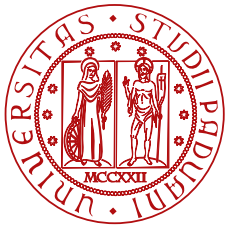

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Laurea Triennale in Ingegneria Informatica

Reti neurali convoluzionali per lo studio di varianti non codificanti in sequenze genomiche

Laureando

Alessandro Trigolo

Matricola 2043049

Relatore

Prof.ssa Cinzia Pizzi

Università degli Studi di Padova

Anno Accademico
2023/2024

Sommario

Questo elaborato mira ad approfondire il funzionamento delle reti neurali convoluzionali e di come questi modelli di deep learning siano in grado di estrarre significative informazioni da sequenze genomiche, analizzandone le zone non codificanti. In particolare, verranno comparati tre tool basati sulle CNN — DeepSEA, Basset e DeepSATA — e sarà fornita una revisione delle loro prestazioni.

Abstract

This thesis aims to deepen the understanding of the functioning of convolutional neural networks and how these deep learning models are able to extract significant information from genomic sequences, analyzing their non-coding regions. In particular, three CNN-based tools — DeepSEA, Basset and DeepSATA — will be compared and a review of their performance will be provided.

Indice

Indice delle Figure

Indice delle Tabelle

Lista degli Acronimi

AI Intelligenza artificiale (*Artificial Intelligence*)

ANN Rete neurale artificiale (*Artificial Neural Network*)

ATP Adenosintrifosfato (*Adenosine TriPhosphate*)

DL *Deep Learning*

DNA Acido desossiribonucleico (*DeoxyriboNucleic Acid*)

GD *Gradient Descent*

GPU Unità di elaborazione grafica (*Graphics Processing Unit*)

mRNA RNA messaggero

NN Rete neurale (*Neural Network*)

RNA Acido ribonucleico (*RiboNucleic Acid*)

SVM *Support Vector Machines*

UTR Regione non tradotta (*UnTranslated Region*)

1 Introduzione

Ad oggi l'avanzamento della genomica — ramo della biologia molecolare che si occupa di studiare il genoma degli esseri viventi — si è rivelato notevolmente significativo al fine di approfondire e comprendere malattie legate alle mutazioni del genoma degli individui. Si stima che solamente una percentuale tra l'1% e il 2% del DNA contiene i *geni*, ovvero particolari regioni che contengono tutte le informazioni necessarie per la sintesi degli aminoacidi che poi comporranno le proteine [1], [2]. Ciò nonostante, la quasi totalità dei disturbi genomici è dovuta alle mutazioni nelle regioni non codificanti [3] — dette *varianti non codificanti*. Le mutazioni in queste zone del genoma, che apparentemente svolgono funzioni marginali, sono responsabili dello sviluppo di disturbi importanti, come le *malattie mendeliane*[1], l'epilessia, malattie cardiovascolari e soprattutto tumori — tra cui il cancro del colon-retto e il tumore al seno [3]–[11].Risulta quindi vitale continuare a studiare gli effetti che le varianti non codificanti in sequenze genomiche hanno sugli individui.

Negli ultimi decenni, il progredire delle tecniche di *sequenziamento* [12] ha dato uno slancio rilevante allo sviluppo della *bioinformatica* — disciplina che unisce informatica e biologia. La bioinformatica si interessa a organizzare dati biologici in modo tale da facilitare l'accesso e l'inserimento di nuove informazioni (come il *PDB* [13]), sviluppare *tool* che permettono l'analisi dei dati e infine fornire una interpretazione significativa dei risultati ottenuti [14]. Più recentemente, l'accrescimento dei dati biologici e il costante avanzamento della potenza di calcolo hanno reso possibile l'applicazione di tecniche di *deep learning* (DL) anche nel campo della bioinformatica. Questo notevole progresso consente di scoprire e perfezionare soluzioni informatiche che permettano di delineare con sempre maggior precisione il ruolo che hanno le mutazioni nelle regioni non codificanti del DNA. Grazie a queste nuove tecnologie, la *genomica funzionale* — area della genomica che si interessa a descrivere le relazioni che ci sono tra i componenti di un sistema biologico, come geni e proteine [15] — ha avuto un forte impulso nell'approfondire le varianti non codificanti, tuttavia rimangono ancora significative lacune

[1] Le malattie mendeliane, causate dalla mutazione di un singolo gene, includono la fibrosi cistica e il morbo di Huntington.

nella comprensione della relazione tra mutazioni genetiche ed espressione genica. L'utilizzo di tecniche di deep learning risulta quindi cruciale per continuare la ricerca in questo ambito. L'obiettivo di questo elaborato è di discutere e confrontare tre tool che utilizzano le *reti neurali convoluzionali* per predire l'effetto delle varianti non codificanti su sequenze genomiche: DeepSEA [16], Basset [17] e DeepSATA [18].

Più precisamente, il Capitolo 2 introdurrà le basi della biologia molecolare, necessarie per comprendere interamente l'importanza delle varianti non codificanti. Successivamente, nel Capitolo 3 saranno approfonditi i principi fondamentali delle reti neurali e il modo in cui le reti convoluzionali possono essere utilizzate come ottimo strumento per predire l'effetto di sequenze genomiche. Il Capitolo 4 invece esaminerà i dettagli implementativi di ciascuno dei tre tool, indagando principalmente sugli aspetti legati alla codifica delle sequenze, alla struttura della rete e al *dataset* utilizzato per allenare il modello. Infine nel Capitolo 5 si riassumono le differenze analizzate nel capitolo precedente, offrendo una visione complessiva del confronto tra i tre tool.

2

Background biologico

La cellula è l'unità fondamentale della vita. La cellula è una piccola miscela acquosa con componenti chimici, racchiusi in una membrana, e possiede l'eccezionale capacità di replicarsi. Il primo elemento che permette di distinguere le cellule è la presenza di un nucleo. Vengono definite *procarioti* le cellule senza nucleo — che sono le più diffuse e compongono organismi unicellulari come i batteri e gli archei — mentre sono chiamate *eucarioti* le cellule che contengono un nucleo — le quali sono in genere più grandi e più complesse e costituiscono forme di vita multicellulari come animali, piante e funghi [19].

All'interno della cellula eucariote (Figura 2.1), immersi nel *citoplasma*, sono presenti diversi *organuli*, i quali svolgono una particolare funzione ciascuno. I *mitocondri* sono gli organuli più diffusi. Il loro compito è quello di generare energia chimica per la cellula: attraverso il processo di ossidazione di zuccheri e grassi, viene creata una sostanza che viene utilizzata nella

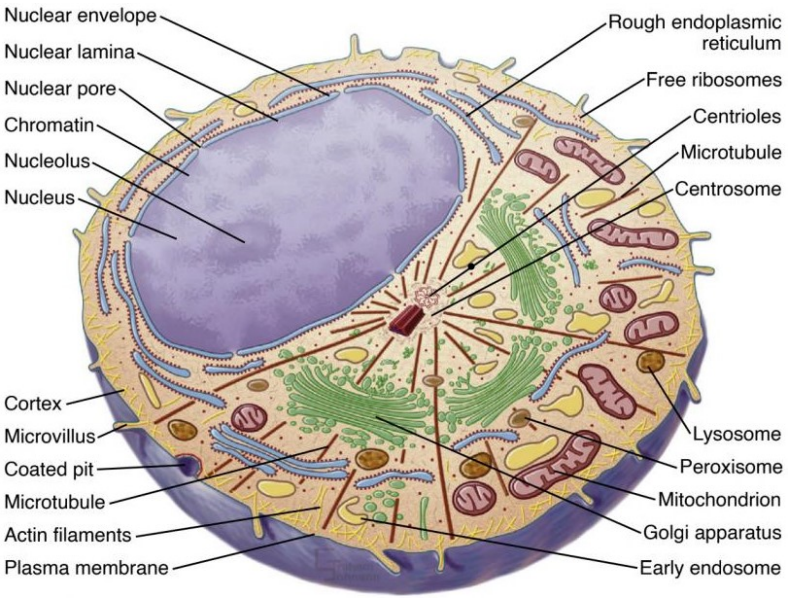

Figura 2.1: Rappresentazione schematica della cellula eucariote; si possono notare i principali organuli tra cui i mitocondri, lisosomi e perossiomi, il reticolo endoplasmatico, e il nucleo [2].

maggior parte delle attività cellulari[1]; questo processo è anche chiamato *respirazione cellulare* perché consumando l'ossigeno viene rilasciata anidride carbonica. Oltre ad essere la fonte energetica primaria della cellula, i mitocondri hanno anche importanti ruoli nella regolazione del metabolismo, del ciclo cellulare, delle risposte antivirali e anche della morte della cellula [19]–[21].

