



UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia “Galileo Galilei”

Corso di Laurea in Fisica

Tesi di Laurea

Deep Neural Networks per la ricostruzione dell’energia di eventi di decadimento beta inverso nell’esperimento JUNO

Relatore

Prof. Alberto Garfagnini

Correlatore

Dr. Yury Malyshkin

Laureando

Francesco Vidaich

Anno Accademico 2017/2018

Indice

1	Introduzione	1
1.1	JUNO	1
1.2	Deep Neural Networks	2
2	Sviluppo Modelli	4
2.1	Presentazione Dati	4
2.2	Discriminazione eventi esterni al volume di fiducia	5
2.3	Ricostruzione Energia	7
2.4	Ricostruzione Vertice	9
3	Risposta dei Modelli al Dark Noise	12
4	Conclusioni	16

Capitolo 1

Introduzione

1.1 JUNO

L'esperimento Jiangmen Underground Neutrino Observatory [1, 2] (JUNO) è stato progettato per studiare la fisica dei neutrini e ha come obiettivo principale la determinazione della gerarchia di massa (Mass Hierarchy) dei neutrini, una delle restanti proprietà fondamentali indeterminate del Modello Standard, ad oggi non conosciuta. JUNO studierà l'interazione dei neutrini prodotti dai reattori nucleari tramite reazioni denominate Inverse Beta Decay (IBD), del tipo $\bar{\nu}_e + p \rightarrow e^+ + n$.

JUNO verrà costruito a Kaiping, in Cina, circa 700 metri sottoterra; si trova a circa 53 km dai reattori nucleari di Yangjiang e Taishan dove vengono prodotti gli antineutrini elettronici necessari per la IBD. L'esperimento, di cui si è iniziata la costruzione nel 2015, dovrebbe entrare in funzione nel 2020.

Il rivelatore centrale di JUNO ha una forma sferica dentro la quale sono contenuti 20000 tonnellate di liquido scintillatore. Sulla superficie sono posti circa 18000 PMT di 20 pollici e 25000 PMT di 3 pollici per la rivelazione dei fotoni. È presente inoltre una sfera acrilica, di diametro interno uguale a 35.4m, il cui scopo è contenere il liquido scintillatore. Questa sfera è immersa in una vasca riempita d'acqua ed è sostenuta da una capriata di acciaio inossidabile, il cui diametro è 40.1 m. L'acqua è usata per schermare i PMT e il terreno circostante dalle radiazioni. Nell'acqua sono inoltre installati altri 2000 fotomoltiplicatori, usati per rivelare radiazioni Cherenkov e identificare eventi di fondo. Si riporta in figura 1 la struttura del rivelatore.

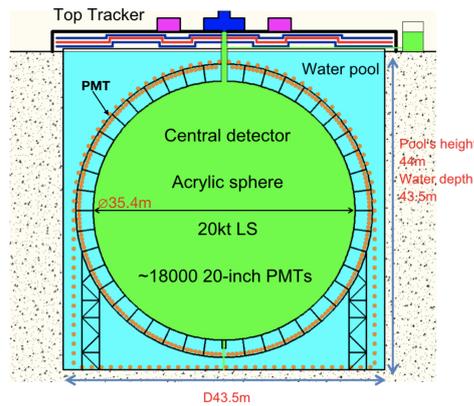


Figura 1: Struttura dell'esperimento JUNO

Nell'esperimento JUNO gli antineutrini elettronici prodotti nel reattore reagiscono con i protoni dello scintillatore attraverso la IBD, producendo positroni e neutroni. Uno degli obiettivi principali di questo lavoro è la ricostruzione dell'energia del positrone: esso annichila con un elettrone e l'energia rilasciata nel liquido scintillatore produce fotoni (di scintillazione) che vengono rivelati dai PMT; per poter studiare la gerarchia di massa dei neutrini, è necessario ottenere una risoluzione di $3\%/\sqrt{E(\text{MeV})}$ su questa energia. Il neutrone invece termalizza e viene catturato da un nucleo producendo altri fotoni, anch'essi rivelati dai PMT, che quindi servono ad identificare le IBD dal fondo.

Per ricostruire con precisione la dinamica dell'evento è necessario stimare oltre all'energia il vertice dell'antineutrino all'interno del rivelatore. Infatti, a causa delle elevate dimensioni di JUNO, parte dell'energia dell'evento potrebbe non venire rivelata dai PMT, poiché alcuni dei fotoni prodotti interagiranno col liquido prima di essere catturati.

Per la ricostruzione di questi eventi sono stati quindi sviluppati tre modelli: uno per la ricostruzione dell'energia, uno per la ricostruzione del vertice e un terzo che viene utilizzato prima di questi e che si occupa di discriminare gli eventi esterni al volume di fiducia del rivelatore.

1.2 Deep Neural Networks

Per la ricostruzione di energia e vertice degli eventi è stato necessario usare algoritmi di machine learning. In particolare, sono state utilizzate delle reti neurali profonde (Deep Neural Networks) che rappresentano una delle strutture più comuni di Deep Learning [4].

In generale una rete neurale consiste in un modello matematico composto di neuroni artificiali interconnessi. I modelli utilizzati (rappresentati in figura 2) sono sequenziali "Feed-Forward", ciò vuol dire che sono strutturati in layer e che le informazioni attraversano la rete mantenendo la stessa direzione. In queste reti ogni neurone è definito da una funzione che prende in ingresso i valori uscenti da tutti i neuroni del layer precedente, ed arriva ad assumere un valore che verrà poi usato dai neuroni del layer immediatamente successivo. In ogni struttura il primo e ultimo layer rappresentano rispettivamente i dati in ingresso e in uscita del modello, chiamati features e labels. Un Deep Neural Network (DNN) rappresenta semplicemente una rete neurale che contiene più di un layer tra quelli di ingresso e uscita (chiamati hidden layers).

