

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA · SEDE DI CESENA
DIPARTIMENTO DI INGEGNERIA DELL'ENERGIA ELETTRICA E
DELL'INFORMAZIONE
- GUGLIELMO MARCONI -

CORSO DI LAUREA TRIENNALE IN
INGEGNERIA ELETTRONICA PER L'ENERGIA E L'INFORMAZIONE

**USO DI TECNICHE DI MACHINE
LEARNING PER VALUTAZIONI
DI COPERTURA WIRELESS
IN AMBIENTE URBANO**

ELABORATO DI
CAMPI ELETTROMAGNETICI

RELATORE:
**CHIAR.MO PROF.
ENRICO MARIA VITUCCI**

PRESENTATA DA:
KLEVIS DUKA

CORRELATORI:
**CHIAR.MO PROF.
VITTORIO DEGLI ESPOSTI
DOTT. ING.
NICOLA DI CICCO**

IV SESSIONE
ANNO ACCADEMICO 2022/2023

Questa pagina è lasciata intenzionalmente bianca.

Keywords:

Machine Learning
Reti Convoluzionali
U-Nets
Ray-Tracing
Copertura Wireless

Questa pagina è lasciata intenzionalmente bianca.

*Dedicata ai miei genitori,
che con amore e sacrifici hanno
reso possibile questo cammino.*

Questa pagina è lasciata intenzionalmente bianca.

Indice

1	Introduzione	1
2	Machine Learning	3
2.1	Generalità	3
2.1.1	Dati	3
2.1.2	Tipologie di algoritmi	4
2.2	Reti convoluzionali	6
2.3	U-Nets	8
3	Ray Tracing	11
3.1	Interazioni delle onde	13
3.1.1	Riflessione e trasmissione	13
3.1.2	Diffrazione	14
3.1.3	Scattering diffuso	14
3.2	Propagazione multicammino	15
3.2.1	Power-delay profile	15
3.2.2	Delay Spread	16
3.2.3	Angle spread	16
3.3	Utilizzo dell'applicativo	17
3.4	Algoritmo di semplificazione della mappa	21
4	Generazione del dataset e addestramento del modello	23
4.1	Generazione del dataset	23
4.1.1	Ritaglio in mappe più piccole	24
4.1.2	Generazione delle mappe di input	25
4.1.3	Simulazioni RT e generazione delle mappe di output	25

4.1.4	Organizzazione del dataset	26
4.2	Addestramento del modello	27
5	Analisi dei risultati	29
5.1	Parametri di valutazione	29
5.1.1	Cross-Entropy Loss	29
5.1.2	F1 Binary score	30
5.2	Risultati	31
5.3	Conclusioni	34

Capitolo 1

Introduzione

Nell'ambito dei sistemi radio, uno dei problemi principali è sempre stato il posizionamento ottimale delle antenne trasmettenti al fine di garantire una copertura adeguata. Nel corso del tempo, per accelerare questo processo, si è gradualmente evoluto da un approccio basato su misurazioni sul campo, non sempre praticabili, ad una pianificazione assistita da tool di previsione di campo deterministici. Fra questi ricoprono un ruolo significativo i modelli a raggi (Ray Tracing) che rappresentano un buon compromesso fra il tempo di calcolo e accuratezza. Tuttavia, negli ultimi anni, con l'accesso a tecnologie sempre più avanzate, si stanno cercando di implementare tecniche di Machine Learning per poter affrontare queste sfide in maniera innovativa.

Considerando la non trascurabile potenza di calcolo richiesta, soprattutto nel caso di scenari di grandi dimensioni (per esempio, un ambiente urbano denso) dagli attuali algoritmi di Ray-Tracing, questo studio si propone di esaminare la fattibilità dell'impiego di specifiche metodologie di Machine Learning per l'analisi della copertura in contesti urbani. L'obiettivo principale è valutare se questa combinazione possa condurre a risultati più efficienti rispetto agli approcci tradizionali, pur mantenendo un livello adeguato di precisione.

Gli ambienti urbani utilizzati per addestrare il modello includono città tipiche europee di medie dimensioni, come Bologna, Monaco e Helsinki, così come ambienti urbani estremamente densi, come il centro della città di San Francisco negli USA. Questa scelta eterogenea è stata fatta per garantire una vasta gamma di configurazioni edilizie, rendendo così la rete neurale prodotta il più generale possibile.

Tutti gli scenari di addestramento prevedono l'uso di un'antenna a dipolo al fine di analizzare casi relativi ad antenne omnidirezionali e semplificare l'analisi senza la necessità di considerare l'orientamento specifico dell'antenna stessa.

Capitolo 2

Machine Learning

2.1 Generalità

Il Machine Learning, o apprendimento automatico, è una disciplina dell'Intelligenza Artificiale che ha rivoluzionato il modo in cui affrontiamo problemi complessi in vari campi, dall'elaborazione del linguaggio naturale all'elaborazione delle immagini, alla previsione finanziaria. Questa disciplina si basa sul concetto fondamentale di far apprendere ai computer come eseguire compiti senza essere esplicitamente programmati per svolgerli. Invece di seguire istruzioni fisse, i sistemi di machine learning quindi si adattano e migliorano le loro prestazioni attraverso l'esperienza e i dati.

2.1.1 Dati

Vista la forte dipendenza di questi algoritmi dai dati risulta quindi fondamentale approfondire la loro struttura e le varie suddivisioni. I dati necessari per l'addestramento possono essere di diversi tipi e una prima suddivisione può essere definita nel seguente modo:

- **Dati numerici:** Questa categoria comprende valori numerici continui o discreti che rappresentano caratteristiche quantitative di un fenomeno in analisi. Temperature, distanze, età, e prezzi sono esempi di dati numerici. Questi dati sono naturalmente rappresentati da vettori numerici multidimensionali, rendendo possibile l'applicazione di algoritmi di Machine Learning, come la regressione, per prevedere valori futuri o scoprire relazioni complesse tra variabili.

- **Dati categorici:** I dati categorici rappresentano valori associati a caratteristiche qualitative o all'assenza/presenza di determinate caratteristiche. Questi dati sono spesso trattati attraverso sistemi di classificazione, come alberi di decisione o regole. Esempi di dati categorici includono il genere di una persona, il suo colore degli occhi, le categorie di prodotti o i tipi di veicoli. L'obiettivo qui è assegnare etichette o categorie ai dati in base alle loro caratteristiche.
- **Sequenze e dati strutturati:** Questa categoria comprende dati che hanno relazioni spazio-temporali o seguono una struttura specifica. Ad esempio, sequenze di dati possono rappresentare suoni che compongono una parola o una frase. Questi dati richiedono spesso approcci più complessi, come le reti neurali ricorrenti per analizzare sequenze o l'uso di algoritmi di elaborazione del linguaggio naturale (NLP) per comprendere testo strutturato..

In conclusione, la diversità dei dati è un elemento fondamentale nel Machine Learning, poiché l'addestramento di algoritmi richiede un'adeguata gestione di dati numerici, categorici e strutturati per ottenere risultati accurati e significativi.

2.1.2 Tipologie di algoritmi

I compiti di apprendimento automatico invece si possono suddividere in tre principali categorie:

- **Algoritmi supervisionati**

Questa procedura prevede una supervisione da parte dell'essere umano che deve etichettare le coppie di input-output con cui l'algoritmo verrà addestrato. In base ai dati di addestramento il modello deve estrarre una regola generale da applicare ai successivi input di test e utilizzo vero e proprio.

- **Algoritmi non supervisionati**

Con questo approccio si intendono categorizzare tutti quei modelli che estrapolano una struttura dei dati di ingresso senza che vi sia un controllo umano in precedenza. I modelli devono quindi estrarre un pattern dei dati senza avere nemmeno il numero di caratteristiche che si cercano nei dati in ingresso.

- **Algoritmi per rinforzo**

Questi modelli vengono addestrati in ambienti dinamici mentre ricevono da parte di un insegnante dei feedback utili per verificare la correttezza dei loro

output. Quindi la supervisione non avviene sui dati in input ma sui risultati dell'algoritmo.

Esistono poi delle vie di mezzo come gli algoritmi semi-supervisionati dove il dataset di allenamento viene solo parzialmente etichettato con i rispettivi output.

Gli algoritmi di interesse in questo elaborato sono quelli supervisionati. Questi in particolare possono essere a loro volta classificati in due diverse categorie in base alla tipologia di dati che producono in output.

- **Classificazione**

Algoritmi atti ad attribuire una classe ad un determinato dato in ingresso. Dati di una stessa classe possiedono proprietà comuni. Tra i più noti possiamo considerare algoritmi di riconoscimento di lettere dell'alfabeto, razza di animale, tipo di indumento, etc. Se i gruppi di suddivisione non sono noti a priori questi algoritmi vengono definiti anche di 'Clustering'.

- **Regressione**

Algoritmi utili in casi in cui è necessario trovare una funzione continua che meglio approssima la relazione dei dati in ingresso con i rispettivi valori in uscita. Esistono vari tipi di regressione, la più semplice tra tutte è quella lineare in cui si cerca la miglior retta che interpola i dati usati per addestrare l'algoritmo. questa tipologia di risoluzione viene usata ad esempio per predire l'andamento di grandezze fisiche, recupero di dati mancanti, estrapolazione relazioni tra diverse grandezze fisiche e altro.

