



SAPIENZA
UNIVERSITÀ DI ROMA

Anti-Adversarial deep learning per stimare i parametri dell'hamiltoniana di Tight-Binding

Facoltà di Scienze Matematiche, Fisiche e Naturali
Laurea Triennale in Fisica

Alberto Colella

Matricola 1814564

Relatore

Stefano Giagu

A handwritten signature in black ink, reading "Stefano Giagu", with a stylized flourish at the end.

Anno Accademico 2023/2024

Sommario

In questa dissertazione tratteremo di un metodo di machine learning per ricavare i parametri dell'hamiltoniana in approssimazione Tight-Binding sviluppato da Andrea Cacioppo nella sua tesi di laurea⁽²⁾. La metodologia implementata è semiempirica e permette di approssimare l'hamiltoniana di una molecola o di un solido cristallino attraverso un set di parametri che ne codificano le proprietà strutturali. Nel contesto ricaveremo l'hamiltoniana approssimata di un semplice solido cristallino. Nella dissertazione dopo aver illustrato gli elementi essenziali alla base del Deep Learning, e dell'approssimazione Tight-Binding per i solidi cristallini andremo ad illustrare la tecnica di addestramento "Anti-Adversarial" proposta nel lavoro di Andrea Cacioppo⁽²⁾. Verrà mostrato come con tale tecnica sia possibile addestrare un modello di rete neurale in grado di apprendere dalle misure delle bande energetiche le principali proprietà elettroniche, e come possono essere utilizzate per ricavare i parametri dell'hamiltoniana che meglio approssimano i dati sperimentali. Dopo aver ragionevolmente verificato il funzionamento del metodo, concluderemo facendo vedere come questo è applicabile al caso reale di un polimero di trans-poliacetilene

Indice

1	Introduzione	1
2	Deep Learning	2
2.1	Neurone artificiale	2
2.2	Reti Neurali Artificiali	3
2.3	Come una ANN può approssimare qualsiasi funzione	5
2.4	Addestramento di una rete neurale	6
3	Hamiltoniana di un solido cristallino e approssimazione Tight-Binding	8
3.1	Approssimazione di Born-Oppenheimer	9
3.2	Approssimazione LCAO	10
3.3	Simmetria nei solidi cristallini	10
3.4	Ricavare l'hamiltoniana Tight-Binding	11
4	Metodologia	14
4.1	H^{NN} nella rappresentazione della DNN	15
4.2	Generazione Dati	15
4.3	Training	16
4.4	Valutazione del modello	16
5	Risultati	17
6	Conclusioni	21
	Bibliografia	22

Capitolo 1

Introduzione

Riuscire a predire importanti proprietà strutturali dei materiali ha comportato sempre una notevole sfida, sia dal punto di vista della modellizzazione del problema che nell'approccio puramente sperimentale. L'importanza di comprendere qualitativamente il problema prima di affrontare la simulazione quantitativa è cruciale.

La complessità dei sistemi di elettroni e nuclei che interagiscono nei materiali richiede un approccio attentamente bilanciato tra modelli teorici e sperimentazione pratica. Il metodo semiempirico Tight-Binding infatti ci offre la possibilità di semplificare il problema e di adattarlo ai dati sperimentali, così da comprendere come elettroni e nuclei interagiscono all'interno di un solido cristallino o di una molecola.

L'uso di questi metodi semiempirici combinano approcci teorici e sperimentali, e permettono la correzione degli errori dovuti all'eccessiva semplificazione. Questi metodi possono includere tecniche di machine learning, simulazioni al computer avanzate o altre strategie innovative. L'approccio combinato permette di sfruttare al meglio sia la comprensione teorica che l'esperienza empirica, superando le limitazioni dei modelli puramente teorici o sperimentali.

Capitolo 2

Deep Learning

Il deep learning è una branca dell'apprendimento automatico (machine learning) che si basa su reti neurali artificiali profonde per elaborare informazioni e apprendere da esse. Questo approccio mira a simulare il modo in cui il cervello umano elabora e apprende da dati, consentendo alle macchine di eseguire compiti complessi, come la classificazione e la regressione

- **Classificazione:** L'obiettivo è assegnare un'etichetta o una classe predefinita a ciascun elemento di un insieme di dati in base alle sue caratteristiche o attributi. In altre parole, si tratta di sviluppare un modello che possa imparare a distinguere tra diverse categorie in modo automatico, basandosi su esempi di dati di addestramento
- **Regressione:** L'obiettivo è quello di stabilire una relazione matematica tra variabili di input e di output, in modo che il modello possa fare previsioni su nuovi dati, stimando un valore numerico anziché assegnare a un'istanza una classe o una categoria predefinita, come avviene nella classificazione

2.1 Neurone artificiale

Il neurone artificiale è un'unità computazionale fondamentale nelle reti neurali artificiali. Svolge la funzione di elaborare l'input ricevuto, applicare una trasformazione e produrre un output. In modo simile al neurone biologico, il neurone artificiale può essere visto come un'unità di calcolo che contribuisce al processo complessivo di apprendimento della rete neurale. Il primo modello di Neurone Artificiale è stato proposto nel 1943 da W.S. McCulloch e W. Pitts⁽⁴⁾.

Il neurone artificiale calcola informazioni attraverso un processo che simula, in modo semplificato, il funzionamento dei neuroni biologici. La computazione all'interno di un neurone artificiale avviene in tre fasi principali che possono essere riassunte in figura 2.1:

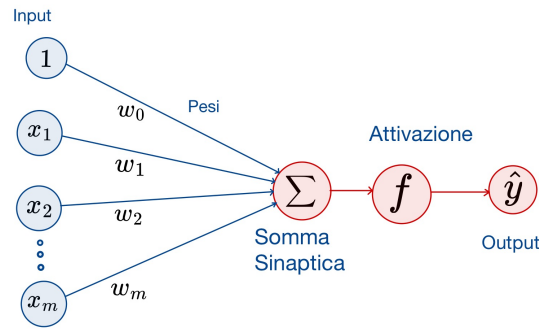


Figura 2.1 Modello del Neurone Artificiale

1. **Input pesato:** Il neurone riceve uno o più input. Ciascun input è moltiplicato per un peso associato alla connessione tra il neurone di input e il neurone in considerazione. I pesi rappresentano l'importanza relativa di ciascun input nel processo decisionale del neurone
2. **Somma sinaptica:** I risultati degli input moltiplicati per i pesi vengono sommati per generare una singola somma ponderata. Questa somma riflette l'importanza combinata degli input ponderati
3. **Funzione di attivazione:** Alla somma ponderata viene poi applicata una funzione di attivazione. Questa funzione introduce non linearità nel modello, consentendo al neurone di apprendere relazioni più complesse e di generare un output che può essere utilizzato come input per altri neuroni o come output finale della rete

Possiamo rappresentare matematicamente l'output di un neurone artificiale nell'equazione 2.1

$$\hat{y} = f \left(\sum_{i=1}^m w_i x_i + w_0 \right) \quad (2.1)$$

Dove \hat{y} è l'output del neurone, f la funzione di attivazione, m il numero di input w_i i pesi associati agli input, x_i gli input e b è il termine di bias.

