

POLITECNICO DI TORINO

**Corso di Laurea
in Ingegneria Aerospaziale**

Tesi di Laurea
**Machine Learning per la fluidodinamica: funzionamento e
applicazioni.**



Relatore

prof.ssa Stefania Scarsoglio

Candidato

Enrico Scribano

Anno Accademico 2019/2020

INDICE

1) Introduzione_____	pag.3
2) Cenni Storici_____	pag.4
3) Machine Learning: fondamenti e principali tipi di algoritmi_____	pag.6
4) Reti neurali_____	pag.14
5) Applicazioni degli algoritmi di apprendimento automatico _____	pag.17
6) Conclusioni_____	pag.25
7) Bibliografia_____	pag.28

1. INTRODUZIONE

Per **intelligenza** si intende il complesso di facoltà psichiche e mentali che consentono all'uomo di pensare, comprendere, spiegare i fatti o le azioni ed elaborare modelli astratti della realtà, caratteristiche che insieme lo rendono capace di adattarsi a situazioni nuove o di modificare la situazione quando questa presenti ostacoli all'adattamento [1]. Di conseguenza, l'**intelligenza artificiale (IA)** è una riproduzione parziale dell'attività intellettuale umana. Ad oggi l'intelligenza artificiale è un campo di ricerca molto vasto, grazie alla facilità con la quale può prestarsi alle varie discipline classiche e non.

Uno dei principali sviluppi dell'intelligenza artificiale è il **Machine learning (ML)**, interpretato in italiano come "apprendimento automatico"; questo tipo di intelligenza artificiale analizza una grande quantità di dati per imparare a svolgere un compito senza che sia stata precedentemente programmata a eseguirlo: attraverso una fase di allenamento fatto su determinati esempi essa apprende la sua mansione.

La necessità di avere molti dati porta subito a pensare, tra i moltissimi campi in cui il ML può essere applicato, alla fluidodinamica, disciplina che ha da sempre a che fare con moltissime quantità di dati provenienti da esperimenti e simulazioni numeriche. La nostra generazione sta, infatti, vivendo un momento favorevole all'incontro tra la meccanica dei fluidi e l'apprendimento automatico grazie allo sviluppo delle architetture informatiche che permettono di immagazzinare, trasformare ed elaborare enormi quantità di dati, insieme alla notevole riduzione del costo computazionale dovuto a nuovi sofisticati algoritmi informatici e ai significativi investimenti delle aziende che hanno intuito i futuri sbocchi [2].

Il metodo di lavoro del machine learning può essere facilmente utilizzato in numerose sfide ancora aperte nella fluidodinamica come la modellizzazione della turbolenza, il controllo, l'ottimizzazione di forma e tutti quei problemi che si sono rivelati finora computazionalmente troppo onerosi. Per garantire lo sviluppo di questo tipo di intelligenza sarà però necessaria una profonda comprensione del funzionamento degli algoritmi, sia nel successo che nel fallimento degli stessi, per evitare di ingigantire le aspettative, ricordando sempre che ci troviamo in un campo aperto di cui dobbiamo imparare a conoscere le potenzialità, ma soprattutto i limiti [2] [3].

2. CENNI STORICI

Il machine learning è un argomento di studio piuttosto recente, la prima volta che venne accostato alla fluidodinamica fu nel 1941 da Kolmogorov, matematico sovietico, che nella sua elaborazione statistica della turbolenza considerò il problema come prima applicazione dell'apprendimento automatico [4].

Nel 1958, Frank Rosenblatt realizzò il **perceptrone**, considerato il primo algoritmo di machine learning, come un classificatore binario con una regola d'apprendimento basata sulla minimizzazione dell'errore. L'emozione per l'uso del perceptrone fu forte, ma venne stroncata qualche anno dopo principalmente perché esso era limitato alle funzioni linearmente separabili (non era, ad esempio, in grado di trattare la funzione XOR) [5]. Una serie di strati di perceptroni sarebbe stata in grado di gestire queste altre funzioni, ma per l'epoca il costo computazionale era troppo alto. Così questo modello fu messo da parte, come l'interesse verso l'intelligenza artificiale che, generalmente, si ridusse negli anni a seguire. Non fu così per tutti: negli anni '60, gli allora studenti dell'Università Tecnica di Berlino **Rechenberg** e **Shwefel** eseguirono degli esperimenti che potrebbero essere visti oggi come precursori dei moderni algoritmi iterativi. I due inserirono in una galleria del vento una struttura formata da cinque pannelli collegati tra loro (figura 2.1), i quali avevano angoli relativi che potevano essere modificati: l'esperimento consisteva nell'aumentare o diminuire (utilizzando un generatore di numeri casuali) gli angoli tra i pannelli e misurare la resistenza aerodinamica della struttura, con l'obiettivo di minimizzarla (figura 2.2). L'incremento o il decremento dell'angolo era auto-appreso dall'algoritmo attraverso il risultato negativo o positivo della precedente misura, tuttavia i due non ottennero un grande riscontro dalla comunità, nonostante le numerose idee e applicazioni

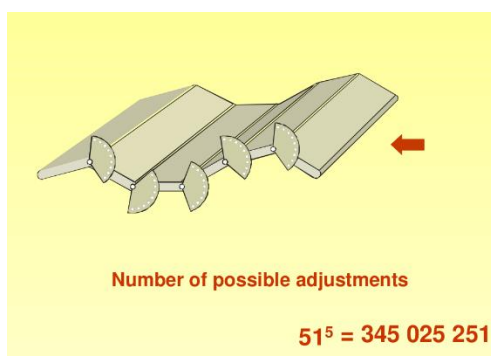


Figura 2.1 Rappresentazione della struttura a cinque pannelli utilizzata da Rechenberg e Shwefel per il loro esperimento.

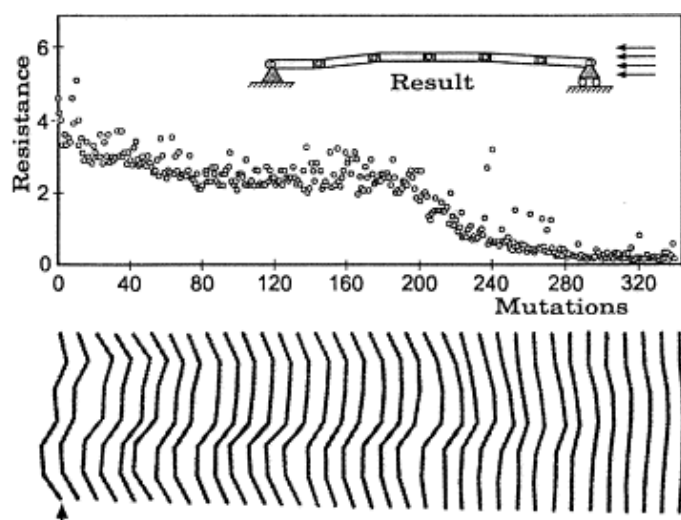


Figura 2.2 Resistenza misurata in funzione degli angoli dei pannelli della struttura utilizzata da Rechenberg e Shwefel.

che si rifanno allo stesso esperimento [3].

Sul finire degli anni '70 gli interessi nella ricerca sull'intelligenza artificiale furono stroncati dall'articolo di J. Lighthill "Artificial Intelligence: A General Survey", nel quale riportò molte critiche sull'argomento, sostenendo che quasi nulla sarebbe stato ricavato da questi studi. Si può però effettivamente notare come, in sua difesa, 40 anni fa fosse ancora impensabile la potenza dei mezzi informatici odierni.

L'interesse oggi è ritornato alto grazie al progresso dell'informatica, partendo dagli anni '90 con lo sviluppo delle **reti neurali**: un'architettura informatica che si basa sul funzionamento biologico delle reti dei neuroni, ispirata dal lavoro dei premi Nobel Hubel e Wiesel sul funzionamento della corteccia visiva primaria dei gatti. Col passare degli anni alle reti neurali vennero affiancati altri strumenti, come la POD (decomposizione propria ortogonale), utilizzati poi per ricostruire i campi di turbolenza e la ricostruzione dei flussi nell'intorno della parete partendo dai soli dati di parete: è la prima volta in cui vennero usati più strati di reti neurali, aprendo la pista al campo del **deep learning** approfonditamente studiato tutt'oggi [8].

È quindi interessante applicarsi allo studio di quali algoritmi e di come essi possano mettersi a disposizione dello studio della fluidodinamica, soprattutto per gli aspetti che risultano ancora oscuri.

Nel capitolo successivo saranno trattati i principali tipi di algoritmi di apprendimento automatico e i problemi risolvibili con gli algoritmi stessi, fondamenti essenziali da cui partire per capire dove e quanto possano essere efficaci nel risolvere i problemi di interesse fluidodinamico.

3. MACHINE LEARNING: FONDAMENTI E PRINCIPALI TIPI DI ALGORITMI

Il funzionamento del Machine Learning si basa in prima battuta sui dati, infatti il comportamento dell'algoritmo non è pre-programmato ma viene appreso dai dati a disposizione. Spesso ci si riferisce agli stessi col termine inglese “**pattern**”, che in italiano può essere tradotto come: campione, esempio, modello o forma. Un pattern è un'entità vagamente definita a cui può essere dato un nome, esso può essere un carattere scritto a mano, un volto, un frammento di testo o il movimento di una particella nello spazio e nel tempo. I pattern possono essere:

- Numerici: valori associati a caratteristiche misurabili e rappresentabili in vettori, quindi soggetti all'ordinamento. Per una persona possiamo considerare pattern numerici tutte quelle informazioni che la caratterizzano fisicamente come: altezza, peso, età, circonferenza toracica, lunghezza del piede ecc.