Il *reticolo endoplasmatico* è invece un organulo molto esteso e svolge molteplici funzioni. Tra questi compiti rientrano quelli di traslocazione di proteine e il ripiegamento delle proteine (*protein folding*) [19], [22]. I *lisosomi* si occupano di degradare e riciclare gli scarti cellulari e giocano un ruolo fondamentale per l'omeostasi della cellula[2], il suo sviluppo e il suo invecchiamento [23]–[25]. Infine, i *perossiomi* sono delle piccole vescicole che forniscono un ambiente protetto per gestire molecole tossiche come gli acidi grassi i quali sono smaltiti tramite la β-ossidazione [19], [26].

L'organulo più importante della cellula rimane il *nucleo*. Racchiuso nell'*involucro nucleare*, all'interno di questo organulo sono presenti tutte le informazioni genetiche, racchiuse in una lunga molecola di acido desossiribonucleico (comunemente noto come DNA), che, una volta impacchettato forma il *cromosoma* [2], [19]. La molecola di DNA è una struttura a doppia elica formata da *nucleotidi*. Osservando la Figura 2.2, i nucleotidi sono composti a loro volta da tre elementi fondamentali: una *base azotata*, uno *zucchero* e un *gruppo fosfato*[3]. Le basi azotate

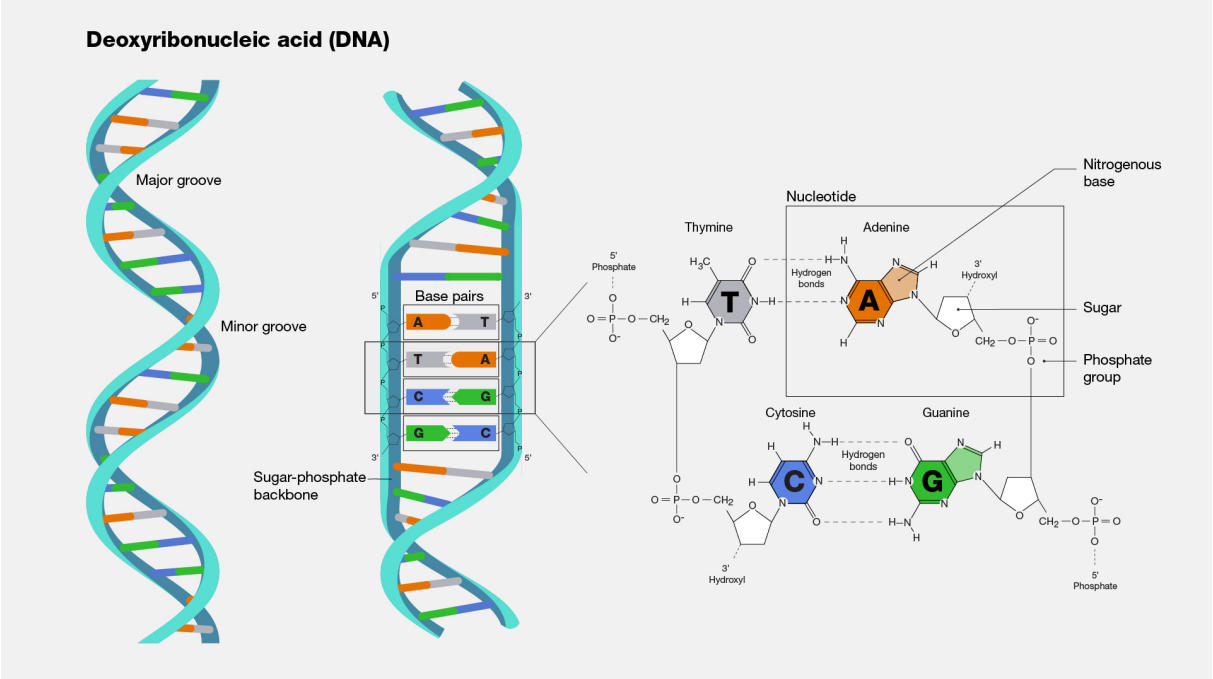

Figura 2.2: Rappresentazione schematica del DNA in cui si possono osservare le coppie di basi azotate, legate tra loro attraverso gli zuccheri e i gruppi fosfati [27].

[1]Questa sostanza è detta *adenosintrifosfato* o ATP ed ha una struttura simile ad un nucleotide: è infatti composta dall'Adenina, da uno zucchero e da tre gruppi fosfati.

[2]Con omeostasi cellulare si intende l'insieme di meccanismi necessari per mantenere ad un livello ottimale le funzioni della cellula.

[3]I gruppi fosfati hanno una carica negativa e forniscono alla molecola le proprietà di un acido.

sono quattro — Adenina (A), Citosina (C), Guanina (G) e Timina (T) — e si uniscono tra loro mediante dei legami ad idrogeno e secondo un preciso criterio: l'Adenina si lega solamente con la Timina (formando il legame *AT*) mentre la Citosina si unisce solo con la Guanina (creando la coppia *CG*) [1], [28]. Si osserva infine che il nucleotide di una coppia e quello successivo si legano mediante zucchero e gruppo fosfato sempre allo stesso modo: il gruppo fosfato di un nucleotide si lega sempre allo zucchero dell'altro. Di conseguenza, preso un filamento della doppia elica, le due estremità non sono uguali in quanto una termina con un gruppo fosfato (terminazione $5'$) e l'altra con uno zucchero (terminazione $3'$).

Attraverso una serie di ripiegamenti, una molecola di DNA lunga circa due metri riesce a raggomitolarsi in un cromosoma di grandezza inferiore a 2 micron (Figura 2.3). Il processo di *DNA-packaging* inizia avvolgendo la doppia elica di DNA attorno a delle proteine dette *istoni* e formando dei *nucleosomi*. In secondo luogo i nucleosomi si ammassano vicini tra loro formando una fibra, chiamata *cromatina* che, a sua volta si impacchetta su se stessa creando il cromosoma [29], [30].

2.1 Dogma centrale

La rilevanza del DNA è data delle informazioni essenziali che questa molecola contiene. Tali informazioni risiedono nei geni, che sono delle sequenze genomiche che codificano uno o più prodotti biologici operativi [32]. L'*espressione genica* è il processo che permette di utilizzare i dati contenuti nel gene per la creazione di macromolecole, come le proteine. Per esempio, le cellule della pelle a contatto con luce solare intensa possono esprimere geni che regolano la pigmentazione della pelle [33]. L'espressione genica è divisa in due fasi principali: la *trascri-*

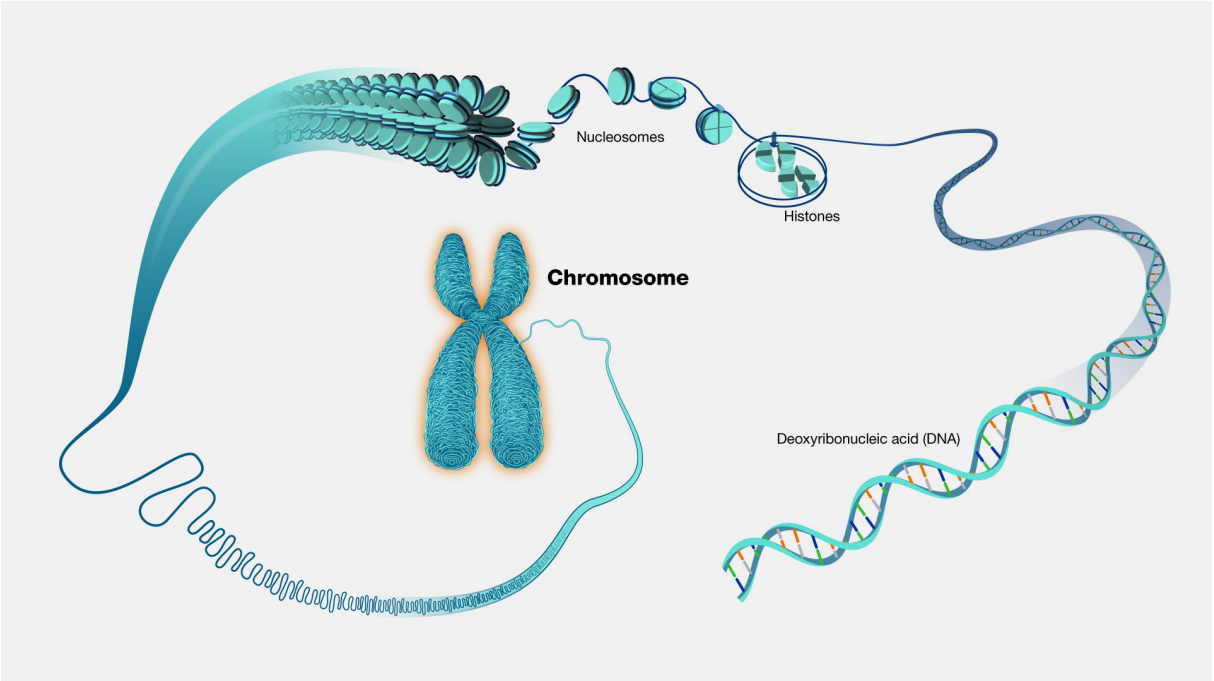

Figura 2.3: Il processo di impacchettamento del DNA che permette di compattare la struttura a doppia eilca nel cromosoma [31].

zione — che si occupa di produrre delle molecole di RNA che rispecchino il gene da esprimere — e la *traduzione* — la quale traduce le informazioni dell'RNA sintetizzando la proteina.

