INAF
ISTITUTO NAZIONALE
DI ASTROFISICA

Classificazione delle Modulazioni Radio tramite Deep Learning per applicazioni radioastronomiche

Autori

ALESSANDRO CABRAS[1], SALVATORE GARAU[2], TONINO PISANU[1], LUCA DIDACI[2], FRANCESCO GAUDIOMONTE[1]

[1]INAF - Osservatorio Astronomico di Cagliari
[2]Università di Cagliari

Revisori

PAOLO MAXIA
PIERLUIGI ORTU

Osservatorio Astronomico di Cagliari
2023

Indice

Abstract

Nel presente report viene illustrato un progetto di ricerca per il rilevamento e la classificazione delle Radio Frequency Interference (RFI) utilizzando algoritmi di machine learning con l'obiettivo di sviluppare un sistema automatizzato in grado di identificare e analizzare segnali indesiderati in modo rapido ed efficiente. La classificazione accurata delle RFI è fondamentale per mitigare le interferenze durante le osservazioni scientifiche con il Sardinia Radio Telescope (SRT).

Nel corso del progetto, sono stati raccolti dati relativi alle interferenze RFI in parte da dataset online e in parte utilizzando strumenti di acquisizione. Sono state esplorate diverse architetture di reti neurali e ne sono state valutate le performance utilizzando misure come l'accuratezza, la perdita (loss) e la matrice di confusione.

Sono state utilizzate anche apparecchiature hardware ad alta fedeltà per confrontare e valutare le prestazioni dei modelli di intelligenza artificiale sviluppati. Questo confronto è stato fondamentale per garantire l'accuratezza dei modelli e consentire una valutazione comparativa con gli strumenti hardware considerati come punto di riferimento.

Elenco degli Acronimi

RFI Radio Frequency Interference

AI Artificial Intelligence

SRT Sardinia Radio Telescope

OAC Osservatorio Astronomico di Cagliari

IQ In-phase and Quadrature

CNN Convolutional Neural Networks

PSK Phase-Shift Keying

BPSK Binary Phase-Shift Keying

QPSK Quadrature Phase-Shift Keying

8PSK 8 Phase-Shift Keying

QAM Quadrature Amplitude Modulation

OFDM Orthogonal Frequency-Division Multiplexing

SNR Signal to Noise Ratio

LFM Linear Frequency Modulation

FMCW Frequency Modulated Countinuous Wave

FMCW Frequency Modulated Countinuous Wave

AM Amplitude Modulation

FM Frequency Modulation

PM Phase Modulation

FSK Frequency-Shift Keying

GSM Global System for Mobile communications

GMSK Gaussian minimum-shift keying

VHF Very High Frequency

DVB-T Digital Video Broadcasting - Terrestrial

RTTY Radio Tele Type

USB Upper Side Band

LSB Lower Side Band

DVB-S2 Next Generation Digital Video Broadcasting Over Satellite

GPU Graphics Processing Unit

CCIR Comitte Consultatif Internationale des Radio

Elenco delle figure

1 Introduzione

1.1 Interferenze in radioastronomia

La ricerca e l'individuazione delle interferenze radio sono diventate di grande importanza nel campo della radioastronomia moderna. Le RFI sono segnali indesiderati che possono disturbare le osservazioni scientifiche effettuate con i radiotelescopi. La classificazione accurata delle RFI è fondamentale per identificare le sorgenti radio che interferiscono con il SRT durante le osservazioni scientifiche.

Presso l'Osservatorio Astronomico di Cagliari (OAC), si svolge un significativo lavoro per il rilevamento e la mitigazione dei segnali RFI, compresa l'individuazione delle sorgenti che emettono tali disturbi. Tra i vari segnali e le diverse tipologie di emissioni rilevate, ad esempio, vi sono le trasmissioni dei radar in banda L (1200 - 1400 MHz) gestiti dall'aeronautica militare. Nel sud-est della Sardegna, sono presenti 9 installazioni radar, ognuna delle quali trasmette su più portanti (più frequenze, fino a 4 per ogni radar). Ogni radar ha una propria modulazione digitale che consente di riconoscere il proprio segnale una volta riflesso dal bersaglio. Uno dei problemi principali è quello di riuscire a classificare e assegnare ogni singola portante ricevuta dal SRT al fine di individuarne la sorgente ed eventualmente mettere in atto delle strategie che permettano di mitigarla. Attualmente, questo compito viene svolto manualmente, con tutte le difficoltà e imprecisioni che questo approccio comporta. Inoltre, quando le emissioni sconfinano dalla banda assegnata ai radar (emissioni spurie), sarebbe utile identificare a quale installazione radar appartengono al fine di prendere le adeguate misure correttive. L'approccio si applica anche ad altre tipologie di segnali digitali, come HiperLan, TETRA, Wimax, ecc., che possono interferire col corretto funzionamento del SRT.

