

UNIVERSITÀ DI PADOVA

DIPARTIMENTO DI FISICA E ASTRONOMIA "G. GALILEI"

Laurea Triennale in Fisica

## Ricostruzione di collisioni protone-protone con CMS con calcolo neuromorfico

Relatrice:

**Dott.ssa Mia Tosi**

Candidato:

**Emanuele Coradin**

Correlatore:

**Dott. Tommaso Dorigo**

ANNO ACCADEMICO 2022/2023

## Abstract

L'esperimento CMS al Large Hadron Collider ricostruisce con algoritmi di *pattern recognition* basato su calcolo convenzionale le tracce di migliaia di particelle cariche che attraversano i suoi rivelatori ogni 25 ns. La ricostruzione e identificazione delle traiettorie delle particelle cariche è un problema particolarmente complesso dal punto di vista computazionale, a causa della grande quantità di dati che devono essere elaborati e dall'elevato numero di combinazioni. Il problema diventerà ancora più difficile da gestire nella fase HL-LHC, in cui è previsto un aumento delle collisioni di circa un fattore tre rispetto alle attuali condizioni sperimentali. In questo lavoro si considererà un modello di calcolo neuromorfico basato su *spiking neural networks* (SNN) per complementare la *pattern recognition* di CMS e identificare tipologie di segnali non normalmente identificati dagli attuali algoritmi in uso. L'efficacia dell'apprendimento non supervisionato delle SNN verrà quantificata in eventi generati tramite simulazioni Monte Carlo.

# Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
1.1	LHC . . . . .	4
1.2	CMS . . . . .	6
1.3	Il Tracciatore . . . . .	7
1.4	Trigger e ricostruzione delle tracce . . . . .	8
<b>2</b>	<b>Il nostro modello per il riconoscimento delle tracce</b>	<b>10</b>
2.1	Modello spiking neural network . . . . .	10
2.2	Primo modello semplificato . . . . .	12
2.2.1	Definizione della rete e del protocollo di apprendimento . . . . .	13
2.2.2	Definizione delle metriche . . . . .	14
2.2.3	Risultati del modello . . . . .	15
2.3	Dalla geometria planare alla geometria cilindrica . . . . .	17
2.3.1	Problema di bordo . . . . .	18
2.4	Gestione della geometria 3D . . . . .	19
2.5	Dati disponibili dalle simulazioni Monte Carlo . . . . .	20
2.6	Generazione del rumore . . . . .	21
2.7	Modifica della rete con taglio pseudocasuale di connessioni . . . . .	22
2.8	Processo di apprendimento e ricerca a griglia . . . . .	22
2.8.1	Hardware e tempi di esecuzione . . . . .	28
<b>3</b>	<b>Conclusioni</b>	<b>29</b>

# Capitolo 1

## Introduzione

### 1.1 LHC

Il "Large Hadron Collider", LHC, è il più grande e potente acceleratore di particelle al mondo attualmente in funzione. Situato presso il CERN (Organizzazione Europea per la Ricerca Nucleare) a Ginevra, in Svizzera, presenta due anelli di 27 chilometri di circonferenza, posti all'interno di un tunnel sotterraneo a circa 100 metri di profondità[5].

Grazie a un sistema di acceleratori, come mostrato in fig. 1.1, è in grado di accelerare due fasci di protoni e ioni pesanti lungo direzioni opposte per poi farli collidere. I fasci, direzionati tramite magneti, non sono continui, bensì sono suddivisi in pacchetti (detti *bunch*), fino a un numero massimo di 2500 circa, separati a intervalli regolari di 25 ns, che si traduce in una frequenza di incrocio pari a 40 MHz. Ciascun *bunch* contiene un elevato numero di protoni (dell'ordine di  $10^{11}$ )[3].

Il risultato di un incrocio tra *bunch* che dia luogo ad almeno una collisione viene detto evento. L'elevata densità di protoni nei *bunch* da origine a collisioni multiple (denominato *pileup*) nello stesso incrocio tra fasci.

Lungo il percorso nei punti di interazione sono situati i quattro rivelatori principali: ATLAS, CMS, LHCb, ALICE[5].

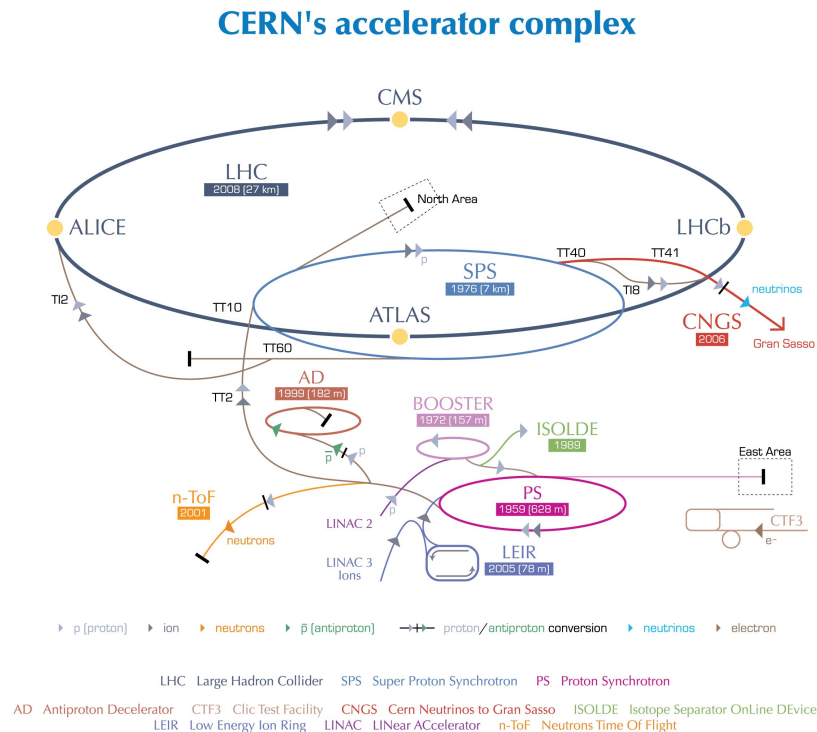


Figura 1.1: Diagramma con gli acceleratori e gli esperimenti di LHC.[9]

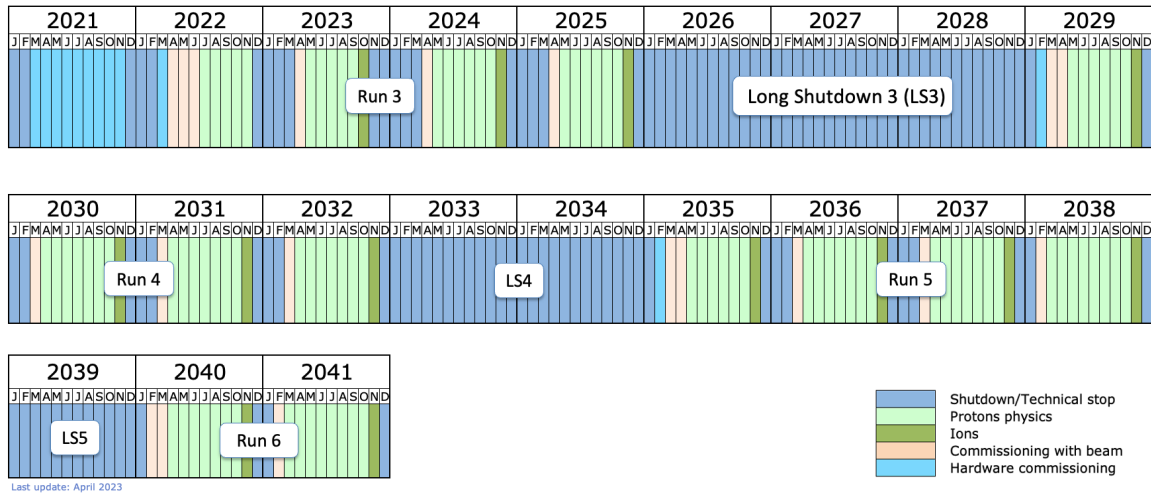


Figura 1.2: Pianificazione di lungo termine delle fasi di LHC.<sup>1</sup>

Come schematizzato in fig. 1.2, attualmente si sta svolgendo il terzo periodo di presa dati, Run3, in cui LHC fornisce collisioni a un'energia nel centro di massa di  $\sqrt{s} = 13.6 \text{ TeV}$ , con un picco di luminosità pari a  $2.1 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ , per una luminosità integrata attesa di  $300 \text{ fb}^{-1}$ [6]. La conclusione di Run3 è prevista per la fine del 2025, con l'inizio di "Long Shutdown 3" (LS3). Durante questo periodo verranno introdotti dei significativi upgrade per entrare nella Fase2, il cui inizio è previsto nel 2029, che segnerà il via dell'era ad "Alta luminosità" di LHC: HL-LHC (vedi fig. 1.2).

L'obiettivo è potenziare l'acceleratore per raggiungere gradualmente nel corso degli anni un picco di luminosità istantanea fino a  $7.5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ [2]. Il progetto originale di LHC prevedeva configurazioni di ottica e intensità dei fasci per fornire in media 25 eventi di *pileup*, ma sin dal Run2 (2015-2018) LHC ha duplicato questo valore, e in Run3 il valore di picco di *pileup* atteso è pari a 62, mentre nella fase HL-LHC si prevedono valori in media pari a 200[2]. Ciò comporta una sfida senza precedenti sia nello studio delle tracce che nella gestione del danno da radiazione dei rivelatori.

La ragione di questi sforzi è la possibilità di studiare eventi rari previsti dal Modello Standard, tra cui l'auto accoppiamento del bosone di Higgs, il suo potenziale, le sue costanti di accoppiamento, etc; contemporaneamente si cercano eventuali fenomeni "Oltre il Modello Standard", che porterebbero quindi alla scoperta di nuova fisica.

<sup>1</sup>Disponibile al link: "<http://lhc-commissioning.web.cern.ch/schedule/LHC-long-term.htm>"

## 1.2 CMS

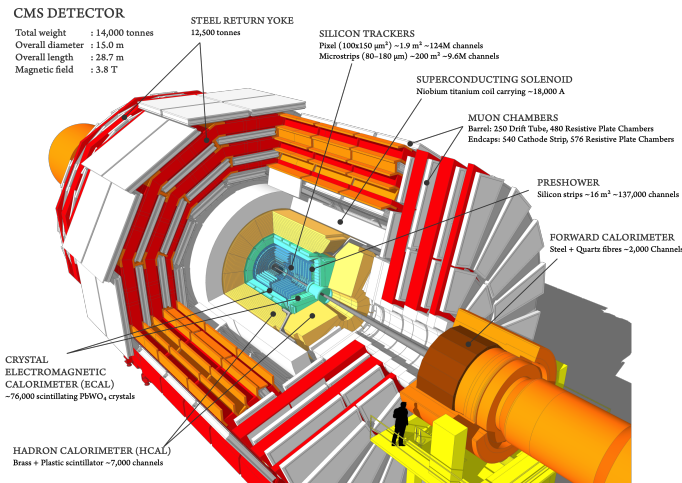


Figura 1.3: *Visione in prospettiva del detector dopo gli aggiornamenti di Fase1[12].*

volume di  $400 \text{ m}^3$ .

Si divide in una parte centrale, detta *Barrel*, e due zone esterne chiamate *Endcap*. Seguendo la direzione radiale si riconoscono diversi strati:

**Punto di interazione** Situato al centro del rivelatore, è la zona dove avviene la collisione tra i due fasci.

**Tracciatore** Composto interamente da rivelatori al silicio, presenta negli strati più interni dei moduli a pixel, mentre in quelli più esterni dei rivelatori a strips. Serve a individuare il percorso seguito dalle particelle cariche e misurarne il momento.

**Calorimetro elettromagnetico (ECAL)** Conta 76000 cristalli scintillatori di tungstato di piombo ( $PbWO_4$ ), assorbe completamente fotoni ed elettroni rilasciando una luce di scintillazione proporzionale all'energia, permettendone la misurazione.

**Calorimetro adronico (HCAL)** Formato da strati di materiale denso come l'ottone intervallati da sottili strati di una particolare plastica scintillante, consente la misura dell'energia dei jet adronici.

**Il solenoide** Costruito con fibre superconduttive, caratterizzato da un'induttanza pari a  $L = 14 \text{ H}$ , vi scorre una corrente di circa  $18000 \text{ A}$ .

**Le camere per muoni** Tracciano il passaggio dei muoni e sono poste all'interno di una struttura di ferro magnetizzato atto a chiudere le linee del campo magnetico.

Nel prosieguo della trattazione si presteranno maggiori dettagli sul tracciatore.

Il sistema di coordinate adottato presenta l'origine nel punto di interazione, l'asse delle  $y$  punta verticalmente verso l'alto mentre quello delle  $x$  è diretto radialmente verso il centro di LHC. In questo modo l'asse  $z$  è allineato alla direzione dei fasci. Si definisce quindi il piano trasverso  $x - y$ , con coordinata radiale  $r$  e angolo azimutale  $\phi \in [-\pi, \pi]$ , misurato a partire dall'asse  $x$ . Chiamato  $\theta$  l'angolo polare misurato a partire dall'asse  $z$ , si definisce la pseudorapidità  $\eta = -\ln \tan(\frac{\theta}{2})$ . Si indica infine con  $p_T$  il momento delle particelle sul piano trasverso[1].

