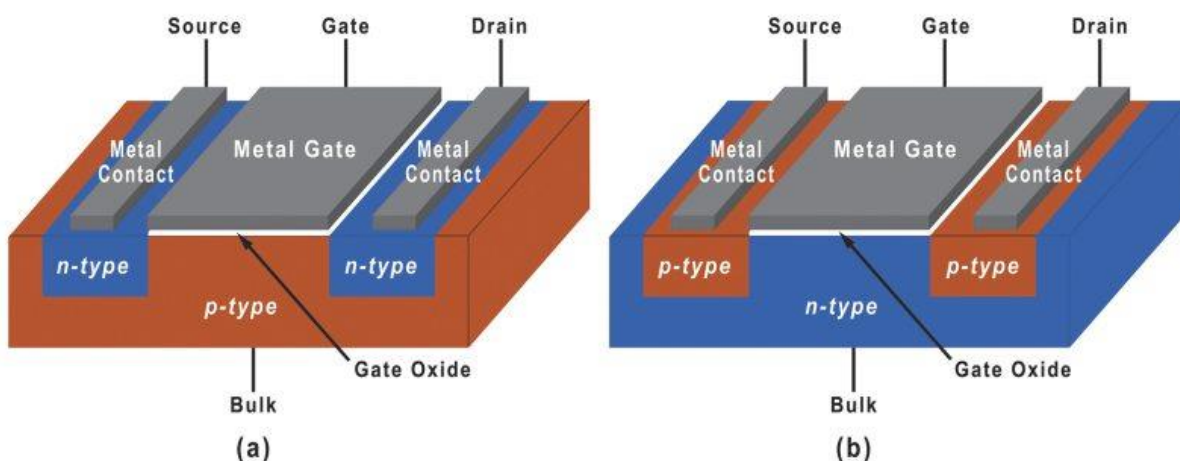


Figura 4 - Schema della struttura di un transistor NMOS (a) e un transistor PMOS (b)

Il MOSFET (acronimo di metal-oxide-semiconductor field-effect transistor) indica una tipologia di transistor a effetto di campo largamente usata nel campo dell'elettronica digitale. È composto da un condensatore Metallo Ossido Semiconduttore, che controlla la formazione di un "canale" conduttivo nel semiconduttore, tra i contatti di drain e source. Il contatto

metallico della struttura MOS prende il nome di gate. L'applicazione di una tensione al gate permette di controllare il passaggio di cariche tra il source e il drain, e quindi la corrente elettrica che attraversa il dispositivo.

Il modello neurale che emula il transistor MOS ha due input, la tensione di drain e la tensione di gate. La corrente di drain è il parametro di output della rete. I dati di addestramento e test vengono generati utilizzando un simulatore. È stato dimostrato [6] che gli errori durante la fase di test sono inferiori allo 0.50%. Questo modello veloce di rete neurale del MOSFET può, quindi, essere utilizzato per prevedere la caratteristica a corrente continua del dispositivo con l'accuratezza di un simulatore fisico.

Capitolo 4

Il Remaining Useful Lifetime

4.1 – Definizione

Una delle principali aree dell'elettronica in cui trovano applicazione le tecniche basate sull'intelligenza artificiale è legata allo studio della degradazione dei dispositivi. Per molte applicazioni, ai dispositivi elettronici è richiesta una lunga durata, con tempi di vita di decine di anni. È chiaro che, non è possibile raccogliere dal campo, archiviare e analizzare un set di dati su un intervallo di tempo così ampio per studiare le proprietà di degrado o la durata dei dispositivi. Ciò è ottenibile attraverso test accelerati, in cui il dispositivo viene sottoposto a diversi tipi di sollecitazioni (termiche o elettriche) per accelerare la degradazione del dispositivo [8]. In questi test accelerati, viene prima identificato un precursore del guasto, e le misurazioni effettuate dai sensori durante il test sono poi analizzate utilizzando metodi basati su modelli e/o dati per prevedere parametri, come il Remaining Useful Lifetime (RUL), dei dispositivi di alimentazione. Nel seguito si farà in particolare riferimento ai dispositivi di

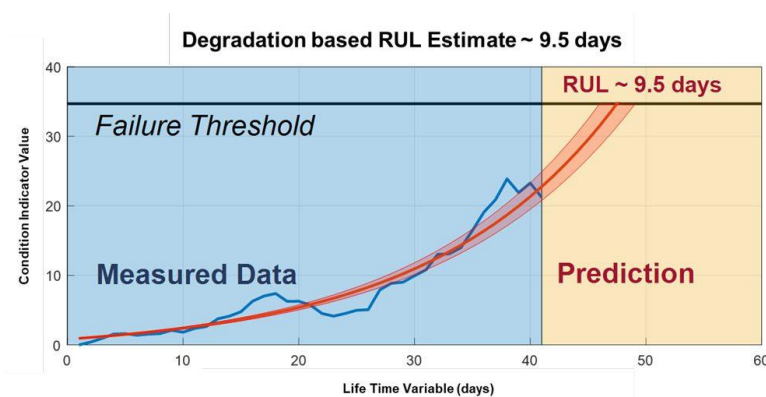


Figura 5 - Esempio di stima del RUL

alimentazione e di conversione dell'energia elettrica.