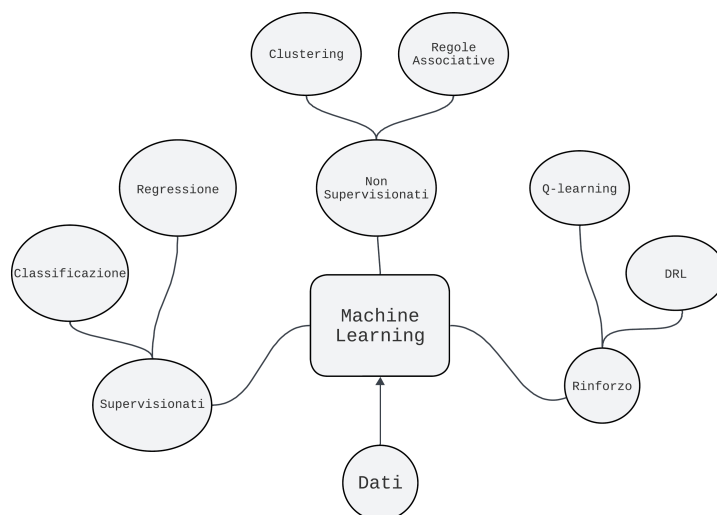


Figura 2.1: *Mapa delle principali suddivisioni degli algoritmi di ML*

2.2 Reti convoluzionali

Una rete neurale convoluzionale (CNN) è un tipo di rete di apprendimento profondo utilizzata principalmente in ambito di classificazione e riconoscimento di pattern all'interno di immagini. Queste reti si distinguono per la loro efficienza dovuta all'abilità di ridurre le dimensioni degli input da cui apprendono. Le Convolutional Neural Networks sono state progettate specificamente per mappare dati di immagini a variabili di output. Sono particolarmente abili nello sviluppare rappresentazioni interne di immagini bidimensionali che possono essere utilizzate per apprendere strutture invarianti rispetto a posizione e scala. Ciò li rende particolarmente adatti a lavorare con dati che presentano una componente di relazione spaziale.

Si chiamano reti convoluzionali perché sfruttano l'operazione di convoluzione per poter imparare. Questa operazione lineare coinvolge la moltiplicazione tra una matrice di pesi (chiamata filtro) con la matrice di input. I filtri vengono tarati in modo tale da rilevare una specifica caratteristica dell'input. Applicando lo stesso filtro per l'intera immagine si ottiene una mappa di attivazioni, chiamata anche feature map. Tale mappa indica la posizione di dove viene rilevato una determinata caratteristica impostata dal filtro e con quale intensità.

Le CNN hanno una struttura composta da 3 livelli: 1) Livello di input, 2) Livello nascosto e 3) Livello di output. I livelli nascosti a loro volta contengono strati convoluzionali, strati di pooling, strati di normalizzazione e strati completamente connessi.

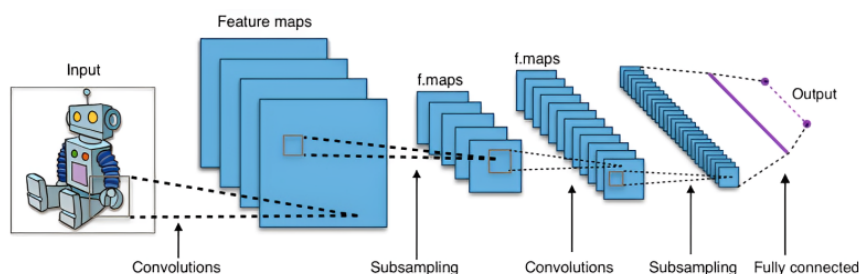


Figura 2.2: Schema di una rete neurale convoluzionale [1]

- Layer convoluzionale: In questo livello, gli input vengono elaborati attraverso una serie di filtri convoluzionali finalizzati all'estrazione delle caratteristiche (features). Dopo ogni passaggio di convoluzione la CNN introduce non linearità nel modello applicando una trasformazione ReLU (Rectified Linear Unit) alla mappa delle funzioni. La trasformazione ReLU è particolarmente apprezzata nelle reti neurali perché aiuta la rete a imparare relazioni complesse

nei dati, rendendo più agevole l'addestramento e migliorando le prestazioni complessive del modello.

- Layer di pooling: Lo strato di pooling riduce progressivamente le dimensioni spaziali della rappresentazione per un calcolo più efficiente. Opera su ciascuna feature map indipendentemente. Un approccio comune utilizzato nel pooling è il max pooling, in cui viene catturato il valore massimo di una sezione della matrice, riducendo il numero di valori necessari per il calcolo.¹
- Layer di normalizzazione: Dopo il layer convoluzionale, l'output potrebbe assumere valori non ammissibili per la rete. Pertanto, vengono applicati i layer di rettifica, che riconducono questi valori a un intervallo ammissibile.
- Layer completamente connesso: Noto anche come “fully-connected layer”, la sua funzione principale è quella di effettuare la classificazione o la regressione, in base alle informazioni estratte dalle fasi di convoluzione e pooling nelle reti neurali convoluzionali.

¹Esempio di max pooling = in una matrice 9x9 ogni ritaglio 3x3 viene trasformato in uno 1x1 che assume il valore massimo tra quelli del ritaglio iniziale, ottenendo quindi una matrice di dimensione 3x3.

2.3 U-Nets

La U-Net è una Fully Convolutional Network (FCN) sviluppata per applicazioni in campo medico: l'individuazione di tumori nei polmoni e nel cervello attraverso l'Image Segmentation [2].

Sono chiamate U-Net perché la loro struttura ricorda la forma della lettera U. La rete è composta da due parti principali: un percorso ascendente e un percorso discendente. Il percorso ascendente, chiamato anche encoder, è responsabile dell'estrazione delle caratteristiche dall'immagine di input. Il percorso discendente, o decoder, è responsabile dell'assemblaggio di queste caratteristiche per creare un'immagine di output, o meglio permette di localizzare queste features nelle immagini.

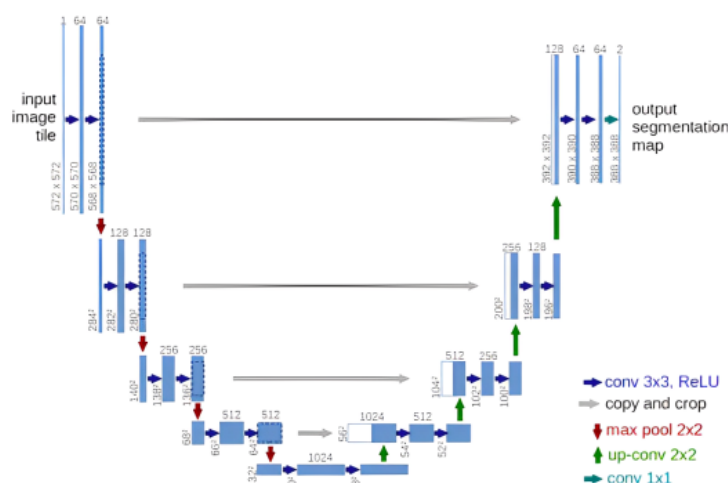


Figura 2.3: Esempio di una U-Net con featuremap multicanale [2]

Il ramo dell'encoder è simile a un classico layer convoluzionale di una Convolutional Neural Network (CNN). Tuttavia, a differenza delle CNN tradizionali, le FCN eliminano il layer completamente connesso e lo sostituiscono con il ramo del decoder. Quest'ultimo è simmetrico rispetto all'encoder e comprende operazioni convoluzionali, layer ReLU e layer di pooling che consentono l'upsampling per ottenere immagini delle stesse dimensioni dell'input.

Un aspetto fondamentale della U-Net è la connessione diretta e simmetrica tra il percorso ascendente (encoder) e il percorso discendente (decoder). Questa connessione permette una comunicazione senza soluzione di continuità tra le due parti

dell'architettura. Durante il processo di estrazione delle caratteristiche nell'encoder, le informazioni rilevanti vengono trasmesse al decoder. Questa connessione diretta consente al decoder di utilizzare dettagli precisi estratti dall'immagine originale per assemblare un'immagine di output che conserva la coerenza spaziale e le caratteristiche chiave. In questo modo, la rete può raggiungere risultati di segmentazione estremamente precisi e dettagliati, rendendo la U-Net ideale per applicazioni che richiedono la localizzazione accurata di oggetti o strutture nell'immagine.

Data la loro grande abilità nell'estrarre delle features, questa tipologia di reti può essere addestrata anche su campioni di immagini relativamente piccoli. Come mostrato nell'articolo "U-Net: Convolutional Networks for Biomedical Image Segmentation" [2], queste reti oltre ad essere veloci risultano anche più efficienti in dimensioni ridotte del dataset.

Capitolo 3

Ray Tracing

La propagazione dei segnali radio è un aspetto fondamentale delle comunicazioni wireless e gioca un ruolo critico nella progettazione e nell'ottimizzazione dei sistemi di comunicazione. La comprensione dei meccanismi di propagazione e dei relativi parametri è essenziale per garantire una comunicazione affidabile. In questo capitolo, esamineremo diversi parametri chiave associati alla propagazione dei segnali radio. Chiameremo d'ora in poi canale il mezzo di comunicazione tra un trasmettitore (Tx) e un ricevitore (Rx).