Al fine di superare le problematiche legate alla rilevazione delle emissioni RFI, si è ritenuto necessario sviluppare uno strumento in grado di identificare e analizzare tali segnali indesiderati in modo rapido e automatizzato. Gli algoritmi di machine learning rappresentano una potente risorsa per la classificazione delle modulazioni digitali [1] [2] [3], consentendo di identificare in modo preciso e automatico il tipo di modulazione presente in un segnale e facilitandone così l'analisi e la comprensione. Nel progetto, le reti neurali sono state impiegate per la classificazione delle RFI, utilizzando i dati IQ che sono caratteristici di ogni modulazione. Attraverso l'addestramento di reti neurali con dati sintetici e una successiva sperimentazione utilizzando dati reali, è stato possibile valutare l'accuratezza e le prestazioni del modello proposto nel riconoscimento e nella classificazione delle diverse tipologie di interferenza.

Durante il processo di addestramento delle reti neurali, è stato fondamentale utilizzare strumentazione di riferimento calibrata per confrontare e valutare i risultati ottenuti. La strumentazione utilizzata, comprende un generatore di segnali Rohde & Schwarz ed un R&S®FSW Signal and Spectrum Analyzer. Il primo è stato utilizzato per generare segnali di prova con modulazioni specifiche, al fine di verificare le prestazioni delle reti neurali nell'identificazione e nella classificazione di tali segnali contribuendo a garantire la riproducibilità e la consistenza dei segnali utilizzati nel processo di addestramento e valutazione; il secondo, invece, è stato impiegato per l'analisi e la valutazione dei segnali rilevati durante il processo di addestramento e durante i test con dati reali.

Le fasi principali del lavoro svolto posso essere così riassunte:

1. Raccolta dei dati relativi alle interferenze RFI: sono stati acquisiti dati che includono segnali sintetici con diverse modulazioni provenienti da diversi dataset.

2. Pre-elaborazione dei dati: i dati raccolti sono stati sottoposti a una serie di operazioni di pre-elaborazione, tra cui la normalizzazione dei segnali e loro suddivisione in set di addestramento e di test.

3. Rete neurale: sono state esplorate diverse architetture di reti neurali, tra cui le Convolutional Neural Networks (CNN) [4] e la rete neurale Inception.

4. Valutazione delle prestazioni: Sono stati utilizzati diversi metodi per valutare le prestazioni delle reti neurali, tra cui l'accuratezza, la perdita (loss) e la matrice di confusione.

5. Uso degli apparati hardware sopra descritti come metodo di confronto.

1.2 Raccolta dei dati

La demodulazione dei segnali in rappresentazione IQ, è una tecnica utilizzata per estrarre i dati da un segnale modulato in frequenza o fase basandosi sulla rappresentazione del segnale modulato come una combinazione di due componenti complesse: una in fase (I) ed una in quadratura (Q).

Per demodulare il segnale, vengono impiegati due mixer in quadratura. Il primo moltiplica il segnale in ingresso (modulato) per una portante in fase, il secondo lo moltiplica invece per una portante in quadratura (sfasata di 90 gradi rispetto a quella in fase).

Dopo il mixing, i segnali risultanti vengono opportunamente filtrati con filtri passa-basso allo scopo di rimuovere le eventuali componenti ad alta frequenza generate nel processo di mixaggio.