Il remaining useful lifetime è un parametro che misura il periodo di tempo in cui è probabile che una macchina funzioni prima che richieda riparazione o sia da sostituire. Tenendo conto del

RUL, gli ingegneri possono programmare la manutenzione, ottimizzare l'efficienza operativa ed evitare tempi di fermo non pianificati. Per questo motivo, la sua stima è una priorità assoluta nei programmi di manutenzione predittiva.

I test di invecchiamento accelerato forniscono una panoramica dei parametri che fungono da precursori di degrado o guasto. Tuttavia, nelle applicazioni del mondo reale, le sollecitazioni che agiscono sul dispositivo non sono controllate (come nei test), senza contare che c'è la possibilità che più sollecitazioni agiscano contemporaneamente. La maggior parte degli studi condotti finora considerano un test accelerato specifico e selezionano un precursore per stimare il RUL, mentre nella pratica le applicazioni indicano che diverse caratteristiche devono essere utilizzate come precursori. Inoltre, quando si studiano diversi MOSFET in parallelo, la quantità di corrente che lasciamo passare attraverso ciascun MOSFET ha un effetto diverso in termini di stress elettrico e termico che agiscono sul dispositivo. Per tali scenari, i metodi per stimare il RUL esistenti potrebbero non essere consistenti a causa della complessa dinamica dello squilibrio di corrente e temperatura.

Per questo motivo, con il continuo e rapido sviluppo delle reti neurali, si è posta molta attenzione ad eventuali loro applicazioni per ottimizzare il processo di stima del remaining useful lifetime.

4.2 – Stima con i filtri di Kalman

Il ciclo termico è una delle principali tecniche per accelerare l'avanzamento del guasto relativo ai dispositivi. In diverse sperimentazioni la variazione della resistenza allo stato ON è stata identificata come il precursore del guasto, per questo si è sviluppato un modello di degradazione con andamento esponenziale che si adatta con successo a tali dati sperimentali.

In passato sono stati utilizzati diversi algoritmi prognostici basati su modelli e dati, utilizzando la resistenza allo stato ON come precursore principale del fallimento. Serkan Dusmez assieme ad altri colleghi [9], ha proposto un metodo basato sul filtro di Kalman per prevedere il remaining useful lifetime di un dispositivo MOSFET, in cui i dati sperimentali, comprendenti anche rumore di misurazione e incertezza, sono stati filtrati con un filtro di Kalman e successivamente un classico algoritmo dei minimi quadrati è stato utilizzato per la previsione. Tuttavia, l'errore di previsione è stato stimato pari al 28,7%. Gli autori hanno quindi esteso il loro lavoro per ridurre l'errore di predizione utilizzando un algoritmo di

consenso del campione casuale (RANSAC) ottimizzando i parametri di sistema attraverso l'utilizzo di algoritmi genetici.

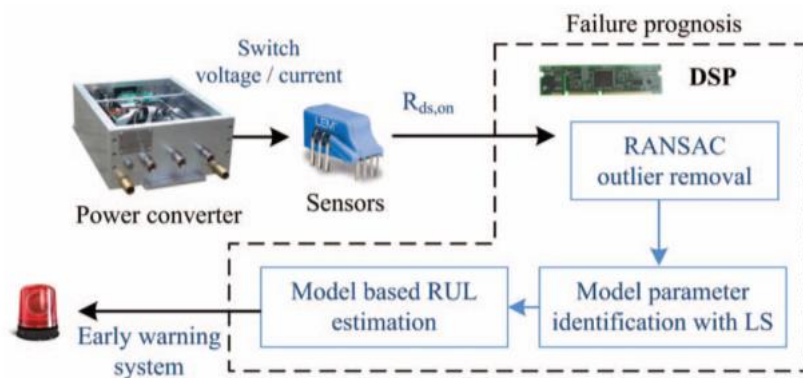


Figura 6 - Schema a blocchi per la stima del RUL

Il filtro di Kalman è un efficiente filtro ricorsivo che valuta lo stato di un sistema dinamico a partire da una serie di misure soggette a rumore. Per le sue caratteristiche

intrinseche è un filtro ottimo per rumori e disturbi agenti su sistemi gaussiani a media nulla.

