



UNIVERSITÀ DEGLI STUDI DI TRIESTE

Dipartimento di Fisica
Corso di Laurea in Fisica

Sviluppo di un potenziale interatomico basato su reti neurali profonde per singoli strati di ditellurio di tungsteno

Relatori:
Dott. Antimo Marrazzo

Presentata da:
Riccardo Dal Molin

Sessione straordinaria
Anno Accademico 2020/2021

Ringraziamenti

Mi è doveroso dedicare questo spazio del mio elaborato alle persone che hanno contribuito, con il loro instancabile supporto, alla realizzazione dello stesso. In primis, un ringraziamento speciale al mio relatore Marrazzo Antimo, per la sua infinita disponibilità, per i suoi indispensabili consigli e per le conoscenze trasmesse durante tutto il percorso di stesura dell'elaborato. Ringrazio infinitamente i miei genitori che mi hanno sempre sostenuto, avendo fiducia in me e appoggiando ogni mia decisione, fin dalla scelta del mio percorso di studi. Spero di rendervi orgogliosi. Ringrazio la mia fidanzata Gloria per avermi trasmesso la sua immensa forza e il suo coraggio. Grazie per tutto il tempo che mi hai dedicato e per il tuo supporto nei momenti più difficili. Grazie perché ci sei sempre stata. Infine un grazie di cuore a tutti i colleghi e amici, con cui ho condiviso l'intero percorso universitario. Grazie per aver condiviso le gioie e difficoltà di questi anni di studi e grazie per tutti i momenti di spensieratezza.

Indice

1	Teoria	4
1.1	Che cos'è la dinamica molecolare	4
1.1.1	L'uso dell'approssimazione classica	5
1.1.2	Algoritmo di integrazione	5
1.2	Potenziale interatomico	6
1.2.1	Potenziali a molti corpi	8
1.3	Dinamica molecolare da principi primi	9
1.3.1	L'approssimazione di Born-Oppenheimer	9
1.3.2	Superficie di energia potenziale	12
1.4	Reti neurali per il calcolo del PES: DeePMD	12
1.5	Deep Neural Network	13
1.5.1	Apprendimento	14
1.6	DeePMD	16
1.6.1	Coordinate	16
1.6.2	Sub-Networks	19
1.6.3	Training	22
2	Risultati	23
2.1	Il ditellurio di tungsteno	23
2.2	Creazione del data set e training	25
2.3	Test del potenziale	26
2.4	Correlazione tra frames	28
2.4.1	Model Deviation	31
2.5	Simulazioni con LAMMPS	33
2.5.1	Potenziale di seconda generazione	35
3	Conclusioni	36

Sommario

La **dinamica molecolare** è, al giorno d'oggi, uno strumento fondamentale in molti ambiti della ricerca, dalla fisica della materia alla chimica fino alla biologia computazionale. Lo sviluppo negli ultimi anni delle reti neurali e del *machine learning* sta rapidamente cambiando i paradigmi della dinamica molecolare, portando all'utilizzo di reti neurali artificiali come strumento per rappresentare la superficie di energia potenziale interatomica, una quantità fondamentale in dinamica molecolare. Trovare un metodo sia preciso che efficiente, dal punto di vista del costo computazionale, per la rappresentazione di quest'ultima è una delle maggiori sfide nella dinamica molecolare. In questo lavoro abbiamo usato il metodo ***deep potential molecular dynamics***, la cui rappresentazione della superficie di energia potenziale interatomica fornisce un pratico modo di ottenere la precisione della dinamica molecolare da principi primi ad una frazione del costo computazionale, grazie alla flessibilità delle reti neurali profonde o ***deep neural networks*** (DNN). La prima parte della tesi si concentra sui fondamenti della dinamica molecolare e delle reti neurali, per poi approfondire nel dettaglio la struttura teorica del ***deep potential molecular dynamics***. La seconda parte è, invece, dedicata all'uso effettivo di questo metodo, nella quale si spiega come abbiamo applicato questa tecnica alla costruzione di un potenziale interatomico per un sistema a monostrato di **ditellurio di tungsteno** (WTe_2), utilizzando come esempi di apprendimento iniziali (*training set*) per DNN, traiettorie atomiche ottenute attraverso dinamica molecolare di Born-Oppenheimer sfruttando la teoria del funzionale densità. Abbiamo, poi, investigato la precisione e l'affidabilità del potenziale così ottenuto, concentrandoci in particolare sugli effetti delle correlazioni tra configurazioni atomiche del *training set* ottenuto con la dinamica molecolare di Born-Oppenheimer. Successivamente abbiamo prodotto, utilizzando LAMMPS, nuove e più lunghe traiettorie, grazie al potenziale prodotto con ***deep potential molecular dynamics***. Da queste traiettorie abbiamo ricavato dei *frames* (posizioni atomiche, forze e energie per ogni time-step) completamente non correlati. Per le configurazioni atomiche di questi *frames*, utilizzando la teoria del funzionale densità, abbiamo calcolato la superficie di energia potenziale interatomica da cui abbiamo ricavato nuove forze interatomiche. Queste forze, caratterizzate dalla precisione tipica della teoria del funzionale densità, sono state quindi confrontate con quelle precedenti, mostrando la precisione della rete nel riprodurre il potenziale interatomico. Infine, utilizzando i *frames* di questo tipo, attraverso un ulteriore apprendimento della DNN su questi ultimi, abbiamo prodotto un PES di seconda generazione e mostrato l'aumento di precisione di questo nuovo potenziale su un largo intervallo di temperature (200K-1000K).

1 Teoria

1.1 Che cos'è la dinamica molecolare

La dinamica molecolare (MD) è una tecnica di simulazione computazionale, dove l'evoluzione temporale di un set di atomi interagenti è ottenuta dall'integrazione numerica delle loro equazioni del moto [4]. Si considera un sistema formato da N atomi interagenti, che seguono le leggi della meccanica classica¹, considerando gli atomi come punti materiali soggetti a certe forze, pertanto descritti da:

$$\mathbf{F}_i = m_i \frac{d^2 \mathbf{r}_i}{dt^2} , \quad (1)$$

per ogni atomo i -esimo nel sistema di N atomi. Dove $\mathbf{F}_i$ descriverà la forza totale agente sull'atomo i -esimo, m_i la sua massa e $\mathbf{r}_i$ la sua posizione nel tempo. Conoscendo la forza $\mathbf{F}_i$ è possibile integrare numericamente l'equazione (1), ottenendo la traiettoria $\mathbf{r}_i$ dell'atomo i -esimo. Per ogni atomo si devono integrare 3 equazioni differenziali del secondo ordine, una per ogni componente cartesiana. Facendo questo per tutti gli N atomi, si ottiene la traiettoria nello spazio delle fasi $6N$ -dimensionale del sistema ($3N$ posizioni e $3N$ momenti). Questo metodo è dunque totalmente deterministico, in contrasto con altre tecniche di simulazione come il metodo Monte Carlo. Date le posizioni e velocità (momenti) iniziali l'evoluzione temporale è in principio² completamente determinata. La dinamica molecolare, pertanto, descrive i sistemi atomici come sistemi di punti materiali che interagiscono fra loro come se fossero degli atomi, dunque, il nucleo concettuale si basa sull'individuazione delle forze $\mathbf{F}_i$ che forniscono la dinamica atomica corretta dei punti materiali. Approfondiremo questo nel dettaglio nel paragrafo 1.2 e successivi.

Principalmente la MD viene utilizzata per lo studio di sistemi in equilibrio termico, nei quali le traiettorie nello spazio delle fasi calcolate attraverso la MD si distribuiscono secondo una funzione di distribuzione statistica (o densità di punti caratteristici ρ) comunemente chiamata ensemble statistico [4]. Un esempio è l'ensemble microcanonico (NVE), caratterizzato da una distribuzione di punti caratteristici che mantiene costante l'energia totale o ancora l'ensemble canonico (NVT) dove la temperatura T è costante e la distribuzione di punti caratteristici segue la distribuzione di Boltzmann:

$$\exp(-\mathcal{H}(\{p, q\})/k_b T) ,$$

dove $\mathcal{H}(\{p, q\})$ è l'hamiltoniana del sistema, funzione di $\{p, q\}$, set di posizioni e momenti nello spazio delle fasi.

In accordo con la fisica statistica, dunque, le grandezze fisiche macroscopiche di sistemi all'equilibrio sono descritte da medie di queste ultime su traiettorie nello spazio delle fasi distribuite secondo un certo ensemble statistico. Tuttavia nelle simulazioni di MD, le traiettorie ottenute non sono mai tutte le traiettorie descritte nel corrispettivo ensemble.

¹Nel paragrafo 1.1.1, verrà approfondito perché e in quali casi si è autorizzati a effettuare questa approssimazione.

²In pratica, la finitezza del tempo di interazione e di memoria disponibile nel computer fa sì che errori aritmetici possano far deviare le traiettorie calcolate da quelle effettive, come approfondito in 1.1.2

Sembrerebbe allora che le medie delle grandezze fisiche di interesse, $f(\{p, q\})$, sui valori che esse assumono sulle traiettorie $\gamma = \{p, q\}$, ottenute dalla MD:

$$\langle f \rangle = \int_{\gamma} f(\{p, q\}) \{dp, dq\} ,$$

non siano analoghe alle medie sugli ensemble statistici. Qui abbiamo denotato con T la durata della simulazione che produce la traiettoria γ . Grazie all'ipotesi di ergodicità che garantisce che traiettorie nello spazio delle fasi, nel limite di tempi lunghi tendono a tutto lo spazio delle fasi, la media precedente converge come

$$\lim_{T \rightarrow \infty} \int_{\gamma} f(\{p, q\}) \{dp, dq\} = \int_{\Gamma} \rho(\{p, q\}) f(\{p, q\}) \{dp, dq\} ,$$

con $\rho(\{p, q\})$ la distribuzione statistica di un certo ensemble e Γ il volume dell'intero spazio delle fasi del sistema. Pertanto nel limite di simulazioni lunghe ci si aspetta di ricavare, dalle medie sulle traiettorie ottenute dalla MD, le stesse proprietà termodinamiche ricavabili dalle medie sugli ensemble. Oltre a questo uso “tradizionale”, la MD può essere utilizzata anche per lo studio di processi non in equilibrio e come strumento efficiente per l'ottimizzazione di strutture superando i minimi di energia locale (*simulated annealing*) [4].

1.1.1 L'uso dell'approssimazione classica

Nel paragrafo precedente abbiamo studiato il sistema atomico classicamente, supponendo che gli atomi seguano le leggi della meccanica newtoniana. Tuttavia gli atomi sono correttamente descritti dalla meccanica quantistica. Sorge spontanea una domanda, perché e quando utilizzare un'approssimazione classica? Una semplice test per verificare la validità di quest'ultima è utilizzare la lunghezza d'onda termica di de Broglie, definita come

$$\Lambda = \sqrt{\frac{2\pi\hbar}{Mk_B T}} \quad (2)$$

Dove M è la massa atomica, T la temperatura del sistema. Essa non è altro che la lunghezza d'onda media di de Broglie in un sistema a temperatura T [4]. Se dunque si ha che, data a la distanza media tra atomi vicini, vale che $\Lambda \ll a$, allora la sovrapposizione delle funzioni d'onda atomiche sarà trascurabile e gli effetti quantistici di conseguenza, permettendo una trattazione classica. Quello che si ottiene è che l'approssimazione classica risulta poco precisa per elementi leggeri (H, He, Ne) ma efficace per elementi più pesanti. Infine, gli effetti quantistici divengono importanti per ogni sistema, con l'abbassamento della temperatura [4].

1.1.2 Algoritmo di integrazione

Il cuore della dinamica molecolare è l'algoritmo di integrazione temporale, necessario per risolvere numericamente equazioni differenziali del secondo ordine, dello stesso tipo dell'equazione (1). L'algoritmo si basa sul concetto di discretizzazione temporale, in cui

la variabile continua tempo viene trasformata in una variabile discreta $t_n = n\delta t$ con $n \in \mathbb{N}$, così che un certo valore di t può sempre essere scritto come $t_{n-1} + \delta t$. A questo punto, le velocità e le posizioni (soluzioni dell'equazione (1)) possono essere ottenute con uno sviluppo di Taylor intorno al punto t_{n-1} . Il più comune algoritmo di questo tipo, che vedremo nel dettaglio, è il cosiddetto algoritmo di Verlet [2, 4]. L'idea generale è di sviluppare in serie la posizione $\mathbf{r}_i(t_n)$, fino al terzo ordine, per ottenere le posizioni nei punti $t_n + \delta t$ e $t_n - \delta t$. Chiamando $\mathbf{a}_i(t_n)$ l'accelerazione e $\mathbf{v}_i(t_n)$ la velocità, si ottiene

$$\begin{aligned}\mathbf{r}_i(t_n + \delta t) &= \mathbf{r}_i(t_n) + \mathbf{v}_i(t_n)\delta t + \frac{1}{2}\mathbf{a}_i(t_n)\delta t^2 + \frac{1}{6}\frac{d^3\mathbf{r}_i}{dt^3}\bigg|_{t_n}\delta t^3 + O(\delta t^4) , \\ \mathbf{r}_i(t_n - \delta t) &= \mathbf{r}_i(t_n) - \mathbf{v}_i(t_n)\delta t + \frac{1}{2}\mathbf{a}_i(t_n)\delta t^2 - \frac{1}{6}\frac{d^3\mathbf{r}_i}{dt^3}\bigg|_{t_n}\delta t^3 + O(\delta t^4) ,\end{aligned}$$

da cui sommando le due equazioni

$$\mathbf{r}_i(t_{n+1}) = \mathbf{r}_i(t_n + \delta t) = 2\mathbf{r}_i(t_n) - \mathbf{r}_i(t_n - \delta t) + \mathbf{a}_i(t_n)\delta t^2 + O(\delta t^4) . \quad (3)$$

Si è ottenuto dunque, il valore della posizione atomica nell'istante $t_n + \delta t$, in funzione della posizione nei momenti antecedenti t_n e t_{n-1} , e dell'accelerazione $\mathbf{a}_i(t_n) = \mathbf{F}_i(t_n)/m_i$. Ripetendo questo procedimento per tutti i t_n voluti, si ottiene la traiettoria nello spazio dell'atomo i -esimo (di durata $n\delta t$), con un errore associato dell'ordine di δt^4 . Un problema con questa versione dell'algoritmo di Verlet è che la velocità non è direttamente calcolata. Essa non è strettamente necessaria per la conoscenza dell'evoluzione temporale del sistema, tuttavia, può essere utile nello studio del comportamento del sistema e necessaria per il computo dell'energia cinetica, essenziale per il controllo della conservazione dell'energia ($E = K + V$) nella MD. La velocità si può ricavare semplicemente da

$$\mathbf{v}_i(t_n) = \frac{\mathbf{r}_i(t_{n+1}) - \mathbf{r}_i(t_{n-1})}{2\delta t} . \quad (4)$$

1.2 Potenziale interatomico

Dall'equazione (1) è evidente che l'ingrediente principale per fare MD sia la forza che agisce su ogni atomo. In sistemi isolati la conoscenza della forza su ogni atomo si riduce alla conoscenza dell'energia potenziale del sistema: ovvero una funzione $V(\mathbf{r}_1, \dots, \mathbf{r}_N)$ della posizione di ogni nucleo atomico, che rappresenta l'energia potenziale del sistema quanto gli atomi sono disposti in una specifica configurazione. La forza su ogni atomo è dunque ricavata dal gradiente del potenziale rispetto alla posizione dell'atomo i -esimo:

$$\mathbf{F}_i = -\nabla_{\mathbf{r}_i} V(\mathbf{r}_1, \dots, \mathbf{r}_N) . \quad (5)$$

Il linea di principio, conoscendo la forma analitica esatta del potenziale, il sistema è completamente descritto e la MD risulterebbe la semplice risoluzione numerica delle equazioni del moto (1). Tuttavia la forma analitica di $V(\mathbf{r}_1, \dots, \mathbf{r}_N)$, per sistemi atomici reali, è impossibile da ottenere a priori. La sfida della MD è costituita, dunque, dalla difficoltà nel formulare un potenziale che descriva il sistema in modo accurato. Per ottenere quest'ultimo si possono costruire dei potenziali *ad hoc*, detti potenziali empirici, attraverso l'analisi

dei dati sperimentali, che siano in grado di descrivere il sistema. Seguendo questa linea di pensiero, senza perdita di generalità, il potenziale può essere diviso nei termini dipendenti dalle coordinate di atomi singoli ϕ_1 , di coppie di atomi ϕ_2 , di triplette di atomi ϕ_3 e così via [2]:

$$V(\mathbf{r}_1, \dots, \mathbf{r}_N) = \sum_i^N \phi_1(\mathbf{r}_i) + \sum_i^N \sum_{j>i}^N \phi_2(\mathbf{r}_i, \mathbf{r}_j) + \sum_i^N \sum_{j>i}^N \sum_{k>j>i}^N \phi_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \dots, \quad (6)$$

dove la notazione $j > i$ e $k > j > i$ ha lo scopo di considerare tutte le coppie e triplette una volta sola. $\phi_1(\mathbf{r}_i)$ rappresenta il contributo di un campo potenziale esterno, mentre gli altri termini rappresentano i contributi di interazione tra atomi. Il secondo termine, ϕ_2 , rappresenta il potenziale di interazione di coppia, detto *pairwise potential*. I termini superiori al secondo vengono chiamati termini di potenziale a molti corpi (*many-body potential*). Considerando l'assenza di un campo esterno, il primo termine dell'equazione (6), è nullo e il potenziale diviene dipendente dalla posizione reciproca degli atomi, in questa forma il potenziale prende il nome di *potenziale interatomico*. In questo modo il potenziale è descritto da

$$V(\mathbf{r}_1, \dots, \mathbf{r}_N) = \sum_i^N \sum_{j>i}^N \phi_2(r_{ij}) + \sum_i^N \sum_{j>i}^N \sum_{k>j>i}^N \phi_3(r_{ij}, r_{ik}, \theta_{ijk}) + \dots, \quad (7)$$

dove $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ e θ_{ijk} é l'angolo tra r_{ij} e r_{ik} . A causa della pesantezza computazionale dei termini a molti corpi, nelle simulazioni di dinamica molecolare vengono raramente utilizzati. Ogni termine k -esimo nell'equazione (7), infatti, porta con sé una somma su $N(N-1)\dots(N-k)$ elementi, pertanto i termini relativi al potenziale a molti corpi ($k = 3, 4, \dots$) implicano una somma su un numero di elementi che scala almeno come $o(N^2)$. Questo porta un enorme dispendio di risorse computazionali che peggiora con l'aumentare delle dimensioni del sistema. Per questo motivo, i termini a molti corpi vengono generalmente trascurati, nonostante si evinca, dall'analisi di un grande numero di dati sperimentali, che il termine ϕ_3 risulti significativo per sistemi con densità tipiche delle fasi liquide e solide³[2]. Il potenziale più utilizzato nelle simulazioni è, dunque, della forma

$$V = \sum_i^N \sum_{j>i}^N \phi_2(r_{ij}), \quad (8)$$

dove il potenziale *pairwise* può essere costruito per rappresentare al meglio la fisica del sistema in studio. Una scelta comune è il potenziale di *Lennard-Jones* (L-J), costruito dall'analisi sistematica di una grande quantità di dati sperimentali⁴ sull'Argon [2, 4]. Da notare che l'equazione (7) è invariante sotto rotazioni, traslazioni e anche sotto permutazioni delle posizioni atomiche (per stessa specie chimica). D'altronde il potenziale atomico deve rispettare queste simmetrie che sono naturali del sistema.