CMS sta per "Compact Muon Solenoid" e, insieme ad ATLAS, è un rivelatore disegnato per studiare un vasto spettro di fisica. È stato ideato per la scoperta e la misura delle proprietà del bosone di Higgs, la ricerca di nuove particelle con massa elevata superiore a  $1 \text{ TeV}$ , la ricerca della materia oscura, lo studio delle interazioni standard a energie dell'ordine del  $\text{TeV}$  e la misura precisa dei vertici secondari di decadimento.

La struttura è cilindrica, con un peso totale di  $12500 \text{ t}$ , un diametro di  $15 \text{ m}$  e una lunghezza di  $21.6 \text{ m}$ , ospita al suo interno un solenoide superconduttivo in grado di produrre un campo magnetico di  $3.8 \text{ T}$  per un

## 1.3 Il Tracciatore

Il tracciatore è il sottosistema di rivelazione più interno. La zona più vicina al punto di interazione è caratterizzata da un'elevata densità di tracce e di radiazione, ed è costituita da dei rivelatori a pixel in grado di fornire una informazione tridimensionale.

Il passaggio di una particella carica provoca la formazione locale di un accumulo di carica detto *cluster*, a cui successivamente viene associata una posizione nello spazio tridimensionale chiamata *hit*.

Si contano quattro strati nel *Barrel* e tre dischi nell'*Endcap*. La disposizione è stata ottimizzata per coprire una regione con pseudorapidità fino a  $|\eta| = 3$ , migliorare la *pattern recognition* e la ricostruzione delle tracce.

Per  $r > 20$  cm invece sono presenti certo numero di strati (10 nel *Barrel*) equipaggiati con un sistema di sensori a strips. Essi possono essere "single sided", se lo strato sensibile è posto da un solo lato del rivelatore, oppure "double sided", se entrambi i lati del modulo sono sensibili[4].

Il sistema si sta gradualmente degradando a causa del danno da radiazione e non sarebbe in grado di operare a lungo durante HL-LHC.

In vista di Fase2 verrà interamente sostituito con un tracciatore costituito da due parti:

**Inner Tracker** Formato da quattro strati nel *Barrel* e 12 dischi nell'*Endcap*, basati su sensori a pixel;

**Outer Tracker** Formato da 6 layer nel *Barrel* e 5 dischi nell'*Endcap*, basato su sensori a strips e macro pixel.

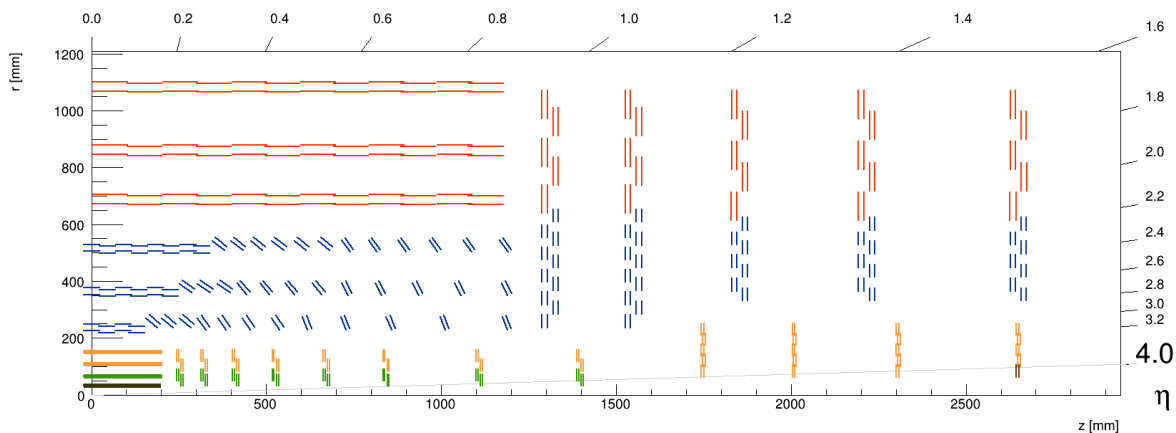


Figura 1.4: Infografica relativa a un quarto del Tracciatore di Fase2 <sup>2</sup>.

Il nuovo sistema di tracciatura è disegnato per aumentare la resistenza alla radiazione, incrementare la granularità mantenendo contenuta l'occupazione per strato in modo da gestire efficacemente livelli di *pileup* in media pari a 140-200 collisioni per evento, a ridurre il materiale presente e ad aumentare la regione coperta fino a  $|\eta| = 4$ .

È inoltre indispensabile migliorare le performance del trigger per aumentare la capacità di selezionare eventi interessanti [1]. In particolare, per ottenere questo risultato, ciascun modulo dell'OT è stato progettato con l'accoppiamento di due rivelatori (pixel e strip, oppure strip e strip) e dotato di un'opportuna elettronica di lettura per poter partecipare al primo livello di trigger (L1) (vedi sez. 1.4).

<sup>2</sup>Disponibile su: [www.cms-tklayout.web.cern.ch/cms-tklayout/layouts/cmssw-models/ZAO\\_T800\\_I\\_T704/info.html](http://www.cms-tklayout.web.cern.ch/cms-tklayout/layouts/cmssw-models/ZAO_T800_I_T704/info.html)

Si riportano nelle tabelle 1.1 e 1.2 le distanze radiali medie dei vari strati del *Barrel* rispetto al centro del rivelatore e l'intervallo di  $z$  coperto.

<b><i>Barile IT</i></b>	<b><i>Pixel</i></b>			
<i>Strato</i>	1	2	3	4
$r[cm]$	3.00	6.15	10.45	14.65
$z_{max}[cm]$	19.79	20.07	20.07	20.07
<i>Numero moduli</i>	108	216	180	252

Tabella 1.1: Distanza radiale media, massimo valore di  $z$  e numero di moduli del *Barrel* di IT di Fase2.

<b><i>Barile OT</i></b>	<b><i>Pixel e strip</i></b>			<b><i>Strip Strip</i></b>		
<i>Strato</i>	1	2	3	4	5	6
$r[cm]$	24.94	37.17	52.27	68.70	86.00	108.30
$z_{max}[cm]$	120.29	120.35	120.19	117.68	117.68	117.68
<i>Numero moduli</i>	558	910	1404	1152	1440	1824

Tabella 1.2: Distanza radiale media, massimo valore di  $z$  e numero di moduli del *Barrel* di OT di Fase2<sup>3</sup>.

## 1.4 Trigger e ricostruzione delle tracce

A causa dell'elevata frequenza di collisione e della quantità di informazione generata da ogni evento (multipli terabytes al secondo), ci si scontra con il problema di aver troppi dati da gestire per essere in grado di salvarli tutti. Per questo viene implementato il Trigger: il suo compito è quello di riuscire a selezionare le informazioni interessanti e, una volta individuate, passarle al sistema di acquisizione dati (DAQ) perché vengano salvate.

È organizzato in due livelli, con diversa complessità computazionale e tempo richiesto:

1. *Trigger di Livello 1 (L1)*;
2. *Trigger di Alto Livello (HLT)*.

L'obiettivo del primo è riuscire a ridurre la frequenza dei dati da 40 *MHz* a  $\sim 110$  *kHz*. Durante Fase2 questo valore verrà aumentato fino a un massimo di 750 *kHz*.

A questo livello si utilizzano più di 500 algoritmi diversi, codificati su hardware appositamente progettato.

Attualmente prende in input i dati provenienti da HCAL, ECAL, dalle camere per muoni e dal monitoraggio del fascio.

<sup>3</sup>Dati reperibili su "[https://cms-tklayout.web.cern.ch/cms-tklayout/layouts/cmssw-models/ZAO\\_T800\\_T704/layoutpixel.html](https://cms-tklayout.web.cern.ch/cms-tklayout/layouts/cmssw-models/ZAO_T800_T704/layoutpixel.html)"



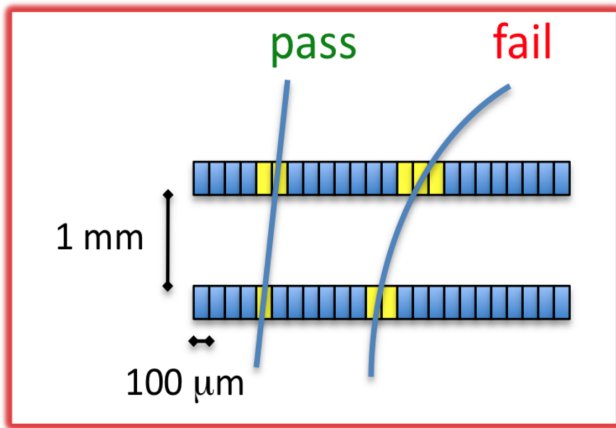


Figura 1.5: *Illustrazione del concetto di stub presa da [8].*

ricostruzione delle tracce.

Questa operazione viene effettuata grazie a un approccio di tipo iterativo, che si compone di quattro passi:

1. **Generazione dei seed:** Si parte da un certo livello di riferimento in base al tipo di traccia cercata e si sfruttano le prime 3/4 hit (*seed*) per ottenere una prima stima dei parametri descrittivi della traiettoria e delle loro incertezze;
2. **Ricerca delle tracce:** si propaga l'informazione ottenuta al passo precedente grazie a specifici algoritmi (in particolare il filtro di Kalman) per trovare nuove hit compatibili e aggiornare le stime dei parametri. In questo passo è permessa al massimo una hit mancante;
3. **Interpolazione delle tracce:** Si sfrutta tutta l'informazione della candidata traccia per effettuare un'interpolazione finale e ottenere così una stima finale dei parametri della traiettoria;
4. **Classificazione delle tracce:** Si valuta la qualità delle tracce trovate in base a vari parametri (tra cui il  $\chi^2$  e algoritmi di machine learning) per eliminare eventuali falsi positivi.

Si parte cercando le tracce più facilmente identificabili (ovvero quelle ad alto  $p_T$ ) e si prosegue man mano risalendo la scala di difficoltà per ricostruire tutti i tipi di traccia. A ogni nuova iterazione vengono nascoste le hit associate a tracce di alta qualità e difficoltà inferiore, in modo tale da ridurre il problema combinatorio.

Per ulteriori informazioni è possibile consultare il testo di riferimento[7].

Nel corso di Fase2 si è progettato di includere in questa fase anche le informazioni provenienti dall'OT. Il passaggio di una particella carica provoca la formazione di una coppia di cluster nei sensori, detta *stub* (vedi fig. 1.5). Questi cluster possono essere più o meno allineati a seconda del  $p_T$  della particella che li ha generati ed è proprio questa l'informazione che verrà sfruttata dal nuovo trigger.

HTL, invece, è progettato per ridurre la frequenza di output a circa 3  $kHz$ . La sua implementazione è puramente software, eseguito in un centro di calcolo con processori commerciali. Attualmente è a questo livello che per la prima volta sono utilizzati anche i dati provenienti dal tracciatore, in particolare per la

# Capitolo 2

## Il nostro modello per il riconoscimento delle tracce

### 2.1 Modello spiking neural network

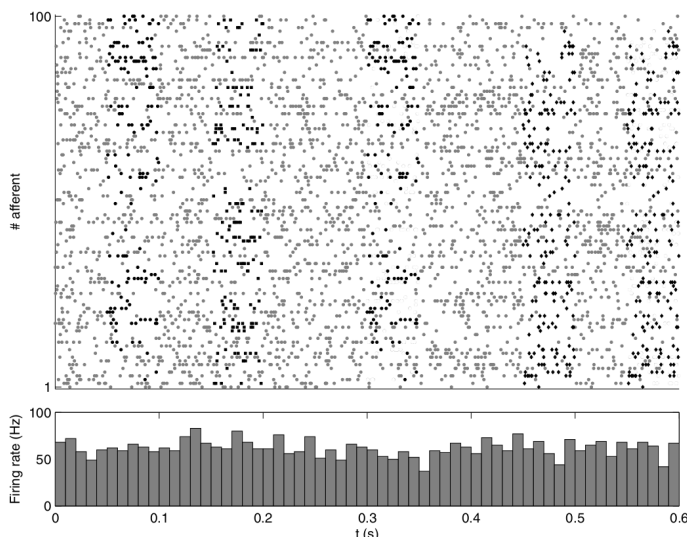


Figura 2.1: *Segnale studiato dal gruppo di Masquelier[10]. Sull'asse delle y viene indicata la fibra afferente responsabile dell'impulso al tempo  $t$  indicato sull'asse delle ascisse. In nero si evidenziano i pattern ripetuti a intervalli casuali sul fondo grigio, costituito da rumore Poissoniano. Il pannello sottostante indica la frequenza media di segnali in ciascun intervallo di 10 ms.*

Esso presenta un rumore poissoniano su cui sono stati inseriti diversi tipi di pattern della durata di 50 ms, che si ripetono in modo casuale nel tempo (fig. 2.1).

Sorprendentemente, grazie ai meccanismi di interazione descritti in seguito, ciascun neurone o sottogruppo di neuroni è stato in grado di auto specializzarsi, emettendo un segnale al riconoscimento di uno specifico pattern.

Il nostro lavoro si basa su una "Spiking neural network" (SNN) costruita su ispirazione di un modello descritto nell'articolo "STDP-Based Spike Pattern Learning" [10].

Questo tipo di reti modella in modo più fedele i meccanismi biologici: l'interazione tra i neuroni è mediata da impulsi elettrici discreti (detti *spikes*) che modificano i valori dei potenziali delle loro membrane.

In particolare, gli autori hanno dimostrato che una SNN equipaggiata con una modellizzazione della "Spike-timing-dependent plasticity" (STDP) e con neuroni del tipo "Leaky integrate-and-fire" (LIF) è in grado di imparare a riconoscere pattern spaziotemporali complessi all'interno di un segnale rumoroso in modo completamente non supervisionato.