Regione	AGGIORNAMENTO 10/03/2020 ORE 17.00							
	POSITIVI AL nCoV				DIMESSI GUARITI	DECEDUTI	CASI TOTALI	TAMPONI
	Ricoverati con sintomi	Terapia intensiva	Isolamento domiciliare	Totale attualmente positivi				
Lombardia	3319	466	642	4427	896	408	5791	21479
Emilia Romagna	669	98	650	1417	31	83	1533	5494
Veneto	204	67	512	783	47	26	856	16643
Piemonte	306	66	64	436		17	453	2374
Marche	152	54	175	381		13	394	1437
Toscana	91	40	129	260	3	1	264	2573
Lazio	50	15	34	99	11	6	116	3591
Campania	33	8	85	126	1		127	1141
Liguria	57	29	42	128	5	8	141	694
Friuli V.G.	27	6	77	110	5	8	116	1602
Sicilia	17	2	41	60	2		62	955
Puglia	28	6	21	55	1	8	59	747
Trento	19	3	28	50	1		52	399
Abruzzo	17	9	11	37		8	38	310
Umbria	8	2	27	37			37	260
Molise	3	3	9	15			15	225
Sardegna	9		11	20			20	204
Valle d'Aosta	2		15	17			17	89
Calabria	8	2	1	11	2		13	360
Bolzano	17	1	20	38			38	36
Basilicata	2		5	7			7	148
TOTALE	5038	877	2599	8514	1004	491	10149	60761

ATTUALMENTE POSITIVI	8514
TOTALE GUARITI	1004
TOTALE DECEDUTI	491
CASI TOTALI	10149

Figura 3.1 Pattern numerici: Positivi al Covid-19 per regione il 10 marzo 2020 [19]. Ogni dato o sequenza di dati in questa tabella è rappresentabile con un vettore o un numero.

- Categorici: associati a caratteristiche qualitative e alla presenza/assenza di una determinata caratteristica. Sempre riferendoci a una persona, appartengono a questa categoria tutte quelle informazioni che la distinguono ma che non sono esprimibili attraverso una misura come ad esempio: sesso, colore degli occhi, colore dei capelli, gruppo sanguigno, cittadinanza ecc.

- Sequenze: pattern sequenziali con relazioni spaziali o temporali, come una sequenza di suoni di una canzone o le immagini che compongono un video.



Figura 3.2 Pattern sequenziale: sequenza di immagini consecutive ritagliate da un video [20] in cui un tennista esegue una battuta.

- Altre strutture quali grafi e alberi.

Per permettere la migliore preparazione possibile della macchina, i dati disponibili, o pattern, vengono divisi in tre gruppi, funzionali alle fasi di apprendimento dell'algoritmo:

- **Training set:** l'insieme di pattern su cui viene addestrato il sistema, trovando il valore di ottimo dei parametri della macchina.
- **Validation set:** i pattern su cui vengono tarati gli iperparametri, ossia parametri che devono essere definiti prima dell'apprendimento vero e proprio e che devono essere ottimizzati per evitare di allungare i tempi computazionali (sovrastimando gli iperparametri), o di non giungere a convergenza (sottostimandoli). Ad esempio, possono essere iperparametri i numeri di neuroni di una rete neurale, il grado del polinomio utilizzato in una regressione o il tipo di loss function usata dall'algoritmo.
- **Test set:** l'insieme di pattern su cui vengono valutate le prestazioni finali [7] [9].

Il problema dell'apprendimento può essere descritto come il processo di stima delle associazioni tra input e output di un sistema utilizzando il minor numero possibile di osservazioni [6].

Il comportamento dell'algoritmo è regolato da una serie di parametri \mathbf{w} (in una rete neurale possono essere i pesi delle connessioni tra neuroni, per esempio), che durante l'apprendimento devono essere determinati. Dato un training set, che chiameremo train, e un insieme di parametri \mathbf{w} , viene indicata una funzione obiettivo $f(\mathbf{train}, \mathbf{w})$. A seconda dello scopo dell'algoritmo la funzione dovrà essere

massimizzata, per cercare l'ottimo della soluzione, o minimizzata per ridurre l'errore (in questo caso si parla di loss function). Dal risultato dell'ottimizzazione della funzione, che può essere fatto esplicitamente attraverso le derivate della stessa o implicitamente utilizzando metodi euristici che operano coerentemente con la funzione f , si ricavano i valori dei parametri w che saranno utilizzati dall'algoritmo per svolgere il suo compito dopo la fase di apprendimento [7].

I vari algoritmi di apprendimento automatico possono essere raggruppati in tre grandi categorie che li discriminano in base al tipo di addestramento [2]:

- Apprendimento supervisionato (supervised).
- Apprendimento non supervisionato (unsupervised).
- Apprendimento semi supervisionato (semi-supervised).

3.1.Apprendimento supervisionato

L'apprendimento supervisionato è la tecnica più comune e basilare del machine learning, la caratteristica fondamentale che distingue questo tipo di algoritmi è insita nei pattern: le classi dei dati utilizzati per l'addestramento sono tutte note e il training set è etichettato con gli output corrispondenti (figura 3.1). La scelta della funzione, che in questo caso dovrà essere minimizzata, riflette la consapevolezza del tipo di dati in esame, determinando anche la linearità o meno della stessa, da cui dipende direttamente il costo computazionale dell'algoritmo da realizzare [2].

Il metodo più conosciuto per usare algoritmi di apprendimento supervisionato sono le reti neurali, alle quali successivamente è dedicato un capitolo.

3.1.1. *Classificazione*

Una delle principali attività che può essere eseguita da un algoritmo ad apprendimento supervisionato è la **classificazione**. La classificazione, uno dei compiti più semplici per gli algoritmi ML (il primo realizzato, il perceptrone, era proprio un classificatore binario), consiste nell'assegnare una classe a un pattern, dove per classe si intende un insieme di proprietà comuni che dipende intrinsecamente dall'applicazione alla quale si lavora. La pratica viene spesso chiamata riconoscimento. Gli algoritmi principali di classificazione sono due: macchine a vettori di supporto SVM e foreste casuali (random forest). L'idea alla base delle SVM è quella di rappresentare i dati in uno spazio di dimensioni coerenti coi pattern e individuare l'iperpiano (un sottospazio di dimensione inferiore di una unità) di separazione tra le classi che massimizzi il margine, ossia la distanza minima tra i punti delle classi nella fase di training [10]. Invece le foreste casuali sono costituite da alberi decisionali (il termine

foresta deriva appunto dal grande numero di alberi decisionali presenti) che, ordinati secondo gerarchie, dividono i dati passo passo. Il risultato in output dal programma sarà la classe di appartenenza del dato in input per entrambi i metodi [2]. La classificazione è ormai utilizzata per risolvere problemi di natura tra loro completamente diversa come ad esempio il riconoscimento facciale di sblocco dei cellulari, programmi per suggerire diagnosi mediche partendo dai sintomi del paziente o banalmente il riconoscimento dello spam nelle caselle di posta elettronica, come mostrato in figura 3.3.

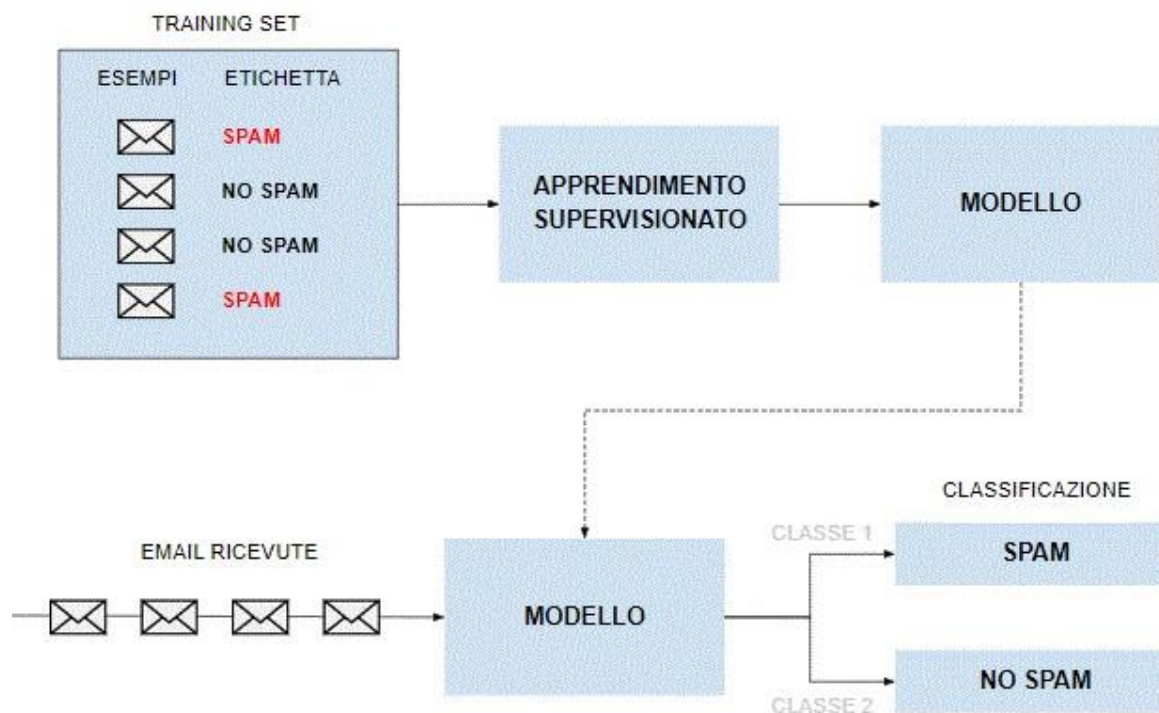


Figura 3.3 Schema del funzionamento di un algoritmo di riconoscimento dello spam [16].