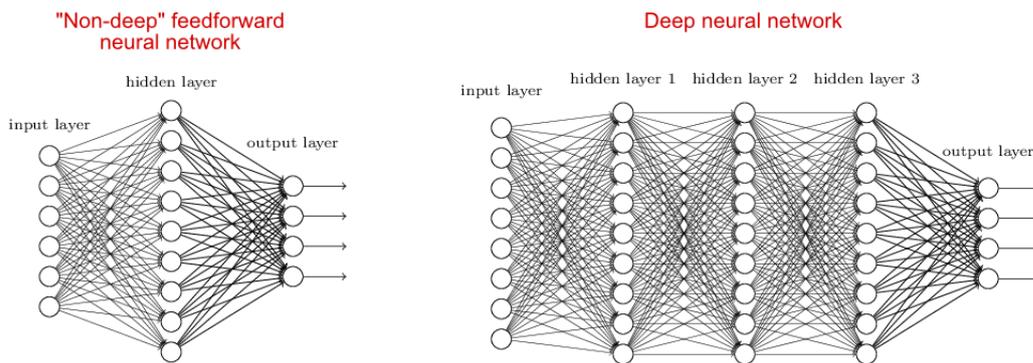


Figura 2: Struttura di una rete neurale semplice e di una profonda

Per sviluppare un modello di questo tipo è necessario effettuare un allenamento utilizzando eventi di cui conosciamo le caratteristiche direttamente misurabili dal rivelatore e quelle da stimare (usati dalla rete neurale come features e labels), in questo modo i parametri che legano i vari neuroni artificiali vengono finemente regolati in modo che le previsioni dei labels per un certo evento corrispondano con quelle vere. Questo processo corrisponde a trovare il minimo di una funzione (chiamata Loss function) che spesso rappresenta l'errore commesso dal modello.

Una volta terminato l'allenamento è possibile utilizzare il modello per stimare le labels di eventi di cui si conoscono solo le features.

Il DNN [5] è un algoritmo di machine learning molto utilizzato per risolvere principalmente due tipi di problemi:

- **Classificazione:** gli eventi analizzati sono suddivisi in categorie e il modello deve riconoscere a quale di queste appartengano. In pratica, l'output è formato da una label per ogni categoria e si stima la probabilità che l'evento appartenga ad ognuna di queste;
- **Regressione:** consiste nella stima di uno o più valori continui sulla base delle features dell'evento. In questo caso è presente un label per ogni grandezza da stimare.

Nel capitolo successivo si vedranno modelli che provano a risolvere entrambi i problemi.

Per lo svolgimento di questo lavoro sono state utilizzate le librerie di Tensorflow [6] e Keras [7] per lo sviluppo di reti neurali.

Capitolo 2

Sviluppo Modelli

2.1 Presentazione Dati

I dati utilizzati per l'allenamento dei modelli sono stati prodotti da simulazioni Montecarlo poiché attraverso esse è possibile descrivere fedelmente il comportamento del rivelatore. Inoltre, utilizzando Montecarlo, si è a conoscenza di grandezze non direttamente misurabili nel rivelatore, come energia e vertice di ogni evento. Utilizzando queste ultime come labels è possibile sviluppare reti neurali che si allenino a stimarle partendo da quantità misurabili nel rivelatore.

L'allenamento viene eseguito utilizzando un file contenente un milione di eventi, nei quali l'energia cinetica E_k del positrone è uniformemente distribuita tra 0 e 10 MeV. È necessario ricordare che l'energia trasportata dai fotoni e raccolta nel rivelatore è uguale a quella del positrone più 1.022 MeV (in quanto positrone ed elettrone annichilano e la loro massa è trasformata in energia).

Attraverso simulazioni Montecarlo, sono stati generati anche altri 11 file, ognuno contenente 2000 eventi con energia E_k fissata (uguale a 0,1,2,...,10 MeV), per verificare la bontà dei modelli in funzione dell'energia. Questa verifica è stata fatta confrontando i valori delle labels stimati dai modelli con i valori veri ottenuti dalla simulazione.

Definiamo ora le grandezze principali che verranno usate come features o labels:

- E_0 : Energia totale del positrone, uguale a $E_k + 0.511$ MeV;
- x, y, z : Coordinate del vertice dell'antineutrino (in millimetri);
- R : Distanza del vertice dal centro del rivelatore (in millimetri);
- TotalPE: Numero di fotoni raccolti da tutti i PMT di 20 pollici (per l'analisi non sono stati considerati i PMT di 3 pollici);
- ht_mean (Hit Time mean): Tempo medio di arrivo dei fotoni ai PMT (si usa come zero l'istante in cui arriva il primo fotone perché non è possibile conoscere quello in cui avviene il decadimento);

- CoH (Center of Hits): Per avere una stima approssimata del vertice si esegue una "media pesata" delle posizioni dei PMT colpiti da fotoni:

$$\vec{CoH} = \sum_{i=0}^{N_{PMT}} \frac{\vec{x}_i n_i}{N_{PE}}$$

Dove \vec{x}_i rappresenta la posizione dell' i -esimo PMT, che raccoglie n_i fotoelettroni. N_{PE} rappresenta invece il numero totale dei fotoelettroni raccolti.

Prima di essere utilizzate, tutte queste grandezze vengono normalizzate in modo che il loro dominio coincida con $[0,1]$ (oppure $[-1,1]$ nel caso in cui assumano anche valori negativi). Questa normalizzazione è necessaria e aumenta notevolmente la precisione del modello in quanto è possibile che, eseguendo l'allenamento senza di essa, le grandezze che assumono valori maggiori abbiano un peso maggiore delle altre.

2.2 Discriminazione eventi esterni al volume di fiducia

Attraverso simulazioni Montecarlo sono stati prodotti eventi distribuiti uniformemente su tutto il volume interno del rivelatore. Tuttavia il suo volume di fiducia (volume all'interno del quale gli eventi rilevati possono essere considerati affidabili) è più piccolo principalmente a causa della notevole differenza tra gli indici di rifrazione del liquido di scintillazione (1.49) e dell'acqua (1.33). Questa differenza comporta fenomeni di rifrazione dei fotoni sul confine tra i due liquidi, modificandone anche il tempo di volo.

