Politecnico di Torino

Politecnico di Torino

Ingegneria Informatica - Computer Engineering

A.a. 2024/2025

Sessione di laurea Luglio 2025

Analisi dei Disturbi della Voce attraverso Explainable-AI

Un approccio concept-based

Relatori:

Tania Cerquitelli

Gabriele Ciravegna

Candidati:

Davide Ghia

Ringraziamenti

Ringrazio tutte le persone che mi sono state vicine in questi anni e che mi hanno permesso di raggiungere questo traguardo. Innanzitutto, ringrazio i miei genitori. Fin dal primo giorno mi avete motivato e sostenuto in questo percorso, senza farmi mai mancare nulla. In particolare, grazie papà per avermi trasmesso la curiosità verso ciò che non conosco e la volontà di imparare costantemente. Grazie mamma per avermi spronato a dare sempre il cento percento, a volte qualcosina in più, anche quando non credevo di farcela. Grazie a tutta la mia famiglia, alle mie sorelle e a mio fratello. Chi da lontano e chi da vicino, mi avete consigliato e preparato al mondo universitario.

Grazie a Freddo, con cui ho condiviso ogni giorno di lezione, di studio, di progetti e soprattutto di riposo. Ancora non ho capito come abbiamo fatto a laurearci. Grazie a tutti i miei compagni di università, per avermi aiutato quando non ci capivo niente e, soprattutto, per aver reso più umano il Politecnico. Grazie ai miei amici, a quelli che conosco da sempre e a quelli che ho conosciuto crescendo, a voi che basta un bar e una birra per farmi dimenticare tutto il resto.

Infine, grazie Valentina. Hai condiviso tutte le mie gioie e le mie delusioni come se fossero tue. Hai sopportato i miei momenti di stress e di nervosismo, riuscendo sempre a calmarmi. Quando l'università si è presa tempo che avremmo potuto e dovuto passare insieme, non me l'hai fatto pesare, ma al contrario, mi hai aiutato nel lavoro. Grazie perché nessuno avrebbe fatto tutto questo per me. Purtroppo ti è capitata la sfortuna più grande: un ragazzo ingegnere.

Indice

Elenco delle figure

Capitolo 1

Introduzione

Obbiettivi della tesi

Nel contesto medico, l'utilizzo dell'intelligenza artificiale - e in particolare del machine learning o apprendimento automatico - ha rivoluzionato molte discipline, aprendo nuove prospettive di analisi e diagnosi. Tuttavia, data l'importanza e la sensibilità dei dati trattati in ambito sanitario, l'opacità che circonda le reti neurali può portare a insicurezza e diffidenza verso l'adozione di queste tecnologie.

Il ramo dell'intelligenza artificiale chiamato X-AI (eXplainable-AI) si occupa di questo problema, cercando di rendere più trasparenti i processi decisionali attraverso spiegazioni, grafici e modellamento della struttura stessa delle reti. In questa tesi, si è cercato di applicare i principi dell'eXplainable-AI nel campo dell'analisi dei disturbi della voce, rimasto relativamente poco studiato rispetto ad altri ambiti clinici. L'obbiettivo principale è quello di trovare una soluzione che garantisca interpretabilità e maggiore chiarezza all'utente umano senza compromettere in modo eccessivo le prestazioni del modello.

L'approccio utilizzato è stato di tipo concept-based. La rete neurale viene progettata per prevedere una serie di concetti - rappresentazioni semantiche comprensibili e familiari per un essere umano - sulla base dei quali avviene la classificazione in voce eufonica o patologica. In questo modo, le decisioni vengono prese riconoscendo una correlazione tra i concetti associati a una determinata voce (*e.g.* disfonia, breathiness) e la relativa classe, garantendo maggiore trasparenza al processo di classificazione. Questa soluzione si rivelerebbe utile non soltanto al paziente, il quale beneficia di una diagnosi più approfondita e articolata, ma anche per il personale medico che può supervisionare meglio il funzionamento del sistema. Nel caso di una diagnosi errata, sarebbe estremamente più facile comprendere perché il modello abbia predetto la classe sbagliata o modificare manualmente la predizione di un concetto per correggere l'errore. Considerando la famiglia delle reti concept-based,

ci concentreremo in particolare su due modelli, ovvero i CBM (Concept Bottleneck Model) e i CEM (Concept Embedding Model).

In sintesi, questa tesi propone l'applicazione di modelli concept-based nella classificazione dei disturbi della voce, con l'obbiettivo di raggiungere maggiore trasparenza e interpretabilità nella predizione mantenendo performance competitive. Il lavoro svolto ha portato alla submission di un workshop paper, intitolato "A Concept-based approach to Voice Disorder Detection", alla conferenza ECMLPKDD 2025, in attesa di essere revisionato.

Struttura della tesi

La tesi è strutturata in cinque diversi capitoli, ciascuno dedicato a un preciso argomento o a una fase specifica del lavoro svolto. In particolare, gli argomenti sono così suddivisi:

- **Capitolo 1** descrive la struttura e gli obbiettivi della tesi.
- **Capitolo 2** introduce gli ambiti in cui spazia la tesi, dando una panoramica generale del contesto attuale e dei lavori precedentemente svolti in questo settore.
- **Capitolo 3** descrive nel dettaglio l'approccio teorico e pratico utilizzato per svolgere gli esperimenti.
- **Capitolo 4** descrive gli esperimenti eseguiti e i risultati ottenuti.
- **Capitolo 5** riassume il lavoro svolto, ne evidenzia le limitazioni e possibili estensioni in lavori futuri.

Capitolo 2

Contesto

2.1 Introduzione all'intelligenza artificiale

Negli ultimi anni, l'intelligenza artificiale (IA) ha conosciuto uno sviluppo vertiginoso, affermandosi in ogni ambito della nostra vita quotidiana. La capacità di analizzare enormi moli di dati le ha permesso di eguagliare, e in alcuni casi di superare, le capacità computazionali del cervello umano in task specifiche.

In questo compito, eccelle in modo particolare una famiglia di modelli che prende il nome di reti neurali. Come suggerisce il nome, questi modelli si ispirano alla struttura del cervello umano: sono composti da diversi strati di nodi (i neuroni), collegati tra loro da un grande numero di legami (assoni). A ciascun collegamento è associato un *peso*, che rappresenta l'influenza esercitata da un neurone su quello a lui connesso. Il valore di ogni nodo, determinato dai valori dei suoi predecessori, viene trasformato attraverso una funzione di attivazione, ovvero un'approssimazione dell'attivazione dei neuroni biologici.

Le reti neurali, nonostante siano state introdotte già negli anni Cinquanta del secolo scorso, sono rimaste per lungo tempo ai margini della ricerca sull'intelligenza artificiale. A causa delle limitazioni teoriche dei modelli iniziali, come il percettrone [1], e della mancanza di risorse computazionali adeguate, queste tecniche vennero progressivamente abbandonate a favore di approcci simbolici o statistici. Il percettrone, in particolare, era composto da un singolo strato con pesi aggiornabili, il che lo rendeva adatto unicamente a problemi in cui i dati erano linearmente separabili[1]. La difficoltà principale che ha impedito la realizzazione di perceptron composti da più layer addestrabili è stata la mancanza di un algoritmo di addestramento efficace. Nel 1986, dopo essere stato formalizzato da un punto di vista matematico

[1]Essendo composto da un singolo layer lineare, eventuali criteri di separazione più complessi (quadratici, cubici, ecc.) non potevano essere appresi.

circa un decennio prima, l'algoritmo noto come *backpropagation* fu applicato con successo all'addestramento delle reti neurali [2]. Questa tecnica, che prevede il calcolo del gradiente procedendo a ritroso attraverso gli strati della rete - da cui il nome -, ha reso possibile l'addestramento dei Multilayer Perceptrons (MLP) e rilanciato l'interesse verso il deep learning.

Solo a partire dai primi anni 2000, con l'avvento di hardware più potenti – in particolare le GPU – e la disponibilità di grandi quantità di dati, le reti neurali hanno conosciuto una nuova diffusione. Grazie anche all'ottimizzazione di tecniche di addestramento come la backpropagation, le deep neural networks – reti neurali composte da molteplici strati – sono oggi diventate il fulcro della ricerca nel campo dell'intelligenza artificiale.

Un grande passo in avanti nella diffusione delle reti neurali è stata l'introduzione delle Convolutional Neural Networks (CNN) [3]. Le CNN sono modelli gerarchici specializzati nell'apprendimento di pattern locali e invarianti a traslazioni, particolarmente adatti all'elaborazione di input con struttura spaziale o temporale, come immagini, segnali audio o video. Queste reti sono composte da tre tipologie principali di layer:

- **layer convoluzionali**: applicano una convoluzione tra l'input del layer e una serie di filtri, detti *kernel*, generando le cosiddette *feature map*, mappe che identificano specifici pattern locali;
- **funzioni di attivazione**: funzioni non lineari (*e.g.* ReLU, Leaky ReLU) applicate elemento per elemento all'output della convoluzione;
- **layer di pooling**: applica funzioni di pooling all'input riducendone la dimensionalità;

In una CNN, generalmente, si susseguono blocchi formati da queste tre tipologie di layer in sequenza. La rete è completata da uno o più layer fully-connected, ovvero dove ogni neurone è collegato - attraverso un peso - a ciascuno degli altri neuroni, i quali permettono la combinazione dell'informazione estratta nei livelli precedenti e la mappatura verso l'output finale. Le reti convoluzionali sono definite gerarchiche perché, proseguendo attraverso i vari layer della rete, la complessità delle caratteristiche apprese aumenta progressivamente. I primi layer della rete tendono a riconoscere pattern semplici e localizzati. Grazie alla convoluzione, ogni neurone aggrega l'informazione di più neuroni appartenenti al layer precedente. Procedendo verso layer più profondi, viene aggregata l'informazione appartenente a regioni sempre più ampie. In altri termini, aumenta quello che viene definito *receptive field*. In questo modo, gli strati successivi riescono a combinare pattern semplici in strutture sempre più complesse, acquisendo una rappresentazione globale dell'input. I kernel, le cui dimensioni variano tra 1x1 e 7x7, vengono appresi tramite backpropagation.

Le Convolutional Neural Network sono state tra i primi modelli a introdurre un'estrazione automatica delle feature, rivoluzionando il machine learning, precedentemente basato sull'estrazione manuale delle feature. Un momento chiave nella diffusione delle CNN è stato il successo di AlexNet [4], presentato nel 2012, che dimostrò l'efficacia di queste reti nella computer vision.

Recentemente, a catturare l'attenzione del dibattito pubblico sono stati i Large Language Models (LLMs), reti neurali di grandi dimensioni in grado di interpretare e generare testo in linguaggio naturale. Nel 2022, l'avvento di ChatGPT - LLM basato su diverse evoluzioni del modello GPT (Generative Pre-Trained Transformer) [5, 6, 7, 8] - e il suo grande successo hanno suscitato un forte slancio nella direzione di questi modelli, il quale si protrae ancora oggi con lo sviluppo di reti sempre più avanzate ed efficienti (*e.g.* Gemini [9], DeepSeek [10]). La quasi totalità dei chat bot rilasciata negli ultimi anni è basata su una famiglia di reti che prende il nome di *transformer* [11]. Questa tipologia di reti, nate per la traduzione del testo e poi diffusesi in moltissimi ambiti, sfrutta un meccanismo noto come *attention mechanism*. L'attenzione consiste nel calcolare quanto la rappresentazione di un componente della sequenza sia influenzata da ciascuno degli altri componenti. I componenti sono definiti come *token*. Nel Natural Language Processing (NLP) - task principale dei chat bot - i tokenizer più semplici fanno corrispondere i token alle parole che costituiscono la frase. Se prendiamo come esempio la frase "Il cane abbaia", il token "abbaia" sarà fortemente influenzato dal token "cane" a causa della loro relazione semantica, mentre, per lo stesso motivo, sarà meno influenzato dal token corrispondente a "Il". In tokenizer più recenti e complessi, le parole vengono a loro volte scomposte in sottoparti, ciascuna delle quali è associata a un token. Sfruttando il meccanismo dell'attenzione, i transformer riescono ad avere una visione più ampia e generale della sequenza, riuscendo a catturare anche le relazioni a lungo raggio. Questa capacità ha permesso la diffusione dei transformer anche nella *computer vision* - object detection, image classifiation, semantic segmentation - e, più recentemente, nell'analisi dei segnali audio. Il riconoscimento di pattern visivi o dell'audio si è rivelato di fondamentale importanza e ha influito nell'affermazione dei transformer come architettura general-purpose.

Le relazioni a lungo raggio, prima dell'introduzione delle reti transformer, hanno rappresentato una delle principali sfide per il deep learning - branca del machine learning che si basa sull'utilizzo delle Deep Neural Network (DNN). Nell'ambito dell'analisi predittiva, le Recurrent Neural Networks (RNN) [12] sono state tra i primi modelli impiegati, mentre la loro evoluzione, le Long Short-Term Memory (LSTM) [13], hanno rappresentato per anni lo standard nell'elaborazione delle sequenze. Entrambe le tipologie di modelli possono essere interpretate come macchine a stati, in cui lo stato interno rappresenta una forma di memoria, utile a catturare le dipendenze temporali all'interno della sequenza. In particolare, l'introduzione delle LSTM ha risolto il problema del *vanishing gradient*: le RNN

cercavano di memorizzare tutta l'informazione in uno stato unico, che, tuttavia, per sequenze di grandi dimensioni, si è rivelato insufficiente. Di conseguenza, l'applicazione di numerose funzioni non lineari sullo stesso stato tendeva a portare i pesi verso valori prossimi allo zero (gradiente che svanisce). Non avendo più informazioni sugli stati iniziali, l'output veniva influenzato principalmente dagli ultimi input ricevuti, perdendo eventuali relazioni temporalmente distanti. Con l'introduzione di più stati specializzati, le LSTM hanno permesso un flusso regolare del gradiente, limitando fortemente il problema del vanishing gradient. Nonostante i miglioramenti introdotti, l'analisi sequenziale di input sequenziali (*e.g.* frasi, video, audio) non ha risolto completamente il problema del bias verso gli input più recenti, a cui viene assegnato un peso maggiore rispetto agli input iniziali. Per questo motivo, l'analisi in parallelo delle sequenze di input, applicata dalle reti transformer tramite l'attenzione, ha permesso di ottenere grandi miglioramenti, rendendole lo standard di fatto per l'analisi di input sequenziali.

2.2 L'Intelligenza artificiale nella medicina

La capacità di analizzare dati particolarmente complessi ha permesso all'intelligenza artificiale di diventare uno strumento utilizzato anche in campo medico. L'analisi di immagini è probabilmente una delle aree più approfondite del settore, e discipline come radiologia, patologia o gastroenterologia hanno potuto beneficiare dei progressi ottenuti dalla ricerca. Attraverso l'analisi della tomografia computerizzata (TAC), per esempio, algoritmi di intelligenza artificiale sono in grado di diagnosticare la presenza di un cancro [14] o addirittura di predirne il rischio nei successivi tre anni [15]. Per la diagnosi della polmonite, invece, il metodo migliore al momento è l'analisi della radiografia toracica del paziente. In [16], si è dimostrato come una grande rete convoluzionale addestrata su un dataset pubblico riuscisse a diagnosticare correttamente la polmonite, superando in media le prestazioni di radiologi professionisti.

Al contrario, l'ambito dell'analisi dei disturbi della voce [17] - complice il minore sviluppo dell'IA applicata ai segnali audio - offre ancora molte opportunità di approfondimento e innovazione. Le diverse patologie che rientrano in questa categoria affliggono una notevole porzione della popolazione, influenzandone la qualità della vita [18, 19]. La diagnosi dei disturbi della voce può richiedere esami invasivi, come la laringoscopia o la biopsia della laringe. Queste procedure, spesso effettuate in anestesia generale, risultano talvolta spiacevoli e impegnative per il paziente. Siccome i modelli complessi basati su reti neurali profonde sono in grado di cogliere caratteristiche vocali e pattern sonori particolarmente articolati, il loro impiego nel riconoscimento automatico delle patologie vocali potrebbe, in alcuni casi, ridurre la necessità di ricorrere a esami invasivi. Un ulteriore aspetto da

considerare è l'affollamento delle strutture ospedaliere: nelle visite tradizionali, i pazienti si devono recare fisicamente in ospedale per essere visitati da un medico specialista. Se fosse possibile ottenere una diagnosi a distanza, ad esempio tramite un'applicazione installata sul telefono cellulare, si otterrebbe un duplice vantaggio: un risparmio di tempo per il paziente e una riduzione del carico di lavoro per il sistema sanitario.

Nel contesto accademico, alcuni studi rilevanti si sono occupati dell'analisi vocale per task affini alla voice disorder detection. Per esempio, interessanti risultati nella disciplina che prende il nome di *computational paralinguistic analysis*, sono stati ottenuti da [20]. In questo studio, si è dimostrato come sia possibile individuare i primi sintomi della malattia di Alzheimer utilizzando esclusivamente il linguaggio parlato. La strategia utilizzata prevede di estrarre lo spettrogramma di Mel - uno spettro di frequenza del suono - dal segnale audio e di analizzarlo con modelli Long-Short Term Memory (LSTM). L'efficacia dello spettrogramma di Mel e dei coefficienti che ne rappresentano le caratteristiche, ovvero i Mel-Frequency Cepstral Coefficients (MFCC), è stato dimostrato anche in [21]. In questo lavoro, gli spettri di frequenza sono forniti come input a un Vision Transformer (ViT), un modello specializzato nell'analisi delle immagini basato su una rete transformer.

I transformer, grazie alla loro grande duttilità, possono essere impiegati direttamente sul segnale audio grezzo. È il caso di HuBERT (Hidden-Unit BERT) [22] e di wav2vec 2.0 [23], due modelli basati su transformer in grado di estrarre rappresentazioni audio efficaci e utili per diversi ambiti in modo *self-supervised*. Questi due modelli, nonostante abbiano una struttura molto simile, si differenziano per la strategia utilizzata per l'addestramento. HuBERT, una volta ricevuto il segnale in input, ne maschera alcune parti. Il transformer, con una tecnica simile a quella usata dal modello BERT (Bidirectional Encoder Representations from Transformers) [24], tenta di predire la rappresentazione delle parti di segnale mascherate, basandosi sulle rappresentazioni non mascherate. Il modello wav2vec 2.0, invece, utilizza un approccio noto come *contrastive learning*. Anche in questo caso, alcune parti del segnale vengono mascherate. Successivamente, vengono generate false rappresentazioni per ogni parte mascherata. Il transformer, ricevute sia le rappresentazioni false che quelle reali, dovrà riconoscere la rappresentazione corretta per ogni parte di segnale mascherata. Entrambi i modelli permettono di iterare il procedimento in modo da ottenere embedding audio sempre più raffinati.

Per quanto riguarda la voice disorder analysis, sono stati condotti diversi studi basati su MFCCs e spettrogrammi di Mel. L'impiego di Multi Layer Perceptron (MLP), come per esempio le Random Forest (ensemble di Decision Trees), è stato esplorato in [25, 26]. In [25] viene applicata una trasformazione nota come *wavelet*, funzione utile per l'analisi di segnali non stazionari. Il segnale trasformato viene fornito in input ad un MLP con una profondità di pochi layer, il quale classifica la voce. Una combinazione di Gaussian Mixture Model (GMM) e Random Forest,

invece, è utilizzata in [26]. La pipeline è suddivisa in due fasi: la prima fase, di distinzione tra voci sane e patologiche (voice disorder detection) attraverso il GMM; la seconda fase, di classificazione del disturbo grazie alla Random Forest. Un'altra tipologia di reti applicata agli spettogrammi di Mel sono le Convolutional Neural Networks (CNN). In [27], lo spettrogramma, rappresentazione bidimensionale del suono, viene estratto dal segnale audio e fornito come input a una rete convoluzionale pre-addestrata nota come OpenL3. La CNN ha il compito di estrarre embedding di alto livello dalla rappresentazione 2D. Dopo una riduzione della dimensionalità, una Support Vector Machine (SVM) classifica la tipologia di disturbo della voce. Un altro interessante metodo di applicazione delle CNN alle rappresentazioni audio è presentato in [28]. In questo lavoro, dopo l'estrazione dei coefficienti MFCC, viene impiegata una *shallow CNN* - ovvero una CNN composta da un unico layer convoluzionale e da tre strati fully-connected per la classificazione. Questo metodo si è rivelato particolarmente veloce nella classificazione, pur raggiungendo performance superiori allo stato dell'arte, rendendolo un approccio particolarmente adatto all'ambito medico.