RANSAC (RANdom SAMple Consensus), è un metodo iterativo per la stima dei parametri di un modello matematico a partire da un insieme di dati contenenti outliers. È un algoritmo non deterministico nel senso che produce un risultato corretto solo con una data probabilità, che aumenta al crescere delle iterazioni consentite. Il concetto di base su cui si fonda questo algoritmo, è che i dati su cui lavora siano divisi tra inlier, cioè dati la cui distribuzione può essere caratterizzata dall'insieme di parametri di un modello, e outlier, cioè dati che non sono rappresentati da tale modello. Inoltre, i dati possono essere affetti da rumore. RANSAC assume inoltre che, dato un insieme (solitamente ridotto) di inlier, esiste una procedura che può stimare i parametri di un modello che rappresenta in modo ottimale i dati.

Nello studio effettuato da Dusmez, i MOSFET sono esposti a cicli termici, e la resistenza allo stato on del precursore di guasto in forma logaritmica viene approssimata da una funzione lineare. I valori anomali presenti nei dati sperimentali vengono rimossi dall'algoritmo RANSAC. I coefficienti empirici del modello lineare si trovano minimizzando la funzione di costo, cioè applicando il metodo dei minimi quadrati solo agli inliers determinati dall'algoritmo RANSAC. Questo approccio ha mostrato risultati migliori e una più rapida convergenza al RUL effettivo rispetto al caso in cui il metodo dei minimi quadrati viene applicato a tutti i punti dati senza rimuovere i valori anomali. I risultati suggeriscono che il metodo di stima RUL proposto può essere integrato in controller a basso costo per la prognosi dei guasti in tempo reale, il che aumenterebbe significativamente l'affidabilità dei convertitori di potenza. Tuttavia, la maggior parte dei valori anomali viene rimossa quando RUL è intorno al 40%. Inoltre, non è possibile tenere traccia della variazione non lineare in alcuni campioni.

Assieme ad altri ricercatori, José Celaya [10] ha esplorato l'uso del filtro di Kalman esteso e del metodo del filtro particolato per la stima del RUL, e lo ha confrontato con un metodo basato sui dati utilizzando la regressione del processo gaussiano (GPR).

Il filtro di Kalman esteso è una versione non-lineare del filtro di Kalman usata quando l'evoluzione o l'osservazione dello stato del sistema sono non-lineari.

Sebbene i metodi basati su modelli fornissero risultati validi, richiedevano solide ipotesi per una migliore accuratezza delle previsioni, il che potrebbe rappresentare una sfida per la prognosi online. Rispetto al metodo basato su modelli, il metodo GPR è stato in grado di fornire le previsioni, ma solo considerevolmente in ritardo, quando il processo di degradazione tende ad essere più vicino al punto di gomito della caratteristica esponenziale. È stato osservato che le previsioni durante i primi cicli di degradazione divergono dal valore reale.

Gli esiti di queste sperimentazioni nel calcolo del remaining useful lifetime sono buoni, ma comunque non ottimi. Questo è ciò che ha motivato molti ricercatori nell'approfondire lo studio sulle applicazioni del machine learning e delle reti neurali. L'obiettivo finale infatti è quello di ottimizzare questa stima, in modo da apportare un considerevole miglioramento alle tecniche finora note.

Capitolo 5

Stima del R.U.L. con reti neurali artificiali

5.1 – Implementazione della rete neurale

La forma più semplice di rete neurale artificiale è la rete neurale feedforward (FFNN), nella quale il flusso di informazioni è unidirezionale dal livello di input al livello di output. La struttura di base della FFNN più semplice comprende un livello di input, un livello nascosto e un livello di output. I parametri di interconnessione tra i diversi strati sono i pesi, che quantificano il contributo di ciascun dato in input calcolando la somma di tutti ingressi. Il bias invece è un valore di soglia che viene sommato dopo la valutazione della somma dei pesi dal livello precedente. La relazione tra i diversi strati è costituita dalle funzioni di attivazione.

Il processo di addestramento di una ANN prevede la ricerca dei parametri, dei pesi e delle distorsioni ottimali che descrivono meglio la relazione tra l'input e l'output. Il modello di rete viene funzionalizzato utilizzando le funzioni di trasferimento dopo l'addestramento. L'FFNN viene addestrato utilizzando il metodo di backpropagation, nel quale l'errore si propaga tra i dati di addestramento e l'output di rete all'indietro. Questo può anche essere visto come un processo di ottimizzazione per trovare i pesi e le distorsioni migliori con lo scopo di ridurre al minimo l'errore quadratico medio tra i valori previsti e il set di dati utilizzato per l'addestramento.