Gli eventi con vertice troppo vicino alla superficie generano fotoni che colpiscono la suddetta con angolo di incidenza molto elevato, spesso causandone la riflessione totale. In questi casi una parte rilevante dei fotoni prodotti verrà riflessa anche più volte sulla superficie interna; si osserva quindi un aumento dell' ht_mean e un abbassamento dell'energia rivelata, in quanto questi fotoni hanno maggiore probabilità di essere assorbiti nel liquido. Per questo motivo, prima della ricostruzione dell'energia e del vertice è stato sviluppato un modello che scartasse gli eventi esterni al volume di fiducia, in modo da eliminare gli effetti indesiderati appena descritti.

Basandosi sul volume di fiducia del rivelatore, definito da una sfera di raggio 17.2 m, è stato sviluppato un primo modello per riconoscere quali eventi avvengono all'interno e quali all'esterno di esso, in modo da utilizzare solo i primi per l'analisi successiva. Per questo compito è stato usato un classificatore con due labels, che rappresentano le probabilità che il vertice dell'evento abbia distanza maggiore e minore di 17.2 m dal centro della sfera: in altre parole, il risultato della previsione è composto da una coppia di valori continui appartenenti all'intervallo $[0,1]$. Si procede poi a salvare in un file gli eventi la cui probabilità del secondo caso (interni al volume di fiducia) sia maggiore, che verrà poi passato ai due modelli successivi per la ricostruzione dell'energia e del vertice.

La rete di questo classificatore è formata dai layer di input e output più 4 hidden layers ognuno formato da 128 nodi. Le grandezze che utilizza come features sono TotalPe, ht_mean , il raggio R_{CoH} calcolato con le componenti del Center of Hits (cioè $R_{CoH} = \sqrt{X_{CoH}^2 + Y_{CoH}^2 + Z_{CoH}^2}$) e la singola componente Z_{CoH} . È necessario aggiungere alle features anche quest'ultima grandezza, in quanto nel rivelatore non è presente una perfetta simmetria sferica, mentre non serve usare X_{CoH} e Y_{CoH} perché viene comunque conservata la simmetria cilindrica lungo l'asse verticale Z . Per questo modello si è utilizzato come loss function l'errore quadratico medio

(MSE), definito come:

$$MSE = \frac{1}{N} \sum_{i=0}^N (Y_i^{pred} - Y_i^{true})^2$$

Dove N rappresenta il numero di dati nel set analizzato, Y_i^{true} rappresenta il valore vero dei label dell' i -esimo evento (che possono essere uguali solo a (0,1) o (1,0)) e Y_i^{pred} è il valore dei label previsti dal modello, cioè la probabilità che l'evento avvenga dentro e che avvenga fuori dal volume di fiducia (formato da due valori continui e appartenenti a [0,1]).

Partendo dagli 11 file con eventi ad energia fissata ne sono stati creati in questo modo altri 11, dopo aver scartato quelli troppo vicini alla superficie sferica. Sapendo a priori la distanza dal centro R di ogni vertice è stato possibile calcolare l'accuratezza del modello per ogni file (quindi in funzione delle energie):

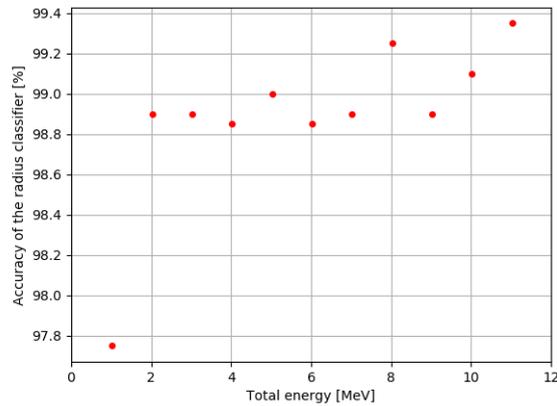


Figura 3: Accuratezza modello per ogni energia

L'accuratezza è calcolata come rapporto tra il numero di eventi riconosciuti in modo corretto dal modello e il numero totale degli eventi analizzati. Questo vuol dire che nell'errore (circa uguale all'1% per quasi tutte le energie) sono compresi sia gli eventi scartati che invece appartenevano al volume di fiducia, sia gli eventi accettati che non ne facevano parte.

Come si vede in figura 3, le accuratze ottenute risultano buone per la maggior parte delle energie, quindi complessivamente gli eventi che verranno passati agli altri modelli corrispondono a quelli avvenuti all'interno del volume di fiducia.

Si nota però più del 2% di errore per gli eventi caratterizzati da $E_k = 0$ MeV; come si vedrà anche più avanti, a basse energie la precisione dei modelli cala. Questo accade perché esiste una dipendenza lineare tra energia dell'evento e numero di fotoni che esso genera; a basse energie verranno generati meno fotoni e il CoH, calcolato come media pesata delle posizioni dei PMT colpiti, avrà quindi risoluzione minore. In questo caso, la precisione del modello scende in modo visibile solo per eventi con energia minima $E_k = 0$ MeV, rimanendo comunque accettabile.

2.3 Ricostruzione Energia

Si riporta ora il primo modello sviluppato per la ricostruzione degli eventi. La rete neurale utilizzata deve quindi eseguire una regressione per stimare il valore del suo unico label: l'energia totale del positrone E_0 . Dal momento che la regressione è un problema leggermente più complesso della classificazione, si è scelto di introdurre un hidden layer aggiuntivo rispetto al classificatore descritto prima, ma le features utilizzate sono le stesse (TotalPE, ht_mean, R_{CoH} e Z_{CoH}).

In questo modello non è stato usato l'errore quadratico medio come loss function, ma si è scelto l'errore percentuale assoluto medio (MAPE), definito come:

$$MAPE = \frac{1}{N} \sum_{i=0}^N \left| \frac{E_i^{pred} - E_i^{true}}{E_i^{true}} \right|$$

Dove N rappresenta il numero di dati nel set analizzato, E_i^{true} rappresenta il valore vero del label (cioè dell'energia) dell' i -esimo evento e E_i^{pred} è il valore previsto dal modello. Si è preferito utilizzare questa loss function al posto del MSE perché si voleva evitare che gli errori sulle stime delle energie più alte pesassero di più degli errori sulle stime di energie più basse, altrimenti su queste ultime si sarebbero ottenuti degli errori percentuali molto maggiori di quelli che verranno mostrati sotto.