L'architettura adottata presenta un certo numero di neuroni tutti connessi tra loro e a 2000 fibre afferenti<sup>1</sup> da cui arrivano gli impulsi di segnale.

---

<sup>1</sup>In anatomia si definiscono fibre afferenti quelle deputate alla trasmissione degli stimoli sensoriali dagli organi periferici al sistema nervoso centrale.

Nel modello considerato, che cerca di approssimare sistemi biologici nel funzionamento e nelle costanti di tempo dei processi che lo compongono, un impulso  $j$  in arrivo al tempo  $t_j$  a livello pre sinaptico<sup>2</sup> aggiunge al potenziale della membrana un "Excitatory Postsynaptic Potential" (EPS):

$$\epsilon(t - t_j) = K \cdot \left[ \exp\left(-\frac{t - t_j}{\tau_m}\right) - \exp\left(-\frac{t - t_j}{\tau_s}\right) \right] \cdot \theta(t - t_j)^3 \quad (2.1)$$

con  $\tau_m$  il tempo caratteristico della membrana,  $\tau_s$  delle sinapsi, mentre  $K$  una costante moltiplicativa scelta in modo tale da normalizzare a 1 il massimo della funzione. Con  $\theta(x)$  si indica la funzione a gradino di Heaviside.

Superata la propria soglia di attivazione  $T$ , il neurone invia un segnale elettrico. Denotato con  $t_i$  il tempo in cui viene inviato l'ultimo impulso, il potenziale della membrana subisce una modifica modellizzata da:

$$\eta(t - t_i) = T \cdot \left\{ K_1 \cdot \exp\left(-\frac{t - t_i}{\tau_m}\right) - K_2 \cdot \left[ \exp\left(-\frac{t - t_i}{\tau_m}\right) - \exp\left(-\frac{t - t_i}{\tau_s}\right) \right] \right\} \cdot \theta(t - t_i) \quad (2.2)$$

$K_1 < K_2$  sono delle costanti positive che determinano la forma del segnale.

Inoltre se un neurone spara viene inviato ai suoi vicini un segnale di inibizione "Inhibitory Post-Synaptic Potential" (IPSP) al tempo  $t_k$ , causando nella membrana degli stessi una riduzione del potenziale data da:

$$\mu(t - t_k) = -\alpha \cdot T \cdot \epsilon(t - t_k) \quad (2.3)$$

con  $\alpha$  costante moltiplicativa che ne regola l'intensità.

Questo modello viene implementato come metodo biologicamente plausibile per introdurre il meccanismo "un vincitore prende tutto": ogni neurone viene messo in competizione con gli altri per essere l'unico attivo. Lo scopo è quello di indurli a specializzarsi.

Si aggiunge che computazionalmente  $\epsilon$ ,  $\eta$ ,  $\mu$  vengono arrotondati a 0 nel caso in cui  $\frac{t-t_i}{\tau_m} > 7$ . Così, a un dato istante temporale il potenziale della membrana di un neurone è dato da:

$$p(t) = \eta(t - t_i) + \sum_{j|t_j > t_i} w_j \cdot \epsilon(t - t_j) + \sum_{k|t_k > t_i} \mu(t - t_k) \quad (2.4)$$

con  $w_j$  i pesi delle connessioni sinaptiche. Ciò definisce il modello di neurone LIF.

Il modello STDP usato per l'apprendimento è definito nel seguente modo:

$$\Delta w_j = \begin{cases} a^+ \cdot \exp\left(\frac{t_j - t_i}{\tau^+}\right) & \text{se } t_j \leq t_i \Rightarrow \text{Potenziamento sinapsi} \\ -a^- \cdot \exp\left(-\frac{t_j - t_i}{\tau^-}\right) & \text{se } t_j > t_i \Rightarrow \text{Depressione sinapsi} \end{cases} \quad (2.5)$$

$a^+$  e  $a^-$  determinano la massima modifica dei pesi, mentre  $\tau^+$  e  $\tau^-$  sono delle costanti che regolano la lunghezza degli intervalli temporali tra gli impulsi presinaptici e postsinaptici entro i quali si verifica il rafforzamento o il deperimento sinaptico. Queste equazioni sono state usate per interpolare delle misure sperimentali su campioni biologici[11]. Anche in questo caso per  $\left| \frac{t_i - t_j}{\tau^+} \right| > 7$  la funzione viene arrotondata a 0.

Il grande pregio del modello è che vengono effettuati dei calcoli sui potenziali solo quando un neurone riceve o emette impulsi, estremamente più efficiente rispetto alla risoluzione di un'equazione differenziale che descriva la dinamica dei potenziali.

<sup>2</sup>Quindi l'origine del segnale può essere sia una fibra afferente che un altro neurone.

In tabella 2.1 sono riportati tutti i valori numerici delle costanti:

<i>Parametro</i>	<i>Valore</i>	<i>Parametro</i>	<i>Valore</i>
$\alpha$	0.25	$\tau_m$	10.0 ms
$K_1$	2	$\tau_s$	2.5 ms
$K_2$	4	$\tau^+$	16.8 ms
$a^+$	0.03125	$\tau^-$	33.7 ms
$a^-$	0.02656	$T$	550 ua <sup>4</sup>

Tabella 2.1: Parametri del modello del gruppo di Masquelier.

Si noti che  $a^- \cdot \tau^- > a^+ \cdot \tau^+$ , il che implica che il sistema tende spontaneamente all'inattività dei neuroni, che in assenza di un segnale ripetitivo continuano ad abbassare i pesi delle connessioni fino a non raggiungere più la soglia di attivazione[13].

## 2.2 Primo modello semplificato

Visti i successi della rete presentata si desidera indagare se un modello simile non possa essere utilizzato per il riconoscimento e la ricostruzione delle tracce all'interno di un rivelatore di particelle come CMS. Nel caso in cui le prestazioni fossero soddisfacenti, infatti, l'abilità della rete di imparare in modo non supervisionato potrebbe aiutare a individuare dei pattern non rintracciabili con gli attuali algoritmi. Inoltre, vista la loro elevata efficienza energetica, se l'hardware si evolvesse sufficientemente, si potrebbe immaginare di riuscire a integrare fisicamente in prossimità del tracciatore un sistema neuromorfico come complemento ai moderni sistemi di tracciamento.

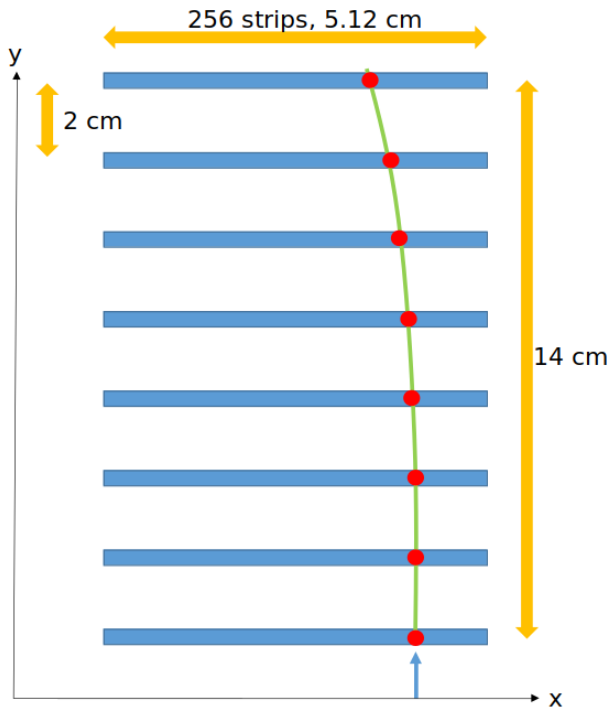


Figura 2.2: Schema del modello semplificato iniziale.

Per iniziare a esplorare questa possibilità si è scritto un codice in C++ basato sul modello semplificato descritto in figura 2.2. Le ipotesi prevedono di lavorare esclusivamente nel piano trasverso, immerso in un campo magnetico di 3.8 T perfettamente ortogonale, in cui sia presente un tracciatore a 8 strati unidimensionali, ortogonali al fascio incidente, equidistanti di 2 cm, lineari e della stessa lunghezza.

A ogni strato viene associato una fibra afferente collegata alla rete neurale. Si suppone inoltre che la densità di rilevatori, detti "strip", sia costante (nel caso specifico 256 strips per una lunghezza di 5.12 cm) e che siano sempre in grado di segnalare il passaggio di una particella (senza falsi negativi).

Vengono quindi simulate diverse classi di particelle, caratterizzate da un  $p_T$  e da una carica elettrica ben definita. Generando uniformemente la posizione iniziale sul primo strato, fissando la direzione iniziale verso le ordinate crescenti e conoscendo il raggio di curvatura del moto definito dalla classe di particella, è possibile calcolare le intersezioni con i vari strati, identificando

quali strips avrebbero registrato un cluster di segnale, detto "hit".

<sup>4</sup>unità arbitrarie.

Non sono previste interazioni tra la particella e il materiale. Per rendere il problema più complesso, si introduce un rumore di fondo costituito da hit generate uniformemente in  $x$  per ogni strato lungo  $y$ .

Il modello prevede di leggere sequenzialmente l'informazione delle strip da sinistra a destra, con una velocità di 1 strip ogni millisecondo per un totale di 256  $ms$  di tempo di lettura, e di poter codificare la presenza di una hit all'interno della strip  $i$  nello strato  $j$  con un impulso nella fibra afferente  $j$  partito al tempo  $t_i = i \cdot 1 \text{ ms}$  con  $i \in [1, 256]$ .

Nella sezione 2.2.1 si presenta il modello di rete implementato.

### 2.2.1 Definizione della rete e del protocollo di apprendimento

La rete neurale adottata è concettualmente molto simile a quella presentata in sezione 2.1.

È organizzata in due strati L0, L1.

I neuroni appartenenti a L1 ricevono i segnali in arrivo da L0 e sia L0 che L1 sono collegati alle fibre afferenti.

Tutti gli  $N_{L0}$  ( $N_{L1}$ ) neuroni di L0 (L1) sono caratterizzati dalla soglia di attivazione T0 (T1).

Un segnale in arrivo al tempo  $t_j$  da un afferente comporta un incremento nel potenziale della membrana neurale dato da  $E_p \cdot \epsilon(t - t_j)$  (vedi eq. 2.1).

$E_p$  è una costante che si configura come un nuovo iperparametro.

Un impulso in arrivo da un altro neurone invece provoca  $\epsilon(t - t_j)$ .

Quando un neurone di L0 supera la propria soglia di attivazione invia un impulso e il suo potenziale diminuisce secondo  $\eta(t - t_i)$  (con  $t_i$  tempo dell'ultimo impulso, eq. 2.1).

Inoltre inibisce esclusivamente gli altri neuroni del suo strato secondo la IPSP (eq. 2.1).

Gli stessi ragionamenti valgono per L1, tranne per l'aggiunta di due nuovi parametri moltiplicativi tali per cui la IPSP diventa:  $\mu_{L1}(t - t_k) = I_{L1} \cdot \mu(D_{L1} \cdot (t - t_k))$ .

Chiamiamo  $I_{L1}$  "Fattore di inibizione di L1" e  $D_{L1}$  "Fattore di dilatazione temporale di L1". Questi parametri vengono introdotti per provare a favorire la specializzazione dei neuroni dello strato L1.

Per l'apprendimento si simulano degli eventi della durata di 256  $ms$ , che possono contenere una e una sola traccia di una particella con del segnale di fondo o solo segnale di fondo con il 50% di probabilità. I tempi sono così lunghi per essere compatibili con i tempi caratteristici del modello in sez. 2.1.

Oltre all'apprendimento della rete tramite STDP, il codice prevede un semplice algoritmo di ottimizzazione degli iperparametri: si effettuano un certo numero  $N_{ep}$  di epoche di apprendimento di  $N_{ev}$  l'una; durante ciascuna epoca si tenta una modifica casuale degli iperparametri e si lascia apprendere la rete tramite STDP.

L'obiettivo è cercare di massimizzare una funzione chiamata Q valore (vedi sez. 2.2.2).

Se nella nuova configurazione provata il Q valore è migliorato allora si effettua un passo verso la direzione trovata nello spazio dei parametri. Altrimenti sono consentiti fino a tre passi verso direzioni non ancora esplorate, prima di tornare alla configurazione migliore e ritentare.

In questo modo, partendo da configurazioni iniziali ragionevoli, si osserva un progressivo aumento del Q valore man mano che si procede nelle epoche, fino a raggiungere un valore di saturazione.

I parametri soggetti a questo processo sono: T0, T1,  $\alpha$ ,  $I_{L1}$ , K, K1, K2,  $E_p$ ,  $D_{L1}$ .

Tutti gli altri non citati rimangono perciò costanti al dato iniziale. All'avvio del programma, T0 e T1 devono essere inizializzati a un valore ragionevole in relazione alla quantità di rumore impostato;  $E_p = 2.5$ ,  $I_{L1} = 1$  e  $D_{L1} = 1$  di default; gli altri parametri comuni con il modello sono stati settati ai valori in tabella 2.1.

Si noti che la rete impara in modo non supervisionato, perciò non è strettamente necessario prevedere fasi di apprendimento e validazione su set di dati distinti.

### 2.2.2 Definizione delle metriche

Considerando L1 come lo strato di uscita, per valutare le prestazioni della rete e monitorarne l'apprendimento si introducono le seguenti definizioni.

Si denota con  $n$  l'indice sugli  $N_{L1}$  neuroni di L1,  $c$  l'indice sulle  $N_{classi}$  classi di particelle (notazione:  $c = 0$  in caso di segnale di fondo),  $\epsilon$  indice sugli eventi,  $\epsilon_c$  indice sugli eventi contenenti una traccia di classe  $c$  fissata.

