Studio sull'interpretabilità della conoscenza di una Rete Neurale attraverso un Albero Decisionale

1st Andrian Melnic
IT Engineering student
UNIVPM, Ancona, Italia
s1098384@studenti.univpm.it

2nd Edoardo Conti
IT Engineering student
UNIVPM, Ancona, Italia
s1100649@studenti.univpm.it

3rd Lorenzo Federici
IT Engineering student
UNIVPM, Ancona, Italia
s1098086@studenti.univpm.it

***Abstract*—Le reti neurali nel tempo si sono dimostrate particolarmente valide in compiti di classificazione, garantendo formidabili capacità di generalizzazione. Purtroppo però non c'è modo di sapere con esattezza perché una rete neurale giunga a determinate conclusioni per via dell'enorme complessità architetturale. Sotto questo aspetto sono una scatola nera dentro la quale non c'è modo di guardare. Nell'ottica di "spiegare" una determinata previsione, si propone un modello che sfrutta la conoscenza acquisita dalla rete per addestrare un albero di decisione in Prolog. Interpretando così la classificazione mediante la combinazione di scelte che maggiormente hanno influito nel processo decisionale che ha condotto all'esito restituito.**

***Index Terms*—Explainable AI, Deep Learning, Percettrone Multistrato, Rete Neurale Convoluzionale, Albero di Decisione**

I. Introduzione

L'apprendimento profondo (in inglese *Deep Learning* - DL) è un campo di ricerca del Machine Learning e dell'Intelligenza Artificiale che si basa su diversi modelli computazionali chiamati Reti Neurali (RN). Quest'ultimi cercano di riflettere il comportamento del pensiero umano, riconoscendo schemi e risolvendo problemi comuni nei campi dell'intelligenza artificiale e dell'apprendimento automatico. Questi algoritmi si sono dimostrati molto efficaci nell'esecuzione di attività di classificazione, eccellendo in presenza di tanti dati di input. Le reti neurali, modellate sulla base della semplificazione di una rete neurale biologica, sono composte da migliaia o addirittura milioni di "neuroni" artificiali densamente interconnessi tra di loro. Ciò gli conferisce un'ottima capacità di generalizzazione, ma d'altra parte introducono anche una difficoltà nell'interpretare i meccanismi di scelta che hanno portato all'esito restituito. Questa problematica è anche data dalla complessità delle rappresentazioni distribuite nei nodi di elaborazioni negli strati nascosti [1], da cui deriva anche la definizione inglese delle reti neurali come "black box". Nasce così un nuovo campo di studio chiamato Explainable AI (XAI), con l'obiettivo di riuscire ad interpretare le decisioni prese in atto dalle RN. XAI è un insieme di metodi e tecniche di intelligenza artificiale, definite a partire dagli anni '80, per risolvere un problema in modo che i risultati della soluzione possano essere compresi dagli esperti di dominio. Ad oggi soluzioni concrete in grado di offrire una spiegazione logica con simbologia comprensibile da una mente umana non sono ancora sufficientemente mature. Dunque, sembra infruttuoso ostinarsi a cercare di capire come una RN prenda una decisione di classificazione comprendendo fedelmente cosa facciano i singoli nodi di elaborazione. Al contrario, invece, è facile spiegare ed interpretare le classificazioni effettuate da un Albero Decisionale (in inglese *Decision Tree* - DT), questo perché le sue decisioni dipendono da una sequenza lineare di scelte regolate direttamente dai dati di input. Sulla base di ciò, si propone un modello (Fig. 1) che sfrutta la conoscenza acquisita della RN per addestrare un albero di decisione, con l'intento di rendere "spiegabili" le previsioni risultanti.