Nella ricerca svolta da Karkulali Pugalenti [11], assieme ad altri collaboratori, ci sono tre variabili di input fornite alla FFNN, vale a dire il training set, il validation set (che utilizza il 30% dei dati dal training set per la convalida), e il set di dati utilizzato per la previsione. La rete neurale viene addestrata utilizzando l'algoritmo Levenberg Marquardt per una

convergenza più rapida. I livelli nascosti e quelli di output vengono elaborati utilizzando come funzione di attivazione la sigmoide, nella sua variante tangente iperbolica:

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{2x}}$$

Il livello nascosto può essere espresso in funzione di *tansig*, cioè la funzione di trasferimento tangente iperbolica:

$$H_j = \text{tansig}\left(\sum_{i=1}^3 \Delta Rds, on_i w_{i,j}^H + b_j^H\right)$$

$$H_j = \sum_{i=1}^3 \frac{1 - e^{-2((\Delta Rds, on)_i w_{i,j}^H + b_j^H)}}{1 + e^{-2((\Delta Rds, on)_i w_{i,j}^H + b_j^H)}}$$

dove H_j si riferisce al valore del j -esimo neurone nel livello nascosto, $\Delta Rds, on$ è l'incremento misurato nella resistenza allo stato ON, $w_{i,j}^H$ è il peso corrispondente al collegamento tra l' i -esimo input e il j -esimo neurone nascosto e b_j^H è il valore di bias per il j -esimo neurone nascosto, dove j varia da 1 a n .

Allo stesso modo, l'output di FFNN, che è l'incremento futuro previsto della resistenza, può essere espresso come:

$$(\Delta Rds, on)_{\text{Predicted}} = \text{tansig}\left(\sum_{j=1}^n H_j w_{j,k}^O + b_k^O\right), \quad k = 1$$

$$(\Delta Rds, on)_{\text{Predicted}} = \sum_{j=1}^n \frac{1 - e^{-2(H_j w_{j,k}^O + b_k^O)}}{1 + e^{-2(H_j w_{j,k}^O + b_k^O)}}$$

dove $w_{j,k}^O$ è il peso del j -esimo neurone nascosto e del k -esimo neurone di output, b_k^O è il valore di polarizzazione del k -esimo neurone nel livello di output, n è il numero di neuroni nascosti e H_j è ottenuto dall'equazione precedente. Poiché il valore previsto è l'unico segnale di uscita dalla FFNN, il valore di k è uguale a 1. I valori dei pesi e le distorsioni vengono determinate dopo aver addestrato la rete neurale.

Ora possiamo rappresentare l'output con una nuova equazione:

$$(\Delta Rds, on)_{\text{Predicted}} = f(W_O^T * f(W_H^T * (\Delta Rds, on) + B_H) + B_O)$$

E, infine, il RUL risultante del MOSFET può essere stimato come:

$$(N_{cy})_{RUL} = N_{cy_{EOL}} - N_0$$

dove $W_0^T = (W_{1,1}^H, \dots, W_{j,k}^H, \dots, W_{j,1}^H)^T$ è il peso della matrice dei neuroni nel livello di output e $W_H^T = (w_{1,1}^o, \dots, w_{i,j}^o, \dots, w_{j,1}^o)^T$ è la matrice dei pesi del livello nascosto. $\Delta R_{ds, on}$ si riferisce all'incremento misurato nella resistenza allo stato ON e $B_H = (b_1^H, \dots, b_j^H)^T$ è l'insieme dei valori di bias dei neuroni nel livello nascosto e $B_O = b_k^O$ è il bias del neurone nel livello di output. Nella seconda equazione, N_0 è il ciclo di temperatura al quale viene effettuata la previsione e $(N_{cy})_{EOL}$ sono i cicli previsti alla fine del ciclo di vita corrispondenti a un criterio di soglia di guasto per il dispositivo/sistema in prova. La differenza tra $(N_{cy})_{EOL}$ e N_0 fornisce il RUL effettivo in termini di numero di cicli, che è l'unità di misura nel caso di studio del MOSFET.

5.2 – Stima del remaining useful lifetime

I test di invecchiamento allo scopo di sviluppare modelli di previsione della durata possono richiedere molto tempo e costi elevati. Pertanto, i test di invecchiamento accelerato sono un metodo efficace per raccogliere dati sul degrado in un breve lasso di tempo. Dusmez [9], congiuntamente con altri ricercatori, ha sviluppato un banco di prova di invecchiamento accelerato con il ciclo termico dei MOSFET. Gli studiosi hanno eseguito una serie di esperimenti in cui una serie di MOSFET era soggetta a degradazione mediante cicli termici con un'ampiezza di oscillazione di 80 °C e una temperatura massima di giunzione di 100 °C. Gli esperimenti sono stati eseguiti su cinque dispositivi per circa tre settimane. È stato osservato che la tendenza al degrado della resistenza allo stato ON stava seguendo un profilo esponenziale in una certa misura e ogni dispositivo aveva una tendenza leggermente diversa all'usura della resistenza nelle stesse condizioni di test. Gli autori nel loro lavoro hanno osservato che il fallimento più comune era la perdita del controllo del gate. Questi dati verranno utilizzati per la stima del RUL con una rete neurale artificiale. I dati MOSFET-1 sono utilizzati come set di dati di addestramento per FFNN mentre i dati MOSFET-2 sono utilizzati per la previsione. Ogni 500 cicli di dati vengono utilizzati nel training set e ogni 1000 cicli di dati vengono utilizzati nel set di dati di previsione. Si assume che il valore di soglia di guasto sia $\Delta R_{on} = 30m\Omega$, ovvero un aumento del 12% rispetto al valore di resistenza iniziale.