Si definiscono inoltre delle funzioni indicatrici utili per selezionare rispettivamente gli eventi dove il neurone  $n$  ha sparato almeno una volta,  $F(n, \epsilon)$ , e quelli dove almeno un neurone di L1 ha sparato almeno una volta,  $F_{L1}(\epsilon)$ , in particolare:

$$F(n, \epsilon) = \begin{cases} 1 & \text{se il neurone } n \text{ ha sparato almeno una volta nell'evento } \epsilon \\ 0 & \text{altrimenti} \end{cases} \quad (2.6)$$

$$F_{L1}(\epsilon) = \begin{cases} 1 & \text{se almeno un neurone di L1 ha sparato almeno una volta nell'evento } \epsilon \\ 0 & \text{altrimenti} \end{cases} \quad (2.7)$$

Allo scopo di caratterizzare le prestazioni della rete si definiscono le seguenti metriche:

- **Accettanza**

- Per neurone per classe:  $A_{n,c} = \frac{\sum_{\epsilon_c} F(n, \epsilon_c)}{\sum_{\epsilon_c} 1}$ ;
- Totale di L1 per classe:  $A_{L1,c} = \frac{\sum_{\epsilon_c} F_{L1}(\epsilon_c)}{\sum_{\epsilon_c} 1}$ ;
- Totale media di L1:  $A_{L1} = \frac{\sum_{c \neq 0} A_{L1,c}}{N_{classi}}$ ;

- **Tasso di errore**

- Del neurone:  $E_n = A_{n,0} = \frac{\sum_{\epsilon_0} F(n, \epsilon_0)}{\sum_{\epsilon_0} 1}$ ;
- Totale medio di L1:  $E_{L1} = A_{L1,0} = \frac{\sum_{\epsilon_0} F_{L1}(\epsilon_0)}{\sum_{\epsilon_0} 1}$ ;

- **Selettività di L1:**  $S_{L1} = \frac{\sum_{n,c} A_{n,c} \cdot \log_2 \left( \frac{A_{n,c} + \delta}{A_c \cdot A_n} \right)}{N_{classi} \cdot N_{L1}}$  con  $A_n = \frac{\sum_{c \neq 0} A_{n,c}}{N_{classi}}$ ,  $A_c = \frac{\sum_n A_{n,c}}{N_{L1}}$ ,  $\delta \ll 1$ ;

- **Q valore:**  $Q = Q_0 + \omega \cdot S_{L1}$  con  $Q_0 = \frac{A_{L1}}{E_{L1}}$ ,  $\omega = \omega_0 \cdot \frac{e^{\frac{Q_0}{4}} - 1}{e - 1}$ ,  $\omega_0 = 5$ .

Delle grandezze analoghe possono essere definite per L0, anche se non assumono importanza per il nostro protocollo.

L'accettanza,  $A_{L1,c}$ , fornisce una misura di quante volte un neurone di L1 si è attivato in eventi contenenti una certa classe di segnale rispetto a quante volte gli è stato presentato.

Il tasso di errore,  $E_n$ , quantifica la frazione di eventi in cui un neurone spara pur non essendo presente nessuna traccia (si tratta quindi di un falso positivo).

La selettività,  $S_{L1}$ , è un numero positivo tanto maggiore quanto più alta è la specializzazione dei singoli neuroni nel riconoscere una certa classe di pattern. Si noti che se tutti i neuroni presentano una risposta uniforme, rappresentata da  $A_{n,c} \equiv \bar{A} \forall n, c$ , allora vale  $S_{L1} = 0$ . Assume invece il suo valore massimo se ogni neurone impara a riconoscere sempre ed esclusivamente un solo pattern e vale  $\max(S_{L1}) = \frac{\min(N_{classi}, N_{L1})^2}{N_{classi} \cdot N_{L1}} \cdot \log_2(N_{classi} \cdot N_{L1})$ .

Il Q valore è definito dalla somma di due contributi che valutano la capacità di riconoscere correttamente i pattern proposti ( $Q_0$ ) e la capacità dei neuroni di specializzarsi ( $\omega \cdot S_{L1}$ ). In particolare,  $Q_0$  è definito come il rapporto tra l'accettanza totale,  $A_{L1}$ , e il tasso d'errore totale,  $E_{L1}$ , mentre il secondo termine  $\omega \cdot S_{L1}$  premia una maggiore selettività una volta che  $Q_0 > 4$ .

$\omega_0$  è una costante moltiplicativa che può essere modificata per calibrare l'importanza della selettività, durante le nostre prove è stata mantenuta fissa a 5.

Queste metriche considerano unicamente eventi con un pattern per classe. La scelta è motivata dal fatto che questo lavoro è da considerarsi come il primo passo per caratterizzare la fattibilità di questo approccio innovativo alla *pattern recognition* in un esperimento ai collisori, che invece sono caratterizzati da eventi multi traccia.

### 2.2.3 Risultati del modello

Si è testata la rete con varie classi di momento e livelli di rumore.

Chiaramente le sue prestazioni possono variare a seconda dei parametri iniziali adottati e dell'occupanza scelta, così a titolo di esempio presentiamo in seguito dei risultati ottenuti nelle seguenti condizioni:

**Parametri**  $N_{L0} = 6$ ,  $N_{L1} = 6$ ,  $T0 = 4$ ,  $T1 = 3$ ; tutti gli altri ai valori di default;

**Tracce** 6 classi di particelle di momento trasverso  $p_T \in \{1, 3, 10\} GeV$ , positive e negative;

**Rumore** Rumore di fondo uniforme con una media di 2 hit per strato;

**Apprendimento** Sono state eseguite 1000 epoche di apprendimento di 100000 eventi l'una.

Qui di seguito si espone la tabella con le metriche migliori ottenute dal test:

<i>Metrica</i>	<i>Valore</i>
$Q_{max}$	6.60
$A_{L1}$	86.7%
$E_{L1}$	4.4%
$S_{L1}$	56.5%

Tabella 2.2: Migliori metriche ottenute dall'allenamento della rete.

Questi risultati esprimono il fatto che almeno un neurone di L1 ha risposto agli eventi in cui era presente una particella nell' 86.7% dei casi, mentre gli eventi catalogati erroneamente come positivi sono circa il 4.4%. L'efficienza della rete, ovvero la capacità di riconoscere correttamente gli eventi di segnale, può quindi essere stimata pari all' 82.3% in prima approssimazione.

Per visualizzare le prestazioni del singolo neurone e analizzare il significato della selettività può essere utile il grafico in fig. 2.3.

Ciascun istogramma è relativo a uno dei neuroni di L1, permette di visualizzare la sua accettazione in relazione ai vari tipi di pattern e il suo tasso d'errore nella configurazione che massimizza il Q valore tra quelle esplorate.

Da fig. 2.3 si nota che il primo neurone è il più attivo di tutti, avendo sparato nel 70% circa degli eventi di segnale. In particolare, è attivo nel 100% degli eventi contenenti particelle positive di  $p_T = 3 GeV$  e  $p_T = 10 GeV$ . Questo neurone è il principale responsabile delle statistiche finali di tabella 2.2.

Man mano che l'indice del neurone aumenta, diminuisce anche la sua attività, fino ad arrivare al numero 6 che si è attivato solo nel 2% di tutti gli eventi di segnale.

È interessante notare come solo nel 60.0% e nel 63.4% degli eventi dove sono presenti particelle di  $p_T = 1 GeV$  almeno un neurone di L1 abbia reagito, con un leggero vantaggio per le particelle negative rispetto alle positive. Questa differenza si presenta invertita per  $p_T = 10 GeV$ , dove l'accettazione per le particelle negative è inferiore di circa il 2%.

In aggiunta a queste considerazioni, è interessante notare che i neuroni 3 e 4 si siano specializzati maggiormente nel riconoscimento delle tracce positive con  $p_T = 1 GeV$ : il primo si attiva nel 97.4% di tali eventi, mentre il secondo nel 77.7%; entrambi risultano meno attivi negli altri

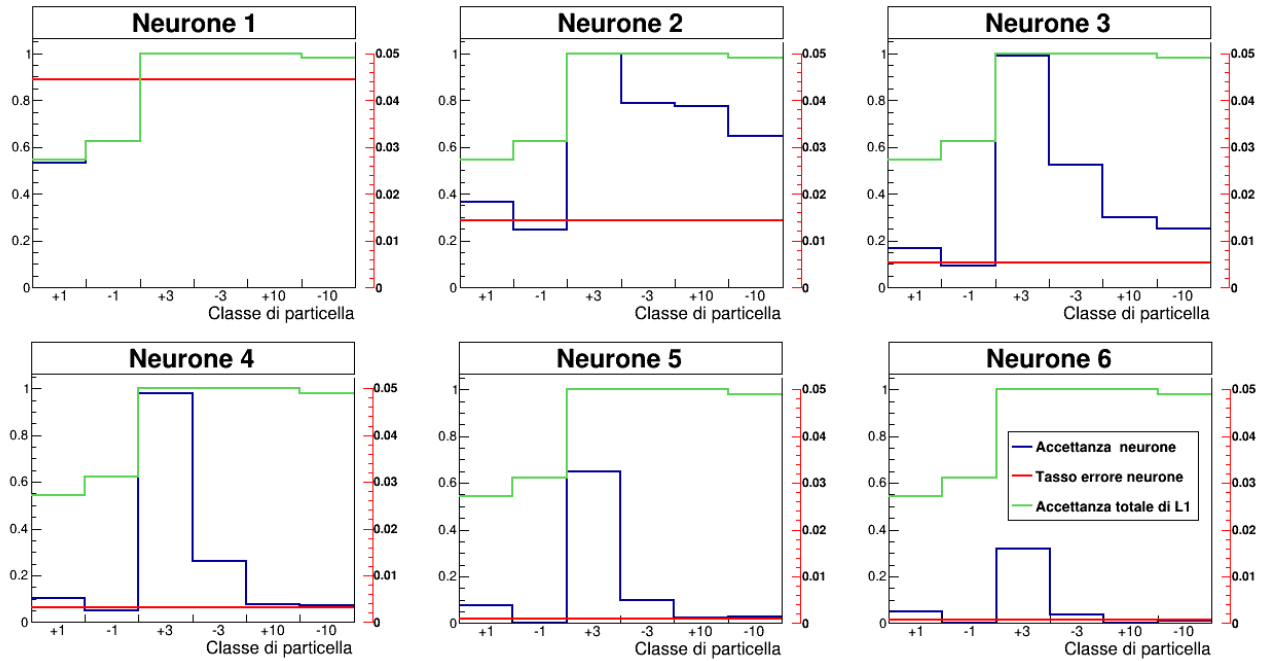


Figura 2.3: Rispettivamente in blu e in rosso l'accettazione per neurone per classe di pattern e il tasso di errore per neurone. In verde l'accettazione totale di L1. Le accettanze sono riferite rispetto alla scala di sinistra, il tasso di errore rispetto a quella di destra.

eventi, con la seconda accettazione più alta che si attesta rispettivamente al 26.5% e al 13.5%. Si è comunque lontani dall'obiettivo ideale di osservare ogni neurone specializzato nel riconoscimento di una particolare classe.

Si presenta infine un grafico, in fig. 2.4, dov'è possibile visualizzare gli impulsi in ingresso dagli afferenti e la conseguente risposta neuronale.

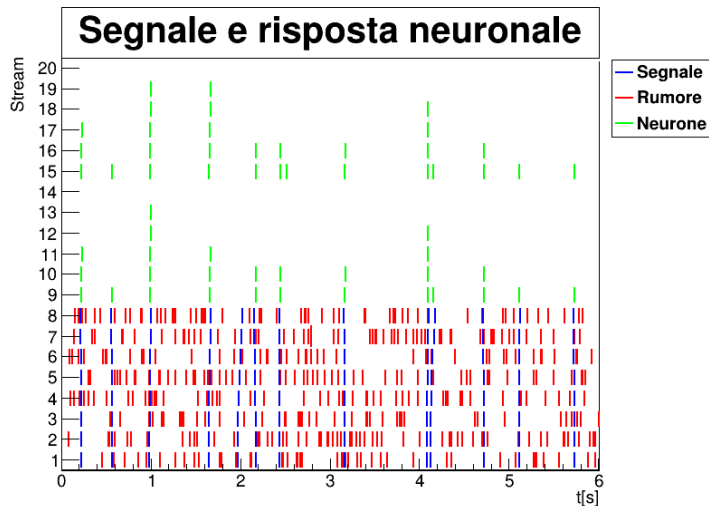


Figura 2.4: Sull'asse delle  $x$  è presente il tempo di codifica, sull'asse delle  $y$  le prime 8 righe rappresentano un afferente, le successive 6 i neuroni appartenenti a L0, le ultime 6 i neuroni appartenenti a L1. Ogni linea disegnata corrisponde a un impulso generato da quella sorgente ("stream") a un dato istante di tempo.

La figura permette di apprezzare come i neuroni reagiscano alle tracce di segnale in un intorno temporale ristretto, rimanendo al contempo poco sensibili al rumore di fondo. Si può notare tuttavia che spesso neuroni dello stesso strato tendono a sparare insieme, comportamento che dovrebbe essere contrastato dalla IPSP.

In conclusione, i risultati ottenuti sono interessanti, perché si dimostra che una SNN potrebbe essere effettivamente in grado di riconoscere delle tracce all'interno di un segnale rumoroso. Per continuare l'indagine occorre raffinare il modello aumentando il grado di realismo.

Il primo obiettivo è l'implementazione di una geometria più vicina a quella di CMS.