³Gli studi sono stati fatti su gas inerti, ci si aspetta che per specie chimiche più interagenti il contributo dei termini di potenziale a multi corpo sia ancora più intenso [13]

⁴I dati sperimentali utilizzati includevano, *molecular beam scattering*, spettroscopia dell'argon molecolare, inversione della dipendenza dalla temperatura dal secondo coefficiente del viriale e proprietà dello stato solido.

1.2.1 Potenziali a molti corpi

Nel paragrafo 1.2, la funzione potenziale viene approssimata ai contributi di coppia atomo-atomo (equazione (8)), dove la forma funzionale di $\phi_2(r_{ij})$ viene ottenuta da dati sperimentali. Per questo, in generale, il potenziale *pairwise* è limitato nell'applicazione a sistemi reali. Un comune e molto utilizzato potenziale di coppia in MD, il potenziale di L-J, trova applicazioni realistiche che tuttavia, si limitano, solo a gas nobili, dove non sono presenti elettroni per formare legami atomici e gli atomi sono attratti tra loro solo attraverso la debole forza di van der Waals. Sistemi più interessanti, come metalli e semiconduttori, non possono essere modellizzati con questo potenziale [4] e nemmeno con altre forme di potenziali di coppia. Un'analisi dei dati sperimentali di gas inerti allo stato solido cristallizzati in un reticolo cubico a facce centrate mostra come, il contributo di ϕ_3 non possa essere trascurato [2]. Un nuovo modo di concepire i potenziali empirici di questi ultimi materiali giunge con il concetto di densità di atomi come variabile chiave da considerare nella costruzione di un potenziale. L'idea fisica principale è che i legami tra atomi diventano più deboli man mano che l'ambiente locale diviene più affollato. Questo implica che l'energia potenziale di un certo atomo i -esimo, dipenderà fortemente dal numero di atomi vicini [4]:

$$E_i \propto \sqrt{\sum_j \langle i|H|j \rangle^2} . \quad (9)$$

Dove $\langle i|H|j \rangle$ è l'integrale di sovrapposizione tra gli orbitali dell'atomo i e del suo vicino j -esimo. Nel formalismo del *tight-binding* gli orbitali sono localizzati, dunque le loro funzioni andranno a zero dopo una certa distanza, per cui gli integrali che contribuiranno all'energia locale dell'atomo i -esimo saranno solo quelli relativi agli atomi all'interno di un raggio di *cutoff*. L'energia totale del sistema può essere ottenuta, allora, come:

$$V = \sum_i^N E_i(\mathbf{R}_i(\mathcal{N}_i)) . \quad (10)$$

Dove $\mathcal{N}_i$ è il numero di atomi interni al raggio di *cutoff*, e di conseguenza

$$\mathbf{R}_i(\mathcal{N}_i) = (\mathbf{r}_{1i}, \dots, \mathbf{r}_{\mathcal{N}_i i}) , \quad (11)$$

con $\mathbf{r}_{ji} = \mathbf{r}_j - \mathbf{r}_i$. Chiaramente anche in questo caso la forma analitica dell'energia locale E_i deve essere prodotta *ad hoc* attraverso lo studio di dati sperimentali relativi ai vari sistemi che si vogliono considerare. Infatti, su questa struttura teorica sono sviluppati vari modelli per V che differiscono per la scelta di E_i , come *embedded atom method*, *glue model*, *Finnis-Sinclair potentials* [4]. Come si può notare, questi potenziali sono una modifica del concetto di *many-body potential* introdotto nel paragrafo 1.2. Ogni atomo contribuisce all'energia potenziale globale con un contributo E_i che sostanzialmente rappresenta tutti i termini superiori al secondo dell'equazione (7), con la differenza che, mentre nel potenziale a molti corpi i contributi dei singoli atomi dipendono da tutti gli atomi del sistema, in questo caso dipenderanno solo dal loro ambiente locale.

1.3 Dinamica molecolare da principi primi

Fin ora il sistema di atomi considerato è stato approssimato a un insieme di punti materiali interagenti tra loro attraverso un potenziale costruito *ad hoc* da dati sperimentali. Abbiamo, infatti, fin qui descritto il sistema atomico in modo molto semplice, trascurando i principi che regolano l'interazione atomica e limitandoci alla costruzione di un modello (classico) che si adatti il più possibile alla descrizione dei dati sperimentali. Questo approccio, è sicuramente efficiente dal punto di vista computazionale, ma limitato nella precisione e nella trasferibilità⁵ dei risultati. Nei paragrafi successivi vedremo come sia possibile descrivere il sistema da principi primi, ovvero attraverso la descrizione dei fenomeni elementari che governano l'interazione atomica, e vedremo come questo approccio ci permetterà di ottenere un potenziale interatomico molto preciso a scapito dell'aumento del costo computazionale.

1.3.1 L'approssimazione di Born-Oppenheimer

Per seguire questo approccio dobbiamo abbandonare l'idea di poter descrivere gli atomi come punti materiali soggetti a un potenziale classico. Volendo descrivere accuratamente un sistema atomico, infatti, non ci si può basare su un'approssimazione di questo tipo. Un sistema, isolato, di atomi interagenti è composto nella realtà da N nuclei e N_e elettroni che interagiscono tra loro attraverso l'interazione coulombiana. La loro dinamica segue le leggi della meccanica quantistica, pertanto il sistema è correttamente descritto dall'operatore hamiltoniano del tipo:

$$\mathcal{H} = \sum_i^N \frac{\mathbf{P}_i^2}{2M_i} + \sum_k^{N_e} \frac{\mathbf{p}_k^2}{2m} + \frac{1}{2} \sum_{i \neq j}^N \frac{Z_i Z_j e^2}{|\mathbf{R}_i - \mathbf{R}_j|} + \frac{1}{2} \sum_{k \neq l}^{N_e} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} - \sum_i^N \sum_k^{N_e} \frac{Z_i e^2}{|\mathbf{R}_i - \mathbf{r}_k|}, \quad (12)$$

con $\mathbf{R}_i$ posizione e $\mathbf{P}_i$ momento dell'atomo i -esimo, $\mathbf{r}_i$ posizione e $\mathbf{p}_k$ momento dell'elettrone k -esimo, Z_i indica il numero di protoni dell'atomo i -esimo (numero atomico), M_i indica la massa del nucleo dell'atomo i -esimo e m indica la massa dell'elettrone. Ricordiamo che il momento va sempre interpretato come operatore ($\mathbf{p}_i = -i\hbar\nabla_i$). Si riconoscono i termini relativi all'energia cinetica degli atomi T_N e degli elettroni T_e e i vari termini relativi alle interazioni coulombiane: atomo-atomo V_{NN} , atomo-elettrone V_{Ne} e elettrone-elettrone V_{ee} .

$$T_N = \sum_i^N \frac{\mathbf{P}_i^2}{2M_i}, T_e = \sum_k^{N_e} \frac{\mathbf{p}_k^2}{2m}$$

$$V_{NN} = \frac{1}{2} \sum_{i \neq j}^N \frac{Z_i Z_j e^2}{|\mathbf{R}_i - \mathbf{R}_j|}, V_{Ne} = \sum_i^N \sum_k^{N_e} \frac{Z_i e^2}{|\mathbf{R}_i - \mathbf{r}_k|}, V_{ee} = \frac{1}{2} \sum_{k \neq l}^{N_e} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|}.$$

In linea di principio, risolvendo l'equazione di Schrödinger per questa hamiltoniana, si ottiene una funzione d'onda $\Psi(\mathbf{R}, \mathbf{r}, t)$ che descrive interamente il sistema, dove $\mathbf{R}$ indica

⁵Con il termine trasferibilità si indica la capacità di un potenziale interatomico di lavorare correttamente su sistemi con differente struttura elettronica [4].

la dipendenza da tutte le coordinate nucleari e $\mathbf{r}$ da quelle elettroniche. Essendo il sistema isolato con un'energia totale E_n , l'equazione di Schrödinger si riduce all'equazione agli autovalori:

$$\mathcal{H}\Psi_n(\mathbf{R}, \mathbf{r}) = [T_N + T_e + V_{NN} + V_{Ne} + V_{ee}] \Psi_n(\mathbf{R}, \mathbf{r}) = E_n \Psi_n(\mathbf{R}, \mathbf{r}) , \quad (13)$$

con n numero quantico del sistema. Risolvere l'equazione (13) risulta considerevolmente complesso, per fortuna il problema può essere notevolmente semplificato apportando alcune approssimazioni. Nel 1923 Max Born e Robert Oppenheimer notarono che i nuclei sono tipicamente più pesanti degli elettroni e si muovono su scale temporali almeno due ordini di grandezza maggiori di quelle degli elettroni [3, 4]

$$\frac{E_e}{E_N} \sim \frac{\omega_e}{\omega_N} \sim \sqrt{\frac{M}{m}} \sim 10^2 - 10^5 , \quad (14)$$

basandosi su questo fatto, ipotizzarono che, dal punto di vista degli elettroni, i nuclei atomici non si muovessero e che d'altro canto gli elettroni reagissero al moto nucleare istantaneamente. Da queste considerazioni si può pensare di descrivere il sistema globale come l'interazione tra il sistema nucleare e il sistema elettronico, dunque descritto da uno spazio di Hilbert dato dal prodotto tensore degli spazi che descrivono il sistema elettronico e nucleare $\mathbb{H}^e \otimes \mathbb{H}^N$. In questo modo, direttamente dai termini dell'hamiltoniana totale $\mathcal{H}$, si può pensare di definire un'hamiltoniana elettronica, dipendente solo dai gradi di libertà elettronici e dalla posizione dei nuclei in modo parametrico:

$$\mathcal{H}^e(\mathbf{R}) = T_e + V_{NN}(\mathbf{R}) + V_{Ne}(\mathbf{R}) + V_{ee} . \quad (15)$$

In questo modo si può costruire un'equazione agli autovalori per gli elettroni, dove le posizioni dei nuclei sono trattate come parametri e non come gradi di libertà dinamici

$$\mathcal{H}^e(\mathbf{R})\Phi_\alpha(\mathbf{R}; \mathbf{r}) = V_\alpha(\mathbf{R})\Phi_\alpha(\mathbf{R}; \mathbf{r}) , \Phi_\alpha(\mathbf{R}; \mathbf{r}) \in \mathbb{H}^e . \quad (16)$$

Dove con α indichiamo il numero quantico del sistema elettronico. In modo analogo si può pensare che esista una hamiltoniana nucleare $\mathcal{H}^N$ che fornisca un'auto-funzione $\Theta_{\alpha\beta}(\mathbf{R}) \in \mathbb{H}^N$, che in modo del tutto generale dipenda sia dal numero quantico β del sistema nucleare che α del sistema elettronico. Dato che il prodotto tensore di autostati è un' autostato, la funzione d'onda descritta in (13), può essere espansa nei gradi di libertà elettronici

$$\Psi_{\alpha\beta}(\mathbf{R}, \mathbf{r}) = \sum_{\alpha'} C_{\alpha'\alpha\beta} \Phi_{\alpha'}(\mathbf{R}; \mathbf{r}) \Theta_{\alpha'\beta}(\mathbf{R}) , \quad (17)$$

dove $C_{\alpha'\alpha\beta}$ sono i coefficienti di Fourier dell'espansione mentre $\alpha\beta$ definiscono il numero quantico totale $n = \alpha\beta$. Ora, utilizzando la più comoda notazione bra-ket⁶, proiettiamo l'equazione (13) in $\langle \Phi_{\alpha''} |$:

$$\langle \Phi_{\alpha''}(\mathbf{R}) | \mathcal{H} | \Psi_{\alpha\beta} \rangle = \langle \Phi_{\alpha''}(\mathbf{R}) | T_N + \mathcal{H}^e | \Psi_{\alpha\beta} \rangle = E_{\alpha\beta} \langle \Phi_{\alpha''}(\mathbf{R}) | \Psi_{\alpha\beta} \rangle . \quad (18)$$

⁶Ricordiamo che il legame tra le due notazioni è dato dalla corrispondenza biunivoca $\langle \mathbf{r} | \Phi_\alpha(\mathbf{R}) \rangle = \Phi_\alpha(\mathbf{R}; \mathbf{r})$

Inserendo ora l'equazione (17), nella (18), si ottiene

$$\sum_{\alpha'} C_{\alpha'\alpha\beta} \langle \Phi_{\alpha''}(\mathbf{R}) | T_N [| \Phi_{\alpha'}(\mathbf{R}) \rangle \otimes | \Theta_{\alpha'\beta} \rangle] + C_{\alpha''\alpha\beta} V_{\alpha''}(\mathbf{R}) | \Theta_{\alpha''\beta} \rangle = C_{\alpha''\alpha\beta} E_{\alpha\beta} | \Theta_{\alpha''\beta} \rangle . \quad (19)$$

A questo punto, è necessario applicare la cosiddetta approssimazione adiabatica, la quale assume che gli stati elettronici non possano essere eccitati dall'energia cinetica dei nuclei. Questo implica che l'operatore T_N agisca solo sulle funzioni d'onda nucleari. L'equazione (19) diviene allora

$$T_N | \Theta_{\alpha''\beta} \rangle + V_{\alpha''}(\mathbf{R}) | \Theta_{\alpha''\beta} \rangle = E_{\alpha\beta} | \Theta_{\alpha''\beta} \rangle . \quad (20)$$

Evidentemente α e α'' identificano lo stesso stato, per cui si prende $\alpha'' = \alpha$. Abbiamo dunque definito l'hamiltoniana nucleare

$$\mathcal{H}_\alpha^N | \Theta_{\alpha\beta} \rangle = [T_N + V_\alpha(\mathbf{R})] | \Theta_{\alpha\beta} \rangle = E_{\alpha\beta} | \Theta_{\alpha\beta} \rangle \quad (21)$$

Quest'equazione mostra come i nuclei si muovano in un potenziale effettivo $V_\alpha(\mathbf{R})$, ottenuto dalla soluzione dell'equazione (16) e dunque dipendente dai differenti autostati contrassegnati con α . Chiamiamo questa funzione potenziale effettivo, al cui interno si trovano sia gli effetti di interazione degli elettroni sia quelli dei nuclei atomici. Infine si nota che, poiché si suppone che gli elettroni si adattino istantaneamente al moto nucleare (approssimazione di Born-Oppenheimer) e l'energia cinetica dei nuclei non ecciti lo stato elettronico (approssimazione adiabatica), si possono considerare gli elettroni sempre nel loro *ground state*, ovvero il loro stato di energia minima⁷[1]. Pertanto $V_\alpha(\mathbf{R}) = V_0(\mathbf{R}) \equiv V(\mathbf{R})$. Dunque, l'equazione (21), proiettando su $\langle \mathbf{R} |$, si riscrive come

$$\mathcal{H}^N \Theta_\beta(\mathbf{R}) = [T_N + V(\mathbf{R})] \Theta_\beta(\mathbf{R}) = E_\beta \Theta_\beta(\mathbf{R}) , \quad (22)$$

in cui il pedice $\alpha = 0$ è stato soppresso. Analogamente l'equazione (17) risulta semplicemente

$$\Psi_{0\beta}(\mathbf{R}, \mathbf{r}) = C_{0\beta} \Phi_0(\mathbf{R}; \mathbf{r}) \Theta_\beta(\mathbf{R}) , \quad (23)$$

In questo modo il problema agli autovalori (13) è disaccoppiato in una parte elettronica, equazione (16), e in una parte nucleare, equazione (22). Dunque, grazie a queste approssimazioni, si riesce a descrivere il sistema atomico come due sistemi separati, quello elettronico e quello nucleare, la cui interazione reciproca è rappresentata, nel sistema elettronico dalla dipendenza parametrica della funzione d'onda dalle posizioni nucleari $\mathbf{R}$, mentre nel sistema nucleare dalla presenza del potenziale effettivo $V(\mathbf{R})$ derivante dalla soluzione dell'equazione di Schrödinger elettronica. Utilizzando questa struttura teorica, tuttavia, la dinamica molecolare che ne risulta è alquanto complessa⁸. Infatti, per ottenere l'equazione d'onda $\Psi_\beta(\mathbf{R}, \mathbf{r})$ che descrive il sistema atomico è necessario risolvere parametricamente l'equazione (16) e quindi usare il potenziale effettivo trovato per risolvere l'equazione (22).