Nella prima fase dell'espressione genica, è necessario trascrivere il DNA in una molecola molto simile ovvero l'RNA — chiamato anche acido ribonucleico. Questa molecola differisce dall'acido desossiribonucleico per una base azotata — anziché la Timina è presente l'Uracile (U) — e per lo zucchero — da desossiribosio a ribosio [34]. La trascrizione del DNA in RNA inizia quando delle proteine, chiamate *fattori di trascrizione*, attratte dagli *enhancer* del DNA, riconoscono la regione che delimita l'inizio della molecola del gene da esprimere, detta *promoter*. Dopo aver riconosciuto l'inizio della sequenza, queste proteine permettono ad un enzima chiamato *RNA polimerasi* di attaccarsi ed aprire la doppia elica del DNA [35]. Una volta aperta la doppia elica, inizia la vera e propria trascrizione in RNA: il filamento del DNA viene preso come modello per la creazione dell'RNA; in particolare il nucleotide dell'RNA sarà il complementare rispetto a quello del DNA (di conseguenza $A \to U$, $C \to G$, $G \to C$ e $T \to A$). Così facendo l'acido ribonucleico viene creato un nucleotide alla volta, analizzando quello del DNA [34]. La trascrizione termina nel momento in cui gli enzimi e le proteine incontrano la regione terminatrice del gene che determina la separazione dal filamento e la terminazione dell'RNA *messaggero* (mRNA) che contiene le informazioni presenti nel gene da esprimere. L'intero processo di trascrizione è illustrato nella Figura 2.4.

Prima di uscire dal nucleo l'RNA messaggero subisce una serie di elaborazioni necessarie per rendere le informazioni immagazzinare sicure: diverse sono le malattie che emergono per mutazioni presenti nell'mRNA tra cui la distrofia miotonica[4] [37]. La prima elaborazione viene

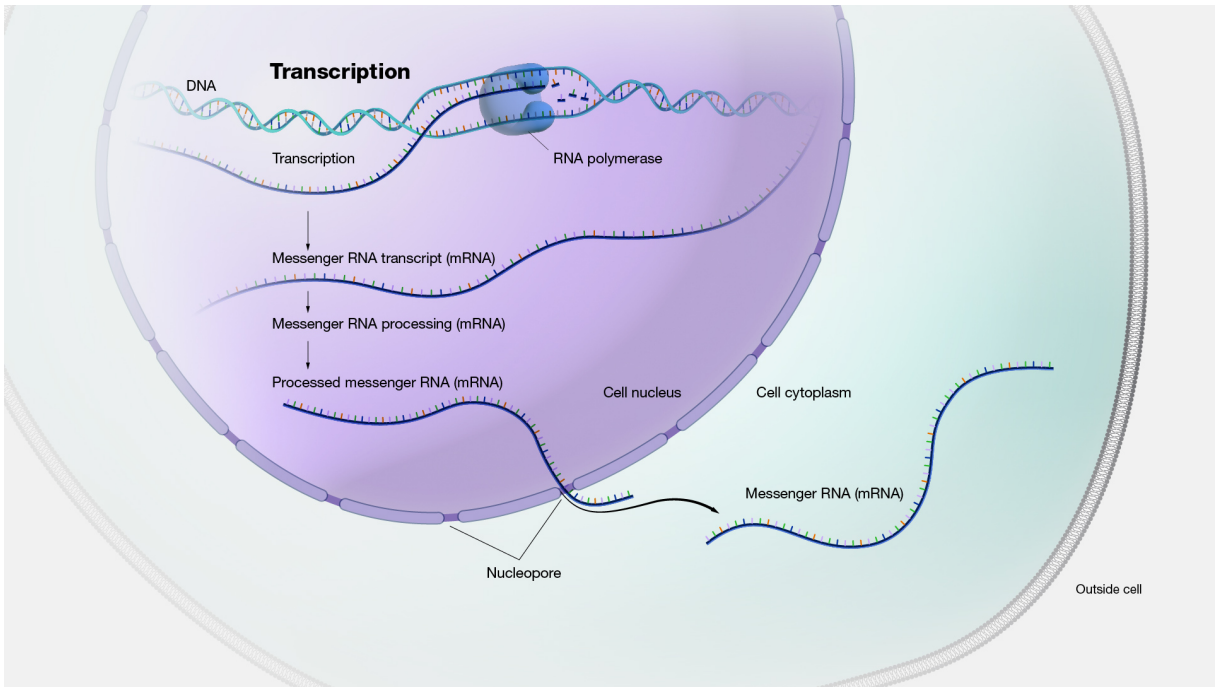

Figura 2.4: Il processo di trascrizione del DNA del gene in RNA mediante la RNA polimerasi [36].

[4]Le distrofie miotoniche sono patologie che colpiscono principalmente l'apparato muscolo-scheletrico.

chiamata 5′-*end capping* e si occupa di aggiungere alla terminazione 5′ dell'mRNA una Guanina attraverso un collegamento inusuale che garantisce maggiore stabilità alla molecola. In secondo luogo avviene lo *splicing* che si occupa di rimuovere le zone non codificanti — dette *introni*— dal gene trascritto mantenendo solo quelle che verranno utilizzate per essere sintetizzate in proteine — gli *esoni* — e quindi facilitando il processo di traduzione. Infine con il 3′-*end processing* viene aggiunta alla terminazione 3′ dell'mRNA una coda di Adenine — datta anche *poly*A *tail* — che, in maniera molto simile al 5′-*end capping* garantisce una stabilità del filamento di acido ribonucleico [38], [39].

Dopo essere uscito dal nucleo attraverso i *pori*, l'RNA messaggero raggiunge il citoplasma ed è pronto per iniziare la seconda fase dell'espressione genica, la traduzione. La traduzione non è altro che la traduzione dell'mRNA in un *polipeptide*, ovvero una sequenza di aminoacidi che compongono la proteina. Gli aminoacidi sono più di 20, di conseguenza anziché codificare un solo nucleotide dell'RNA messaggero, vengono codificati tre nucleotidi alla volta: questa tripletta viene chiamata *codone*. Durante la fase della traduzione, giocano un ruolo fondamentale i *ribosomi* i quali sono degli organuli nei quali avviene la traduzione. I ribosomi sono composti da due sotto unità, ciascuna delle quali ha tre siti per l'RNA di *trasporto* (*tRNA*). Delle due sotto unità del ribosoma, quella dimensionalmente minore si lega all'mRNA e agli *anticodoni* (sequenze specifiche di tre basi nel tRNA) e controlla che la traduzione avvenga con successo. La sotto unità più voluminosa invece si prende carico di catalizzare il legame peptidico tra l'aminoacido trasportato dal tRNA e la catena di aminoacidi in crescita [39]–[41]. In questo modo i ribosomi, analizzando codone dopo codone riescono a creare la catena polipeptidica mediante l'RNA di trasporto, come mostrato nella Figura 2.5.

Una volta creata la sequenza polipeptidica, inizia il processo di ripiegamento della proteina.

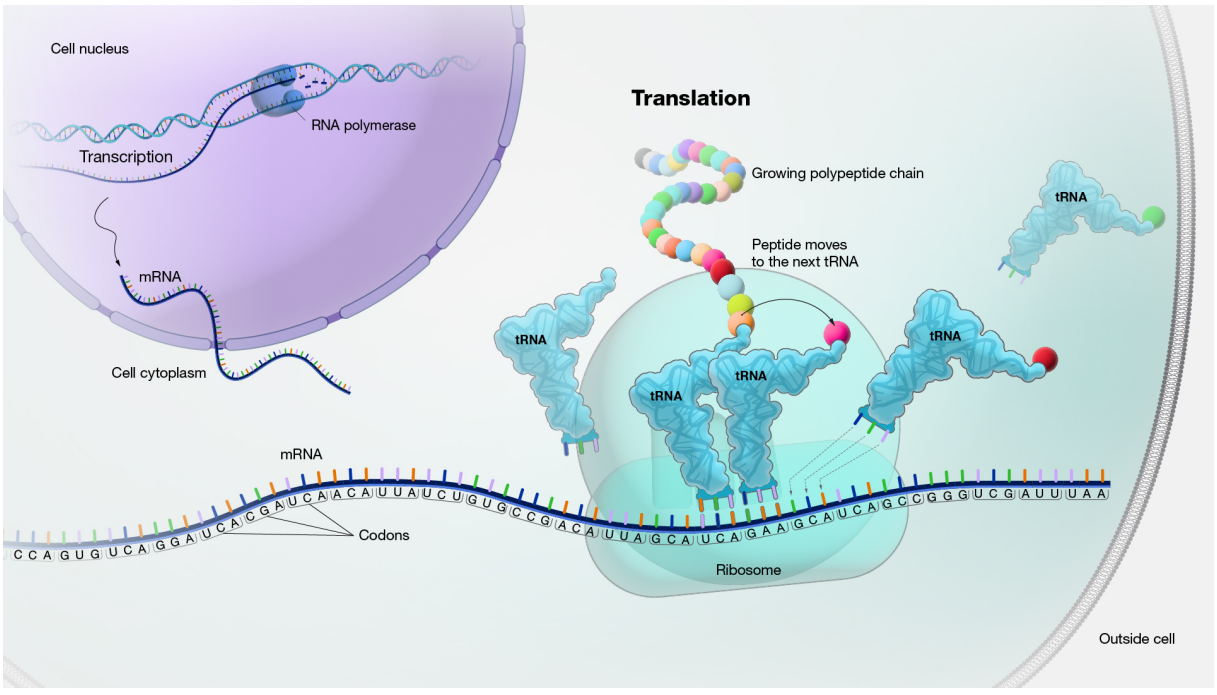

Figura 2.5: Il processo di traduzione da RNA messaggero a polipeptide attraverso il tRNA e i ribosomi [42].