Ad ogni pattern del training set, che in questo caso sono delle mail, viene assegnata una classe: “spam” o “no spam”. L’algoritmo di apprendimento supervisionato studia il set ed elabora un modello per il riconoscimento dello spam. Una volta eseguite le altre fasi dell’apprendimento (attraverso validation set e test set), il modello è pronto per essere collegato alla casella di posta elettronica e svolgere il suo compito di classificatore.

3.2.Apprendimento non supervisionato

Nell’apprendimento non supervisionato ci si trova in una situazione opposta a quella della categoria precedente: le classi dei pattern utilizzati non sono note, e il training set non è etichettato. I compiti

che questo tipo di algoritmi sono deputati a svolgere si basano sull'estrazione di informazioni dai dati, seguendo certi criteri, senza l'aiuto di supervisione o etichette.

I problemi che vengono fatti risolvere dagli algoritmi non supervisionati sono principalmente due: **riduzione di dimensionalità** e **clustering** (raggruppamento).

3.2.1. Riduzione di dimensionalità

L'obiettivo dei metodi di riduzione di dimensionalità (dall'inglese "dimensionality reduction") è quello di mappare i dati iniziali da uno spazio di dimensioni d a uno spazio con dimensione inferiore k . Questo tipo di pratica può essere considerata una specie di compressione dei dati, con conseguente perdita di informazioni. Il problema da risolvere è quello di scartare solamente quelle informazioni che non sono pertinenti o scarsamente rilevanti al caso di studio. Questo tipo di compito può essere svolto come processo prima dell'utilizzo di algoritmi supervisionati, poiché operare in spazi ad alte dimensioni richiede, oltre a lunghi tempi di esecuzione, imponenti quantità di dati etichettati per l'addestramento. Dunque, operare in spazi a dimensioni inferiori rende più semplice e veloce l'addestramento, scartando dati ridondanti, informazioni correlate non utili e dati rumorosi, facendo però attenzione a non cancellare e preservare informazioni, ma combinarle in modo da ottenere la risoluzione ottimale. La tecnica di riduzione di dimensionalità più conosciuta è la PCA, analisi delle componenti principali, che esegue una mappatura lineare con l'intento di preservare la maggior parte delle informazioni del pattern. Un metodo diverso per effettuare la riduzione di dimensionalità è quello di filtrare le informazioni in input attraverso un codificatore (coder) e successivamente attraverso un decodificatore (decoder) che ricodifichi solamente le informazioni utili allo scopo. La riduzione di dimensionalità non lineare può essere fatta con la tecnica encoder-decoder attraverso reti neurali profonde [2].

3.2.2. Raggruppamento

Il clustering, o meglio raggruppamento in italiano, è una famiglia di tecniche ad apprendimento non supervisionato che ha lo scopo di individuare raggruppamenti propri di pattern ed eventualmente definire la corrispondenza tra raggruppamenti e classi (che non sono note) [7]. Il problema è piuttosto complesso, date le molteplici soluzioni possibili, e spesso si può trovare una soluzione esaustiva solamente in presenza di pochi pattern. I criteri di clustering si basano su due osservazioni principali:

- I. I pattern di uno stesso cluster devono essere il più simile tra loro rispetto a pattern di cluster diversi.
- II. I cluster sono costituiti da nuvole di punti a densità elevata, mentre le zone esterne ai raggruppamenti devono avere densità molto più bassa.

Criteri più utilizzati sono quelli che minimizzano la distanza tra gli elementi del cluster e il centroide del gruppo assegnato, oppure quelli che minimizzano le distanze tra elementi dei cluster [7].

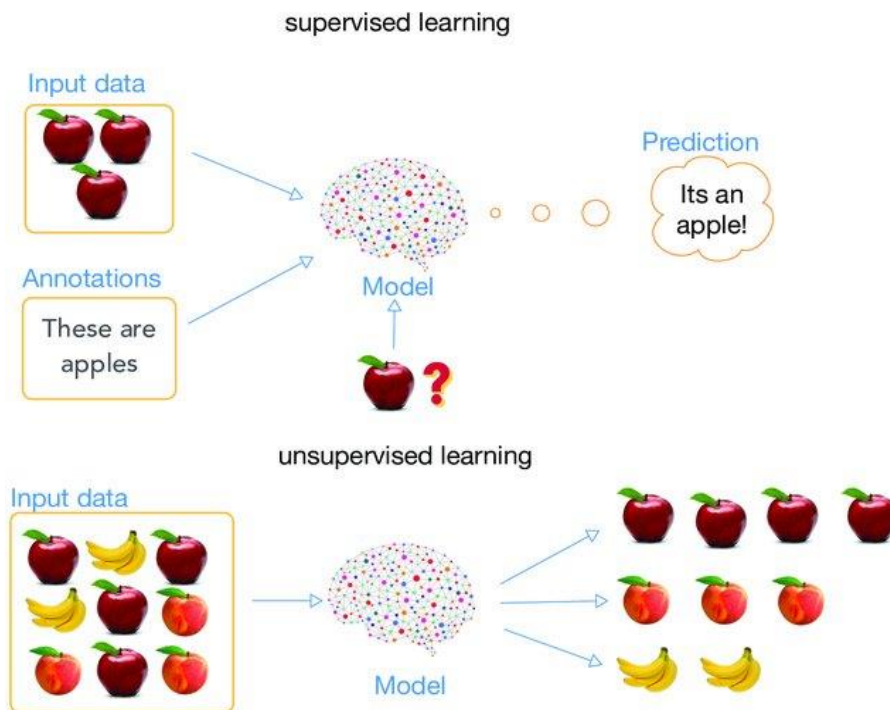


Figura 3.4 Differenza tra apprendimento supervisionato e non supervisionato [21]. Il primo, grazie all'etichetta è capace di riconoscere una mela, il secondo invece non sa cosa sia o quale sia una mela, ma riconosce le differenze tra mela, pesca e banana e le può raggruppare tra loro.

3.3. Apprendimento semi-supervisionato

L'apprendimento semi-supervisionato è un ibrido delle due precedenti, infatti il training set è parzialmente etichettato o riceve informazioni correttive dall'esterno. Gli algoritmi semi supervisionati principali sono: **reti generative avversarie** e **l'apprendimento per rinforzo**.

3.3.1 Reti generative avversarie

Le reti generative avversarie (GAN, dal loro nome inglese) o reti antagoniste generative, sono classi di algoritmi ad apprendimento semi-supervisionato che funzionano attraverso un training set parzialmente etichettato. L'algoritmo è composto da due reti neurali avversarie che interagiscono tra di loro: un generatore e un discriminatore. Il generatore opera concependo dati che siano sempre il più simile possibile a quelli etichettati; invece il discriminatore deve apprendere come distinguere i dati etichettati inseriti in input da quelli generati dal generatore [2]. Il processo finisce quando il discriminatore non è più in grado di distinguere i dati generati da quelli reali: ciò significa che il generatore ha imparato a stimare correttamente i dati. Poiché le reti si autogestiscono, questo tipo di

algoritmo è anche chiamato apprendimento auto-supervisionato. Come le altre classi di apprendimento, anche le GAN hanno bisogno di moltissimi dati in entrata, ma al livello attuale esse non danno garanzia di convergenza.

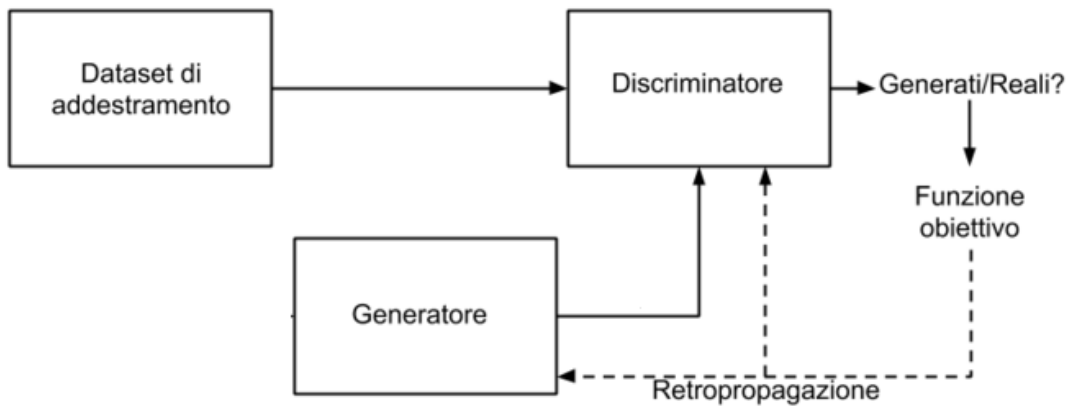


Figura 3.5 Schema del funzionamento di una GAN

3.3.2 Apprendimento per rinforzo

L'apprendimento per rinforzo, in inglese reinforcement learning (RL), è un metodo d'apprendimento che prevede l'interfacciarsi tra l'esterno e l'algoritmo stesso, chiamato agente. L'agente ha un repertorio di azioni e conoscenze base che, diversamente dagli altri semi-supervisionati, non apprende da alcun pattern etichettato, ma impara dall'esperienza che gli deriva dal collegamento esterno [2].