Lo scenario principale da cui si cominciano tutti gli studi è quello detto Spazio Libero, ovvero quando il collegamento tra Tx e Rx è di tipo LoS (Line of Sight, Linea di Vista). Una prima approssimazione per lo studio delle Onde Elettromagnetiche (E.M. o EM) è quella di porsi in un mezzo omogeneo e in condizioni di campo lontano. La seconda condizione è verificata per:

$$r \gg \lambda, \quad r \gg D, \quad r \gg D^2/\lambda \quad (3.1)$$

dove λ rappresenta la lunghezza d'onda, r la distanza tra il trasmettitore e un punto nello spazio e D la massima estensione dell'antenna.

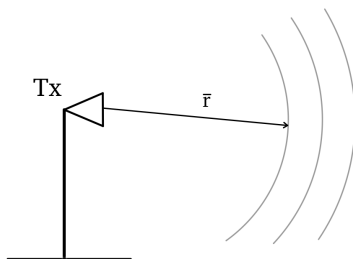


Figura 3.1: Antenna Tx che genera un'onda E.M.

A distanze sufficientemente grandi il campo generato dal trasmettitore viene considerato come un'onda quasi sferica, trascurando il fattore di attenuazione α il campo elettrico può essere espresso come:

$$\vec{E}(r, \theta, \phi) = \vec{E}_0(\theta, \phi) \frac{e^{-j\beta r}}{r} \quad (3.2)$$

Definendo $\eta = \sqrt{\frac{\mu}{\epsilon}}$ come l'impedenza intrinseca del mezzo possiamo semplicemente ricavare l'equazione del fronte d'onda del campo magnetico nel seguente modo:

$$\vec{H} = \frac{1}{\eta} \hat{i}_r \times \vec{E} \quad (3.3)$$

Possiamo ora definire il Vettore di Poynting, cioè la grandezza vettoriale che rappresenta la densità di potenza trasportata dall'onda E.M. nel seguente modo:

$$\begin{aligned} \vec{S} &= \frac{\vec{E} \times \vec{H}}{2} \\ &= \frac{|\vec{E}|^2}{2\eta} \end{aligned} \quad (3.4)$$

Un approccio più semplice allo studio della radiazione E.M. è quello della teoria a raggi. Questo metodo discretizza l'onda in fasci di raggi e studia le interazioni di ogni raggio con l'ambiente. Viene definito come raggio la curva della quale ogni punto risulta essere perpendicolare al fronte d'onda. In condizioni di spazio libero e mezzo di propagazione omogeneo, il raggio elettromagnetico risulta essere quindi un cammino rettilineo associato al fronte d'onda, e la radiazione di un'antenna trasmittente risulta essere un fascio proprio di raggi che ha come origine proprio il trasmettitore.

Si nota che direzione e verso del raggio sono uguali a quelle del vettore di Poynting associato all'onda. Solitamente infatti si dice anche che il raggio trasporta l'energia del campo EM.

Mediante legge di intensità che lega in maniera inversamente proporzionale la sezione di un tubo di flusso e l'intensità di potenza che lo attraversa, si definisce lo Spreading factor (SF) nel seguente modo:

$$A = \sqrt{\frac{|\vec{S}_2|}{|\vec{S}_1|}} = \sqrt{\frac{d\Sigma_1}{d\Sigma_2}} \quad (3.5)$$

Con Σ che rappresenta le basi del tubo di flusso che ha come superficie laterale i raggi che collegano due fronti d'onda. In particolare nel caso di onda sferica, data la proporzionalità tra distanza e dimensione della sezione del tubo di flusso, otteniamo:

$$A = \sqrt{\frac{d\Sigma_1}{d\Sigma_2}} = \frac{r_1}{r_2} \quad (3.6)$$

3.1 Interazioni delle onde

A differenza dello spazio libero dove le onde non incontrano ostacoli di nessun tipo nel loro cammino, in ambiente urbano questo non si verifica a causa delle interazioni con ogni tipo di ostacolo incontrato. Tra le principali riconosciamo le seguenti.

3.1.1 Riflessione e trasmissione

Assumendo localmente un'onda piana e considerando la superficie di interazione liscia, si può analizzare il fenomeno di riflessione di un'onda. Questo meccanismo di propagazione segue le leggi della Riflessione e quella di Snell.

Consideriamo i due mezzi in cui l'onda viaggia come omogenei con coefficienti di rifrazione rispettivamente n_1 e n_2 , e nominiamo con pedice i , r e t rispettivamente gli elementi di onda incidente, riflessa e trasmessa abbiamo che:

- Legge della Riflessione

$$\theta_i = \theta_r \quad (3.7)$$

- Legge di Snell

$$n_1 \cdot \sin\theta_i = n_2 \cdot \sin\theta_t \quad (3.8)$$

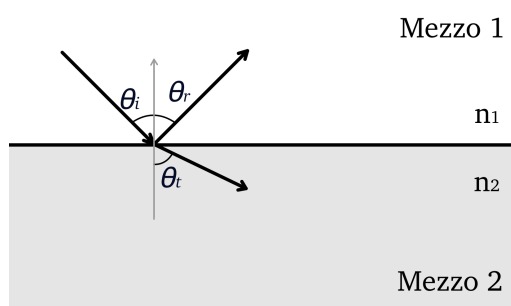


Figura 3.2: Fenomeno della riflessione e trasmissione di un'onda alla superficie di incidenza

3.1.2 Diffrazione

La diffrazione, un fenomeno fisico simile alla riflessione ma che si verifica quando un'onda incide sullo spigolo di un ostacolo. Ciò dà origine alla creazione di diversi raggi che si dispongono in un cono noto come cono di Keller, come rappresentato nella Figura 3.3. Ciascun raggio all'interno del cono forma un angolo θ_d con lo spigolo generatore. La legge della diffrazione, basata sulla legge di Snell e applicata a questo fenomeno, può essere definita come segue:

$$n_1 \cdot \sin\theta_i = n_2 \cdot \sin\theta_d \quad (3.9)$$

3.1.3 Scattering diffuso

Questo fenomeno, invece, è causato dalla rugosità della superficie di incidenza. Come abbiamo visto precedentemente, in caso di superficie perfettamente liscia avremo solo due raggi generati da quello incidente: il riflesso e il trasmesso. Nel mondo reale, ogni superficie presenta piccole imperfezioni dovute sia alla natura dei materiali che a depositi di polvere, usura nel tempo, etc. Queste micro imperfezioni generano, oltre alla riflessione speculare secondo la legge di Snell, numerosi raggi in direzioni diverse, dando origine al fenomeno di scattering diffuso.

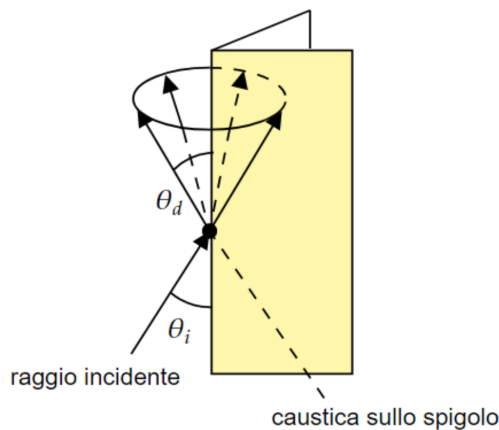


Figura 3.3: Cono di Keller

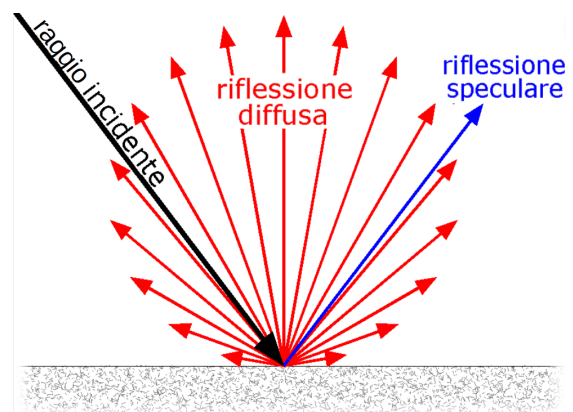


Figura 3.4: Scattering

3.2 Propagazione multicammino

Nell'ambiente urbano, i raggi generati da un'antenna trasmittente interagiscono con l'ambiente circostante in molteplici modi. Si verifica un fenomeno noto come multicammino, in cui i segnali radio percorrono diversi cammini dalla sorgente alla destinazione a causa di riflessioni, diffrazioni e rifrazione. Queste interazioni danno origine a onde elettromagnetiche di diversa natura che si sovrappongono tra loro. Di conseguenza, il campo misurato in un punto specifico dello spazio non è più semplicemente il risultato dell'onda iniziale precedentemente definita, ma piuttosto una combinazione coerente dei campi associati a ciascun singolo percorso.