In maniera molto simile a quanto visto per l'impacchettamento del DNA nel cromosoma, la sequenza di polipeptidi inizialmente si arrotola creando delle bobine che sono comunemente chiamate α-*helix*. Queste ultime poi si ripiegano nuovamente arrivando alla struttura terziaria della proteina, ovvero la proteina tridimensionale effettiva [43]. Una volta creata la proteina il gene è stato espresso definitivamente. Questo passaggio di informazioni dal DNA alla creazione della proteina è gergo definito come il *dogma della biologia molecolare*.

Come accennato all'inizio del capitolo, la cellula possiede la notevole capacità di replicarsi. In genere una cellula si duplica durante la crescita e lo sviluppo dell'organismo, quando deve essere rimpiazzata o rigenerata oppure nella riproduzione asessuata di alcuni micro organismi [44]. Il processo replicazione cellulare, chiamato *mitosi*, è preceduto dall'*interfase*, processo fondamentale in cui la cellula cresce di dimensioni e il DNA nei cromosomi si duplica, favorendo la replicazione cellulare. La mitosi può essere suddivisa in quattro fasi principali [44]–[47] le quali sono riassunte anche nella Figura 2.6:

1. Nella *profase* i cromosomi duplicati si condensano nel nucleo ed iniziano ad avvicinarsi dei microtuboli al nucleo, chiamati *centrosomi*; allo stesso tempo la membrana nucleare inizia a svanire;
2. Dopo che i microtuboli si sono attaccati ai cromosomi (fase intermedia detta *prometafase*) si giunge alla *metafase*, situazione in cui tutti i cromosomi sono allineati lungo la linea equatoriale della cellula;
3. Durante l'*anafase*, ciascuna coppia di cromosomi si divide e raggiunge i poli della cellula;
4. La fase finale della mitosi è la *telofase* nella quale le due cellule si dividono; le membrane nucleari delle due cellule si riformano attorno ai cromosomi divisi.

Affinché la mitosi abbia successo, è prima necessario duplicare il DNA all'interno della cellula. Il processo di replicazione di DNA che precede la mitosi è anche definito come *fase*

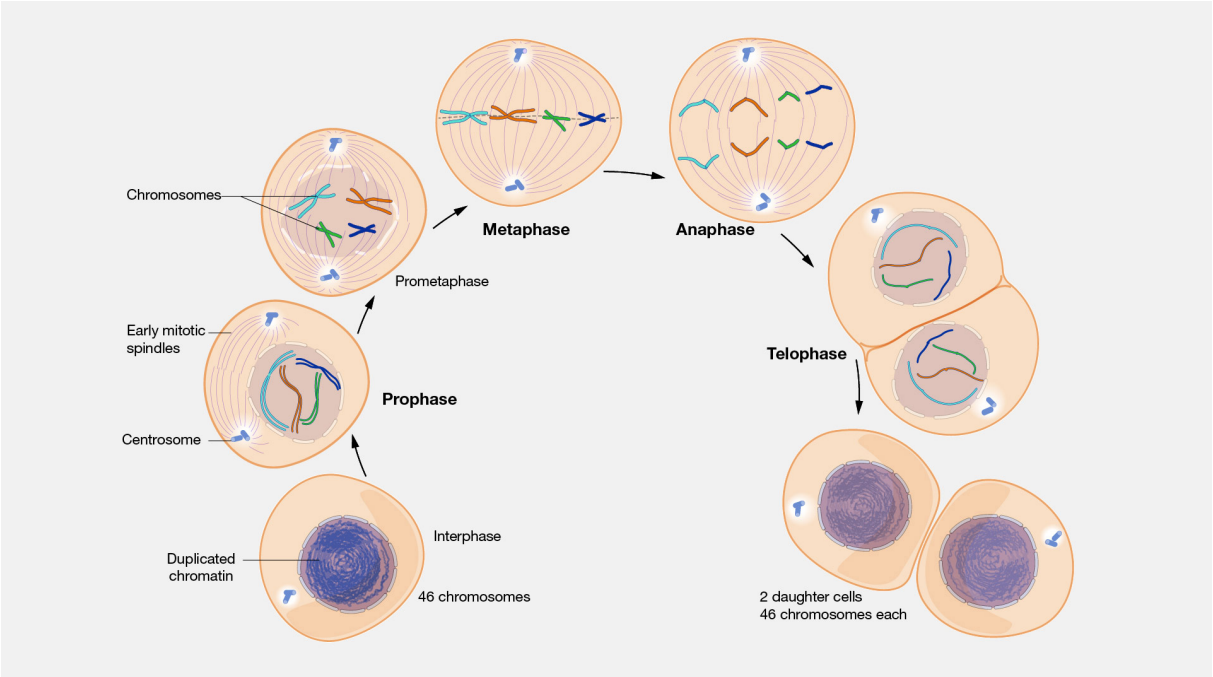

Figura 2.6: Rappresentazione delle quattro fasi che comprendono la mitosi cellulare [48].

di sintesi — S-phase. La duplicazione del DNA inizia con l'identificazione dell'*origine della replicazione*, ovvero una sequenza del DNA che specifica da quale punto della sequenza il DNA deve essere replicato (ci sono più di cento mila siti che segnalano un punto di orine nel DNA di una cellula). Una *proteina iniziatrice* è legata al punto di origine promuovendo l'attaccamento al DNA del *replisoma* che è composto da un enzima chiamato *elicasi* che si occupa di dividere i due filamenti di DNA procedendo nella direzione $5' \rightarrow 3'$. A questo punto il *RNA prime* inizia la sintesi del DNA favorendo l'attaccamento della *DNA polimerasi* entrambi i filamenti per duplicare il DNA. Essendo che il genoma è complementare, un filamento avrà un verso $5' \rightarrow 3'$ (*leading strand*) mentre l'altro filamento avrà verso opposto, $3' \rightarrow 5'$ (*lagging strand*). Di conseguenza, nel filamento concorde al replisoma, la polimerasi non incontrerà problemi nella duplicazione, invece nel filamento $3' \rightarrow 5'$ il DNA dovrà essere duplicato a segmenti, detti *frammenti di Okazaki* che verranno collegati tramite la *DNA ligasi* [49]–[52]. La Figura 2.7 racchiude quanto descritto sulla fase di sintesi.

2.2 VARIANTI NON CODIFICANTI

Come descritto fino ad ora, il ruolo del DNA è fondamentale in quanto trasmesso da cellula a cellula durante la replicazione per poi essere utilizzato nell'espressione genica creando le proteine. Alle regioni del DNA che prendono parte all'espressione genica, si contrappongono le regioni che non vengono codificate, in particolare gli enhancer ed i promoter, ma anche gli introni all'interno di un gene. Una mutazione in queste zone può causare diversi disturbi genetici in quanto l'espressione genica o il processo di replicazione del DNA possono essere compromessi.