La rappresentazione IQ presenta diversi vantaggi. Ad esempio, consente una maggiore efficienza spettrale, in quanto un eventuale segnale modulato viene rappresentato in uno spazio bidimensionale anziché unidimensionale. Inoltre, l'utilizzo di due componenti in quadratura permette una migliore resistenza ai disturbi e una maggiore tolleranza agli errori di fase. Poiché la rappresentazione IQ è bidimensionale, è possibile applicare le stesse tecniche di apprendimento automatico utilizzate per la classificazione delle immagini.

Per il training delle reti neurali possono essere utilizzati diversi dataset, i quali contengono una serie di esempi di forme d'onda IQ corrispondenti a diverse modulazioni, al fine consentire alla rete di classificarle correttamente. Questo processo di classificazione basato sulle reti neurali consente di identificare in modo automatico e preciso la modulazione presente nelle forme d'onda, facilitando così la comprensione e l'analisi dei segnali digitali. I principali tipi di modulazione raggruppati per utilizzo sono:

- Phase-Shift Keying (PSK): Modulazione di fase a 2/4/8/16 stati utilizzata in Wi-Fi, Bluetooth, ecc. Ad es. Binary Phase-Shift Keying (BPSK), Quadrature Phase-Shift Keying (QPSK), 8 Phase-Shift Keying (8PSK).
- Quadrature Amplitude Modulation (QAM): Modulazione d'ampiezza in quadratura a 16/64/256 stati utilizzata in Wi-Fi, Digital Video Broadcasting - Terrestrial (DVB-T), ecc. Usa due portanti SF in quadratura.
- Frequency-Shift Keying (FSK): Modulazione di frequenza a 2/4 stati. Ad es. Gaussian minimum-shift keying (GMSK) usata in Global System for Mobile communications (GSM).

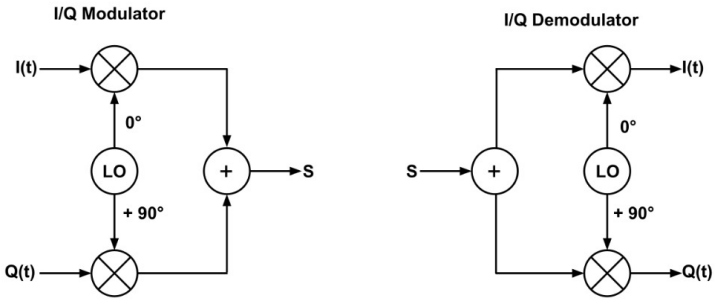


Figura 1.1: Modulazione e Demodulazione in quadratura IQ

- Linear Frequency Modulation (LFM): Variazione lineare della frequenza utilizzata in radar Frequency Modulated Countinuous Wave (FMCW).
- Phase-coded waveforms: Vari codici di modulazione di fase utilizzati in radar come Frank, P1/P2/P3/P4, Zadoff-Chu, Barker, ecc.
- Stepped-Frequency radar: Trasmette impulsi a frequenze discrete ("gradini") per misurare il ritardo di ritorno.
- Amplitude Modulation (AM): Modulazione d'ampiezza utilizzata nelle radio AM e trasmissioni Very High Frequency (VHF).
- Frequency Modulation (FM): Modulazione di frequenza utilizzata nelle radio FM e trasmissioni TV analogiche.
- OFDM: Tecnica di multiplexing con sottocanali ortogonali utilizzata in Wi-Fi, telefonia mobile, DVB-T, ecc.

Le modulazioni digitali come PSK, QAM e FSK vengono utilizzate principalmente per le comunicazioni Wi-Fi, Bluetooth e cellulari. Le modulazioni radar come LFM, stepped-frequency e phase-coded waveform consentono la misurazione della distanza di oggetti in movimento. AM e FM sono ancora utilizzate per le trasmissioni radio e TV analogiche. OFDM è una tecnica avanzata utilizzata sia per le comunicazioni digitali che per i radar.