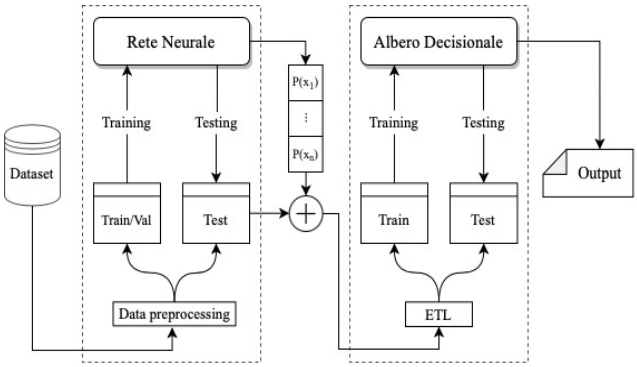

Fig. 1: Soluzione che massimizza sia la generalizzazione del modello che l'interpretabilità dell'output

II. Metodologia

A. Dataset

Per la realizzazione del progetto si è sfruttata la banca dati *UCI Machine Learning Repository*, dalla quale si è reperito un dataset sulla quale condurre l'esperimento. Si tratta di dati relativi alla stima dell'obesità di individui sulla base delle loro abitudini alimentari e condizioni fisiche. Il dataset è frutto di uno studio condotto su soggetti con età compresa tra 14 e 61 anni provenienti dal Messico, Perù e Colombia [2]. I dati disponibili contano 17 attributi (compreso il campo target) e più di 2000 record. Le features in questione descrivono condizioni intelligibili al fine di offrire una panoramica sullo stato della persona, ad esempio: se un membro della famiglia ha sofferto o soffre di sovrappeso oppure quanto spesso l'individuo mangia cibi fuori pasto. I significati associati alle features sono indicati nell'articolo di riferimento [2].

B. Data Preprocessing

Per poter essere analizzato da una rete neurale profonda, un dataset deve prima essere elaborato con una serie di trasformazioni. Risulta infatti necessario codificare i valori non numerici, un metodo che prevede quindi la conversione di campi categorici in campi numerici mantenendo consistenza tra le varie associazioni. Sono anche stati opportunamente arrotondati i valori numerici dei record generati sinteticamente tramite il tool Weka, come segnalato nello studio di riferimento del dataset [2]. In seguito sono state impiegate tecniche di analisi esplorativa dei dati (in inglese *Exploratory Data Analysis* - EDA) al fine di comprendere al meglio il dataset e riepilogarne le caratteristiche principali, come connessioni e relazioni tra variabili indipendenti e dipendenti. Questo passaggio risulterà fondamentale, nello specifico per implementare moderne reti neurali convoluzionali partendo da dati di natura tabellare. Quindi si è dapprima generata una matrice di correlazione per poi ordinare in una lista decrescente le features più statisticamente correlate al campo target. Il risultato è stato visualizzato attraverso una *heatmap* come riportato in Figura 2, una tecnica di visualizzazione dei dati in due dimensioni che riporta il magnitudo di un fenomeno attraverso gradienti di colori.

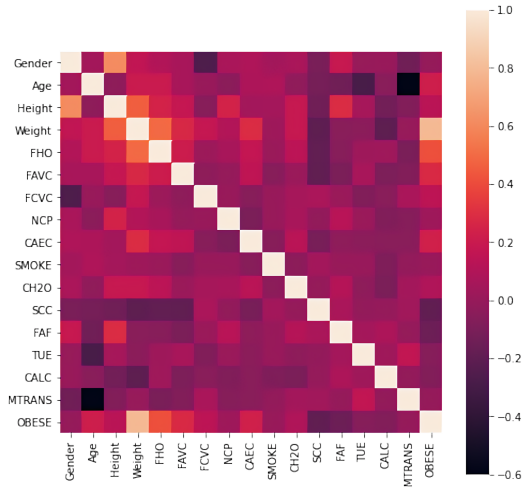

Fig. 2: Mappa di correlazione

I primi cinque attributi che mostrano una maggiore correlazione con il campo target sono `WEIGHT`, `FHO`, `FAVC`, `CAEC` e `AGE` che rispettivamente indicano: il peso, se un famigliare soffre di sovrappeso, se consuma spesso cibi altamente calorici, quanto spesso mangia cibi fuori pasto ed infine l'età. Per motivi direttamente dipendenti dalla natura dell'esperimento, in questa circostanza è necessario un insieme di test (*test set*) più copioso rispetto agli standard. Dato che questo fungerà come parte del nuovo dataset per addestrare l'albero decisionale. Dunque, la ripartizione dei dati per addestrare la rete neurale, che verrà discussa nella Sottosezione II-C, prevede che il 40% di questi formino il *test set* (845) ed il 20% della parte restante il *validation set* (254), lasciando al *train set* un totale di 1012 campioni.