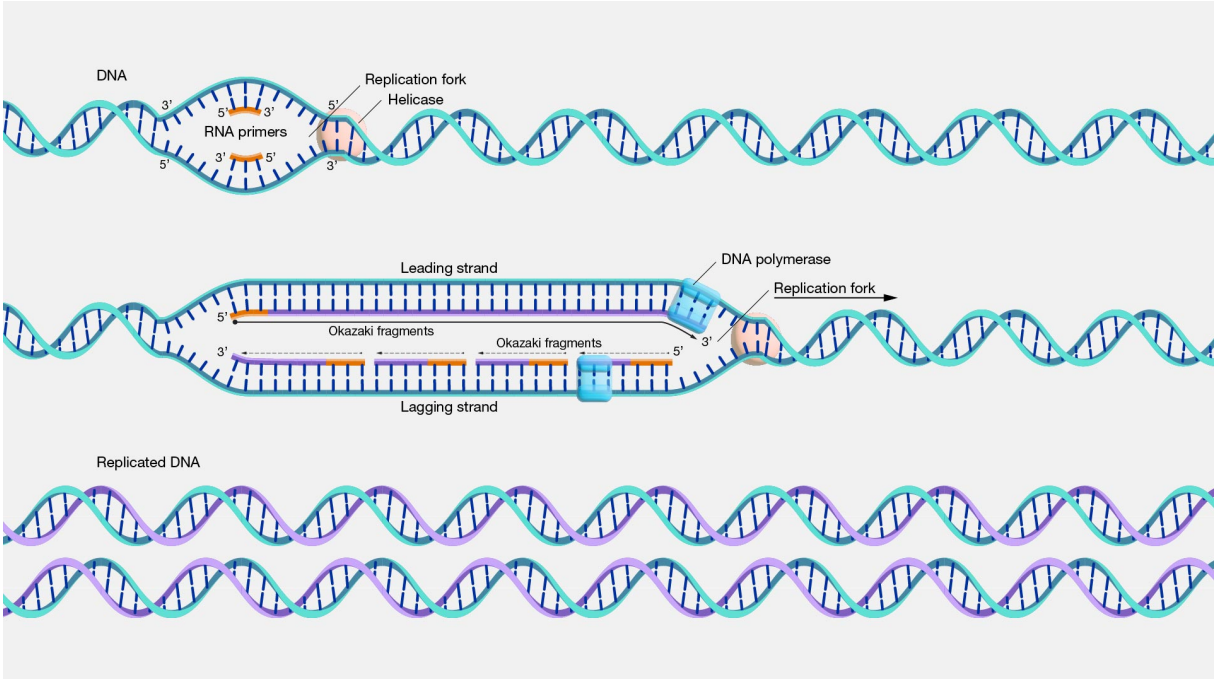

Figura 2.7: Il processo di replicazione del DNA durante la fase di sintesi [53].

Alcune varianti non codificanti possono alterare lo splicing di un gene trascritto. Come accennato precedentemente, lo splicing è il processo che permette la rimozione delle zone non codificanti del gene (gli introni) prima che questo sia tradotto in una proteine. Lo splicing fa parte dei processi che l'mRNA subisce prima di lasciare il nucleo ed iniziare la traduzione. All'interno degli introni sono presenti diverse sequenze che sottolineano l'inizio di un esone, tra cui il *donor*, l'*acceptor*, i *branch points* e i *tratti polipirimidinici*. Le variazioni in queste regioni possono condurre alla non codifica di una esone o alla conservazione di un introne: più del 15% dei disturbi ereditari sono dovuti proprio a questi inconvenienti. Oltre agli introni, anche altre regioni non tradotte (UTR) dell'mRNA possono condurre ugualmente a disturbi ereditari. Queste regioni sono fondamentali per gestire il processo seguente alla trascrizione: si occupano infatti di rendere il filamento di mRNA più stabile e robusto e rendono la sua localizzazione più semplice per poter iniziare il processo di traduzione [4].

[4]

spiega in maniera più approfondita la varianti non codificianti

3

Reti neurali

Il primo modello di intelligenza artificiale (AI) risale al 1943, dove E. McCulloch e W. Pitts cercarono di modellare un neurone come una semplice funzione predefinita. Nel modello, il neurone, generava un valore in output nel caso in cui le variabili booleane di input, una volta elaborate, superavano una soglia prestabilita [54, "A logical calculus of the ideas immanent in nervous activity"]. Poco dopo, nel 1950, Alan Turing, pubblicò un articolo che definiva una metodologia per testare l'intelligenza di un modello [55, "Computing machinery and intelligence"]. Questo test — noto anche come *The imitation game* — consisteva nel valutare se una macchina potesse imitare l'intelligenza umana tabilendo così un obiettivo per il campo dell'intelligenza artificiale, termine che venne conianto per la prima volta nella conferenza di Dartmouth nel 1956.

Dopo due anni, nel 1958, lo psicologo F. Resenblatt introdusse il *percettrone* che, a differenza del modello del '43, processava input non booleani e disponeva di pesi per bilanciare l'output [56, "The perceptron: a probabilistic model for information storage and organization in the brain."]. Anche se il percettrone sarà alla base delle reti neurali artificiali moderne, nei dieci anni successivi alla pubblicazione dell'articolo, le aspettative iniziali non vennero soddisfatte. Nel 1968 venne pubblicato un libro il quale analizzava le prestazioni del percettrone e constatava le forti limitazioni del modello, come l'impossibilità di risolvere problemi non linearmente separabili [57, "Perceptrons"]. In seguito ad un secondo articolo del 1973, dove si evidenziavano gli scarsi risultati ottenuti in paragone con le grandi aspettative, iniziò il *Primo Inverno dell'Intelligenza Artificiale* dove fino alla metà degli anni Ottanta molte organizzazioni governative smisero di finanziare la ricerca sull'Intelligenza Artificiale (AI). L'Inverno della AI terminò nel 1985 con l'introduzione del *Gradient Descent Optimization*, algoritmo che permetteva di aggiornare i pesi in modo tale da minimizzare l'errore in una rete. Un anno dopo venne introdotto l'algoritmo della *back-propagation*, fondamentale per lo sviluppo di reti neurali, costituite da più livelli di neuroni, ciascuno dei quali è collegato al livello successivo [58, "Learning representations by back-propagating errors"].

Nonostante il grande sviluppo nella parte algoritmistica, l'hardware non era computazionalmente prestante da supportare le richieste di calcolo delle reti neurali artificiali. Questa carenza

nella potenza di calcolo portò al *Secondo Inverno dell'Intelligenza Artificiale*, periodo in cui l'interesse scientifico si spostò su modelli che richiedevano meno potenza di calcolo, come le *Support Vector Machines* (SVM) introdotte nel 1963. Il Secondo Inverno della AI terminò a metà degli anni Novanta quando il progresso dell'hardware riuscì a soddisfare i requisiti computazionali dei modelli basati su reti neurali. Il costante sviluppo culminò nell'ultimo ventennio quando venne introdotta la GPU, che, insieme all'aumento dei dati disponibili, accelerò notevolmente i progressi nel campo dell'AI [59], [60].

3.1 PRINCIPI DI BASE ED EVOLUZIONE

Le reti neurali artificiali (ANN) — comunemente note come reti neurali (NN) — mirano a rappresentare un modello semplificato del cervello, trattato come una struttura composta da neuroni. Risulta quindi essenziale comprendere il funzionamento del singolo *neurone artificiale* per poi esplorare la struttura di una rete neurale, che è un collegamento tra più neuroni artificiali.

Come un neurone biologico, il neurone artificiale (Figura 3.1) riceve un numero indefinito n di segnali in *input*, che possono essere rappresentati con la notazione $x_0, x_1, \ldots, x_n$. Tali valori possono essere raggruppato nel vettore di input $\mathbf{x} = [x_0, x_1, \ldots, x_n]$. Per descrivere il risultato di tutti i segnali in ingresso del neurone, si introduce una funzione g, che si suppone essere una semplice somma algebrica dei segnali di input x_i, con $i \in [0, n]$. Ogni segnale di input, prima di essere sommato viene moltiplicato per il rispettivo peso (*weight*) w_i, con $i \in [0, n]$, che appartiene al vettore dei pesi $\mathbf{w} = [w_0, w_1, \ldots, w_n]$. Ne consegue che il segnale in uscita, indicato con v, comprenderà la somma del prodotto del segnale i-esimo (x_i) con il rispettivo peso i-esimo (w_i):

$$v = g(\mathbf{w}, \mathbf{x}) = \sum_{i=0}^{n} w_i x_i \tag{3.1}$$

Per attivare un neurone, è necessario che il segnale v prodotto in *output* sia superiore ad una soglia prescelta, solitamente rappresenta dalla prima componente del vettore di input (x_0). Questo valore, noto come *bias*, è scelto arbitrariamente ma spesso viene impostato ad 1. Il criterio che

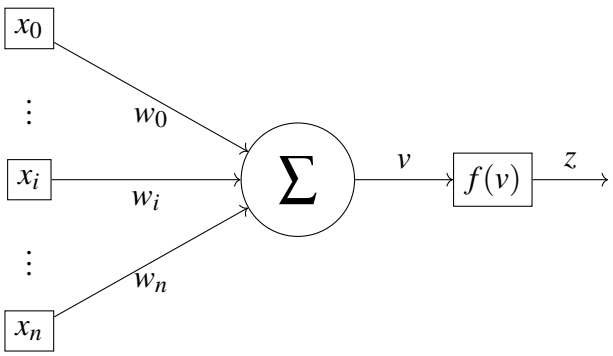

Figura 3.1: Rappresentazione schematica del funzionamento di un neurone artificiale. I segnali in input X_i vengono moltiplicati con i rispettivi pesi W_i e sommati tra loro; il risultato v viene processato dalla funzione di $f(v)$ che restituisce l'output z.