Questo processo di calcolo è ripetuto per ogni neurone all'interno di una rete neurale durante la fase di addestramento, in questa fase i pesi delle connessioni vengono regolati per minimizzare l'errore tra l'output previsto e l'output desiderato.

2.2 Reti Neurali Artificiali

Nel 1958 Frank Rosenblatt nell'articolo⁽⁶⁾, introduce il primo modello di rete neurale artificiale chiamato "Perceptron".

Questa rete rappresenta un progresso notevole rispetto al modello di McCulloch e Pitts. La combinazione di diversi neuroni consente infatti alla rete di risolvere problemi complessi con variabili continue o discrete. Inoltre i pesi della rete sono ora variabili e possono essere modificati in modo da migliorare i risultati adattandosi al

problema specifico. L'idea di Rosenblatt è alla base di tutte le reti neurali che si utilizzano oggi.

Lo studio in questo campo del Machine Learning ha portato successivamente allo sviluppo delle Deep Neural Network. Le reti neurali più utilizzate sono le "Fully Connected" sono composte da unità chiamate "neuroni" descritti nella 2.1, organizzate in strati come in 2.2

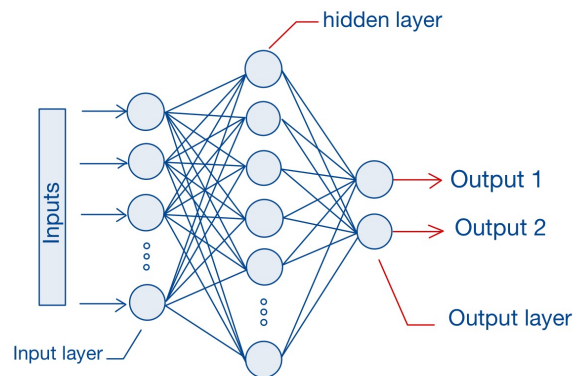


Figura 2.2 modello a grafo di una rete neurale

1. **Input Layer:** Questo strato riceve i dati in ingresso, che possono rappresentare, ad esempio, le caratteristiche di un'immagine o i valori numerici di un insieme di dati.
2. **Hidden Layer:** Questi strati elaborano le informazioni ricevute dallo strato di input. Ciascun neurone in uno strato nascosto è collegato a tutti i neuroni nello strato successivo e nello strato precedente. Ogni connessione ha un peso associato, che modifica l'importanza di ciascun input.
3. **Output Layer:** Questo strato produce il risultato finale della rete neurale. Può essere una classificazione, una previsione numerica o un'altra forma di output, a seconda del tipo di problema che la rete neurale sta risolvendo.

Il processo attraverso cui una rete neurale apprende dai dati è noto come "addestramento". Durante l'addestramento, la rete neurale aggiorna i pesi delle connessioni in modo che l'output prodotto si avvicini quanto possibile al risultato desiderato. Ciò viene fatto utilizzando algoritmi di ottimizzazione, come la Discesa lungo il gradiente.

In sintesi, una rete neurale apprende dai dati attraverso connessioni pesate tra neuroni, adattando i pesi durante l'addestramento per migliorare la sua capacità di fare previsioni o compiere compiti specifici.

2.3 Come una ANN può approssimare qualsiasi funzione

In questa dissertazione utilizzeremo una Dense Neural Network per approssimare l'hamiltoniana di un solido cristallino date le sue bande energetiche. Queste tipologia di reti sono molto semplici e generalizzabili, e di conseguenza offrono una buona adattabilità in molti tipi di compiti, come nel nostro caso la regressione. E' interessante vedere come una rete abbastanza densa, cioè composta da un numero cospicuo di neuroni nell'hidden layer può approssimare qualsiasi tipo di funzione. L'idea è descritta nel libro "Neural Network and Deep Learning"⁽⁵⁾ di Michael Nielsen.

Iniziando con il considerare un semplice NN con 1 solo hidden layer formato da 2 soli neuroni e attivazione sigmoide 2.2, questo produrrà come output la sua specifica funzione di attivazione

$$f(y) = \frac{1}{1 + e^{-y}} \quad (2.2)$$

La variabile y è l'output della funzione sinaptica definita in precedenza nella eq:2.1. Cambiando i valori di w e di b è possibile quindi modificare la forma della sigmoide in uscita dal neurone

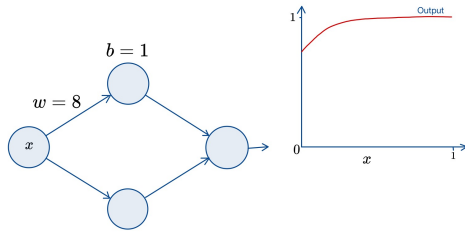


Figura 2.3 output di un neurone con pesi inizializzati

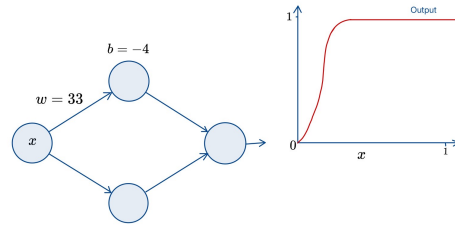


Figura 2.4 w cambia la pendenza della funzione di attivazione

Utilizzando un valore molto grande per w e definendo $S = -\frac{b}{w}$ come la posizione dello scalino, è possibile far tendere le sigmoidi a delle step-function in ogni punto del dominio. Lo stesso principio può essere applicato con un secondo neurone il cui output può essere combinato con quello del primo. Imponendo $w_2 = -w_1$, è possibile generare degli output uniformi che variano da S_1 a S_2 con un'altezza w_1 . Incorporando diverse coppie di neuroni in parallelo e regolando accuratamente i parametri, è fattibile approssimare la funzione utilizzando rettangoli. Incrementando il numero di coppie di neuroni nella rete, si migliora progressivamente l'approssimazione della funzione.