La selezione di un numero appropriato di neuroni durante la modellazione della rete neurale è essenziale per prevenire l'overfitting e anche per ridurre il carico computazionale. Sono stati proposti diversi approcci costruttivi e di pruning per correggere il numero di neuroni nascosti per un particolare problema. In questo studio, Karkulali Pugalenti e gli altri ricercatori hanno adottato il metodo proposto da Hagan per determinare il limite superiore sui neuroni nascosti usando l'equazione:

$$N_h = \frac{N_s}{\alpha * (N_i + N_o)}$$

dove N_i è il numero di neuroni di input, N_s è il numero di campioni nel training set, N_o è il numero di neuroni di output e α è un fattore di scala arbitrario che di solito varia da 2 a 10. In questo studio, N_i e N_o sono uguali a 1 e ci sono 47 punti dati nel training set. Quindi, il limite superiore per il numero di neuroni nascosti trovato è 11.75. La FFNN è stata allenata per 500 ripetizioni e l'accuratezza relativa, insieme al tempo di calcolo, è stata utilizzata come metrica delle prestazioni per il confronto. L'accuratezza relativa (RA) può essere espressa come misura dell'errore relativo tra il RUL previsto e quello effettivo, infatti la sua espressione è data da:

$$RA = 1 - \frac{|True\ RUL - Predicted\ RUL|}{True\ RUL}$$

Al fine di migliorare la qualità della previsione, sono stati incorporate due regolazioni della previsione per eliminare i valori anomali. Innanzitutto, i valori della previsione che superano 2σ vengono eliminati dai risultati. In secondo luogo, vengono selezionati solo i campioni che aumentano/diminuiscono in modo monotono nei risultati della previsione. Sulla base delle suddette regolazioni, i risultati indicano che per una FFNN con un singolo neurone nello strato nascosto, anche se il carico computazionale è inferiore, solo 24 delle 500 ripetizioni rientrano nell'intervallo di confidenza accettabile, ottenendo così una percentuale di successo di circa il 4,8%. Aumentare il numero di neuroni nascosti a 5 o 10 migliora l'accuratezza con un tasso di successo della previsione di circa il 22%. Con un numero ancora maggiore di neuroni nascosti, si osserva un'accuratezza di previsione molto scarsa. Pertanto, si può concludere che la scelta di 5 neuroni nascosti per questo particolare scenario è ottimale. Anche se il set di dati per la previsione ha una grande quantità di rumore, l'accuratezza della previsione è considerevolmente buona, il che implica che l'impatto del rumore sul sistema di rete neurale è trascurabile.

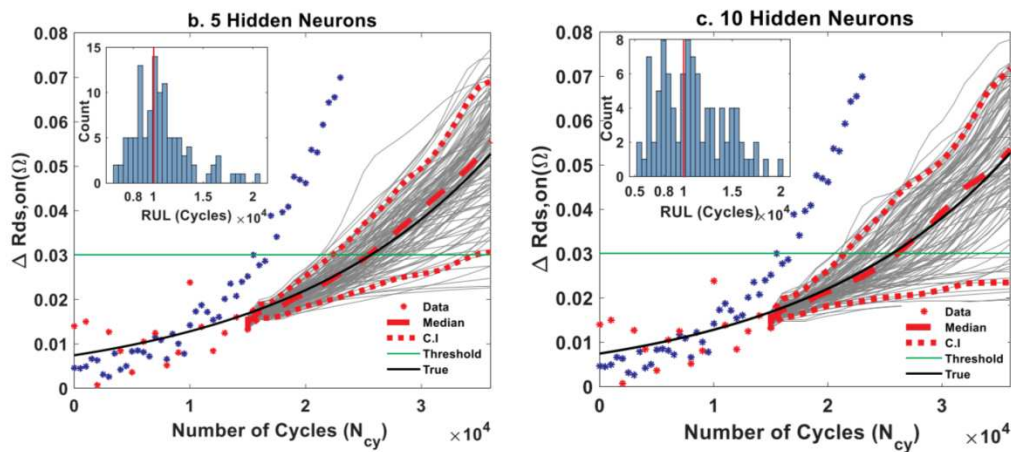


Figura 7 - Confronto stima RUL con 5 neuroni nascosti (sinistra) e 10 neuroni nascosti (destra)

Per un modello di rete neurale efficace, il training set dovrebbe coprire tutte le variazioni statistiche nei dati con il rumore in essi incluso. In passato sono stati eseguiti diversi studi per determinare l'efficienza del modello di rete neurale con insiemi di dati di addestramento molto piccoli, ma l'impatto di insiemi di dati di previsione molto piccoli sull'accuratezza della previsione è relativamente inesplorato. Oltre alla ricerca relativa all'applicazione delle reti neurali artificiali per la stima del RUL, Karkulali Pugalenti e altri hanno cercato di colmare tale lacuna prevedendo i risultati con solo quattro punti nel set di dati per la previsione. I risultati mostrano che i valori previsti non divergono dal valore reale anche se l'accuratezza relativa per la previsione è bassa. Il tasso di successo della previsione è inferiore al 4%, che aumenta al 16% con 8 punti dati. Pertanto, con un buon training set, il modello di rete neurale proposto può prevedere efficacemente i risultati anche con pochissimi punti nel validation set, senza imbattersi in fenomeni di underfitting.