## 2.3 Dalla geometria planare alla geometria cilindrica

Per poter studiare le prestazioni di questo nuovo approccio nel contesto di CMS, come primo passo si è implementata una geometria del piano trasverso più simile al caso reale (vedi sez. 1.3).

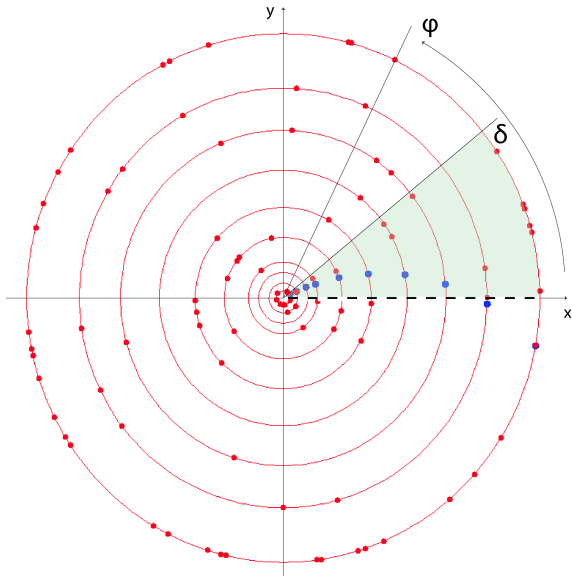


Figura 2.5: *Evento generato in geometria cilindrica, con traccia di  $p_T = 1 \text{ GeV}$  e fondo uniforme in  $\phi$ . Viene messo in evidenza il problema di bordo discusso nella sottosezione 2.3.1.*

associato un picco al tempo  $t_j = \frac{\phi_j + 2\pi \cdot \theta(-\phi_j)}{\omega}$ .

Questa scelta appare naturale perché mappa in modo monotono le tracce di segnale, modulo una discontinuità nel passaggio tra  $\phi = 2\pi \text{ rad}$  e  $\phi = 0$ . Si rimanda alla sez. 2.3.1 la trattazione del problema.

Per rendere la rete compatibile con questi nuovi tempi caratteristici si sono riscalati i parametri temporali  $\tau_m$ ,  $\tau_s$ ,  $\tau^+$  e  $\tau^-$  di un fattore  $10^{-7}$ .

Abbiamo testato il codice con un rumore uniforme in  $\phi$  a vari gradi di occupanza. Trattandosi di un passaggio intermedio, in questa fase non ci si è preoccupati di ottenere il rendimento della rete migliore possibile, ma si è comunque verificato che l'apprendimento continuasse a funzionare.

A titolo esemplificativo si presentano i migliori risultati di un test eseguito con le stesse condizioni di sez. 2.2.3, tracce uniformi in  $\phi$  e risoluzione del problema di bordo proposta in sez. 2.3.1.

<i>Metrica</i>	<i>Valore</i>
$Q_{max}$	7.65
$A_{L1}$	95.4%
$E_{L1}$	1.8%
$S_{L1}$	58.3%

Tabella 2.3: Migliori metriche ottenute dall'allenamento della rete.

Come si può notare sono addirittura migliori di quelle riportati in tab. 2.2. Probabilmente la rete ha beneficiato delle maggiori informazioni della nuova geometria.

Abbiamo previsto 10 strati di rivelazione, idealmente senza spessore, a forma di circonferenza, con raggi compatibili con le distanze radiali medie previste dal progetto del nuovo tracciante di Fase2. Per semplicità ci siamo limitati a considerare la zona del *Barrel* sia per IT che per OT (vedi tabelle 1.1 e 1.2).

A ciascuno di essi è associata una fibra afferente, per un totale di 10 fibre.

Le tracce generate hanno  $p_T$  fissato e sono vincolate a partire dall'origine posta al centro del tracciante; la direzione iniziale segue una distribuzione uniforme in  $\phi$  e le intersezioni con gli strati sono calcolate numericamente con il metodo della bisezione.

Il valore del campo magnetico è stato impostato a  $3.8 \text{ T}$ .

Per la codifica temporale degli impulsi, si prevede di leggere il tracciante in senso antiorario a partire dall'asse delle x. Definendo quindi  $\omega = 2\pi \cdot f$  la velocità angolare di lettura, dove  $f = 40 \text{ MHz}$  corrisponde alla frequenza degli eventi (vedi sez. 1.1), alla j-esima hit nello strato i-esimo viene as-

### 2.3.1 Problema di bordo

Osservando attentamente fig. 2.5, si può intuire il principale problema della codifica adottata. Supponiamo per esempio che una particella di carica negativa segua un moto tale per cui le prime  $n$  hits avvengano ad angoli  $\phi_j > 0$  e le ultime  $10 - n$  a  $\phi_j < 0$ . Ciò implica la formazione di una discontinuità nella codifica del segnale: i primi impulsi si collocheranno temporalmente all'inizio dell'evento mentre gli ultimi vicino alla sua conclusione (vedi grafico a sinistra di fig. 2.6).

Considerando questa geometria e assumendo una distribuzione uniforme delle direzioni delle tracce in  $\phi$ , è possibile stimare che la frazione di particelle con  $p_T$  pari a 1, 3 e 10  $GeV$  coinvolte nel problema di bordo sia approssimativamente del 10%, 3% e 1% rispettivamente.

Come soluzione si è adottata la strategia di proseguire la codifica delle hit una volta aver completato l'angolo giro rileggendo la regione  $\phi \in [0, \delta]$  (vedi la regione verde di fig. 2.5) dove  $\delta$  è un angolo fissato. Il suo valore va scelto in base al  $p_T$  delle tracce che si desidera considerare.

Supponendo di essere interessati a tracce di  $p_T \geq 1 GeV$  (vedi sez. 2.5), si è calcolato che un buon valore è  $\delta = 0.7 rad$ , pari alla massima differenza angolare attesa tra la hit più interna e più esterna della traccia più un piccolo margine dell'ordine del centesimo di radiante.

In questo modo si riesce sempre a presentare alla rete una traccia continua all'interno dell'evento, pena la presenza di duplicati di porzioni di segnale (grafico di destra di fig. 2.6).

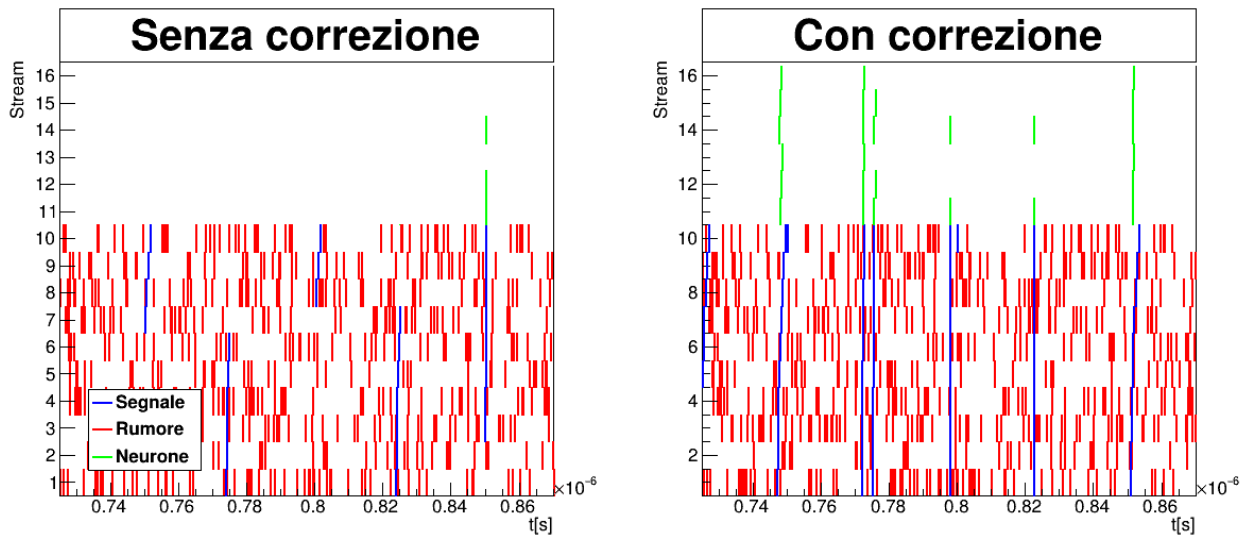


Figura 2.6: Grafici dimostrativi del nostro algoritmo di gestione del bordo. Sull'asse delle  $x$  è presente il tempo di codifica, sull'asse delle  $y$  le prime 10 righe rappresentano un afferente, le successive 3 i neuroni appartenenti a  $L0$ , le ultime 3 i neuroni appartenenti a  $L1$ . Ogni linea disegnata corrisponde a un impulso generato da quella sorgente ("stream") ad un dato istante di tempo.

Per testare gli effetti della modifica si è creato il seguente ambiente standard:

**Parametri**  $N_{L0} = 3$ ,  $N_{L1} = 3$ ,  $T0 = 6$ ,  $T1 = 6$ ; tutti gli altri ai valori di default;

**Tracce** Tre classi di particelle solo negative, di momento trasverso  $p_T \in \{1, 3, 10\} GeV$ , generate in modo tale che per ogni classe il 50% di esse fossero presentassero il problema di bordo;

**Rumore** Rumore di fondo con una media di 1 hit per strato a distribuzione uniforme in  $\phi$  (per mantenere bassa la difficoltà);

**Apprendimento** Sono state eseguite 100 epoche di apprendimento di 100000 eventi l'una.

I risultati del test con e senza correzione sono riportati nella seguente tabella:

	<i>Senza correzione</i>	<i>Con correzione</i>
$Q_{max}$	6.22	7.76
$A_{L1}$	84.1%	97.6%
$E_{L1}$	4.7%	1.1%
$S_{L1}$	53.8%	54.4%

Tabella 2.4: Confronto migliori risultati con e senza correzione del problema di bordo.

Come si può osservare il miglioramento è sostanziale: nel test la correzione ha consentito di aumentare l'accettanza del 13.5% e diminuire il tasso di errore a quasi un quinto, con un leggero miglioramento della selettività.

## 2.4 Gestione della geometria 3D

Per poter ambire a lavorare con dei dati veri o provenienti da una simulazione Monte Carlo rimangono due problemi da risolvere: la gestione della terza dimensione "z" e come mappare correttamente l'informazione radiale all'interno degli afferenti dovendo lavorare con dati provenienti da sensori su strati caratterizzati da spessori e posizioni molte più complesse rispetto al caso ideale di sez. 2.3.

Si noti che la soluzione non può essere cercata nella codifica temporale, perché grazie a essa viene gestita  $\phi$  e non è possibile racchiudervi anche tutte le informazioni contenute nelle rimanenti due dimensioni.

Il nostro approccio che permette di risolvere entrambi i problemi contemporaneamente prevede di sfruttare invece i nervi afferenti.

Prendendo in prestito il linguaggio tipico dell'analisi, si immagini di considerare un dominio di lavoro  $z \in [-a, a]$ ,  $r \in [0, b]$ , di discretizzarlo dividendolo in  $N_z$  e  $N_r$  sezioni dello stesso spessore e di associare a ogni intersezione tra una z e una r sezione un afferente. Così una hit registrata nella posizione determinata da  $(r, \phi, z)$  verrà incasellata in una z-r sezione e codificata in un impulso proveniente al tempo t calcolato come in sez. 2.3 dall'afferente associato.

Questa tecnica ha il pregio di rendere il software indipendente dalla geometria del rilevatore, a meno di impostare adeguatamente il dominio e il numero di sezioni.

Lo scotto principale da pagare è un certo grado di distorsione e perdita dell'informazione originale causato dalla discretizzazione; inoltre il numero di afferenti necessari è pari a  $N_r \cdot N_z$ , che potrebbe diventare certamente consistente. Ciò però non causa un aumento importante della complessità computazionale perché, come detto in sez. 2.1, i calcoli vengono effettuati solo all'occorrenza di impulsi.

## 2.5 Dati disponibili dalle simulazioni Monte Carlo

Per testare la rete si dispone di alcune simulazioni Monte Carlo effettuate con la geometria di Fase2 (vedi sez. 1.3). Gli eventi sono estremamente semplici:

si suppone in ognuno la produzione di un solo tipo di particella, senza *pileup*;

il campione presenta una distribuzione uniforme in  $\phi \in [-\pi, \pi]$ , mentre  $\eta \in [-1, 1]$  per illuminare prevalentemente la zona del *Barrel*, evitando i dischi e i rivelatori inclinati.

Sono state prese in considerazione sei classi di particelle: muoni e antimuoni con momento trasverso  $p_T \in \{1, 3, 10\} \text{ GeV}$ .

Si è mantenuta la possibilità di interazione tra le particelle prodotte e il materiale del rivelatore, come si può vedere in fig. 2.8.

Su ogni evento è stato poi lanciato un algoritmo di associazione a cui ci affidiamo per la distinzione tra cluster dovuti o meno all'evento principale.