L'allenamento è stato svolto utilizzando il file contenente un milione di eventi, poi si sono usati i prodotti del primo modello per verificarne il funzionamento per ogni energia. Ognuno di questi è stato analizzato allo stesso modo: sono state fornite al modello le features di ogni evento in modo che esso facesse una stima della loro energia. Nel caso ideale si sarebbero dovuti trovare valori uguali di energia totale E_0 del positrone per tutti gli eventi in ogni file; con il modello sviluppato si vede che le stime si dispongono seguendo una distribuzione gaussiana centrata sul valore nominale di energia corrispondente al file analizzato. Si riporta un esempio in figura 4 per il file con $E_k = 5$ MeV:

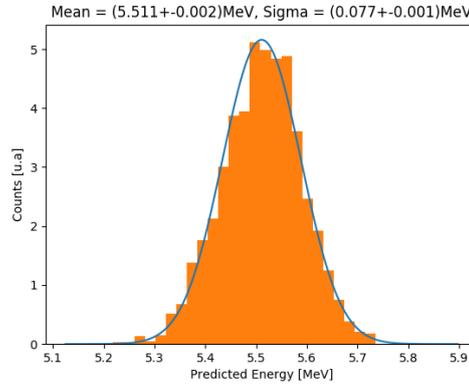


Figura 4: Previsione di E_0 del file con $E_k = 5$ MeV

Eseguendo un fit gaussiano è stato quindi possibile valutare la precisione e l'accuratezza del modello, rispettivamente rappresentate dalla sigma e dalla differenza tra media del fit e valore nominale dell'energia.

Eseguendo la stessa analisi su tutti i dataset contenenti eventi ad energia costante e confrontando i risultati si ottengono i seguenti plot, dove si è preferito utilizzare l'energia totale

trasportata dai fotoni e rivelata dai PMT (uguale a $E_{vis} = E_0 + 0.511$ MeV, dove il secondo termine è dovuto all'energia dell'elettrone in quiete che annichila col positrone di energia E_0):

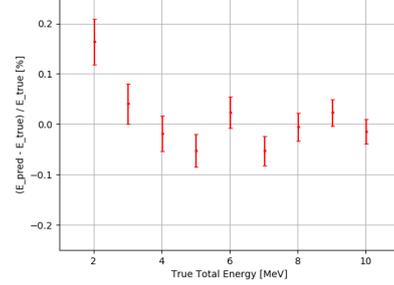
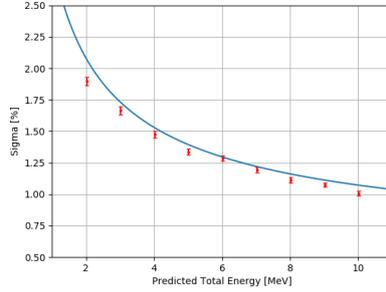


Figura 5: Sigma percentuali ottenute per ogni E_0 Figura 6: Differenza percentuale media tra valore previsto, sovrainposte all'andamento atteso

Si consideri inizialmente la figura 5: essa raccoglie le sigma percentuali, ottenute come rapporto tra sigma e media ricavati dal fit gaussiano, per ogni energia. Nel plot è presente anche una curva di colore blu che rappresenta l'andamento atteso delle sigma percentuali in funzione dell'energia [1], ed è definita come:

$$\frac{\sigma_{E_{vis}}}{E_{vis}} = \sqrt{\left(\frac{2.821}{\sqrt{E_{vis}}}\right)^2 + 0.5947^2 + \left(\frac{0.0}{E_{vis}}\right)^2}$$

La forma generica della funzione è stata ricavata considerando, oltre alla fluttuazione statistica presente nel caso ideale, tutti i disturbi presenti nell'esperimento reale, come dark noise dovuto ai PMT e alla strumentazione, uniformità imperfetta del rivelatore e risoluzione finita dei PMT sulla carica raccolta. I valori inseriti all'interno della funzione invece sono stati ottenuti attraverso una parametrizzazione fatta utilizzando modelli che non facevano uso di algoritmi basati su machine learning.

Dal grafico, si vede che i valori delle sigma percentuali ricavati dai fit gaussiani seguono un andamento molto simile a quello previsto. È lecito quindi pensare che il modello sviluppato funzioni in maniera ottimale su tutto lo spettro energetico.

In figura 6 viene riportata una stima dell'errore sistematico del modello per diversi valori di energia, calcolato come differenza percentuale tra valore medio delle energie previste dal modello e quello nominale degli eventi contenuti in ogni file, cioè come $\frac{E_{vis}^{pred} - E_{vis}^{true}}{E_{vis}^{true}}$. Dal grafico

si nota che quasi tutti i valori medi di E_0^{pred} sono molto vicini a quelli nominali, quindi il modello risulta accurato per la maggior parte di energie.

Dai due grafici mostrati nelle figure sopra sono stati rimossi i punti corrispondenti agli eventi con $E_k = 0$ MeV e $E_k = 10$ MeV in quanto, essendo uguali agli estremi dell'intervallo energetico su cui è stato allenato il modello, non venivano ricostruiti in modo fedele.

2.4 Ricostruzione Vertice

Il secondo modello per la ricostruzione dell'evento prevede la stima delle tre componenti del vertice. Come prima, si dovrà eseguire una regressione, ma in questo caso il modello sarà fornito di tre label. Inoltre, si sono mantenute le features dei modelli precedenti, ma sono state aggiunte anche le variabili X_{CoH} e Y_{CoH} . Nonostante la struttura interna della rete non sia stata cambiata (è sempre formata da 5 hidden layer da 128 nodi), è stato possibile allenare il modello per più tempo senza osservare fenomeni di overfitting.