5.3 – Discussione dei risultati

In questo studio è stato presentato l'approccio basato su una rete neurale artificiale per la prognosi della degradazione della resistenza ON nei MOSFET. Viene considerata una rete neurale feedforward e tre diversi set di dati provenienti da esperimenti di invecchiamento accelerato, i quali vengono utilizzati per l'addestramento e la previsione.

L'efficienza del modello può essere migliorata scegliendo un numero ottimale di neuroni nascosti. Le prestazioni sono state valutate utilizzando l'accuratezza relativa e il tempo di calcolo come parametri prognostici chiave. I risultati mostrano chiaramente che con pochi

neuroni nascosti, il tasso di successo della previsione è molto basso anche se il carico computazionale è inferiore. Tuttavia, per 5-10 neuroni, l'accuratezza migliora un po'.

Le prestazioni della rete peggiorano ancora una volta per un maggior numero di neuroni nascosti. I risultati indicano inoltre che la FFNN ha un vantaggio rispetto ai metodi basati su modelli per la previsione di dati rumorosi e in piccola quantità. Le previsioni sono abbastanza buone anche sui set di dati che non seguono la stessa tendenza alla degradazione del training set.

Come si può notare comparando questi risultati con quelli trattati nel capitolo precedente, la stima del RUL con il metodo delle reti neurali è decisamente più conveniente. Oltre che in termini di tempo, infatti, la loro applicazione in questo tipo di procedura dimostra di avere percentuali di errore molto inferiori a quelle calcolate con il filtro di Kalman. Costruendo una rete neurale artificiale con un adeguato numero di neuroni nel livello nascosto, e fornendogli in input un dataset di dimensioni corrette si possono quindi migliorare sia le prestazioni che l'accuratezza.

Sicuramente questa tecnica può essere migliorata, ad esempio utilizzando approcci guidati per mettere a punto ulteriormente la struttura e la complessità della rete neurale al fine di migliorare ulteriormente l'accuratezza per set di dati con tendenze di degrado molto diverse, garantendo allo stesso tempo che il carico computazionale sia mantenuto basso.

In particolare, è stato dimostrato che anche solo cambiando tipologia di rete neurale è possibile ottenere prestazioni ancora migliori.

5.4 – Che tipologia di rete neurale utilizzare?

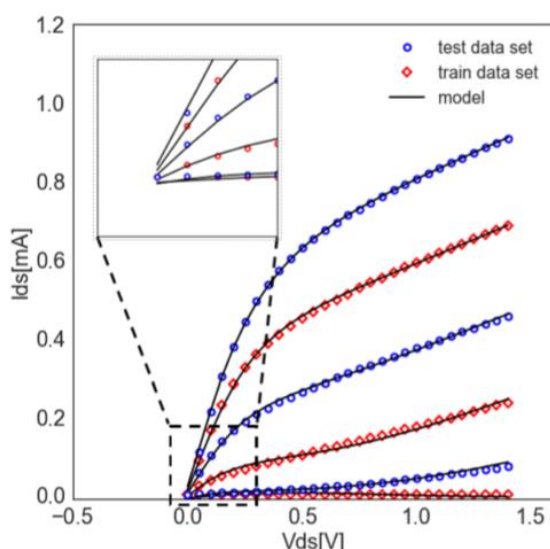


Figura 8 - Caratteristica I_{ds} - V_{ds} per V_{gs} differenti, dove l'ingrandimento mostra il comportamento non fisico nell'intorno di $V_{ds}=0$

In un modello di rete neurale artificiale per transistor, i dati misurati delle tensioni ai terminali e delle geometrie dei dispositivi vengono solitamente utilizzati come elementi in input nel livello di ingresso, mentre quello della corrente al terminale viene utilizzato come uscita. Tuttavia, se

un modello ANN viene sviluppato solo sulla base dei dati senza considerare la fisica nei dispositivi, potrebbe mostrare comportamenti non fisici e causare errori nella simulazione del circuito. Ad esempio, un modello basato su ANN può prevedere un I_{ds} finito quando $V_{ds} = 0$ per la curva $I_{ds} - V_{ds}$, violando la legge fisica di base.

Per eliminare questo comportamento non fisico efficacemente, un nuovo approccio viene illustrato in [12]. Questo infatti utilizza la transconduttanza come parametro per il fitting. Utilizzare come funzione di costo il logaritmo della transconduttanza può infatti essere d'aiuto nel migliorare l'accuratezza del modello, in quanto riduce la scala dei dati.