Ogni canale radio può essere descritto da una funzione di trasferimento equivalente passa-basso. Consideriamo le due ipotesi di canale statico, cioè con proprietà invarianti nel tempo, e di antenne, amplificatori e modulatori/demodulatori (modem) non distorcenti. Sotto queste ipotesi possiamo ricavare una risposta impulsiva del canale:

$$h_0(t) = \sum_{i=1}^{N_r} \rho_i \delta(t - t_i) e^{j\theta_i - 2\pi f_0 t_i} \quad (3.10)$$

Dove ρ_i rappresenta l'attenuazione dovuta alla propagazione in spazio libero e alla interazione con gli ostacoli, t_i il ritardo dovuto al tempo di propagazione e θ_i lo sfasamento iniziale della portante.

Se chiamiamo ξ come il tempo relativo al cammino principale (LoS), una risposta impulsiva ideale del canale $h_i(\xi)$ e $h_a(\xi)$ la risposta impulsiva di antenne, amplificatori e mo-dem otteniamo la risposta del sistema:

$$h(\xi) = h_i(\xi) * h_a(\xi), \quad (3.11)$$

3.2.1 Power-delay profile

Il Power Delay Profile (PDP) è una misura che descrive come la potenza del segnale radio varia nel tempo a causa delle riflessioni e dei ritardi nei percorsi di propagazione. Il PDP fornisce informazioni preziose per la progettazione dei sistemi di comunicazione e la compensazione degli effetti del multipath. Grazie alla risposta impulsiva del canale possiamo calcolare il power-delay profile come:

$$p(\xi) = \frac{|h(\xi)|^2}{\int |h(\xi)|^2 d\xi} \quad (3.12)$$

3.2.2 Delay Spread

Tale fenomeno è dovuto al diverso tempo di arrivo di ciascun cammino rispetto al cammino più veloce (solitamente determinato dalla LoS). Infatti, anche nel caso più semplice dove consideriamo una sola riflessione con il terreno, la distanza percorsa dai vari cammini è diversa e, nell'ipotesi di mezzo omogeneo, avendo la stessa velocità alcuni raggi arriveranno prima di altri. Questo viene solitamente chiamato anche eco. Semplificando un po' possiamo dire che il delay spread (DS) è la differenza dei tempi tra il cammino più veloce e il più lento.

3.2.3 Angle spread

L'ampiezza angolare (Angle Spread, AS) dei canali wireless si riferisce alla distribuzione di potenza su una serie di angoli di incidenza ed è causata dalla propagazione multicammino. Solitamente, questo parametro è espresso come una percentuale dell'angolo solido, indicando la porzione dell'angolo solido da cui provengono i cammini più significativi. Possiamo considerare l'angle spread come l'equivalente del delay spread nello spazio. Questo parametro riveste una notevole importanza, poiché se i componenti più rilevanti del campo provengono da direzioni diverse, potrebbero interagire in maniera distruttiva tra loro, causando una distorsione nel segnale ricevuto. Bisogna mantenere perciò questo valore il più basso possibile.

Considerando $\hat{k} = \hat{k}(\theta, \phi)$ il versore che unisce l'origine con un punto nello spazio in posizione (r, θ, ϕ) e $p_k(\hat{k})$ l'analogo del PDP ma per l'angolo di arrivo dei raggi, possiamo scrivere in formule l'AS nel caso 2D come:

$$AS^{2D} = \sigma_k = \sqrt{\int_{4\pi} |\hat{k} - \langle \hat{k} \rangle|^2 p_k(\hat{k}) d\Omega} = \sqrt{1 - |\langle \hat{k} \rangle|^2} \quad (3.13)$$
$$\langle \hat{k} \rangle = \int_{4\pi} \hat{k} p_k(\hat{k}) d\Omega$$

3.3 Utilizzo dell'applicativo

Per l'analisi della propagazione in ambiente urbano è stato utilizzato un programma di Ray-Tracing (RT). In particolare, il software di previsione della propagazione deterministica si basa sull'Ottica Geometrica (GO) e sulla Teoria Uniforme della Diffrazione (UTD). Nella metodologia GO/UTD, durante la propagazione dei raggi, essi interagiscono con le strutture geometriche presenti nell'ambiente, compiendo riflessioni, diffrazioni e trasmissioni. Gli effetti di tali interazioni vengono calcolati mediante l'utilizzo dei relativi coefficienti di riflessione, diffrazione e trasmissione, calcolati in base alle proprietà dei materiali di ciascun oggetto. Gli effetti di scattering possono essere considerati anche in relazione a percorsi aggiuntivi che si diffondono dalle superfici caratterizzate da rugosità in rapporto alla lunghezza d'onda.

Il programma di RT utilizzato è un eseguibile lanciato tramite Matlab, esso opera in due principali step:

- Un primo step geometrico in cui traccia tutte le traiettorie tra il trasmettitore e i ricevitori con un numero di interazioni massime impostate dall'utente.
- Uno step elettromagnetico dove calcola l'evoluzione del campo emesso dall'antenna Tx per ogni raggio fino al raggiungimento dell'antenna Rx.

Per poter utilizzare questo software sono necessari dei file strutturati in un modo ben preciso che descrivano le caratteristiche geometriche ed elettromagnetiche dell'ambiente. Riportiamo di seguito spiegazioni ed esempi dei principali file necessari per utilizzare l'applicativo.

MAP file Il file principale, con estensione *.map*, contiene le informazioni cruciali sulla posizione dei vari file necessari. Al suo interno sono elencati gli indirizzi di tali file, che contengono le informazioni richieste per la simulazione. La struttura di questo file deve seguire il seguente formato:

```
#----- Building information -----  
  
# Name of the file with the buildings geometrical information ("Vector File" PLANET)  
  
Building file: address of the file buildings.bld  
  
# Name of the file with rooftop building heights ("Attribute File" PLANET).  
# Heights can be absolute or relative to the terrain.
```

```

# In any case, the choice needs to be consistent with the ABSOLUTE_H flag
# contained in param.dat file and terrain database.

Height max file: address of the file building_roof_rel.atr or building_roof_abs.atr

# Name of the file with floor building (relative) levels
#"none" = not available

Height min file: address of the file building_floor.atr

# Name of the file containing electromagnetic characteristics of walls

Electromagnetic file: address of the file buildings.elm

#----- Terrain information -----

# Terrain file name (binary). "none" = not available
# Pixel resolution (meters). "none" = not available

Terrain file:                address of the file terrain.bin if available
X resolution:                resolution along x axis [m]          if available
Y resolution:                resolution along y axis [m] if available

# Size of terrain database. "none" = not available

X_MIN:                       smallest x coordinate
X_MAX:                       largest x coordinate
Y_MIN:                       smallest y coordinate
Y_MAX:                       largest y coordinate

# Byte order in DTM binary file. "M" = Motorola (BIG ENDIAN);
# "I" = Intel (LITTLE ENDIAN); "none" = not available

BYTE_ORDER:                  byte order: M, I or none

```

BLD file Il file *.bld* contiene le coordinate dei vertici dei poligoni che costituiscono la descrizione geometrica 2D degli edifici. Ciascun blocco deve essere descritto nel seguente modo:

Building ID	Type of surface	Number of vertices
Xv1	Yv1	
Xv2	Yv2	
...	...	
Xvn	Yvn	

Nota bene che l'ultima coordinata del vertice deve corrispondere con la prima per chiudere il poligono

ROOF_ABS.ATR file Tale file contiene le altezze assolute [m] degli edifici definiti nel file .bld. Ecco un esempio:

```
1 "building" height_bld_1
2 "building" height_bld_2
...      ...      ...
n "building" height_bld_n
```

ROOF_REL.ATR file Questo file è simile a quello precedente, ma contiene altezze in [m] relative al terreno.

FLOOR.ATR file Il seguente file contiene la quota dei pavimenti [m] degli edifici definiti nel file .bld. Di seguito un esempio:

```
1 "building" floor-level_bld_1
2 "building" floor-level_bld_2
...      ...      ...
n "building" floor-level_bld_n
```

ELM file Questo file contiene le proprietà elettromagnetiche delle superfici nel contesto in termini di permittività relativa e conducibilità. Il tipo di interazioni abilitate per ciascuna parete è definito anche secondo un codice (r=riflessione, d=difrazione, s=dispersione, n=nessuna). Inoltre, è possibile aggiungere come parametri opzionali il coefficiente di scattering diffuso S_R, lo spessore della parete e il coefficiente di trasmissione diffuso S_T. La struttura per un edificio con n pareti è la seguente:

Building ID	Type of surface	Number of walls
σ_1 ϵr_1	enabled interaction (S_R)	(thickness) (S_T)
σ_2 ϵr_2	enabled interaction (S_R)	(thickness) (S_T)
...
σ_n ϵr_n	enabled interaction (S_R)	(thickness) (S_T)

TX file Il file .tx contiene le informazioni relative al trasmettitore in termini di posizione, potenza di trasmissione, frequenza e caratteristiche di radiazione. Di seguito, la struttura:

```

Position:      x   y   z   coordinates [m] of the transmitter
Power:         transmitted power [dBm] or [dBW]
Frequency:     [MHz]
Radiation File: address of the .ant file with Tx antenna radiation pattern
Angle 1:        $\alpha$ Za rotation angle following convention defined in Appendix A2
Angle 2:        $\alpha$ Ya rotation angle following convention defined in Appendix A2
Angle 3:        $\alpha$ Xa rotation angle following convention defined in Appendix A2