C. Rete neurale a Percettrone multistrato (MLP)

L'architettura di rete neurale artificiale più impiegata per dati tabulari è il Percettrone multistrato (in inglese *Multy Layer Perceptron* - MLP) [3]. Il modello MLP implementato consiste in quattro livelli: un input layer, due hidden layer ed un output layer. In ogni layer tutti i neuroni sono connessi con quelli del layer successivi, motivo per cui vengono definiti *fully connected*. Ad eccezione dei nodi di input, ogni neurone utilizza la funzione d'attivazione non lineare *Rectified Linear Unit* (ReLU) ed invece il nodo di output la funzione sigmoidea. Per la fase di training i modelli MLP utilizzato una tecnica di supervised learning chiamata *backpropagation*, una procedura per regolare ripetutamente i pesi dei nodi in modo da ridurre al minimo la differenza tra il l'output della RN e l'output desiderato. L'input layer conta 16 nodi, pari al numero di features in ingresso dal dataset. I due strati interni sono composti da 32 neuroni ciascuno, un opportuno compromesso di densità che esclude un eccessivo "attaccamento" ai dati utilizzati per l'addestramento. L'output layer $\hat{y}$ ha un'unica uscita dato che il problema di classificazione è binario. Lo schema dell'architettura MLP impiegata è riportata in Figura 3. L'intera implementazione è stata poi ottimizzata introducendo negli strati densi dei *Kernel Regularizers* ed aggiungendo fra questi dei *Dropout layers*. La prima miglioria è volta ad applicare piccole penalità sui pesi dei neuroni che verranno poi sommate alla funzione di loss [4]. La seconda invece rimuove il contributo del 20% di nodi scelti casualmente ad ogni step durante il training, riducendo la possibilità di *overfitting* [5].

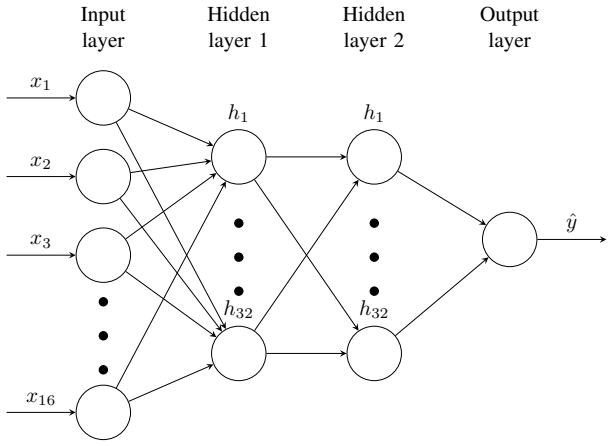

Fig. 3: Architettura MLP implementata

D. Secondo approccio: Rete Neurale Convoluzionale (CNN)

Le RN hanno dimostrato enormi progressi nell'ambito della classificazione, in particolar modo le Reti Neurali Convoluzionali (in inglese *Convolutional Neural Networks* - CNN). Le quali riescono ad avvicinarsi, o in alcuni casi superare, l'accuratezza umana nella classificazione delle immagini. Pertanto, data la loro ottima efficienza e generalizzazione, si è scelto di testare anche un modello CNN. Lo scopo era di constatare se effettivamente questo approccio alternativo offra

un miglioramento significativo dei risultati (nell'ambito di questo progetto), confrontandoli con quelli del MLP. Ad oggi però, non ci sono stati progressi sostanziali nell'applicazione di CNN su dati tabulari, dunque si è fatto ricorso al metodo proposto da Buturović et al. [6]. Questo propone una nuova tecnica chiamata *TAbular Convolution* (TAC) che permette la trasformazione di dati descrittivi in immagini. La conversione avviene trattando ciascun vettore di features dei dati come un filtro (*kernel*), che sarà poi convoluto ad un'unica immagine di base. Al fine di ottenere un dataset di immagini che rappresentino, univocamente, le righe dei dati tabulari. Più nello specifico la creazione del kernel avviene riorganizzando il vettore delle caratteristiche (normalizzato sottraendo il suo valore medio μ) in una matrice quadrata dispari (Fig. 4).