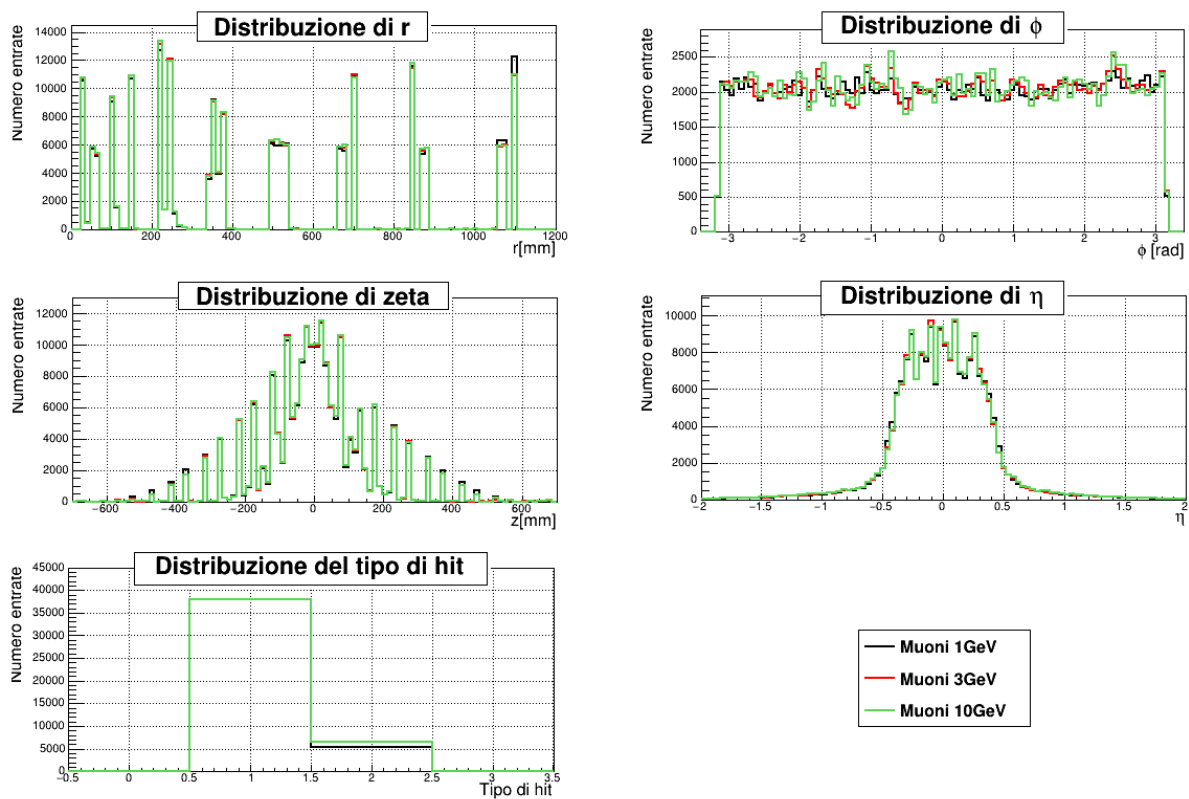


Figura 2.7: Grafici che mostrano le distribuzioni degli eventi Monte Carlo. Il tipo di hit 1 è associato alla traccia principale, 2 invece ad altre fonti (in particolare quelli dovuti all'interazione col materiale).

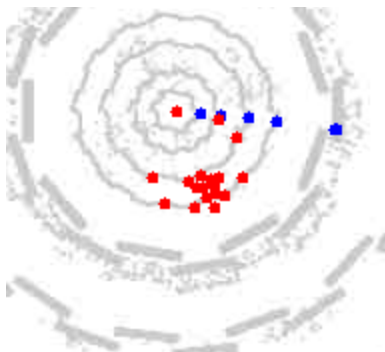


Figura 2.8: Dettaglio di un evento in cui l'urto col materiale del tracciatore ha causato l'emissione di un elettrone.

## 2.6 Generazione del rumore

Come visibile in figura 2.7, i cluster non associati all'evento principale sono in numero decisamente inferiore (in un rapporto di circa 1 a 6). Per incrementarne il numero si necessita di generare altre hit casualmente nello spazio.

Sarebbe poco coerente, però, produrlo in modo uniforme su tutte le r-z sezioni, perché si andrebbero a trattare allo stesso modo tutti i punti dello spazio, indipendentemente dal fatto che siano o meno interessati dalla presenza di tracciatori. L'idea è che per essere certi di rispettare la sottostante geometria di CMS occorrerebbe pescare una collezione di cluster dagli eventi generati.

Chiamata  $\rho(\vec{r})$  la densità di hit per unità di volume in una data posizione dello spazio, nella collezione di eventi generati  $\rho(\vec{x}) = \frac{dN_{hit}}{dV} \propto F(\vec{x}) \cdot G(\vec{x})$  con  $F(\vec{x})$  fattore di forma e  $G(\vec{x})$  fattore geometrico.

Il primo è dovuto a come è fatto il tracciatore (per esempio idealmente varrà 0 nelle zone dove non sono presenti sensori in grado di rivelare le particelle); il secondo dipende dal fatto che le tracce seguono una traiettoria che parte dal centro del rivelatore per poi allontanarsi verso l'esterno e deve essere corretto, altrimenti la densità risulterebbe molto maggiore nell'IT rispetto all'OT.

Dato che esse sono state generate uniformemente in  $\phi$  e  $\eta$ ,  $G(\vec{x}) \propto \frac{1}{r}$ .

Così per l'estrazione del rumore si può utilizzare il metodo dell'inversione:

1. Si salvano all'interno di una lista tutte le hit generate dalla Simulazione Monte Carlo;
2. Si associa a ciascuna hit un peso moltiplicativo pari a  $r$ ;
3. Si definisce la funzione cumulativa  $P(i) = \sum_{j < i} r_j$  (che è monotona e dunque invertibile);
4. Si calcola  $P_{max} = \max(P(i))$ ;
5. Si genera un numero casuale  $x \in [0, P_{max}]$  a distribuzione uniforme;
6. Si estrae  $i_{hit} = P^{-1}(x)$ .
7. Si ripete il procedimento dal punto 5 fino a raggiungere il numero di hit di rumore desiderato.

In fig. 2.9 si può osservare un evento ottenuto sovrapponendo alla traccia principale del rumore generato secondo questo procedimento con 200 cluster.

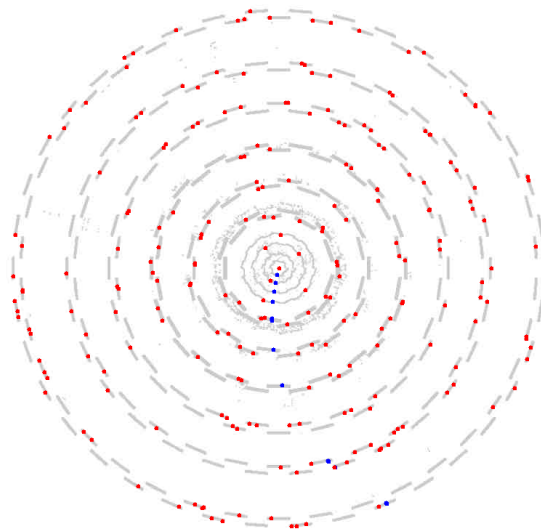


Figura 2.9: *Proiezione sul piano trasverso di un evento con rumore generato, distribuito uniformemente, con una media di 200 cluster per evento. In blu i cluster associati all'evento principale, in rosso il rumore di fondo.*

## 2.7 Modifica della rete con taglio pseudocasuale di connessioni

Per provare ad aumentare la differenziazione dei vari neuroni e favorire un aumento della selettività si è pensato di introdurre la possibilità di eliminare pseudocasualmente alcune connessioni neurali. Ciò dovrebbe in linea di principio forzare neuroni diversi a vedere porzioni diverse del segnale, aiutandoli nella specializzazione. Questa operazione viene gestita da 3 nuovi iperparametri, che assumono il significato di probabilità di sopravvivenza delle connessioni:

$CF_{i,0}$  gestisce quelle tra afferenti e neuroni di L0;

$CF_{i,1}$  gestisce quelle tra afferenti e neuroni di L1;

$CF_{0,1}$  gestisce quelle tra neuroni di L0 e neuroni di L1;

Essi devono essere inizializzati all'avvio del programma.

Per chiarezza espositiva si riporta un elenco finale di tutti gli iperparametri del modello:

**Tipo A: ottimizzabili nel corso dell'esecuzione**  $T0, T1, \alpha, I_{L1}, K, K1, K2, E_p, D_{L1}$ ;

**Tipo B: costanti per tutto il processo:**  $N_{L0}, N_{L1}, CF_{i,0}, CF_{i,1}, CF_{0,1}, a^+, a^-, \tau_m, \tau_s, \tau^+, \tau^-$ .

## 2.8 Processo di apprendimento e ricerca a griglia

Per testare il modello si sono generati eventi con la presenza di una singola traccia sovrapposta a del rumore di fondo ed eventi con solo rumore di fondo, in proporzione 1:1 in successione casuale. Si sono utilizzate tutte le 6 classi di particelle disponibili dalle simulazioni Monte Carlo con egual frequenza (per maggiori dettagli si rimanda alla sez. 2.5).

Il numero di hit di rumore presente segue una distribuzione poissoniana.

Si sono così prodotti tre set di dati da  $N_{ev}^{gen} = 100000$  eventi con numero medio di hit di rumore  $N_{hit}^{fondo} \in \{50, 100, 200\}$ . Nelle figure 2.10, 2.11, 2.12 si riporta una visualizzazione grafica di tre eventi proiettati sul piano trasverso.

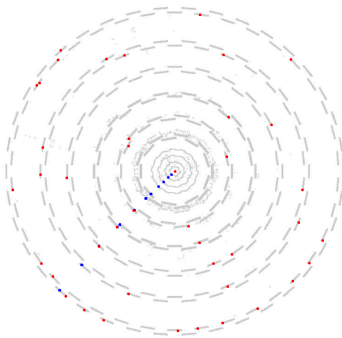


Figura 2.10: *Evento con rumore generato, distribuito uniformemente, con una media di 50 cluster per evento, proiettato nel piano trasverso.*

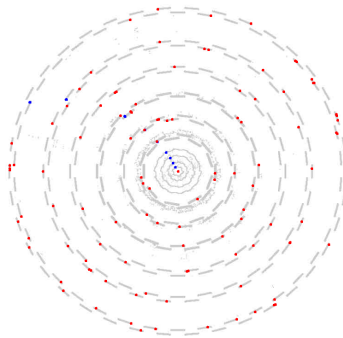


Figura 2.11: *Evento con rumore generato, distribuito uniformemente, con una media di 100 cluster per evento, proiettato nel piano trasverso.*

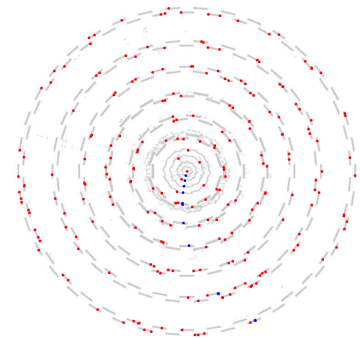


Figura 2.12: *Evento con rumore generato, distribuito uniformemente, con una media di 200 cluster per evento, proiettato nel piano trasverso.*

L'apprendimento della rete richiede un numero significativo di eventi. Allo scopo di massimizzare il numero di eventi di segnale disponibili si è costruito il set di dati in modo da riutilizzare il pattern delle tracce ruotando di un angolo  $\Delta\phi \cdot k$  solo le hit associate alla traccia principale, con  $k = 1, 2, \dots$  che tiene conto del numero di iterazioni e  $\Delta\phi = \frac{2\pi}{1 + \frac{N_{ev} \cdot N_{ep}}{N_{gen}^{gen}}}$ .

In questo modo si riesce a presentare alla rete eventi con la medesima topologia ma diversi, minimizzando il costo in termini di risorse.

Nella ricerca a griglia si è utilizzato  $\Delta\phi = \frac{\pi}{3}$  con  $k = 1, \dots, 5$ , mentre nell'apprendimento finale  $\Delta\phi = \frac{\pi}{8}$  con  $k = 1, \dots, 15$ .

La discretizzazione dello spazio è stata fissata considerando  $N_r = 50$  sezioni con  $r \in [0, 120]$  cm e  $N_z = 50$  sezioni con  $z \in [-60, 60]$  cm (vedi fig. 2.7).

Occorre inoltre stabilire con che parametri di Tipo B sia più opportuno lavorare (vedi sez. 2.7). Per ridurre la dimensionalità del problema si considerano  $a^+ = 0.03125$ ,  $a^- = 0.02656$ ,  $\tau_m = 1.00$  ns,  $\tau_s = 0.25$  ns,  $\tau^+ = 1.68$  ns e  $\tau^- = 3.37$  ns, compatibilmente con i parametri di tab. 2.1 a meno di riscaldamento temporale.

In lavori futuri si potrebbe studiare se questi siano effettivamente i migliori anche per il nostro problema.

Restano quindi da stabilire i valori di  $N_{L0}$ ,  $N_{L1}$ ,  $CF_{i,0}$ ,  $CF_{i,1}$  e  $CF_{0,1}$ .

Sarebbe stato interessante effettuare una ricerca a griglia testando le possibili combinazioni di tutti i parametri contemporaneamente, ma il numero di combinazioni possibili sarebbe stato troppo elevato e il tempo necessario sarebbe diventato considerevolmente lungo.

Si è deciso quindi di effettuare prima una ricerca facendo variare il solo numero di neuroni, testando tutte le configurazioni in un intervallo  $N_{L0}, N_{L1} \in \{6, 7, 8, 9, 10\}$ .

Fissato poi il numero di neuroni alla configurazione migliore si sono testate tutte le possibili combinazioni di  $CF_{i,0}$ ,  $CF_{i,1}$ ,  $CF_{0,1} \in \{0.6, 0.85, 1\}$ . L'unico fattore preso in considerazione per la valutazione è il massimo raggiunto dal Q valore.

Per controllare se livelli di rumore diverso potessero condurre a parametri diversi si è effettuata la stessa ricerca per tutti e tre i casi considerati.