⁷Questa condizione è dimostrata per la maggior parte dei cristalli in condizioni normali.

⁸La dinamica molecolare interamente quantistica di questo tipo viene prodotta utilizzando il metodo dell'integrale sul cammino di Feynman. Questo tipo di dinamica molecolare è comunemente detta PIMD dall'inglese *path integral molecular dynamics*.

1.3.2 Superficie di energia potenziale

Con la struttura teorica precedente si considerano sia i nuclei che gli elettroni come oggetti quantistici, dunque descritti da equazioni d'onda. Tuttavia nel paragrafo 1.1.1, si è visto come la dinamica di atomi pesanti a temperature non troppo vicine allo zero assoluto possa essere trattata classicamente. Pertanto, limitandoci allo studio di sistemi solidi cristallini, si può semplificare la descrizione dei nuclei atomici, e trattarli classicamente. Dunque, l'equazione di Schrödinger (22) può essere sostituita con l'equazione di Newton classica, ovvero si considera $\mathcal{H}^N = T_N + V(\mathbf{R})$ non più un operatore ma l'hamiltoniana classica del sistema (l'energia totale). Così facendo si ricava attraverso le equazioni di hamilton

$$\mathbf{F}_i = \frac{d\mathbf{P}_i}{dt} = -\nabla_{\mathbf{r}_i} \mathcal{H}^N = -\nabla_{\mathbf{r}_i} V(\mathbf{R}) . \quad (24)$$

Pertanto è evidente come i nuclei atomici si muovano su una superficie (o meglio iper-superficie) di energia potenziale efficace, che in questo contesto è comunemente chiamata *potential energy surface* (PES) o più correttamente, *adiabatic potential energy surface* [16]. In questo modo i nuclei sono ancora trattati come punti materiali classici soggetti adesso però ad un potenziale quantistico derivante dalla soluzione dell'equazione di Schrödinger elettronica. La dinamica molecolare che ne risulta, detta appunto dinamica molecolare da principi primi o dinamica molecolare *ab initio* (AIMD dall'inglese *ab initio molecular dynamics*), unisce la descrizione quantistica a quella classica, utilizzando l'approssimazione classica dove risulta ancora molto efficiente, ovvero sui nuclei, e descrive, invece, quantisticamente gli elettroni. Quello che si è ottenuto non è molto diverso dalla dinamica molecolare descritta nel paragrafo 1.1, con la differenza sostanziale però che il potenziale interatomico è ora ricavato da principi primi. Per ogni step di simulazione viene risolta numericamente l'equazione agli autovalori (22) per le posizioni atomiche in quell'istante ($\mathbf{R}_t$), ottenendo l'equazione d'onda $\Phi(\mathbf{R}_t, \mathbf{r})$ e il valore del PES $V(\mathbf{R}_t)$, da questo si ricavano le forze dall'equazione (24), usate nell'algoritmo di interazione temporale introdotto nel paragrafo 1.1.2. La necessità di risolvere numericamente l'equazione (22) per ogni step di simulazione, tuttavia, complica notevolmente la MD, poiché impone la determinazione della funzione d'onda elettronica dipendente da $3N_e$ coordinate cartesiane. La teoria del funzionale densità (DFT dall'inglese *density functional theory*), fornisce un valido modo per ottenere il PES $V(\mathbf{R}_t)$ e conseguentemente $\Phi(\mathbf{R}_t, \mathbf{r})$, attraverso la descrizione dell'autovalore energia $V(\mathbf{R})$ (al *ground state*) come funzionale della densità elettronica [6].

1.4 Reti neurali per il calcolo del PES: DeePMD

Avendo introdotto la AIMD si può pensare che scompaia la necessità di costruire potenziali interatomici empirici. L'uso dell'approssimazione di Born-Oppenheimer nella dinamica molecolare, a cui ci si riferisce in letteratura anche come *Born-Oppenheimer molecular dynamics* (BOMD), insieme alla DFT produce risultati molto accurati ma limitati dal costo computazionale a sistemi di centinaia di atomi, per scale temporali nell'ordine dei ~ 100 ps [17]. Pertanto questo tipo di MD risulta molto lenta e difficilmente applicabile, anche con l'utilizzo dei computer più moderni, allo studio di fenomeni fisici su scale

temporali superiori all'ordine dei 100 ps e su sistemi con scale di lunghezza estese [8, 17]. D'altra parte, la costruzione empirica di potenziali interatomici, descritti solitamente come potenziali di coppia che siano in grado di interpolare i dati sperimentali, porta, come già detto, a una MD sicuramente più efficiente dal punto di vista computazionale, ma limitata nella precisione e nella trasferibilità dei risultati. Dalle considerazioni precedenti, è dunque evidente che il problema più impegnativo della MD è rappresentare il PES, sia in modo accurato, che efficiente. I moderni sviluppi nell'utilizzo di reti neurali e *machine learning* (ML) [7] come strumento per il calcolo del *potential energy surface* portano nuove speranze nella soluzione di questo dilemma. L'idea è quella di utilizzare reti neurali artificiali (ANN), che da un set di esempi di configurazioni atomiche (*training data*), ottenute con AIMD, possano “apprendere” la forma del PES. In questo modo, producendo col minimo sforzo computazionale un set di traiettorie da principi primi, è possibile interpolare il PES, eliminando così la necessità di risolvere l'equazione (16), per ogni step di simulazione. Negli ultimi anni questi metodi, basati sul ML, hanno ottenuto alcuni notevoli successi [11, 17, 18], in particolare il metodo *Deep Potential Molecular Dynamics* (DeePMD), usato in questo lavoro. Molti sono, infatti, gli esempi in letteratura che mostrano come questo metodo permetta di eseguire simulazioni di dinamica molecolare con una precisione comparabile a quella della BOMD attraverso DFT e un'efficienza competitiva con MD basata su potenziali empirici [18]. La DeePMD è basata sulla costruzione di un potenziale interatomico a molti corpi, nella forma (10), dunque, ottenibile come somma di contributi atomici:

$$V = \sum_i E_i ,$$

dove E_i rappresenta il contributo energetico dell'atomo i -esimo, che seguendo il modello esposto nel paragrafo 1.2.1, è una funzione che dipende solo dalle posizioni degli atomi nell'ambiente locale dell'atomo i -esimo. Questa funzione è ottenuta attraverso il *machine learning* di una *Deep Neural Network* (DNN), che date in input le posizioni atomiche nell'ambiente locale di un atomo i -esimo, fornirà il valore dell'energia locale E_i . Vedremo nel dettaglio come è costruita la DNN che produce E_i nel paragrafo 1.6. Si noti come questa descrizione formale del PES lo renda scalabile con il numero di atomi indipendentemente dalla forma di E_i . Questo assicura che il potenziale sia estensivo, proprietà fondamentale dell'energia in un sistema termodinamico.

1.5 Deep Neural Network

Che cos'è una DNN e come può “imparare” a riprodurre una funzione arbitraria? Una rete neurale non è altro che una composizione di funzioni elementari parametrizzate, che da degli input $\mathbf{d} \in \mathbb{R}^N$ calcolano un certo output⁹ $L(\mathbf{d}) \in \mathbb{R}$. Ogni funzione elementare, detta anche neurone, è definita come:

$$l^i(\mathbf{d}) = \xi(\mathbf{w}_i \mathbf{d} + b_i) . \quad (25)$$

Dove $\mathbf{w}_i \in \mathbb{R}^N$ è un vettore di parametri relativo alla funzione elementare i -esima, $b_i \in \mathbb{R}$ è un altro parametro, detto *bias* o soglia, ξ una funzione continua e derivabile, detta

⁹In generale $L(\mathbf{d}) \in \mathbb{R}^k$ con $k \in \mathbb{N}$, tuttavia per i nostri scopi è comodo considerare fin da subito un output scalare.

funzione di attivazione, tale che $\xi(x) \rightarrow \pm 1$ per $x \rightarrow \pm\infty$ e $\xi(0) = 0$. Generalizzando questo concetto ad output a più dimensioni possiamo definire uno strato (in letteratura *layer*) di M neuroni, detto anche *perceptron*¹⁰ (o più propriamente *simple perceptron*) [7]:

$$\mathcal{L}(\mathbf{d}) = \xi(\mathbf{W}\mathbf{d} + \mathbf{b}) . \quad (26)$$

Dove $\mathbf{W}$ è una matrice di pesi $N \times M$, formata da M vettori $\mathbf{w}_i$, $\mathbf{b}$ è un vettore degli M elementi b_i e $\mathcal{L}(\mathbf{d}) \in \mathbb{R}^N$. A questo punto la composizione di h strati va a creare l'architettura della DNN

$$L(\mathbf{d}) = l^{out}(\mathbf{d}^h) \circ \mathcal{L}^h(\mathbf{d}^{h-1}) \circ \mathcal{L}^{h-1}(\mathbf{d}^{h-2}) \circ \dots \circ \mathcal{L}^1(\mathbf{d}) , \quad (27)$$

$$\mathbf{d}^p = \mathcal{L}^p(\mathbf{d}^{p-1}) = \xi(\mathbf{W}^p \mathbf{d}^{p-1} + \mathbf{b}^p) , \quad (28)$$

dove il simbolo “ $\circ$ ” indica la composizione di funzioni, $\mathbf{d}^p \in \mathbb{R}^{N_p}$ l'output dello strato p -esimo e $\mathbf{W}^p \in \mathbb{R}^{M_p \times M_{p-1}}$, matrice dei pesi del p -esimo strato, con M_p neuroni. L'ultimo strato l^{out} serve a cambiare la dimensione dell'output così da avere $L(\mathbf{d}) \in \mathbb{R}$ (Figura 1). Da notare che, per facilitare la notazione, il *bias* $\mathbf{b}$ può essere incorporato nella matrice $\mathbf{W}$, ridefinendo il vettore degli input come $\mathbf{d} = (\mathbf{d}, 1) \in \mathbb{R}^{N+1}$. In questo modo

$$\mathbf{W}\mathbf{d} + \mathbf{b} = (\mathbf{W}, \mathbf{b}) \cdot (\mathbf{d}, 1) = (\mathbf{W}, \mathbf{d}) . \quad (29)$$

Dove la matrice dei pesi a destra dell'equazione è definita come $\mathbf{W} = (\mathbf{W}, \mathbf{b})$, mentre il vettore degli input è dato dalla nuova definizione precedentemente introdotta. Così facendo il p -esimo strato si può scrivere come:

$$\xi(\mathbf{W}^p \mathbf{d}^{p-1} + \mathbf{b}^p) = (\xi(\mathbf{w}_1^p \mathbf{d}^{p-1}), \xi(\mathbf{w}_2^p \mathbf{d}^{p-1}), \dots, \xi(\mathbf{w}_M^p \mathbf{d}^{p-1})) = \xi(\mathbf{W}^p \mathbf{d}^{p-1}) . \quad (30)$$

Con questa struttura si può dimostrare che esistono valori di W^p , b^p e h per ogni p tali che $L(\mathbf{d})$ approssima a piacere qualunque funzione reale continua definita sullo stesso dominio [7]. Ovviamente il teorema non fornisce un algoritmo per trovare questi valori, pertanto, è qui che entra in gioco il concetto di “apprendimento” di una DNN, il *machine learning*, ovvero degli algoritmi che permettano di trovare i migliori valori per i parametri tali che la rete approssimi una certa funzione, o più correttamente, che minimizzi una specifica funzione, come vedremo nel paragrafo successivo.

Vale la pena aggiungere, che questa architettura di rete è comunemente chiamata in generale *feed-forward*, col nome specifico di DNN¹¹ per $h > 2$. Queste reti sono caratterizzate dall'avere un flusso di informazioni che si muove in un'unica direzione lungo i *layer* della rete, come si vede in figura 1, gli input della rete si muovono dall'*input layer* in verde, ai vari *hidden layers* blu fino all'*output layer* in rosso.

1.5.1 Apprendimento

Come trovare, dunque, i parametri che permettano alla rete di approssimare una certa funzione? Supponiamo di avere una funzione $\mathcal{F}(\mathbf{x})$ e vorremmo che la rete la riproduca,

¹⁰Introdotta per la prima volta nel 1958 dallo psicologo statunitense Frank Rosenblatt

¹¹L'aggettivo *deep*, profondo, sta proprio ad indicare il fatto che l'input si propaga attraverso vari strati di neuroni prima di “riemergere” dall'*output layer*.

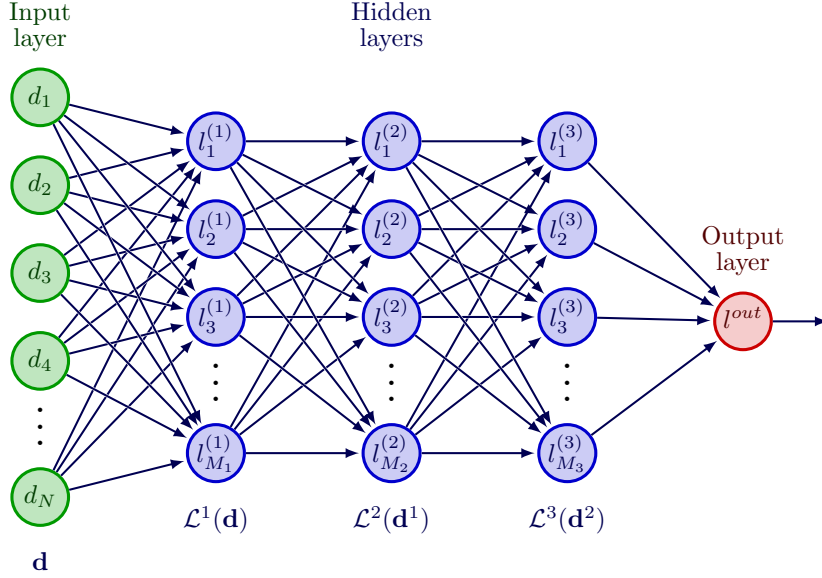


Figura 1: Schema di una DNN con $h = 3$ layers, ogni cerchio rappresenta un neurone, ovvero la funzione (25). In verde il vettore di input $\mathbf{d} \in \mathbb{R}^N$, in blu i vari layers rappresentati dalla funzione (28) di $M_p = M_1, M_2, M_3$ numero di neuroni, in rosso l'ultimo strato che trasforma l'output in un numero reale.

ovvero che per ogni $\mathbf{x} \in \mathbb{R}^N$, $\mathcal{F}(\mathbf{x}) = L(\mathbf{x})$. Supponiamo inoltre di conoscere dei punti di questa funzione, ovvero conoscere per certi $\mathbf{x}_m$, con $m = 1, 2 \dots M$, il valore $\mathcal{F}(\mathbf{x}_m)$. Chiamiamo questi M punti, esempi (*training set*), e costruiamo una funzione costo o *loss function*:

$$C(\mathbf{W}^1, \mathbf{W}^2 \dots \mathbf{W}^h) = \frac{1}{2} \sum_{m=1}^M [\mathcal{F}(\mathbf{x}_m) - L(\mathbf{x}_m, \{\mathbf{W}^1, \mathbf{W}^2 \dots \mathbf{W}^h\})]^2, \quad (31)$$

dove abbiamo esplicitato la dipendenza funzionale di L dalle matrici dei pesi. Questa funzione rappresenta l'errore che la rete produce nell'approssimare la funzione $\mathcal{F}$ sugli esempi dati. Evidentemente, essendo definita positiva, il punto di minimo di questa funzione rappresenta i migliori valori dei parametri della rete per interpolare gli esempi dati e dunque approssimare la funzione $\mathcal{F}$. L'apprendimento della rete neurale si trasforma, dunque, nella ricerca di un minimo per la funzione (31). Essendo C una funzione a più variabili reali, il minimo sarà tale che

$$\nabla_{\mathbf{w}_i^p} C(\mathbf{w}_i^p) \big|_{\mathcal{W}} = 0, \quad (32)$$

dove con $\mathbf{w}_i^p$ indichiamo tutti i vettori delle matrici $\mathbf{W}^1, \mathbf{W}^2, \dots, \mathbf{W}^h$, e con $\mathcal{W}$ indichiamo il punto di minimo della funzione C . Il nucleo dell'apprendimento della rete neurale risiede, dunque, nella risoluzione di questo problema. Un modo tipico di procedere è utilizzare il metodo della discesa del gradiente: i valori dei pesi vengono inizializzati randomicamente (o ad un certo valore scelto a priori), generalmente appartenenti a $[-1, 1]$;

a questo punto per questi valori viene calcolato il gradiente di C e successivamente i pesi vengono aggiornati con la regola:

$$w_{ij}^p(n+1) = w_{ij}^p(n) - \eta \frac{\partial C}{\partial w_{ij}^p} \bigg|_{w_{ij}^p(n)} \quad (33)$$

dove n indica il numero di aggiornamenti già fatti, e η , comunemente detto *learning rate* è un parametro scelto a priori o variabile durante l'apprendimento, che indica quanto veloce possono variare i pesi in uno step di aggiornamento. Iterando questo procedimento si vanno a modificare i pesi verso valori che portano C al minimo. È importante notare che la funzione costo, definita in equazione (56), non è una funzione semplice con un solo minimo ma può presentare numerosi minimi locali, anche molto profondi, dove il gradiente della funzione costo si annulla. Questi minimi locali non rappresentano i pesi ottimali¹², pertanto varie alternative e modifiche sono state apportate alla regola generale (33), così da rendere più efficiente la determinazione del minimo $\mathcal{W}$. In particolare, in questo lavoro verrà utilizzato l'algoritmo per l'ottimizzazione di funzioni ad argomenti stocastici *Adam* [10], del quale discuteremo brevemente nel paragrafo 1.6.3.