Anche nella voice disorder detection le architetture transformer hanno trovato, recentemente, grande impiego. La possbilità di applicarle direttamente sul segnale grezzo, senza la necessità di ricorrere all'estrazione di feature tramite pre-processing, è un notevole passo avanti nell'analisi della voce. In [29], vengono estratte feature dalla *raw waveform* per essere successivamente fornite ad un transformer leggero, un Class-Token (CT) transformer, che esegue la classificazione. I transformer normalmente richiederebbero grani moli di dati per essere addestrati *from scratch*. Questo rete, tuttavia, è una versione leggera di transformer, ovvero costituito da pochi layer. Inoltre, i dati in input sono embedding provenienti da un feature extractor pre-addestrato. Il CT transformer non deve, perciò, applicare l'attenzione sull'intero segnale, ma esclusivamente sulle feature estratte. E' stato preferito questo approccio per poter sfruttare l'attenzione, cogliendo le relazioni globali senza richiedere un numero eccessivo di dati di training. In [30] viene affrontata la task di voice disorder detection e di voice disorder classification mediante l'impiego di architetture transformer. Gli autori dimostrano come i transformer, e in particolare HuBERT, ottengano prestazioni superiori rispetto a modelli basati su CNN applicate a rappresentazioni 1D o 2D del segnale audio. Il lavoro introduce anche una pipeline che prevede l'addestramento indipendente di due reti transformer: una specializzata su registrazioni di frasi e l'altra su vocali sostenute. Entrambi i modelli producono una predizione, e tramite un framework noto come Mixture of Experts (MoE), viene selezionata quella associata alla maggiore confidenza, valutata tramite l'entropia delle probabilità predette. Questo approccio, combinato a tecniche di data augmentation, generazione sintetica di dati e alla scelta di dataset ad-hoc per il pre-training di HuBERT, ha permesso di ottenere risultati notevolmente superiori allo stato dell'arte. Il lavoro risulta particolarmente rilevante

per questa tesi, poiché uno dei dataset utilizzati negli esperimenti è l'IPV dataset, lo stesso impiegato in questo studio e che verrà analizzato nel dettaglio nel capitolo successivo.

Altri studi [31, 32] hanno dimostrato come l'Elettroglottografia (EGG), anche nota come laringografia, contenga informazioni utili per identificare la patologia della voce. L'EGG consiste nella misurazione del contatto delle corde vocali attraverso elettrodi posizionati sul collo del paziente e rappresenta quindi un esame semplice e non invasivo. Tuttavia, viene analizzato un segnale correlato alla fonazione e non direttamente il segnale audio. La grande differenza tra modelli basati su dati della EGG o sugli MFCC e modelli come HuBERT o wav2vec è che i primi dispongono di rappresentazioni statiche e derivanti da misurazioni, mentre i secondi imparano rappresentazioni che possono essere migliorate e adattate alla task specifica.

Considerando i vantaggi e gli svantaggi di ogni metodo e il lavoro svolto in [30], in questa tesi si è deciso di adottare l'approccio diretto al segnale audio grezzo, siccome offriva maggiore adattabilità al rilevamento dei disturbi della voce e, generalmente, migliori prestazioni. In particolare, si è utilizzato il modello pre-addestrato HuBERT come feature extractor, data anche la limitatezza dei dati a disposizione per un eventuale addestramento from scratch.

L'applicazione dell'intelligenza artificiale a un ambito delicato come quello medico solleva tuttavia nuove criticità. I modelli più avanzati, composti da numerosi strati e caratterizzati da milioni di parametri, risultano spesso opachi nel loro funzionamento. Il processo decisionale, che conduce alla classificazione in una determinata categoria piuttosto che in un'altra, non è facilmente interpretabile da parte di un essere umano. Le reti neurali possono essere percepite come *scatole nere* che ricevono un input e, senza fornire spiegazioni, producono un output. Nonostante possano essere valutate attraverso test su grandi moli di dati, il dubbio su quale sia il processo decisionale seguito permane: il modello potrebbe seguire un ragionamento errato che, tuttavia, conduce ugualmente a predizioni corrette. Affidarsi ciecamente alle sue decisioni, senza comprenderne le motivazioni, rappresenta un rischio eccessivo, soprattutto quando si ha a che fare con dati sensibili come quelli medici. L'interpretabilità e l'affidabilità delle Deep Neural Network, per i suddetti motivi, rappresentano un ostacolo all'applicazione e diffusione delle reti neurali.

2.3 Explainable AI

Un settore dell'intelligenza artificiale, noto come eXplainable AI (XAI), si è sviluppato parallelamente alla diffusione delle reti neurali profonde e affronta il problema dell'interpretabilità dei modelli complessi. Il suo obbiettivo è rendere più trasparenti e comprensibili le cosiddette *scatole nere*, come le deep neural networks, in modo da offrire maggiore chiarezza agli utenti e aumentarne la fiducia nell'IA. In base

allo stage in cui vengono applicate, le tecniche di XAI possono essere divise in due grandi gruppi: **post hoc** e **explainable-by-design**.

Al primo gruppo appartengono tutte quelle tecniche e strumenti che vengono applicati su modelli già esistenti e quindi applicati dopo l'addestramento, al fine di estrarre dati informativi. Viene in particolare sottolineata la relazione tra le feature di input e l'output del modello, analizzando l'influenza delle prime sul processo decisionale. Tra i metodi post hoc maggiormente diffusi possiamo trovare la *feature importance analysis*, che identifica le caratteristiche più significative per la predizione, *example-based explanations*, che forniscono esempi presi dal dataset utilizzato simili a quello classificato, *perturbation-based methods*, nei quali l'input viene perturbato per osservarne l'impatto sull'output. A quest'ultima tipologia di metodi appartiene LIME (Local Interpretable Model-agnostic Explanations) [33]. Questa tecnica di spiegazione locale, dopo aver perturbato i dati, costruisce un modello semplice e interpretabile basato sulle predizioni del modello complesso. Una spiegazione è definita locale quando la sua validità è limitata alla relazione tra un singolo input e l'output corrispondente, e non è generalizzabile all'intero dominio dei dati. Nel dettaglio, LIME genera un insieme di dati perturbati nell'intorno dell'input, assegnando a ciascuno di questi punti un certo peso. Su questi dati perturbati - e sul loro output corrispondente - viene addestrato un modello facilmente interpretabile (reti lineari o poco profonde). I coefficienti del modello interpretabile rappresentano l'influenza delle feature sulla predizione della classe. Anche SHAP (SHapley Additive exPlanations) [34], framework molto diffuso basato sulla feature importance analysis, allena un'approssimazione del modello originale. In questo caso, però, l'importanza delle caratteristiche viene calcolata utilizzando la teoria dei giochi, e non la perturbazione. A ogni caratteristica (giocatore) viene assegnato il valore di Shapley corrispondente, ovvero un punteggio che ne descrive il contributo alla predizione (vincita). SHAP è in grado di fornire sia spiegazioni locali, ovvero sulla singola predizione, sia spiegazioni globali, ovvero il ruolo delle caratteristiche per tutte le previsioni. Entrambe queste tecniche sono definite *model agnostic*, ovvero possono essere applicate indipendentemente dal tipo di modello. Anche i valori assegnati da SHAP possono variare significativamente da modello a modello. Nonostante forniscano indicazioni su quali caratteristiche influenzino maggiormente le predizioni, LIME e SHAP, come evidenziato da [35], si imbattono nel problema che cercano di risolvere, ovvero l'opacità delle decisioni. Non è chiaro all'utente finale, infatti, come i valori siano assegnati a ciascuna caratteristica [36].

A differenza delle tecniche post hoc, i modelli *explainable-by-design* prevedono un fattore di interpretabilità integrato direttamente nella struttura del modello durante la fase di progettazione. Un esempio di questo tipo di reti sono i Decision Trees (DT). La loro struttura ricorda quella di un grafo ad albero - da cui il nome - dove ogni nodo corrisponde a una condizione. Da ogni nodo si generano due

rami, terminanti a loro volta in un nodo. Partendo da una condizione iniziale (nodo radice), a seconda che sia soddisfatta o meno, si prosegue lungo uno dei due rami. Lo stesso processo viene ripetuto fino ad arrivare a un nodo finale, il quale rappresenta la classe predetta. Il processo decisionale di un Decision Tree è molto semplice e intuitivo, e per questo rientra nella categoria sopra citata. Tuttavia, come dimostrato in [37], in alcuni casi la profondità e la larghezza di questi alberi può crescere linearmente con il numero di parametri dell'input, rendendo questi modelli tutt'altro che comprensibili o interpretabili. Un altro approccio è utilizzato da ProtoPNet (Prototypical Part Network) [38], una rete specializzata nella task di *image classification*. Il modello ispeziona l'immagine individuando parti visivamente significative e successivamente le confronta con i prototipi appresi durante l'addestramento, ovvero esempi significativi di determinate caratteristiche. Basandosi sulla somiglianza tra parti individuate nell'input e i prototipi appresi, il modello assegna una classe all'immagine.

Il problema principale delle tecniche finora presentate, sia di quelle post hoc che explainable-by-design, consiste nel fatto che si limitano a individuare quali parametri influenzino maggiormente la predizione. Questo molto spesso non è sufficiente a comprendere come un modello *percepisca* un input. Per questo motivo, recentemente, la ricerca si sta spostando verso differenti tipologie di interpretazioni. In particolare, un settore in grande crescita è quello della *concept-based explainability* (C-XAI). Un concetto può essere descritto come un attributo di alto livello che descrive un'idea, una caratteristica o una qualità, ed è comprensibile da un essere umano. Prendendo come esempio una classificazione di immagini di animali, una serie di concetti potrebbe essere costituita da *pelo, piume, ali, colore, artigli*. Questo tipo di rappresentazioni è facilmente comprensibile dall'utente finale. ProtoPNet si avvicina, idealmente, alla concept-based explainability, senza però ricorrere a concetti espliciti.

In [39] vengono identificate tre categorie di concept-based explanations in base alla modalità con cui i concetti vengono utilizzati: relazione classe-concetto, associazione nodo-concetto e infine visualizzazione del concetto. La relazione classe-concetto può essere paragonata alla *feature importance analysis*, ma in questo caso, al posto dell'importanza delle caratteristiche, viene analizzata l'influenza di ciascun concetto sulla predizione finale. L'associazione nodo-concetto prevede che ogni concetto sia associato a un componente della struttura del modello, come ad esempio un filtro. La visualizzazione del concetto, impiegata soprattutto in caso di concetti non supervisionati, fornisce una rappresentazione grafica del concetto imparato dal modello, evidenziando quali feature lo rappresentano.

Nel campo della C-XAI, la linea di confine tra metodi post hoc ed explainable-by-design è più sottile: i primi individuano i concetti analizzando le rappresentazioni imparate, mentre la strategia explainable-by-design forza il modello ad apprendere i concetti come rappresentazioni intermedie.

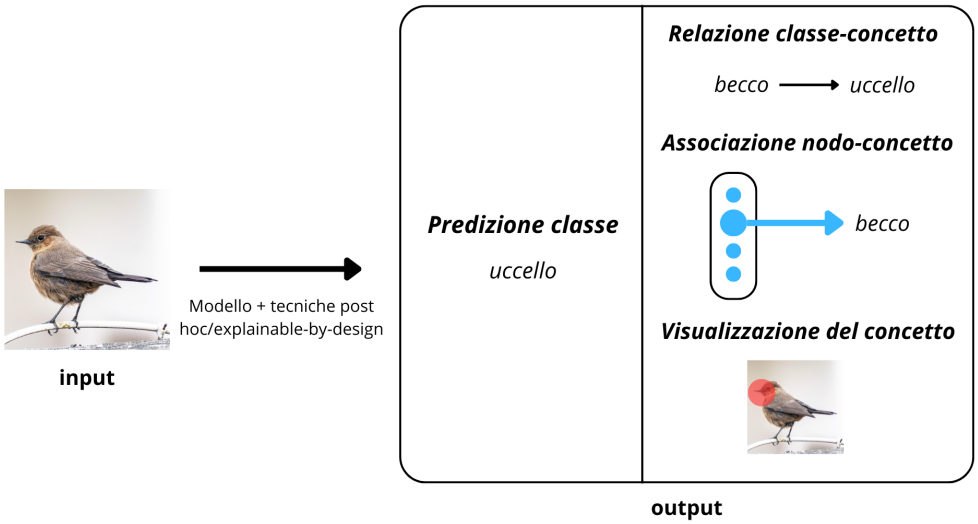

Figura 2.1: I tre tipi di spiegazioni basate sui concetti fornite dalle tecniche di Explainable AI.

Un esempio di C-XAI post hoc è il Quantitative Testing with Concept Activation Vectors (TCAV) [40]. Questa tecnica mira a interpretare lo stato interno di una rete neurale in termini di concetti facilmente comprensibili da un utente umano. Per farlo, utilizza i *concept activation vectors*, vettori che descrivono la presenza di un concetto all'interno della struttura del modello e la sua influenza sulla predizione finale. Per ottenere questi vettori, si deve prima selezionare il concetto di interesse. Successivamente, si addestra un classificatore lineare su un dataset composto da due tipi di esempi, alcuni che rappresentano il concetto, altri che non lo rappresentano. La task del modello lineare sarà classificare le due tipologie di esempi. Ogni layer del modello, dopo l'addestramento, avrà un vettore dei pesi corrispondente. A seconda del layer che si sceglie per individuare il concetto, si ottiene un CAV differente. Questo metodo garantisce grande flessibilità per quanto riguarda la scelta dei concetti, non più limitata a quelli presenti all'interno del dataset di riferimento. Inoltre, il problema dell'annotazione di nuovi dataset viene parzialmente risolta grazie alla possibilità di estrarre CAV da dataset esterni, oppure di utilizzare vettori pre-addestrati. L'approccio TCAV richiede però che i concetti siano linearmente separabili, condizione non sempre soddisfatta. Alcuni concetti, infatti, possono essere semanticamente correlati. L'introduzione delle Concept Activation Regions (CAR) [41] mitiga questa separazione netta, permettendo una rappresentazione dei concetti sparsa su più cluster nello spazio latente della rete neurale.

Mentre TCAV evidenzia la presenza di determinati concetti nelle informazioni apprese dal modello, Causal Concept Effect (CaCE) [42] individua le relazioni

causali tra i concetti e le classi. Questa tecnica si basa sulla perturbazione di un concetto per osservarne l'influenza sull'output: tutti i concetti rimanenti vengono lasciati al valore originale e solo il concetto di interesse viene modificato. Viene poi calcolata la differenza tra la predizione originale e la predizione con il valore perturbato. In questo modo CaCE apprende l'effettiva importanza di ciascun concetto nella predizione finale, evitando correlazioni tra classe e concetto. Per generare sample in cui è presente/assente un determinato concetto, CaCE offre due opzioni: avere diretto controllo sullo stato dei concetti, potendoli inserire o rimuovere manualmente; utilizzare un modello generativo addestrato a rappresentare i sample con e senza determinati concetti. Nel primo caso la perturbazione opererebbe con i valori reali, mentre nel secondo con approssimazioni.

Alcuni studi [43, 44] hanno dimostrato come le reti neurali tendano a imparare autonomamente rappresentazioni di concetti all'interno dei layer. A seconda della profondità a cui si trova il layer, le rappresentazioni apprese sono più o meno complesse. Nei primi layer il modello tende a imparare a distinguere feature di basso livello come ad esempio il colore, mentre nei layer più avanzati riesce a distinguere anche concetti più articolati (*e.g.* oggetti). Basandosi sul lavoro di questi studi, la tecnica nota come Network Dissection (ND) [45] tenta di associare ogni neurone del modello con un concetto specifico. La ND utilizza il BRODEN (BROad and DENsly labled) dataset - che contiene una vasta gamma di annotazioni multimodali - come un dizionario dei concetti. Attraverso l'Intersection over Union (IoU), viene calcolato quanto un nodo corrisponda a un concetto presente nel BRODEN dataset. Il nodo sarà associato al concetto con maggiore IoU. Ovviamente questo metodo non consente di verificare se il modello abbia imparato a rappresentare concetti non presenti nel dizionario ma comprensibili dall'essere umano.

Le tecniche post hoc sono particolarmente utili nel caso si abbiano a disposizione modelli già perfettamente funzionanti e si voglia lasciarne la struttura intatta. In questo modo, si riesce a ottenere un certo grado di interpretabilità dal modello senza doverlo modificare o riprogettare. Tuttavia, non vi è alcuna garanzia che il modello impari sempre concetti interpretabili dall'essere umano. Inoltre, come dimostrato da [46], tecniche come il TCAV sono suscettibili ad attacchi noti come *adversarial attacks*. Essi consistono nel fornire al modello esempi che lasciano invariata la predizione, ma ne modificano l'intepretazione. Questi esempi sono creati appositamente per questo scopo, e possono portare a conseguenze catastrofiche nell'interpretabilità del modello. Nello studio sopra citato, è stato dimostrato come sia possibile rendere il concetto di *strisce* non influente per la predizione della classe *zebra*, e viceversa, rendere il concetto *a pallini* molto rilevante.

I modelli explainable-by-design integrano la rappresentazione concettuale direttamente all'interno della loro struttura. Similmente a quanto accadeva nell'associazione nodo-concetto, anche in questo approccio è previsto che ci sia una correlazione tra i nodi del modello e i concetti. A differenza della tecnica post hoc,

tuttavia, questa associazione viene *forzata* in fase di addestramento e imparata dal modello. L'addestramento di una rete che individui concetti potrebbe sembrare una task impossibile senza annotazioni all'interno di un dataset. Nella pratica, l'addestramento è realizzabile attraverso l'applicazione di tecniche di unsupervised learning, come ad esempio il clustering, adattate alla task. E' il caso delle Interpretable CNN [47]. Senza l'ausilio di annotazioni, questo metodo modifica le reti CNN (Convolutional Neural Network) convenzionali al fine di renderle capaci di apprendere rappresentazioni più significative. Grazie all'aggiunta di una loss per ogni feature map, ciascun filtro viene incoraggiato a codificare una parte specifica di un oggetto, appartenente esclusivamente a una categoria di oggetti. Un'altra rete che adotta un approccio unsupervised basato sui concetti è Self-Explaining Neural Network (SENN) [48]. Attraverso un encoder, il modello apprende autonomamente una serie di concetti interpretabili. Un'altra sezione della rete, il parametrizer, assegna un coefficiente a ciascuno dei concetti predetti, il quale rappresenta il peso del concetto nella predizione finale. Quest'ultima è data dalla somma di ogno concetto moltiplicato per il peso corrispondente. Pesi e concetti vengono estratti *on-the-fly* per ogni sample, rendendoli dipendenti dall'input. SENN, pertanto, fornisce esclusivamente spiegazioni locali.

Tra i modelli che, invece, necessitano di un dataset comprensivo di concetti annotati troviamo i Concept Bottleneck Models (CBM) [49]. Questi modelli prendono il nome dal *bottleneck layer*, un layer dove ogni nodo corrisponde univocamente a un concetto. I CBM si differenziano dalle precedenti tecniche che dividevano l'addestramento in due parti nettamente distinte, la prima in cui si allenava un modello sulla predizione dei concetti e una seconda in cui si allenava un altro modello sulla predizione della classe. I Concept Bottleneck Model forniscono la possibilità di allenare il modello su queste due task in maniera congiunta. Poniamo caso che f descriva come viene imparata la classificazione finale e g la classificazione dei concetti. A seconda di come vengono addestrati, i modelli bottleneck possono essere distinti in tre diverse categorie:

- **independent bottleneck**: per addestrare f vengono utilizzati i concetti reali; in fase di test, però, la predizione è basata sui concetti predetti da g.
- **sequential bottleneck**: viene prima imparata g; successivamente, viene imparata f utilizzando i concetti precedentemente predetti da g.
- **joint bottleneck**: vengono minimizzate le loss di f e g contemporaneamente utilizzando una loss combinata, somma delle due loss.

Nel joint bottleneck, la loss combinata può essere bilanciata grazie all'introduzione di un coefficiente moltiplicativo. In questo modo, si assegna un peso alle due loss e se ne modifica l'influenza nella somma. L'applicazione di modelli CBM, data la rigidità della relazione nodo –> concetto, spesso porta a un calo nelle performance.