```

RX file Il file .rx contiene informazioni sui ricevitori in termini di numero, indirizzo e delle caratteristiche di radiazione dell'antenna. La struttura di base è per posizioni multiple dei ricevitori su una linea retta ed è la seguente:

```

Point A:      x   y   z   coordinates [m] of the first Rx point
Point B:      x   y   z   coordinates [m] of the last Rx point
RX Number:    number of Rx positions in the line
Radiation File: address of the .ant file with Rx antenna radiation pattern
Angle 1:       $\alpha$ Za rotation angle following convention defined in Appendix A2
Angle 2:       $\alpha$ Ya rotation angle following convention defined in Appendix A2
Angle 3:       $\alpha$ Xa rotation angle following convention defined in Appendix A2

```

ANT file Per simulare la propagazione in un ambiente tridimensionale, è importante utilizzare una descrizione tridimensionale dei diagrammi di radiazione dell'antenna, sia sul lato di trasmissione che sul lato di ricezione, inclusa l'informazione sulla polarizzazione. Purtroppo, non sempre è possibile disporre di un diagramma di radiazione tridimensionale completo dell'antenna. Pertanto, è possibile utilizzare anche diagrammi classici nei due tagli principali (piano E e piano H) e ricavare da essi il diagramma di radiazione tridimensionale. In base a ciò, ci sono due formati possibili per il file .ant da utilizzare come input: E_H e 3D. Nel nostro caso, il formato è il seguente:

```

#Antenna_File_Format: [3D or E_H]
3D

#Sampling Information:
#Frequency[MHz]:
start freq.      n° of steps      stop freq.

#Theta[deg]
1st angle      n° of steps      last angle

```

```

#Phi[deg]
1st angle          n° of steps          last angle

# For each frequency the data structure is repeated identically
# EThij stands for the field in the direction (thetai,phij) with i=0,...,M j=0,...,N
# Theta angles are scanned first
# after each theta full iteration is completed phi step is increased

Freq_id      Frequency[MHz]      Max_Gain[dBi]      Max_Field_Strength[V/m]
Abs(ETH00) [V/m]  Phase(ETH00) [rad]  Abs(EPh00) [V/m]  Phase(EPh00) [rad]
Abs(ETH10) [V/m]  Phase(ETH10) [rad]  Abs(EPh00) [V/m]  Phase(EPh10) [rad]
...          ...
Abs(ETHM0) [V/m]  Phase(ETHM0) [rad]  Abs(EPhM0) [V/m]  Phase(EPhM0) [rad]
Abs(ETH01) [V/m]  Phase(ETH01) [rad]  Abs(EPh01) [V/m]  Phase(EPh01) [rad]
...          ...
Abs(ETHM1) [V/m]  Phase(ETHM1) [rad]  Abs(EPhM1) [V/m]  Phase(EPhM1) [rad]
...          ...
Abs(ETHij) [V/m]  Phase(ETHij) [rad]  Abs(EPhij) [V/m]  Phase(EPhij) [rad]
...          ...
Abs(ETHMN) [V/m]  Phase(ETHMN) [rad]  Abs(EPhMN) [V/m]  Phase(EPhMN) [rad]

```

Tutte le informazioni sopra citate sono state prese dalla guida utente del file di RT, per ulteriori approfondimenti fare riferimento ad essa.

3.4 Algoritmo di semplificazione della mappa

Per generare un dataset consistente sono state usati dei ritagli di mappe esistenti di grandi città. Per poter generare queste mappe si è potuto usare un algoritmo di semplificazione già presente tra gli eseguibili del tool di RT. Questo eseguibile è nato per poter fare delle simulazioni semplificate in uno scenario molto grande (ad es. la città di Bologna) dove sono presenti tantissimi edifici che renderebbero le simulazioni molto onerose.

Il file di semplificazione sceglie nella mappa degli edifici o ostacoli solo quelli che sono interessati dalla propagazione. Ovvero vengono presi solo quegli edifici o ostacoli che hanno maggiore probabilità di interagire con il campo EM generato dal trasmettitore e che verrà ricevuto dal ricevitore. Possono essere quindi definite tre diverse categorie di edifici cosiddetti “attivi”, cioè:

- A. Gli edifici situati all’interno dell’ellisse avente i fuochi posti sulla coppia ricevitore-trasmettitore. Questi edifici sono quelli più vicini alla coppia Tx-Rx e di conseguenza è molto probabile che verranno coinvolti in riflessioni importanti necessarie alla comunicazione tra i due terminali. È a cura di chi



Figura 3.5: Esempio di edifici “attivi” secondo il parametro A dove in verde viene rappresentata la mappa completa e in rosso quella semplificata [3]

ha progettato tale algoritmo dimensionare l’eccentricità dell’ellisse per avere il valore ottimo di edifici di questa categoria

- B. Un altro importante insieme di edifici sono quelli che si trovano in LoS tra Tx e Rx, generalmente questi edifici sono coinvolti nella generazione di interazioni del primo o secondo ordine.
- C. Gli ultimi edifici sono quelli che appartengono alla categoria di oggetti molto alti e opportunamente ruotati. Questi sono particolarmente importanti nel caso di copertura macro-cellulare dove il trasmettitore si trova oltre l’altezza del tetto, perciò è molto probabile che un edificio opportunamente orientato generi una riflessione importante per la propagazione del segnale.

Questa tecnica permette di velocizzare in maniera significativa l’esecuzione delle simulazioni di RT senza perdere di significativa qualità. Ma nel nostro caso questo algoritmo è stato utilizzato per generare dei ritagli di mappa a partire da una più grande. Sono stati impostati quindi i parametri necessari all’eseguibile che genera i ritagli per delimitare le porzioni di mappa di interesse in modo tale da ottenere in output i file necessari per eseguire le simulazioni. Approfondiremo meglio l’argomento nel capitolo successivo.

Capitolo 4

Generazione del dataset e addestramento del modello

4.1 Generazione del dataset

Nel processo di sviluppo di un sistema di previsione della propagazione radio in ambienti urbani, una fase cruciale consiste nella generazione del dataset di addestramento per l'algoritmo di Machine Learning (ML). Questo dataset permette all'algoritmo di apprendere le relazioni tra le condizioni ambientali e i risultati delle simulazioni di propagazione radio.

Per la previsione di tali scenari, è stata adottata una tecnica di segmentazione di immagini, in particolare si è usata una U-Net. Come discusso nel capitolo 2, le U-Nets sono reti completamente convoluzionali particolarmente adatte al compito specifico di segmentazione. Essendo algoritmi supervisionati, richiedono perciò un dataset etichettato.

Per generare i file di addestramento, si è optato per la tecnica di Ray-Tracing (RT), essenziale per produrre le mappe di copertura necessarie in tempi ragionevoli. Sfruttando scenari geometrici ampi, come le mappe di grandi città, sono state creati nuovi scenari di dimensioni ridotte. Si sono scelte queste dimensioni per concentrarsi sulla copertura microcellulare¹, evitando un'eccessiva complessità computazionale legata allo studio di aree estese. Il funzionamento del sistema creato per generare questi scenari ritagliati è molto semplice e segue il principio logico descritto in Figura 4.1.

¹La copertura urbana viene definita microcellulare quando l'antenna trasmittente si trova ad un'altezza mediamente inferiore a quella dei tetti delle abitazioni, in particolare quando ha anche una copertura ridotta di circa qualche centinaio di metri

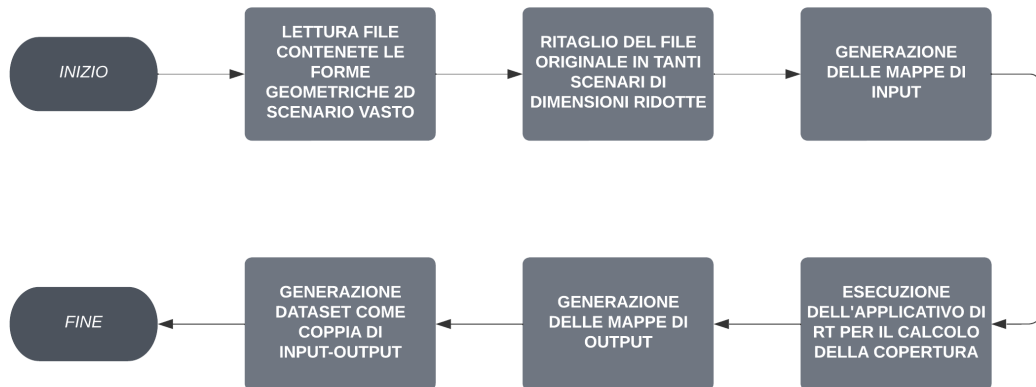


Figura 4.1: Diagramma a blocchi del processo di generazione di mappe

4.1.1 Ritaglio in mappe più piccole

Per eseguire il ritaglio degli ambienti urbani, è stato utilizzato l’eseguibile precedentemente menzionato nel paragrafo “*Algoritmo di ritaglio*” (vedi Cap. 3). Inizialmente, sono state estratte tutte le informazioni necessarie, tra cui la posizione degli edifici, le loro dimensioni e l’estensione massima della mappa di origine. La scelta è ricaduta sulla generazione di mappe con dimensioni massime di circa 600 metri, tenendo presente che la copertura di una microcella solitamente ha un diametro attorno a tali valori. Con queste informazioni, è stato configurato un ciclo automatico che adatta le impostazioni da fornire all’eseguibile. I parametri da impostare nel file “building.city_param.dat” comprendono le coordinate minime e massime dello scenario, che vanno compilati con valori reali in metri seguendo un sistema di coordinate UTM o equivalente, e il parametro ENABLE_SIMPLIFICATION. Quest’ultimo abilita l’algoritmo di semplificazione, impostato su ‘0’ per evitare la selezione di ostacoli particolarmente lontani. A seguire, viene riportato un estratto del codice per maggiore chiarezza.

```

XMIN_MAP 0.0
XMAX_MAP 0.0
YMIN_MAP 0.0
YMAX_MAP 0.0
ENABLE_SIMPLIFICATION 0
  
```

È stato poi eseguito un controllo sul numero minimo di edifici, o più in generale ostacoli, presenti in un singolo ritaglio. Questo per mantenere ambienti urbani comunque popolati e non creare scenari poco popolati dove la previsione ricadrebbe

semplicemente nel caso di spazio libero, o comunque di poco interesse dal punto di vista propagativo.