1.6 DeePMD

Abbiamo ora tutti gli strumenti per studiare nel dettaglio come è costruita la DNN che dovrà calcolare il PES e come lavora. In particolare per questo lavoro si utilizzerà un pacchetto software chiamato **DeePMD-kit**¹³ [18], basato sul framework per il *deep learning* **TensorFlow**. Vedremo di seguito la struttura teorica del *DeePMD*, come vengono rappresentate le coordinate atomiche, come sono costruite le DNN che da queste ultime calcolano il PES e la particolare famiglia di funzioni costo utilizzata per l'ottimizzazione dei pesi.

1.6.1 Coordinate

Consideriamo un sistema di N atomi, le cui posizioni nello spazio euclideo sono descritte da $\mathbf{r} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\}$. Definiamo la matrice delle coordinate $\mathcal{R} \in \mathbb{R}^{N \times 3}$, la cui i -esima colonna contiene le 3 coordinate cartesiane di $\mathbf{r}_i = (x_i, y_i, z_i)$

$$\mathcal{R} = \begin{bmatrix} x_1 & x_2 & \dots & x_N \\ y_1 & y_2 & \dots & y_N \\ z_1 & z_2 & \dots & z_N \end{bmatrix} \quad (34)$$

Il PES sarà in generale funzione di queste variabili $V(\mathcal{R})$. Poiché quello che si vuole costruire è una DNN che calcola l'energia locale E_i dell' i -esimo atomo, come anticipato nel paragrafo 1.4, formalizziamo la descrizione dell'ambiente locale dell'atomo. Definiamo un raggio di *cutoff* r_c che delimita l'ambiente locale dell'atomo. Passiamo al sistema di

¹²Spesso questi minimi forniscono le comunemente dette soluzioni spurie della rete. Questi pesi possono far produrre alla rete risultati molto diversi da quelli voluti [7].

¹³Il codice è interamente disponibile all'indirizzo <https://github.com/deepmodeling/deepmd-kit>

riferimento centrato nell'atomo i -esimo $(x_{ji}, y_{ji}, z_{ji}) = \mathbf{r}_{ji} = \mathbf{r}_j - \mathbf{r}_i$, in questo modo il set di atomi interni al raggio di *cutoff* dell' i -esimo atomo è $\mathcal{N}_{r_c}(i) = \{k \in [1, N] \mid \|\mathbf{r}_{ki}\| = r_{ki} < r_c\}$ e ne indichiamo la cardinalità con $N_i = |\mathcal{N}_{r_c}(i)|$. Così facendo $j \in \mathcal{N}_{r_c}(i)$, sarà l'indice che identifica il j -esimo atomo vicino al i -esimo atomo. Possiamo ora definire la matrice delle coordinate locali $\mathcal{R}^i \in \mathbb{R}^{N_i \times 3}$:

$$\mathcal{R}^i = \begin{bmatrix} x_{1i} & x_{2i} & \dots & x_{N_i i} \\ y_{1i} & y_{2i} & \dots & y_{N_i i} \\ z_{1i} & z_{2i} & \dots & z_{N_i i} \end{bmatrix} \quad (35)$$

In questo modo, seguendo l'equazione (10), possiamo scrivere

$$V(\mathcal{R}) = \sum_i^N E_i(\mathcal{R}^i) , \quad (36)$$

A questo punto potrebbe sembrare ovvio costruire la DNN in modo che mappi le posizioni atomiche $\mathcal{R}^i$ in $E_i(\mathcal{R}^i)$, tenendo conto della specie chimica dell'atomo i -esimo, ovvero (usando la notazione del paragrafo 1.5)

$$E_i(\mathcal{R}^i) = L^{\alpha_i}(\mathcal{R}^i) , \quad (37)$$

dove con l'apice α_i indichiamo come i pesi della rete dipendano solamente dalla specie chimica dell'atomo i -esimo. Il PES mappato dalla rete sarà allora

$$V(\mathcal{R}) = \sum_i^N L^{\alpha_i}(\mathcal{R}^i) . \quad (38)$$

Seguendo questo approccio pertanto si costruirà una rete per ogni specie chimica del sistema, che mappa ogni posizione atomica, della stessa specie, nell'energia potenziale E_i . Tuttavia, conviene seguire un approccio differente, modificando le coordinate. Trasformiamo le variabili locali $\mathcal{R}^i$ nelle variabili generalizzate $\tilde{\mathcal{R}}^i \in \mathbb{R}^{N_i \times 4}$:

$$\{x_{ji}, y_{ji}, z_{ji}\} \rightarrow \{s(r_{ji}), \tilde{x}_{ji}, \tilde{y}_{ji}, \tilde{z}_{ji}\} \quad (39)$$

dove $s(r_{ji}) : \mathbb{R} \rightarrow \mathbb{R}$ è una funzione continua

$$s(r_{ji}) = \begin{cases} \frac{1}{r_{ji}}, & r_{ji} < r_{cs} \\ \frac{1}{r_{ji}} \left\{ \frac{1}{2} \cos \left[\pi \frac{(r_{ji} - r_{cs})}{(r_c - r_{cs})} \right] + \frac{1}{2} \right\}, & r_{cs} < r_{ji} < r_c \\ 0, & r_{ji} > r_c \end{cases} \quad (40)$$

che mappa la distanza radiale dall'atomo i -esimo. Mentre le altre componenti di $\tilde{\mathcal{R}}^i$ sono date da:

$$\begin{cases} \tilde{x}_{ji} = \frac{s(r_{ji})x_{ji}}{r_{ji}} \\ \tilde{y}_{ji} = \frac{s(r_{ji})y_{ji}}{r_{ji}} \\ \tilde{z}_{ji} = \frac{s(r_{ji})z_{ji}}{r_{ji}} \end{cases} \quad (41)$$

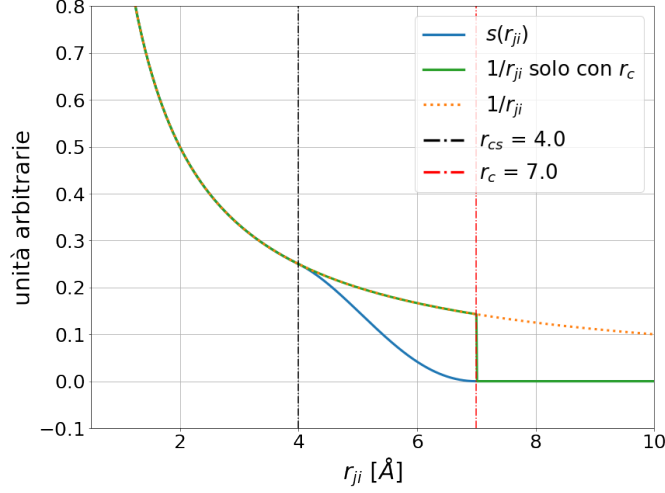


Figura 2: Confronto tra l'andamento di $s(r_{ji})$ (blu) e $1/r_{ji}$ solo con il raggio di *cutoff* (verde). In arancione andamento di $1/r_{ji}$ per riferimento. Si vede come $s(r_{ji})$ va a zero in modo continuo al valore di r_c .

Notiamo che abbiamo preso r_{cs} come parametro per avere uno *smooth cutoff*, permettendo così alle componenti di $\tilde{\mathcal{R}}^i$ di andare in modo continuo a zero a r_c . Queste nuove coordinate, oltre a permettere la scelta tra l'utilizzo di coordinate radiali o coordinate angolari ed aver eliminato la discontinuità a r_c , forniscono anche un peso ($s(r_{ji})$) per il contributo dell'atomo j -esimo all'energia locale E_i . Infatti più l'atomo j -esimo è distante da quello i -esimo, meno contribuirà all'energia locale (figura 2).

Seguendo questo approccio, tuttavia, sorge un problema: E_i è invariante sotto traslazione, permutazione degli atomi con la stessa specie chimica e rotazione, che sono le simmetrie naturali del sistema (già introdotte nel paragrafo 1.2.1). Tuttavia con la definizione (40), le coordinate generalizzate non sono più invarianti sotto queste ultime. Per preservare le simmetrie naturali del sistema la DNN viene costruita attraverso due *sub-networks*. La prima, detta *encoding network*, costruita per trasformare le coordinate locali $\mathcal{R}^i$ in coordinate *ad hoc* ($\mathcal{D}^i$) che preservano le simmetrie traslazionali, rotazionali e le permutazioni del sistema. La seconda, detta *fitting network*, sarà una classica DNN *feed-forward*¹⁴ che mapperà le nuove coordinate $\mathcal{D}^i$ nella funzione E_i [11].

Funzioni che preservano le simmetrie Prima di entrare nei dettagli di come sono costruite le *sub-networks* per E_i , vediamo come costruire una funzione scalare $f(\tilde{\mathcal{R}}^i)$ (come E_i), invariante sotto traslazione, rotazione e permutazione. Innanzitutto, queste trasformazioni agiscono rispettivamente :

$$\hat{T}_t f(\tilde{\mathcal{R}}^i) = f(\tilde{\mathcal{R}}^i + \mathbf{t}), \quad \hat{R}_u f(\tilde{\mathcal{R}}^i) = f(\mathcal{U}\tilde{\mathcal{R}}^i), \quad \hat{P}_\sigma f(\tilde{\mathcal{R}}^i) = f(\tilde{\mathcal{R}}_{\sigma(1)}^i, \tilde{\mathcal{R}}_{\sigma(2)}^i, \dots, \tilde{\mathcal{R}}_{\sigma(N_i)}^i) \quad (42)$$

¹⁴In realtà la rete avrà anche dei “salti” tra *layers*, ovvero la possibilità che l'output di un strato di neuroni (p -esimo) non vada a quello successivo ma direttamente allo strato ($p+2$)-esimo. Questa più complessa architettura forma la cosiddetta *Residual Neural Network* (ResNet) [11].

Dove $\mathbf{t} \in \mathbb{R}^3$ rappresenta un vettore di traslazione generico, $\mathcal{U} \in \mathbb{R}^{3 \times 3}$ una matrice ortogonale di rotazione e σ una arbitraria permutazione del set di indici. Quello che si vuole ottenere è dunque una struttura di $\tilde{\mathcal{R}}^i$ che renda f invariante sotto queste trasformazioni.

Per fare questo notiamo che si può costruire la matrice simmetrica $\Omega^i \in \mathbb{R}^{N_i \times N_i}$

$$\Omega^i = \tilde{\mathcal{R}}^i \left(\tilde{\mathcal{R}}^i \right)^T, \quad (43)$$

che sarà tale $\hat{T}_t \Omega^i = \Omega^i$ e $\hat{R}_{\mathcal{U}} \Omega^i = \Omega^i$. In questo modo si preserva completamente l'informazione sulle coordinate locali dell'atomo i -esimo, permettendo la costruzione di una funzione di $\tilde{\mathcal{R}}^i$ simmetrica per traslazioni e rotazioni, ovvero $f(\Omega^i) = f(\Omega^i(\tilde{\mathcal{R}}^i))$. Tuttavia questa matrice simmetrica inverte le colonne con le righe sotto permutazione.

Per ottenere una $f(\tilde{\mathcal{R}}^i)$ invariante sotto permutazioni, si può usare il teorema riportato in [11], il quale afferma che ogni funzione invariante sotto permutazione (42), può essere rappresentata come

$$f(\tilde{\mathcal{R}}^i) = \rho \left(\sum_j \phi(\tilde{\mathcal{R}}_j^i) \right), \quad (44)$$

dove ϕ , funzione multi-dimensionale e ρ , scalare, sono funzioni opportunamente scelte. Vedremo successivamente come queste affermazioni ci saranno utili a creare una rete che preserva le simmetrie.

1.6.2 Sub-Networks

Ispirati dalle considerazioni del paragrafo precedente, creiamo una *deep neural network* $G^{\alpha_i \alpha_j}(s(r_{ji}))$ che mappa il singolo valore $s(r_{ji})$ in un vettore di dimensione $(\mathcal{G}^i)_j \in \mathbb{R}^{M_1}$, (Figura 3) i cui elementi sono

$$(\mathcal{G}^i)_{jk} = (G(s(r_{ji})))_k \quad (45)$$

dove abbiamo tolto gli apici α per facilità di notazione. Da notare che questi indici indicano la dipendenza dei parametri della rete dalla specie chimica del atomo i -esimo e del suo vicino j -esimo. Per cui, immaginando di avere k specie chimiche si costruiscono $k^2 - k$ reti G . Per ogni atomo i -esimo si è definita, dunque, una matrice $\mathcal{G}^i \in \mathbb{R}^{N_i \times M_1}$. A questo punto si osserva che si può porre

$$\sum_j^{N_i} \phi(\tilde{\mathcal{R}}_j^i) = \sum_j^{N_i} (\mathcal{G}^i)_{kj} \tilde{\mathcal{R}}_{jl}^i = \left[(\mathcal{G}^i)^T \tilde{\mathcal{R}}^i \right]_{kl}. \quad (46)$$

con l indice degli elementi $\{s(r_{ji}), \hat{x}_{ji}, \hat{y}_{ji}, \hat{z}_{ji}\}$, e $\left[(\mathcal{G}^i)^T \tilde{\mathcal{R}}^i \right]_{kl}$ elemento di $(\mathcal{G}^i)^T \tilde{\mathcal{R}}^i \in \mathbb{R}^{M_1 \times 4}$. Pertanto grazie all'equazione (44), la funzione arbitraria

$$\rho \left((\mathcal{G}^i)^T \tilde{\mathcal{R}}^i \right), \quad (47)$$

funzione di ogni elemento della (46), è invariante sotto permutazione degli N_i atomi. Volendo rendere la funzione (47) invariante anche sotto rotazione e traslazione, possiamo

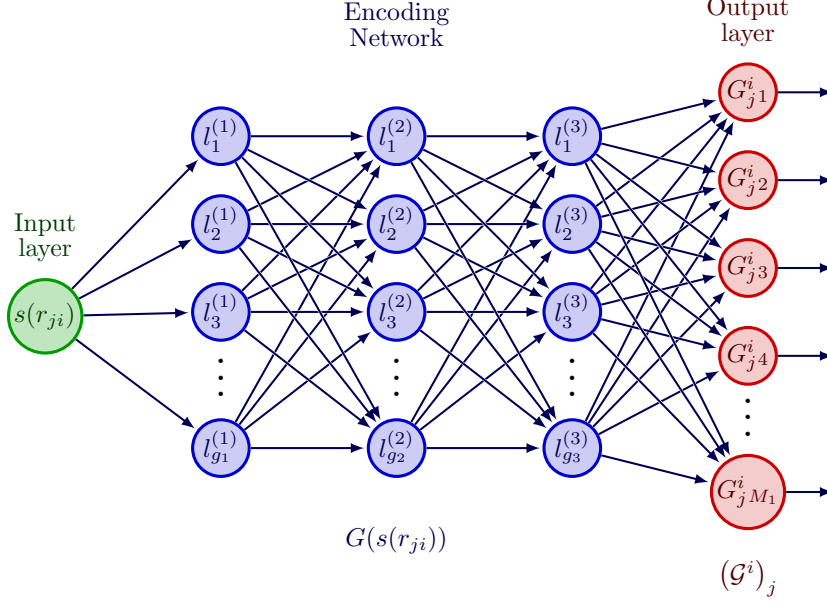


Figura 3: Schema esempio della sub-network $G^{\alpha_i \alpha_j}(s(r_{ji}))$. In questo caso si hanno 3 layers di dimensioni g_1, g_2, g_3 , che mappano un valore scalare in un array di M_1 componenti

ragionare in questo modo: poiché le matrici del tipo introdotto in equazione (45), sono per costruzione invarianti sotto rotazione e traslazione, vorremmo inserire all'interno dell'equazione (46) la matrice Ω^i al posto della dipendenza diretta da $\tilde{\mathcal{R}}^i$. Definendo due matrici $\mathcal{G}^{i1}$ e $\mathcal{G}^{i2}$ della forma introdotta in equazione (45), rispettivamente di M_1 e M_2 colonne, possiamo definire

$$\phi_{km}(\tilde{\mathcal{R}}_j^i) = \sum_l^4 (\mathcal{G}^{i1})_{kj} \tilde{\mathcal{R}}_{jl}^i \tilde{\mathcal{R}}_{lj}^i (\mathcal{G}^{i2})_{jm} \quad (48)$$

così che

$$\sum_j^{N_i} \phi_{km}(\tilde{\mathcal{R}}_j^i) = \sum_j^{N_i} \sum_l^4 (\mathcal{G}^{i1})_{kj} \tilde{\mathcal{R}}_{jl}^i \tilde{\mathcal{R}}_{lj}^i (\mathcal{G}^{i1})_{jm} \equiv \mathcal{D}_{km}^i \quad (49)$$

in questo modo abbiamo definito una matrice $\mathcal{D}^i \in \mathbb{R}^{M_1 \times M_2}$, invariante sotto rotazioni e traslazioni grazie alla comparsa della matrice (43)

$$\mathcal{D}^i = (\mathcal{G}^{i1})^T \tilde{\mathcal{R}}^i (\tilde{\mathcal{R}}^i)^T \mathcal{G}^{i2} = (\mathcal{G}^{i1})^T \Omega^i \mathcal{G}^{i2} \quad (50)$$