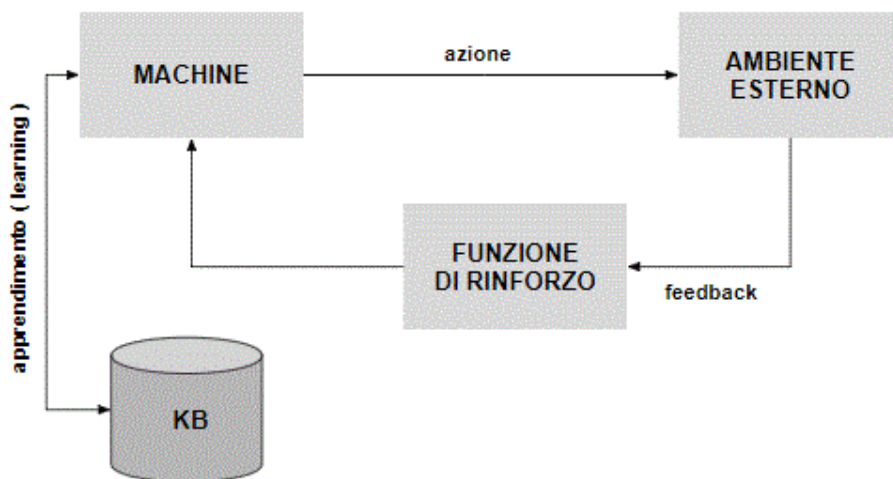


Figura 3.6 Schema del funzionamento di un algoritmo di apprendimento per rinforzo. Attraverso le conoscenze di base (KB) l'algoritmo compie un'azione sull'ambiente esterno, il quale darà una risposta (feedback) che ricompenserà o penalizzerà l'algoritmo attraverso una funzione di rinforzo [16].

Attraverso le conoscenze base, l'algoritmo compie delle azioni sull'ambiente. L'ambiente risponde all'azione dell'algoritmo dando un feedback, che ricompenserà o penalizzerà l'algoritmo attraverso una funzione di rinforzo. Ad azioni corrette dell'algoritmo corrispondono delle "ricompense", esattamente come nell'addestramento dei cani. L'algoritmo è programmato non solo per eseguire azioni e ricevere riscontri, ma anche per massimizzare le ricompense a lungo termine. Prime applicazioni di questi algoritmi sono sicuramente i giochi: negli anni '90 un algoritmo a RL fu programmato con le basi del backgammon [18], all'inizio venne fatto giocare contro sé stesso milioni di volte fino ad assumere esperienza di gioco tale da poter competere coi migliori giocatori umani. Al giorno d'oggi, grazie alle reti neurali è possibile sviluppare videogiochi molto più complicati del backgammon.

È però importante notare come un gioco relativamente semplice come il backgammon impieghi milioni di partite per far sviluppare sufficiente esperienza alla macchina, di conseguenza applicarlo alle simulazioni fluidodinamiche risulterebbe proibitivo con la potenza di calcolo attuale.

4 RETI NEURALI

Le reti neurali, NN dall'inglese neural network, sono uno dei metodi più studiati per eseguire algoritmi di apprendimento supervisionato. Come specificato in precedenza, prendono il nome dalla struttura biologica alla quale sono ispirate: i neuroni.

Negli anni '60 i neuroscienziati Hubel e Wiesel studiarono approfonditamente la corteccia visiva primaria scoprendone una organizzazione gerarchica strutturata in diversi livelli, o strati. È utile riprodurre una breve descrizione del neurone animale per meglio comprendere l'analogia con le reti neurali artificiali [8].

I neuroni sono cellule della corteccia cerebrale animale costituiti da un corpo centrale nel quale è contenuto il nucleo della cellula, circondato da brevi ramificazioni chiamate dendriti e con un prolungamento più lungo detto assone. La parte terminale dell'assone possiede altre piccole diramazioni che sono in connessione con i dendriti di altre cellule neurali, queste connessioni prendono il nome di sinapsi. Attraverso questi collegamenti i neuroni si scambiano segnali elettrici grazie ai quali possono comunicare. Pur essendo cellule piuttosto semplici, i neuroni, grazie alla loro organizzazione ramificata costituita da miliardi di cellule riescono a eseguire operazioni decisamente complesse. Ogni neurone è collegato ad altri attraverso le sinapsi, che nel cervello umano sono stimate in un numero tra 10^{13} e 10^{15} [11].

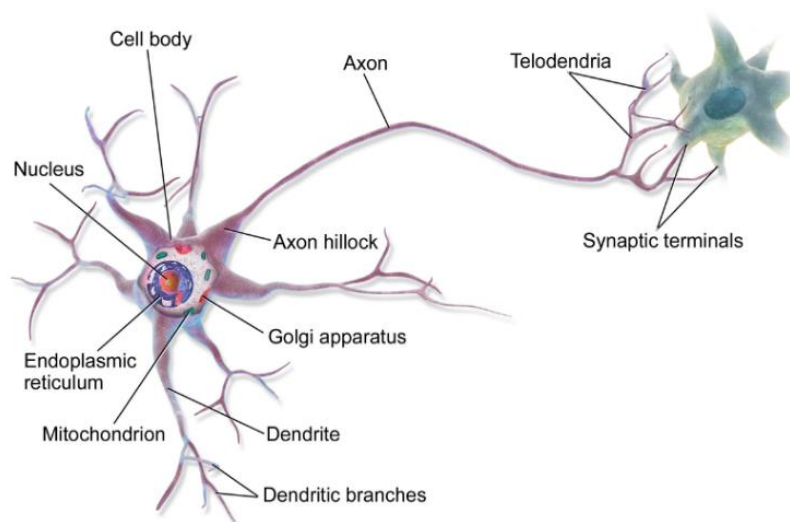


Figura 4.1 Schema di un neurone umano: si notino il nucleo, i dendriti e l'assone con le sue terminazioni[5].

Così come il neurone riceve impulsi, anche l'unità base della rete neurale riceve in input dei valori numerici x_n attraverso un sistema di pesi w_n . L'unità effettua una somma pesata degli input x sui pesi w e in seguito applica una funzione di attivazione per poi ottenere l'output. Il valore dei pesi w viene modificato durante l'apprendimento, così come nei neuroni umani le sinapsi si rinforzano o si indeboliscono. Infatti, le sinapsi seguono la regola di Hebb: se due neuroni interconnessi tra loro attraverso collegamenti sinaptici sono ripetutamente attivati, le sinapsi che li collegano saranno rinforzate [7]. Analogamente, l'apprendimento fa sì che i pesi w delle unità che producono risultati giusti vengano aumentati per avere maggiore impatto nel risultato finale, mentre i pesi delle unità che sbagliano saranno diminuiti.

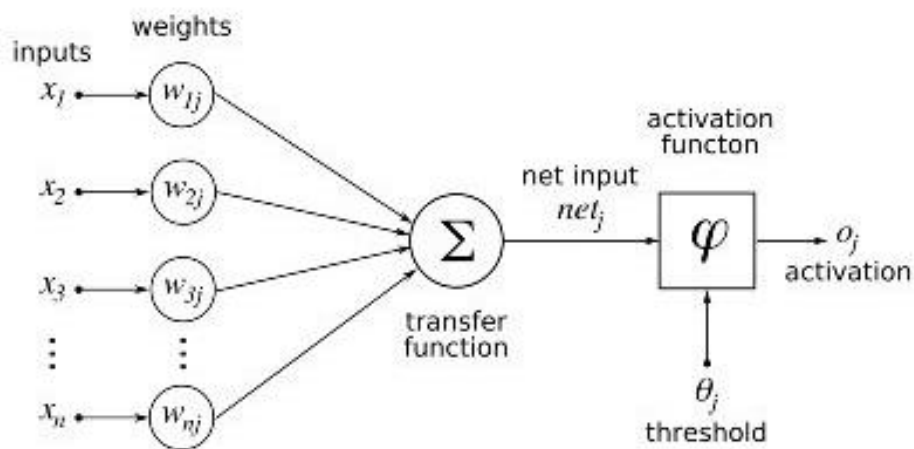


Figura 4.2 Schema di una unità base di una rete neurale, "x" sono gli input, "w" i pesi assegnati a ogni linea d'input e "o" identifica l'output dell'unità [5].

Più unità disposte su uno stesso strato, layer in inglese, costituiscono un perceptrone[13]. Per risolvere problemi sempre più complessi (e soprattutto non lineari), sono state realizzate reti neurali a più strati di neuroni; questo tipo di reti è costituito da uno strato di input, uno di output e un numero variabile di strati nascosti, chiamati hidden layer. Una rete composta da più strati nascosti è detta deep neural network DNN, ossia rete neurale profonda. La potenza delle DNN è dovuta alla loro flessibilità,

poiché è possibile combinare le unità base in livelli a seconda del tipo di problema che l'algoritmo dovrà risolvere [7].