4.1.2 Generazione delle mappe di input

Ci occupiamo ora della generazione dei file ausiliari per creare il dataset. In base alla formattazione definita nel Capitolo 3, si sono generati i file del trasmettitore per ogni scenario e il file contenente il grigliato di ricevitori. Una volta note le dimensioni e ottenuti i file dall'eseguibile che si occupa del ritaglio si sono potute creare le mappe di input per la rete convoluzionale.

Come forma di rappresentazione delle informazioni è stata scelta la matrice. Scelta effettuata per la loro maggiore flessibilità in fase di post-processing. Ogni elemento della matrice rappresenta quindi un pixel dell'immagine seguendo una particolare codifica. Per le matrici di input della rete convoluzionale il trasmettitore viene rappresentato con il valore '3', gli edifici con il valore '2' e il terreno con il valore '0'. Mentre per quelle di output cambia il significato in base al tipo di problema che si vuole affrontare, per esempio in caso di analisi di presenza o meno di LoS il valore '0' è riservato all'assenza di campo e il valore '1' alla presenza. Di seguito verrà riportato un esempio di una coppia di matrici di input e output del medesimo scenario.

4.1.3 Simulazioni RT e generazione delle mappe di output

Sono poi state fatte le simulazioni Ray Tracing su tutti gli scenari impostando il numero di interazioni con l'ambiente desiderato. Per un primo approccio si è voluto studiare il problema della linea di vista, si impostano i parametri necessari perciò in modo tale da non considerare riflessioni, diffrazioni o altri fenomeni del multipath. Una volta completate tutte le simulazioni è nota la potenza ricevuta da ciascun ricevitore e, grazie a ciò, si possono costruire anche le mappe di output. Si faccia riferimento alla Figura 4.2 per un esempio.

Durante la fase di generazione, potrebbero verificarsi arrotondamenti che portano alla presenza di elementi errati nelle matrici, generando potenziale rumore durante l'addestramento. Un altro aspetto critico identificato riguarda le dimensioni del trasmettitore. Inizialmente, si potrebbe pensare di riservare un singolo pixel con coordinate (x_{tx}, y_{tx}) per rappresentare il trasmettitore sulla mappa; tuttavia, questa informazione potrebbe andare persa durante l'addestramento. Ciò è dovuto alle fasi di down-sampling, che comportano la perdita di informazioni specifiche di ciascun pixel. Per risolvere questo problema, è sufficiente aumentare

le dimensioni dell'antenna tx sulla mappa. Al fine di affrontare questi dettagli, è stata implementata una fase di post-processing su tutte le matrici.

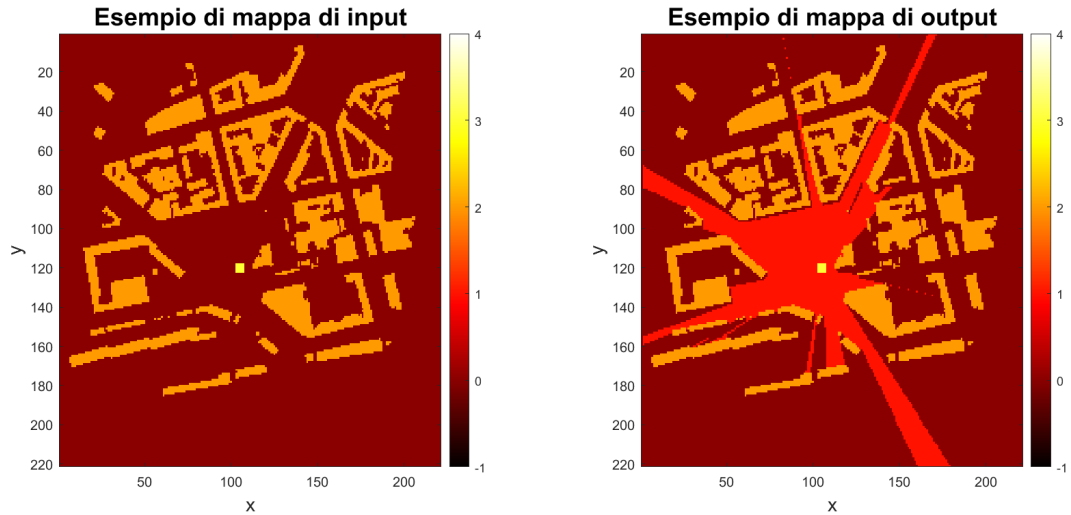


Figura 4.2: *Esempio di coppia di mappe input-output del dataset*

4.1.4 Organizzazione del dataset

Una volta acquisite tutte le coppie di ingresso e uscita, è essenziale organizzare il dataset in modo appropriato per l'addestramento dell'algoritmo. Tradizionalmente, il dataset viene suddiviso in tre parti: una per la fase di training, una per la fase di test e un'ultima per la fase di valutazione. Una suddivisione comune potrebbe essere basata su proporzioni come 70-80% per il training e 10-20% per le fasi di test e valutazione.

Nel nostro caso specifico, a causa della limitata disponibilità di dati, si è optato per una suddivisione del 98-12-0. Questa scelta dedica la maggior parte delle coppie input-output alla fase di addestramento e il resto per quella di test, obbligando l'utilizzo delle medesime mappe anche per la fase di valutazione. È importante notare che in futuro è sempre possibile aumentare la dimensione del dataset con ulteriori simulazioni, per ora non è stato possibile farlo per ragioni di tempo.

4.2 Addestramento del modello

Per implementare il modello di machine learning, è stato utilizzato il framework di sviluppo Pytorch. Grazie alle sue librerie integrate, che operano su matrici multidimensionali definite come classi tensor, è possibile effettuare calcoli come somme e prodotti tra elementi con tecniche ottimizzate, riducendo i tempi di calcolo. Questo framework offre inoltre la possibilità di applicare trasformazioni durante il pre-processing dei dati e semplifica la creazione di nuovi dataset.

Per quanto riguarda la fase di addestramento del modello, si è concentrati sulla struttura del codice. Dopo l'importazione delle librerie necessarie come Pytorch e Numpy, è stata definita una classe "LoS_Dataset". Questa classe ottimizza il caricamento delle mappe di input e delle relative maschere di copertura, consentendo al calcolatore di caricare i file man mano che sono richiesti dal modello, evitando la necessità di memorizzare tutti i file fin dall'inizio.

Per adattare i dati a un problema di classificazione binaria, è stata applicata una trasformazione alla maschera, impostando a 0 tutti gli elementi che assumono un valore diverso da 1, dove 1 indica la presenza di Linea di Vista (LoS). Inoltre, sono state applicate operazioni di ritaglio per ridurre le dimensioni della maschera su cui l'algoritmo si deve addestrare, insieme a ribaltamenti orizzontali e/o verticali casuali.

Sono state definite variabili di dataset e dataloader per le diverse fasi di addestramento e valutazione richieste alla rete. La struttura del modello è stata definita utilizzando una U-Net, come descritto nel Capitolo 2, con quattro strati di Down e relativi strati di Up.

La funzione di addestramento è stata implementata per ottimizzare la funzione di perdita durante il processo di apprendimento. In ogni epoca, vengono valutate la loss sia per la fase di training che per la fase di valutazione, al fine di ottimizzare i pesi dei vari strati della rete. Il modello viene salvato insieme ai relativi pesi per un utilizzo futuro senza la necessità di riaddestrare la rete.

Infine, il modello è stato testato su alcune immagini del dataset per valutare le prestazioni, sia visualmente attraverso plot dei tensori ottenuti confrontati con quelli reali ottenuti con il Ray Tracing (RT), sia mediante il calcolo della metrica F1, trattata in dettaglio nel capitolo successivo. Per generare i grafici della Loss Function, sono stati caricati tutti i dati online tramite un API chiamato W&B, che consente di tenere traccia degli sviluppi del modello, tra le altre funzionalità.

Capitolo 5

Analisi dei risultati

Il processo di addestramento del modello per la classificazione binaria tra Linea di Vista (LoS) e Non Linea di Vista (NLoS) ha fornito risultati significativi, e in questo capitolo esamineremo le principali osservazioni e conclusioni emerse. Per arrivare a queste conclusioni sono stati analizzati due principali parametri: la “cross-entropy loss” e la “F1 Binary score”.