Il guadagno che si ha in termini di interpretabilità, a seconda dell'ambito e della task, può giustificare questo calo. Un altro fattore da considerare, è che non sempre il set di concetti individuati è sufficiente a determinare la classe.

L'approccio dei Concept Bottleneck Model richiede un'annotazione completa e precisa, decisamente complessa da ottenere quando ci si interfaccia con realtà pratiche. Per sopperire in parte alla scarsità di annotazioni, gli Incremental Residual CBM [50], oltre al bottleneck, utilizzano dei vettori ottimizzabili che apprendono l'informazione residua non rappresentata dai concetti annotati. Oltre che a migliorare la predizione mantenendo interpretabilità, questi vettori possono essere successivamente tradotti in concetti comprensibili dall'essere umano e aggiunti alla *candidate concept bank*, ovvero all'insieme dei concetti utilizzabili per la predizione.

I modelli Concept Embedding Model (CEM) [51] introducono maggiore flessibilità alla relazione tra nodo e concetto, cercando di limitare o eliminare il calo in accuracy della task, mantenendo, però, una buona interpretabilità. I concetti, invece che associati direttamente ai nodi, sono rappresentati da vettori supervisionati (o *embedding*). In particolare, a ogni concetto corrispondono due embedding, uno rappresentante lo stato attivo del concetto, mentre il secondo lo stato inattivo. Una rappresentazione vettoriale dei concetti incrementa la capacità di apprendimento. L'approccio è simile a un CBM ibrido, con la differenza che l'informazione è completamente dipendente dai concetti e non *concept-agnostic*. Siccome le decisioni non vengono prese in base alla presenza o assenza di un concetto, ma considerando rappresentazioni concettuali più complesse, l'interpretabilità del modello è inferiore rispetto a un CBM.

I modelli concept-based, progettati per essere explainable-by-design, offrono una notevole adattabilità ai cambiamenti di dominio (*domain shift*) tra dataset diversi. Il domain shift si verifica quando dati della stessa tipologia vengono raccolti in condizioni differenti, alterando la distribuzione degli input. Ad esempio, un dataset potrebbe contenere immagini di animali su sfondo neutro, un altro immagini di animali nel loro habitat naturale, e un terzo disegni stilizzati degli stessi animali. Sebbene il soggetto rimanga invariato, cambia il modo in cui esso si manifesta visivamente. Un discorso analogo vale per i segnali audio: si pensi a un dataset con registrazioni vocali pulite in studio e un altro con registrazioni in ambienti rumorosi o con dispositivi diversi. Classificando in base a concetti discriminanti, i modelli concept-based riescono a isolare caratteristiche rilevanti per la task, riducendo l'influenza di informazioni non pertinenti come lo sfondo o il rumore.

Un'altra funzionalità particolarmente interessante di questi modelli è l'*intervention*. Questa tecnica consiste nel sostituire, durante la fase di testing, uno o più concetti predetti dal modello con i corrispondenti concetti reali. In tal modo, è possibile valutare l'influenza esercitata da un singolo concetto, o da un gruppo di concetti, sull'output finale del modello, evidenziando eventuali relazioni causali tra concetti

e classe. L'approccio è simile a quanto visto per alcune strategie di eXplainable AI post-hoc. L'idea della perturbazione dell'input è comune a LIME, con la differenza che quest'ultimo agisce direttamente su caratteristiche dell'input - come pixel, parole ecc. - e non sui concetti. CaCE, viceversa, analizza le relazioni causali dei concetti. I concetti, tuttavia, sono estratti da uno o più dataset esterni, differenziandosi in questo dettaglio dai modelli CBM e CEM. Ciò che unisce concettualmente queste tecniche è l'approccio di *counterfactual reasoning*, ovvero ragionare in funzione ipotetica su un eventuale cambiamento dell'output al variare dell'input.

L'*intervention* si presta in modo naturale ad applicazioni in ambito medico: la possibilità di correggere le predizioni concettuali errate mediante una supervisione umana rende l'utilizzo dell'intelligenza artificiale più sicuro e affidabile. Personale medico esperto può intervenire direttamente sui concetti errati, verificando l'effetto di tali modifiche sulla decisione finale del modello. Una stretta cooperazione tra intelligenza artificiale e operatori umani può così condurre a prestazioni superiori rispetto a quelle ottenute singolarmente, mantenendo trasparenza nel processo.

I Concept Bottleneck Model e i Concept Embedding Model offrono una soluzione semplice e particolarmente adatta al problema dell'interpretabilità delle DNN nella voice disorder detection. La possibilità di giustificare una decisione attraverso la presenza o l'assenza di certe caratteristiche vocali rappresenta un sostanziale miglioramento in termini di chiarezza e trasparenza del modello. La predizione prodotta dalla rete assumerebbe una forma più vicina a una diagnosi medica: il medico (ovvero, il modello concept-based) formula una diagnosi sulla base di determinate caratteristiche rilevate durante l'esame clinico (i concetti). Una pipeline di questo tipo non solo rende più comprensibile il processo decisionale, ma contribuisce anche ad accrescere la fiducia nel modello da parte sia del personale esperto che dei pazienti, i quali non si trovano più davanti a una scatola nera, bensì a una decisione motivata. Come già evidenziato, i modelli concept-based — in particolare i CBM — richiedono dataset dotati di annotazioni dettagliate a livello di concetto. Tuttavia, nell'ambito della speech analysis, i dataset annotati sono ancora molto limitati e spesso di difficile accesso, rappresentando un ostacolo concreto alla diffusione delle tecniche concept-based nel dominio dei segnali audio. Alla scarsità di dataset annotati, si aggiunge la difficoltà nell'identificare concetti validi, da un punto di vista medico, per la predizione di disturbi della voce. Quest'ultimo problema è stato risolto grazie al supporto di personale medico altamente specializzato, che ha seguito lo sviluppo di questo progetto sin dalle fasi iniziali. Per quanto riguarda il dataset, il personale medico ci ha fornito l'accesso all'Italian Pathological Voice (IPV), un dataset proprietario dotato di referti clinici di cui parleremo nel prossimo capitolo, che ha sopperito alla mancanza di annotazioni concettuali.

Grazie al supporto del personale medico e alla disponibilità del dataset IPV, si è scelto in questa tesi di esplorare un approccio concept-based, concentrandosi in particolare sui modelli CBM e CEM, che presentano caratteristiche particolarmente

adatte al problema affrontato. Il metodo seguito e il materiale a disposizione per implementare questa soluzione saranno argomento del prossimo capitolo.

Capitolo 3

Materiale e Metodologia

3.1 Dataset

Il dataset utlizzato per l'annotazione dei concetti e per l'addestramento di tutte le diverse reti impiegate è l'IPV (Italian Pathological Voice), introdotto in [30]. Al suo interno si trova una raccolta di file audio, file di testo, screening e altre informazioni riguardanti visite foniatriche effettuate in diversi ospedali italiani. In particolare, nella cartella di ciascun paziente sottoposto ad una visita, sono presenti i seguenti file:

- **registrazione frasi CAPE-V**: registrazione audio del paziente mentre legge una frase foneticamente bilanciata;
- **registrazione vocale sostenuta**: registrazione audio del paziente mentre pronuncia la vocale /a/ in modo sostenuto;
- **referto medico**: file PDF contenente l'anamnesi del paziente.

Il CAPE-V [52] è una scala standardizzata utilizzata nella valutazione percettiva della voce. Questa tecnica prevede la lettura di una serie di frasi foneticamente bilanciate. Presentando una grande varietà di suoni, le frasi utilizzate da questa scala si rivelano particolarmente utili per riconoscere alterazioni nel parlato. Nel caso dell'IPV dataset, è stato utilizzato l'adattamento italiano di questo standard. Gli audio, in formato ".wav", hanno una durata compresa tra gli 8 secondi e i 24 secondi, siccome la velocità con la quale vengono pronunciate le frasi varia significativamente da un paziente all'altro. Di seguito sono elencate le frasi utilizzate all'interno delle registrazioni:

– Il nuovo libro verde è sulla scatola.

– L'uomo e la donna mangiano le uova.

– Che cosa ha rotto il gatto.

– Le mie nonne non vanno mai al mare.

– Lo zoppo ha toccato il letto.

La pronuncia di una vocale sostenuta è utile per carpire alcune caratteristiche della voce che difficilmente potrebbero essere identificate esclusivamente attraverso l'analisi della lettura delle frasi, come ad esempio vibrazione o instabilità nella voce. Anche in questo caso, i file audio hanno un formato ".wav" e sono compresi tra i 5 secondi e i 10 secondi.

I file contenenti le anamnesi dei pazienti, scritti direttamente dal medico che ha effettuato la visita, si presentano in forma discorsiva e non strutturata. Data la limitata disponibilità di dati annotati in ambito medico a livello globale, i referti clinici si sono rivelati una risorsa fondamentale per lo sviluppo di questo progetto. Poiché l'approccio basato su concetti rappresenta una soluzione ancora poco esplorata nell'ambito dell'elaborazione di segnali audio, è risultato impossibile reperire annotazioni adeguate da dataset esterni. Sebbene esistano alcune annotazioni multi-etichetta nel contesto della classificazione dei sentimenti, l'applicazione di modelli concept-based in ambito medico costituisce, ad oggi, un campo di ricerca ancora largamente inesplorato.

Oltre ai file precedentemente elencati, la documentazione include anche i risultati di esami strumentali e invasivi, che tuttavia non sono stati considerati ai fini del presente lavoro. Il dataset presenta i dati raccolti su un totale di 513 visite, di cui 170 relative a voci eufoniche mentre 343 a voci patologiche. I casi che presentano anche un referto clinico sono 312, di cui 70 appartenenti a voci eufoniche e 242 a voci patologiche. La forte differenza nella distribuzione delle due classi è un fattore di cui si è dovuto tener conto nel corso degli esperimenti.

Nella fase finale di questo lavoro, il personale medico è riuscito a fornirci alcuni dei referti clinici mancanti, ampliando così la dimensione del dataset a disposizione per il training. Il numero dei casi provvisti di anamnesi è così salito da 312 a 385. In particolare, la maggior parte dei referti aggiunti corrisponde a casi di voci eufoniche, bilanciando parzialmente la distribuzione delle classi - 134 eufoniche e 251 patologiche. E' stato successivamente notato che alcuni referti clinici, appartenenti sia al dataset originale che a quello esteso, nonostante contenessero una descrizione di esami di vario tipo e le informazioni generali del paziente, non contenevano una descrizione dettagliata dell'esame percettivo della voce. Pertanto, non venivano citati i concetti definiti come predicibili dalla voce. Nell'annotazione automatica, Gemini classificava - in modo corretto per il suo addestramento - questi concetti come non presenti, falsando sia la fase di training che la fase di evaluation. Nonostante questo problema coinvolgesse solo una piccola frazione dei casi totali, in un dataset di limitate dimensioni come l'IPV avrebbe potuto influire significativamente sui

risultati. Per questo motivo, è stata effettuata una parziale revisione manuale dei dati, soffermandosi in modo particolare sui casi che presentavano valori ambigui - casi patologici che presentavano tutti i concetti predicibili come assenti. Avendo

Data	n° Sani	n° Malati	n° Totale
IPV file audio	170	343	513
IPV referti	70	242	312
Extended IPV referti	134	251	385

Tabella 3.1: La distribuzione delle classi all'interno dell'IPV dataset. Si può notare come, dopo l'aggiunta dei nuovi dati, la distribuzione sani/malati dei casi con referto sia simile a quella del dataset completo

avuto accesso a questi nuovi dati solo verso la conclusione del lavoro, non è stato possibile rieseguire gli esperimenti sul dataset completo nella loro interezza, bensì solo gli esperimenti più significativi. Per maggiore comprensione, nei relativi esperimenti si farà riferimento all'insieme dei dati già in nostro possesso e di quelli nuovi come *extended IPV* o *dataset esteso*. Nella Tabella 3.1 è presente un riassunto del numero di casi totali (IPV file audio), di casi provvisti di referto (IPV referti) e infine dei casi provvisti di referto dopo l'aggiunta dei nuovi dati (Extended IPV referti).

3.2 Modelli

In questa sezione si procederà a un'analisi dettagliata dei modelli impiegati nel presente lavoro, con un approfondimento della loro struttura architetturale e dei fondamenti teorici su cui si basano.

3.2.1 HuBERT

Hidden Unit BERT (HuBERT) [22] è un modello self-supervised per l'apprendimento di rappresentazioni audio. Un modello è definito self-supervised quando non necessita di etichette esplicite per i dati di addestramento, bensì utilizza come target delle label - o pseudo-label - generate dal modello stesso. Si differenzia dalle tecniche unsupervised siccome queste ultime cercano di estrarre rappresentazioni informative dai dati senza ricorrere ad alcun tipo di etichetta. In realtà, HuBERT sfrutta una tecnica unsupervised (clustering) per generare le pseudo-etichette utilizzate per il training auto-supervisionato. Nonostante il primo step non supervisionato, HuBERT rientra comunque nella categoria di modelli self-supervised. La sua struttura segue un approccio basato su un algoritmo di clustering e sulle reti *transformer*,

con una strategia chiamata *masked prediction* che permette di imparare relazioni temporali a lungo raggio.

HuBERT è composto da un encoder convoluzionale del segnale e da un transformer encoder con struttura simile alla rete BERT [24] - modello specializzato in token testuali. Nella Figura 3.1 è rappresentata la pipeline seguita per il pre-addestramento di HuBERT. Alla prima iterazione il segnale grezzo viene trasformato in MFCCs (Mel-Frequency Cepstral Coefficients), ovvero rappresentazioni compatte che descrivono le caratteristiche spettrali della voce sulla scala Mel, ispirata alla percezione uditiva umana. Questa trasformazione viene applicata a ciascun frame[1] del segnale. Dal segnale audio di input $X = [x_1, ..., x_T]$, dove T rappresenta il numero di frame di cui è composto il segnale, si otterrà un numero T di rappresentazioni MFCC. Su queste rappresentazioni viene applicato un algortimo di clustering non supervisionato, *k-means*, il quale assegna a ciascun frame pseudo-etichette discrete, dette *hidden units*. Descritto in termini matematici, $C(X) = [c_1, ..., c_T]$, dove C rappresenta l'algoritmo di clustering e c_t la pseudo-etichetta assegnata al frame. Una volta ottenute le pseudo-etichette, il segnale audio grezzo viene fornito come input all'encoder convoluzionale, il quale genera un embedding per ogni frame. Prima di essere passati al transformer, gli embedding vengono mascherati a blocchi. Nello specifico, ogni embedding, con una probabilità $p = 8\%$, ha la possibilità di diventare l'indice di partenza per un blocco di 10 embedding mascherati. La mascheratura viene effettuata prendendo blocchi contigui di frame sia per rendere più complessa la task - un frame per volta sarebbe facilmente predicibile dai frame vicini - sia per catturare relazioni temporali tra i frame. Il trasformer, ricevuta la sequenza di embedding, cercherà di predire le etichette dei frame mascherati, sulla base del contesto fornito dai frame non mascherati, apprendendo in questo modo rappresentazioni audio contestualizzate e in grado di catturare relazioni temporali a lungo raggio. La loss function viene calcolata esclusivamente sui frame mascherati. Dopo una fase iniziale di pre-training, il clustering non viene più effettuato sulle rappresentazioni MFCC, contenenti informazioni poco complesse, ma bensì sulle rappresentazioni generate dal CNN encoder. Questi embedding contengono informazioni più articolate e specifiche, permettendo un clustering più efficace. Il processo è iterabile: ad ogni iterazione, le rappresentazioni prodotte dall'encoder convoluzionale saranno sempre più efficaci, in un ciclo di continuo raffinamento. Nella Figura 3.1 viene anche rappresentato l'output del modello se usato come encoder di rappresentazioni audio - come nel nostro caso. Viene considerata l'ultima *hidden unit* del transformer HuBERT, composta da embedding audio la cui dimensione varia in base al modello considerato (768 per HuBERT BASE, 1024

[1]Un frame è un intervallo temporale di lunghezza fissa sul segnale audio. In genere, la lunghezza di questo intervallo è di 25ms

HuBERT LARGE, 1280 X-LARGE). Queste rappresentazioni prendono il nome di *hidden states*. Il modello genererà, come output, un hidden state per ogni frame audio ricevuto in input.

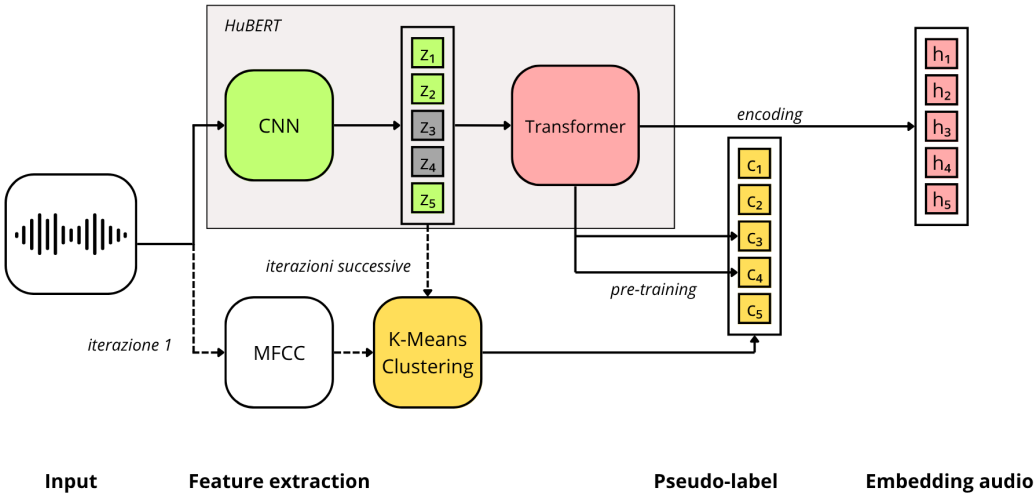

Figura 3.1: Pipeline del modello HuBERT durante il pre-training e la generazione di embedding. Le linee tratteggiate indicano azioni mutualmente esclusive: alla prima iterazione, la feature extraction avviene tramite MFCC, mentre nelle iterazioni successive l'algoritmo di clustering viene applicato direttamente sulle rappresentazioni generate dai layer convoluzionali.

Siccome l'utilizzo di un singolo modello di clustering non offrirebbe abbastanza flessibilità, HuBERT sfrutta un insieme di modelli di clustering, come ad esempio più modelli k-means con diversa dimensione, ottenendo maggiore granularità e permettendo l'apprendimento di informazioni complementari presenti nel segnale audio. L'impiego di rappresentazioni più o meno granulari favorisce l'estrazione di pattern acustici più complessi e strutturati.

Il meccanismo dell'attenzione, fulcro del funzionamento delle reti transformer, permette di stabilire l'influenza di ciascun token sul token preso in esame $x^{(c)}$. Il procedimento prevede il calcolo degli *attention scores*, a_i i quali indicano il peso del token $x^{(i)}$ nella rappresentazione di $x^{(c)}$. Ogni attention score viene calcolato attraverso l'impiego della tripletta *query* (Q), *key* (K) e *value* (V), e delle rispettive tre matrici dei pesi, W_q, W_k, W_v, condivise per tutti i token di input. Il primo step, definendo $x^{(c)}$ il token su cui vogliamo calcolare l'attenzione, è calcolare Q,K e V

per ogni token:

$$Q^{(i)} = W_q x^{(i)}$$
$$K^{(i)} = W_k x^{(i)}$$
$$V^{(i)} = W_v x^{(i)}$$

dove $i \in [1, T]$ con T numero di token. Successivamente, viene calcolato il prodotto scalare tra la query $Q^{(c)}$ del token $x^{(c)}$ e la key $K^{(i)}$ del token $x^{(i)}$. Il prodotto viene normalizzato sulla radice quadrata di d_k, dimensione della matrice K, e infine viene applicata una funzione *softmax* per ottenere gli attention scores:

$$a_{c,i}(Q^{(c)}, K^{(i)}) = softmax(\frac{Q^{(c)} K^T_{(i)}}{\sqrt{d_k}})$$

Per ottenere la rappresentazione finale del token $x^{(c)}$, viene affettuata una somma pesata dei valori di ciascun token, moltiplicandoli per il loro rispettivo attention score:

$$z^{(c)} = \sum_{i=1}^{T} a_{c,i} V^{(i)}$$

I pesi delle matrici W vengono aggiornati ad ogni iterazione tramite backpropagation. Siccome l'input spesso presenta diverse caratteristiche importanti nella determinazione dell'influenza dei token, HuBERT utilizza la *multi-head attention*. Ogni *attention head* è composta dalle tre matrici dei pesi (W_q, W_k, W_v). Aggiungendo attention head differenti, ognuna di esse può focalizzarsi su una determinata caratteristica dell'input, rendendo la rappresentazione più ricca e specifica.