Si sono svolte delle prove preliminari, grazie alle quali si sono stabilite tre coppie di valori iniziali ragionevoli per le soglie di attivazione in relazione al rumore. In ordine crescente si sono utilizzate  $(T0 = 15, T1 = 10)$ ,  $(T0 = 30, T1 = 20)$ ,  $(T0 = 45, T1 = 30)$ .

Si specifica inoltre che durante la ricerca si sono effettuate un numero  $N_{ep} = 5$  di epoche di apprendimento con all'interno un numero di eventi pari a  $N_{ev} = 100000$ . Si è fatta questa scelta per consentire all'algorithmo di ottimizzazione di accomodare i parametri di default se necessario, in modo da non escludere configurazioni interessanti che avrebbero necessitato di un bilanciamento leggermente diverso.

I risultati migliori raggiunti tramite questo processo sono riassunti in tabella 2.5.

$\overline{N_{hit}^{fondo}}$	$N_{L0}$	$N_{L1}$	$CF_{i,0}$	$CF_{i,1}$	$CF_{0,1}$	$Q_{max}$
50	7	8	1	0.85	0.6	7.179
100	7	10	0.6	1	1	6.897
200	7	10	0.8	1	0.6	6.092

Tabella 2.5: Migliori parametri trovati con la ricerca a griglia al variare del fondo.

Per concludere, si è effettuato un ultimo processo di apprendimento un po' più lungo con tutti i migliori parametri trovati. Si sono impostati  $N_{ep} = 15$  e  $N_{ev} = 100000$ .

Segue una tabella con una lista completa degli iperparametri migliori e dei valori raggiunti dalle metriche.

	<i>Parametro</i>	$\overline{N_{hit}^{fondo}} = 50$	$\overline{N_{hit}^{fondo}} = 100$	$\overline{N_{hit}^{fondo}} = 200$
<i>Tipo A</i>	$N_{L0}$	7	7	7
	$N_{L1}$	8	10	10
	$CF_{i,0}$	1	0.6	0.8
	$CF_{i,1}$	0.85	1	1
	$CF_{0,1}$	0.6	1	0.6
<i>Tipo B</i>	$T0$	15.0	30	45
	$T1$	13.8	20	30
	$\alpha$	0.241	0.25	0.25
	$I_{L1}$	1.29	1	1
	$K$	1.04	1	1
	$K_1$	2.46	2	2
	$K_2$	4.00	4	4
	$E_p$	2.50	2.5	2.5
	$D_{L1}$	0.792	1.024	1.016
<i>Metriche</i>	$Q_{max}$	7.17	6.90	6.09
	$A_{L1}$	91.5%	87.1%	82.2%
	$E_{L1}$	2.2%	2.1%	3.9%
	$S_{L1}$	53.5%	53.4%	53.1%

Tabella 2.6: Tabella riassuntiva dei parametri della rete e dei risultati ottenuti.

Dalla tabella si osserva che per le due condizioni di rumore più elevate tutti i parametri di Tipo A tranne  $D_{L1}$  sono rimasti ai valori assegnati in partenza.

Anche nel caso di  $\overline{N_{hit}^{fondo}} = 50$  comunque l'escursione è rimasta contenuta nella maggior parte di essi. Probabilmente ciò è dovuto al fatto che i dati di partenza sono già molto vicini a quelli ottimali e il semplice algoritmo di ottimizzazione adottato, non è in grado di individuare la direzione corretta nello spazio degli iperparametri per garantire un ulteriore aumento del Q valore massimo con il numero ridotto di epoche adottato.

Si potrebbe allora tentare di aumentarle, ma con l'hardware a nostra disposizione ciò necessiterebbe una quantità di tempo superiore a quella di cui si dispone per la presentazione dei risultati (vedi sez. 2.8.1). Occorrerebbe pensare a un metodo più raffinato per garantire prestazioni migliori in tempi ragionevoli.

Comunque la prestazione della rete complessiva è incoraggiante: la percentuale media di eventi di segnale correttamente identificati può essere stimata effettuando una sottrazione tra l'accettanza e il tasso di errore, ottenendo circa 89.3%, 85.0% e 78.3% per i tre set di dati.

Si osserva inoltre che, come è facile aspettarsi, all'aumentare del numero di hit di rumore il problema diventa sempre più complesso, il che si traduce in una diminuzione del Q valore massimo.

Anche la selettività tende leggermente a scendere, ma si tratta di variazioni dell'ordine del per cento.



Per indagare più nel dettaglio il comportamento della rete, segue un grafico (fig. 2.13) dove è possibile osservare l'accettazione e il tasso di errore per classe di pattern di ogni neurone di L1 e confrontarla con quella totale.

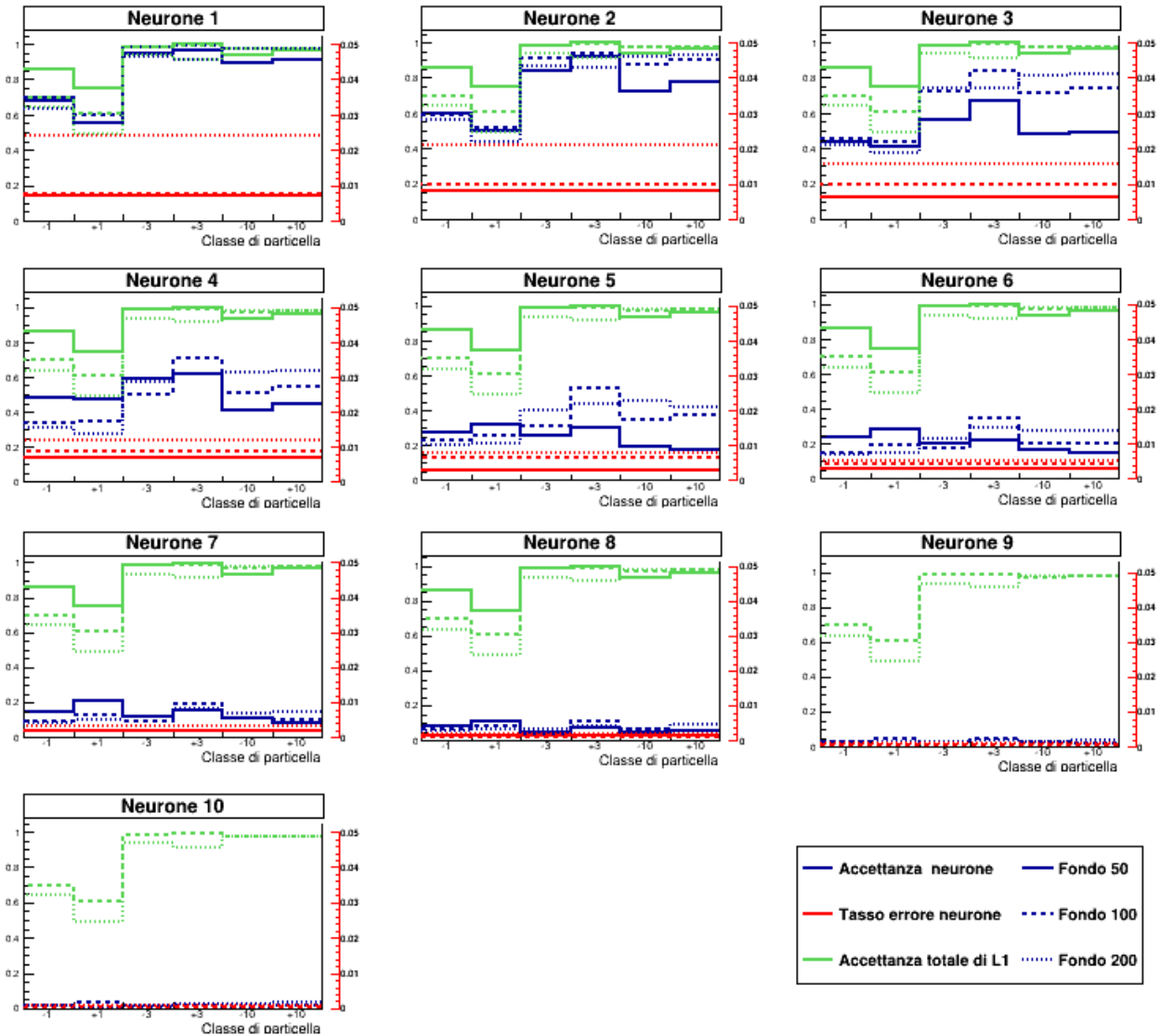


Figura 2.13: Rispettivamente in blu e in rosso l'accettazione per neurone per classe di pattern e il tasso di errore per neurone. In verde l'accettazione totale di L1. Le accettanze sono riferite rispetto alla scala di sinistra, il tasso di errore rispetto a quella di destra. Si identificano con tratti diversi i risultati dei tre set di dati.

Ciascun istogramma è relativo a uno dei neuroni di L1, permette di visualizzare la sua accettazione in relazione ai vari tipi di pattern e il suo tasso d'errore nella configurazione che massimizza il Q valore tra quelle esplorate.

Si osserva che per tutti i livelli di rumore il neurone 1 è quello più attivo, con una media intorno all'80/85% di reazioni a eventi di segnale; è sulle sue prestazioni che si basano principalmente i risultati della rete. A mano a mano che si esaminano i neuroni successivi si nota che essi sono sempre meno attivi, fino ad arrivare agli ultimi che sparano molto raramente: il decimo neurone dei casi  $\overline{N_{hit}^{fondo}} = 100, 200$  spara solo circa nel 2% degli eventi di segnale.

Complessivamente non si osservano neuroni particolarmente specializzati, nonostante l'introduzione del taglio pseudocasuale delle connessioni neurali (vedi sez. 2.7).

Anche in questo caso si sono riscontrate le maggiori difficoltà di apprendimento con le particelle di  $p_T = 1 \text{ GeV}$ , specialmente nel caso dei muoni.

Per analizzare meglio questi aspetti si riporta una tabella con l'accettazione totale di L1 per classe di pattern.

$N_{hit}^{fondo}$	$A_{L1,-1}$	$A_{L1,+1}$	$A_{L1,-3}$	$A_{L1,+3}$	$A_{L1,-10}$	$A_{L1,+10}$
50	86.0%	75.7%	98.6%	99.9%	93.5%	96.4%
100	70.1%	60.7%	98.4%	98.9%	97.0%	97.6%
200	63.9%	49.4%	93.5%	91.6%	97.5%	97.4%

Tabella 2.7: Accettazione totale di L1 per classe di pattern in relazione al set di dati.

Dalla tabella si può notare come l'aumento del rumore renda particolarmente più difficile l'individuazione delle particelle a basso momento trasverso, con differenze di accettazione anche di 25 punti percentuali tra i due campioni estremi (vedi  $A_{L1,+1}$ ). Questo invece non vale quelle con  $p_T = 10 \text{ GeV}$ , dove anzi sembrerebbe quasi valere l'opposto.

Si evidenzia come la differenza di carica nel caso  $p_T = 1 \text{ GeV}$  comporti una differenza di accettazione dell'ordine del 10%; a momento trasverso più elevato le differenze sono invece dell'ordine di qualche punto percentuale al massimo.

Per verificare che le metriche siano coerenti con l'effettivo comportamento della rete si sono prodotti i seguenti grafici (fig. 2.14, 2.15, 2.16), in cui è possibile vedere il segnale trasmesso in ingresso dagli afferenti e gli impulsi emessi in risposta dei neuroni. L'elevato numero di afferenti complica la visualizzazione rispetto al caso precedente di fig. 2.4. Si è scelto di ordinarli per gruppi di  $z$  sezioni: le prime 50 righe sono associate a  $z = z_{min} = -60 \text{ cm}$  al variare di  $r \in [0, r_{max} = 120 \text{ cm}]$ , le successive 50 sono associate a  $z = z_{min} + \Delta z$  al variare di  $r$  e così via. Perciò la riga centrale numero 1250 corrisponde alla  $z$ - $r$  sezione con  $z = 0$  e  $r = 0$ .

Tipicamente le tracce di segnale della simulazione Monte Carlo hanno origine a  $z$  vicini a 0, per cui la prima hit verrà codificata in un afferente intorno alla 1250. Gli impulsi successivi potrebbero rimanere in afferenti limitrofi nel caso in cui la particella non vari troppo la sua coordinata  $z$ , oppure risultare più dispersi in caso contrario.