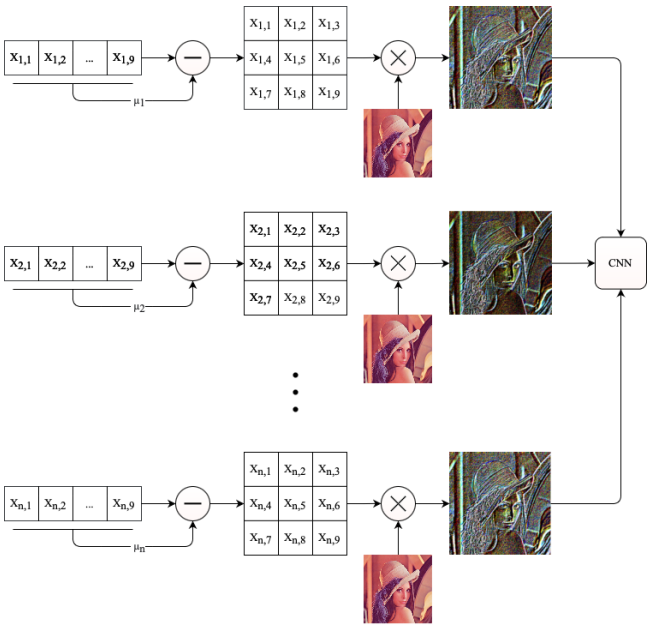

Fig. 4: TAbular Convolution

Nel caso in cui il numero delle features è un quadrato di un numero intero dispari maggiore di 1, può essere convertito senza attuare nessun espediente. La situazione cambia nel momento in cui il numero delle features di input non è un quadrato dispari, infatti in questa situazione bisognerà ricorrere a tecniche di:

- Padding (riempimento): aggiunta di zeri, rumore casuale o features ingegnerizzate;
- Trimming (taglio): rimozione features;

Il dataset impiegato conta 16 features, quindi il kernel quadratico dispari più vicino è una matrice 3x3, composta da $3^2 = 9$ features (Fig. 5). Come anticipato nella Sottosezione II-B, tramite tecniche di analisi dei dati abbiamo ottenuto le 9 features più correlate al campo target che, oltre a quelle già indicate nella sezione appena citata, figurano `FCVC`, `Height`, `CH2O` e `NCP`. Le quali indicano rispettivamente: frequenza di consumo di vegetali, altezza, consumo di acqua giornaliera e numero medio di pasti giornalieri.

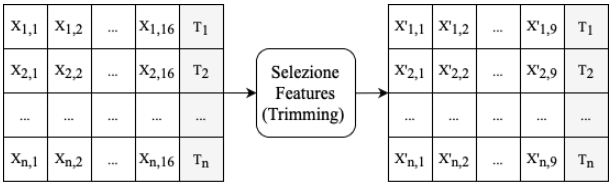

Fig. 5: Schema rimozione features

Formulati i kernel e convoluti all'immagine di riferimento (Lenna, figura classica della Computer Vision) si è ottenuto il dataset di immagini che poi potrà essere fornito come input alla rete convoluzionale (Fig. 4). In questo progetto sono state implementate e valutate due backbone CNN, una VGG-19 e una ResNet-50, entrambe pre-allenate su imageNET.

E. ETL e Albero Decisionale PROLOG

Per l'allenamento dell'albero decisionale sono stati impiegati i dati di test, sostituendo i valori del campo target originali con le predizioni della rete neurale. Tuttavia, prima di fornire questi dati all'algoritmo, è necessaria una fase di ETL (Extract, Transform, Load) nella quale sono state applicate le seguenti trasformazioni tramite un programma Prolog dedicato:

1) i valori continui degli attributi numerici sono stati discretizzati in intervalli categorici;
2) ogni elemento del test set è stato convertito in un predicato che include, come termini, la predizione della classe di appartenenza dell'elemento fornita dalla rete neurale e la lista delle coppie `attributo = valore`.