L'intero processo di masking, clustering e di predizione delle pseudo-etichette avviene esclusivamente durante la fase di pre-addestramento del modello. Nel nostro caso, HuBERT è stato utilizzato come pre-trained encoder su cui è stato effettuato fine-tuning. I pesi delle matrici dell'attenzione sono aggiornati attraverso la backpropagation ed il modello non fa più ricorso alla mascheratura o al clustering. L'addestramento *from scratch*, specialmente nel nostro caso, non sarebbe stata la soluzione migliore, date le limitate dimensioni del dataset a disposizione. I transformer, a causa dell'elevato numero di parametri (95 M per HuBERT BASE), richiedono enormi quantità di dati per un training corretto ed evitare overfitting. Avendo imparato, attraverso il pre-addestramento, rappresentazioni efficaci e generalizzabili per diverse task, un ulteriore addestramento su una limitata quantità di dati risulta sufficiente per raffinare le suddette rappresentazioni.

L'architettura di HuBERT presenta tre possibili configurazioni, ognuna delle quali possiede un diverso livello di complessità: *BASE, LARGE e X-LARGE*. Confrontando il numero di parametri di ciascuna configurazione, HuBERT BASE possiede 95M di parametri, mentre HuBERT LARGE e X-LARGE possiedono rispettivamente 317M e 964M. Data la grande complessità degli ultimi due modelli,

per questa tesi si è scelto di utilizzare il modello HuBERT BASE. L'encoder convoluzionale, comune a ciascuna configurazione, è composto da 7 layer con stride e dimensione del kernel decrescenti. L'encoder transformer del modello BASE presenta 12 *transformer blocks* e 8 *attention heads*. A differenza delle altre configurazioni, il modello BASE utilizza una tecnica di regolarizzazione nota come LayerDrop [53]. Il LayerDrop introduce la possibilità di saltare un intero layer del transformer durante l'addestramente con una probabilità, nel caso di HuBERT, del 5%. Questa tecnica facilita notevolmente anche un eventuale processo di *pruning*, che consiste nel rimuovere parti superflue di un modello per semplificarne la struttura e renderlo più efficiente. Nei modelli più complessi, come HuBERT LARGE e X-LARGE, al LayerDrop vengono preferite tecniche di regolarizzazione più adatte a reti di grandi dimensioni, come ad esempio un dropout più aggressivo.

3.2.2 Rete neurale convenzionale

La rete neurale convenzionale è stata realizzata utilizzando come base il modello pre-addestrato HuBERT nella configurazione BASE. Considerata la limitata disponibilità di dati di addestramento, si è preferito effettuare esclusivamente il fine-tuning sul IPV dataset piuttosto che allenare un modello *from scratch*. Per la task binaria, è stata aggiunta una testa di classificazione composta da layer lineari. Il numero e le dimensioni di questi layer intermedi sono stati oggetto degli esperimenti condotti su questo tipo di rete. L'output finale del modello è composto dai due logit appartenenti alle classi *sano* e *malato*. La rete lavora in modalità end-to-end, ricevendo come input il segnale audio grezzo e restituendo direttamente la classe predetta, senza fornire rappresentazioni intermedie.

3.2.3 CBM e CEM

Come menzionato nel primo capitolo, i Concept Bottleneck Models e i Concept Embedding Models appartengono alla famiglia dei modelli *explainable-by-design*. Queste reti incorporano fattori interpretabili direttamente nella loro struttura, rendendo intrinsecamente più chiaro e trasparente il processo decisionale. In particolare, i modelli concept-based sfruttano l'idea di *concetto*, ovvero una rappresentazione - astratta o meno - comprensibile da un essere umano dell'informazione appresa da un modello. L'architettura delle reti concept-based può essere schematizzata come:

$$x - - > c - - > y$$

dove x rappresenta l'input, c le rappresentazioni dei concetti e y la classe finale predetta. Le rappresentazioni dei concetti - diverse a seconda del modello concept-based impiegato - sono ottenute attraverso un mapping dell'input, definito come $c = g(x)$. La classificazione finale, invece, avviene mappando le reppresentazioni

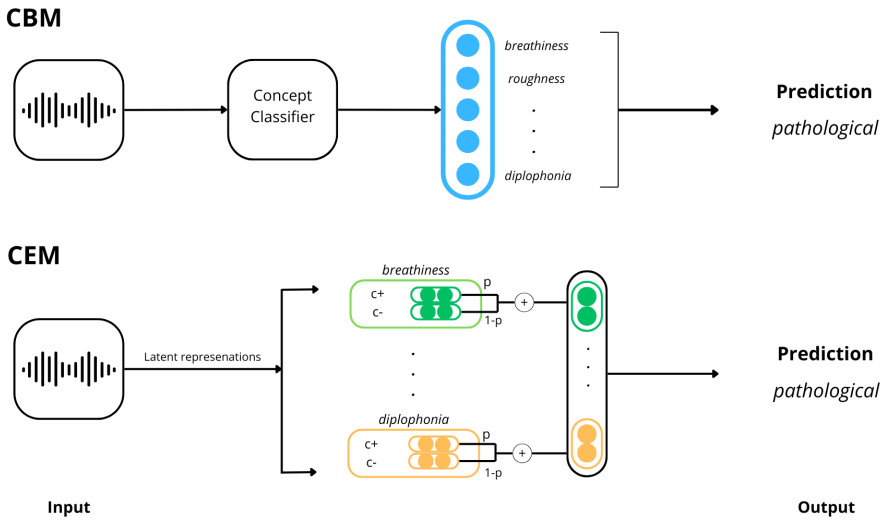

Figura 3.2: Le due architetture messe a confronto. Mentre il CBM predice direttamente i concetto come presenti o assenti, il CEM calcola la probabilità di attivazione p per ciascun concetto; tramite una somma pesata, l'embedding corrispondente allo stato attivo e allo stato negativo vengono combinati.

c nello spazio delle classi: $y = f(c) = f(g(x))$. Lo schema precedente può quindi essere riscritto come:

$$x - - > g(x) - - > f(g(x))$$

I Concept Bottleneck Model [49] prendono il nome dal cosiddetto *bottleneck layer*, uno strato della rete in cui ogni neurone rappresenta un concetto interpretabile e semanticamente definito. Questo strato costituisce un vero e proprio "collo di bottiglia", in quanto il numero di neuroni viene ridotto per coincidere esattamente con il numero di concetti utilizzati come supervisione. A ciascun concetto, attraverso l'applicazione di funzioni non lineari all'output del bottleneck layer, viene assegnato un valore *attivo* (1) o *inattivo* (0). La classificazione avviene esclusivamente sulla base di questi valori dei concetti. La funzione $f(c)$ non agisce più su rappresentazioni dell'audio, ma su rappresentazioni binarie dei concetti. La relazione tra la classe e i concetti è quindi immediata, e il processo decisionale del modello è facilmente comprensibile. Per incoraggiare un corretto apprendimento dei concetti, questa tipologia di modelli utilizza una loss dedicata - la concept loss. Il processo di addestramento prevede l'utilizzo di una loss dedicata anche per quanto riguarda la classificazione binaria sano/malato - la task loss. In questo lavoro si è scelto di utilizzare un CBM di tipo *joint bottleneck*, il quale ottimizza contemporaneamente

sia la concept loss che la task loss, utilizzando una loss combinata:

$$\hat{f},\ \hat{g} = \arg\min_{f,g} \sum_{i} \left[\mathcal{L}_Y(f(g(x^{(i)})), y^{(i)}) + \sum_{j} \lambda\, \mathcal{L}_{C_j}(\hat{c}; c^{(i)}) \right]$$

con $\lambda > 0$. Il coefficiente λ è un parametro che *pesa* il contributo della concept loss ($\mathcal{L}_{C_j}$) nel calcolo della loss totale, assegnandole più o meno importanza all'interno della somma. Ponendo $\lambda = 0$, il Concept Bottleneck Model sarebbe equivalente ad un modello standard, siccome le rappresentazioni intermedie non sarebbero incoraggiate ad apprendere i concetti e la predizione si baserebbe su rappresentazioni audio ottimizzate esclusivamente per la task finale. Nel CBM utilizzato in questo lavoro, si è scelto di pesare la loss utilizzando due parametri λ: λ_c per la concept loss e λ_{sm} per la task loss. La loss ottimizzata dal Concept Bottleneck Model, quindi, può essere definita come:

$$\mathcal{L}_{tot} = \lambda_c \mathcal{L}_c + \lambda_{sm} \mathcal{L}_{sm}$$

Si è scelta questa strategia per mantenere una loss totale normalizzata piuttosto che limitarsi a scalare numericamente solo la concept loss, garantendo maggiore stabilità e chiarezza alla loss totale.

Nei Concept Embedding Model [51] il bottleneck layer nodo-concetto non è più presente: la corrispondenza diretta tra neurone e concetto è sostituita da una rappresentazione più flessibile dei concetti. Prendendo come esempio un classificatore di mezzi di trasporto basato su immagini, il concetto *vola* comprende diverse sfaccettature. La modalità di volo di un elicottero e di un aereo sono completamente diverse, e questa varietà in un modello bottleneck andrebbe perduta. Per questo motivo, i CEM sostituiscono la rappresentazione binaria dei concetti con una rappresentazione multidimensionale, in cui ciascun concetto viene descritto da due embedding di valori, uno associato all'attivazione positiva del concetto ($\hat{c}^+$) e l'altro alla sua non attivazione ($\hat{c}^-$). Se un concetto è assente, il modello si basa esclusivamente sull'embedding negativo; viceversa, in presenza del concetto, viene utilizzato l'embedding positivo. Nei casi pratici, spesso un concetto non è né completamente assente, né completamente attivo. Per questo motivo, i CEM calcolano anche la probabilità di attivazione p per ciascun concetto. Questa probabilità viene calcolata attraverso uno scoring system basato sulla concatenazione dei due embedding: $p_i = s([\hat{c}_i^+, \hat{c}_i^-]^T)$. Come è possibile osservare nella Figura 3.2, per ottenere l'embedding finale rappresentante il concetto, il modello esegue una somma pesata degli embedding, ciascuno scalato, rispettivamente, per la probabilità di attività p_i e per quella di inattività $1 - p_i$:

$$\hat{c}_i = (p_i \hat{c}_i^+ + (1 - p_i)\hat{c}_i^-)$$

Gli embedding ottenuti per ciascun concetto vengono concatenati in un singolo layer, simile al bottleneck layer, il quale costituirà l'input del classificatore finale

$f(c)$. La dimensione di questo layer sarà, quindi, uguale alla dimensione di un singolo embedding[2] moltiplicata per il numero di concetti. La predizione della classe avverrà, perciò, sulla base di rappresentazioni più dettagliate e flessibili dei concetti, ovvero gli embedding. Per garantire l'associazione embedding-concetto, l'embedding layer predice comunque ogni concetto come attivo o inattivo - in modo binario - attraverso il vettore delle probabilità calcolate secondo lo scoring system descritto precedentemente, sul quale viene ottimizzata la concept loss. Come fatto per i CBM, i CEM sono stati implementati come modelli end-to-end, ovvero ottimizzando una loss combinata definita come la somma della concept loss e della task loss. Generalmente, l'impiego dei Concept Embedding Model comporta un incremento dell'accuratezza sulla task prediction rispetto all'utilizzo dei Concept Bottleneck Model, potendo sfruttare rappresentazioni più complesse e articolate. Allo stesso tempo, l'interpretabilità risente di questa maggiore complessità nelle rappresentazioni siccome la relazione concetto-classe non è immediata come nel bottleneck layer.

Entrambe le tipologie di reti basate su concetti - sia i CBM classici che i CEM - consentono l'intervento diretto sui concetti a test-time. In presenza di predizioni errate, è possibile correggere manualmente i concetti predetti e osservarne l'impatto sulla classificazione finale. Questa pratica, nota come *intervention*, consente di valutare quanto la predizione dipenda da ciascun concetto e se il modello abbia appreso correttamente le relazioni causali tra concetti e classe. La possibilità di modificare i concetti a test time permette il *counterfactual reasoning*. Questa pratica esplora scenari differenti di predizione modificando parzialmente l'input e osservandone le conseguenze sull'output. E' quindi possibile modificare alcuni concetti predetti e osservare come cambi la predizione finale del modello, senza la necessità di doverlo riaddestrare. Il counterfactual reasoning è utile soprattutto per individuare le relazioni causali nei modelli, evidenziando quanto un concetto incida nella predizione della classe. L'intervention si rivela particolarmente utile solo nei casi in cui l'accuratezza dei concetti sia elevata. Nel caso contrario, non ci sarebbe allineamento tra concetti reali e concetti predetti dal modello e fornire l'input corretto al classificatore finale non porterebbe ad un miglioramento delle prestazioni.

Per la costruzione dei modelli concept-based è stato impiegato HuBERT, utilizzato in questo contesto come feature extractor. Le rappresentazioni audio da esso ottenute si sono dimostrate adatte a un'analisi concettuale, come richiesto dalla task. Il CBM è stato realizzato aggiungendo una testa di classificazione dei concetti composta da due layer lineari, in cui il secondo rappresenta il cosiddetto bottleneck

[2]La dimensione degli embedding è arbitraria e può essere settata prima del training. Dimensioni comunemente utilizzate sono 16 e 32.

layer. La testa di classificazione affronta il problema in modalità multi-etichetta, producendo per ciascun concetto un logit che ne indica la presenza o l'assenza. In aggiunta al classificatore dei concetti, è stata inserita un'altra testa di classificazione incaricata di predire se la voce è eufonica o patologica. Questo classificatore finale, che riceve in input il tensore dei concetti, è composto da due layer lineari. Per il modello CEM si è seguita una struttura pressochè identica a quella del CBM. Il bottleneck layer è stato sostituito da un *embedding layer*, che associa a ciascun concetto un vettore in uno spazio latente condiviso. Gli embedding ottenuti vengono concatenati e forniti in input al classificatore finale sotto forma di tensore lineare. Oltre a generare tali embedding, il layer è anche incaricato di predire la presenza o l'assenza di ciascun concetto. Anche in questo caso, il classificatore finale è composto da due layer lineari. Tuttavia, a differenza del modello CBM, l'input ricevuto non è costituito da valori binari relativi ai concetti, bensì dalla concatenazione degli embedding corrispondenti.

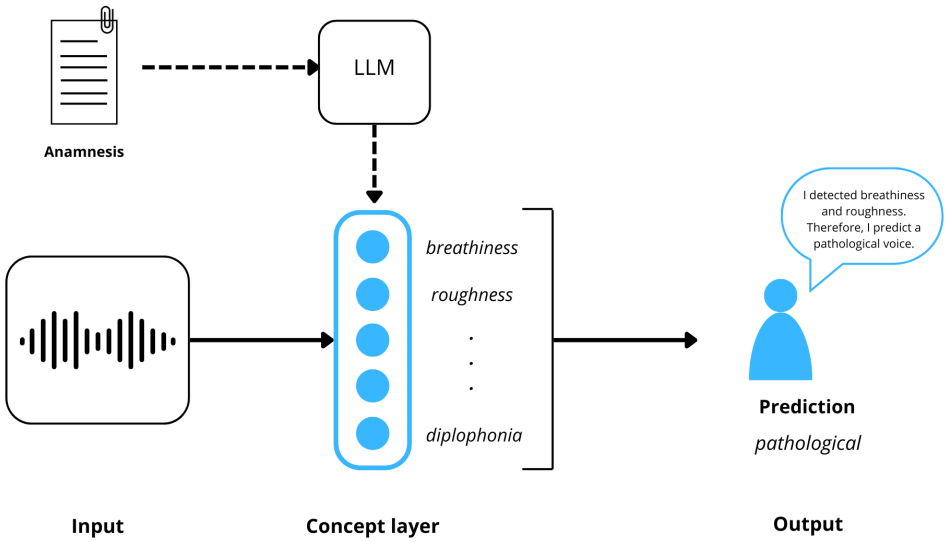

Figura 3.3: Il concept-based framework utilizzato in questo lavoro. La linea tratteggiata indica le procedure adottate esclusivamente in fase di addestramento del modello.

3.3 Metodo

L'approccio adottato in questa tesi, riassunto dalla Figura 3.3, consiste nell'addestramento di due tipologie di reti concept-based, i Concept Bottleneck Model (CBM) e i Concept Embedding Model (CEM), confrontandone le prestazioni con

quelle di un modello standard. I concetti, estratti dai referti clinici attraverso un Large Language Model, vengono forniti al modello esclusivamente in fase di training, in modo da incoraggiarne l'apprendimento. In fase di test, i concetti vengono predetti dal modello e le annotazioni vengono utilizzate per calcolare la concept accuracy. La task finale presa in esame è la classificazione binaria delle voci in eufoniche o patologiche. L'obbiettivo è verificare se sia possibile raggiungere prestazioni competitive mantenendo un certo grado di interpretabilità.

In primo luogo, si sono identificati i concetti candidati all'estrazione, ovvero concetti contenuti nelle anamnesi che possiedono valenza clinica e utili alla classificazione. I concetti sono stati poi estratti dai referti, confrontando i risultati ottenuti da due diversi LLMs. E' stata addestrata una rete neurale convenzionale sul medesimo dataset, al fine di valutare l'efficacia dell'approccio interpretabile rispetto al modello end-to-end. Data la contenuta dimensione del dataset, si è ritenuto opportuno utilizzare una rete pre-addestrata specializzata in rappresentazioni audio (HuBERT). Questa tipologia di modello è stata impiegata sia per l'approccio concept-based che per quello standard. Successivamente, sono stati addestrati i due modelli concept-based utilizzando le annotazioni generate dal LLM. Vediamo ora più nel dettaglio ciascuna delle fasi.