Così facendo si ottengono le variabili $\mathcal{D}^i$ che permettono di costruire una funzione $\rho(\mathcal{D}^i)$ invariante sia sotto permutazione degli elementi $\tilde{\mathcal{R}}_j^i$ che sotto traslazione e rotazione. Si costruisce adesso la *fitting network* (L) che mappa le nuove coordinate in E_i , ovvero

$$E_i(\tilde{\mathcal{R}}^i) = L^{\alpha_i}(\mathcal{D}^i) = L^{\alpha_i} \left(\sum_j^{N_i} \phi_{km}(\tilde{\mathcal{R}}_j^i) \right) \quad (51)$$

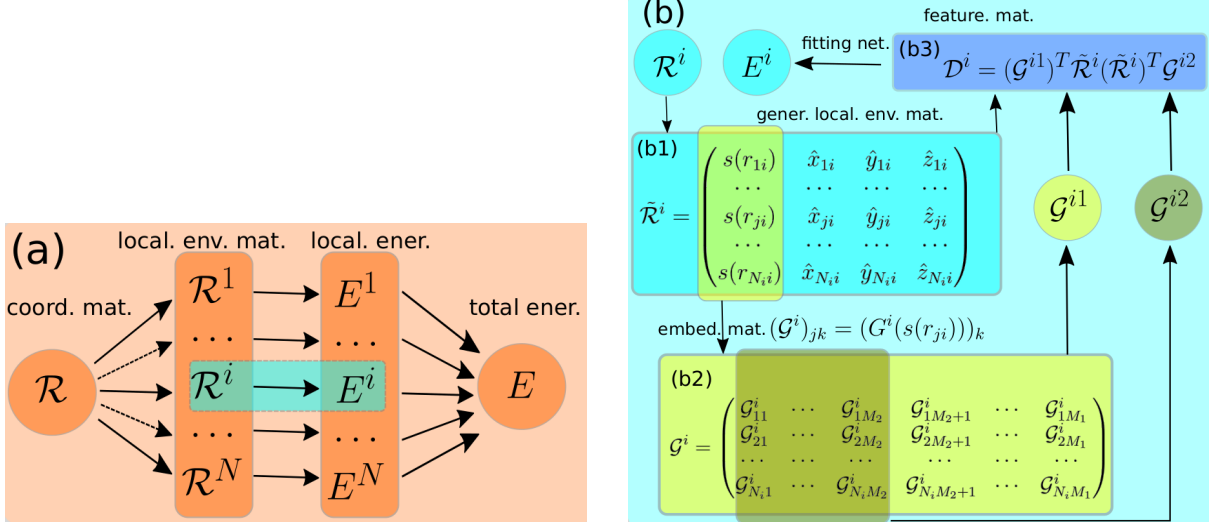


Figura 4: (a) Mappa delle coordinate $\mathcal{R}$ al PES E : prima si trasformano in coordinate locali $\mathcal{R}^i$, successivamente vengono mappate dalle *sub-networks* nell'energia atomica E_i , infine si ottiene $E = \sum_i E_i$. (b) Schema dell'*encoding network*: (b1) le coordinate locali $\mathcal{R}^i$ vengono trasformate in quelle generalizzate $\tilde{\mathcal{R}}^i$; (b2) la parte radiale di $\tilde{\mathcal{R}}^i$ viene mappata attraverso l'*encoding network* alle matrici $\mathcal{G}^{i1}$ e $\mathcal{G}^{i2}$; (b3) le nuove coordinate $\mathcal{D}^i$ sono ottenute dal prodotto tra $\mathcal{G}^{i1}$, $\mathcal{G}^{i2}$ e $\tilde{\mathcal{R}}^i$. (Figure adattate da [11])

dove riconosciamo la forma dell'equazione (44). L'*encoding network* è dunque costituita da una rete *feed-forward* $G^{\alpha_i \alpha_j}$, il cui *output layer* ha dimensione M_1 e M_2 (figura 3 e 4), mentre la *fitting network* sarà una DNN del tipo (27), dove ogni *layer* p -esimo $\mathcal{L}_p^{\alpha_i}$, della forma (28), mappa

$$\mathbf{d}_p^i = \mathcal{L}_p^{\alpha_i}(\mathbf{d}_{p-1}^i) = \tanh(\mathbf{W}_{\alpha_i}^p \mathbf{d}_{p-1}^i + \mathbf{b}_{\alpha_i}^p), \quad (52)$$

dove con $\mathbf{d}_p^i$ indichiamo l'output del p -esimo *layer* relativo all'atomo i -esimo e con $\mathbf{W}_{\alpha_i}^p$, $\mathbf{b}_{\alpha_i}^p$ la matrice dei pesi e la soglia del *layer* p -esimo, dove abbiamo esplicitato la dipendenza dalla specie chimica dell'atomo i -esimo con il pedice α_i . Abbiamo posto la tangente iperbolica come funzione di attivazione $\xi = \tanh$. Da notare che l'ultimo layer $\mathcal{L}_{out}^{\alpha_i}$ è infine una semplice trasformazione lineare

$$E_i = \mathcal{L}_{out}^{\alpha_i}(\mathbf{d}_h^i) = \mathbf{W}_{\alpha_i}^{out} \mathbf{d}_h^i + b_{\alpha_i}^{out}. \quad (53)$$

Dunque, quello che si ottiene è una DNN che può essere vista come due reti distinte, *fitting* e *encoding network*. Quest'ultima serve solamente alla trasformazione delle coordinate cartesiane "canoniche" degli atomi dell'ambiente locale dell'atomo i -esimo, in coordinate ottimali ($\mathcal{D}^i$) per la descrizione dell'ambiente locale. Questo avviene attraverso una trasformazione non analitica e univoca ma ottimizzata dall'apprendimento della rete. Qualunque sia la forma analitica della trasformazione $\tilde{\mathcal{R}}^i \rightarrow \mathcal{D}^i$, grazie alla struttura di $\mathcal{D}^i$, $L^{\alpha_i}(\mathcal{D}^i)$ risulta sempre rispettare le simmetrie del sistema già introdotte. Pertanto a noi non interessa che forma ha questa trasformazione fintanto che la rete riproduce l'energia locale adeguatamente.

Da notare che ottenuto il PES, si possono ricavare, direttamente da quest'ultimo, la forza totale su ogni atomo e il viriale. Possiamo definire, infatti, una matrice forza $\mathcal{F} \in \mathbb{R}^{N \times 3}$ e il tensore 3×3 del viriale:

$$\mathcal{F}_{ij}(\mathcal{R}) = -\nabla_{\mathcal{R}_{ij}} V, \quad \Xi_{ij}(\mathcal{R}) = \sum_k^N \mathcal{R}_{ki} \mathcal{F}_{kj}. \quad (54)$$

dove $\mathcal{F}_{ij}$ è la componente j -esima della forza totale agente sull'atomo i -esimo. Indicando, con un abuso di notazione, $L(\tilde{\mathcal{R}}^k)$, la rete totale (*encoding* più *fitting*) che mappa le coordinate $\tilde{\mathcal{R}}^k$ locali dell'atomo k -esimo nell'energia potenziale locale, l'elemento $\mathcal{F}_{ij}$ è esplicitamente dato da:

$$\mathcal{F}_{ij}(\mathcal{R}) = -\frac{\partial V(\mathcal{R})}{\partial \mathcal{R}_{ij}} = -\sum_k^N \frac{\partial L(\tilde{\mathcal{R}}^k)}{\partial \mathcal{R}_{ij}} = -\sum_k^N \sum_l^4 \sum_{\beta \in N_{rc}(k)} \frac{\partial L}{\tilde{\mathcal{R}}_{\beta l}^k} \frac{\tilde{\mathcal{R}}_{\beta l}^k}{\partial \mathcal{R}_{ij}}, \quad (55)$$

con $\mathcal{R}_{ij}$ componente cartesiana j -esima della posizione dell'atomo i -esimo, elemento della matrice (34). Usando l'equazione (55) nella (54), si ricava direttamente il viriale. In questo modo si ottengono grandezze fisiche essenziali per la MD, in particolare la forza, necessaria per la soluzione numerica dell'equazione (1). In oltre la matrice delle forze sarà di particolare interesse in questo lavoro, come mezzo per il controllo della precisione della rete.

Da notare, infine, che la rete così costruita non dipende dal numero totale di atomi del sistema. La DNN non ha nessun parametro dipendente dal numero di atomi del sistema usato per l'apprendimento, essa infatti mappa un certo numero di posizioni atomiche locali N_i nell'energia potenziale E_i , indipendentemente dal numero totale di atomi del sistema. Pertanto sia l'apprendimento che l'uso effettivo della rete può essere effettuato su sistemi con diverse quantità di atomi. Il PES riprodotto dalla rete è dunque trasferibile a sistemi di dimensioni diverse, il che permette di effettuare l'apprendimento su sistemi contenuti, più facilmente simulabili con MD da principi primi.

1.6.3 Training

Per ottimizzare sia i parametri della *fitting network*, che i parametri delle *encoding network*, che indichiamo globalmente con ω , viene costruita una funzione costo del tipo introdotto con l'equazione (31), utilizzando le tre funzioni (energia, forza e viriale) calcolate dalla rete

$$\mathcal{C}(p_e, p_f, p_\xi) = \frac{1}{|\mathcal{B}|} \sum_{l \in |\mathcal{B}|} p_e |E_l - E_l^\omega|^2 + \frac{p_f}{3} |\mathcal{F}_l - \mathcal{F}_l^\omega|^2 + \frac{p_\xi}{9} \|\Xi_l - \Xi_l^\omega\|^2. \quad (56)$$

Dove $E_l - E_l^\omega$, $\mathcal{F}_l - \mathcal{F}_l^\omega$, $\Xi_l - \Xi_l^\omega$ sono le differenze tra i valori di energia, forza e viriale del *training set* e quelli calcolati dalla rete, dipendenti dai parametri ω . $\mathcal{B}$ e $|\mathcal{B}|$ indicano la *minibatch* e la sua dimensione, ovvero, un insieme di esempi scelti tra tutti quelli del *training set*. Il set consiste in istantanee, chiamate *frames*, delle configurazioni atomiche (date dalle posizioni atomiche, le specie chimiche e le dimensioni della cella) e delle funzioni forza, energia e viriale. Ci riferiremo in seguito all'insieme dei *frames* provenienti da

simulazioni di MD con il nome di traiettorie. I fattori p_ϵ , p_f , p_ξ dipendenti dal *learning rate* della rete, indicano il contributo di ogni componente alla funzione costo. Essi sono liberi di cambiare durante la minimizzazione di $\mathcal{C}$ secondo la

$$p(t) = p^{limit} \left[1 - \frac{\eta(t)}{\eta(0)} \right] + p^{start} \left[\frac{\eta(t)}{\eta(0)} \right], \quad (57)$$

dove $\eta(t)$ è il *learning rate* allo step t e $\eta(0)$ all'inizio dell'apprendimento. I fattori p variano da un valore p^{start} iniziale a un valore p^{limit} finale, seguendo il decadimento esponenziale dell'*learning rate*

$$\eta(t) = \eta(0) d_r^{\frac{t}{d_s}}, \quad (58)$$

con d_r e d_s , *decay rete*¹⁵ e *decay steps* rispettivamente, iperparametri della rete.

Come anticipato nel paragrafo 1.5.1, DeepMD-kit utilizza l'algoritmo per la minimizzazione di funzioni stocastiche *Adam* (*adaptive moment estimator*) per minimizzare la funzione costo (56) [10, 11]. L'uso di questo algoritmo è dettato dal carattere stocastico della funzione costo, dato dalla valutazione di $\mathcal{C}$, per ogni step di apprendimento, su una *minibatch*, scelta in modo casuale. Pertanto la funzione costo, valutata su tutto il *training set*, rappresenta il valore medio intorno al quale sono distribuite le funzioni costo valutate sulle varie *minibatch*. Ovvero, indicando sempre con w i pesi della rete e $\mathcal{C}_t(w)$ la funzione costo valutata al t -esimo step (valutata su una certa *minibatch*), si avrà che $\mathcal{C}_t(w)$ saranno distribuite intorno al valore medio $\mathbb{E}[\mathcal{C}(w)]$. L'algoritmo *Adam*, sempre basato sulla discesa del gradiente, si propone, allora, come metodo efficiente per la minimizzazione del valore di aspettazione $\mathbb{E}[\mathcal{C}(w)]$, aiutando a superare i minimi locali e ad evitare l'*overfitting* [10].

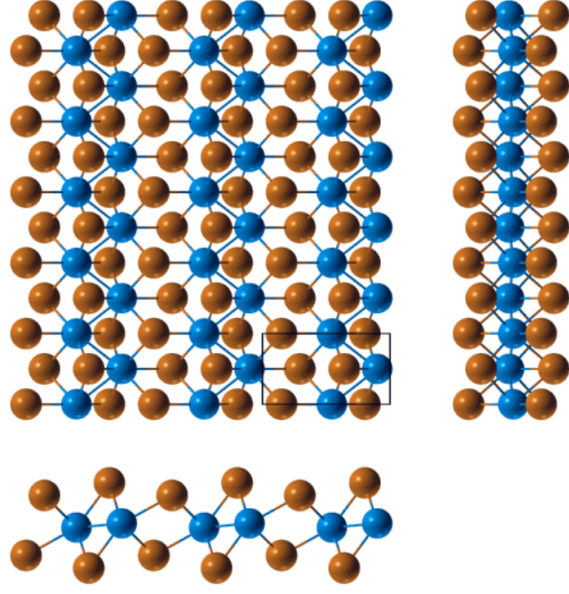
2 Risultati

In questo capitolo, applicheremo il metodo DeePMD, a un sistema fisico reale: il **ditellurio di tungsteno** (WTe₂). Spiegheremo l'interesse fisico nello studio di questo specifico materiale, come si è prodotto il set di traiettorie da principi primi utilizzato per l'apprendimento della rete neurale e poi, nel dettaglio l'apprendimento di quest'ultima per il WTe₂. Successivamente si discuterà come testare l'efficacia del potenziale DeePMD prodotto, e i problemi associati alla stima dell'errore commesso dalla rete nella predizione del PES. Infine si utilizzerà il potenziale ottenuto per la MD.

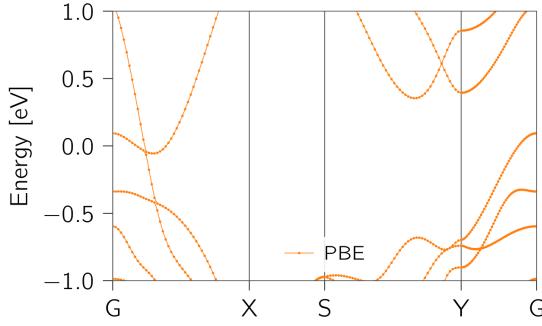
2.1 Il ditellurio di tungsteno

Il ditellurio di tungsteno è un composto chimico facente parte della famiglia dei dicalcogenuri dei metalli di transizione (TMDC dall'inglese: *transition metal dichalcogenide*). Questi materiali sono caratterizzati dall'avere una struttura cristallina a strati, dallo spessore di qualche atomo (figura 5a per il WTe₂). Come nella grafite, gli atomi nello stesso strato sviluppano legami covalenti mentre gli atomi tra gli strati interagiscono attraverso

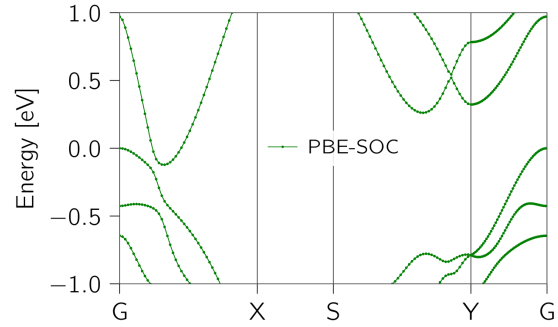
¹⁵È ovviamente richiesto che $d_r < 1$



(a) Proiezione ortogonale della struttura cristallina del WTe_2



(b) Struttura a bande senza effetto spin-orbita



(c) Struttura a bande con effetto spin-orbita

Figura 5: (a) Proiezione ortogonale della struttura cristallina del WTe_2 . Gli atomi marroni sono Te mentre i blu W. (figura adattata da [12])

interazioni di van der Waals. Grazie a questa struttura gli strati di questi materiali possono essere separati (sfogliazione). Pertanto è possibile studiare le proprietà fisiche dei singoli strati, che in generale sono diverse da quelle del materiale originario. Per questo motivo, in questo lavoro, si è studiato un sistema di WTe_2 a monostrato, dunque, con geometria dei legami atomici che si sviluppa in 2 dimensioni. In particolare il WTe_2 studiato, sviluppa una struttura chiamata $1\text{T}'$, che consiste in una catena monodimensionale a zigzag lungo l'asse y [14] (Figura 5a). L'interesse nello studio di questo materiale deriva proprio dal fatto che il WTe_2 con questa struttura risulta essere un isolante topologico (in 2D) [14], ovvero un materiale la cui struttura a bande si presenta come una normale struttura a bande di un isolante, ovvero con un *gap* tra la banda di valenza e quella di conduzione, con l'aggiunta però di *edge states*, causati dall'interazione spin-orbita degli elettroni, che permettono la conduzione (di tipo metallico) sul bordo. Il WTe_2 è dunque

caratterizzato da una complessa struttura elettronica, il cui comportamento in funzione della temperatura tutt'ora è soggetto al dibattito. Proprio per questo motivo, studiare il sistema attraverso la MD ad alte temperature risulta di forte interesse, in particolare nel verificare la presenza di effetti anarmonici.