Per soddisfare i requisiti effettivi della tecnologia e del design dei circuiti integrati, devono essere incluse caratteristiche di input aggiuntive come la temperatura e le dimensioni del dispositivo. Pertanto sarebbe necessario un modello ANN complesso con dimensioni del livello nascosto più grandi per garantire elevata qualità al modello. Spesso viene utilizzata la regolarizzazione per prevenire l'overfitting, che spesso si verifica quando il modello è eccessivamente complesso. Infatti, introdurre più funzioni nella funzione obiettivo introduce un termine di costo maggiore.

Questo problema viene risolto costruendo una rete neurale profonda per costruire il modello del transistor. Avendo quindi a disposizione una rete con decine di livelli nascosti e utilizzando l'equazione logistica come funzione di attivazione, si può ottenere maggiore qualità e accuratezza nei risultati. I comportamenti non fisici e l'overfitting non si manifestano in questo nuovo modello. Inoltre, esso può essere facilmente esteso con diverse features in ingresso per rendere la simulazione più dettagliata e prossima alla realtà.

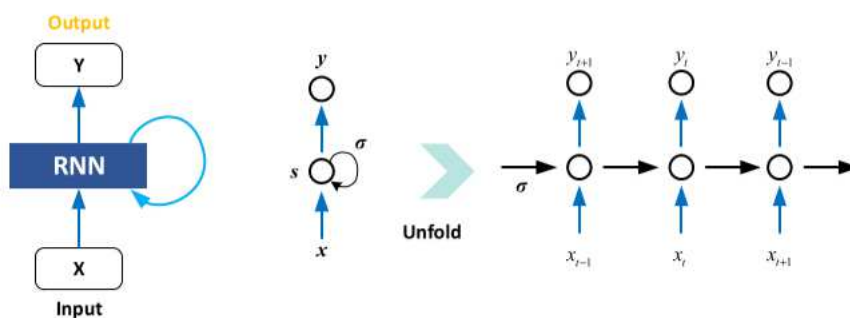


Figura 9 - Struttura di una RNN

Un'altra tipologia di rete molto utile in questo campo è la rete neurale ricorrente (RNN), particolarmente

utile in quanto ha la capacità di memorizzare i dati, dato che i suoi livelli possono comunicare anche all'indietro con i livelli precedenti. Questa è in grado di modellare dati sequenziali, che vengono studiati per la previsione RUL con features statistiche dei segnali di vibrazione nel dominio del tempo e della frequenza.

Lo svantaggio dell'utilizzo di queste features però è che portano ad ignorare delle informazioni sulla frequenza temporale, la quale è fondamentale nella previsione RUL perché i segnali di vibrazione non sono stazionari quando si verifica il guasto. Per risolvere questo problema, in [13] viene proposta un'ulteriore architettura profonda, denominata rete neurale convoluzionale ricorrente profonda (DRCNN). Incorporando il funzionamento convoluzionale nel processo di transizione di stato di una RNN, le informazioni spaziali nel dominio tempo-frequenza possono essere apprese automaticamente dai segnali di vibrazione, il che contribuisce al miglioramento delle prestazioni di previsione. Con l'operazione convoluzionale in RNN, sia le informazioni spaziali nel dominio della frequenza temporale che le informazioni precedenti vengono utilizzate per la previsione RUL. Inoltre, impilando la rete neurale convoluzionale ricorrente strato per strato, l'architettura profonda può apprendere funzionalità di alto livello nel dominio della frequenza temporale. Rispetto ai risultati del metodo RNN profondo convenzionale, il metodo proposto mostra la sua efficacia e superiorità. In questo studio, vengono comparati poi gli errori compiuti da una rete profonda ricorrente e da una DRCNN, dimostrando che quest'ultima tipologia diminuisce considerevolmente il tasso di errore.

Questi due esempi chiariscono il fatto che le reti neurali artificiali migliorano notevolmente le prestazioni nel processo di stima del remaining useful lifetime, ma esiste comunque un margine di miglioramento. Dato il gran numero di tipologie di reti esistenti, che si distinguono tra di loro anche per dettagli implementativi come il numero di neuroni o la funzione di attivazione, sicuramente per ogni specifico problema da risolvere esisterà la rete neurale più adatta a quel caso. Il concetto fondamentale quindi si riduce al conoscere, attraverso sperimentazioni come quelle appena viste, le possibilità di questo strumento in continuo sviluppo, per poterlo calibrare al meglio secondo le esigenze.

Capitolo 6

Conclusioni

Con il progresso della tecnologia, si sono moltiplicati anche gli strumenti e le tecniche per eseguire tantissimi processi. In particolare, la creazione e lo sviluppo dell'intelligenza artificiale offre diversi vantaggi, tra cui il risparmio di tempo e risorse. L'apprendimento automatico, con le sue reti neurali, sta prendendo piede molto velocemente e sembra avere le potenzialità di cambiare completamente il lavoro dell'essere umano.