Una volta completata la trasformazione, il 70% dell'insieme totale dei predicati (845) è stato utilizzato come *train set*, mentre il restante 30% come *test set*.

Il programma di induzione dell'albero decisionale è stato implementato secondo il seguente algoritmo ricorsivo:

Algorithm 1 Induzione Albero di Decisione

Require: $crit, EsG, Es, Att$
  $Albero \leftarrow null$
  if $Es = \varnothing$ **then**
    return $classi(EsG) \leftarrow ???$
  else if Stessa classe per ogni $e \in Es$ **then**
    return $classificazione$
  else if $Att = \varnothing$ **then**
    return $classi(Es) \leftarrow$ Classe dominante
  else
    $A \leftarrow ScegliAttributo(crit, Att, Es)$
    $Albero \leftarrow$ nuovo albero con radice $test(A)$
    $AttR \leftarrow Att - A$
    for all $val \in Valori(A)$ **do**
      $EsAval \leftarrow \{e : e \in Es$ e $e.A = val\}$
      $SAlbero \leftarrow InduceAlbero(crit, Es, EsAval, AttR)$
      Aggiungi $SAlbero$ ad $Albero$ con etichetta $\{A = val\}$
    end for
  end if
  return $Albero$

Il programma partiziona l'insieme degli esempi applicando i valori dell'attributo risultato più significativo secondo il criterio indicato dal termine `crit`. Quindi, per ogni valore di tale attributo si avrà un sotto albero in cui saranno presenti gli esempi (`Es`) dove l'attributo più significativo ha quel valore. I criteri implementati sono *Gini Index*, *Information Gain* e *Gain Ratio*. Si procede quindi con l'applicazione ricorsiva dell'algoritmo ad ogni sotto albero così ottenuto. La ricorsione termina in un nodo se in esso si verifica una delle seguenti condizioni:

- Non ci sono più esempi da discriminare, gli esempi del nodo genitore (`EsG`) vengono classificati con l'etichetta '`???`';
- Tutti gli esempi presenti in `Es` sono etichettati con la stessa classe, perciò non è più necessario discriminare;
- Ci sono ancora esempi misti che devono essere partizionati ma `Att` è vuoto, dunque non ci sono più attributi per discriminare. In questo caso si assegna a tutti gli esempi la classe dominante (quella più frequente).

I codici sorgente, con i due approcci di rete neurale implementati in notebook Python e l'albero in Prolog, sono liberamente fruibili dalla repository online pubblicata su GitHub[1].

III. Risultati

In Tabella I sono mostrate le performance delle classificazioni delle varie reti neurali implementate. Tali risultati sono stati ottenuti sfruttando l'algoritmo di ottimizzazione stocastica *Adam*, mentre la funzione di loss utilizzata è la *Binary Cross Entropy*. L'addestramento delle RN sfrutta meccanismi di riduzione dinamica del learning rate sulla base di metriche specifiche. Inoltre, un ulteriore callback è impostato per interrompere il training nella casistica in cui non si presentino miglioramenti nella funzione obiettivo per 10 epoche consecutive. Infine, la varie reti sono state addestrate per 100 epoche, una dimensione confacente al dataset impiegato.