descrive l'attivazione del neurone artificiale è riassunto nella *funzione di attivazione*, chiamata $z = f(v)$. La funzione di attivazione più semplice che descrive tale criterio la funzione *gradino* $\mathbf{1}(v)$, descritta graficamente nella Figura 3.2:

$$z(v) = \mathbf{1}(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases}$$

Come per un cervello umano, anche la rete neurale deve essere in grado di impararare. In particolare un neurone artificiale deve essere in grado di reagire in un determinato modo quando in input riceve determinati *pattern*. Affinchè il neurone sia in grado di comporatarsi correttamente di fronte a dei pattern, è fondamentale allenarlo: è dunque necessario utilizzare un *training dataset* che contenga numerosi vettori di input $\mathbf{x}$ e, per ciascuno di essi sia presente la risposta corretta che il neurone dovrebbe fornire. Formalmente definiamo il dataset $\mathscr{D}$ come:

$$\mathscr{D} = \left\{ \{X_1, y_1\}, \{X_2, y_2\}, \ldots \left\{X_j, y_j\right\}, \ldots \{X_M, y_M\} \right\}$$

Il dataset[1] è composto da M vettori di input, definiti con la notazione X_j ($j \in [1, M]$), e da esattamente M risposte y_j, che rappresentano il comportamento atteso del neurone se il vettore di ingresso è X_j. È quindi possibile definire la matrice $\mathbf{X}$ che contiene esattamente M vettori — ciascuno in una riga — i quali sono composti esattamente da n componenti (o *feature*), che sono le componenti associate al vettore dei pesi $\mathbf{w}$, precedentemente introdotto. Alla matrice $\mathbf{X}$ è associato il vettore $\mathbf{y}$, anch'esso di dimensione M tale per cui la componente y_j sia la risposta

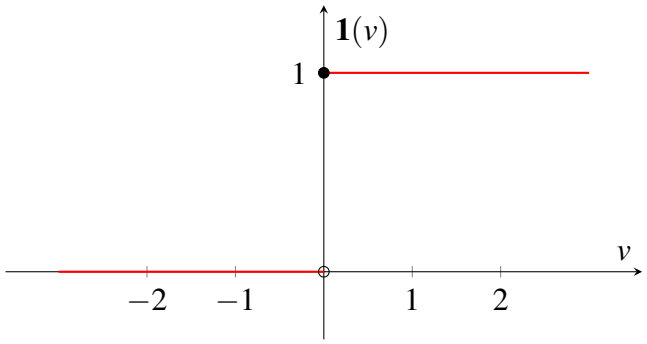

Figura 3.2: Grafico della funzione gradino $\mathbf{1}(v)$. Si osserva che vale zero per valori strettamente minori di zero e uno per valori maggiori o ugali a zero.

[1] Un dataset così definito è utilizzato nel *supervised learning*, dove all'interno del dataset sono presenti sia i vettori in ingresso che la risposta attesa. Si contrappone l'*unsupervised learning* — che non verrà trattato — dove sono presenti solo i vettori in ingresso e sono sconosciute le risposte attese.

attesa del vettore di input X_j.

$$\mathbf{X} = \begin{bmatrix} X_1^0 & X_1^1 & \cdots & X_1^i & \cdots & X_1^n \\ X_2^0 & X_2^1 & \cdots & X_2^i & \cdots & X_2^n \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_j^0 & X_j^1 & \cdots & X_j^i & \cdots & X_j^n \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_M^0 & X_M^1 & \cdots & X_M^i & \cdots & X_M^n \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_j \\ \vdots \\ y_M \end{bmatrix}$$

Da questo deriva che il vettore X_j è dimensionalmente compatibile con il vettore di pesi $\mathbf{w}$ in quanto hanno esattamente la stessa dimensione.

$$X_j = \left[X_j^0, X_j^1, \ldots, X_j^n\right] \qquad \mathbf{w} = [w_0, w_1, \ldots, w_n]$$

Perciò il dataset $\mathscr{D}$ può essere riscritto attraverso la matrice $\mathbf{X}$ ed il vettore di risposte attese associato $\mathbf{y}$.

$$\mathscr{D} = \{\mathbf{X}, \mathbf{y}\}$$

Con un dataset di partenza si può definire un algoritmo generico che sia in grado di descrivere il processo di apprendimento di neurone artificiale. Come descritto nell'Algoritmo 1, dopo aver inizializzato il vettore di pesi $\mathbf{w}$, per ogni vettore X_j del dataset $\mathscr{D}$ vengono calcolati il segnale v e il valore della funzione di attivazione $z = f(v)$. L'idea alla base dell'algoritmo è quella di modificare il vettore di pesi $\mathbf{w}$ in funzione del risultato della funzione di attivazione rispetto al segnale processato. In questo modo, una volta processati tutti i vettori presenti nel dataset, è

Algoritmo 1 Allenamento del neurone artificiale

Require: Dataset $\mathscr{D}$
  Inizializza $\mathbf{w}$ con numeri casuali
  $j \leftarrow 1$
  while $j \leq M$ **do**
    Calcola v in funzione di X_j e $\mathbf{w}$
    Determina z in rapporto a v e y_j
    Modifica $\mathbf{w}$ a seconda del risultato z
    $j \leftarrow j + 1$
  end while

possibile constatare se il neurone sia stato in grado di apprendere il comportamento desiderato in funzione del vettore in ingresso.

Il modello di neurone artificiale che viene tuttora utilizzato nelle reti neurali è il percettrone. Dato un vettore in ingresso X_j e un vettore di pesi $\mathbf{w}$, il segnale v è dato dalla somma algebrica

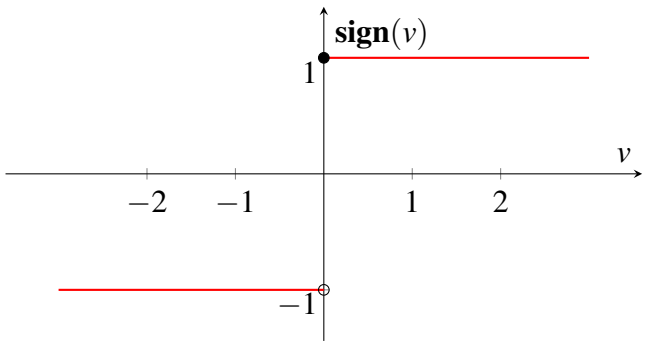

Figura 3.3: Grafico della funzione segno **sign**(v). Si osserva che vale -1 per valori strettamente minori di zero e 1 per valori maggiori o ugali a zero.

dei prodotti di ciascuna delle componenti:

$$v = g\left(\mathbf{w}, X_j\right) = \sum_{i=0}^{n} w_i X_j^i$$

La funzione di attivazione $f(v)$ è la funzione *"segno"* (Figura 3.3), chiamata anche *gradino bipolare* e definita come segue.

$$z(v) = \mathbf{sign}(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ -1 & \text{se } v < 0 \end{cases}$$

Il particolare più importante però è la *learning rule*, ovvero il criterio secondo il quel il vettore di pesi è aggiornato a seconda della predizione del percettrone. Se la predizione z_j rispetto al vettore X_j è diversa dalla risposta attesa y_j, allora il vettore di pesi è modificato come segue.

$$w_i = w_i + y_j X_j^i$$

Al modello di neurone artificiale del percettrone si aggiunge anche l'*Adaline* (*Adaptive Linear Neuron*) la cui learning rule

$$w_i = w_i + \eta \left(y_j - v_j\right) X_j^i$$

Oltre a differire per la learning rule, alcuni modelli differiscono per l'activation function. Oltre alla funzione "segno" anche la funzione *sigmoide* (Figura 3.4) è largamente utilizzata.

$$\sigma(v) = \frac{1}{1+e^{-v}}$$

Una rete neurale non è altro che una struttura composta da neuroni artificiali i quali sono organizzati in livelli (*layers*). In una rete neurale multilivello, è sempre presente un input layer e un output layer. I livelli che si trovano tra questi sono detti livelli nascosti — *hidden* layers. Va osservato che i neuroni dello stesso livello non sono mai collegati ma sono collegati con quelli

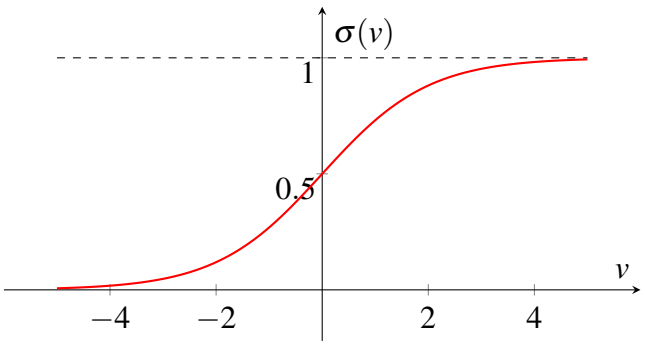

Figura 3.4: Grafico della funzione gradino $\sigma(v)$.