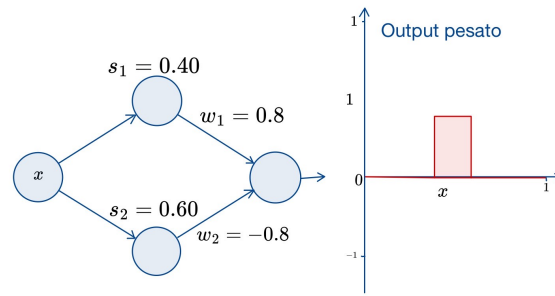


Figura 2.5 con due neuroni è possibile generare una step-function

Questo approccio può essere esteso a problemi di dimensioni maggiori, offrendo una procedura generalizzabile per l'approssimazione di funzioni complesse.

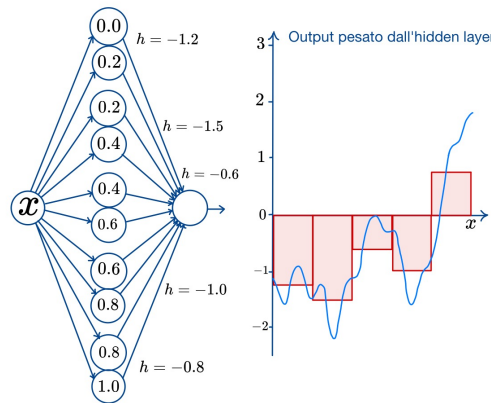


Figura 2.6 esempio dell'approssimazione di una generica funzione utilizzando 10 neuroni

2.4 Addestramento di una rete neurale

L'addestramento di una rete neurale è il processo che coinvolge l'ottimizzazione dei pesi delle connessioni tra i neuroni in accordo ad una data funzione di perdita o loss function che misura le prestazioni del modello rispetto ad uno specifico compito.

La loss function è una misura della discrepanza tra le previsioni del modello di rete neurale e i valori desiderati (etichette o target) associati ai dati di addestramento. La loss function valuta quanto il modello si discosta dalla soluzione corretta. Un esempio di loss function è riportato nell'eq 2.3, che rappresenta la loss MAE (Mean Absolute Error), ed è la loss che utilizzeremo successivamente nell'applicazione

pratica descritta nel capitolo 4.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i(w)| \quad (2.3)$$

dove N è il numero totale di esempi che viene fornito alla rete, y_i sono i valori reali per ogni esempio e \hat{y}_i le previsioni del modello per ogni esempio cioè l'output della rete. Possiamo dividere l'addestramento in due fasi principali:

- **FeedForward**: i pesi sono fissati (se è il passo zero vengono fissati casualmente) e l'input viene propagato layer per layer fino ai neuroni di output (function signal)
- **BackPropagation**: viene calcolato l'errore confrontando l'output con il target e il risultato viene propagato indietro, layer per layer (error signal). Durante questa fase, l'algoritmo di retropropagazione regola i pesi delle connessioni per ridurre l'errore. Questo processo di aggiornamento dei pesi è realizzato utilizzando algoritmi di ottimizzazione come la discesa del gradiente.

I pesi della rete w vengono aggiornati di modo che minimizzino la funzione 2.3. Questo minimo può essere cercato con tecniche di discesa lungo il gradiente. I pesi vengono aggiornati muovendosi nella direzione opposta al gradiente della loss function rispetto al parametro considerato, e di una quantità proporzionale al gradiente moltiplicato per un tasso di apprendimento (learning-rate) che regola la dimensione del passo di aggiornamento. Ad ogni iterazione scendiamo lungo le zone dove la funzione decresce, scegliendo quindi i pesi che minimizzano l'errore di predizione della rete neurale.

$$w_{i+1} = w_i - lr \frac{\partial L(n)}{\partial w_i(n)} \quad (2.4)$$

Dove abbiamo indicato con w_{i+1} il peso aggiornato, con w_i il vecchio peso e lr il tasso di apprendimento.

Una singola epoca rappresenta un ciclo completo di propagazione in avanti e retropropagazione dell'errore attraverso tutti i dati di addestramento. Le fasi di FeedForward e di BackPropagation vengono ripetute per molte epoche o fino a quando l'errore raggiunge un livello accettabile.

Per rendere più veloce e robusta la ricerca del minimo della loss function, la discesa lungo il gradiente viene eseguita in maniera stocastica, aggiornando i pesi dopo aver utilizzato solo una frazione (detta batch) degli eventi del training set.

Non tutti i parametri che definiscono la rete neurale vengono allenati tramite la discesa lungo il gradiente. Alcuni parametri (detti Iperparametri), come ad esempio il numero di batch, il tasso di apprendimento, il numero di neuroni nel singolo layer e il numero di hidden layer, vengono scelti in modo empirico provando diverse configurazioni e scegliendo quella che fornisce le prestazioni migliori rispetto allo specifico compito da risolvere.

Capitolo 3

Hamiltoniana di un solido cristallino e approssimazione Tight-Binding

Per predire le proprietà strutturali di una molecola o di un solido è fondamentale conoscere la sua hamiltoniana che codifica tali proprietà. L'hamiltoniana di un sistema è una funzione che restituisce l'energia totale del sistema sotto forma della somma di energia cinetica e potenziale, in funzione delle sue variabili dinamiche come la posizione e la velocità delle particelle

$$H(p, q) = \sum_{i=1}^N \frac{p_i^2}{2m_i} + V(q_1, q_2, \dots, q_n) \quad (3.1)$$

dove p_i sono i momenti coniugati alle coordinate q_i , m_i sono le masse delle particelle, e V è il potenziale del sistema.

Per iniziare a inquadrare il problema si può prendere l'hamiltoniana di un atomo di idrogeno, cioè un sistema composto da un protone e un elettrone, sapendo che l'interazione tra particelle con cariche diversa è data dalla legge di Coulomb il potenziale è di tipo Coulombiano e l'hamiltoniana risulta tale:

$$H(p, q) = \frac{P^2}{2M} + \frac{p^2}{2m} - \frac{ke^2}{r} \quad (3.2)$$

dove con M, P e m, p si è indicato rispettivamente la massa e l'impulso del protone e quelli dell'elettrone. Con r si è indicato la posizione relativa dell'elettrone rispetto al protone $|q - Q|$. Questo è un solito problema che ricorre spesso nella fisica e viene chiamato problema a due corpi come ad esempio il Sole e la Terra. Questi problemi possono essere risolti facilmente cambiando sistema di riferimento, infatti mettendosi nel centro di massa si può eliminare le coordinate assolute dei due corpi e trattare solo le coordinate relative, inoltre se si considerasse la massa del corpo più massiccio molto più grande del secondo $M \gg m$ il centro di massa cade proprio nella posizione del primo corpo, in questo caso l'hamiltoniana può essere riscritta come segue:

$$H(p, q) = \frac{p^2}{2m} - \frac{ke^2}{r} \quad (3.3)$$

L'analogia con il Sole e la Terra è solo qualitativa rispetto al problema a cui siamo interessati, questo perché nel trattare protoni ed elettroni si deve necessariamente usare la meccanica quantistica. L'approccio all'hamiltoniana non segue più quello classico di considerarla come una funzione dello stato del sistema, ma piuttosto come un operatore che agisce sulla funzione d'onda del sistema $|\psi\rangle$, e il modulo quadro di quest'ultima $|\psi|^2$ è la densità di probabilità di trovare una particella in una determinata posizione dello spazio. In questo caso il ruolo dell'operatore hamiltoniana è di natura duplice: i suoi autovalori forniscono le energie accessibili al sistema e le soluzioni dell'equazione di Schrödinger 3.4 per ogni autovalore forniscono gli orbitali atomici, nel caso dell'atomo di idrogeno.