	MLP	ResNet-50	VGG-19
Accuracy	0.9065	0.9384	0.9173
Loss	0.2656	0.1389	0.1603

TABLE I: Performance di classificazione delle reti neurali sul test set

Come è possibile notare dalla Tabella I, le due CNN hanno un leggero vantaggio in termini di performance rispetto a MLP. C'è da sottolineare però, che le reti convoluzionali sono notevolmente più complesse di una semplice rete neurale composta dai soli strati densi. È lecito pensare quindi, che tale aumento di complessità sia il motivo di questo miglioramento dei risultati, nonostante la perdita di informazioni dovuta al processo di feature selection descritto nella Sottosezione II-B e la possibile introduzione di rumore a seguito dell'applicazione dei kernel sull'immagine di riferimento. Riflettendo su questo piccolo margine di miglioramento da parte delle due CNN rispetto il metodo MLP, si può evincere che ai fini dell'obiettivo esposto nella Sezione I, le prestazioni delle reti neurali convoluzionali non sono così superiori da far preferire quest'ultime ad una rete neurale a percettrone multistrato. Infatti, le due CNN implementate, benché generalizzino meglio, sono caratterizzate da tempistiche di esecuzione maggiori e da una maggiore occupazione di memoria. Tutto ciò è influenzato sia dalla presenza dell'algoritmo TAC [6] per la generazione delle immagini da dati tabulari che dalla maggior complessità dei modelli CNN. Pertanto si è preferito continuare lo studio analizzando, attraverso un albero decisionale, le predizioni ottenute dall'approccio MLP. Di seguito sono mostrati i risultati ricavati con i vari criteri di divisione dell'albero. Questi mostrano come l'albero riesca a ottenere ottime performance, con un'accuratezza massima di circa il 96% utilizzando il Gain Ratio come criterio di split (Tab. IIc).

		Predicted: Positive	Predicted: Negative		
Actual	Positive	TP:128	FN:8	Accuracy	0.9430
	Negative	FP:6	TN:104	NC	8

(a) Indice di Gini

		Predicted: Positive	Predicted: Negative		
Actual	Positive	TP:129	FN:7	Accuracy	0.9508
	Negative	FP:5	TN:103	NC	10

(b) Information Gain

		Predicted: Positive	Predicted: Negative		
Actual	Positive	TP:132	FN:3	Accuracy	0.9591
	Negative	FP:7	TN:103	NC	9

(c) Gain Ratio

TABLE II: Matrici di confusione degli alberi di decisione

IV. Discussioni

Il processo d'interpretazione delle previsioni della rete neurale consiste nel ripercorrere puntualmente il grafo generato dall'albero di decisione addestrato sulla conoscenza della RN stessa. In questa sezione si cercherà di esporre tale procedimento prendendo in esame una riga casuale dal dataset, riportata nella Tabella III. L'approccio sarebbe identico nel caso in cui si tratti di una misurazione di uno o più soggetti nuovi.

GENDER	AGE	WEIGHT	FHO	...	TUE	CALC	SCC
1	23	91.07	1	...	1	3	0

TABLE III: Estratto del record d'esempio

[1]https://github.com/SasageyoOrg/explainable-ai

Il primo passo prevede quindi di ingaggiare la rete neurale pre-addestrata, sfruttandone l'elevata capacità di generalizzazione, per classificare se il campione appartiene alla classe OBESE. In effetti, l'esempio preso in considerazione risulta essere appartenente alla suddetta classe. Così facendo si acquisisce una conoscenza sufficientemente attendibile che tuttavia è priva di argomentazioni, o meglio, pecca di giustificazioni rappresentando squisitamente il risultato di un'analisi predittiva della rete neurale. Processo su cui è impraticabile una qualsivoglia serie di indagini per estrarre la logica delle scelte per via dell'enorme complessità dell'architettura. Tuttavia qui interviene il grafo prodotto dall'albero decisionale offrendo un metodo sistematico per interpretare tale predizione. Infatti, ripercorrendo i nodi del grafo sulla base dei dati del record preso in esame fino al raggiungimento della foglia (Fig. 6), si formulerà la combinazione di features che hanno portato il classificatore ad assegnare tale responso. Si vuole sottolineare che a fronte dell'implementazione dei tre diversi criteri di partizionamento (citati nella Sottosezione II-E) è possibile estrapolare tre diverse interpretazioni. Fermo restando che tali soluzioni portano alle medesime classificazioni, è possibile giungere alle stesse conclusioni basandosi su combinazioni di scelte diverse. Nello specifico gli alberi generati tramite l'indice di Gini e la variazione d'entropia risultano altamente paragonabili, con delle differenze marginali al crescere della profondità. Viceversa, Gain Ratio produce un albero con differenze significative; probabilmente per via della formulazione matematica che differisce sensibilmente dai due precedenti. Garantendo un'accuratezza superiore (come riportato in Tab. IIc), per l'analisi che segue si adotta l'albero generato da quest'ultimo criterio.