La scelta della loss function tra MSE e MAPE non è stata però immediata: teoricamente parlando, ci si aspetta che utilizzando il MAPE si arrivi ad ottenere un errore percentuale delle tre componenti x, y e z circa costante per tutti i loro valori; utilizzando il MSE invece vengono considerati solo i quadrati dei valori assoluti dei loro errori, quindi ci si aspetta che l'errore percentuale diminuisca al crescere delle tre componenti. Sono stati costruiti due grafici per verificare la dipendenza tra l'errore percentuale del raggio stimato (considerando $R_{pred} = \sqrt{x_{pred}^2 + y_{pred}^2 + z_{pred}^2}$) e il valor vero del raggio per le due loss function utilizzate. Si riportano i due grafici trovati usando il set di dati con $E_k = 7$ MeV:

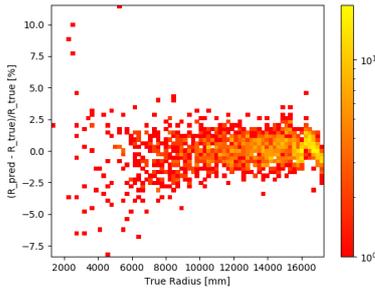


Figura 8: Errore percentuale in funzione del raggio usando il MSE come loss function

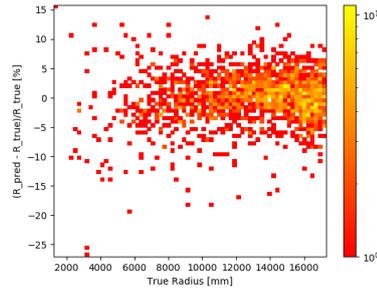


Figura 9: Errore percentuale in funzione del raggio usando il MAPE come loss function

Complessivamente si nota che gli errori percentuali sulla stima del raggio seguono un andamento simile a quello descritto sopra. Dal momento che gli eventi vengono simulati in modo uniforme su tutto il volume sferico, la maggior parte degli eventi avviene a distanze abbastanza elevate dal centro del rivelatore quindi, anche se i due modelli sembrano fornire errori simili nella stima dei vertici vicini al centro, si è preferito utilizzare il MSE durante lo sviluppo e l'ottimizzazione del modello.

L'allenamento si è svolto anche in questo caso utilizzando il file contenente un milione di eventi, valutando poi il modello ottenuto utilizzando i vari set di dati con eventi ad energia costante. A differenza dell'energia, le grandezze da stimare ora non sono costanti all'interno dei set, perciò le stime non si distribuiranno seguendo una curva gaussiana. Per verificare precisione e accuratezza del modello, si sono andati a studiare gli scarti delle stime, cioè le componenti di $x_{pred} - x_{true}$. Si riportano in figura 10, 11 e 12 gli scarti delle stime delle tre componenti del vertice per il set di dati con $E_k = 7$ MeV:

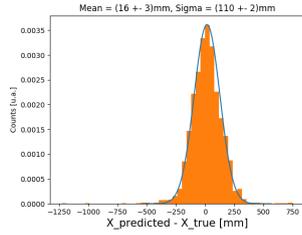


Figura 10: Residui $x_{pred} - x_{true}$ per $E_k = 7$ MeV

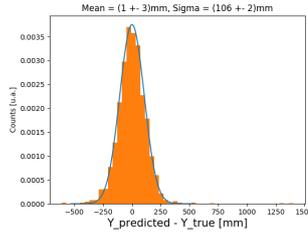


Figura 11: Residui $y_{pred} - y_{true}$ per $E_k = 7$ MeV

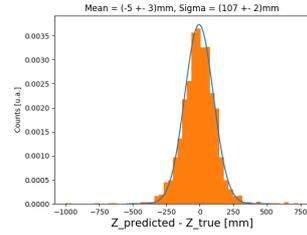


Figura 12: Residui $z_{pred} - z_{true}$ per $E_k = 7$ MeV

Come già fatto per la ricostruzione dell'energia, si è svolto un fit gaussiano usando i dati raccolti negli istogrammi, ma in questo caso non viene utilizzata la sigma come valore percentuale rispetto alla media del fit in quanto ci si aspetta che questa sia uguale a zero per gli scarti. Analizzando i set di dati e confrontando i risultati di ogni fit, si ottengono i seguenti plot:

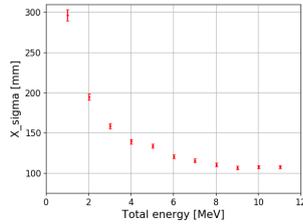


Figura 13: Sigma ottenute dai fit di $x_{pred} - x_{true}$ ad ogni energia

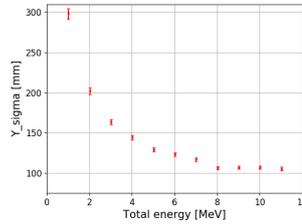


Figura 14: Sigma ottenute dai fit di $y_{pred} - y_{true}$ ad ogni energia

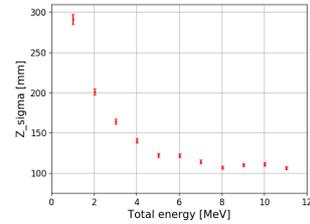


Figura 15: Sigma ottenute dai fit di $z_{pred} - z_{true}$ per ogni energia

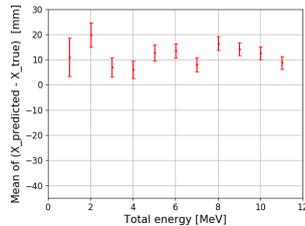


Figura 16: Medie ottenute dai fit di $x_{pred} - x_{true}$ per ogni energia

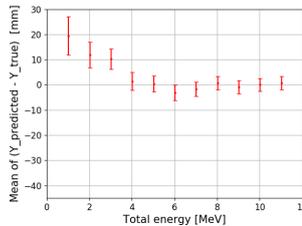


Figura 17: Medie ottenute dai fit di $y_{pred} - y_{true}$ per ogni energia

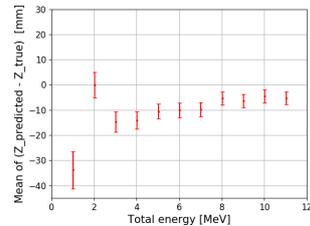


Figura 18: Medie ottenute dai fit di $z_{pred} - z_{true}$ per ogni energia

Si prendano in considerazione i risultati appena mostrati: si nota subito che in tutti i casi analizzati le sigma seguono un andamento simile a quello ottenuto durante la ricostruzione delle energie, quindi anche questo modello risulta più preciso per energie elevate. Per quanto riguarda la differenza media tra valore stimato e valore vero per le tre componenti del vertice, si nota che è presente un leggero bias: per tutti i valori di energia x_{pred} viene in media sovrastimata di circa 1cm, z_{pred} viene sottostimata in media della stessa quantità, invece per y_{pred} è presente un bias ancora minore. Questo errore sistematico del modello non rappresenta però un problema date le notevoli dimensioni del rivelatore.