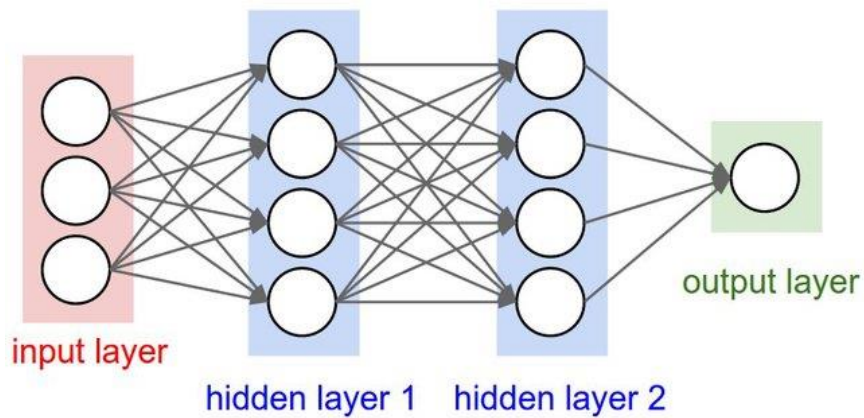


Figura 4.3 Rappresentazione sintetica di una Rete Neurale profonda, alimentata in avanti, a due livelli nascosti [11].

Quando una rete ha solamente neuroni con collegamenti in avanti, ossia solamente con il livello successivo e non con livelli precedenti o sullo stesso layer, si chiama rete feedforward (alimentata in avanti). Se invece sono previsti collegamenti all'indietro si parla di **rete neurale ricorrente** RNN, di elevata utilità nella fluidodinamica poiché è in grado di operare su pattern sequenziali, come immagini di un video o dati che cambiano nel tempo, modificando i parametri w grazie alla retro-propagazione. Recentemente gli algoritmi RNN hanno visto un maggiore impiego, grazie allo sviluppo di memorie LSTM (long short term memory) in grado di memorizzare dati per diversi intervalli di tempo per poi eliminarli quando non più utili, aumentando di molto la velocità computazionale dei problemi [7].

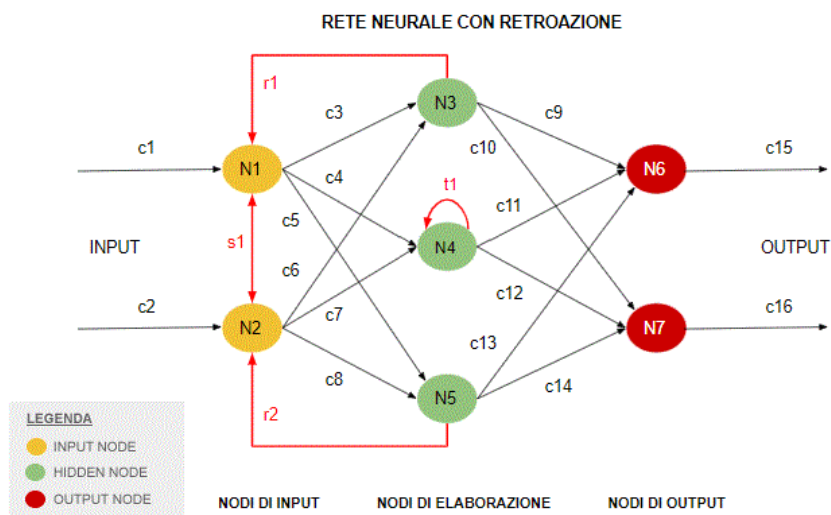


Figura 4.4 Schema di una Rete Neurale Ricorrente, in rosso sono evidenziati i collegamenti all'indietro[16].

5. APPLICAZIONI DEGLI ALGORITMI DI APPRENDIMENTO AUTOMATICO

Le applicazioni del machine learning sono tutt'oggi sparse in moltissime discipline e campi di studio completamente diversi tra loro. Ciò è dovuto alla differenza sostanziale che questo metodo trova con quelli classici, che sono stati sempre adottati. Il suo punto di forza principale è proprio la versatilità: un classico programma informatico non può agire al di fuori di ciò per cui è stato programmato, infatti esso non impara nulla con l'esperienza, tutto ciò di cui ha bisogno lo conosce già e trovandosi in una situazione non prevista si blocca, non potendo scegliere come agire in quel caso. Al contrario, un algoritmo machine learning consente al programma di affrontare situazioni non previste inizialmente, prendendo decisioni non predette dal programmatore. Ad esempio, sarebbe impensabile poter costruire un'automobile a guida autonoma senza apprendimento automatico perché gli scenari davanti ai quali può trovarsi sono pressoché infiniti ed è assolutamente impossibile che un programmatore li preveda tutti.

Al di fuori della fluidodinamica, l'apprendimento automatico è comunque un mondo da scoprire, ma le applicazioni studiate finora lasciano ben sperare per il futuro del campo: una di quelle più sorprendenti sono sicuramente gli assistenti virtuali. Appartengono a questa categoria i più celebri Assistente Google, Siri di Apple e Cortana di Microsoft, ma anche altri utilizzati in grandi aziende di servizi per garantire un primo livello base di interazione col cliente (per citarne uno, Conversica AI). Questi assistenti sono in grado di interagire con gli esseri umani attraverso telefoni, pc o altri dispositivi elettronici ascoltando ed eseguendo compiti come effettuare chiamate o rispondere a domande attraverso una voce sintetizzata. La maggior parte di questi utilizza le registrazioni delle conversazioni con gli utenti per apprendere di volta in volta a interagire e rispondere nel miglior modo possibile. Altri algoritmi sono utilizzati per riconoscere la scrittura umana e trascriverla, o anche tradurla in un'altra lingua (Google traduttore).

Le applicazioni degli algoritmi di apprendimento automatico nella fluidodinamica sono molte e in diversi ambiti, ma sono unite tutte da uno scopo comune: aiutare a trovare un modello capace di descrivere la realtà di un problema in un modo migliore di quanto si sia riusciti a fare senza. Modellizzare, secondo il dizionario Zingarelli, significa “trarre da una realtà fisica o concettuale complessa un modello che la rappresenti descrivendone gli aspetti sostanziali e le relazioni funzionali”. Il tassello fondamentale che sta alla base dello studio dei moti fluidi sono le leggi di conservazione, ricavate nei secoli passati, le cui equazioni di bilancio sono utilizzate per formare le famose leggi di Navier-Stokes. Compiere delle simulazioni utilizzando quel modello, soprattutto per numeri di Reynolds piuttosto elevati, va oltre le attuali capacità computazionali. La prima alternativa è quella di svolgere le simulazioni attraverso delle approssimazioni appositamente selezionate delle

equazioni di Navier-Stokes (ad esempio le equazioni di Eulero che descrivono il moto di un fluido inviscido). Anche utilizzando le dovute approssimazioni, le simulazioni risulterebbero comunque molto costose in termini di calcoli se utilizzate iterativamente per risolvere i problemi di ottimizzazione e molto lente se utilizzate per realizzare il controllo in tempo reale [2], presentando così una delle prime difficoltà da risolvere.

Nello studio di un qualsiasi problema attraverso tecniche di machine learning, soprattutto per l'impiego nella fluidodinamica, il primo passo da compiere dopo la definizione del problema stesso è quello di ottenere i dati da utilizzare per le fasi di apprendimento e la loro riduzione per diminuire il costo computazionale dell'algoritmo. Questo processo viene chiamato "estrazione dei dati".

5.1 Estrazione dei dati e modellizzazione

L'estrazione dei dati è un processo comune a tutti gli algoritmi. Ricavare dati utili è il primo compito da svolgere per poter studiare un fenomeno fluidodinamico attraverso il machine learning e solo in seguito, avendo conoscenza del tipo di pattern che si andranno ad utilizzare si potrà elaborare il programma.

Però, come accennato in precedenza, trattare i dati così come raccolti non è possibile a causa delle dimensioni troppo grandi che questi occuperebbero, serve dunque ridurre le dimensioni, ed è qui che vengono in aiuto gli algoritmi accennati in precedenza: per giungere a dei risultati è necessario ottenere dei modelli che catturino l'essenza del meccanismo del fluido e che siano fruibili a un costo computazionale molto minore, dunque è opportuno agire operando delle riduzioni di dimensionalità.

Gli algoritmi più utilizzati sono quelli già citati, la PCA (analisi delle componenti principali) o la tecnica encoder-decoder attraverso reti neurali. La seconda, utilizzando reti neurali ha il vantaggio di risultare più flessibile ed è usata in casi non lineari: la sua prima applicazione è stata ricavare la velocità in prossimità della parete di un flusso in un canale turbolento [12].

Un ulteriore aiuto nella ricerca dei modelli può essere dato dall'applicazione degli algoritmi di raggruppamento e classificazione, i primi sono già stati utilizzati per riconoscere e separare le regioni di un flusso con caratteristiche diverse [2], mentre la classificazione è spesso utilizzata per riconoscere e individuare i comportamenti e i regimi di flusso, di cui conosciamo già la fisica che li governa, o per identificare disturbi e rumori.

La riduzione di dimensionalità ha dato nuova linfa al campo dell'apprendimento automatico riducendo i costi degli algoritmi, tuttavia presenta ancora delle limitazioni che escludono la risoluzione di alcuni tipi di problemi. Infatti, le tecniche di decomposizione propria ortogonale POD

non sono in grado di catturare con buona accuratezza transitori e fenomeni intermittenti. Questi problemi sono però risolvibili attraverso le reti neurali e il deep learning. A causa degli elevati costi computazionali e delle grandi quantità di dati, le potenzialità di queste tecniche sono ancora poco esplorate ma da qualche decennio a questa parte stanno trovando applicazione nello studio delle turbomacchine, dei trasferimenti di calore, della turbolenza e varie applicazioni aeronautiche [8].

I fenomeni fluidodinamici che è possibile e utile studiare attraverso tecniche di apprendimento automatico sono molti, ma finora solo per alcuni di essi si stanno sfruttando tutte le potenzialità del machine learning, nei paragrafi successivi ne vengono presentati due: il controllo dei flussi (paragrafo 5.2) e i flussi turbolenti (paragrafo 5.3).