del livello precedente e del livello successivo. In particolare, data una rete neurale, i neuroni del livello r ricevono il segnale dai neuroni del livello $r-1$ e, dopo aver elaborato le informazioni, inviano il segnale processato i neuroni del livello $r+1$. Osservando la Figura 3.5 si introduce una nuova notazione per le reti neurali: si definisce con $N^{(r)(k)}$ il neurone k-esimo che si trova nel livello r; di conseguenza il suo output è identificato dalla notazione $y^{(r)(k)}$ e il suo input è $X_i^{(r)(k)}$, con $i \in [1,n]$, associato al peso $w_i^{(r)(k)}$. Si osserva che n è il numero di input del neurone, e di conseguenza è il numero di neuroni presenti nel livello precedente; particolarmente l'input i-esimo di un neurone generico k in un livello r è uguale all'output del neurone k al livello $r-1$:

$$y^{(r-1)(i)} = X_i^{(r)(k)}$$

Utilizzando la somma algebrica, illustrata nell'equazione 3.1, l'output di un neurone k, in un

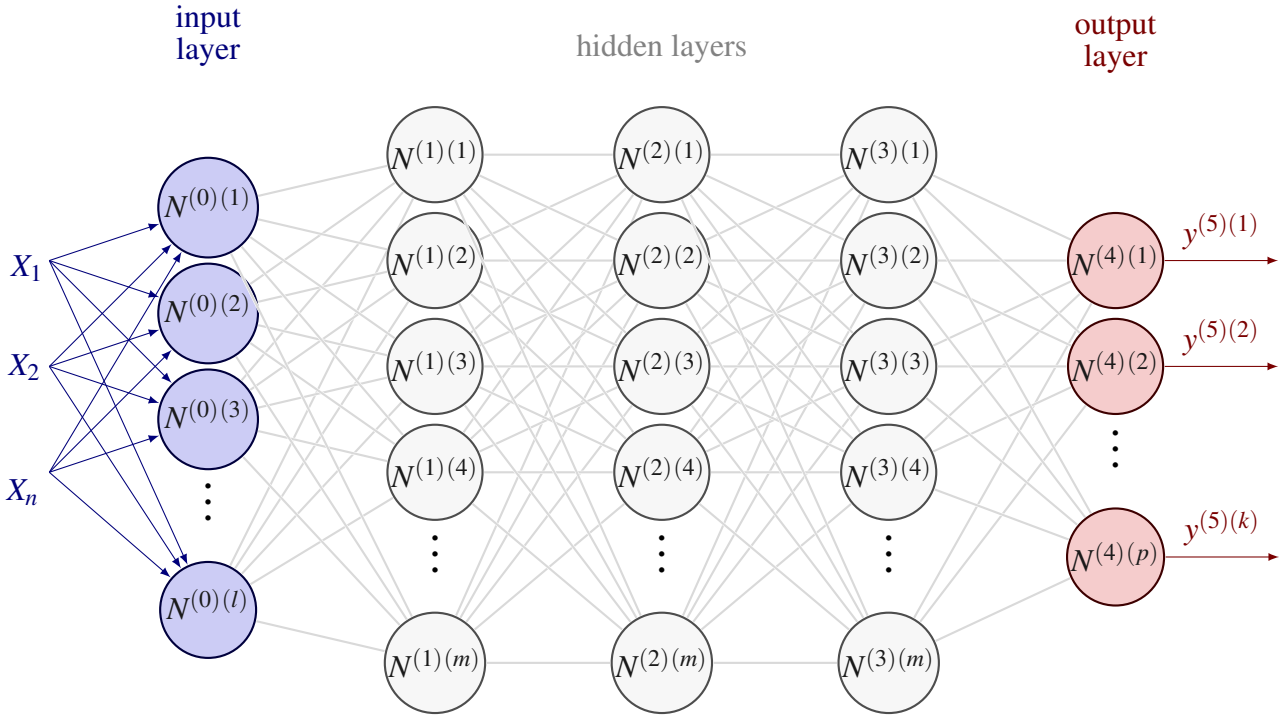

Figura 3.5: Rappresentazione di una rete neurale multilivello.

livello r è dato da:

$$y^{(r)(k)} = f\left(\sum_i w_i^{(r)(k)} X_i^{(r)(k)}\right) = f\left(\sum_i w_i^{(r)(k)} y^{(r-1)(i)}\right)$$

Tale risultato può essere osservato graficamente nella Figura.

[59]

Come nel caso di un singolo neurone artificiale, anche una rete neurale deve essere allenata. Per allenare una NN i pesi della rete devono essere impostati in modo tale da *minimizzare* l'errore dato un determinato dataset di partenza. I pesi vengono man mano rifiniti durante la fase di allenamento attraverso la derivata parziale della funzione "errore" calcolate sui pesi. Questo approccio è del tutto analogo all'approccio del *Gradient Descent* (GD)

si utilizza l'algoritmo della *backpropagation*, introdotto per la prima volta nel 1986. L'idea alla base di questo algoritmo è quella di minimizzare una funzione di errore attraverso l'approccio del *Gradient Descent* (GD). Essendo che questo approccio richiede il calcolo del gradiente, è necessario che la funzione di attivazione sia continua e differenziabile: a questo proposito si utilizza la funzione sigmoide (Figura 3.4) anziché la funzione "segno" adottata nel percettrone.

$$s(x) = \sigma(x) = \frac{1}{1+e^{-x}} \qquad \frac{d}{dx}s(x) = \frac{e^{-x}}{(1+e^{-x})^2} = s(x)\,(1-s(x))$$

Per una rete compota da percettroni, la funzione "segno" può essere approssimata dalla *sigmoide simmetrica*, che non è altro che la tangente iperbolica di $\frac{x}{2}$:

$$S(x) = 2s(x) - 1 = \tanh\left(\frac{x}{2}\right)$$

[61]

3.2 RETI NEURALI CONVOLUZIONALI

parla anche delle metodologie di evaluation

convolution: filters sono di fatto dei profile sulle sequenze

[59], [62]

4

Reti convoluzionali e varianti non codificanti

5

Discussione

Tabella che specifica e riassume per ogni tool encoding, dataset etc

riassume quanto analizzato prima, riporta i risultati del paper piu recente in modo da avere un momento in cui riassumo la situa

Sperimentalmente, per risorse a disposizione, per il confronto ci si basa sui risultati dell'ultimo paper

6

Conclusioni

Limiti, punti di forza e future ricerche. Se ci sono in ballo futuri lavori, nuove direzioni da seguire, da osservare dagli articoli dei 3 tool

Bibliografia

[1] S. S. Sahu e G. Panda, «Identification of protein-coding regions in DNA sequences using a time-frequency filtering approach,» *Genomics, Proteomics and Bioinformatics*, vol. 9, n. 1-2, pp. 45–55, 2011.

[2] T. D. Pollard, W. C. Earnshaw, J. Lippincott-Schwartz e G. Johnson, *Cell Biology E-Book: Cell Biology E-Book*. Elsevier Health Sciences, 2022.

[3] F. Zhang e J. R. Lupski, «Non-coding genetic variants in human disease,» *Human molecular genetics*, vol. 24, n. R1, R102–R110, 2015.

[4] J. French e S. Edwards, «The role of noncoding variants in heritable disease,» *Trends in Genetics*, vol. 36, n. 11, pp. 880–891, 2020.

[5] H. Chial, «Mendelian genetics: patterns of inheritance and single-gene disorders,» *Nature Education*, vol. 1, n. 1, p. 63, 2008.

[6] S. Pagni, J. D. Mills, A. Frankish, J. M. Mudge e S. M. Sisodiya, «Non-coding regulatory elements: Potential roles in disease and the case of epilepsy,» *Neuropathology and Applied Neurobiology*, vol. 48, n. 3, e12775, 2022.

[7] A. Kapoor et al., «An enhancer polymorphism at the cardiomyocyte intercalated disc protein NOS1AP locus is a major regulator of the QT interval,» *The American Journal of Human Genetics*, vol. 94, n. 6, pp. 854–869, 2014.

[8] E. Khurana, Y. Fu, D. Chakravarty, F. Demichelis, M. A. Rubin e M. Gerstein, «Role of non-coding sequence variants in cancer,» *Nature Reviews Genetics*, vol. 17, n. 2, pp. 93–108, 2016.

[9] J. Tian et al., «Systematic functional interrogation of genes in GWAS loci identified ATF1 as a key driver in colorectal cancer modulated by a promoter-enhancer interaction,» *The American Journal of Human Genetics*, vol. 105, n. 1, pp. 29–47, 2019.

[10] S. E. Bojesen et al., «Multiple independent variants at the TERT locus are associated with telomere length and risks of breast and ovarian cancer,» *Nature genetics*, vol. 45, n. 4, pp. 371–384, 2013.

[11] K. Michailidou et al., «Association analysis identifies 65 new breast cancer risk loci,» *Nature*, vol. 551, n. 7678, pp. 92–94, 2017.

[12] C. S. Pareek, R. Smoczynski e A. Tretyn, «Sequencing technologies and genome sequencing,» *Journal of applied genetics*, vol. 52, pp. 413–435, 2011.