Utilizzando le tre componenti del vertice, per ogni evento è stato possibile ricostruire una stima della sua distanza dal centro del rivelatore come $R_{pred} = \sqrt{x_{pred}^2 + y_{pred}^2 + z_{pred}^2}$. Anche in questo caso sono stati studiati gli scarti dal suo valore vero, cioè $R_{pred} - R_{true}$, e sono stati analizzati seguendo la procedura utilizzata per le tre componenti del vertice. I risultati ottenuti sono mostrati nelle figure 19 e 20:

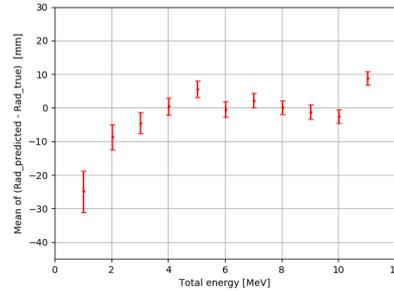
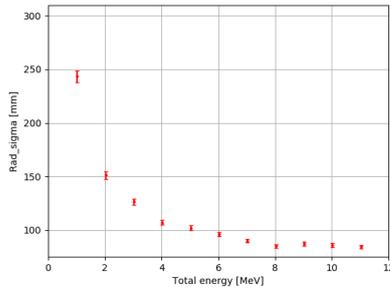


Figura 19: Sigma ottenute dai fit di $R_{pred} - R_{true}$ per ogni energia

Figura 20: Medie ottenute dai fit di $R_{pred} - R_{true}$ per ogni energia

Si può notare immediatamente che i grafici ottenuti per R seguono un andamento simile a quelli delle tre componenti del vertice, ma i risultati dei fit sono migliori rispetto a prima in quanto le componenti sovrastimate e quelle sottostimate tendono a compensarsi. Infatti i valori delle sigma risultano più piccoli per ogni energia e arrivano ad essere minori di 10 centimetri per una buona parte di esse, inoltre i valori medi degli scarti non risultano essere sistematicamente sovrastimati o sottostimati su tutto lo spettro delle energie analizzate. È giusto osservare che gli eventi con $E_k = 0$ MeV e $E_k = 10$ MeV sono caratterizzati dagli scarti medi maggiori, ma rimanendo comunque sotto 1.5 cm non rappresentano un problema.

Cercando di riassumere i risultati ottenuti, è stato costruito un ulteriore grafico per mostrare quanto vale la distanza media tra il vertice di un evento e il vertice previsto dal modello (cioè $Dist = |x_{pred} - x_{true}|$) per ogni energia.

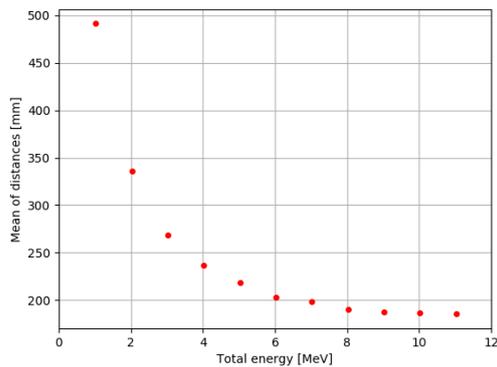


Figura 21: Distanze medie tra vertice vero e previsto per ogni energia

Come previsto, l'andamento delle distanze medie tra i vertici è molto simile a quello delle sigma ottenute dai fit gaussiani. Ovviamente le distanze assumono valori maggiori, ma si può dire che complessivamente il modello funziona correttamente su tutto lo spettro energetico analizzato (a differenza del modello per la ricostruzione che presentava anomalie per gli eventi con $E_k = 0$ MeV o $E_k = 10$ MeV).

Capitolo 3

Risposta dei Modelli al Dark Noise

Il dark noise è un tipo di rumore generato per effetti termici all'interno dei PMT [8]. In particolare, esso è causato da elettroni che vengono generati indipendentemente dai fotoni che colpiscono il PMT. Questo causa una corrente anche in assenza di luce e, più in generale, un rumore sul segnale acquisito, rappresentato come numero di fotoelettroni raccolti in funzione del tempo.

I dati sono stati simulati come nel caso precedente, ma si è proceduto ad aggiungere manualmente i fotoelettroni raccolti a causa del dark noise. Dal momento che non tutti i PMT da 20 pollici sono uguali (1/4 dei PMT è prodotto dalla Hamamatsu e 3/4 dalla NNVT), il disturbo è stato inserito in maniera diversa: per i primi si sono aggiunti fotoelettroni con una frequenza di 16 Hz, per i secondi è stata usata una frequenza di 50 Hz. In figura 22 si mostra una simulazione della corrente generata da un evento (rappresentata dal numero di fotoelettroni raccolti in un unità di tempo) in presenza di dark noise:

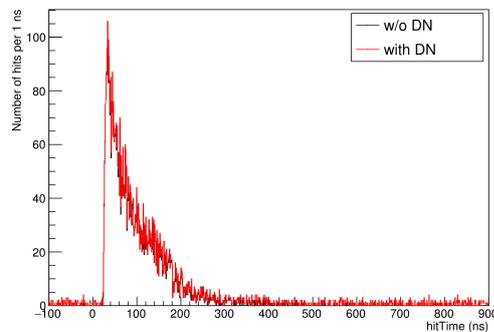


Figura 22: Numero di fotoelettroni raccolti in 1ns in funzione del tempo in presenza di dark noise

Per valutare l'effetto di questo disturbo sulla ricostruzione dell'energia e del vertice è stato prodotto, sempre attraverso simulazioni Montecarlo, un nuovo dataset contenente un milione di eventi affetti da dark noise e altri 11 contenenti gli eventi con E_k fissata. Sono stati sviluppati altri tre modelli utilizzando questi ultimi file: la struttura delle reti è rimasta uguale a prima e l'allenamento è stato svolto allo stesso modo, ma utilizzando i nuovi dati affetti da

dark noise. Si è studiato il funzionamento dei nuovi modelli per ogni energia usando la stessa procedura di prima utilizzando i nuovi 11 set di dati con E_k costante.