3.3.1 Annotazione dei concetti

L'addestramento dei modelli concept-based richiede che i concetti da imparare siano annotati. Come detto in precedenza parlando della struttura di questi modelli, la limitata disponibilità di dati pubblici ha escluso la possibilità di ricavare queste annotazioni da dataset esterni. All'interno dell'IPV, per la maggior parte delle visite, si può trovare un file contenente il referto medico, contente le informazioni generali del paziente e i risultati degli esami condotti. In particolare, i referti contengono una descrizione di varie caratteristiche percebili attraverso un esame percettivo della voce. Combinando un'analisi di frequenza delle parole all'interno dei referti con un'analisi manuale, è stato possibile selezionare un set di 14 concetti candidati, ovvero concetti che potessero descrivere determinate caratteristiche della voce e permetterne una distinzione in eufoniche e patologiche. Una volta individuati i concetti di interesse, si è reso necessario assegnare un valore a ciascuno di essi per ogni referto presente all'interno del dataset. Non essendo i dati strutturati secondo un formato specifico e condiviso, ma scritti in forma discorsiva, si è dovuto ricorrere ad un approccio interpretativo. Una famiglia di modelli specializzata nella comprensione e generazione del linguaggio naturale è quella dei Large Language Models (LLMs). Questi modelli vengono impiegati nella realizzazione dei modelli conversazionali noti come *chatbot* (*e.g.* ChatGPT), sistemi che simulano l'interazione con un essere umano intepretando quanto comunicato dell'utente e generando una risposta coerente in linguaggio naturale. L'input fornito ai chatbot sotto forma

di testo è noto come *prompt*, e a seconda della sua struttura, articolazione e sintassi, l'output generato dal modello può variare significativamente. Questa caratteristica ha dato origine ad una disciplina nota come *prompting*, la quale studia le tecniche e le strategie per formulare prompt efficaci al fine di ottenere l'output desiderato. Per estrarre i concetti dai referti medici presenti nell'IPV dataset è stato adottato questo approccio. In particolare, si è fatto uso della strategia del *few-shot prompting*, la quale consiste nell'includere all'interno del prompt alcuni esempi di input e del relativo output atteso. Ecco il prompt utilizzato per il modello Gemini:

```
prompt = f"""

Estrai i seguenti concetti dal testo fornito, restituendo un oggetto JSON completo. Ogni concetto deve avere esattamente uno dei valori ammessi.
[
  "fumo": unici possibili valori: "si"/"no",
  "uso voce professionale": unici possibili valori: "si"/"no",
  "disfonia": unici possibili valori: "no"/"lieve"/"lieve-moderata"/"moderata"/"severa",
  "onda mucosa irregolare": unici possibili valori: "si"/"no" ,
  "mucose": unici possibili valori: "rosee"/"eutrofiche"/"iperemiche"/"non presente",
  "diplofonia": unici possibili valori: "si"/"no",
  "strain": unici possibili valori: "si"/"no",
  "roughness": unici possibili valori: "si"/"no",
  "breathiness": unici possibili valori: "si"/"no",
  "asthenicity": unici possibili valori: "si"/"no",
  "fonastenia": unici possibili valori: "si"/"no",
  "atteggiamento glottico a clessidra": unici possibili valori: "si"/"no",
  "disodia": unici possibili valori: "si"/"no",
  "sesso": unici possibili valori: "maschio"/"femmina"
]
Se i concetti non sono espressamente citati nel testo, il loro
valore deve essere "no" oppure 'non presente' solo per concetti che includono esplicitamente 'non presente' tra i valori ammessi (come 'mucose').
Attenzione: concetti come asthenicity, fonastenia, roughness o strain vanno marcati "sì" solo se sono espressamente citati nel testo.
Ecco un esempio del formato output da ritornare e seguire fedelmente:
{{
  "fumo": (tua risposta),
  "uso voce professionale": (tua risposta),
  "disfonia": (tua risposta),
  "onda mucosa irregolare": (tua risposta),
  "mucose": (tua risposta),
```

```
  "diplofonia": (tua risposta),
  "strain": (tua risposta),
  "roughness": (tua risposta),
  "breathiness": (tua risposta),
  "asthenicity": (tua risposta),
  "fonastenia": (tua risposta),
  "atteggiamento glottico a clessidra": (tua risposta),
  "disodia": (tua risposta),
  "sesso": (tua risposta)
}}

Esempio di input/output:
Input:
{testo1}
Output:
{{
    "fumo": "no",
    "uso voce professionale": "si",
    "disfonia": "no",
    "onda mucosa irregolare": "si",
    "mucose": "rosee",
    "diplofonia": "no",
    "strain": "no",
    "roughness": "si",
    "breathiness": "si",
    "asthenicity": "no",
    "fonastenia": "si",
    "atteggiamento glottico a clessidra": "no",
    "disodia": "no",
    "sesso": "femmina"
}}

Esempio di input/output:
Input:
{testo2}
Output:
{{
    "fumo": "no",
    "uso voce professionale": "si",
    "disfonia": "lieve",
    "onda mucosa irregolare": "si",
    "mucose": "non presente",
    "diplofonia": "no",
    "strain": "si",
    "roughness": "si",
    "breathiness": "si",
    "asthenicity": "no",
    "fonastenia": "si",
    "atteggiamento glottico a clessidra": "si",
```

```
    "disodia": "no",
    "sesso": "femmina"
}}

Esempio di input/output:
Input:
{testo3}
Output:
{{
    "fumo": "si",
    "uso voce professionale": "no",
    "disfonia": "severa",
    "onda mucosa irregolare": "no",
    "mucose": "iperemiche",
    "diplofonia": "si",
    "strain": "si",
    "roughness": "no",
    "breathiness": "no",
    "asthenicity": "si",
    "fonastenia": "no",
    "atteggiamento glottico a clessidra": "no",
    "disodia": "si",
    "sesso": "maschio"
}}

Esempio di input/output:
Input:
{testo4}
Output:
{{
    "fumo": "no",
    "uso voce professionale": "no",
    "disfonia": "lieve-moderata",
    "onda mucosa irregolare": "si",
    "mucose": "non presente",
    "diplofonia": "no",
    "strain": "si",
    "roughness": "no",
    "breathiness": "no",
    "asthenicity": "no",
    "fonastenia": "no",
    "atteggiamento glottico a clessidra": "no",
    "disodia": "no",
    "sesso": "maschio"
}}

Testo da cui estrarre i concetti:

{ text }
```

```
128
129 Output:
130 """
```

Con una corretta struttura del prompt e fornendo esempi rappresentativi, il modello generalizza il comportamento desiderato, restituendo le informazioni estratte nel formato richiesto. Per gli esempi sono stati utilizzati quattro testi presi dal dataset e leggermente modificati al fine di rappresentare nel modo più chiaro e completo possibile ciascun concetto da estrarre. La prima sezione del prompt è dedicata ad una breve descrizione della task da svolgere. Sono indicati esplicitamente e in modo strutturato i concetti da identificare all'interno del testo fornito come input. Successivamente, vengono riportate alcune precisazioni su come valorizzare il set di concetti. Per aiutare il modello a comprendere la struttura dati desiderata in output, si sono inseriti quattro esempi testuali di input con il corrispondente output atteso. Infine, si è inserito il testo di input. Fornire un prompt con una struttura ben definita, chiara e coerente è fondamentale per aiutare il modello a generalizzare in modo ottimale. Durante la fase sperimentale sono stati considerati due modelli di chatbot: Gemini 1.5-pro [9] e Mistral 7B-v0.1-hf [54]. Gemini è una famiglia di modelli proprietari, particolarmente potenti e accessibili esclusivamente tramite API. Al contrario, Mistral 7B è un modello open-source progettato per bilanciare efficienza e complessità. Grazie alla sua accessibilità, Mistral risulta particolarmente adatto a scopi accademici, come la ricerca e la sperimentazione. Al fine di individuare il modello più adatto all'annotazione automatica dei concetti, è stato creato un test set composto da 69 casi selezionati dal dataset principale, per i quali è stata effettuata un'annotazione manuale di ciascun concetto. Entrambi i modelli sono stati utilizzati per annotare i sample presenti nel test set, impiegando prompt leggermente differenti nella struttura. Poichè i modelli concept-based richiedono che ciascun concetto sia espresso in forma binaria (0 = assente, 1 = presente), si è reso necessario applicare il *one-hot encoding* ai concetti che potevano assumere più di due valori. Ad esempio, il concetto "disfonia" può assumere i valori *assente, lieve, moderata* e *severa*. Attraverso il one-hot encoding, un singolo concetto multiclasse viene trasformato in quattro concetti distinti (disfonia assente, disfonia lieve, ecc.), ciascuno rappresentato come presente o assente. In questo modo, il numero di concetti candidati è salito da 14 a 20. Dei 513 casi presenti all'interno del dataset, solo 312 possiedono il relativo referto scritto. E' stato possibile, di conseguenza, effettuare l'annotazione automatica esclusivamente su questa frazione del dataset. Vediamo più nel dettaglio i concetti candidati:

- **Fumo, Uso professionale della voce, Sesso**: informazioni generali sul paziente;

- **Disfonia assente/lieve/moderata/severa**: grado di alterazione della voce percepito;
- **Onda mucosa irregolare**: onda prodotta dalla mucosa che ricopre le corde vocali durante la loro vibrazione;
- **Mucose rosee/iperemiche/eutrofiche**: stato delle mucose orofaringee. *Rosee* e *eutrofiche* rappresentano mucose sane, mentre *iperemiche* mucose arrossate;
- **Diplofonia**: emissione simultanea di due suoni a tonalità differente dalla laringe, voce sdoppiata;
- **Strain**: voce pressata, sforzata;
- **Roughness**: voce rauce, raucedine;
- **Breathiness**: voce soffiata, fuga d'aria in fonazione;
- **Asthenicity**: voce debole, mancanza di volume;
- **Fonastenia**: affaticamento nel controllo della voce;
- **Atteggiamento glottico a clessidra**: la chiusura glottica assume una forma simile ad una clessidra;
- **Disodia**: alterazione della voce durante il canto;

Per garantire che i concetti candidati avessero una rilevanza clinica effettiva e fossero pertinenti alla task analizzata, è stato chiesto un consulto tecnico al personale medico che collabora allo sviluppo di questo lavoro e che ha curato l'annotazione dei dati. Il risultato di questa consulenza è stato un sostanziale ridimensionamento del numero dei concetti da utilizzare per la classificazione sani/malati, riducendoli a 14. Come mostrato nella Tabella 4.4, alcuni concetti (*e.g.* mucose rosee, mucose eutrofiche, ecc), come indicato dal medico, non possono essere rilevati tramite un esame percettivo della voce, trattandosi di caratteristiche fisiche osservabili solo tramite indagini strumentali o esami invasivi. Questa tipologia di concetti è stata esclusa dal set destinato alla classificazione. Un secondo gruppo di concetti ricade sotto la categoria dei sintomi (*e.g. fonastenia*) riferiti dal paziente e risulta pertanto non inferibile da un'analisi acustica automatica. In ultima istanza, concetti come *fumo* o *uso professionale della voce* rappresentano abitudini del paziente. Queste ultime due categorie rivestono grande importanza nella determinazione della qualità vocale e possono essere fornite direttamente dal paziente, rendendo superflua la loro predizione automatica mediante una rete neurale. A dimostrazione della validità dei concetti individuati per il cadidate set finale, è necessario sottolineare che 8

Tabella 3.2: Scelta dei concetti per delineare il concept set finale.

Predicibili	Forniti dal paziente	Esclusi
Disfonia assente	Fumo	Onda mucosa irregolare
Disfonia lieve	Uso professionale della voce	Mucose rosee
Disfonia moderata	Sesso	Mucose iperemiche
Disfonia severa	Fonastenia	Mucose eutrofiche
Diplofonia	Disodia	Atteggiamento glottico a clessidra
Strain		
Roughness		
Breathiness		
Asthenicity		

dei 9 concetti predicibili rientrano tra i parametri della scala GRBAS [55], uno strumento utilizzato nell'esame percettivo della voce. In particolare, i 4 concetti legati alla disfonia corrispondono alla lettera "G" della scala (grado di disfonia), *roughness* alla "R", *breathiness* alla "B", *asthenicity* alla "A" e infine *strain* alla "S".

Il procedimento adottato per integrare i concetti forniti dal paziente con quelli predetti automaticamente sarà oggetto di trattazione nel paragrafo successivo.

3.3.2 Addestramento dei modelli

Il passo successivo all'annotazione dei concetti è stato l'addestramento delle diverse tipologie di reti neurali. Come prima cosa, si è deciso di addestrare una rete neurale convenzionale end-to-end in modo da avere una base di confronto per le prestazioni dei modelli interpretabili. Nonostante il dataset sia composto da più di 500 esempi, solo 312 di questi erano forniti di un referto clinico che ne ha permesso l'annotazione dei concetti. Per garantire un'uguale condizione di partenza per i modelli standard e interpretabili, si è deciso di addestrare la rete convenzionale esclusivamente sugli esempi che possedevano anche l'anamnesi scritta. I segnali audio utilizzati sono quelli contenenti le frasi CAPE-V. Rispetto alle vocali sostenute, le frasi contengono maggiore informazione e permettono una classificazione più accurata sia per la task che per i concetti. Nonostante le vocali sostenute contengano comunque informazioni importanti e necessarie per la classificazione di alcuni concetti, come è stato evidenziato in [30], un approccio che concatena entrambi i segnali audio in un input combinato per un unico modello non garantisce migliori prestazioni. Siccome la loro lunghezza può variare notevolmente, sì è selezionata la lunghezza massima tra tutti i campioni e successivamente è stato applicato il padding a tutti gli audio di durata inferiore. Grazie all'impiego di maschere e meccanismi di attenzione,

i modelli basati su transformer sono in grado di individuare correttamente le porzioni informative del segnale, ignorando il padding. Per l'addestramento delle reti basate sui concetti si è usato lo stesso metodo di processamento dei dati audio. Per quanto riguarda i concetti, si sono individuati due gruppi distinti: i concetti forniti direttamente dal paziente, tra cui i sintomi e le informazioni generali, e i concetti identificabili dalla voce. Nella rete CBM, il primo gruppo non viene predetto dal modello, ma fornito direttamente al classificatore finale come ground truth. Il secondo gruppo, viceversa, viene predetto dal classificatore di concetti sulla base del segnale audio. I due gruppi di concetti vengono concatenati e forniti al classificatore finale, che effettua la predizione della classe potendo disporre di tutti i concetti. Questa tecnica simula l'approccio adottato negli esami percettivi della voce, in cui il medico dispone sia delle informazioni generali sul paziente, sia della sintomatologia da lui riferita. Per la rete CEM, il procedimento differisce per come vengono gestiti i concetti forniti direttamente dal LLM. Gli embedding, uno rappresentante lo stato attivo e uno quello non attivo per ciascun concetto, vengono appresi tramite backpropagation. In questa tesi, si è deciso di adottare lo stesso metodo anche per i concetti forniti come ground truth. La differenza nell'addestramento risiede nel fatto che gli embedding dei concetti predicibili dipendono direttamente dal segnale audio. Difatti, questi embedding devono essere adatti anche a stabilire se il concetto che rappresentano sia attivo o meno nell'input analizzato - processo incoraggiato dall'utilizzo della concept loss. Al contrario, gli embedding dei concetti non predicibili contengono esclusivamente l'informazione relativa allo stato corrispondente siccome il fattore predittivo viene meno - lo stato reale è conosciuto dal modello. Si è preferito adottare la tecnica degli embedding automatici per i concetti ground truth rispetto ad altre - come concatenare i concetti forniti dal LLM in forma binaria agli embedding dei concetti predetti - siccome rappresentava una soluzione sperimentale meno approfondita dalla letteratura.

Capitolo 4

Esperimenti

In questa sezione della tesi si descrivono le varie tipologie di esperimenti svolti. In particolare, ogni esperimento verrà analizzato nel dettaglio nella sottosezione ad esso dedicata, comprendente una descrizione dei modelli utilizzati, del setup e dei risultati ottenuti.

4.1 Setup degli esperimenti

Per quanto riguarda gli esperimenti di annotazione dei concetti, sono stati considerati i modelli Mistral 7B-v0.1-hf e Gemini-pro generative. Per testarli, si è utilizzato il test set composto da 69 casi annotati manualmente. Gli esperimenti hanno coinvolto l'annotazione di tutti i concetti appartenenti al candidate set originale, il quale comprende i concetti predicibili, i concetti corrispondenti a sintomi del paziente e ad informazioni generali ed infine i concetti non predicibili tramite esame percettivo della voce. In questo modo, si è potuta testare l'efficienza e la comprensione del testo dei LLM su una più ampia gamma di concetti, assicurandoci che questo metodo potesse essere eventualmente ampliato a più concetti in caso di necessità. Siccome l'output desiderato richiedeva una struttura coerente, la temperatura[1] è stata settata a 0.1.

Per svolgere gli esperimenti di voice disorder classification, si sono utilizzati gli audio e i referti presenti all'interno del IPV dataset. Siccome sia gli audio delle vocali sostenute che quelli delle frasi CAPE-V presentano durate molto variabili, e siccome la dimensione dei campioni dev'essere costante, per tutti gli esperimenti si è

[1]La temperatura è un parametro che controlla il grado di casualità nella generazione del testo da parte di un LLM. Valori più alti di temepratura (*e.g.* 0.7-0.8) favoriscono risposte più varie e meno prevedibili. Un valore basso di temperatura, invece, conferisce maggiore determinismo al modello.

deciso di fare *padding* sui sample. Come misura di riferimento per il padding, è stata selezionata la durata massima tra tutti i sample. Le reti transformer possiedono la capacità di *ignorare* il padding utilizzando una *attention mask*, ovvero una maschera che riconosce le aree del segnale prive di informazioni rilevanti grazie al meccanismo dell'attenzione. L'attention mask corrisponde ad un tensore della stessa grandezza del sample paddato, in cui ogni posizione è valorizzata ad "1" se rappresenta un frame che contiene informazione, o "0" se rappresenta un frame di padding. Tutti i frame che corrispondono al valore "0" vengono ignorati.

Essendo pre-addestrato su esempi campionati a 16kHz, HuBERT garantisce rappresentazioni coerenti solo per audio campionati alla stessa frequenza. Gli audio presenti nel IPV dataset sono stati campionati a 44,1kHz. E' stato necessario, quindi, effettuare un resample su ognuno di questi segnali audio, in modo da poterli fornire in input al feature extractor senza riscontrare problemi.

Per quanto riguarda l'input fornito al classificatore finale o al classificatore di concetti - a seconda del modello utilizzato - è necessario che sia un tensore monodimensionale. HuBERT estrae un rappresentazione per ogni frame del segnale, ovvero un intervallo comprendente più campioni. Attraverso l'encoder CNN, in pratica, HuBERT applica un downsampling sul segnale. In output, quindi, si ha un numero di rappresentazioni che dipende dalla durata dell'audio. Per trasformare questo insieme di rappresentazioni in una rappresentazione unica, si è applicato il *max pooling*. Questa tecnica di riduzione della dimensionalità, in questo caso applicata alla dimensione temporale, consiste nel prendere esclusivamente i valori maggiori lungo una certa dimensione. Si è preferito il max pooling rispetto ad altre tecniche, come ad esempio l'*average pooling*, dal momento che le caratteristiche di una voce patologica tendono a manifestarsi in picchi, piuttosto che in modo costante.

Per il train/val/test split si è optato per 64/18/18. Data la ridotta dimensione del dataset, si è scelto questo split rispetto ai più comuni 70/15/15 o 80/10/10 per garantire affidabilità ai risultati ottenuti sul validation set e sul test set. Per mantenere una distribuzione delle classi costanti in tutti e tre i set, si è utilizzato lo *stratified splitting*. Siccome le classi sono fortemente sbilanciate, uno squilibrio tra la distribuzione del train set e del test set avrebbe influito pesantemente sulle prestazioni del modello ed avrebbe reso i risultati ottenuto meno affidabili.

4.2 Rete neurale convenzionale

Considerando che i referti a disposizione erano solamente 312 su 513, per avere un confronto basato sullo stesso dataset, la rete neurale convenzionale è stata addestrata esclusivamente sui casi che disponevano di un referto scritto. Come precedentemente descritto, la struttura della rete prevede un feature extractor

composto da un modello HuBERT pre-addestrato e da una testa di classificazione, composta da un numero variabile di layer lineari. Negli esperimenti, sono stati utilizzati rispettivamente 2,3,4 layer lineari sequenziali. Per quanto riguarda le dimensioni delle hidden unit intermedie, si è scelta [128] per l'architettura 2-layer, [256,128] per quella 3-layer, e infine [512,256,128] per l'architettura 4-layer. Considerando che la dimensione degli embedding che HuBERT fornisce in output è 768, si sono scelti questi valori per una progressiva riduzione della dimensionalità. Un numero maggiore di layer avrebbe potuto portare ad una over-parametrizzazione del modello, introducendo un numero di parametri decisamente maggiore rispetto a quelli necessari al modello per ottenere rappresentazioni intermedie efficaci. La *loss function* adottata per questo esperimento è una cross-entropy loss. Considerando il grande sbilanciamento tra le due classi - 70 sample sani contro 242 malati - è stato necessario utilizzare dei pesi all'interno della funzione di loss. Per il feature extractor, siccome si tratta di una rete pre-addestrata molto complessa, si è scelto di utilizzare un learning rate di 5e-5, mentre per il classificatore finale si è ricorso ad un learning rate di 5e-4. Non essendo particolarmente complesso e avendo un numero di parametri realtivamente basso, quest'ultima sezione del modello necessita di un learning rate maggiore rispetto ad HuBERT.

Nella tabella 4.1 sono riportate le prestazioni delle tre diverse configurazioni. Per il calcolo dell f1 score, si è utilizzato la macro average, ovvero la media aritmetica tra gli f1 score delle due classi. In questo modo, le classi meno rappresentate influiscono ugualmente sul calcolo del totale. Gli f1 score racchiusi dalle parentesi appartengono rispettivamente alla classe *sano* e alla classe *malato*.

Classifier	Val Accuracy	Test Accuracy	Test F1 score (macro avg)
2-LAYER config	80.36%	82.46%	**0.76** (0.64-0.88)
3-LAYER config	86.31%	**84.80%**	**0.76** (0.62-0.90)
4-LAYER config	**86.91%**	81.87%	0.70 (0.50-0.89)

Tabella 4.1: I risultati dei diversi classificatori messi a confronto. Tra parentesi sono indicati gli F1 score delle due classi (sano-malato).