2.2 Creazione del data set e training

Per l'apprendimento del potenziale DeePMD sono state usate traiettorie¹⁶ prodotte con BOMD attraverso DFT, utilizzando il software **Quantum ESPRESSO** [5]. Il sistema simulato è formato da 216 atomi che compongono un monostrato di WTe₂ in geometria 1T'. Le traiettorie sono state calcolate nell'ensemble canonico (NVT) dunque a temperatura e volume costante, a 200 K, 300 K e 500 K, e nel ensemble isobarico (NPT) tenendo, dunque, costante la pressione (a 0 bar), per temperature sui 1000 K (tabella 1). L'insieme di queste traiettorie forma il *data set* utilizzato per l'apprendimento e il test della rete. Proprio per questo motivo le traiettorie vengono divise in 3 sottoinsiemi: *training*, *validation* e *test*. Il *training set*, composto dal 70 % di ogni traiettoria, è utilizzato per l'apprendimento della DNN; il *validation set*, composto solo dal 10 %, è utilizzato per un test in tempo reale della rete, durante l'apprendimento, che fornisce una stima di come la rete si comporta su un set di dati non utilizzati per l'apprendimento. Il *test set*, composto dal 20 % dei frames di ogni traiettoria, è utilizzato per il test della rete una volta completato l'apprendimento. Prima di poter far apprendere la rete su questi dati, tuttavia,

ensemble	temperatura [K]	lunghezza [ps]	numero time-step
NVT	200	2.347	970
NVT	300	1.836	759
NVT	500	2.318	958
NPT	1000	1.040	430

Tabella 1: Tabella riassuntiva delle traiettorie BOMD utilizzate per il training e test iniziale della DNN. Ogni time-step corrisponde a 2.42×10^{-3} ps

è necessario definire¹⁷ alcuni iperparametri. Vediamo qui i più importanti, dividendoli in: parametri dell'*encoding network*, parametri della *fitting network* e parametri globali. I parametri più significativi dell'*encoding network*, riportati sotto **descriptor**, e della *fitting network*, sotto **fitting_net**, sono riportati in seguito:

```
"descriptor" :{
  "sel":          [100, 100],
  "rcut_smth":    6.0,
  "rcut":          7.00,
  "neuron":       [32,64,128],
}
"fitting_net" : {
```

¹⁶Specifichiamo che le traiettorie utilizzate non sono state calcolate da me ma erano preesistenti.

¹⁷tutti i parametri necessari all'apprendimento della rete sono specificati in un file di testo .json

```

"activation_function" :    "tanh",
"neuron":                [256, 256, 256],
    }

```

Il parametro `sel` definisce il numero massimo di ogni specie chimica (W, Te) all'interno del raggio di cutoff; `rcut` e `rcut_smth` definiscono r_c e r_{cs} rispettivamente, in unità di ångström; `neuron` definisce il numero di neuroni in ogni *layer*, dunque anche il numero di *layers*, di entrambe le reti; `activation_function` definisce la funzione di attivazione definita in (52). I parametri globali significativi sono riportati qui di seguito:

```

"start_pref_e": 0.001,
"limit_pref_e": 1,
"start_pref_f": 1000,
"limit_pref_f": 1,
"start_pref_v": 0,
"limit_pref_v": 0,
"numb_steps": 200000,
"batch_size": 1,

```

I parametri che iniziano per `start_pref_` e `limit_pref_` rappresentano p^{start} e p^{limit} definiti in equazione (57), per ogni grandezza presente nella funzione costo. `numb_steps` indica il numero di step di apprendimento, mentre `batch_size` il valore di $|\mathcal{B}|$ definito in equazione (56). I valori di questi parametri sono stati presi da [9], data l'efficienza riscontrata nell'apprendimento della rete.

Con questi parametri e con il *training set* definito precedentemente, si ha la convergenza della funzione costo durante l'apprendimento, come si può vedere nella parte superiore della figura 6, dopo circa 1.5×10^5 step. Inoltre, anche la funzione costo calcolata sul *validation set* converge, nello stesso modo della funzione costo calcolata sul *trainig set*. Questo implica, non solo che la rete ha interpolato il PES sui dati del *trainig set*, ma anche che questa interpolazione è corretta per configurazioni nello spazio delle fasi del sistema (*frames* delle traiettorie) che la rete non ha mai usato per l'apprendimento: ovvero non c'è *overfitting*. La funzione costo, per come è definita in (56), è la somma dell'errore quadratico medio (RSME) tra i valori predetti dalla rete e quelli del *data set* (dell'energia, della forza e del viriale), moltiplicati per i fattori p_e , p_f , p_ξ . Non avendo utilizzato il viriale, nel nostro caso, la funzione costo risulta solo la somma dell'RSME dell'energia e della forza moltiplicati per i corrispettivi fattori. Nella parte sotto della figura 6, si vede come si comporta l'RSME, calcolato sia sul *training set* che sul *validation set*, di queste due grandezze. L'errore sulla forza converge quasi subito, rimanendo costante per la maggior parte dell'apprendimento, mentre l'errore sull'energia converge più lentamente analogamente alla funzione costo. Si nota infine, come era prevedibile, che l'errore si comporta allo stesso modo sia se calcolato sul *training set* che sul *validation set*.

2.3 Test del potenziale

A questo punto, ottenuto il potenziale DeePMD del sistema, si può utilizzare il *test set* per verificare l'accuratezza di quest'ultimo. Per ogni l -esimo frame del *test set*, si ha un

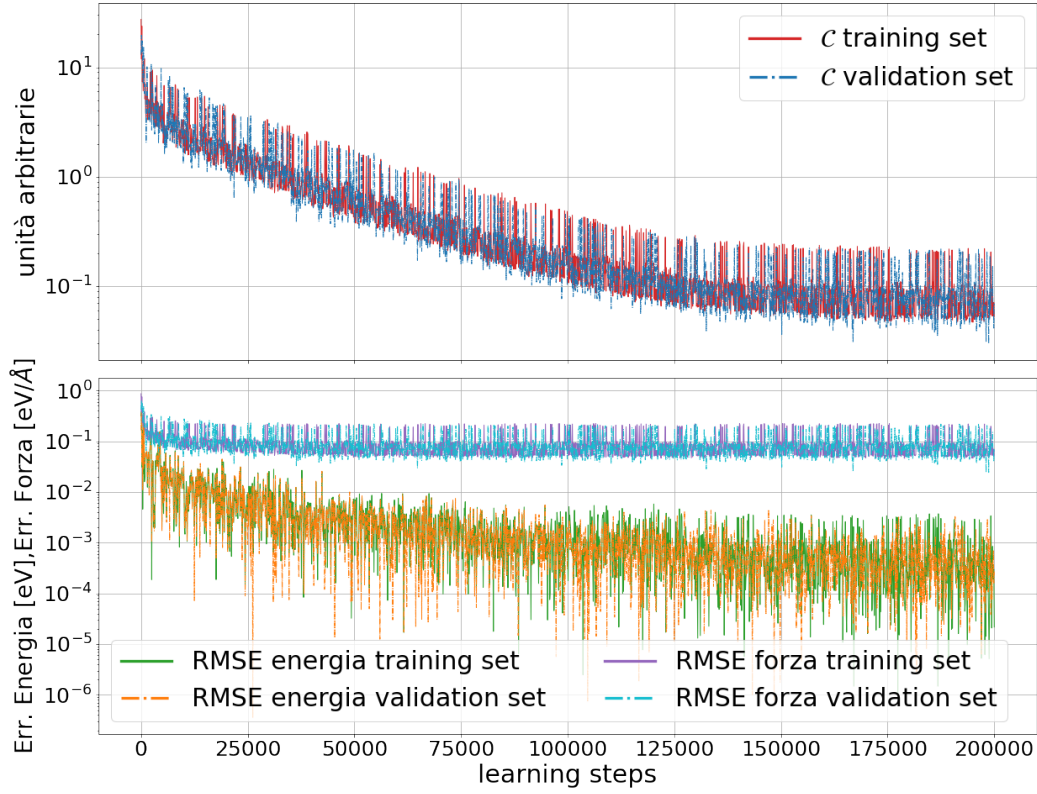


Figura 6: Sopra: La funzione costo $\mathcal{C}$ della rete calcolata durante l'apprendimento, per il *trainig set* (rosso) e per il *validation set* (blu), si vede che entrambe le funzioni si comportano allo stesso modo. Sotto: errore quadratico medio (RMSE) dell'energia e della forza rispetto al *trainig set* (verde e viola) e rispetto il *validation set* (arancione e azzurro)

valore del PES E_l e un valore E_l^w prodotto dalla rete utilizzando l'insieme delle coordinate atomiche $\mathcal{R}_l$, contenute nell' l -esimo frame. Dunque per ogni configurazione atomica $\mathcal{R}_l$ contenuta *test set* possiamo confrontare il PES calcolato con la DFT $V(\mathcal{R}_l) = E_l$ con quello ottenuto con DeePMD (E_l^w). Graficando le coppie (E_l, E_l^w) si ottiene la figura 7a. In linea di principio, se la rete predice perfettamente l'energia, avremo $E_l = E_l^w$, che rappresenta la bisettrice nera in figura, indicando un errore nullo del PES DeePMD prodotto. Come si vede dalla figura 7a, i punti (E_l, E_l^w) , non si discostano troppo dalla bisettrice per energie relative a temperature fino ai 500 K, mentre per 1000 K è evidente un aumento dell'errore. Inoltre, si vede chiaramente come l'energia del sistema aumenti all'aumentare della temperatura e come la distribuzione dei punti lungo la bisettrice sia sempre meno densa all'aumentare della temperatura, indicando un aumento delle energie accessibili al sistema. Un grafico del genere risulta ancora più significativo se si utilizzano

le forze invece che l'energia. Andando a confrontare le componenti delle forze di ogni atomo ottenute dal PES predetto dalla rete con quelle date dal *test set*, per temperature sui 300 K, 500 K e 1000 K, si ottiene il grafico in figura 7b, dove ogni punto rappresenta la coppia (F_j^i, F_j^{iw}) . F_j^i rappresenta la componente cartesiana $j = x, y, z$ dell'atomo i -esimo del *test set* (calcolata con BOMD) mentre F_j^{iw} quella ottenuta dal potenziale calcolato dalla rete. Come si vede nella figura 7 la precisione del PES diminuisce al crescere della temperatura, sia nel predire le forze che le energie, l'errore quadratico medio sulle forze riportato in figura 7b, aumenta, infatti, di un ordine di grandezza tra il test a 300 K e 1000 K. Questo comportamento si può razionalizzare pensando che all'aumentare della temperatura l'ensemble esplora uno spazio delle fasi sempre più grande, che non è più ben rappresentato dalle corte traiettorie usate per l'apprendimento. Ricordiamo, inoltre, che il potenziale a cui è soggetto un singolo atomo (allo stato solido cristallino) può essere approssimato a un potenziale parabolico, per basse temperature del sistema [6]. Questo porta alla descrizione armonica del comportamento degli atomi e ai così detti fononi del materiale. Tuttavia, all'aumentare della temperatura quest'approssimazione viene meno e la dinamica atomica non può più essere descritta attraverso semplici oscillatori armonici. Il PES che descrive il sistema a temperature più alte è, dunque, molto più complesso di quello a basse temperature, pertanto è logico aspettarsi, come si vede in figura 7, una maggiore difficoltà della DNN nel riprodurre quest'ultimo potenziale.

2.4 Correlazione tra frames

Un osservazione importante che si può fare ora è notare che il test del potenziale nel paragrafo precedente è stato fatto grazie *test set*, ottenuto dalle traiettorie calcolate con BOMD. I *frames* che formano queste traiettorie atomiche, derivando da soluzioni della hamiltoniana elettronica tramite DFT, risultano fortemente correlati tra loro. Per quantificare questa correlazione possiamo calcolare la funzione di autocorrelazione delle forze (FACF), definita come

$$\rho(k) = \frac{1}{(N_f - k)\sigma^2} \sum_t^{N_f - k} \frac{1}{N} \sum_i^N \mathbf{F}_i(t) \mathbf{F}_i(t + k) , \quad (59)$$

dove la somma più interna somma su tutti gli atomi N mentre quella più esterna su tutti gli $N_f - k$ *frames* con N_f *frames* totali e $\mathbf{F}_i(t)$ è la forza nell'atomo i -esimo nel *frame* t -esimo. Questa funzione fornisce il coefficiente di correlazione (o coefficiente di Pearson) $-1 \leq \rho(k) \leq 1$ delle forze distanti k *frames*. Come si vede dalla figura 8, la correlazione fatica ad andare a zero, per cui risulta che i punti nello spazio delle fasi rappresentati dai *frames* non sono indipendenti gli uni dagli altri. Come si vede, sempre dalla figura 8, anche le forze nei *frames* distanti 100 *time steps* risultano correlate. Questo fatto mette in dubbio la validità del test sul potenziale, poiché non si è utilizzato un *test set* che la rete non ha mai usato, bensì di un set di *frames* fortemente dipendente da quelli utilizzati per l'apprendimento del potenziale.

Inoltre, questa dipendenza tra *frames* fa supporre che non tutte le configurazioni atomiche, le energie e le forze, calcolate tramite BOMD, che la rete usa per apprendere, portino nuova informazione alla rete, ovvero contribuiscano ad un effettivo miglioramento

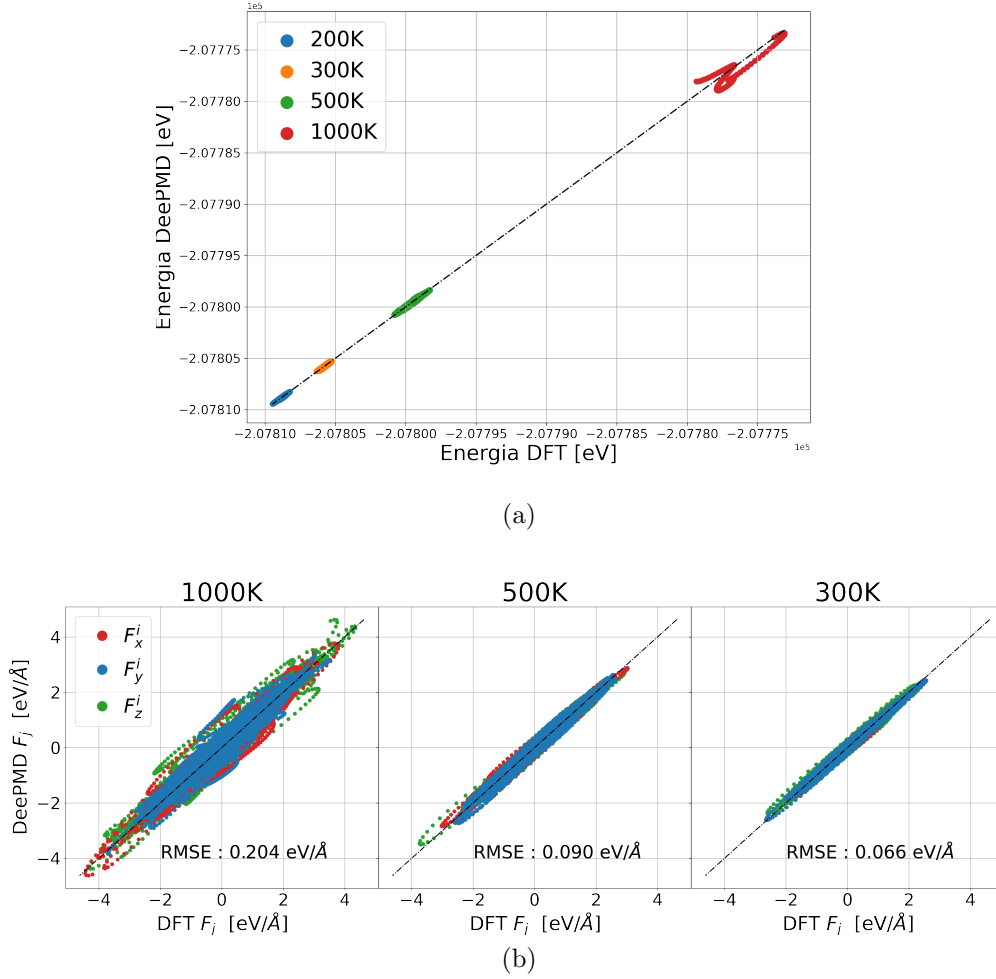


Figura 7: (a) Confronto tra le energie predette dalla rete e del *test set* per varie temperature. (b) Confronto tra le forze predette dalla rete e del *test set* per varie temperature. Sulle ascisse e ordinate si hanno le componenti cartesiane della forza (F_j) del *test set* (DFT F_j) e prodotte dal potenziale DeePMD (DeePMD F_j). Ogni punto nel grafico rappresenta una componente della forza di un atomo del sistema, in rosso le forze lungo x , in blu lungo y e in verde lungo z .