5.2 Ottimizzazione e controllo dei flussi con l'apprendimento automatico

La riduzione della resistenza e il controllo dei flussi sono tecniche di eccezionale interesse per l'industria. Il controllo su un flusso ne modifica il comportamento attraverso sistemi mossi da sensori, ed è necessario in risposta a instabilità, rumori e disturbi che possono modificare drasticamente il moto. È però fondamentale che il controllo avvenga con il ritardo minore possibile per far sì che la correzione possa essere efficace e non scateni reazioni non volute [12].

La resistenza aerodinamica è circa responsabile del 20% dell'energia persa nel moto di un moderno veicolo pesante su ruote, ed è anche la principale fonte di perdite di energia per tutti i velivoli.[9] La resistenza danneggia anche il moto di alcuni animali ed è la natura stessa a suggerirci la possibilità di ridurre le perdite ricercando l'ottimizzazione: ad esempio, è ormai noto che gli squali, quelli in grado di nuotare ad alte velocità, hanno la pelle con delle zigrinature in grado di interagire con lo strato limite intorno allo squalo stesso, riducendo la resistenza dell'acqua del 9% [13].

L'ottimizzazione e il controllo dei flussi sono due campi di studio che, con l'aumento delle capacità informatiche, stanno diventando sempre più indistinguibili: se si avesse la capacità di risolvere un problema di ottimizzazione con una velocità tale da controllare completamente un volo, allora questo sarebbe certamente anche una regola sul controllo del mezzo, facendo sì che lo stesso algoritmo possa essere interpretato indifferentemente come risoluzione di uno dei due problemi [14]. Purtroppo, ricercare una strategia per applicare il controllo attivo su un flusso è un problema alquanto complicato; utilizzare semplicemente delle misure di pressione e velocità intorno al nostro oggetto potrebbe anche risultare utile, ma ripresenta nuovamente il problema dell'altissimo costo computazionale, che può essere visto anche come lentezza dell'algoritmo, andando così a contrastare con la definizione di controllo attivo. Per questo motivo non possono essere utili tutti gli algoritmi

che fanno dell'iterazione il loro punto di forza, che però possono essere usati per ricercare soluzioni a tipi di problemi in cui non è richiesta una velocità notevole, o dove è possibile utilizzare più architetture in parallelo. Utilizzare una rete neurale per applicare il controllo di un flusso richiederebbe esagerate quantità di dati per istruire adeguatamente l'algoritmo, ma soprattutto, all'aumentare dei parametri cresce significativamente il tempo di apprendimento, rendendo questo tipo di approccio impossibile in un futuro prossimo [2].

Un suggerimento sul tipo di architettura che può risultare vantaggiosa per questo tipo di problemi ci viene nuovamente fornito dalla natura: pesci, uccelli, animali in genere e lo stesso essere umano apprendono a spostarsi nello spazio grazie a sistemi che supervisionano il movimento, come la vista per gli uccelli e gli umani o la linea laterale sensoriale dei pesci, che trasmettono i dati come dei sensori, lasciando al cervello la decisione su cosa sia giusto e sbagliato, facendo compiere successivamente il movimento corretto. Principio che ricalca esattamente la teoria utilizzata dall'apprendimento rinforzato. Fornendo all'algoritmo le conoscenze di fluidodinamica esso può, attraverso le ricompense dovute alle azioni svolte correttamente, implementare queste conoscenze comportandosi esattamente come potrebbe fare la linea laterale di un pesce.

5.2.1 Il primo algoritmo ad apprendimento per rinforzo per il controllo di flussi

Esempio di come un algoritmo ad apprendimento per rinforzo possa essere utilizzato per il controllo dei flussi e per la riduzione della resistenza è uno studio pubblicato nel 2018 da Rabault, Kuchta et al.: gli autori sono stati i primi a provare realmente l'efficacia di un algoritmo per rinforzo su un problema di controllo attivo. Lo studio si basa sulla ricerca, attraverso un appropriato utilizzo del machine learning, di un sistema di controllo in 2D che possa ridurre la resistenza di un cilindro investito da un flusso. Il sistema di controllo è attuato tramite due getti posizionati simmetricamente sopra e sotto il cilindro, attaccati alla parete dello stesso, come rappresentato dai puntini rossi in figura 5.1.

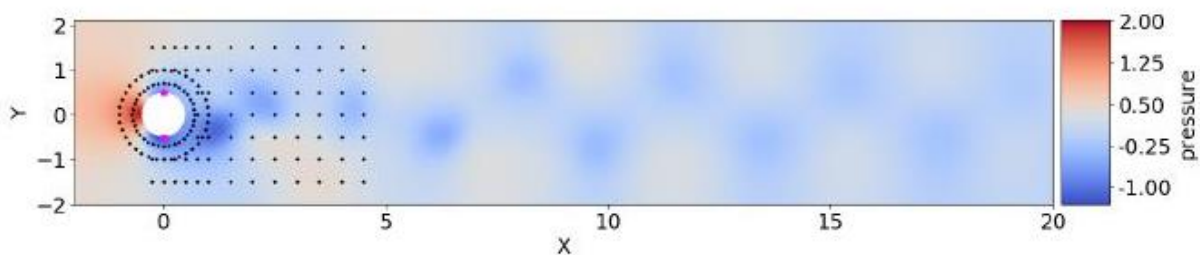


Figura 5.1 Rappresentazione adimensionalizzata del cilindro in fase di inizio simulazione, senza controllo attivo. I punti rossi indicano la posizione dei getti del sistema di controllo. La posizione delle sonde di velocità è indicata dai punti neri [15]. Il flusso che investe il cilindro proviene da sinistra.

L'architettura utilizzata è composta da una rete neurale a due livelli nascosti di 512 unità base, un livello di input e uno di output (simile alla figura 4.3); questa configurazione è stata trovata empiricamente: l'utilizzo di architetture più grandi o più piccole è risultato meno accurato di questa [15]. L'agente, ossia la parte del programma che riceve le ricompense, è stato scelto in modo da ridurre la complessità matematica del programma e per evitare tempi di apprendimento piuttosto lunghi. La funzione di ricompensa scelta non prevede l'uso di sanzioni in caso di risultati errati, ma solo di ricompense se l'algoritmo funziona correttamente, perché complicherebbe troppo il programma allungando i tempi d'apprendimento.

L'agente è stato in grado di elaborare una strategia di controllo dopo 24 ore di apprendimento in un computer moderno, tempo che può essere ancora ridotto utilizzando altri calcolatori in parallelo [15].

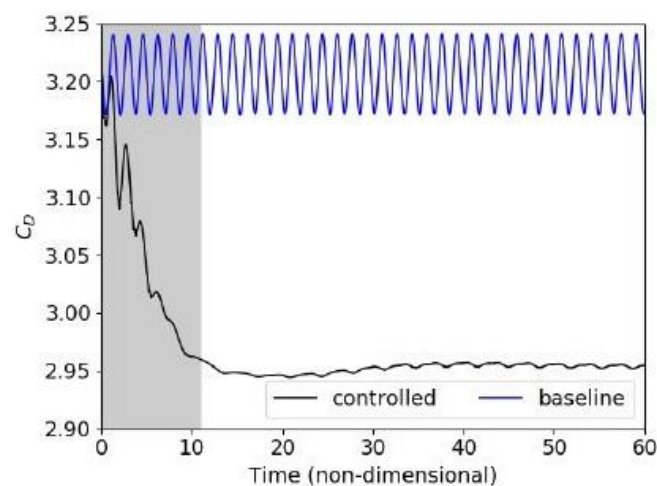


Figura 5.2 Coefficiente di resistenza in funzione del tempo: la linea in blu rappresenta l'andamento senza controllo, quella in grigio con il controllo attivo [15].

Nella simulazione senza controllo la resistenza ha un andamento periodico, dettato dal movimento dei vortici, e il valore medio del coefficiente di resistenza $\langle C_D \rangle \approx 3.205$ con un'ampiezza di 0.034. Invece l'andamento del coefficiente di resistenza con controllo attivo non ha più carattere periodico, ma si riduce fino a stabilizzarsi vicino al valore $\langle C_D' \rangle \approx 2.95$ (figura 5.2), ossia con una diminuzione quasi dell'8% [15].

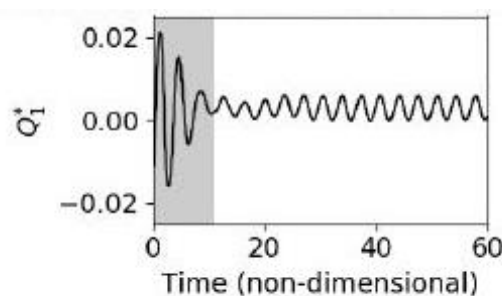


Figura 5.3 Andamento del flusso di massa nel getto superiore rispetto al tempo [15].

Questo studio ha quindi dimostrato come si possa applicare l'apprendimento per rinforzo ai problemi di controllo dei flussi. Pure essendo un'analisi in due dimensioni, con una maggiore potenza di calcolo è possibile studiare il problema reale in tre dimensioni e valutarne applicazioni tecniche. Ad esempio, gli autori suggeriscono che possa gettare le basi per lo studio di sistemi di controllo per la riduzione della resistenza in piccole barche.