Si confrontano ora i risultati della ricostruzione dell'energia e del vertice utilizzando i modelli originali e quelli che utilizzano dati affetti da dark noise, rappresentati nei grafici come punti rispettivamente di colore rosso e blu:

- **Ricostruzione Energia:**

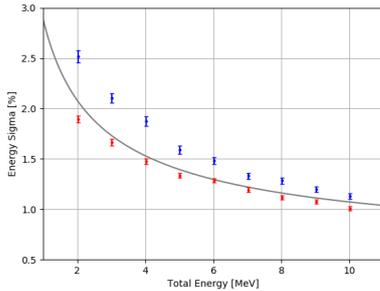


Figura 23: Confronto sigma dei fit sulle energie E_0 ottenute in presenza (blu) e in assenza (rosso) di dark noise

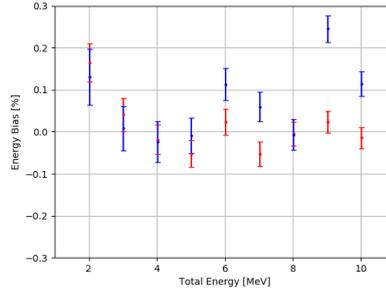


Figura 24: Confronto delle differenze percentuali tra valore medio stimato di E_0 e quello nominale ottenute in presenza (blu) e in assenza (rosso) di dark noise

Si consideri inizialmente il confronto tra le sigma mostrato in figura 23: ricordando che la curva di colore grigio rappresenta l'andamento atteso delle sigma (definito nella sezione 2.3), si vede che l'introduzione del disturbo causa un leggero aumento delle sigma, che comunque continuano a seguire un andamento simile a quello atteso.

In figura 24 vengono confrontati invece i bias dei due modelli per l'energia: complessivamente non si notano grandi differenze.

Anche in questo caso i punti con $E_k = 0$ MeV e $E_k = 10$ MeV sono stati rimossi per lo stesso motivo spiegato alla fine del capitolo 2.3. L'introduzione del disturbo nella ricostruzione dell'energia sembra quindi non influenzare troppo le previsioni del modello, che rimangono simili a quelle ottenute in precedenza.

- **Ricostruzione Vertice:**

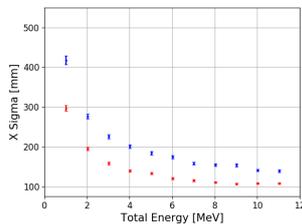


Figura 25: Confronto sigma del fit di $x_{pred} - x_{true}$ in presenza (blu) e in assenza (rosso) di dark noise

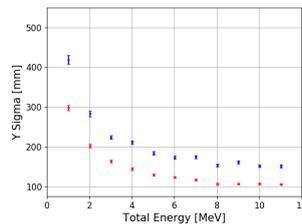


Figura 26: Confronto sigma del fit di $y_{pred} - y_{true}$ in presenza (blu) e in assenza (rosso) di dark noise

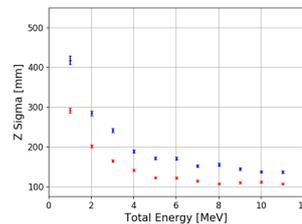


Figura 27: Confronto sigma del fit di $z_{pred} - z_{true}$ in presenza (blu) e in assenza (rosso) di dark noise

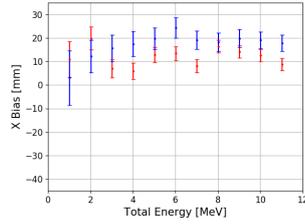


Figura 28: Confronto medie del fit di $x_{pred} - x_{true}$ in presenza (blu) e in assenza (rosso) di dark noise

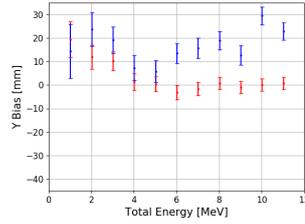


Figura 29: Confronto medie del fit di $y_{pred} - y_{true}$ in presenza (blu) e in assenza (rosso) di dark noise

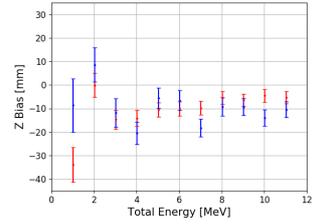


Figura 30: Confronto medie del fit di $z_{pred} - z_{true}$ in presenza (blu) e in assenza (rosso) di dark noise

Nei sei grafici mostrati sopra si esegue un confronto delle sigma e dei bias dei due modelli per la ricostruzione delle tre componenti del vertice. Si nota subito che, come per la ricostruzione delle energie, l'aggiunta del dark noise ai dati aumenta le sigma per tutte e tre le componenti, questa volta in maniera più consistente. Questo aumento non sembra dipendere eccessivamente dall'energia, quindi in presenza di dark noise le sigma seguono un andamento simile a quello ottenuto usando il modello originale poichè, a parte le basse energie, l'aumento delle sigma sembra essere costante per tutti i set di dati analizzati. Per quanto riguarda il bias (ottenuto come scarto medio tra valore stimato e valore vero delle componenti), il modello su cui agisce il dark noise tende a sovrastimare ulteriormente i valori delle componenti x e y del vertice rispetto al modello originale, rimanendo comunque su valori accettabili (al massimo 3 cm). Non si nota però una differenza sostanziale lungo la componente verticale z.