2
Dataset

2.1 Dataset utilizzati nel progetto

Di seguito vengono elencati i diversi dataset utilizzati:

2.1.1 Dataset Panoradio HF

Il dataset Panoradio [1] contiene 172800 segnali radio digitali, ciascuno con 2048 campioni IQ (complessi), acquisiti con una frequenza di campionamento di 6000 Hz. L'energia di ciascun segnale è normalizzata a 1 e c'è uno spostamento casuale di frequenza di +/- 250 Hz per simulare la deriva di frequenza che si verifica nelle trasmissioni radio reali. I segnali hanno un SNR di 25, 20, 15, 10, 5, 0, -5 e -10 dB per riflettere vari livelli di rumore e interferenza. I segnali sono stati trasmessi attraverso un canale Comitte Consultatif Internationale des Radio (CCIR) 520 per simulare l'effetto di percorsi multipli e fading. Il dataset contiene 17 tipi di modulazioni, sia digitali che analogiche, utilizzate nelle comunicazioni radio:

- Morse: codifica digitale on-off (OOK)
- PSK31 e PSK63: modulazione di fase a 31 e 63 baud
- QPSK31: modulazione di fase a 4 stati a 31 baud
- Radio Tele Type (RTTY)45_170 e RTTY50_170: telescrivente a 45 e 50 baud, 170 Hz di shift Mark-Space modulazione FSK

- RTTY100_850: telescrivente a 100 baud, 850 Hz di shift Mark-Space modulazione FSK
- OLIVIA8_250, OLIVIA16_500 e OLIVIA32_1000: OLIVIA a 8, 16 e 32 toni, 250, 500 e 1000 Hz di larghezza di banda
- DOMINOEX11: a 11 frequenze
- MT63_1000: MT63 a 1000 Hz di larghezza di banda
- NAVTEX: sistema di trasmissione di messaggi di sicurezza marittima
- Upper Side Band (USB) e Lower Side Band (LSB): banda laterale unica superiore e inferiore con modulazione d'ampiezza.
- AM: modulazione d'ampiezza
- FAX: trasmissione di segnali per FAXSIMILE via radio

2.1.2 OFDM dataset

Il dataset [5] contiene 6 tipi di segnali OFDM modulati in banda base con intestazione e payload modulati rispettivamente come BPSK+BPSK, BPSK+QPSK, BPSK+8PSK, QPSK+BPSK, QPSK+QPSK, QPSK+8PSK. L'intervallo SNR di ciascun segnale è da -10 dB a +20 dB con incrementi di 2 dB. Per ogni tipo di segnale in uno specifico SNR, sono stati generati 4096 campioni di dati, ognuno dei quali ha 1024 attributi. In totale ci sono $6 \times 16 \times 4096 = 393216$ pezzi. In questo set di dati, sono presenti sei tipi di segnali OFDM modulati di banda base. L'OFDM è una tecnica di modulazione utilizzata in molti sistemi di comunicazione, come Wi-Fi, 4G/5G e radio digitale terrestre, per trasmettere dati ad alta velocità su canali multipli. I segnali sono combinati utilizzando diverse modalità di modulazione per l'intestazione (header) e il payload. Di seguito, una descrizione dei segnali presenti nel set di dati e le loro caratteristiche:

- BPSK+BPSK: Entrambe l'intestazione e il payload sono modulati utilizzando la modulazione BPSK (Binary Phase Shift Keying). La BPSK è una tecnica di modulazione di fase che utilizza due fasi distinte per rappresentare i bit 0 e 1. Viene utilizzata in sistemi di comunicazione semplici e robusti dove la larghezza di banda e l'efficienza spettrale non sono una priorità.
- BPSK+QPSK: L'intestazione è modulata con BPSK, mentre il payload è modulato con QPSK (Quadrature Phase Shift Keying). La QPSK utilizza quattro fasi diverse per rappresentare i dati (due bit per simbolo), offrendo un'efficienza spettrale doppia rispetto alla

BPSK. La QPSK è comunemente utilizzata in sistemi di comunicazione digitali come Wi-Fi e satellitari.

- BPSK+8PSK: L'intestazione è modulata con BPSK, e il payload è modulato con 8PSK (8 Phase Shift Keying). L'8PSK utilizza otto fasi diverse per rappresentare i dati (tre bit per simbolo), offrendo un'efficienza spettrale maggiore rispetto alla BPSK e QPSK. L'8PSK viene utilizzato principalmente in sistemi di comunicazione ad alta velocità e ad alta capacità, come la trasmissione satellitare Next Generation Digital Video Broadcasting Over Satellite (DVB-S2).