Gli orbitali atomici in generale dipendono da tre numeri quantici i quali sono n :numero quantico principale, l :numero quantico angolare, m :numero quantico magnetico. Il concetto di orbitale atomico è molto utilizzato in chimica e in fisica e la forma degli orbitali è la superficie attorno al nucleo dove l'elettrone ha una certa probabilità di trovarsi, e questo concetto può essere esteso anche ai legami chimici nelle molecole.

$$\hat{H}|\psi\rangle = E_n|\psi\rangle \quad (3.4)$$

L'eq:3.4 per un atomo di idrogeno può essere risolta per cui ricaviamo lo spettro (l'insieme discreto di energie accessibili) e la funzione d'onda di un elettrone in un campo centrale attrattivo. Il problema diventa più complesso se si inizia a considerare i successivi elementi della tavola periodica, cioè se andiamo ad aggiungere elettroni e protoni all'interno del sistema, la difficoltà non sarà nell'aggiungere protoni al nucleo, questo perché si possono sempre utilizzare gli accorgimenti descritti sopra, ma quanto nelle interazioni tra elettroni: infatti questi avendo la stessa carica si repellono e in generale per un sistema ad N elettroni l'operatore hamiltoniana può essere scritta come segue.

$$\hat{H} = \sum_{i=1}^N \hat{H}_i^{1e} + \sum_{i,j=1}^N \hat{H}_{ij}^{2e} \quad (3.5)$$

con $i \neq j$, dove si definisce $H_i^{1e} = \frac{p_i^2}{2m} - \frac{e^2 z}{4\pi\epsilon_0 r_i}$ l'hamiltoniana dell' i -esimo elettrone e $H_{ij}^{2e} = \frac{1}{2} \frac{e^2}{4\pi\epsilon_0} \frac{1}{|r_i - r_j|}$ l'hamiltoniana di interazione fra due diversi elettroni ed N il numero di elettroni.

Il problema non può essere risolto in modo esatto, ma si possono ottenere delle soluzioni approssimate utilizzando le teorie di campo medio. Queste approssimazioni consistono nel rimpiazzare l'hamiltoniana di interazione con un'hamiltoniana efficace ad un elettrone. L'idea è perciò quella di utilizzare gli N orbitali atomici (ricavati dall'idrogeno) per ottenere i livelli energetici di un atomo ad N elettroni (approssimazione di elettroni quasi indipendenti)

3.1 Approssimazione di Born-Oppenheimer

Nelle molecole e nei solidi non è possibile separare l'hamiltoniana nelle variabili elettroniche e nucleari.

$$H(\vec{r}, \vec{R}) = T_e(\vec{r}) + T_N(\vec{R}) + V(\vec{r}, \vec{r}) + V(\vec{r}, \vec{R}) + V(\vec{R}, \vec{R}) \quad (3.6)$$

questo perché i termini potenziali coulombiani attrattivi mescolano le coordinate degli elettroni e dei nuclei. Born e Oppenheimer sfruttano il fatto che le masse associate alle due diverse coordinate sono molto diverse. In prima approssimazione possiamo considerare i protoni con una massa infinita, perciò la loro posizione è fissata nello spazio, in questo modo possiamo raggruppare i tre termini che dipendono dalle coordinate \vec{r} dell'equazione 3.6 in una hamiltoniana puramente elettronica

$$\hat{H}^e \psi_i^e(\vec{r}; \vec{R}) = \epsilon_i^e \psi_i^e(\vec{r}; \vec{R}) \quad (3.7)$$

Le autofunzioni elettroniche a nuclei fissi che risolvono l'equazione 3.7 e i suoi relativi autovalori (che dipendono parametricamente dalle posizioni dei nuclei) permettono di ottenere soluzioni approssimate all'equazione di Schrödinger completa (elettronica e nucleare)

3.2 Approssimazione LCAO

Sebbene l'equazione 3.7 sia risolvibile analiticamente per una molecola come H_2^+ , per molecole e solidi a molti elettroni non esiste una soluzione analitica. Data la completezza degli autostati dell'atomo di idrogeno isolato si possono approssimare gli autostati di una qualunque molecola o solido cristallino con una combinazione lineare infinita di orbitali atomici, ottenendo la soluzione esatta con metodi numerici. Se però ci interessa una trattazione approssimata in grado di tradursi in un calcolo semplice e in un'intuizione semplice della fisica del sistema in esame possiamo adottare una base con pochi stati capace di rappresentare i primi stati dell'hamiltoniana elettronica.

Questo metodo è noto come LCAO (Linear Combination of Atomic Orbitals) consiste nell'approssimare la funzione d'onda elettronica di una molecola o di un solido a molti elettroni in una combinazione lineare di orbitali atomici definiti nello spazio di Hilbert a un elettrone diminuendo enormemente la dimensionalità del problema. L'idea è quella di prendere la base più piccola possibile data dagli orbitali atomici centrati sui nuclei della molecola (per N autostati servono almeno N orbitali atomici). Prendendo in esempio una molecola biatomica i primi due stati dell'energia possono essere ricavati dalla combinazione lineare di due orbitali atomici 1s centrati nei siti A e B

$$\begin{aligned} \psi_+ &= c_A \psi_A + c_B \psi_B \\ \psi_- &= c_A \psi_A - c_B \psi_B \end{aligned} \quad (3.8)$$

Si può notare come l'autostato del primo livello energetico essendo dispari cambia di segno, questo comporta che rispetto allo stato fondamentale (stato legante) la densità di probabilità di trovare un elettrone tra i due atomi è più piccola (stato anti-legante). Ritrovando un'efficace e semplice spiegazione di come possono formarsi i legami chimici

3.3 Simmetria nei solidi cristallini

Per trattare gli elettroni in solidi cristallini si può utilizzare il Teorema di Bloch, tuttavia una formulazione equivalente è fornita dal libro "Elementi di fisica atomica,

molecolare e dei solidi" ⁽¹⁾

Un aiuto fondamentale per capire le proprietà elettroniche di un solido cristallino sono le proprietà di simmetria di cui essi godono, questo perché gli atomi non sono disposti a caso bensì seguono un passo ben preciso come le mattonelle di un pavimento, e una volta nota la posizione e la forma di una di queste cellette, chiamate celle primitive, sono note tutte le altre.