È stato eseguito anche un confronto usando le stime di R ottenute dai due modelli per cercare di riassumere gli effetti del dark noise osservati sulle tre componenti utilizzando una singola grandezza. in figura 31 e 32 si riporta un confronto tra le sigma e gli scarti medi di $R_{pred} - R_{true}$ ottenuti con i due modelli:

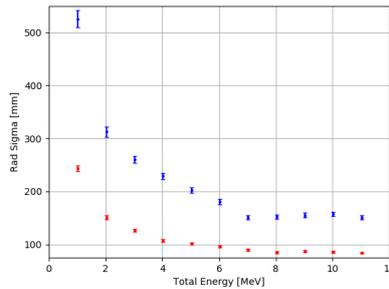


Figura 31: Confronto sigma ottenute dai fit di $R_{pred} - R_{true}$ in presenza (blu) e in assenza (rosso) di dark noise

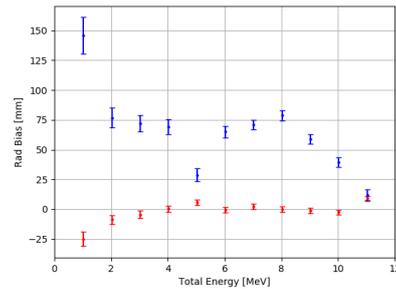


Figura 32: Confronto medie ottenute dai fit di $R_{pred} - R_{true}$ in presenza (blu) e in assenza (rosso) di dark noise

Se per il modello originale i valori delle sigma di $R_{pred} - R_{true}$ erano minori di quelli delle singole componenti, in questo caso vediamo che con lo stesso procedimento le sigma ottenute sono leggermente maggiori. Questo è dovuto all'aumento delle sigma delle singo-

le componenti, infatti calcolando $R_{pred} = \sqrt{x_{pred}^2 + y_{pred}^2 + z_{pred}^2}$ le sovrastime e sottostime delle tre componenti non si compensano più in modo da ottenere un valore preciso del raggio a causa della maggiore incertezza su di esse.

Per quanto riguarda il bias, si nota che l'ulteriore sovrastima delle componenti x e y si traduce in un errore sistematico di R_{pred} quasi costante su buona parte delle energie.

Oltre a questo, è presente anche un fenomeno particolare: se il modello originale tendeva a sottostimare R_{pred} a basse energie, ora si osserva un comportamento opposto. Per capire la causa di questo fenomeno sarebbe necessario approfondire lo studio di questi nuovi modelli allenati con dati affetti da dark noise, ma ciò uscirebbe dagli scopi di questo lavoro, che si è concentrato sullo sviluppo dei modelli utilizzando dati non disturbati da questo rumore.

Capitolo 4

Conclusioni

In JUNO, per poter ricostruire l'energia e il vertice degli eventi IBD (Inverse Beta Decay) si è ricorso all'utilizzo di algoritmi di machine learning. In particolare, per questo lavoro è stato scelto l'utilizzo delle reti neurali profonde (DNN), una delle strutture più comuni di Deep Learning. Basandosi su di esse, sono stati sviluppati 3 modelli: uno per la ricostruzione dell'energia, uno per la ricostruzione del vertice e un terzo che viene utilizzato prima di questi e che si occupa di discriminare gli eventi esterni al volume di fiducia del rivelatore.

Le sigma ottenute attraverso modello per la ricostruzione dell'energia del positrone seguono i valori ricavati dagli ultimi metodi non basati su algoritmi di machine learning, quindi è lecito dire che il modello funziona correttamente. In particolare, si nota che per ogni energia i valori delle sigma sono minori di $3\%/\sqrt{E(\text{MeV})}$, risoluzione energetica necessaria per studiare la gerarchia di massa dei neutrini.

Per quanto riguarda la ricostruzione del vertice, i valori delle sigma su ogni componente minore di 15 cm per quasi tutte le energie analizzate. Se però si utilizzano le tre stime per calcolare la componente radiale di ogni evento, la sigma di quest'ultima diventa inferiore a 10 cm per diversi valori di energia. Nonostante questi risultati siano accettabili, è possibile che con ulteriore lavoro si possa rendere la ricostruzione del vertice ancora più precisa.

Introducendo il dark noise, si nota un aumento delle incertezze su tutte le stime, ma complessivamente i risultati non si discostano troppo da quelli ottenuti in assenza di questo rumore, in particolare vediamo che nella ricostruzione dell'energia i valori delle sigma sono ancora inferiori a $3\%/\sqrt{E(\text{MeV})}$.

Si può quindi dire che l'utilizzo dei DNN per lo sviluppo dei modelli sia una scelta valida in quanto le previsioni ottenute risultano soddisfacenti; in particolare, dal momento che i modelli sembrano lavorare correttamente anche sotto l'influenza di dark noise, è lecito aspettarsi che questi possano funzionare anche in presenza di altri disturbi. Per verificare quest'ultima ipotesi potrebbe rendersi necessario un ulteriore sviluppo del lavoro in modo da capire come adattare e migliorare i modelli per lavorare con dati sempre più simili a quelli reali.

Bibliografia

- [1] F. An et al, *Neutrino Physics with JUNO*, J. Phys. G43(2016)030401, [arXiv/1507.05613].
- [2] T. Adam et al, *JUNO Conceptual Design Report*, [arXiv/1508.07166].
- [3] Qin Liu, Miao He, Xuefeng Ding, Weidong Li, Haiping Peng. *A vertex reconstruction algorithm in the central detector of JUNO*, 2018, [arXiv:1803.09394].
- [4] By Shai Shalev-Shwartz, Shai Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014.
- [5] By Charu C. Aggarwal, *Neural Networks and Deep Learning*, Springer, September 2018.
- [6] TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [7] François Chollet, Keras, 2015, GitHub repository, <https://github.com/keras-team/kerass>
- [8] A.G.Wright, *The Photomultiplier handbook*, Oxford Univ. Press, 2017