5.1 Parametri di valutazione

5.1.1 Cross-Entropy Loss

La cross-entropy loss, “perdita di entropia incrociata” in italiano, è una funzione di perdita comunemente utilizzata nei problemi di classificazione, in particolare in quelli in cui ci sono più classi. Esistono due principali varianti di cross-entropy loss: la “Binary Cross-Entropy Loss” e la “Categorical Cross-Entropy Loss”.

La Categorical Cross-Entropy Loss è utilizzata in problemi di classificazione multi-classe, dove esistono più di due classi da riconoscere. Per il calcolo viene utilizzata la seguente formula

$$L(y, \hat{y}) = - \sum_i y_i \cdot \log(\hat{y}_i) \quad (5.1)$$

Dove y_i è l’indicatore della classe reale (1 se l’istanza appartiene alla classe i , altrimenti 0) e \hat{y}_i è la probabilità predetta dal modello per la classe i .

Quando le classi di categorizzazione degenerano in una viene usata la Binary Cross-Entropy Loss (BCE Loss). Utilizzata quindi per problemi di classificazione binaria, dove bisogna definire se l’elemento appartiene o meno alla classe. La

formula della BCE Loss è la seguente:

$$L(y, \hat{y}) = -[y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})] \quad (5.2)$$

Dove y è l'etichetta vera (0 o 1) e \hat{y} è la previsione del modello, una probabilità compresa tra 0 e 1.

In entrambe le formule, l'obiettivo è minimizzare la loss. Quando il modello fa previsioni accurate, la loss si avvicina a zero. La funzione di perdita incoraggia il modello a fare previsioni con probabilità alte per le classi corrette e basse per le classi errate. Per il primo approccio di LoS o NLoS si può applicare quindi una cross-entropy loss binaria visto che le due classi sono complementari.

5.1.2 F1 Binary score

L'F1 score è una metrica di valutazione delle prestazioni utilizzata comunemente in problemi di classificazione binaria, ma può essere estesa anche a problemi di classificazione multi-classe. È particolarmente utile quando c'è uno sbilanciamento tra le classi, cioè quando una classe ha molte più istanze dell'altra.

L'F1 score è basato su due altre metriche: la precisione (precision) e il richiamo (recall), che sono definite come segue:

- **Precisione:** Misura la frazione di istanze positive predette correttamente rispetto al totale delle istanze predette come positive. In formule possiamo scriverlo come:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (5.3)$$

- **Richiamo (Sensibilità):** Misura la frazione di istanze positive predette correttamente rispetto al totale delle istanze effettivamente positive. La formula è:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (5.4)$$

L'F1 score combina precisione e richiamo in un singolo valore che tiene conto sia dei falsi positivi che dei falsi negativi. La formula dell'F1 score è la seguente:

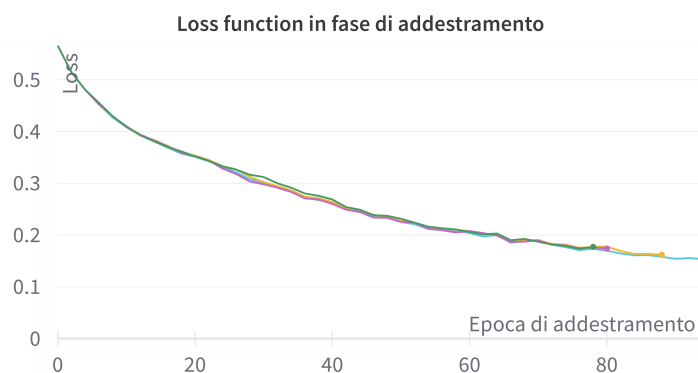
$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5.5)$$

L'F1 score restituisce quindi un valore compreso tra 0 e 1, dove 1 indica una perfetta precisione e richiamo e 0 indica che almeno una di queste metriche è 0.

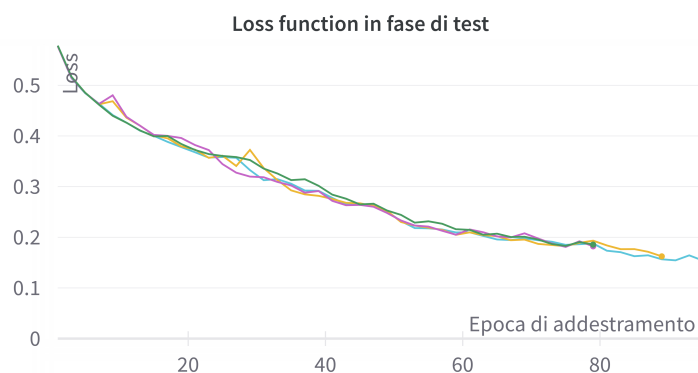
L'utilità dell'F1 score diventa evidente quando c'è uno sbilanciamento tra le classi, poiché in situazioni in cui una classe è molto più frequente dell'altra, la precisione da sola potrebbe essere ingannevole. Ad esempio, se la classe positiva è molto rara, un classificatore che predice sempre la classe negativa otterrebbe un'alta precisione, ma il suo richiamo sarebbe basso. Analogamente per situazioni con bassa precisione ma alto richiamo. L'F1 score fornisce perciò un modo di bilanciare queste due misure.

5.2 Risultati

Si riportano di seguito il grafico dell'andamento della cross-entropy loss e alcune immagini di confronto tra ground truth e predizione del modello ottenuto.



(a) Fase di addestramento



(b) Fase di test

Figura 5.1: Andamento della loss function al passare dei cicli di addestramento

L'andamento della loss sia sul set di addestramento che su quello di test mostra una decrescita coerente, indicando che le decisioni apprese durante l'addestramento sono in grado di generalizzarsi bene anche al test. Ciò suggerisce che il modello è in grado di discriminare tra LoS e NLoS in maniera affidabile su nuovi dati.

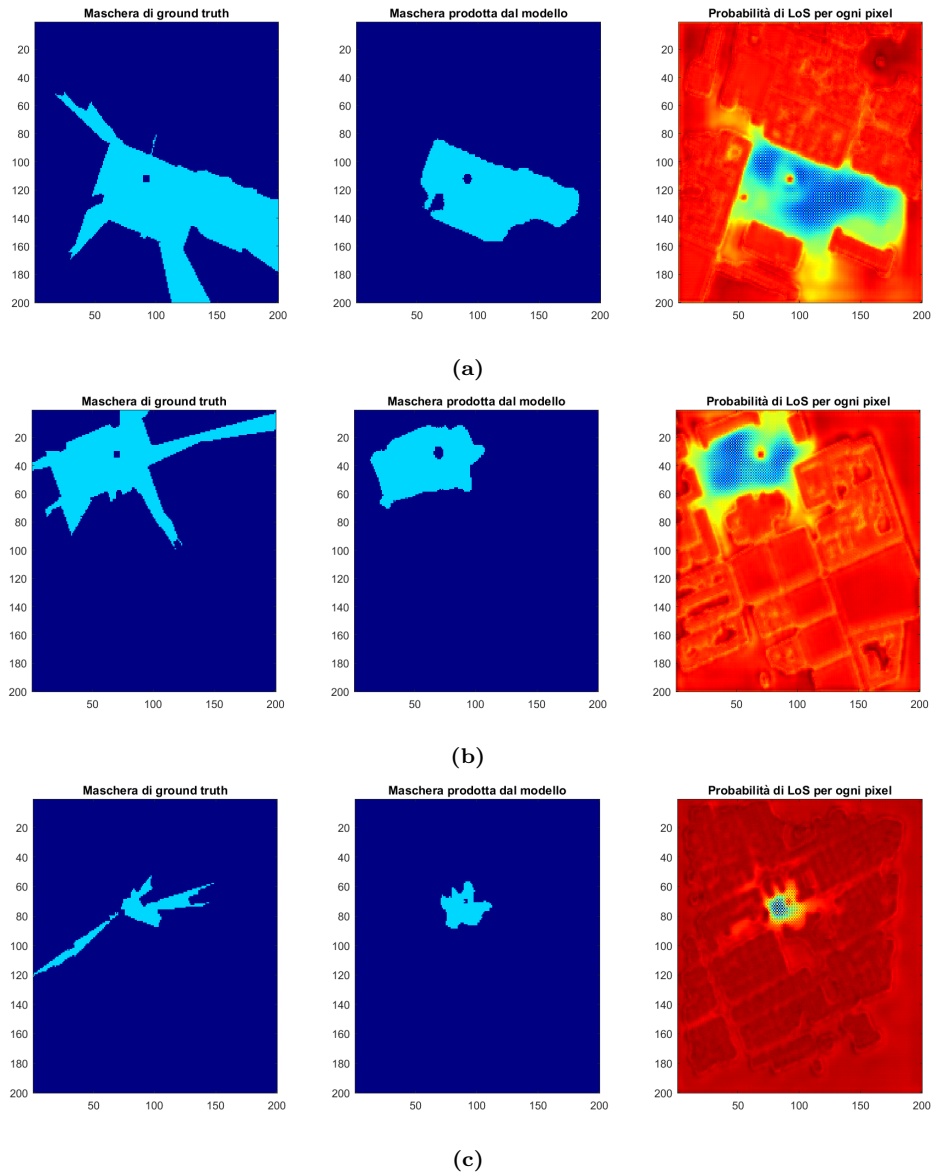


Figura 5.2: Tre esempi di maschere ottenute con il modello addestrato

Come si può notare dalla figura sopra riportata, il modello presenta alcune difficoltà nella corretta propagazione del segnale su lunghe distanze. Questo è

un problema noto delle reti convoluzionali, in quanto per stabilire la linea di vista, il modello deve propagare l'informazione di pixel in pixel della trasmittente fino all'ultimo punto della mappa dove c'è LoS. Questa osservazione suggerisce che potrebbero essere necessarie modifiche o tecniche specifiche per migliorare la gestione delle lunghe distanze nel modello.