Per vedere come sfruttare la simmetria di traslazione si può partire da una catena di atomi a distanza costante a (passo della catena) chiusa su sé stessa. In questo modo il perimetro della catena vale Na dove N è il numero di siti atomici che stiamo considerando e se si prende il limite $N \rightarrow \infty$ il poligono tende ad una circonferenza di raggio $r = \frac{Na}{2\pi}$, e se si proietta la catena lungo una linea orizzontale la simmetria di rotazione diventa indistinguibile dalla simmetria di traslazione per a o multipli di a . L'hamiltoniana elettronica sarà invariante di conseguenza per rotazioni di $\frac{2\pi}{Na}$ attorno al suo asse di rotazione, questo vuol dire che le soluzioni dell'equazione l'operatore hamiltoniana condivide una base di autostati con l'operatore $\hat{R}_{\frac{2\pi}{N}}$ con autovalore $e^{\pm 2\pi i m/aN}$ ottenendo stati distinti per ogni valore di $m = 0, \pm a, \pm 2a, \dots \pm (N-1)a/2$ se N è dispari e $m = 0, \pm a, \pm 2a, \dots \pm (N-1)a/2, \pm Na/2$ se N è pari, e con l'analogia con la catena unidimensionale descritta precedentemente si può riscrivere il numero quantico m come un numero d'onda $k(a)$ in funzione di a

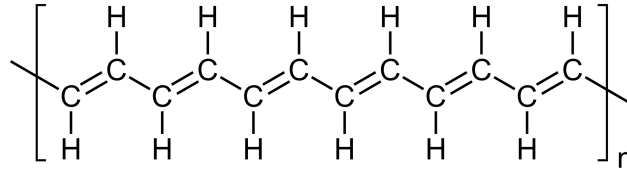
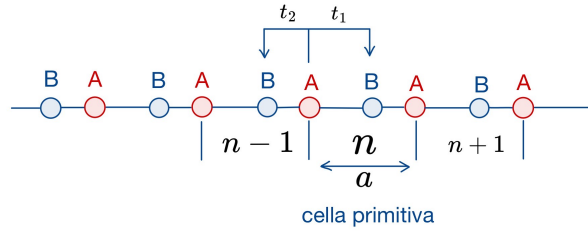
- $k_m = \frac{2\pi m}{Na} = 0, \pm \frac{2\pi}{Na}, \pm \frac{4\pi}{Na}, \dots, \pm \frac{(N-1)\pi}{Na}$ per N dispari
- $k_m = \frac{2\pi m}{Na} = 0, \pm \frac{2\pi}{Na}, \pm \frac{4\pi}{Na}, \dots, \pm \frac{(N-1)\pi}{Na}, + \frac{\pi}{a}$ per N pari

Prendendo N abbastanza grande il numero d'onda varia con continuità nella regione $[-\frac{\pi}{a}, +\frac{\pi}{a}]$ chiamata convenzionalmente prima zona di Brillouin. Con questo si vuol dire che gli autostati della hamiltoniana elettronica sotto una traslazione pari al passo della catena acquistano un fattore e^{ika} , a questo punto se si adotta una base minima LCAO con ad esempio un solo orbitale per sito atomico $|n\rangle$, $n = 0, 1, 2, \dots, N$ la simmetria sotto rotazione $\hat{R}_{\frac{2\pi}{Na}}$ è sufficiente a determinare gli autostati dell'hamiltoniana elettronica che corrispondono alla combinazione lineare:

$$|\phi_n(k)\rangle = \frac{1}{\sqrt{N}} \sum_{n=1}^N e^{ikna} |n\rangle \quad (3.9)$$

3.4 Ricavare l'hamiltoniana Tight-Binding

A questo punto si introdurrà subito il modello che verrà utilizzato per il trans-poliacetilene. Si considera quindi un cristallo uni-dimensionale con due atomi per sito e un orbitale $1s$ per atomo, in questo caso la catena è composta da legami covalenti di due tipi, infatti nel caso del trans-poliacetilene $[C_2H_2]_N$ considereremo sia il legame C-H che il legame C-C

**Figura 3.1** struttura del trans-poliacetilene**Figura 3.2** Modello di un polimero composto da una catena 1D con due atomi per cella primitiva

Si considerano solamente i legami, cosiddetti stretti, cioè si prendono diversi da zero gli elementi di matrice dell'hamiltoniana dati dall'interazione degli orbitali con quelli primi vicini. Andiamo a considerare quindi nella base LCAO minimale gli autovettori approssimati

- $|\phi_A(k)\rangle = \frac{1}{\sqrt{N}} \sum_n e^{ikna} |n, A\rangle$
- $|\phi_B(k)\rangle = \frac{1}{\sqrt{N}} \sum_n e^{ikna} |n, B\rangle$

dalla sovrapposizione di queste si ottiene la funzione d'onda del cristallo

$$|\Phi\rangle = C_A |\phi_A(k)\rangle + C_B |\phi_B(k)\rangle \quad (3.10)$$

Lo scopo è quello di risolvere l'equazione di Schrödinger con la 3.10 ricavandone i suoi autovalori, cioè le bande energetiche

$$H|\Phi\rangle = E|\Phi\rangle \quad (3.11)$$

moltiplicando la 3.11 per $\langle\phi_j|$ Per ogni $j = A, B$ si ottiene:

$$\sum_{j'}^{A,B} H_{jj'} C_{j'} = E \sum_{j'}^{A,B} S_{jj'} C_{j'} \quad (3.12)$$

La 3.12 è chiamata equazione agli autovalori generalizzata dove si può definire la matrice hamiltoniana $H_{j,j'} = \langle\phi_j|H|\phi_{j'}\rangle$ e la matrice di overlap come $S_{j,j'} = \langle\phi_j|\phi_{j'}\rangle$ e le bande energetiche sono date dal determinante della seguente matrice:

$$\begin{pmatrix} H_{AA} - \lambda S_{AA} & H_{AB} - \lambda S_{AB} \\ H_{BA} - \lambda S_{BA} & H_{BB} - \lambda S_{BB} \end{pmatrix} \quad (3.13)$$

Sebbene la presenza della matrice di overlap è proprio dovuta al fatto che la base LCAO non è ortogonale, si può assumere quest'ultima come una matrice unitaria, questo è supportato dal fatto che non porta risultati qualitativi apprezzabili considerarla o meno, e quindi le bande energetiche saranno date dalle soluzioni del polinomio caratteristico:

$$\det[H - \lambda] = 0 \quad (3.14)$$

La parte più difficile è quella di andare a calcolare gli elementi di matrice, perché quest'ultimi non sono banali.