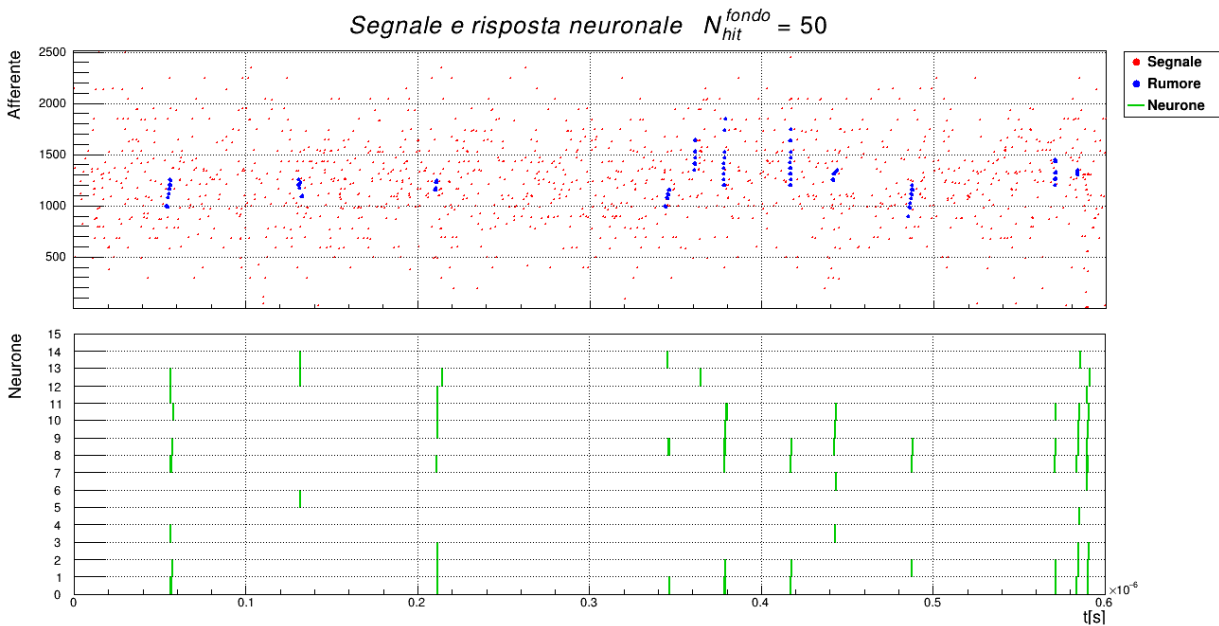


Figura 2.14: Sull'asse delle  $x$  è presente il tempo di codifica, sull'asse delle  $y$  dell'istogramma inferiore le prime 7 righe rappresentano i neuroni di L0, le successive 8 quelli di L1. Nell'istogramma superiore ogni riga corrisponde a un afferente. Ciascuna linea disegnata corrisponde a un impulso generato da quella sorgente ad un dato istante di tempo.

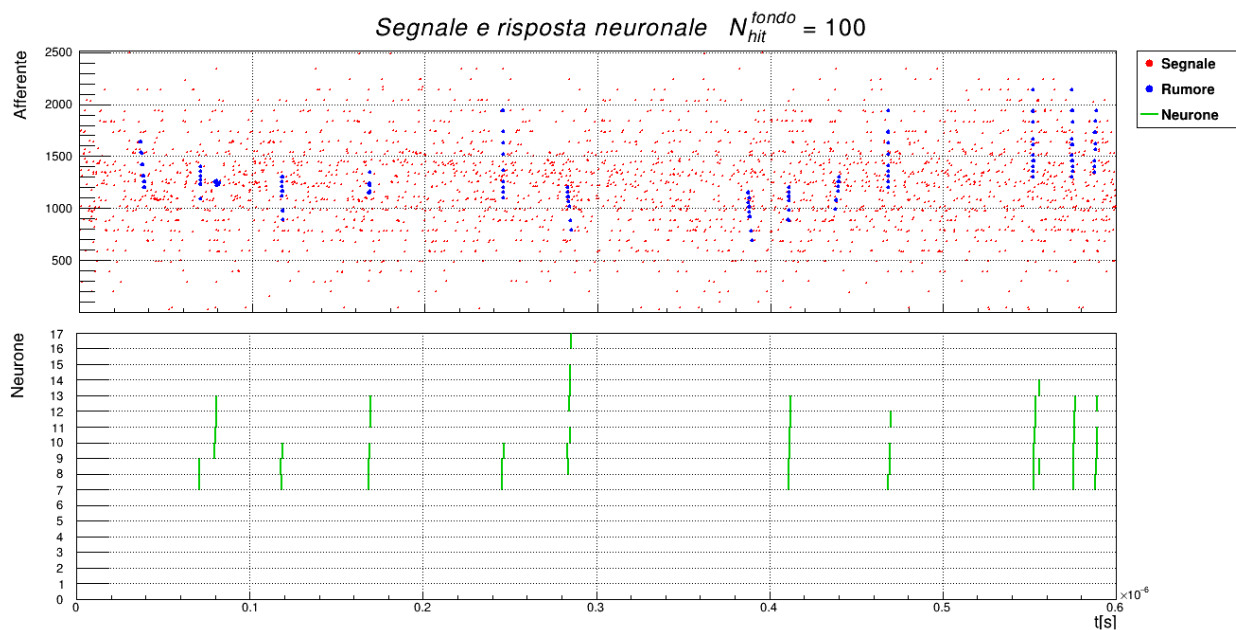


Figura 2.15: Sull'asse delle  $x$  è presente il tempo di codifica, sull'asse delle  $y$  dell'istogramma inferiore le prime 7 righe rappresentano i neuroni di L0, le successive 10 quelli di L1. Nell'istogramma superiore ogni riga corrisponde a un afferente. Ciascuna linea disegnata corrisponde a un impulso generato da quella sorgente ad un dato istante di tempo.

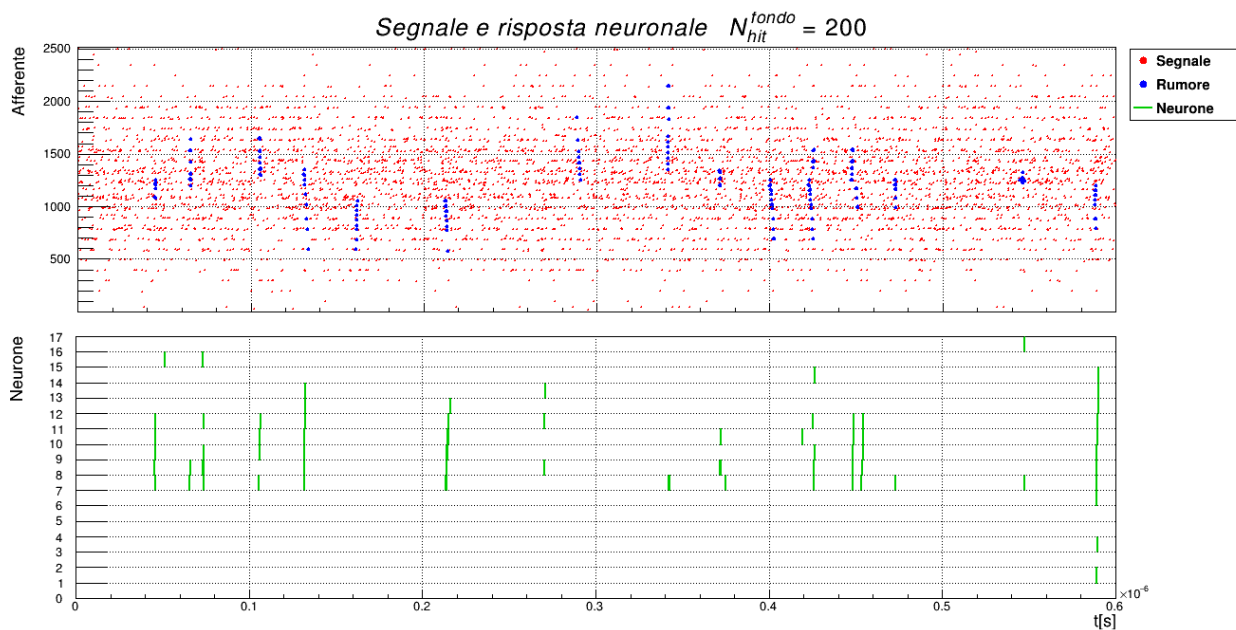


Figura 2.16: Sull'asse delle  $x$  è presente il tempo di codifica, sull'asse delle  $y$  dell'istogramma inferiore le prime 7 righe rappresentano i neuroni di L0, le successive 10 quelli di L1. Nell'istogramma superiore ogni riga corrisponde a un afferente. Ciascuna linea disegnata corrisponde a un impulso generato da quella sorgente ad un dato istante di tempo.

I grafici confermano che la rete sta rispondendo bene, vista la buona corrispondenza tra i tempi di arrivo delle hit di segnale e quelli di attivazione dei neuroni di L1. Si nota inoltre un fatto interessante: a livelli maggiori di rumore i neuroni di L0 sono più soggetti a disattivarsi. Ciò è confermato anche dalle statistiche ottenute dal programma, secondo le quali il numero di impulsi che mediamente si sono originati in L0 nel corso di 100000 eventi sono circa 40000 per  $N_{hit}^{fondo} = 50$  (stesso ordine di grandezza degli impulsi di L1), ma solo 2000 circa negli altri due casi.

Si nota nuovamente la tendenza indesiderata dei neuroni di rispondere in cascata alle tracce, cosa che suggerisce di tentare un potenziamento della IPSP in esecuzioni future.

### 2.8.1 Hardware e tempi di esecuzione

Sia la ricerca a griglia che i processi di apprendimento finali sono stati eseguiti su macchine facenti parte dell'infrastruttura informatica del gruppo CMS di Padova, dotate di 32 CPU "AMD Opteron(tm) Processor 6378" con frequenza di clock di 2.4 GHz e 64 GB di RAM.

Considerando un test con  $N_{ep} = 15$  epoche di apprendimento da  $N_{ev} = 100000$  eventi, il tempo impiegato per la ricerca del set di valori ottimali degli iperparametri di tipo A è riportato in tabella 2.8:

$\overline{N}_{hit}^{fondo}$	<i>Tempo [ore]</i>
50	5.5
100	7.5
200	19

Tabella 2.8: Tempi di elaborazione suddivisi per set di dati.

Il tempo necessario per l'elaborazione di un singolo evento da parte del programma varia in base al numero di impulsi che si sono verificati, i quali dipendono sia dal numero di hit che dalla risposta dei neuroni che cambia nel corso dell'apprendimento.

Questi tempi rendono dispendioso ottenere dei risultati consistenti dall'algoritmo di ottimizzazione degli iperparametri di tipo A, che necessiterebbe di qualche centinaia se non migliaia di epoche di allenamento.

# Capitolo 3

## Conclusioni

In questo lavoro si è iniziata a esplorare la possibilità di utilizzare un modello di calcolo neuro-morfico basato su *spiking neural networks* per complementare la *pattern recognition* nel contesto del complesso problema della ricostruzione di traiettorie di particelle cariche negli esperimenti di fisica delle particelle agli acceleratori.

Si è costruito un modello in grado di gestire la geometria tridimensionale del rivelatore di CMS di Fase2 e lo si è testato delle simulazioni Monte Carlo.

I risultati dimostrano che la rete è stata in grado di imparare a riconoscere diverse tipologie di particelle in modo non supervisionato sotto certe ipotesi semplificative di lavoro.

In particolare, in condizioni di rapporto segnale rumore pari a 1:5, 1:10 e 1:20, si è stimato che mediamente sia stata in grado di identificare correttamente rispettivamente il 89.3%, 85.0% e 78.3% degli eventi contenenti particelle con momento trasverso  $p_T \in \{1, 3, 10\} \text{ GeV}$ .

Le limitazioni del progetto su cui concentrarsi in lavori futuri sono:

- Scarsa specializzazione dei neuroni;
- Ipotesi di lavorare con eventi a singola traccia, contrariamente a ciò che avviene in una tipica collisione;
- Elevato tempo di elaborazione;
- Inefficienza dell'algoritmo di ottimizzazione degli iperparametri;
- Aver limitato il numero di iperparametri soggetti a ottimizzazione.

# Bibliografia

- [1] The Phase-2 Upgrade of the CMS Tracker. Technical report, CERN, Geneva, 2017.
- [2] CERN Yellow Reports: Monographs. Cern yellow reports: Monographs, vol. 10 (2020): High-luminosity large hadron collider (hl-lhc): Technical design report, 2020.
- [3] The CMS Collaboration. Description and performance of track and primary-vertex reconstruction with the CMS tracker. *Journal of Instrumentation*, 9(10):P10009–P10009, October 2014.
- [4] The CMS Collaboration. The CMS phase-1 pixel detector upgrade. *Journal of Instrumentation*, 16(02):P02027–P02027, February 2021.
- [5] Lyndon Evans and Philip Bryant. LHC machine. *Journal of Instrumentation*, 3(08):S08001–S08001, August 2008.
- [6] Stephane Fartoukh, Sofia Kostoglou, Matteo Solfaroli Camillocci, Gianluigi Arduini, Hanes Bartosik, Chiara Bracco, Krzysztof Brodzinski, Roderik Bruce, Xavier Buffat, Marco Calviani, Francesco Cerutti, Ilias Efthymiopoulos, Brennan Goddard, Giovanni Iadrola, Nikos Karastathis, Anton Lechner, Elias Metral, Nicolas Mounet, Francois-Xavier Nuiiry, Parthena Stefania Papadopoulou, Yannis Papaphilippou, Brian Petersen, Tobias Hakan Bjorn Persson, Stefano Redaelli, Giovanni Rumolo, Benoit Salvant, Guido Sterbini, Helga Timko, Rogelio Tomas Garcia, and Jorg Wenninger. LHC Configuration and Operational Scenario for Run 3. Technical report, CERN, Geneva, 2021.
- [7] Rudolf Frühwirth and Are Strandlie. *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*. Springer International Publishing, 2021.
- [8] M Jeitler. The upgrade of the CMS trigger system. *Journal of Instrumentation*, 9(08):C08002–C08002, August 2014.
- [9] Christiane Lefèvre. The CERN accelerator complex. Complexe des accélérateurs du CERN. 2008.
- [10] Timothée Masquelier, Rudy Guyonneau, and Simon J. Thorpe. Competitive STDP-based spike pattern learning. *Neural Computation*, 21(5):1259–1276, May 2009.
- [11] Guo qiang Bi and Mu ming Poo. Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience*, 24(1):139–166, March 2001.
- [12] Tai Sakuma and Thomas McCauley. Detector and event visualization with SketchUp at the CMS experiment. *Journal of Physics: Conference Series*, 513(2):022032, June 2014.
- [13] Sen Song, Kenneth D. Miller, and L. F. Abbott. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9):919–926, September 2000.