Come era prevedibile, è possibile notare come, per ciascuna configurazione, ci sia una grande differenza tra gli f1 score delle due classi. I modelli, infatti, faticano maggiormente a riconoscere correttamente la classe *sano*, essendo decisamente meno rappresentata. L'architettura 3-layer è quella che in generale performa meglio sulle tre diverse metriche. Per questo motivo, è stata scelta come architettura di confronto per i risultati dei modelli concept-based.

Per dimostrare come le registrazioni delle frasi CAPE-V contengano maggiore informazione rispetto alle registrazioni delle vocali sostenute, ho ripetuto l'esperimento utilizzando queste ultime come input per il modello. I risultati sono mostrati

Input	Val Accuracy	Test Accuracy	Test F1 score (macro avg)
Frasi CAPE-V	**86.31%**	**84.80%**	**0.76** (0.62-0.90)
Vocali sostenute	78.28%	81.31%	0.70 (0.50-0.89)

Tabella 4.2: I risultati ottenuti utilizzando le due diverse tipologie di registrazioni come input.

nella Tabella 4.2. Il modello addestrato sulle vocali sostenute raggiunge un'accuracy e un F1 score significativamente inferiori rispetto alla sua controparte addestrata su frasi CAPE-V, confermando l'ipotesi iniziale. Per questo motivo, gli esperimenti successivi sono stati eseguiti utilizzando esclusivamente queste ultime come input.

Siccome il dataset a disposizione per il training è composto da un numero ridotto di sample, il modello potrebbe risentire di overfitting e di focalizzarsi, quindi, su caratteristiche non rilevanti del training set - come ad esempio rumore. Per questo motivo, si è deciso di applicare *data augmentation* (DA) sui sample. Con l'obbiettivo di stabilizzare il processo di training e osservare se ci fossero miglioramenti nelle prestazioni, sono state impiegate trasformazioni come il *time stretch*, *pitch shift*, e l'aggiunta di rumore. Queste trasformazioni sono state applicate su ciascun sample: il valore dei parametri di ciascuna traformazione (*e.g.* p% di pitch shift) sono stati selezionati in modo casuale, in un certo range, utilizzando una distribuzione gaussiana di probabilità. Per osservare l'impatto della data augmentation, è stato introdotto un iper-parametro definito *intensità*, il quale rappresenta il fattore moltiplicativo del massimo valore del range. Per esempio, se di base il range di time stretch è [0.00;0.10], con *intensità* = 2.0 il range diventa [0.00;0.20]. La distribuzione di probabilità rimane invariata, a cambiare è esclusivamente l'intervallo tra valore massimo e minimo. Siccome trasformazioni come time stretch e pitch shift richiedono un tempo di computazione elevato, si sono applicate a sample troncati a 10 secondi di audio. Il modello è stato addestrato anche sulla versione troncata dell'input, ma senza l'applicazione delle trasformazioni di data augmenation, al fine di poter comparare i risultati su dati equivalenti.

I risultati mostrati nella Figura 4.1 riassumono l'andamento della data augmentation a vari livelli di intensità. Le trasformazioni applicate ai dati non sembrano migliorare le prestazioni del modello addestrato sui dati originali, anzi, le performance sono inferiori su ogni metrica considerata. Una causa di questi risultati potrebbe essere la grande sensibilità del modello alla qualità vocale: alterazioni come il pitch shift e il rumore, sommati a disturbi già presenti nelle registrazioni audio, possono influenzare negativamente la capacità del modello di valutare la qualità vocale dal segnale audio. Il modello appare comunque stabile, con valori costanti di accuratezza, dimostrando di non overfittare sul dataset di training. Una data augmentation non eccessivamente aggressiva - ovvero con *intensità* = 1.0

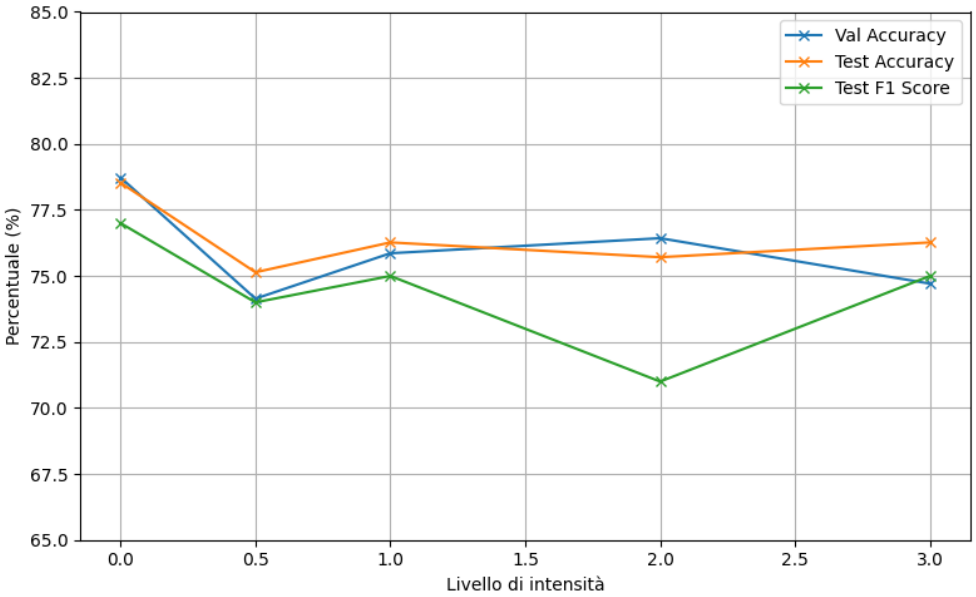

Figura 4.1: Validation accuracy, test accuracy e f1 score per le diverse intesità di data augmentation. In ognuna delle metriche considerate, la DA non migliora le prestazioni.

- risulta essere la più efficiente. Osservando la Figura 4.2, possiamo comparare l'andamento della train/validation accuracy e della train/validation loss del modello addestrato senza DA e con DA con *intensità* = 1.0. Applicando la data augmentation, il processo di addestramento risulta più stabile. Il modello, come ci si poteva aspettare, tende a convergere più lentamente, raggiungendo, nel caso mostrato nella figura, un risultato ottimale con *epoch* = 14, mentre senza data augmentation il risultato ottimale è stato raggiunto già con *epoch* = 5. Questo comportamento è perfettamente giustificato dall'incremento della varianza dei dati di input introdotto dalla data agumentation, il quale rallenta la convergenza. Considerando la maggiore complessità e l'incremento del tempo di addestramento introdotti dalla DA, si è preferito non applicare questa tecnica e, pertanto, di procedere negli esperimenti sui dati non trasformati.

4.3 Annotazione dei concetti

Le metriche considerate per la valutazione dei due modelli sono l'accuracy media dei concetti, il macro F1 score medio dei concetti e gli errori totali. Quest'ultima misura rappresenta la somma degli errori commessi dal modello per ciascun concetto. L'insieme di queste metriche permette di avere una panoramica sia globale che

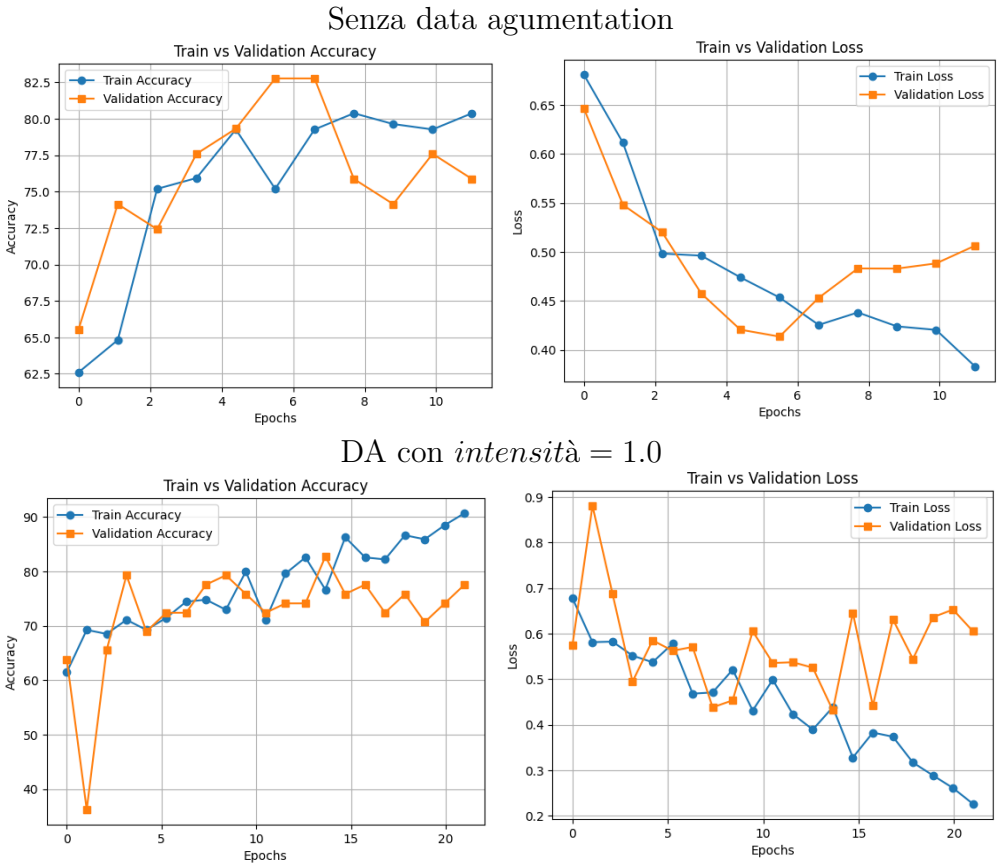

Figura 4.2: I grafici della train/validation accuracy e train/validation loss relativi all'addestramento del modello senza data augmentation (in alto) e con data augmentation ad *intensità* = 1.0 (in basso).

locale delle prestazioni dei modelli, considerando anche fattori come la distribuzione dei concetti nel test set.

Come mostrato nella tabella 4.3, il modello Gemini performa leggermente meglio rispetto a Mistral a livello di accuracy e di f1 score. Analizzando più nel dettaglio i risultati ottenuti, si può però notare come gli errori commessi da Gemini siano meno della metà. Per i successivi esperimenti, si è scelto quindi di utilizzare i concetti estratti da Gemini.

Model	Concept Acc (avg)	Concept F1 score (avg)	Total Errors
Mistral 7B-v0.1-hf	97.1%	0.97	42
Gemini-pro	**98.7%**	**0.98**	**20**

Tabella 4.3: Risultati degli esperimenti sui due LLMs.

Concept	Accuracy	F1 score	Errors
Fumo	97.1	0.95	2
Uso voce professionale	97.1	0.97	2
Diplofonia	100.0	1.00	0
Strain	100.0	1.00	0
Roughness	100.0	1.00	0
Breathiness	100.0	1.00	0
Asthenicity	100.0	1.00	0
Fonastenia	100.0	1.00	0
Atteggiamento gottico a clessidra	100.0	1.00	0
Disodia	97.1	0.94	2
Sesso	100.0	1.00	0
No disfonia	94.2	0.91	4
Disfonia lieve	92.8	0.93	5
Disfonia lieve-moderata	98.6	0.96	1
Disfonia moderata	100.0	1.00	0
Disfonia severa	100.0	1.00	0
Onda mucosa irregolare	94.2	0.94	4
Mucose rosee	100.0	1.00	0
Mucose iperemiche	100.0	1.00	0
Mucose eutrofiche	100.0	1.00	0

Tabella 4.4: Risultati del modello Gemini pro su ciascun concetto

Nella Tabella 4.4 possiamo osservare più nello specifico i risultati ottenuti sui singoli concetti dal modello Gemini. E' necessario considerare che, non essendo stati scritti al fine di estrarne concetti, i referti possono contenere descrizioni non perfettamente chiare di alcune caratteristiche. E' il caso, ad esempio, del grado di disfonia, che può non essere indicato esplicitamente, ma piuttosto in modo generico. L'annotazione è quindi frutto di un'interpretazione dei dati, la quale può generare incongruenze tra annotazione automatica e manuale. Si prestano particolarmente ad errori di questo genere concetti come *fumo* o *uso professionale della voce*. Nel caso di un ex fumatore, il modello potrebbe avere un dubbio su come etichettarlo. Lo stesso ragionamento può essere fatto nel caso in cui un paziente studiasse canto: la voce non è utilizzata in modo professionale in senso stretto, tuttavia ne viene fatto un uso intensivo.

4.4 Predizione classe con concetti reali (c—>y)

Per garantire che i concetti individuati nel concept set finale fossero effettivamente discriminativi per le due classi - ovvero se si riuscisse ad ottenere una classificazione soddisfacente basandosi esclusivamente sui concetti - è stato addestrato un classificatore *sani/malati* utilizzando i concetti ground truth. Il modello, composto da due layer lineari di dimensione [256,128], riceve come input i concetti estratti dal Large Language Model (Gemini) e, basandosi su di essi, predice la classe corrispondente. Questa rete simula la capacità di predizione di un modello concept-based ideale che predice i concetti con una precisione del 100%. Per questo motivo, ho deciso di definirla *IdealCBM*.

Model	Test Class Accuracy	Test Class F1 score
HuBERT	84.80%	0.76 (0.62-0.90)
IdealCBM	**88.30%**	**0.80** (0.67-0.93)

Tabella 4.5: Risultati degli esperimenti sulla rete convenzionale HuBERT e sull'IdealCBM.

Come mostrato nella Tabella 4.5, l'IdealCBM supera le prestazioni di HuBERT sia in termini di class accuracy che di class F1 score. Questi risultati dimostrano che i concetti individuati nel candidate set finale permettono, nella maggior parte dei casi, di distinguere correttamente le due classi.

4.5 Predizione dei concetti (x—>c)

Dopo averne appurato l'efficacia, è stato necessario verificare se i concetti individuati potessero essere appresi correttamente da una rete neurale. Per raggiungere questo scopo, è stato addestrato un classificatore di concetti. La task di questo modello, riprendendo lo schema citato nella sottosezione 3.2.3, può essere definita come x —> c. In questo caso, la task finale — la classificazione sano/malato — non viene considerata dal modello, il quale ottimizza una funzione di loss che riguarda esclusivamente i concetti. L'architettura della rete si divide in un feature extractor, composto da una rete HuBERT pre-addestrata, è da una testa di classificazione, composta da due layer lineari di dimensione [256,128]. Per HuBERT si è usato un learning rate pari a 5e-5, mentre per la testa di classificazione si è scelto un valore di 5e-4. La classificazione dei concetti è stata trattata come una task multi-label binaria: l'ultimo layer della testa di classificazione restituisce come output i logit di ciascun concetto predicibile; attraverso una sigmoide e l'applicazione di una soglia di attivazione - nel nostro caso, per ogni concetto è stata usata la soglia 0.5 - avviene

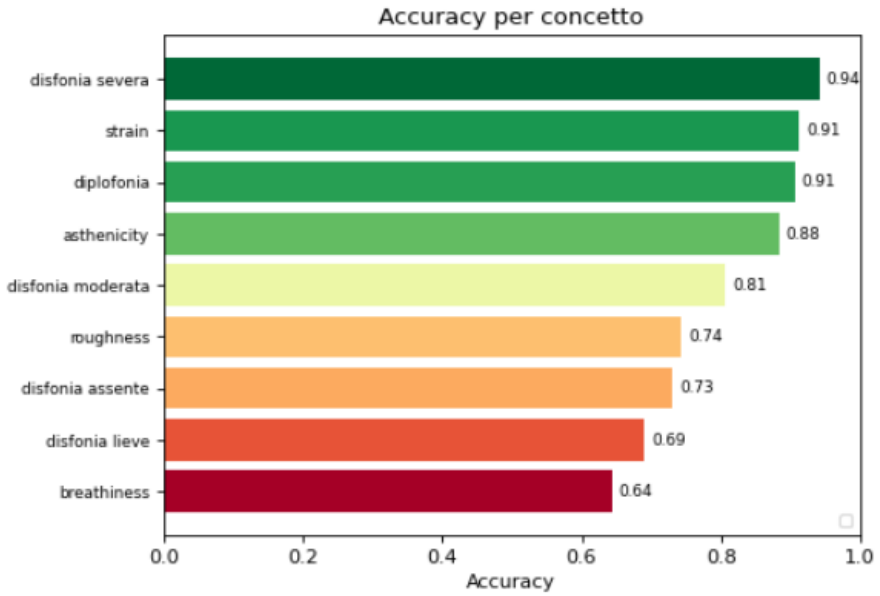

Figura 4.3: Test accuracy per ogni concetto predicibile. I risultati sono ordinati secondo un valore di accuracy decrescente.

la predizione binaria di ciascun concetto. La funzione di loss utilizzata è la Binary Cross-Entropy con logits. La loss viene calcolata su tutti i concetti predetti, che non vengono quindi ottimizzati in modo indipendente. E' stato scelto di adottare questa strategia per diverse ragioni. Questo approccio simula l'apprendimento del modello concept-based x—>c—>y, permettendo un confronto diretto tra le concept accuracy dei modelli. Inoltre, un apprendimento combinato dei concetti permette di ottenere rappresentazioni che meglio generalizzano su tutti i concetti, migliorando le performance complessive del modello. Infine, una loss combinata tiene conto di possibili dipendenze esistenti tra concetti.

Il modello ha raggiunto un'accuracy media sui concetti di 80.64%. Per maggiore dettaglio, nella Figura 4.3 possiamo osservare l'accuracy ottenuta sui singoli concetti. Come si può notare, i concetti vengono generalemente riconosciuti in modo corretto. In particolare, per quanto riguarda i concetti legati alla disfonia, *disfonia severa* e *disfonia moderata* raggiungono valori di accuracy maggiori rispetto a *disfonia assente* e *disfonia lieve*. È probabile che il modello incontri maggiori difficoltà nel distinguere tra una disfonia lieve ed una voce eufonica, mentre i gradi più marcati risultano più facilmente identificabili e sono associati a rappresentazioni concettualmente più distinte. Il concetto *breathiness* è quello con l'accuracy più bassa. Questo comportamento potrebbe essere causato dall'utilizzo del max pooling: la breathiness, solitamente, si manifesta in modo diffuso e, a meno di picchi particolarmente marcati, potrebbe non essere identificata.

Conseguentemente ai risultati raggiunti, il modello ha dimostrato di essere capace di apprendere i concetti selezionati per il candidate set finale. Il passo successivo è applicare la classificazione dei concetti all'interno dei concept-based models.

4.6 Modello CBM (x–>c–>y)

Nel modello CBM, come descritto precedentemente, si è adottata una strategia mista per quanto riguarda i concetti: 9 concetti sono predetti dal modello, mentre 5 concetti sono forniti al modello direttamente dalle annotazioni di Gemini. La struttura del modello è divisa in concept classifier - costituito da encoder HuBERT e testina di classificazione dei concetti - e dal classificatore finale, il quale riceve in input i 14 concetti e predice la classe. Il concept classifier è pressochè identico a quello utilizzato nella sezione precedente. E' stata utilizzata una Binary Cross-Entropy loss con logits per ottimizzare l'apprendimento dei concetti, con un learning rate di 5e-5. Per quanto riguarda il task classifier, invece, si è scelta una Cross-Entropy loss con i pesi corrispondenti alle due classi, e un learning rate di 5e-3. La grande differenza tra i due learning rate è dovuta al fatto che il concept classifier è composto da una rete pre-addestrata come HuBERT, la quale necessita esclusivamente di fine-tuning a learning rate basso, mentre il task classifier è addestrato *from scracth*. La tipologia di modello CBM impiegata negli esperimenti è il joint bottleneck, dove viene ottimizzata una loss totale, somma delle due loss. Si è deciso di assegnare ai parametri λ i valori $\lambda_c = 0.9$ e $\lambda_{sm} = 0.1$, favorendo maggiormente l'apprendimento dei concetti. Questa scelta, comune nei modelli concept-based, evita che il modello raggiunga ottime prestazioni sulla task finale sfruttando rappresentazioni indipendenti dai concetti, perdendo, così, l'alto livello di intepretabilità fornito da quest'ultimi.