Esiste tuttavia una strategia chiamata Tight-Binding semi empirico che consiste per prima cosa a considerare soltanto le interazioni tra gli atomi primi vicini e di considerare questi elementi di matrice come parametri da determinare con i dati sperimentali. Questa idea di utilizzare gli elementi di matrice come parametri fu introdotta da J. C. Slater e G. F. Koster in un fondamentale articolo del 1954⁽⁷⁾. Il punto è che una volta che sia stato stabilito quali debbano essere gli elementi di matrice non nulli, trattare questi come parametri empirici significa, in qualche modo, riassorbire in essi parte del ruolo di quelli che si sono forzati a zero. In questo modo si hanno due tipi di legami, quelli inter-sito e extra-sito che sono codificati dai seguenti elementi di matrice

$$\begin{aligned} \langle \phi_A | H | \phi_A \rangle &= \frac{1}{N} \sum_{nn'}^N e^{ika(n-n')} \langle n' A | H | n A \rangle \\ \langle \phi_A | H | \phi_B \rangle &= \frac{1}{N} \sum_{nn'}^N e^{ika(n-n')} \langle n' A | H | n B \rangle \end{aligned} \quad (3.15)$$

Usando esplicitamente l'approssimazione Tight-Binding si possono assegnare i seguenti parametri empirici

$$\langle n' A | H | n B \rangle = \begin{cases} t_1 & \text{se } n' = n \text{ (legame inter-sito)} \\ t_{12} & \text{se } n' = n \pm 1 \text{ (legame extra-sito)} \\ 0 & \text{altrimenti} \end{cases} \quad (3.16)$$

$$\langle n A | H | n A \rangle = \frac{1}{N} \sum_n^N \epsilon_0 = \epsilon_0 \quad (3.17)$$

Si possono ricavare gli elementi della matrice opposti a questi facendone il complesso coniugato, dato che la matrice è hermitiana. Per la periodicità del polimero gli elementi di matrice non dipendono da n. Ottenendo in questo modo la matrice di Tight-Binding per la quale nel prossimo capitolo verrà spiegato come è possibile ricavare i parametri migliori

$$H^{TB} = \begin{pmatrix} \epsilon_0 & t_1 + t_{12}e^{-ika} \\ t_1 + t_{12}e^{+ika} & \epsilon_0 \end{pmatrix} \quad (3.18)$$

Da cui si possono ricavare le bande energetiche con pochi passaggi, che dipendono con continuità da k nella prima zona di Brillouin

$$\lambda_{12}(k) = \epsilon_0 \pm \sqrt{t_1^2 + t_{12}^2 + 2t_1t_{12}\cos(ka)} \quad (3.19)$$

Capitolo 4

Metodologia

Come già introdotto precedentemente siamo interessati a ricavare la matrice 3.18 utilizzando una rete neurale. L'architettura segue l'idea illustrata nella tesi di Andrea Cacioppo⁽²⁾ può essere riassunta come in figura 4.1.

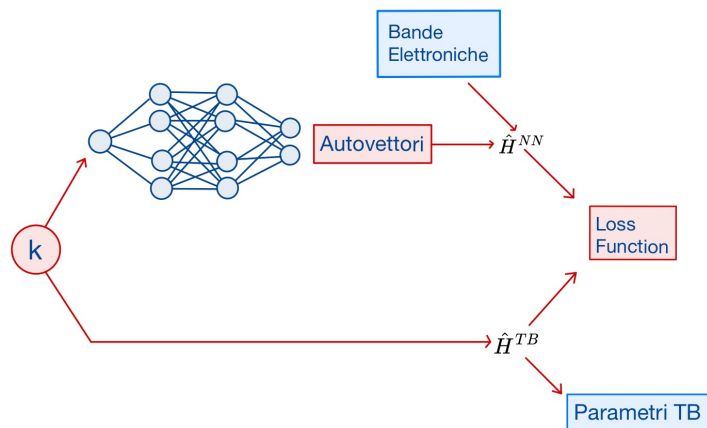


Figura 4.1 Architettura del modello Anti-Adversarial

Le informazioni a disposizione sono le bande energetiche $\lambda_{12}(k)$ e la forma della matrice 3.18. I parametri della matrice vengono inizializzati casualmente secondo una distribuzione piatta in $[0,5]$ ad eccezion fatta del parametro ϵ_0 che viene preso da una distribuzione piatta in $[0,1]$, quest'ultimo perché si è notato che l'algoritmo è molto suscettibile a questo parametro sebbene non cambia la forma delle bande, infatti essendo una costante può essere messa a 0.

Il Procedimento che esegue l'algoritmo è il seguente: Generati i valori di k nella prima zona di Brillouin la DNN approssima gli autovettori della matrice di Tight-Binding e con la conoscenza delle bande energetiche viene costruita una matrice H^{NN} nella rappresentazione della rete neurale, parallelamente con la conoscenza della forma funzionale della matrice attraverso i valori precedentemente generati di k viene costruita la matrice H^{TB} nella rappresentazione parametrica.

Ad ogni passo di allenamento viene calcolata la loss function come la distanza tra le due rappresentazioni. In questo modo la rete neurale e la rappresentazione parametrica sono allenate a cooperare minimizzando la distanza tra esse, restituendo i parametri della matrice di Tight-Binding che meglio approssimano le bande energetiche