- QPSK+BPSK: L'intestazione è modulata con QPSK e il payload è modulato con BPSK. Questa combinazione può essere utilizzata per aumentare l'affidabilità dell'intestazione in presenza di rumore e interferenze.

- QPSK+QPSK: Entrambe l'intestazione e il payload sono modulati utilizzando QPSK. Questa combinazione è comune in molti sistemi di comunicazione digitale che richiedono un equilibrio tra efficienza spettrale e robustezza.

- QPSK+8PSK: L'intestazione è modulata con QPSK e il payload è modulato con 8PSK. Questa combinazione può essere utilizzata per massimizzare l'efficienza spettrale del payload mentre si mantiene una maggiore affidabilità nella trasmissione dell'intestazione. L'intervallo di rapporto segnale-rumore (SNR) varia da -10 dB a +20 dB con incrementi di 2 dB per ciascun tipo di segnale. Un SNR più elevato indica una migliore qualità del segnale rispetto al rumore di fondo, il che può influenzare la capacità di ricevere e decodificare correttamente i dati trasmessi.

2.1.3 DeepRadar2022 dataset

Il dataset DeepRadar2022 [6] contiene 782000 segnali generati con 23 diverse modulazioni ampiamente utilizzate nei radar. Questo dataset è stato creato attraverso simulazioni MatLab con l'obiettivo di supportare la classificazione dei segnali radar e fornire un ampio set di dati per ulteriori ricerche nel campo. È stato sviluppato dal gruppo Radar e Microonde presso l'Università Politecnica di Madrid.

Il dataset comprende una serie di segnali, ognuno dei quali è rappresentato da un vettore 1024x2 che contiene i campioni temporali in fase e quadratura (I + jQ) del segnale modulato, incluso un contributo di rumore. Per ogni modulazione considerata, sono stati generati segnali con valori di

SNR compresi tra -12 dB e 20 dB, con incrementi di 2 dB. Questa variazione degli SNR consente di coprire una vasta gamma di condizioni realistiche che si possono incontrare nelle applicazioni radar. Le modulazioni considerate come classi sono:

- FSK a 2/4/8 stati
- PSK a 2/4/8 stati
- Modulazioni di fase con codici Barker, Huffman, Frank, Px, Zadoff-Chu, T1/2/3/4
- Frequency Modulation con codice Costas
- LFM
- Rumore bianco gaussiano complesso

Per ogni modulazione e per ogni SNR sono stati generati 1200, 400 e 400 segnali rispettivamente per training, convalida e test. Pertanto, il set di dati finale è distribuito come segue:

- Set di training -> 469.200 segnali
- Set di convalida -> 156.400 segnali
- Set di test -> 156.400 segnali

2.2 Ambiente di sviluppo

Si è deciso di utilizzare un ambiente di sviluppo online per sopperire alla mancanza di hardware all'avanguardia dedicato al training di algoritmi di apprendimento. Google Colaboratory, noto come Colab, è una piattaforma basata su cloud che permette di scrivere ed eseguire codice Python direttamente nel browser, senza richiedere la configurazione di un ambiente di sviluppo locale.

L'utilizzo di Colab presenta diversi vantaggi come ad esempio l'accesso alle risorse di calcolo gratuite, inclusa l'accelerazione Graphics Processing Unit (GPU) consentendo di eseguire addestramenti impegnativi senza dover investire in hardware dedicato. Inoltre, Colab fornisce una vasta selezione di librerie preinstallate, semplificando il processo di configurazione dell'ambiente di sviluppo. Questo permette di risparmiare tempo e di concentrarsi maggiormente sullo sviluppo del codice e sull'esplorazione dei modelli.

Tuttavia, l'utilizzo di Colab presenta anche alcuni svantaggi. Innanzitutto, essendo un ambiente basato su cloud, è necessaria una connessione a Internet per accedere ai servizi offerti. Inoltre, l'uso delle risorse di calcolo gratuite può essere soggetto a limitazioni, ad esempio per quanto riguarda il tempo massimo di esecuzione consentito per una singola sessione.