Nella Tabella 4.6 si possono osservare i risultati ottenuti dal Concept Bottlenck Model. Per quanto riguarda la concept accuracy media, il CBM raggiunge il 78.71% contro l'80.64% raggiunto dal concept classifier x—>c, che rappresenta la massima concept accuracy raggiungibile da questa tipologia di modello. Nella rete CBM, infatti, l'ottimizzazione coinvolge due loss differenti: nonostante la concept loss abbia un peso maggiore della task loss, il modello non ottimizza esclusivamente la prima, risentendone lievemente in termini di concept accuracy. Nella Figura 4.4 sono comparate le accuracy dei singoli concetti, dove si può notare come, in media, i risultati siano leggermente inferiori rispetto al concept classifier puro. Come era possibile prevedere, la task accuracy di HuBERT rimane superiore a quella del CBM, che tuttavia raggiunge risultati comparabili con uno scarto di circa 3.75%. Questa differenza è presente anche per quanto riguarda l'F1 score, con un decremento del 0.07 nel caso del CBM. I risultati del Concept Bottleneck Model,

Model	Concept Acc	Task Acc	Task F1 score
HuBERT	-	84.80%	0.76
CBM	78.71%	81.05%	0.69
Concept Classifier	80.64%	-	-
IdealCBM	-	88.30%	0.80

Tabella 4.6: Comparazione dei risultati ottenuti con HuBERT e CBM. Sono indicati anche i risultati del Concept Classifier e dell'IdealCBM, utili per il confronto dei primi due con prestazioni ideali.

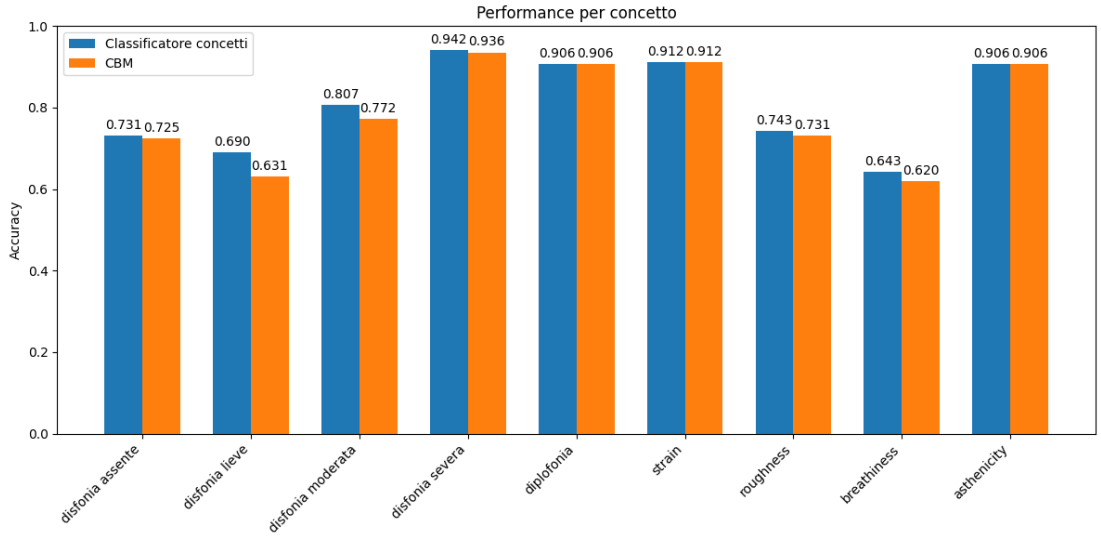

Figura 4.4: Le accuracy ottenuto per ciascun concetto con il modello x—>c (concept classifier) e con il CBM.

nonostante questa rete raggiunga una concept accuracy di poco inferiore all'80%, dimostrano la possibilità di avvicinare le performance di un modello end-to-end assicurando un'alta interpretabilità.

4.7 Modello CEM

Anche per il modello CEM è stato utilizzato un approccio misto. Gli embedding dei 9 concetti predicibili e dei 5 embedding relativi ai concetti forniti dal LLM vengono

trasformati automaticamente in embedding[2], i quali vengono successivamente aggiornati durante il training tramite backpropagation. La struttura del modello è pressochè identica a quella del CBM. Il concept classifier è composto dall'encoder HuBERT e da una testa di classificazione, in questo caso composta da un solo layer, il concept embedding layer. Questo layer fornisce sia gli embedding di input per il classificatore finale, che le predizioni binarie dei concetti. Il classificatore finale è composto da 3 layer lineari, con hidden unit rispettivamente di dimensioni [256,128]. Il learning rate utilizzato per il concept classifier è stato settato a 5e-5, mentre per la testa di classificazione si è scelto 5e-4. Si è optato per questi valori per le stesse motivazioni riportate nella sezione precedente. Anche in questo caso, come pesi delle due loss sono stati scelti i valori $\lambda_c = 0.9$ e $\lambda_{sm} = 0.1$. Per l'addestramento, si è adottata una strategia nota come *warmup*: per le prime 2 epoch di training, la task è stata congelata ponendo $\lambda_{sm} = 0$, per essere poi riattivata dalla terza epoch in poi. In questo modo, la task loss ha iniziato ad essere attiva solo dopo che il modello ha appreso rappresentazioni migliori per i concetti, evitando che imparasse una classificazione basata su rappresentazioni errate nelle prime epoche. Come possiamo osservare nella Tabella 4.7, il Concept Embedding Model supera le

Model	Concept Acc	Task Acc	Task F1 score
HuBERT	-	84.80%	0.76
CBM	78.71%	81.05%	0.69
CEM	79.38%	82.46%	0.71

Tabella 4.7: Comparazione dei risultati ottenuti con HuBERT, CBM e CEM.

prestazioni del modello CBM sia in termini di concept accuracy - 79.38% contro 78.71% - che di task accuracy - 82.46% contro 81.05% - avvicinandosi ulteriormente ai risultati ottenuti con il modello end-to-end HuBERT. Anche l'F1 score migliora, passando da 0.69 a 0.71. Complessivamente, il modello CEM offre prestazioni migliori rispetto al CBM, a scapito, tuttavia, di un certo grado di interpretabilità.

4.8 Migliori esperimenti su dataset esteso

Come già anticipato nella sezione dedicata alla descrizione del dataset (Sezione 3.1), il personale medico che ha seguito lo sviluppo di questo progetto ha, successivamente, fornito ulteriori dati utili per gli esperimenti. In particolare, i nuovi dati consistevano

[2]La creazione di embedding automatici è stata eseguita mediante l'utilizzo del modulo torch.nn.Embedding.

in alcuni dei referti mancanti di casi già presenti nel dataset originale - e tuttavia non utilizzati per l'addestramento dei modelli. Siccome l'estensione del dataset è avvenuta ad esperimenti pressoché conclusi, si è reputato opportuno ripetere esclusivamente gli esperimenti più significativi sul dataset ampliato. E' doveroso citare che il dataset, oltre ad essere stato esteso, è stato oggetto di una mia revisione manuale. Questa revisione si è resa necessaria dopo aver notato che alcuni report clinici, nonostante presentassero i risultati di varie tipologie di esami, non contenevano una descrizione dell'esame percettivo della voce, causando una valorizzazione a "0" di tutti i concetti siccome non venivano citati nel documento. Specialmente per i casi patologici, questo poteva influire pesantemente sia sul training che sull'evaluation dei concetti. Questi casi sono stati considerati come carenti di report clinico - sarebbe meglio dire di esame percettivo - e quindi non considerati negli esperimenti. Nella Tabella 4.8 si possono osservare i risultati

End-to-End HuBERT			
Dataset	**Concept Accuracy**	**Class Accuracy**	**F1 Score**
Originale	-	84.80%	0.76
Esteso	-	91.33%	0.91
Classificazione dei concetti			
Dataset	Concept Accuracy	Class Accuracy	F1 Score
Originale	80.64%	-	-
Esteso	85.80%	-	-
Concept Bottleneck Model (CBM)			
Dataset	Concept Accuracy	Class Accuracy	F1 Score
Originale	78.71%	81.05%	0.69
Esteso	84.43%	87.76%	0.86
Concept Embedding Model (CEM)			
Dataset	Concept Accuracy	Class Accuracy	F1 Score
Originale	79.38%	82.46%	0.71
Esteso	84.50%	87.30%	0.86

Tabella 4.8: Confronto delle performance sulle quattro task precedentemente analizzate sul dataset originale e sul dataset dopo l'estensione.

ottenuti utilizzando il dataset originale e il dataset esteso, considerando tutte le task valutate nelle sezioni precedenti. Considerato che nel dataset esteso: erano

disponibili più esempi per il training; le classi erano più bilanciate; le annotazioni sono state parzialmente riprocessate con supervisione umana; i risultati mostrano un netto miglioramento sulla totalità delle task. In particolare, si può notare un incremento dell'F1 score di ~0.15 nelle performance di ogni modello, decisamente maggiore rispetto all'incremento della class accuracy. Questo fenomeno è dovuto principalmente al bilanciamento delle classi introdotto dai nuovi dati del dataset esteso, quasi interamente appartenenti alla classe *sano* (0), decisamente meno rappresentata. Nei modelli concept-based, la crescita della concept accuracy (+5.72% per il CBM) è parallela all'incremento della class accuracy (+6.71% per il CBM). Questo netto miglioramento nel riconocimento delle classi può essere attribuito a due fattori: apprendimento di pattern di classificazione *sano/malato* più efficaci e riconoscimento più preciso di concetti cruciali per la classificazione finale. Per quanto riguarda il primo fattore, la disponibilità di classi più bilanciate ha sicuramente influito sull'apprendimento di pattern più differenziati. Il secondo fattore sembra essere confermato se analizziamo più nel dettaglio le concept accuracy singole dei concetti. Le accuracy cresciute maggiormente rispetto agli esperimenti sul dataset originale sono quelle dei concetti *disfonia assente* (+12.41%) e *disfonia lieve* (+10.71%). Il grado di disfonia rappresenta un indicatore essenziale della qualità della voce. In particolare, il concetto *disfonia assente* è spesso associato alla classe *sano*. Un incremento così significativo nel riconoscimento di questi due concetti giustifica e spiega il corrispondente miglioramento nelle predizioni della classe. Limitiamo ora la nostra attenzione al confronto dei risultati ottenuti sul

Model	Concept Acc	Task Acc	Task F1 score
HuBERT	-	91.33%	0.91
CBM	84.43%	87.76%	0.86
CEM	84.50%	87.30%	0.86

Tabella 4.9: Comparazione dei risultati ottenuti con HuBERT, CBM e CEM sul dataset esteso.

dataset esteso. Come possiamo osservare dalla Tabella 4.9, il modello end-to-end mantiene il primato nelle prestazioni sia sulla class accuracy che sull'F1 score. Per quanto riguarda i modelli concept-based, invece, il CBM supera, seppur con un margine ristretto, la sua controparte basata sugli embedding nella class accuracy. La concept accuracy è pressoché identica. Generalmente, i modelli CEM superano, in class accuracy, quelli basati sul bottleneck layer. La limitata rappresentabilità imposta dall'associazione 1:1 tra nodi e concetti spesso non permette di cogliere particolari *sfumature* di quest'ultimi. Nel nostro caso, i concetti sono ben definiti e difficilmente scomponibili in sotto-concetti (procedimento in parte già compiuto

con la divisione della disfonia in gradi), e risultano particolarmente adatti alla classificazione della voce[3]. Per questo motivo, probabilmente, il modello CBM e il modello CEM raggiungono la stessa class accuracy. I risultati ottenuti dai due modelli dimostrano, anche dopo l'estensione del dataset originale, la possibilità di raggiungere prestazioni competitive con i modelli end-to-end, pur mantenendo un elevato grado di interpretabilità.

Intervention. Il concetto di *intervention*, introdotto nella sezione 2.3, rappresenta una funzionalità cruciale dei modelli Concept Bottleneck Model. Essa consiste nel sostituire, a test time, i valori di determinati concetti predetti dal modello con il loro valore reale. Osservando come cambia l'output del classificatore in base all'input (i concetti), è possibile individuare relazioni causali tra concetti e classe, evidenziando quali concetti siano più rilevanti. Il processo si adatta particolarmente ad uno scenario come la diagnosi dei disturbi della voce: una supervisione umana specializzata può correggere eventuali errori riconosciuti nella predizione dei concetti, e correggere, conseguentemente, una predizione finale causata dall'errore in input. Una combinazione di intelligenza artificiale e personale specializzato potrebbe superare le prestazioni dei singoli.

Per osservare l'influenza di ciascun concetto sulla predizione finale, in un primo momento, si è applicata l'intervention sui concetti singolarmente. Nella fase successiva, si è applicata l'intervention su un numero di concetti gradualmente crescente, fino a raggiungere l'interezza del candidate set. L'intervention risulta particolarmente efficace quando la concept accuracy raggiunge valori elevati ($>$ 90%). Nel nostro caso, la concept accuracy è del 84.43%. E' probabile, pertanto, che l'impatto dell'intervention non sia del tutto positivo sul classificatore finale. Il modello, infatti, ha imparato a distinguere le classi utilizzando la distribuzione dei concetti fornitagli in input dal concept classifier. Un cambiamento, seppur leggero, di questa distribuzione potrebbe influire negativamente sul suo processo decisionale. Nella Figura 4.5 possiamo osservare i risultati ottenuti con l'intervention sui singoli concetti. E' possibile notare come l'unico concetto su cui l'intervention si rivela effettivamente utile è *disfonia assente*. Intuitivamente, è facile comprendere la forte correlazione tra questo concetto e la classe *sano*. Il modello ha imparato correttamente, nonostante gli errori di predizione, ad associare l'assenza di disfonia con una voce non patologica, relazione che persiste nella maggior parte dei casi. Si può aggiungere che il concetto *disfonia assente* presenta una delle concept accuracy più basse: avendo imparato la corretta associazione e avendo a disposizione le label corrette, il modello riesce a classificare con precisione maggiore i segnali audio in input. Per quanto riguarda i concetti rimanenti, la concept accuracy

[3]Ricordiamo che i concetti selezionati appartengono ad una scala utilizzata in campo medico (GRBAS) di cui è stata provata l'efficacia.

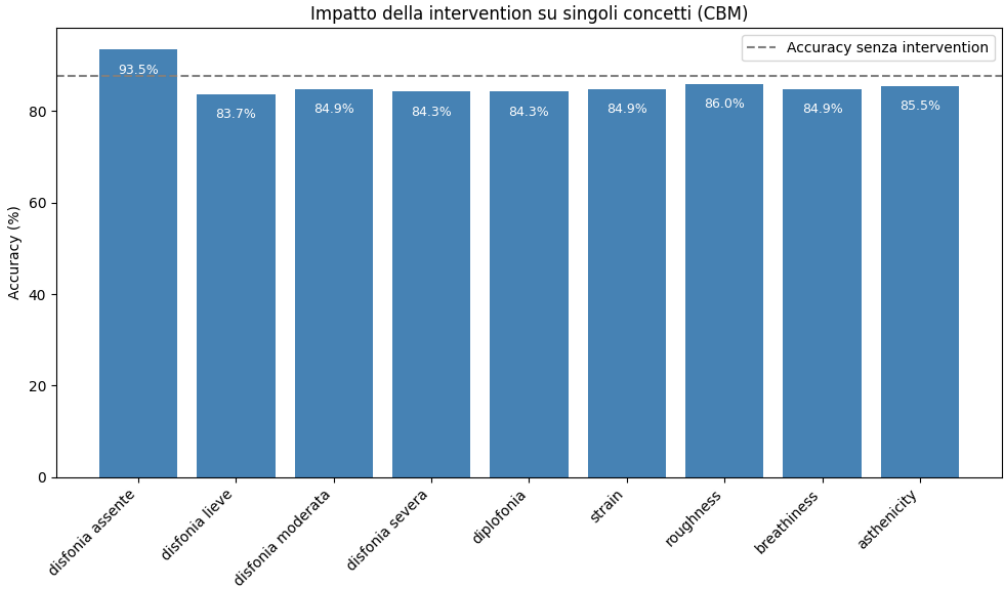

Figura 4.5: Accuracy raggiunte con l'intervention sui singoli concetti. In grigio è indicata l'accuracy del modello senza intervention.

raggiunge valori elevati durante l'addestramento. La correzione non porta, perciò, a differenze significative. E' possibile evidenziare, tuttavia, come nessuno dei concetti (eccetto, in parte, *disfonia assente*) sia sufficiente, da solo, per una classificazione corretta. Dopo aver osservato l'applicazione dell'intervention sui singoli concetti, vediamo adesso come si comporta il modello quando la correzione avviene su più concetti contemporaneamente. Nella Figura 4.6 è rappresentata la task accuracy raggiunta dal Concept Bottleneck Model tramite l'applicazione dell'intervention su un numero gradualmente maggiore di concetti. Alla prima iterazione è stato corretto un singolo concetto, alla seconda iterazione i primi due concetti e così via fino a una intervention completa. L'ordine dei concetti seguito è quello rappresentato nella Figura 4.5. L'accuracy tende a crescere con un intervention sui primi 2 concetti, mentre l'ulteriore estensione al resto dei concetti provoca una lieve diminuzione delle prestazioni. Questo risultato dimostra nuovamente l'importanza dei concetti legati al grado di disfonia nella determinazione della classe. Concetti come *strain* o *breathiness*, viceversa, presentano relazioni più complesse e articolate. Il modello, addestrato come un joint bottleneck, ha imparato a riconoscere queste relazioni dalla distribuzione dei concetti predetta dal concept classifier. Cambiando questa distribuzione, il modello fatica a riconoscere le stesse relazioni nei concetti corretti, trovandosi di fronte configurazioni che potenzialmente non apparivano durante la fase di training.

Gli esperimenti di intervention evidenziano come la classificazione tra *sano* e

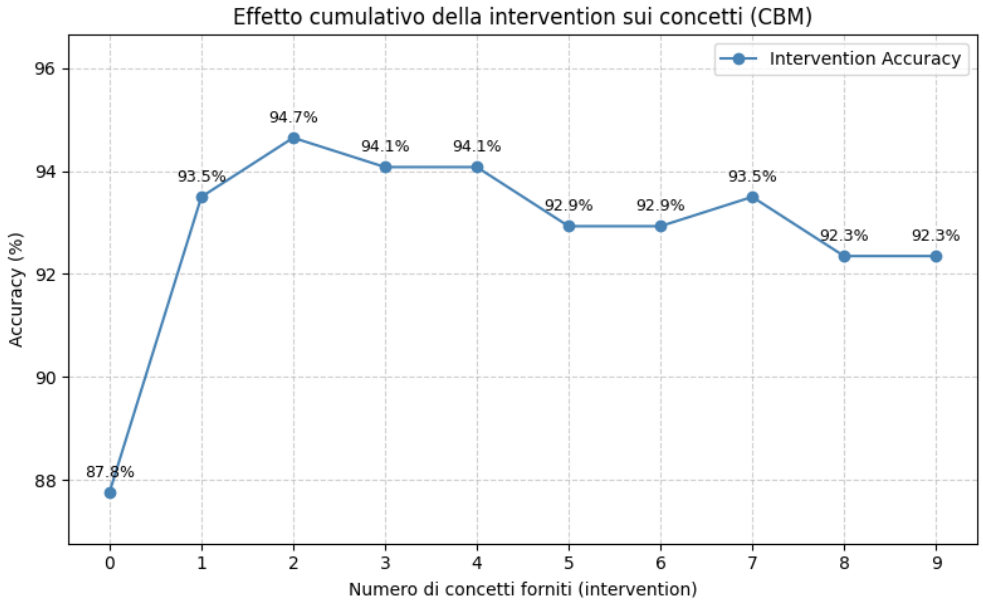

Figura 4.6: Effetti di un intervention progressiva sulla task accuracy raggiunta dal modello CBM.

malato sia guidata principalmente dai concetti relativi al grado di disfonia. In particolare, si osserva una forte correlazione tra l'assenza di disfonia e la classe *sano*, così come tra i gradi più elevati di disfonia e la classe *malato*. Il concetto di *disfonia lieve* si configura invece come un caso limite: un livello lieve può infatti essere associato a entrambe le classi. In queste situazioni ambigue, i concetti ausiliari — ovvero quelli non direttamente legati alla disfonia — assumono un ruolo più rilevante, contribuendo alla determinazione della classe in modo complementare.