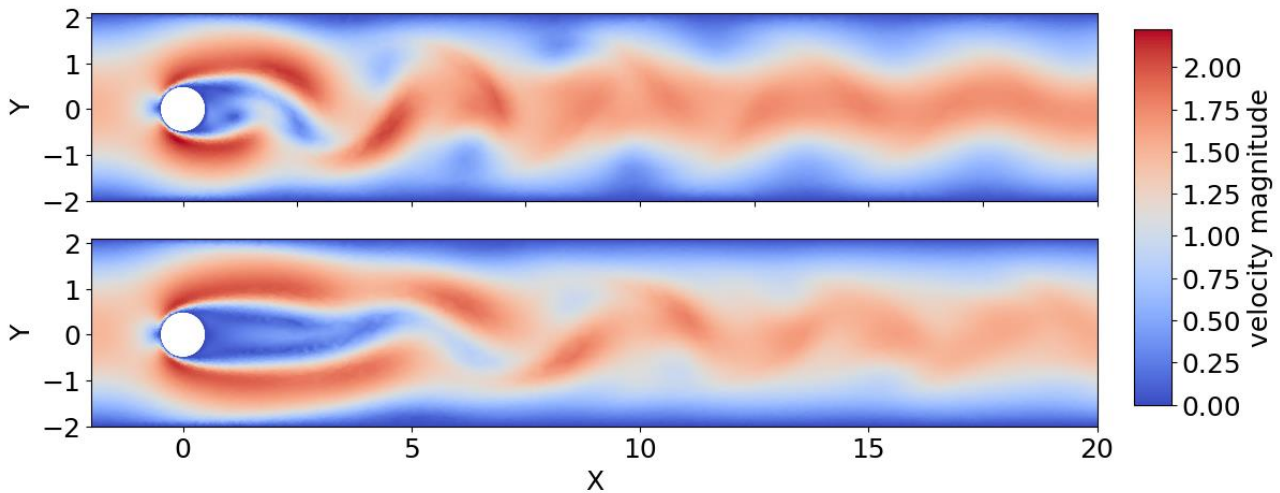


Figura 5.4 Differenza di velocità nei casi senza controllo (sopra) e con il controllo (sotto) [15].

5.3 Studio dei flussi turbolenti con gli algoritmi di apprendimento automatico

L'analisi dei flussi turbolenti è uno dei più importanti campi di studio della fluidodinamica moderna. La complessità del problema stesso ha come conseguenza diretta il tentativo di utilizzare i metodi più disparati per giungere ad una soluzione generale. La maggior parte degli studi nel campo hanno come scopo l'individuazione di correlazioni e modelli comuni, i quali possono avere sia natura geometrica che statistica [23]. Dato lo scopo comune, gli algoritmi di apprendimento automatico si prestano bene alle applicazioni nel campo della turbolenza, come predetto da Kolmogorov negli anni '40. Molte delle leggi utilizzate per studiare i flussi turbolenti sono state ottenute traendo ispirazione dall'analisi di grandi quantità di dati e i concetti che sono derivati da essi potrebbero risultare più chiari attraverso i metodi del machine learning, rendendo questo tipo di analisi automatizzata. Inoltre, l'impiego di questi algoritmi potrebbe risultare fondamentale per analizzare ad un livello più profondo le relazioni tra i diversi termini delle equazioni di governo per risolvere il problema della chiusura della turbolenza. Per alcune applicazioni è stato proposto l'utilizzo delle equazioni di Navier Stokes mediate alla Reynolds (RANS) con lo scopo di rappresentare il tensore degli sforzi utilizzando una rete neurale con 8-10 livelli nascosti [8]. Questa architettura si è dimostrata molto più accurata nelle soluzioni rispetto ai modelli classici con le

RANS in differenti test, ed è stata in grado di prevedere i comportamenti in un condotto nonostante fosse tarata con esempi a numeri di Reynolds molto diversi da quelli utilizzati.

5.3.1 Confronto delle soluzioni di diversi algoritmi per un flusso turbolento di taglio

Per esaminare il potenziale degli algoritmi ad apprendimento automatico viene presentato uno studio effettuato confrontando l'analisi di un semplice flusso turbolento di taglio attraverso tre tipi di algoritmi diversi: una rete neurale profonda (nello studio indicata come MLP multi-layer perceptron), una rete neurale ricorrente LSTM e una echo state network ESN, una rete simile a una RNN ma nella quale lo strato nascosto è scarsamente collegato e gli unici pesi modificabili sono quelli che collegano gli stati nascosti con lo strato di output, mentre gli altri sono fissi e casuali. Per semplificare il flusso esso viene descritto trasportando la soluzione da uno spazio continuo a uno discreto tramite i metodi di Galerkin, riducendo così gli output a nove coefficienti soluzione. Il campo di velocità viene dunque descritto dalla formula $\mathbf{u}(\mathbf{x}, \mathbf{t}) = \sum_{k=1}^9 \mathbf{a}_k(\mathbf{t}) * \mathbf{u}_k(\mathbf{x})$ dove $\mathbf{u}_k(\mathbf{x})$ descrive la dipendenza dallo spazio e $\mathbf{a}_k(\mathbf{t})$ sono i 9 coefficienti che governano la dinamica del flusso. L'obiettivo è ottenere una soluzione del metodo di Galerkin attraverso i tre algoritmi precedentemente citati.

Tutti e tre gli algoritmi hanno svolto 100 serie da 4000 step di apprendimento. Le dimensioni dei dati in input dipendono dal tipo di architettura dell'algoritmo, come riferito in figura 5.5.

Parameter	MLP	LSTM	ESN
Input dimension	4500	90	9
Output dimension	9	9	9
Hidden layers	4	1	
Neurons in hidden layers	90	90	–
Reservoir nodes	–	–	600
Activation function	Hyperbolic tangent	Hyperbolic tangent	Hyperbolic tangent

Figura 5.5 Parametri dei diversi algoritmi utilizzati per lo studio [23].

Computazionalmente la rete neurale profonda è la più onerosa mentre la ESN risulta la più veloce a svolgere la fase di apprendimento [23], avendo dimensioni di input nettamente minori delle altre. La rete neurale ricorrente presenta un errore minore rispetto alle altre due nella fase di training (figura 5.6).

Dopo aver svolto la fase di apprendimento agli algoritmi viene somministrato un flusso di cui è nota la soluzione (nelle figure indicata con GT, ground truth) e vengono analizzati i risultati.

Network	Training loss	Validation loss	Training time (minutes)	$E(\bar{u}_x)$ (%)
MLP	2.0×10^{-5}	3.1×10^{-5}	$O(10^2)$	12.24
LSTM	2.8×10^{-8}	3.6×10^{-8}	$O(10^1)$	1.01
ESN	2.0×10^{-7}	2.4×10^{-7}	$O(10^0)$	3.96

Figura 5.6 Risultati della fase di training delle tre architetture comparate [23].

Le predizioni effettuate nei parametri a_1 e a_9 mostrano come nessuno dei tre algoritmi si avvicini all'andamento esatto dei parametri reali nel tempo (figura 5.7), tuttavia i risultati generali si accostano alla soluzione esatta (figura 5.8) e la ragione di tali disparità può nascere alla natura fortemente caotica del modello, che comunque risulta semplificato rispetto a un normale flusso [23].

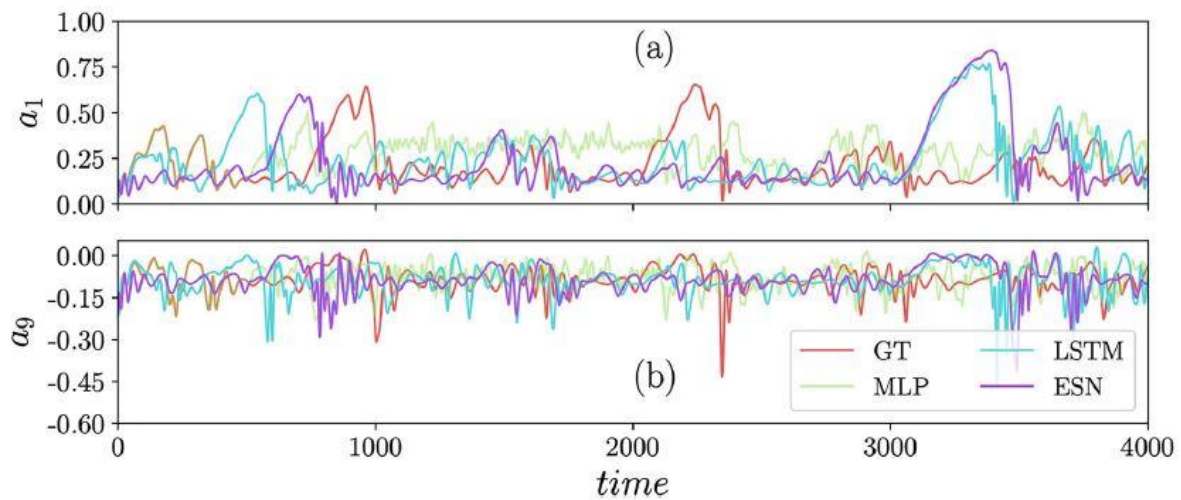


Figura 5.7 Confronto degli andamenti delle predizioni dei parametri a_1 e a_9 dei tre algoritmi con la soluzione reale GT [23].