dell'efficienza della DNN. In altre parole, lo spazio delle fasi esplorato dal sistema dalle traiettorie calcolate, è una piccola parte dello spazio effettivo, e non rappresenta un buon campione di quest'ultimo per l'apprendimento della rete neurale. Per mostrarlo esplicitamente si può pensare di costruire dei nuovi *training set*, costituiti considerando solo un *frame* ogni n *frames* del *training set* iniziale. Abbiamo costruito tre *training set* formati da *frames* distanti 10, 20, 110 e 210 time-step, che chiameremo *training set* 10, 20, 110, 210. Dunque costituiti da *frames* sempre più distanti l'uno dall'altro e pertanto sempre meno correlati tra loro. Con questi set abbiamo prodotto quattro reti

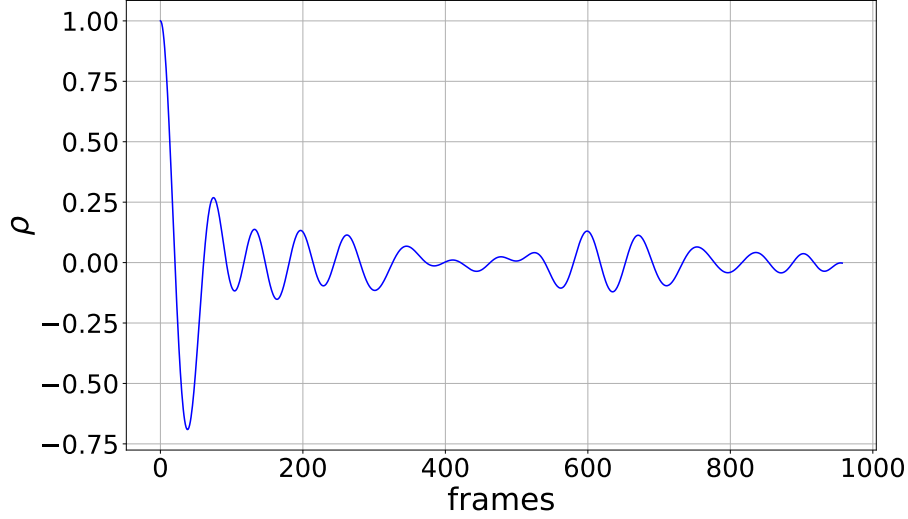


Figura 8: Funzione di autocorrelazione delle forze per la traiettoria NVT a 500 K.

(ovvero quattro potenziali) che abbiamo poi testato sullo stesso *test set*¹⁸. Dunque, se non tutti i *frames* del *training set* iniziale contribuiscono a diminuire l'errore del potenziale, quello che ci si aspetta è che potenziali (reti) costruiti con i nuovi *training set* si comportino come il potenziale iniziale fintanto che non si eliminino dal *training set* *frames* non correlati fra loro. In figura 9a, si è graficato l'errore assoluto E_f^i in funzione dei *frames* del *test set* e dei potenziali costruiti sui vari *training test*, definito come

$$E_f^i = \langle |\mathbf{F}^i - \mathbf{F}^i(w)| \rangle, \quad (60)$$

dove $\langle |\mathbf{F}^i - \mathbf{F}^i(w)| \rangle$ è la norma della differenza tra forze del *test set* ($\mathbf{F}^i$) e quelle ottenute dal PES predetto dalla rete ($\mathbf{F}^i(w)$) rispetto al *frame* i -esimo, mediata su tutti gli atomi del sistema. Come si vede dalla figura 9a, E_f^i non aumenta per i potenziali costruiti con i *training set* 10, 20 rispetto all'errore del potenziale di riferimento costruito con tutto il *training set*. Per i potenziali successivi si vede invece che l'errore aumenta, sensibilmente per il potenziale costruito con il *training set* 210, il che conferma che i *frames* che contribuiscono effettivamente all'apprendimento della rete sono almeno uno ogni 100 time step, coerentemente con il fatto che la correlazione, come si vede in figura 8, diminuisce sensibilmente dopo i 100 frames. Interessante è notare che i *training set* 10, 20 utilizzano rispettivamente il 9.0% e 4.6% dei *frames* del *training set* iniziale e nonostante questo i potenziali prodotti hanno la stessa precisione del potenziale di riferimento. Allo stesso modo il potenziale costruito con il *training set* 110, utilizzando solamente il 0.77% dei *frames*, produce un E_f^i che risulta maggiore di quello di riferimento al massimo di 0.01 eV/Å. Dunque, è evidente che traiettorie ottenute da BOMD non sembrano essere un set di esempi efficiente per l'apprendimento della DNN.

¹⁸Il *test set* è rimasto sempre lo stesso, quello ottenuto dal 20% delle traiettorie calcolate nel paragrafo 2.2 e utilizzato per il test nel paragrafo 2.3.

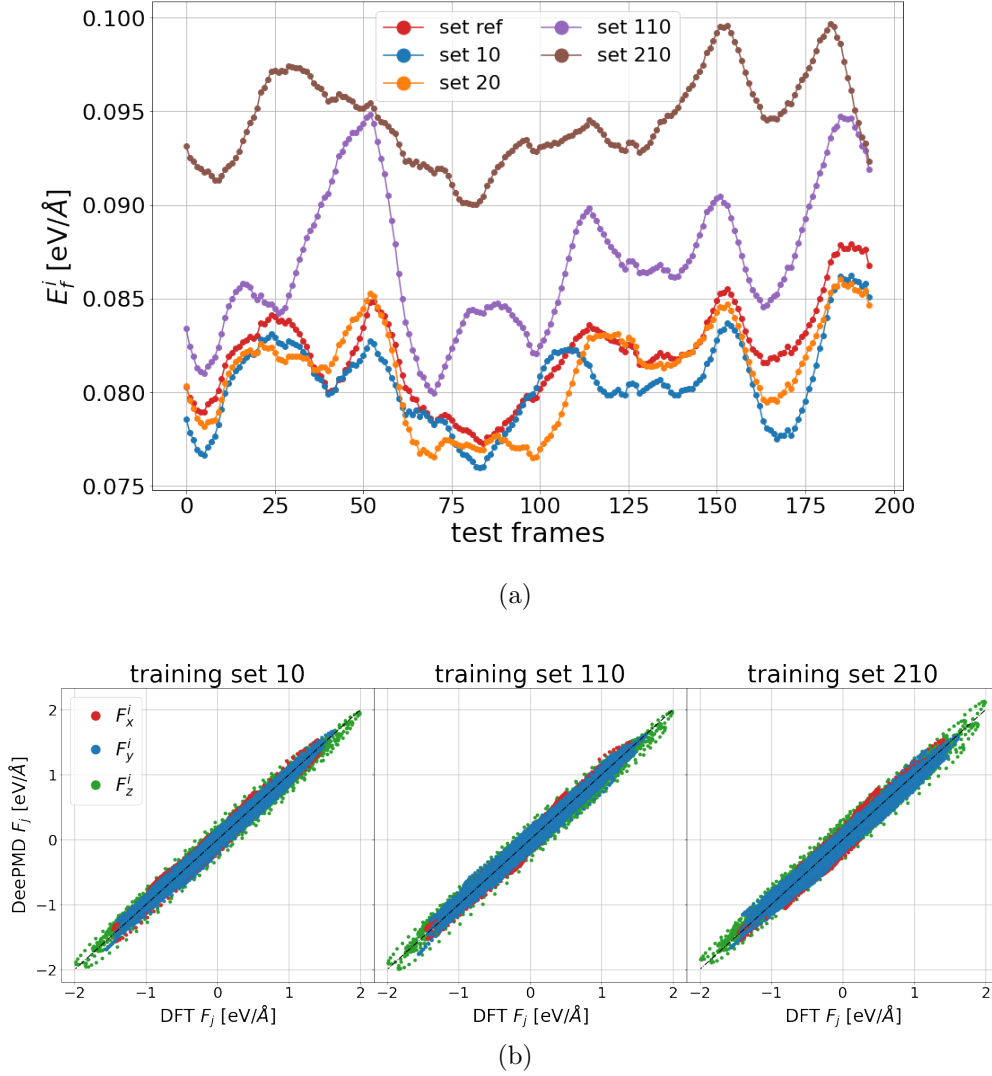


Figura 9: Confronto tra i test dei potenziali costruiti sui vari *training set* a 200 K. (a) Errori assoluti della forza a confronto per i potenziali costruiti con i vari *training set*, in rosso l'errore sul potenziale costruito con tutti i *frames* del *training set* per riferimento. (b) Confronto tra i vari test dei potenziali costruiti con i 3 nuovi *training set*.

2.4.1 Model Deviation

In questo capitolo introduciamo un altro metodo, comunemente usato, per la stima dell'affidabilità di un potenziale DeePMD. Si tratta del calcolo della così detta *model deviation*, ovvero lo scarto quadratico medio su un ensemble di potenziali [9, 19]. L'idea è, infatti, quella di costruire varie reti sullo stesso *training set*, che differiscono le una dalle altre per l'inizializzazione dei valori dei pesi. Ottenendo, dunque, un ensemble di potenziali (reti che riproducono il PES) che testati su un comune *test set* portano a risultati simili, ma non uguali. I risultati, infatti, non sono esattamente gli stessi a causa della natu-

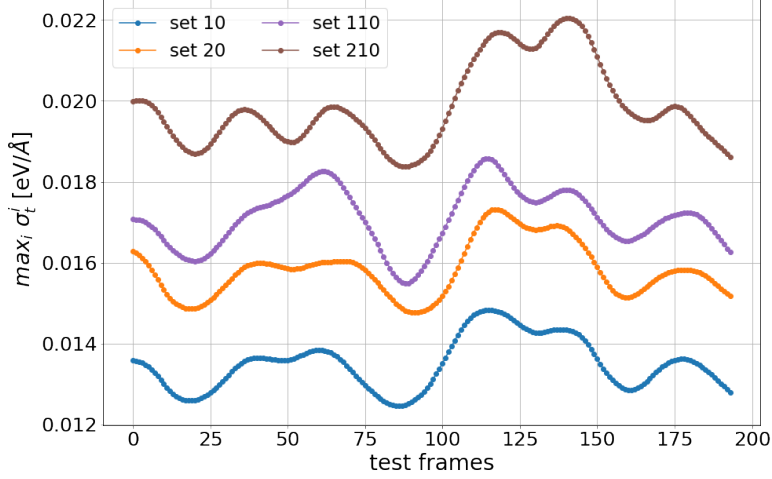


Figura 10: *Model deviation* (σ_t) per gli ensemble di potenziali costruiti con i *training set* 10, 20, 110 e 210, in funzione dei *frames* del *test set*.

ra stocastica dell'apprendimento delle reti. Questo porta reti con valori iniziali di pesi diversi a minimizzare la funzione costo in punti diversi, ma distribuiti in modo stocastico attorno a un valore medio. Ovvero, modifiche della condizione iniziale dell'algoritmo di minimizzazione (paragrafo 1.6.3) della funzione costo, porta a fluttuazioni intorno a un valore medio dei pesi della rete, dunque fluttuazioni del valore del PES calcolato da quest'ultima e di conseguenza fluttuazioni delle forze e del viriale. Pertanto andando a testare su un medesimo *test set* un ensemble di potenziali così costruiti, si può calcolare la deviazione standard (scarto quadratico medio), rispetto al valore medio, della forza su ogni atomo per ogni *frame* del *test set*:

$$\sigma_t^i = \sqrt{\langle |\mathbf{F}_{i,t}^w - \langle \mathbf{F}_{i,t}^w \rangle|^2 \rangle}, \quad (61)$$

dove $\mathbf{F}_{i,t}^w$, indica la forza sull'atomo i -esimo, per il *frame* t -esimo, calcolata dal PES prodotto dalla rete con pesi w , mentre la media $\langle \mathbf{F}_{i,t}^w \rangle$ è effettuata sull'ensemble di M potenziali

$$\langle \mathbf{F}_{i,t}^w \rangle = \frac{1}{M} \sum_{\alpha=1}^M \mathbf{F}_{i,t}^{w_\alpha}, \quad (62)$$

dove si indicano i pesi della rete α -esima, che riproduce il corrispettivo PES, con w_α . Volendo paragonare la *model deviation* con l'errore assoluto in figura 9a, si può, per ogni *training set* utilizzato nel paragrafo precedente, costruire un ensemble di 4 potenziali e graficare $\sigma_t = \max_i \sigma_t^i$ in funzione dei *frame* del *test set* (uguale per tutti gli ensemble e uguale a quello utilizzato nel calcolo degli errori assoluti in figura 9a). Come si vede dalla figura 10, σ_t risulta sempre minore dell'errore assoluto in figura 9a, mostrando come la rete sia molto affidabile, ma non altrettanto precisa. Ovvero le fluttuazioni sugli ensemble sono piccole rispetto all'errore assoluto, il che ci assicura che l'errore che vediamo in

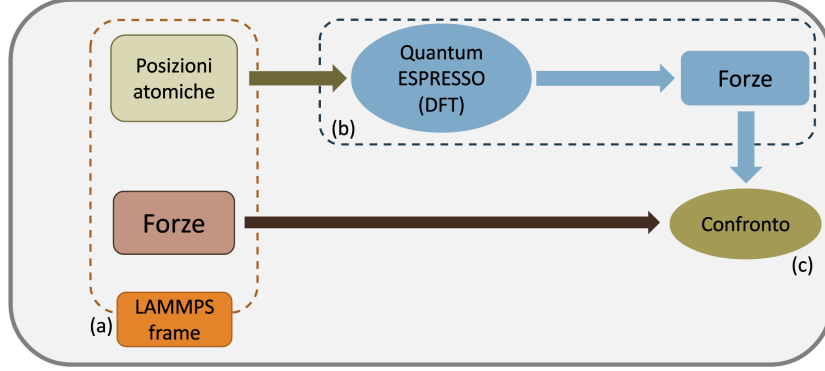


Figura 11: Schema concettuale del test sulla precisione di DeePMD mediante confronto tra forze prodotte da LAMMPS e da Quantum ESPRESSO. (a) Usando il potenziale DeepMD, prodotto dalla DNN, si producono vari *frames* contenenti posizioni atomiche e forze del sistema. (b) Tramite Quantum ESPRESSO, si calcola PES da cui le forze, per le varie posizioni atomiche. (c) Le forze prodotte da LAMMPS e dalla DFT tramite Quantum ESPRESSO vengono confrontate producendo il grafico 13.

figura 9a non derivi da fluttuazioni stocastiche della rete, dovute alle condizioni iniziali di apprendimento, ma sia intrinseco della DNN¹⁹. Il grafico 10 mostra inoltre che per ogni *training set* σ_t aumenta, al contrario dell'errore assoluto E_f^i . L'eliminazione di *frames* correlati del *training set*, nonostante non produca un grande effetto sull'errore assoluto del potenziale DeePMD, dunque sui valori medi delle forze dei vari ensemble come definiti nell'equazione (62), produce un aumento delle fluttuazioni stocastiche, peggiorando l'affidabilità del PES prodotto dalla rete.

2.5 Simulazioni con LAMMPS

Come detto nel paragrafo 2.4, la correlazione tra *frames* delle traiettorie utilizzate per l'apprendimento e il test del PES, mette in dubbio la validità dei test effettuati su quest'ultimo con i *test set* descritti nel paragrafo 2.3. Per poter ottenere una stima effettiva della precisione del potenziale DeePMD, è necessario usare un *test set* non correlato con il *training set* usato per l'apprendimento dalla DNN. Risulta pertanto necessario produrre un *test set* di questo tipo.

Innanzitutto si utilizza il PES ottenuto, per eseguire delle simulazioni del sistema di WTe_2 . Utilizzando il software per la dinamica molecolare LAMMPS [15] che si interfaccia direttamente con la rete creata con DeePMD-kit, abbiamo ottenuto delle nuove traiettorie. Avendo alleggerito notevolmente il carico computazionale, non dovendo utilizzare la DFT per calcolare il PES per ogni configurazione atomica, queste nuove traiettorie sono molto più lunghe di quelle usate per l'apprendimento della DNN. Sono state calcolate due traiettorie nell'ensemble canonico (NVT) a 200 K e 500 K, con un time step di 2.42×10^{-3} ps, per 10^5 step, dunque ottenendo traiettorie dell'ordine di 240 ps. Grazie alla notevole

¹⁹Questo non implica che la rete non possa essere migliorata, con un *training set* più ampio e meno correlato.