Capitolo 5

Conclusione

In questa tesi si è esplorato un approccio concept-based alla task della voice disorder detection, con l'obbiettivo di raggiungere un trade-off soddisfacente tra performance e interpretabilità. Il lavoro si è concentrato su due tipologie di modelli concept-based, i Concept Bottleneck Model (CBM) e i Concept Embedding Model (CEM), scelti per l'alto livello di trasparenza dell'architettura, per la somiglianza del processo decisionale con una diagnosi umana e per la possibilità - tramite intervention - di integrare la supervisione da parte di personale medico specializzato durante l'inferenza. Siccome per l'addestramento dei due modelli era necessario un dataset provvisto di annotazioni concettuali, si è proceduto con un'annotazione automatica tramite LLM (Gemini) dei concetti, utilizzando come fonte i referti clinici contenuti nell'Italian Pathological Voice (IPV) dataset. I concetti da estrarre sono stati identificati tramite la consulenza di personale medico specializzato. Per poter confrontare i risultati dei modelli concept-based con le performance di un approccio standard, si è addestrata una rete neurale convenzionale basata su HuBERT, modello pre-addestrato specializzato nella speech analysis. Gli esperimenti hanno dimostrato che un approccio basato sui concetti permette di raggiungere prestazioni comparabili a una rete convenzionale, garantendo, però, un alto grado di interpretabilità e offrendo la possibilità di giustificare le decisioni prese attraverso la presenza o l'assenza dei concetti identificati.

Nonostante i risultati ottenuti, il lavoro svolto presenta alcune criticità. Innanzitutto, una prima limitazione significativa riguarda il dataset utilizzato, che non è stato concepito originariamente per un approccio concept-based. I referti clinici non sono file strutturati e contengono una valorizzazione dei concetti spesso ambigua e non sempre coerente. L'impiego di un Large Language Model ha introdotto, seppur in misura ridotta, un certo grado di errore nell'annotazione automatica dei concetti. Questo problema è stato parzialmente arginato attraverso una revisione manuale delle annotazioni, senza, però, raggiungere una risoluzione completa e definitiva. Un altro fattore da considerare è che il lavoro è stato interamente compiuto operando

su piattaforme con risorse computazionali limitate. Per questo motivo, esperimenti con batch size elevate non sono stati possibili, perdendo la possibilità di valutare i modelli al massimo della loro capacità. Inoltre, i modelli raggiungono accuracy sui concetti ancora distanti da quelle ottimali per modelli concept-based, a causa di una combinazione delle ragioni sopra citate. Questo aspetto, come evidenziato dai risultati, ha compromesso l'efficacia dell'intervention, uno strumento cruciale per la collaborazione tra medico e macchina.

Sviluppi futuri. La progettazione di un dataset strutturato e ottimizzato per la classificazione dei concetti permetterebbe un sostanziale miglioramento delle prestazioni e, di conseguenza, maggiore affidabilità. La scala GRBAS, utilizzata come spunto per la scelta dei concetti candidati, presenta un'ulteriore divisione di concetti come *strain, breathiness, ecc.* in gradi di intensità. Questa sottodivisione non è stata applicabile all'annotazione concettuale automatica siccome non citata esplicitamente nei referti clinici a disposizione. La realizzazione del dataset, tuttavia, richiede la disponibilità di personale medico specializzato per l'annotazione, rendendo questa direzioni di sviluppo difficile da intraprendere. Per valutarne le prestazioni su una più ampia gamma di dati, si potrebbero eseguire test dei modelli su dataset differenti dall'IPV, come per esempio l'Advanced Voice Function Assessment Database (AVFAD) [56], dataset molto utilizzato nella voice disorder analysis. L'annotazione dei concetti non è strettamente necessaria a test time per valutare le performance sulla task finale, rendendo adottabili i modelli concept-based anche su dataset che dispongono esclusivamente di segnali audio - come l'AVFAD. Un interessante direzione in cui estendere questo lavoro potrebbe essere l'implementazione di un'interfaccia per spiegare il processo decisionale. Attraverso un modello generativo - come un LLM per esempio, si potrebbe produrre un report simile a una diagnosi medica, contenente una descrizione dell'analisi della voce, linee guida da seguire, e altre informazioni clinicamente rilevanti. Per quanto riguarda la trasparenza del modello, si potrebbero applicare tecniche di eXplainable-AI post-hoc al fine di evidenziare relazioni causali tra le predizioni e i concetti (*e.g.* CaCE), espandendo il lavoro nella direzione intrapresa con gli esperimenti sull'intervention. Raggiungendo una accuracy e una granularità dei concetti maggiore, si potrebbe estendere l'approccio concept-based alla task della voice disorder classification, ovvero il riconoscimento della tipologia di disturbo della voce.

In definitiva, questo lavoro esplora nuove soluzioni nell'ambito della voice disorder analysis, proponendo l'utilizzo di modelli concept-based come strumento per conciliare prestazioni competitive e trasparenza nel processo decisionale. L'adozione di un approccio X-AI basato sui concetti rappresenta un passo in avanti verso l'integrazione dell'intelligenza artificiale in ambito clinico, dove la possibilità di motivare e giustificare le decisioni è spesso tanto importante quanto la correttezza della diagnosi. Le numerose direzioni di sviluppo evidenziate suggeriscono che l'unione

tra eXplainable AI e voice disorder detection sia un ambito ancora ampiamente esplorabile, con l'obbiettivo di migliorare sensibilmente la qualità della vita delle persone e garantire l'affidabilità dell'intelligenza artificiale.

Bibliografia

[1] Frank Rosenblatt. «The perceptron: A probabilistic model for information storage and organization in the brain». In: *Psychological Review* 65.6 (1958), pp. 386–408. DOI: 10.1037/h0042519 (cit. a p. 3).

[2] David E. Rumelhart, Geoffrey E. Hinton e Ronald J. Williams. «Learning representations by back-propagating errors». In: *Nature* 323.6088 (1986), pp. 533–536. DOI: 10.1038/323533a0 (cit. a p. 4).

[3] Y. Lecun, L. Bottou, Y. Bengio e P. Haffner. «Gradient-based learning applied to document recognition». In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791 (cit. a p. 4).

[4] Alex Krizhevsky, Ilya Sutskever e Geoffrey E Hinton. «ImageNet Classification with Deep Convolutional Neural Networks». In: *Advances in Neural Information Processing Systems*. A cura di F. Pereira, C.J. Burges, L. Bottou e K.Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (cit. a p. 5).

[5] Alec Radford, Karthik Narasimhan, Tim Salimans e Ilya Sutskever. *Improving Language Understanding by Generative Pre-Training*. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf. Accessed: 2025-06-23. 2018 (cit. a p. 5).

[6] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei e Ilya Sutskever. *Language Models are Unsupervised Multitask Learners*. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. Accessed: 2025-06-23. 2019 (cit. a p. 5).

[7] Tom B Brown et al. «Language Models are Few-Shot Learners». In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 1877–1901 (cit. a p. 5).

[8] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin et al. «Training language models to follow instructions with human feedback». In: *arXiv preprint arXiv:2203.02155* (2022) (cit. a p. 5).

[9] Gemini Team. *Gemini: A Family of Highly Capable Multimodal Models*. 2025. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805 (cit. alle pp. 5, 33).

[10] DeepSeek-AI Team. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. arXiv: 2501.12948 [cs.CL]. URL: https://arxiv.org/abs/2501.12948 (cit. a p. 5).

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser e Illia Polosukhin. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762 (cit. a p. 5).

[12] Robin M. Schmidt. *Recurrent Neural Networks (RNNs): A gentle Introduction and Overview*. 2019. arXiv: 1912.05911 [cs.LG]. URL: https://arxiv.org/abs/1912.05911 (cit. a p. 5).

[13] Sepp Hochreiter e Jürgen Schmidhuber. «Long Short-Term Memory». In: *Neural Computation* 9 (nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735 (cit. a p. 5).

[14] Diego Ardila et al. «End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography». In: *Nature Medicine* 25 (2019), pp. 954–961. DOI: 10.1038/s41591-019-0447-x. URL: https://doi.org/10.1038/s41591-019-0447-x (cit. a p. 6).

[15] Peng Huang et al. «Prediction of lung cancer risk at follow-up screening with low-dose CT: a training and validation study of a deep learning method». In: *The Lancet Digital Health* 1.7 (nov. 2019), e353–e362. DOI: 10.1016/S2589-7500(19)30159-1. URL: https://doi.org/10.1016/S2589-7500(19)30159-1 (cit. a p. 6).

[16] Pranav Rajpurkar et al. *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*. 2017. arXiv: 1711.05225 [cs.CV]. URL: https://arxiv.org/abs/1711.05225 (cit. a p. 6).

[17] Johns Hopkins Medicine. *Voice Disorders*. https://www.hopkinsmedicine.org/health/conditions-and-diseases/voice-disorders. Accessed on March 24, 2022. 2022 (cit. a p. 6).

[18] Nelson Roy, Ray M Merrill, Steven D Gray e Elaine M Smith. «Voice disorders in the general population: prevalence, risk factors, and occupational impact». In: *The Laryngoscope* (2005) (cit. a p. 6).

[19] Seth M Cohen. «Self-reported impact of dysphonia in a primary care population: An epidemiological study». In: *The Laryngoscope* (2010) (cit. a p. 6).

[20] Amit Meghanani, Anoop C. S e A. G. Ramakrishnan. «An Exploration of Log-Mel Spectrogram and MFCC Features for Alzheimer's Dementia Recognition from Spontaneous Speech». In: *Proceedings of the 8th IEEE Spoken Language Technology Workshop (SLT)*. Medical Intelligence e Language Engineering Lab. Shenzhen, China: IEEE, 2021. DOI: 10.1109/SLT48900.2021.9383491 (cit. a p. 7).

[21] Loukas Ilias, Dimitris Askounis e John Psarras. «Detecting dementia from speech and transcripts using transformers». In: *Computer Speech amp; Language* 79 (apr. 2023), p. 101485. ISSN: 0885-2308. DOI: 10.1016/j.csl.2023.101485. URL: http://dx.doi.org/10.1016/j.csl.2023.101485 (cit. a p. 7).

[22] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov e Abdelrahman Mohamed. *HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units*. 2021. arXiv: 2106.07447 [cs.CL]. URL: https://arxiv.org/abs/2106.07447 (cit. alle pp. 7, 20).

[23] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed e Michael Auli. *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*. 2020. arXiv: 2006.11477 [cs.CL]. URL: https://arxiv.org/abs/2006.11477 (cit. a p. 7).

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee e Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805 (cit. alle pp. 7, 21).

[25] Lotfi Salhi, Talbi Mourad e A. Cherif. «Voice disorders identification using hybrid approach: Wavelet analysis and multilayer neural networks». In: *World Academy of Science, Engineering and Technology* 45 (gen. 2008), pp. 330–339 (cit. a p. 7).

[26] Julián David Arias Londoño, Jorge Andrés Gómez García, Laureano Moro Velázquez e Juan Ignacio Godino Llorente. «ByoVoz Automatic Voice Condition Analysis System for the 2018 FEMH Challenge». In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE Xplore, dic. 2018, pp. 5228–5232. ISBN: 978-1-5386-5035-6. DOI: 10.1109/BigData.2018.8622498. URL: https://oa.upm.es/55117/ (cit. alle pp. 7, 8).

[27] Xiaoyan Peng, Hongjun Xu, Jianjun Liu et al. «Voice disorder classification using convolutional neural network based on deep transfer learning». In: *Scientific Reports* 13.1 (2023), p. 7264. DOI: 10.1038/s41598-023-34461-9 (cit. a p. 8).

[28] Xiaoping Xie, Hao Cai, Can Li, Yu Wu e Fei Ding. «A Voice Disease Detection Method Based on MFCCs and Shallow CNN». In: *Journal of Voice* (2023). ISSN: 0892-1997. DOI: https://doi.org/10.1016/j.jvoice.2023.09.024. URL: https://www.sciencedirect.com/science/article/pii/S0892199723003016 (cit. a p. 8).

[29] Dayana Ribas, Miguel Pastor, Antonio Miguel, David Martinez, Alfonso Ortega e Eduardo Lleida. «Automatic Voice Disorder Detection Using Self-Supervised Representations». In: *IEEE Access* PP (gen. 2023), pp. 1–1. DOI: 10.1109/ACCESS.2023.3243986 (cit. a p. 8).

[30] Alkis Koudounas, Gabriele Ciravegna, Marco Fantini, Erika Crosetti, Giovanni Succo, Tania Cerquitelli e Elena Baralis. «Voice Disorder Analysis: a Transformer-based Approach». In: *Interspeech 2024*. interspeech$_2$024. ISCA, set. 2024, pp. 3040–3044. DOI: 10.21437/interspeech.2024-1122. URL: http://dx.doi.org/10.21437/Interspeech.2024-1122 (cit. alle pp. 8, 9, 18, 35).

[31] Sudarsana Reddy Kadiri e Paavo Alku. «Analysis and Detection of Pathological Voice Using Glottal Source Features». In: *IEEE Journal of Selected Topics in Signal Processing* 14.2 (2020), pp. 367–379. DOI: 10.1109/JSTSP.2019.2957988 (cit. a p. 9).

[32] Rumana Islam, Esam Abdel-Raheem e Mohammed Tarique. «Voice Pathology Detection Using Convolutional Neural Networks with Electroglottographic (EGG) and Speech Signals». In: *Computer Methods and Programs in Biomedicine Update* 2 (2022). CC BY-NC-ND 4.0, p. 100074. DOI: 10.1016/j.cmpbup.2022.100074 (cit. a p. 9).

[33] Marco Tulio Ribeiro, Sameer Singh e Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. 2016. arXiv: 1602.04938 [cs.LG]. URL: https://arxiv.org/abs/1602.04938 (cit. a p. 10).

[34] Scott Lundberg e Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI]. URL: https://arxiv.org/abs/1705.07874 (cit. a p. 10).

[35] Ahmed M. Salih, Zahra Raisi-Estabragh, Ilaria Boscolo Galazzo, Petia Radeva, Steffen E. Petersen, Karim Lekadir e Gloria Menegaz. «A Perspective on Explainable Artificial Intelligence Methods: SHAP and LIME». In: *Advanced Intelligent Systems* 7.1 (giu. 2024). ISSN: 2640-4567. DOI: 10.1002/aisy.

202400304. URL: http://dx.doi.org/10.1002/aisy.202400304 (cit. a p. 10).

[36] Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Vaughan e Hanna Wallach. *Manipulating and Measuring Model Interpretability*. 2021. arXiv: 1802.07810 [cs.AI]. URL: https://arxiv.org/abs/1802.07810 (cit. a p. 10).

[37] Yacine Izza, Alexey Ignatiev e Joao Marques-Silva. *On Explaining Decision Trees*. 2020. arXiv: 2010.11034 [cs.LG]. URL: https://arxiv.org/abs/2010.11034 (cit. a p. 11).

[38] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su e Cynthia Rudin. *This Looks Like That: Deep Learning for Interpretable Image Recognition*. 2019. arXiv: 1806.10574 [cs.LG]. URL: https://arxiv.org/abs/1806.10574 (cit. a p. 11).

[39] Eleonora Poeta, Gabriele Ciravegna, Eliana Pastor, Tania Cerquitelli e Elena Baralis. *Concept-based Explainable Artificial Intelligence: A Survey*. 2023. arXiv: 2312.12936 [cs.AI]. URL: https://arxiv.org/abs/2312.12936 (cit. a p. 11).

[40] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas e Rory Sayres. *Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)*. 2018. arXiv: 1711.11279 [stat.ML]. URL: https://arxiv.org/abs/1711.11279 (cit. a p. 12).

[41] Jonathan Crabbé e Mihaela van der Schaar. *Concept Activation Regions: A Generalized Framework For Concept-Based Explanations*. 2022. arXiv: 2209.11222 [cs.LG]. URL: https://arxiv.org/abs/2209.11222 (cit. a p. 12).

[42] Yash Goyal, Amir Feder, Uri Shalit e Been Kim. *Explaining Classifiers with Causal Concept Effect (CaCE)*. 2020. arXiv: 1907.07165 [cs.LG]. URL: https://arxiv.org/abs/1907.07165 (cit. a p. 12).

[43] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva e Antonio Torralba. *Object Detectors Emerge in Deep Scene CNNs*. 2015. arXiv: 1412.6856 [cs.CV]. URL: https://arxiv.org/abs/1412.6856 (cit. a p. 13).

[44] Ruth Fong e Andrea Vedaldi. *Net2Vec: Quantifying and Explaining how Concepts are Encoded by Filters in Deep Neural Networks*. 2018. arXiv: 1801.03454 [cs.CV]. URL: https://arxiv.org/abs/1801.03454 (cit. a p. 13).

[45] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva e Antonio Torralba. «Network Dissection: Quantifying Interpretability of Deep Visual Representations». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6541–6549. DOI: 10.1109/CVPR.2017.354 (cit. a p. 13).

[46] Davis Brown e Henry Kvinge. *Making Corgis Important for Honeycomb Classification: Adversarial Attacks on Concept-based Explainability Tools*. 2022. arXiv: 2110.07120 [cs.LG]. URL: https://arxiv.org/abs/2110.07120 (cit. a p. 13).

[47] Quanshi Zhang, Ying Nian Wu e Song-Chun Zhu. *Interpretable Convolutional Neural Networks*. 2018. arXiv: 1710.00935 [cs.CV]. URL: https://arxiv.org/abs/1710.00935 (cit. a p. 14).

[48] David Alvarez-Melis e Tommi S. Jaakkola. *Towards Robust Interpretability with Self-Explaining Neural Networks*. 2018. arXiv: 1806.07538 [cs.LG]. URL: https://arxiv.org/abs/1806.07538 (cit. a p. 14).

[49] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim e Percy Liang. *Concept Bottleneck Models*. 2020. arXiv: 2007.04612 [cs.LG]. URL: https://arxiv.org/abs/2007.04612 (cit. alle pp. 14, 25).

[50] Chenming Shang, Shiji Zhou, Hengyuan Zhang, Xinzhe Ni, Yujiu Yang e Yuwang Wang. *Incremental Residual Concept Bottleneck Models*. 2024. arXiv: 2404.08978 [cs.LG]. URL: https://arxiv.org/abs/2404.08978 (cit. a p. 15).

[51] Mateo Espinosa Zarlenga et al. *Concept Embedding Models: Beyond the Accuracy-Explainability Trade-Off*. 2022. arXiv: 2209.09056 [cs.LG]. URL: https://arxiv.org/abs/2209.09056 (cit. alle pp. 15, 26).

[52] Gail Kempster, Bruce Gerratt, Katherine Verdolini Abbott, Julie Barkmeier-Kraemer e Robert Hillman. «Consensus Auditory-Perceptual Evaluation of Voice: Development of a Standardized Clinical Protocol». In: *American Journal of Speech-Language Pathology* 18 (mag. 2009), pp. 124–132. DOI: 10.1044/1058-0360(2008/08-0017) (cit. a p. 18).

[53] Angela Fan, Edouard Grave e Armand Joulin. *Reducing Transformer Depth on Demand with Structured Dropout*. 2019. arXiv: 1909.11556 [cs.LG]. URL: https://arxiv.org/abs/1909.11556 (cit. a p. 24).

[54] Albert Q. Jiang et al. *Mistral 7B*. 2023. arXiv: 2310.06825 [cs.CL]. URL: https://arxiv.org/abs/2310.06825 (cit. a p. 33).

[55] Nicolas Saenz-Lechon, Juan godino llorente, Víctor Osma-Ruiz, Manuel Blanco-Velasco e Fernando Cruz-Roldan. «Automatic Assessment of Voice Quality According to the GRBAS Scale». In: *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference* 1 (feb. 2006), pp. 2478–81. DOI: 10.1109/IEMBS.2006.260603 (cit. a p. 35).

[56] Luis M.T. Jesus, Inês Belo, Jessica Machado e Andreia Hall. «The Advanced Voice Function Assessment Databases (AVFAD): Tools for Voice Clinicians and Speech Research». In: *Advances in Speech-language Pathology*. A cura di Fernanda Dreux M. Fernandes. Rijeka: IntechOpen, 2017. Cap. 14. DOI: 10.5772/intechopen.69643. URL: https://doi.org/10.5772/intechopen.69643 (cit. a p. 55).