Il risultato positivo che si cercava invece, emerge riguardo alla capacità del modello di riconoscere la presenza di edifici e la conseguente interruzione della linea di vista. Questa caratteristica è cruciale per l'applicazione del modello nel contesto delle reti di comunicazione wireless in ambienti urbani, dove la presenza di edifici è un fattore determinante nella propagazione del segnale.

Inoltre, il modello ha dimostrato di apprendere correttamente che il segnale ha origine dalla trasmittente. In tutte le predizioni, la probabilità di LoS più alta è stata assegnata all'area circostante alla trasmittente. Questa è un'altra capacità molto importante per l'applicazione del modello in ambito wireless.

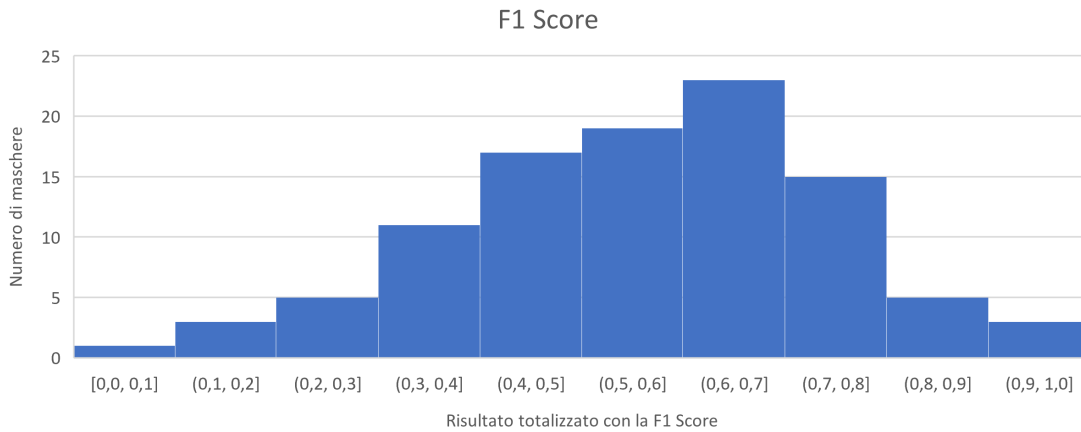


Figura 5.3: *Istogramma valori F1*

Dai test eseguiti con la F1 è risultato che il modello deve ancora migliorare un po', come detto poco prima in particolare a dare fastidio sono le propagazioni in percorsi lontani e stretti. In questo caso queste reti non riescono a continuare la propagazione, generando errore e di conseguenza falsi negativi. Per questo motivo la F1 Score presenta dei valori che arrivano anche fino a 0.95 ma con una media di soli 0.56. Proprio perché in alcuni scenari in cui il trasmettitore ha un'area circostante abbastanza ampia il modello ha prestazioni molto buone, mentre appunto nei casi particolari di cui discutevamo prima non riesce ancora ad ottenere risultati sufficienti.

5.3 Conclusioni

L'obiettivo primario di questo lavoro non è stato la risoluzione completa del problema di previsione della copertura in ambiente urbano, ma piuttosto la presentazione di un approccio basato sui dati come possibile soluzione. Questo primo tentativo ha aperto la strada a una comprensione più approfondita delle dinamiche della propagazione radio in contesti urbani mediante tecniche di machine learning.

In conclusione, nonostante alcune sfide legate alla propagazione del segnale a lunga distanza, il modello ha dimostrato una buona capacità di discriminare tra Linea di Vista e Non Linea di Vista. Esso mostra buone performance in ambienti meno densamente urbanizzati, con un numero moderato di edifici. Pertanto, potrebbe trovare utilizzo in scenari come piccole città, comuni rurali o quartieri urbani meno densi.

Va sottolineato che, a causa delle limitazioni temporali, non è stato possibile esplorare completamente le potenzialità di miglioramento dell'architettura di rete, degli iperparametri o la generazione di ulteriori dati. Questi aspetti rappresentano chiaramente aree di sviluppo futuro, dove ulteriori indagini potrebbero portare a un perfezionamento del modello e, di conseguenza, a una maggiore accuratezza nelle previsioni.

Un ulteriore aspetto suscettibile di miglioramenti e che richiede ulteriori approfondimenti è l'insieme di dati di addestramento. L'espansione del dataset attraverso simulazioni su diverse porzioni di mappa o su scenari con diverse conformazioni edilizie potrebbe arricchire la capacità predittiva del modello. Inoltre, l'aumento della risoluzione delle immagini potrebbe offrire al modello maggiore flessibilità durante la fase di addestramento.

Come prossimo passo si pensa di applicare le stesse tecniche per problemi di stima di parametri propagativi più complessi come il path loss o il delay spread, trasformando il problema da classificazione binaria in regressione lineare. Visto il comportamento del modello valutato sulla probabilità della linea di vista, vedi Figura 5.2, si ipotizza che per questa nuova tipologia di problemi il modello potrebbe fornire risultati qualitativamente migliori.

Bibliografia

- [1] Wikipedia. «Convolutional neural network.» (2023), indirizzo: https://en.wikipedia.org/wiki/Convolutional_neural_network.
- [2] O. Ronneberger, P. Fischer e T. Brox, «U-Net: Convolutional Networks for Biomedical Image Segmentation,» *CoRR*, vol. abs/1505.04597, 2015. arXiv: 1505.04597. indirizzo: <http://arxiv.org/abs/1505.04597>.
- [3] V. Degli-Esposti, F. Fuschini, E. M. Vitucci e G. Falciasecca, «Speed-Up Techniques for Ray Tracing Field Prediction Models,» *IEEE Transactions on Antennas and Propagation*, vol. 57, n. 5, pp. 1469–1480, 2009. DOI: 10.1109/TAP.2009.2016696.

Elenco delle figure

2.1	Mappa delle principali suddivisioni degli algoritmi di ML	5
2.2	Schema di una rete neurale convoluzionale [1]	6
2.3	Esempio di una U-Net con featuremap multicanale [2]	8
3.1	Antenna Tx che genera un'onda E.M.	11
3.2	Fenomeno della riflessione e trasmissione di un'onda alla superficie di incidenza	13
3.3	Cono di Keller	14
3.4	Scattering	14
3.5	Esempio di edifici “attivi” secondo il parametro A dove in verde viene rappresentata la mappa completa e in rosso quella semplificata [3]	22
4.1	Diagramma a blocchi del processo di generazione di mappe	24
4.2	Esempio di coppia di mappe input-output del dataset	26
5.1	Andamento della loss function al passare dei cicli di addestramento	31
5.2	Tre esempi di maschere ottenute con il modello addestrato	32
5.3	Istogramma valori F1	33

Ringraziamenti

Ringrazio in primo luogo il Professor Enrico Maria Vitucci per la disponibilità che ha mostrato nei miei confronti in questi giorni di lavoro e per tutto il supporto da lui fornitomi.

Un ringraziamento sentito va anche al Dott. Nicola Di Cicco per avermi guidato nella mia iniziale avventura nel campo del machine learning e per avermi trasmesso tanto in questi pochi mesi.

Ringrazio inoltre anche il Professor Vittorio Degli Esposti per aver seguito il progetto e fornito il materiale necessario per la comprensione dei fenomeni che stanno alla base della propagazione radio.

Ringrazio poi mia mamma e la mia famiglia in generale per essermi stati accanto in tutti questi anni, che hanno sempre accettato tutte le mie scelte accompagnandomi in questo percorso.

Un grandissimo grazie va la mia ragazza per avermi supportato, e sopportato, anche nei momenti più difficili e bui rendendoli più affrontabili e sereni, rallegrandomi tutte le giornate e dando un significato ad esse.

Una menzione speciale va anche ai miei amici Ion, Matteo e Mattia con cui ho condiviso dei momenti magici in questi ultimi anni e con cui spero di dividerne tanti altri.

Ringrazio poi tutti i miei compagni di corso che mi hanno tenuto compagnia durante le lezioni e regalato dei momenti indimenticabili. Un grazie va in privato a Federico Cobianchi con cui ho passato tutte le giornate, sopportandoci a vicenda e confortandoci quando necessario.

Due righe le vorrei dedicare anche ai professori delle superiori, in particolare al Prof. Daniele Casadei Lelli che ha fatto scattare in me la scintilla che mi ha portato poi fin qui oggi. Grazie per avermi mostrato quanto è affascinante questa materia.