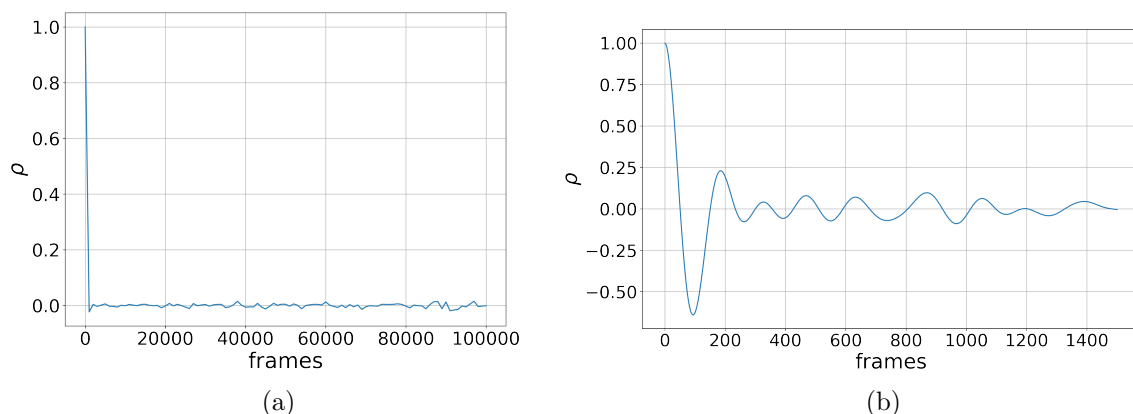


Figura 12: (a) Funzione di autocorrelazione delle forze per la traiettoria ottenuta con LAMMPS a 500K (NVT), calcolata considerando un *frames* (una configurazione di forze) ogni 1000. È evidente come la lunghezza della traiettoria permetta alla funzione di convergere a zero. (b) Dettaglio della Funzione di autocorrelazione graficata a sinistra, considerando tutti i primi 1400 *frames*.

lunghezza delle traiettorie rispetto quelle presentate nel paragrafo 2.2 abbiamo ottenuto *frames* molto meno correlati di quelli ottenuti dalle traiettorie precedenti, come si vede dalla funzione di autocorrelazione delle forze in figura 12a, calcolata per l'ensemble NVT a 500K. Si vede dalla figura 12b che la funzione di autocorrelazione delle forze ha un comportamento simile a quello precedentemente osservato nel paragrafo 2.4, con la differenza che ora abbiamo *frames* distanti più di 1000 *time steps* l'uno dall'altro, il che ci assicura l'assenza di correlazione tra le configurazioni atomiche di questi ultimi. A questo punto, avendo a disposizione un set di configurazioni atomiche non correlate tra loro, utilizziamo la DFT per ottenere da queste ultime il PES, e di conseguenza le forze, ad essa relative, con la precisione caratteristica della dinamica molecolare da principi primi, ovvero con la stessa precisione delle traiettorie usate precedentemente per il *training* e per il test del potenziale DeePMD. Ovvero, prendiamo una configurazione atomica ogni 1000 *time step*, il che ci assicura la non correlazione tra queste configurazioni come ci garantisce la figura 12b, e usiamo la DFT (attraverso Quantum ESPRESSO) per ottenere le forze su ogni atomo del sistema per questa configurazione atomica. Otteniamo dunque per ogni configurazione atomica non correlata le relative forze agenti su ogni atomo. In questo modo possiamo usare questo set di forze come riferimento per la precisione del PES calcolato con DeePMD. Se infatti la rete è molto precisa e riproduce fedelmente il potenziale interatomico, le forze calcolate con questo potenziale non devono discostarsi da quelle calcolate con la DFT, per le medesime configurazioni atomiche. Da notare che questo *test set* di forze è del tutto indipendente da come è costruita la rete, ovvero il potenziale atomico, poiché derivano dalla soluzione dell'equazione di Schrödinger elettronica per una determinata configurazione spaziale degli atomi del sistema. Si può pertanto costruire un grafico come quello presentato nel paragrafo 2.3 in figura 7b, dove ogni punto rappresenta una coppia

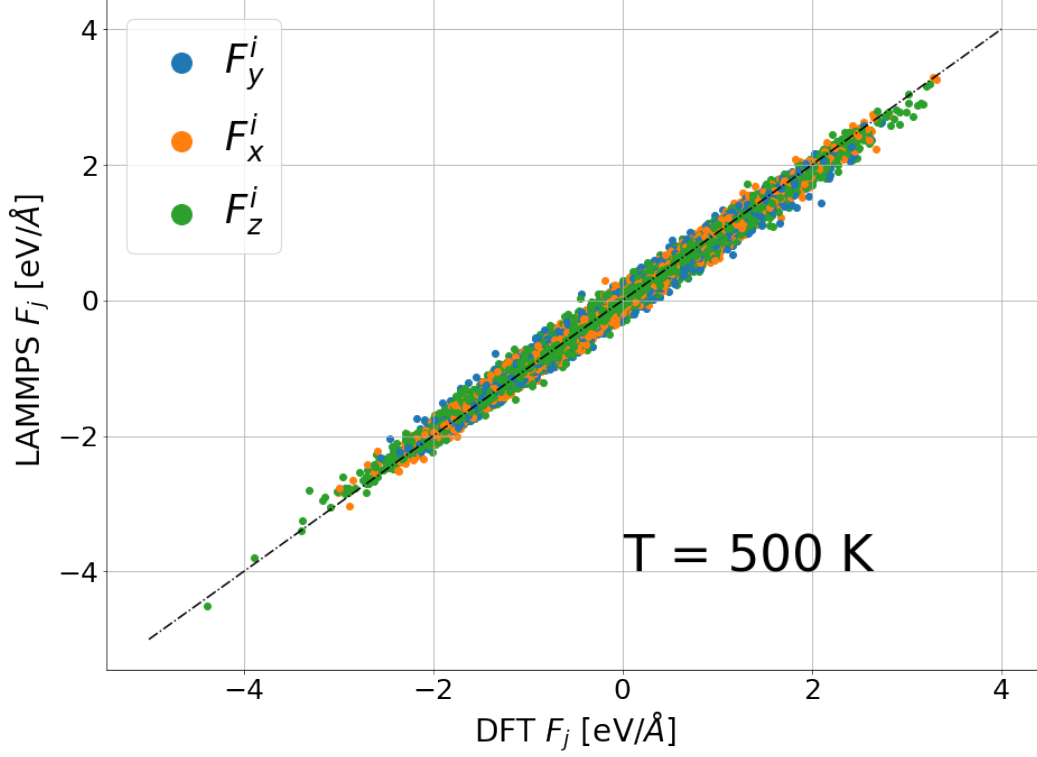


Figura 13: Confronto tra le forze ottenute con LAMMPS con l'utilizzo del potenziale intermolecolare prodotto da DeePMD e le forze ottenute, per le stesse configurazioni atomiche, tramite DFT (con Quantum Espresso), per un sistema NVT a 500 K su 100 *frames* per 216 atomi.

$(F_j^i, F_j^i(w))^{20}$, con F_j^i che rappresenta la componente della forza dell'atomo i -esimo lungo una coordinata $j = x, y, z$, calcolata con Quantum ESPRESSO attraverso DFT, mentre $F_j^i(w)$ rappresenta l'esatto analogo però calcolato usando il potenziale DeePMD. Come si vede dalla figura 13, i punti si distribuiscono intorno alla diagonale, dunque confermando un ottimo comportamento della rete neurale nella predizione del potenziale interatomico. Da notare che questo confronto può essere fatto direttamente confrontando il PES ottenuto con la DFT e quello prodotto con DeePMD, senza calcolare le forze. Tuttavia data l'importanza delle forze nella dinamica del sistema abbiamo preferito fare un confronto direttamente con queste ultime.

2.5.1 Potenziale di seconda generazione

Abbiamo mostrato come il PES, prodotto con il metodo DeePMD per il sistema di WTe_2 , abbia una precisione paragonabile a quella del PES calcolato tramite DFT, almeno per

²⁰La variabile w è stata aggiunta per esplicitare la dipendenza delle forze dalla DNN e dunque dai pesi della rete.

temperature non superiori ai 500 K (figure 7 e 13). È evidente, tuttavia, che per temperature più alte, come discusso nel paragrafo 2.3, il potenziale fatica a riprodurre i risultati della DFT (figura 7b per i 1000 K). Per utilizzare il PES per simulazioni del WTe_2 a temperature dai 500 K ai 1000 K, dove si è più interessati allo studio delle proprietà del sistema, è necessario migliorare quest'ultimo. L'idea è quella di utilizzare le lunghe traiettorie, prodotte da LAMMPS usando il PES già calcolato con DeePMD, per ottenere molte configurazioni atomiche non correlate. Utilizzare poi, queste ultime, per calcolare tramite DFT (sempre attraverso Quantum ESPRESSO) il corrispettivo PES e le forze. Si ottengono, dunque, *frames* non correlati con energie e forze con precisioni tipiche della DFT, esattamente come nel test del potenziale nel paragrafo precedente. Tuttavia, ora, utilizziamo questi *frames* come nuovo *training set*, non correlato, della rete. Si calcolano 5 traiettorie, lunghe 240 ps, di un sistema di WTe_2 di 270 atomi, per le temperature di 200 K, 400 K, 600 K, 800 K e 1000 K (una per ogni temperatura) sia nell'ensemble canonico che isobarico, dunque per un totale di 10 traiettorie. Per un *frames* ogni 5000 *time steps* si calcola tramite DFT il PES e le corrispettive forze, ottenendo, dunque, 20 *frames* per ogni temperatura per entrambi gli ensemble (ovvero 100 *frames* non correlati totali). Di questi 20, 15 vengono usati per l'apprendimento (*training set*), 3 per i test e 2 per la *validation*. Partendo dal PES già costruito e ricominciando l'apprendimento con questo nuovo *training set*, si ottiene un nuovo potenziale DeePMD, che possiamo chiamare di seconda generazione. Testandolo sul *test set* si ottengono le figure 14, analoghe alle figure 7b nel paragrafo 2.3. Queste rappresentano il test delle forze prodotte dal PES di seconda generazione un intervallo di temperature dai 200 K ai 1000 K su entrambi gli ensemble utilizzati. È evidente come il PES sia migliorato per tutte le temperature in entrambi gli ensemble, particolarmente per le alte temperature. Il potenziale DeePMD prodotto ha dunque una precisione comparabile con quella di un PES ottenuto tramite DFT per tutto l'intervallo di temperature considerato.

3 Conclusioni

In questo lavoro si è studiato e applicato il metodo per la dinamica molecolare *deep potential molecular dynamics* (DeePMD) basato sull'uso di reti neurali profonde per il calcolo del *potential energy surface* (PES) con la precisione tipica della teoria del funzionale densità (DFT). Nella prima parte della tesi si introducono le strutture teoriche della dinamica molecolare, delle reti neurali e del DeePMD, andando a porre l'attenzione sulla difficoltà computazionale di ottenere il PES da principi primi e come il metodo DeePMD offra un valido modo per il calcolo di quest'ultimo con la stessa precisione, ma senza il pesante dispendio di risorse computazionali. Nella seconda parte ci si concentra sull'applicazione del metodo a un materiale specifico: il ditellurio di tungsteno (WTe_2) a monostrato. Il PES prodotto viene testato e se ne discute la precisione, mostrando come l'utilizzo di traiettorie ottenute tramite dinamica molecolare da principi primi (con l'uso della DFT), per l'apprendimento del potenziale DeePMD, non sia efficiente, a causa della forte correlazione tra le forze dovuta alla ridotta lunghezza di queste ultime traiettorie. Si approfondisce, inoltre, l'effetto che questa correlazione ha sul test del PES, mostrando come ottenere una valutazione della sua precisione priva di questo *bias*, attraverso la produ-

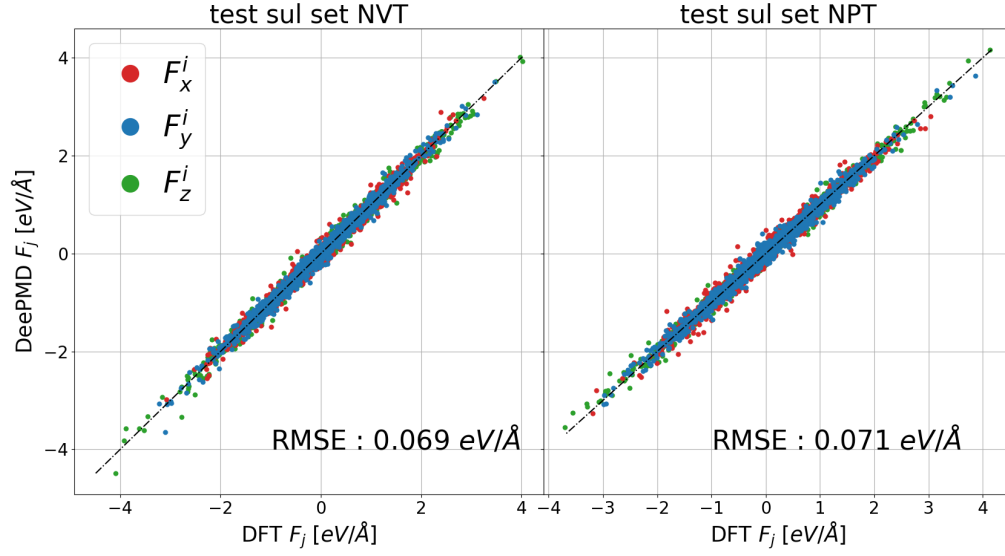


Figura 14: Confronto tra le componenti delle forze prodotte dal PES di seconda generazione e quello calcolato tramite DFT per l'ensemble isobarico (NPT) e quello canonico (NVT). Su entrambi gli ensemble il test è stato effettuato su 15 *frames*, 3 per ogni temperatura (200 K, 400 K, 600 K, 800 K, 1000 K)

zione di configurazioni atomiche non correlate usando il medesimo potenziale DeePMD. Infine, mostratane l'efficacia per il sistema di WTe_2 a basse temperature, si mostra come l'apprendimento da nuovi dati non correlati permette di ottenere un PES di seconda generazione con un notevole miglioramento della precisione per temperature dai 200 K ai 1000 K. In conclusione, quello che si è mostrato in questa tesi è l'efficacia dell'uso del metodo DeePMD e la sua effettiva applicabilità a un materiale bidimensionale, riuscendo a costruire un PES della precisione tipica della DFT, che in futuro potrà essere usato per lo studio delle proprietà e dei comportamenti anarmonici di singoli strati di WTe_2 ad alte temperature.

Riferimenti bibliografici

- [1] Unai Aseguinolaza Aguirreche. Anharmonic effects in thermoelectric and 2d materials. Master’s thesis, University of the Basque Country, 2020.
- [2] M.P. Allen and D.J. Tildesley. *Computer simulation of liquids*. Oxford University Press, 1987.
- [3] B.H. Bransden and C.J. Joachain. *Physics of Atoms and Molecules*. Logman scientific & Technical imprint, 2nd Edition, 2003.
- [4] Furio Ercolessi. *A molecular dynamics primer*. Spring College in Computational Physics, ICTP, Trieste, 1997.
- [5] P. Giannozzi, O. Andreussi, and T. Brumme *et al.* Advanced capabilities for material modelling with Quantum ESPRESSO. *J. Phys.: Condens. Matter*, 29, 2017.
- [6] Feliciano Giustino. *Materials Modelling using Density Functional Theory*. Oxford university press, 2014.
- [7] Jhon Hertz, Andres Krogh, and Richard G. Palmer. *Introduction to the theory of neural computation*. Santa Fe Insititute, 1991.
- [8] Radu Iftime, Peter Minary, and Mark E. Tuckerman. Ab initio molecular dynamics: Concepts, recent developments, and future trends. *PNAS*, 102:6654–6659, 2005.
- [9] Leonid Kahle and Federico Zipol. On the quality of uncertany estimates from neural networks potential ensembles. *Phys. Rev. E*, 105, (2021).
- [10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
- [11] L.Zhang, J. Han, H.Wang, W.A. Saidi, R.Car, and W. E. End to end symmetry preserving inter-atomic potential energy model for finite and extended systems. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 2018.
- [12] A. Marrazzo, M. Gibertini, D. Campi, N. Mounet, and N. Marzari. Relative abundance of Z2 topological order in exfoliable two dimentional insulators. *Nano Lett.*, 19:8431–8440, 2019.
- [13] Doran M.B and Zucker I.J. Higher order multipole three-body van der waals’ interaction and stability of rare gas solids. *Journal of Physics*, C4:307, 1971.
- [14] Xiaofeng Qian, Junwei Liu, Liang Fu, and Ju Li. Quantum spin hall effect in two-dimensional transition metal dichalcogenides. *Science*, 346:1344–1347, 2014.
- [15] A. P. Thompson, H. M. Aktulga, and R. Berger *et al.* LAMMPS - a flexible simulation tool for particle-based materials modelings at the atomic, meso, and continuum scales. *Computer Physics Communications*, 271, 2022.

- [16] Donald G. Truhlar. Potential energy surfaces. *The encyclopedia of physical science and technology*, 13:9–17, 2001.
- [17] Han Wang, Roberto Car, and Weinan E. Deep potential molecular dynamics : A scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.*, 120, 2018.
- [18] Han Wang, Linfeng Zhang, Jiequn Han, and Weinan E. Deepm-kit: A deep learning package for many-body potential energy rapresentation and molecular dynamics. *Computer Physics Communications*, 228:178–184, 2018.
- [19] Yuzhi Zhang, Haidi Wang, Weijie Chen, Jinzhe Zeng, Linfeng Zhang, Han Wang, and Weinan E. DPGEN: A concurrent learning platform for the generation of reliable deep learning based potential energy models. *Computer Physics Communications*, 253(4A), 2020.