Tra i diversi campi in cui il machine learning trova impiego, sicuramente la sua applicazione alla modellistica permette di ottenere notevoli benefici. Come è stato evidenziato nei precedenti capitoli, infatti, non ci si limita ad un miglioramento qualitativo dei processi di costruzione e simulazione, ma si riesce ad ottenere addirittura un miglioramento delle prestazioni significativo.

Le reti neurali artificiali ora sembrano essere diventate una tecnica molto utilizzata per modellare componenti dinamici non-lineari, come i transistor. Su questo sicuramente influisce il fatto che la rete, una volta costruita, è anche semplice da rimodellare e adattare a seconda delle specifiche necessità. Rimangono ancora alcune incertezze, date dal fatto che questo metodo sembrerebbe non rispecchiare perfettamente i dispositivi reali. Un esempio ne è infatti il comportamento non fisico del transistor MOSFET implementato con una ANN. La soluzione potrebbe essere, come visto, l'utilizzo di una tipologia di rete neurale differente, più adatta al caso specifico che si sta trattando, ma a volte questo potrebbe non bastare.

Il caso della stima del remaining useful lifetime è solo una delle infinite applicazioni che l'apprendimento automatico può trovare nell'ambito dell'elettronica, ma si è già dimostrato essere un metodo rivoluzionario. Il progresso di questa disciplina subirebbe un'accelerazione

senza paragoni avendo a disposizione degli strumenti come questo ma nella loro versione ottimizzata.

Per questo motivo il progresso fatto finora è decisamente notevole, ma il margine di miglioramento è ancora molto ampio. In pochissimi anni il mondo dell'intelligenza artificiale si è ampliato in modo impressionante, quindi certamente si arriverà anche ad un punto dello sviluppo in cui il perfezionamento di queste tecniche diventerà realtà. Le tecniche utilizzate finora diverrebbero obsolete, avendo come concorrenti le reti neurali che, una volta affinata anche la loro accuratezza, sarebbero sicuramente l'opzione migliore.

Bibliografia

- [1] – Taiwo Oladipupo Ayodele (2010), *Machine learning overview, New advances in machine learning*, Yagang Zhang (Ed.)
- [2] - Baştanlar Yalin, Ozuysal Mustafa (2014), *Introduction to machine learning*, Methods Mol. Biol.
- [3] - Taiwo Oladipupo Ayodele (2010), *Types of machine learning algorithms, New advances in machine learning*, Yagang Zhang (Ed.)
- [4] – Mingda Li, Ozan Irsoy, Claire Cardie, Huili Grace Xing (2017), *Physics-inspired neural networks for efficient device compact modeling*, School of electrical and computer engineering, Cornell University, Ithaca, NY
- [5] - Q.J. Zhaog, P. Wang, Viaya K. Devabhaktuni (1999), *Neural network structures for EM/Microwave modeling*, Department of electronics, Carleton University, Ottawa, Canada
- [6] - Qi-Jun Zhang, Kuldip C. Gupta, Vijay K. Devabhaktuni (2003), *Artificial neural networks for RF and microwave design*, Department of electronics, Carleton University, Ottawa, Canada
- [7] – David E. Root (2012), *Future device modeling trends*, IEEE Microwave Magazine 13, pagine 45-59
- [8] – Jayanth Reddy Regatti, Alec Schnabel, Haoyang You, Boxue Hu, Han Wang, Julia Zhang, Jon Hall, Jin Wang, Abhishek Gupta (2021), *A discussion of artificial intelligence applications in SiC MOSFET device operation*, Center for high performance power electronics, The Ohio State University, Columbus, OH, USA

- [9] - Serkan Dusmez, Mehrdad Heydarzadeh, Mehrdad Nourani, Bilal Akin (2016), *A robust remaining useful lifetime estimation method for discrete power MOSFETs*, Department of electrical engineering, University of Texas-Dallas, Dallas, Texas
- [10] - José R. Celaya, Abhinav Saxena, Chetan S. Kulkarni, Sankalita Saha, Kai Goebel (2011), *Prognostics approach for power MOSFET under thermal-stress aging*, NASA Ames Research Center, Prognostics Center of Excellence, CA, USA
- [11] - Karkulali Pugalenthia, Hyunseok Parkb, Nagarajan Raghavana (2019), *Prognosis of power MOSFET resistance degradation trend using artificial neural network approach*, Microelectronics Reliability 100–101 113467
- [12] - Yuan Lei, Xiao Huo, Beiping Yan (2018), *Deep neural network for device modeling*, IEEE Electron Devices Technology and Manufacturing Conference Proceedings of Technical Papers
- [13] – Meng Ma, Zhu Mao (2019), *Deep recurrent convolutional neural network for remaining useful life prediction*, IEEE International Conference on Prognostics and Health Management (ICPHM)