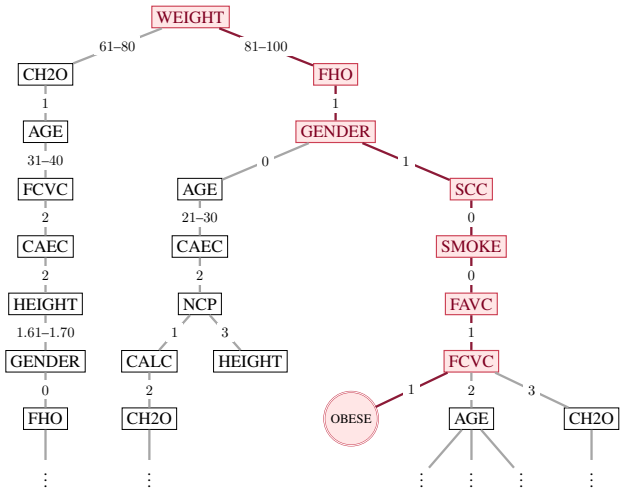

Fig. 6: Estrazione serie di scelte che conducono alla classificazione del campione d'esempio

Di conseguenza è ora possibile argomentare l'esito e il processo decisionale della rete artificiale. In pratica oltre che offrire una previsione sui dati in ingresso sarà anche possibile rispondere al perché di tale output. A titolo esemplificativo, i dati dell'individuo preso in considerazione indicano che soffre di obesità; prevalentemente per via del peso che ricade nella fascia 81-100Kg (WEIGHT) e anche perché un membro della famiglia soffre o ha sofferto di sovrappeso (FHO), è di genere maschile (GENDER), non monitora il consumo di calorie (SCC), non è un fumatore (SMOKE), consuma spesso cibi ad alto contenuto calorico (FAVC) ed infine perché non mangia mai verdure (FCVC).

WEIGHT	FHO	GENDER	SCC	SMOKE	FAVC	FCVC
81-100	1	1	0	0	1	1

TABLE IV: Attributi che "spiegano" l'esito della rete

Come accennato in precedenza, gli alberi prodotti con indice di Gini o variazione d'entropia avrebbero proposto una spiegazione alternativa. Infatti quest'ultimi avrebbero associato all'obesità dell'individuo in esame il suo peso, l'altezza, l'età e anche il consumo giornaliero d'acqua (CH2O).

V. Conclusioni

In questo articolo si è descritto un metodo per combinare le capacità di generalizzazione di una rete neurale con quelle interpretative dell'albero decisionale. Elaborando così un approccio in grado di fornire risposte più informative. Infatti, nell'eventualità in cui siano richieste le cause per cui il classificatore abbia restituito un particolare esito piuttosto che un altro, l'albero generato è in grado di formulare una combinazione logica di scelte che maggiormente hanno influito nel processo decisionale. Alla luce delle riflessioni riportate nella Sezione IV, il suddetto metodo trova un possibile campo d'applicazione in tutti quegli ambiti in cui è tassativa un'interpretabilità degli esiti forniti in output. Si pensi al settore medico, dove il referto di un'analisi prodotta da un sistema d'apprendimento automatico profondo sia coadiuvato da una serie di probabili cause in grado di offrire maggiore supporto decisionale agli esperti di dominio. In modo analogo anche nel contesto di Industria 4.0, dove processi d'automazione industriale per aumentare la produttività degli impianti e/o migliorare la qualità dei prodotti possono essere rafforzati mirando a risultati caratterizzati da una maggiore granularità; anche qui facilitando le diagnosi degli addetti ai lavori.

References

[1] Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree, 2017.

[2] Fabio Mendoza Palechor and Alexis de la Hoz Manotas. Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from colombia, peru and mexico. *Data in Brief*, 25:104344, 2019.

[3] Popescu Marius, Valentina Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8, 07 2009.

[4] Russell Reed and Robert J MarksII. *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press, 1999.

[5] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 06 2014.

[6] Ljubomir Buturović and Dejan Miljković. A novel method for classification of tabular data using convolutional neural networks. *bioRxiv*, 2020.