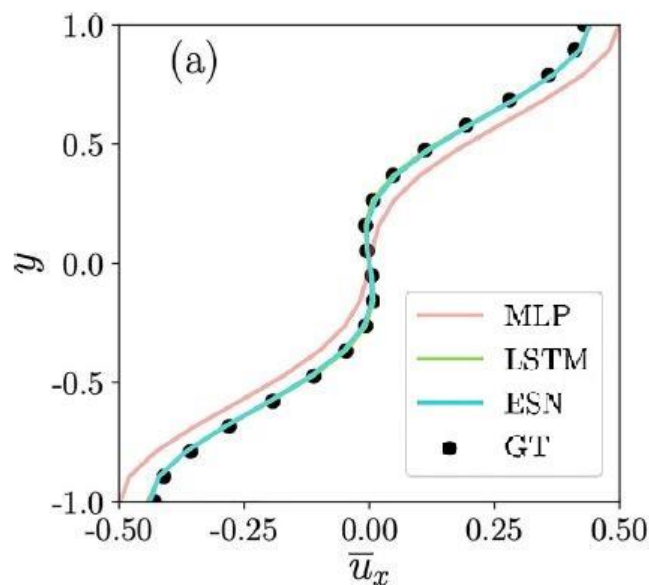


Figura 5.8 Confronto delle predizioni dei tre algoritmi con la soluzione reale della componente della velocità u_x [23].

6. CONCLUSIONI

La forza e la versatilità dell'apprendimento automatico stanno proiettando il campo in moltissime discipline le quali, a loro volta, hanno molteplici applicazioni possibili per questo nuovo tipo di approccio, per le quali può risultare vincente spostare il livello del problema in sé, cambiando gli interrogativi da porsi: la domanda non è più “come posso capire e/o risolvere il problema?”, ma diventa “come posso aiutare qualcun'altro (l'algoritmo) ad imparare a risolvere un problema di cui non conosco la soluzione?”.

La velocità con cui gli algoritmi si sviluppano è impressionante: fino a nemmeno un decennio fa gli assistenti vocali erano poco sviluppati e interagivano solamente con messaggi di testo, mentre oggi hanno una propria voce caratteristica e sono in grado di svolgere compiti piuttosto complessi; le auto a guida autonoma erano ancora in fase di progettazione mentre ora sono argomento all'ordine del giorno.

L'obiettivo da raggiungere è sicuramente quello di una maggiore comprensione degli algoritmi [2] e lo sviluppo di questi verso un migliore utilizzo dei dati, per ridurre i tempi e i costi computazionali dell'apprendimento dei programmi, ma anche imparare a formulare diversamente i problemi per renderli adeguati all'apprendimento automatico e capire come selezionare accuratamente i dati da utilizzare nella fase di training.

L'incontro tra il machine learning e la fluidodinamica ha ricevuto nuovo interesse nella comunità scientifica solo da pochi anni ed è importante intrecciare i legami tra i due partendo da ciò che finora è stato realizzato e che può incentivare la risoluzione di problemi che non siamo stati capaci di risolvere attraverso un metodo classico. Sono di recente sviluppo studi sul controllo della separazione del flusso sui profili alari, sull'attenuazione dei vortici d'estremità delle ali o, ancora, uno studio sull'ottimizzazione di forma di componenti meccanici con cavità rettangolari esposte a correnti, che può facilmente essere adattato ai carrelli dei velivoli [22]. È sicuramente interessante riuscire a capire come, partendo da questi studi di base si possa applicare il metodo a questioni più grandi, ad esempio la relazione tra il controllo 2D della scia del cilindro possa essere esteso e studiato per il controllo della scia di una barca.

Un aspetto che va però tenuto in conto è lo scopo principale della fluidodinamica, ossia imparare e capire i meccanismi dei fluidi, riconoscendone gli aspetti predominanti. Utilizzando solamente gli algoritmi ad apprendimento automatico, alcuni dei quali possono risolvere i problemi senza ricevere alcuna nozione di fisica, si rischia di venire meno al principio della scoperta fisica in sé, favorendo

l'utilizzo di questi mezzi come puri risolutori di quesiti posti in maniera congeniale ai propri algoritmi risolutori [24].

É anche interessante mettere in luce come finora tutti gli studi compiuti sugli algoritmi ad apprendimento automatico nel campo della fluidodinamica siano applicati a principi di cui abbiamo già una discreta conoscenza, speculando sui risultati ottenibili in campi non ancora del tutto conosciuti [24].

Ovviamente l'apprendimento automatico non può sostituire tutti gli altri metodi di studio precedentemente utilizzati, ma deve essere usato come un potente mezzo complementare in grado di aiutarci a comprendere e risolvere quelli di un certo tipo. La strada è ancora lunga ed è necessario il coinvolgimento di tutta la comunità scientifica per progettare uno sviluppo futuro, come ad esempio la creazione di un enorme disco online di dati ottenuti da esperimenti, simulazioni e ricerche varie da condividere apertamente con tutta la comunità scientifica, in modo che gli algoritmi possano attingere da esso per aumentare il loro training set [2] e aumentare la loro precisione, in analogia con quello che fanno molti programmi di riconoscimento fotografico attingendo da Google Immagini per effettuare i loro confronti.

“Il progresso deriva da un intelligente uso dell’esperienza”

Aforisma di E. Hubbard, suggerito dall’assistente vocale del mio pc.

BIBLIOGRAFIA

- [1] Dizionario Treccani, Intelligenza
- [2] Brunton, S.L., Noack, B.R., Koumoutsakos, P., Machine Learning for fluid mechanics; Annual Review of Fluid Mechanics, 50 (477-508), 2020
- [3] Brunton, S.L., Hemati, M.S., Taira, K, Special issue on machine learning and data-driven methods in fluid dynamics. Theor. Comput. Fluid Dyn., 34 (333–337), 2020
- [4] Yaglom, A.M., A.N. Kolmogorov as a fluid mechanician and founder of a school in turbulence research; Annual Reviews of fluid mechanics, 26 (1-23), 1994
- [5] Percettrone, Enciclopedia della Scienza e della Tecnica Treccani, 2008
- [6] Cherkassky, V., Mulier, F.M., Learning from data; John Wiley & sons inc, 1998
- [7] Maltoni, D., Dispense, Machine Learning; *UniBo*
- [8] Kutz, J.N., Deep learning in fluid dynamics; Journals of fluid mechanics, Cambridge University Press, 814, (1-4), 2017
- [9] Kubat, M., An introduction to machine learning; Springer international publishing, 2017
- [10] Vapkin, V., The nature of statistical learning theory, Springer, 1995
- [11] Sistema nervoso-Neurone, Enciclopedia Treccani
- [12] Muller, S., Milano, M., Koumoutsakos, P., Application of machine learning algorithms to flow modeling and optimization; Center for Turbulence Research Annual Research Briefs, 1999
- [13] Dean, B., Bushan, B., Shark-skin surfaces for fluid-drag reduction in turbulent flow: a review. Philosophical transactions. Series A, Mathematical, physical, and engineering sciences, 2010
- [14] Tsiotras, P., Mesbahi, M., Toward an algorithmic control theory. Journal of guidance, control and dynamics, 40 (194) 2017
- [15] Rabault, J., Kuchta, M., Jensen, A., et al., Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control; arXiv:1808.07664 (physics) Cornell University, 2018
- [16] Minini, A., Machine Learning, 2020
- [17] Musiol, M.J., Speeding up Deep Learning Computational Aspects of Machine Learning, 2016
- [18] Tesauro, G., Temporal Difference Learning and TD-Gammon; Association for Computing Machinery, 1995
- [19] Coronavirus in Italia, 10.149 casi e 631 morti: il bollettino del 10 marzo; *Corriere della Sera* 10 marzo 2020
- [20] Pereira, Frame, Knoow Enciclopedia tematica

- [21] Yan, M., Liu, K., Guan, Z., Background Augmentation Generative Adversarial Networks (BAGANs): Effective Data Generation Based on GAN-Augmented 3D Synthesizing; *Symmetry*, 10 (734), 2018
- [22] Taira, K., Hemati, M.S., Brunton, S.L., et al; Modal analysis of Fluid Flows: Applications and Outlook; *AIAA Journal*, 58, (998-1022), 2020
- [23] Pandey, S., Schumacher, J., Sreenivasan, K.R., A perspective on machine learning in turbulent flows, *Journal of Turbulence*, 2020
- [24] Brenner, M.P., Eldredge, J.D., Freund, J.B., Perspective on machine learning for advancing fluid mechanics, *Physical review fluids* 4, 2019

Desidero utilizzare questo spazio per ringraziare tutti coloro che, direttamente o indirettamente, mi hanno permesso di raggiungere questo primo importantissimo traguardo.

Un sentito ringraziamento alla relatrice, professoressa Scarsoglio che, oltre ad aver seguito attentamente e con pazienza la stesura di questo elaborato, mi ha permesso di scoprire il mondo del machine learning dal quale sono rimasto intimamente affascinato.

Ringrazio profondamente mia madre Maria, che regalandomi il mio primo libro mi ha avviato allo studio e per l'incontrastata fiducia che ha riposto in me in tutti i gli anni scolastici e universitari.

Ringrazio infinitamente mio padre Enzo per aver eretto il basamento su cui si fonda la mia vita e per la passione che mi ha trasmesso verso gli aspetti pratici, essenziale per la scelta del mio percorso di studi.

Ringrazio mia sorella Serena, il più grande regalo che mi abbiano fatto i miei genitori, per condividere con me amarezze e gioie.

Un ringraziamento particolare ai miei Nonni per avermi appoggiato nei momenti bui, e non solo, illuminando la strada verso l'obbiettivo.

E, infine, un grazie speciale a tutti gli amici, in particolare Sofie, Lorenzo, Martina, Anna, Giacomo e Matteo per aver reso più sereno il mio percorso.