[13] S. K. Burley, H. M. Berman, G. J. Kleywegt, J. L. Markley, H. Nakamura e S. Velankar, «Protein Data Bank (PDB): the single global macromolecular structure archive,» *Protein crystallography: methods and protocols*, pp. 627–641, 2017.

[14] N. M. Luscombe, D. Greenbaum e M. Gerstein, «What is bioinformatics? A proposed definition and overview of the field,» *Methods of information in medicine*, vol. 40, n. 04, pp. 346–358, 2001.

[15] C. Caudai et al., «AI applications in functional genomics,» *Computational and Structural Biotechnology Journal*, vol. 19, pp. 5762–5790, 2021.

[16] J. Zhou e O. G. Troyanskaya, «Predicting effects of noncoding variants with deep learning–based sequence model,» *Nature methods*, vol. 12, n. 10, pp. 931–934, 2015.

[17] D. R. Kelley, J. Snoek e J. L. Rinn, «Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks,» *Genome research*, vol. 26, n. 7, pp. 990–999, 2016.

[18] W. Ma et al., «DeepSATA: A Deep Learning-Based Sequence Analyzer Incorporating the Transcription Factor Binding Affinity to Dissect the Effects of Non-Coding Genetic Variants,» *International Journal of Molecular Sciences*, vol. 24, n. 15, p. 12 023, 2023.

[19] B. Alberts et al., *Essential cell biology*. Garland Science, 2015.

[20] P. F. Chinnery e E. A. Schon, «Mitochondria,» *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 74, n. 9, pp. 1188–1199, 2003.

[21] H. M. McBride, M. Neuspiel e S. Wasiak, «Mitochondria: more than just a powerhouse,» *Current biology*, vol. 16, n. 14, R551–R560, 2006.

[22] G. K. Voeltz, M. M. Rolls e T. A. Rapoport, «Structural organization of the endoplasmic reticulum,» *EMBO reports*, vol. 3, n. 10, pp. 944–950, 2002.

[23] A. Ballabio, «The awesome lysosome,» *EMBO molecular medicine*, vol. 8, n. 2, pp. 73–76, 2016.

[24] C. Yang e X. Wang, «Lysosome biogenesis: Regulation and functions,» *The Journal of cell biology*, vol. 220, n. 6, 2021.

[25] E. C. Dell'Angelica, C. Mullins, S. Caplan e J. S. Bonifacino, «Lysosome-related organelles,» *The FASEB Journal*, vol. 14, n. 10, pp. 1265–1278, 2000.

[26] M. Islinger, S. Grille, H. D. Fahimi e M. Schrader, «The peroxisome: an update on mysteries,» *Histochemistry and cell biology*, vol. 137, pp. 547–574, 2012.

[27] National Human Genome Research Institute, *Deoxyribonucleic acid (DNA) Image*, `https://www.genome.gov/genetics-glossary/Deoxyribonucleic-Acid`, 2024.

[28] C. Fonseca Guerra, F. M. Bickelhaupt, J. G. Snijders e E. J. Baerends, «Hydrogen bonding in DNA base pairs: reconciliation of theory and experiment,» *Journal of the American Chemical Society*, vol. 122, n. 17, pp. 4117–4128, 2000.

[29] A. Jansen e K. J. Verstrepen, «Nucleosome positioning in Saccharomyces cerevisiae,» *Microbiology and molecular biology reviews*, vol. 75, n. 2, pp. 301–320, 2011.

[30] G. Zheng, *The packaging of DNA in chromatin*. Rutgers The State University of New Jersey, School of Graduate Studies, 2010.

[31] National Human Genome Research Institute, *Chromosome Image*, `https://www.genome.gov/genetics-glossary/Chromosome`, 2024.

[32] M. B. Gerstein et al., «What is a gene, post-ENCODE? History and updated definition,» *Genome research*, vol. 17, n. 6, pp. 669–681, 2007.

[33] R. J. White, *Gene transcription: mechanisms and control*. John Wiley & Sons, 2009.

[34] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts e P. Walter, «From DNA to RNA,» in *Molecular Biology of the Cell. 4th edition*, Garland Science, 2002.

[35] P. Cramer, «Organization and regulation of gene transcription,» *Nature*, vol. 573, n. 7772, pp. 45–54, 2019.

[36] National Human Genome Research Institute, *Transcription Image*, `https://www.genome.gov/genetics-glossary/Transcription`, 2024.

[37] A. Philips e T. Cooper*, «RNA processing and human disease,» *Cellular and Molecular Life Sciences CMLS*, vol. 57, pp. 235–249, 2000.

[38] S. Hocine, R. H. Singer e D. Grünwald, «RNA processing and export,» *Cold Spring Harbor perspectives in biology*, vol. 2, n. 12, a000752, 2010.

[39] M. Livingstone, E. Atas, A. Meller e N. Sonenberg, «Mechanisms governing the control of mRNA translation,» *Physical biology*, vol. 7, n. 2, p. 021 001, 2010.

[40] V. Ramakrishnan, «Ribosome structure and the mechanism of translation,» *Cell*, vol. 108, n. 4, pp. 557–572, 2002.

[41] J. Lemonnier, N. Lemonnier, S. Pascolo e C. Pichon, «The Marathon of the Messenger,»

[42] National Human Genome Research Institute, *Translation Image*, `https://www.genome.gov/genetics-glossary/Translation`, 2024.

[43] G. E. Schulz e R. H. Schirmer, *Principles of protein structure*. Springer Science & Business Media, 2013.

[44] R. M. Bavle, «Mitosis at a glance,» *Journal of Oral and Maxillofacial Pathology*, vol. 18, n. Suppl 1, S2–S5, 2014.

[45] C. E. Walczak, S. Cai e A. Khodjakov, «Mechanisms of chromosome behaviour during mitosis,» *Nature reviews Molecular cell biology*, vol. 11, n. 2, pp. 91–102, 2010.

[46] X. Li, F. Yang e B. Rubinsky, «A theoretical study on the biophysical mechanisms by which tumor treating fields affect tumor cells during mitosis,» *IEEE Transactions on Biomedical Engineering*, vol. 67, n. 9, pp. 2594–2602, 2020.

[47] M. Sullivan e D. O. Morgan, «Finishing mitosis, one step at a time,» *Nature reviews Molecular cell biology*, vol. 8, n. 11, pp. 894–903, 2007.

[48] National Human Genome Research Institute, *Mitosis Image*, https://www.genome.gov/genetics-glossary/Mitosis, 2024.

[49] R. A. Laskey, M. P. Fairman e J. J. Blow, «S phase of the cell cycle,» *Science*, vol. 246, n. 4930, pp. 609–614, 1989.

[50] S. P. Bell e A. Dutta, «DNA replication in eukaryotic cells,» *Annual review of biochemistry*, vol. 71, n. 1, pp. 333–374, 2002.

[51] A. Dutta e S. P. Bell, «Initiation of DNA replication in eukaryotic cells,» *Annual review of cell and developmental biology*, vol. 13, n. 1, pp. 293–332, 1997.

[52] «Chapter 42 - S Phase and DNA Replication,» in *Cell Biology (Third Edition)*, T. D. Pollard, W. C. Earnshaw, J. Lippincott-Schwartz e G. T. Johnson, cur., Third Edition, Elsevier, 2017, pp. 727–741, ISBN: 978-0-323-34126-4. DOI: https://doi.org/10.1016/B978-0-323-34126-4.00042-6. indirizzo: https://www.sciencedirect.com/science/article/pii/B9780323341264000426.

[53] National Human Genome Research Institute, *DNA Replication Image*, https://www.genome.gov/genetics-glossary/DNA-Replication, 2024.

[54] W. S. McCulloch e W. Pitts, «A logical calculus of the ideas immanent in nervous activity,» *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[55] A. M. Turing, *Computing machinery and intelligence*. Springer, 2009.

[56] F. Rosenblatt, «The perceptron: a probabilistic model for information storage and organization in the brain.,» *Psychological review*, vol. 65, n. 6, p. 386, 1958.

[57] M. Minsky e S. A. Papert, *Perceptrons, reissue of the 1988 expanded edition with a new foreword by Léon Bottou: an introduction to computational geometry*. MIT press, 2017.

[58] D. E. Rumelhart, G. E. Hinton e R. J. Williams, «Learning representations by back-propagating errors,» *nature*, vol. 323, n. 6088, pp. 533–536, 1986.

[59] M. Flasiński, *Introduction to artificial intelligence*. Springer, 2016.

[60] N. Muthukrishnan, F. Maleki, K. Ovens, C. Reinhold, B. Forghani, R. Forghani et al., «Brief history of artificial intelligence,» *Neuroimaging Clinics of North America*, vol. 30, n. 4, pp. 393–399, 2020.

[61] S. Sathyanarayana, «A gentle introduction to backpropagation,» *Numeric Insight*, vol. 7, pp. 1–15, 2014.

[62] R. Rojas e R. Rojas, «The backpropagation algorithm,» *Neural networks: a systematic introduction*, pp. 149–182, 1996.