4.1 H^{NN} nella rappresentazione della DNN

In questo paragrafo vedremo come si può ricavare la matrice H^{NN} attraverso gli autovettori approssimati della DNN e gli autovalori $\lambda_{12}(k)$. La matrice in forma diagonale è data da:

$$A_D = V^{-1}AV \quad (4.1)$$

Dove V è la matrice le cui colonne sono gli autovettori della matrice A e A_D è la matrice in forma diagonale. Invertendo la relazione 4.1 e inserendo gli autovettori approssimati e la matrice A_D otteniamo

$$H^{NN} = VA_DV^{-1} \quad (4.2)$$

In generale per una matrice complessa 2x2 i suoi autovettori ortonormalizzati possono essere scritti in funzione di due angoli compresi in $[0, 2\pi]$ come segue:

$$\begin{aligned} v_1 &= (\cos(\theta(k)), \sin(\theta(k))e^{i\phi(k)}) \\ v_2 &= (-\sin(\theta(k))e^{-i\phi(k)}, \cos(\theta(k))) \end{aligned} \quad (4.3)$$

dove la matrice A_D è data da:

$$A_D = \begin{pmatrix} \lambda_1(k) & 0 \\ 0 & \lambda_2(k) \end{pmatrix} \quad (4.4)$$

eseguendo i prodotti tra matrici della 4.2 otteniamo la matrice:

$$\begin{aligned} H^{NN} &= \begin{pmatrix} \lambda_1 \cos^2(\theta) + \lambda_2 \sin^2(\theta) & \mu \cos(\theta) \sin(\theta) \cos(\phi) \\ \mu \cos(\theta) \sin(\theta) \cos(\phi) & \lambda_1 \cos^2(\theta) + \lambda_2 \sin^2(\theta) \end{pmatrix} \\ &+ i \begin{pmatrix} 0 & \mu \cos(\theta) \sin(\theta) \sin(\phi) \\ -\mu \cos(\theta) \sin(\theta) \sin(\phi) & 0 \end{pmatrix} \end{aligned} \quad (4.5)$$

dove per compattezza di notazione abbiamo indicato con $\mu = \lambda_1 - \lambda_2$

4.2 Generazione Dati

L'input dell'algoritmo sono le bande energetiche $\lambda_{12}(k)$, dato che si vuol far vedere la ragionevolezza del metodo utilizzato, genereremo i dati sinteticamente e testeremo infine il modello allenato su un esempio della vita reale, cioè sul polimero del trans-poliacetilene. Prenderemo quindi come veri i parametri della 3.18 estratti casualmente da una distribuzione piatta tra $[0,5]$ ad eccezion fatta per il parametro ϵ_0 che viene preso da una distribuzione piatta in $[0,1]$. Presi 500 punti nella zona di Brillouin per k vengono generate le matrici per ognuno di questi punti e gli autovalori corrispondenti per ogni punto cioè le bande energetiche

4.3 Training

Il modello Anti-Adversarial viene allenato come segue : Una batch di valori di k viene estratto casualmente dai dati in input, vengono generati a partire da questi N_{batch} matrici $H^{NN}(k)$ e $H^{TB}(k)$. Viene calcolata la MAE elemento per elemento tra le due matrici. I parametri del modello vengono aggiornati contemporaneamente con l'ottimizzatore adam, considerando il rapporto tra il learning rate della NN e della H^{TB} come un iperparametro da ottimizzare nel prossimo capitolo

4.4 Valutazione del modello

Per valutare le prestazioni del modello si può avere una stima di quanto i parametri ricavati si discostino da quelli veri secondo la metrica:

$$A = \frac{100}{1 + ||\vec{\Delta p}||_2} \quad (4.6)$$

dove $||\vec{\Delta p}||_2$ è la distanza tra il vettore dei parametri di H^{TB} e quelli veri, specificato come segue:

$$||\vec{\Delta p}||_2 = \sqrt{(\epsilon_{0true} - \epsilon_0)^2 + (t_{1true} - t_1)^2 + (t_{12true} - t_{12})^2 + (a_{true} - a)^2} \quad (4.7)$$

Con $A = 100$ i due vettori sono completamente sovrapposti, mentre con $A = 0$ i due vettori sono a distanza infinita. La 4.6 è da intendere come una indicazione di quanto il modello funzioni correttamente

Capitolo 5

Risultati

Per questo tipo di compito è fondamentale ricavare gli iperparametri migliori. Il set di iperparametri utilizzati sono: Batch-size, n-neurons, learning rate, il ratio e il dropout. Dove il ratio è definito come segue:

$$ratio = \frac{lr^{TB}}{lr^{NN}} \quad (5.1)$$

Ottenendo dal learning rate, e la 5.1 i parametri:

- $lr^{NN} = (lr)$
- $lr^{TB} = (lr) * (ratio)$

E' stato osservato empiricamente che l'iperparametro 5.1 è fondamentale per la stabilità dell'algoritmo, infatti sono stati scelti soltanto valori $ratio > 1$ così che $lr^{TB} > lr^{NN}$, facendo sì che l'apprendimento dei parametri della H^{TB} sia più veloce dell'apprendimento della rete H^{NN} , questo porta ad un apprendimento congiunto migliore. Inoltre un altro iperparametro cruciale è quello di Dropout: Il parametro di dropout è quantificato come un numero $|p| < 1$ e corrisponde alla percentuale delle connessioni sinaptiche che vengono eliminate in un determinato layer, la tecnica di dropout rientra nelle tecniche di regolarizzazione, eliminare delle connessioni sinaptiche infatti rende più difficile il compito della rete neurale, questo aiuta il modello a non focalizzarsi troppo sulle caratteristiche specifiche del campione (di dimensione finita) di training, e in ultima analisi ad evitare fenomeni di overfitting. In questo problema si è riscontrato un effettivo beneficio dall'utilizzo del Dropout, andando così a bilanciare l'apprendimento dei parametri di Tight-Binding.

Per trovare gli iperparametri migliori, questi ultimi sono stati variati individualmente, variandone uno e fissando i rimanenti. Per ogni set di iperparametri sono stati utilizzate 5 configurazioni dei parametri dell'hamiltoniana, dopo di che viene selezionato e salvato il modello migliore (con l'accuracy più alta), e di conseguenza il set di iperparametri che ne consegue per poter ricavare i parametri della hamiltoniana del trans-poliacetilene. Gli iperparametri utilizzati e il loro intervallo di variazione è indicato di seguito:

- $lr = [0.001, 0.01]$
- $ratio = [5, 10]$

- $bs = [16, 32]$
- $neurons = [52, 128]$
- $dropout = [0.1, 0.3]$

Di seguito viene riportato il set degli iperparametri che ha presentato un'accuracy maggiore (su 20 epoche di addestramento):

Accuracy	95.2193
loss	0.6542
lr	0.01
ratio	5
bs	32
neurons	52
dropout	0.1

Tabella 5.1 iperparametri ottimali

Inoltre possiamo verificare nel grafico riportato in figura 5.1 che la loss function del modello migliore ha un andamento decrescente rispetto all'accuracy, questa è un indicazione a posteriori che la loss function utilizzata per addestrare la rete neurale è un buon surrogato dell'accuracy.

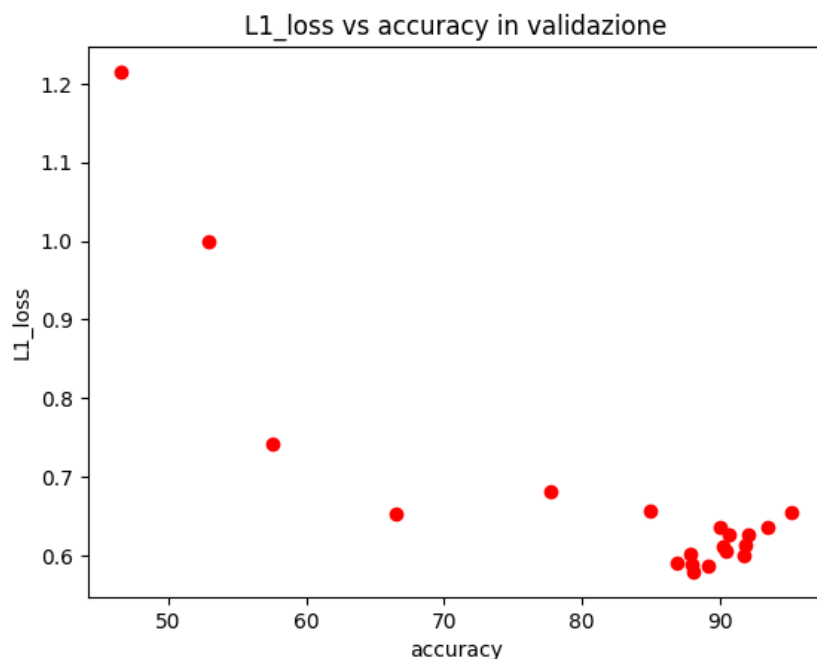


Figura 5.1 andamento della Loss function in funzione dell'accuracy per il modello migliore

Di seguito vengono riportati i migliori risultati delle 10 simulazioni fatte sul transpoliacetilene

Accuracy	Loss
80.09	2.52
66.25	2.38
86.82	2.27
86.91	2.25
89.99	2.18
92.67	2.15
81.35	2.41
89.00	2.256
86.84	2.347
85.81	2.176

Tabella 5.2 10 simulazioni per il poliacetilene

Inoltre possiamo a posteriori valutare le prestazioni dell'algoritmo osservando come le bande elettroniche ricostruite coi parametri ricavati si accordano a quelle reali:

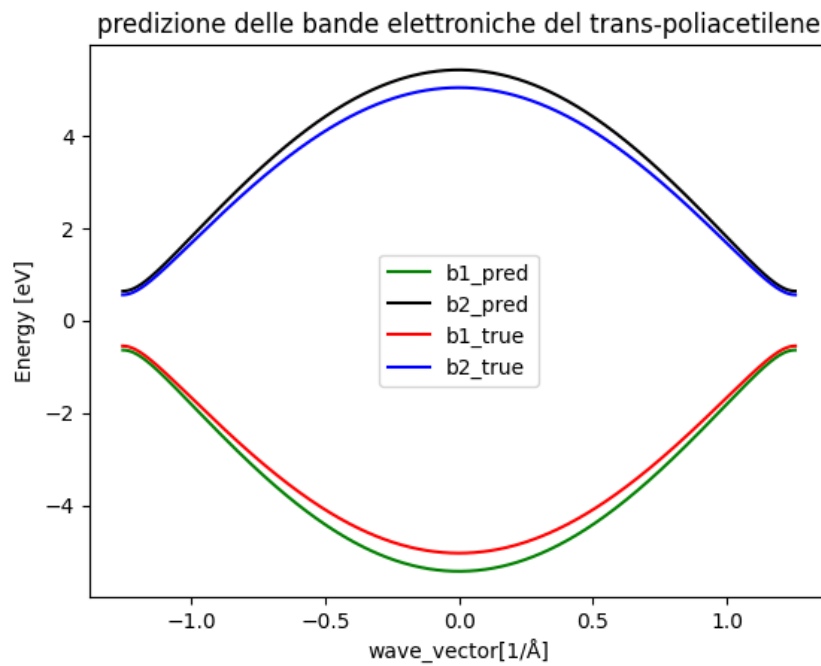


Figura 5.2 Sovrapposizione delle bande predette da quelle reali, per il trans-poliacetilene

Di seguito vengono mostrati i parametri ricavati dall'algoritmo e quelli reali (presi dall'articolo⁽³⁾)

	reali	fit
ϵ_0	0	-0.0046
t_1	2.796	3.0321
t_{12}	2.242	2.3945
d	2.5	2.5016

Tabella 5.3 Parametri ricavati

Come si può notare dalla 5.2 che le bande predette non si sovrappongono perfettamente su quelle reali sebbene i risultati soddisfano le aspettative, questo potrebbe essere da imputare alla mole di parametri da ricavare, un passo in avanti in questo studio sarebbe quello di analizzare l'importanza di uno specifico parametro rispetto agli altri, così da poter agire individualmente e avere uno spazio di ricerca migliore.

Capitolo 6

Conclusioni

In questa dissertazione si è descritto un nuovo approccio per ricavare i parametri di Tight-Binding di un cristallo uni-dimensionale, che abbraccia il metodo semi empirico a quello del machine learning.

É stato discusso nel capitolo 5 come è possibile ricavare l'hamiltoniana di un cristallo uni-dimensionale nell'approssimazione di tight-binding, per poi applicare il metodo un polimero reale, il Trans-Poliacetilene. Si é mostrato come le bande energetiche che codificano le proprietà strutturali del Trans-poliacetilene possono essere ricostruite attraverso l'apprendimento dei parametri dell'hamiltoniana nell'approssimazione tight-binding. Un possibile sviluppo del metodo, per renderlo di maggior interesse pratico, potrebbe consistere nel lavorare direttamente con le bande energetiche misurate sperimentalmente.

Inoltre questo tipo di approccio inverso, può essere utilizzato per cercare particolari materiali date le proprietà strutturali codificate dalle bande energetiche.

Bibliografia

- [1] Giovanni Battista Bachelet, Vito DP Servedio, et al. Elementi di fisica atomica, molecolare e dei solidi. In *Elementi di fisica atomica, molecolare e dei solidi*, pages 1–159. Aracne Editrice Internazionale srl, 2014.
- [2] Andrea Cacioppo. Deep learning for the parameter estimation of tight-binding hamiltonians, 2020.
- [3] M Ali M Keshtan and Mahdi Esmailzadeh. Tight-binding hamiltonian considering up to the third nearest neighbours for trans polyacetylene. *Journal of Physics: Condensed Matter*, 32(28):285401, 2020.
- [4] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biology*, 52:99–115, 1990.
- [5] Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA, 2015.
- [6] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [7] John C Slater and George F Koster. Simplified lcao method for the periodic potential problem. *Physical review*, 94(6):1498, 1954.