Nonostante questi potenziali svantaggi, l'utilizzo di questo tool si è rivelato una soluzione efficace e conveniente per l'addestramento dei modelli di machine learning sui diversi dataset, consentendo di ottenere risultati significativi anche in assenza di una macchina locale dedicata.

2.3 Pre-elaborazione dei dati

La pre-elaborazione dei dati è un passaggio fondamentale nella preparazione di un dataset per il machine learning. In questa fase vengono caricati i dati e le etichette di classe e opportunamente mescolati per garantire una distribuzione casuale. Questi passaggi sono volti a preparare il dataset per l'addestramento del modello di Artificial Intelligence (AI), garantendo che i dati siano coerenti, casualmente distribuiti e adeguati alla struttura richiesta dall'algoritmo utilizzato.

Nel dettaglio, in questa fase sono state svolte le seguenti operazioni per il dataset deepRadar:

1. Caricamento dei dati: I dati vengono caricati dai file X_train.mat, X_val.mat e X_test.mat utilizzando la libreria h5py. Questi contengono i campioni di addestramento, validazione e test relativi al dataset DeepRadar. I dati vengono caricati come array NumPy e successivamente viene effettuata una trasposizione.

2. Caricamento delle etichette: Le etichette associate ai campioni di addestramento, validazione e test vengono caricate dai file Y_train.mat, Y_val.mat e Y_test.mat utilizzando la funzione io.loadmat del modulo io. Le etichette vengono salvate come array NumPy.

3. Caricamento delle etichette di classe: Le etichette di classe associate ai campioni di addestramento, validazione e test vengono caricate dai file lbl_train.mat, lbl_val.mat e lbl_test.mat utilizzando la funzione io.loadmat. Le etichette di classe vengono salvate come array NumPy.

4. Mescolamento dei dati: Viene utilizzata la funzione sklearn.utils.shuffle del modulo sklearn per mescolare casualmente i dati di addestramento, validazione e test insieme alle rispettive etichette e etichette di classe. Questo passaggio è importante per evitare che il modello apprenda un ordine specifico durante l'addestramento.

Figura 2.1: Analizzatore di segnali FSW67 Rohde & Schwarz

5. Trasposizione dei dati: Utilizzando la funzione np.transpose(), viene effettuata una trasposizione degli array dei dati X_train, X_val e X_test. Questa operazione scambia l'ordine del secondo e terzo asse, effettuando così la trasposizione dei dati (fondamentale per adattare i dati alla struttura richiesta dalla rete neurale).

2.4 Apparati Hardware OAC

L'FSW67 della Rohde & Schwarz è uno strumento per l'analisi dei segnali e dello spettro radio, in grado di campionare e analizzare segnali radio a larga banda, sia il dominio del tempo che della frequenza.

Per visualizzare la costellazione IQ, l'FSW67 campiona il segnale radio misurandone ampiezza e fase istantanea. Il ricevitore non è in grado di classificare il segnale in quanto è necessario a priori conoscere la modulazione per mappare il segnale nel display. È tuttavia uno strumento importante per l'analisi del segnale e può essere utilizzato come gold standard per verificare la modulazione di un segnale sconosciuto una volta trovata la modulazione.

Per questa ragione, è stato utilizzato in combinazione con il SMW200A, un generatore di segnali vettoriali ad alte prestazioni. Per creare segnali modulati e verificarne l'analisi della costellazione, è necessario collegare correttamente questi due strumenti attraverso l'uscita RF del generatore all'ingresso RF dell'FSW67 utilizzando un cavo coassiale. Il generatore SMW200A è stato utilizzato per generare segnali modulati in base ai parametri desiderati, come frequenza portante, tipo

di modulazione e frequenza dei simboli. Successivamente, l'FSW67 analizza il segnale ricevuto, permettendo di visualizzare la costellazione e verificare la qualità del segnale modulato.

Per registrare i dati IQ, si configura l'FSW67 in modo da catturare i campioni del segnale ricevuto. Questi campioni possono essere salvati su un dispositivo di archiviazione esterno. Una volta raccolti i dati, è stato utilizzato il modello addestrato per classificare la modulazione del segnale, confrontando i risultati ottenuti con quelli forniti dall'analizzatore FSW67, utilizzato come riferimento. È stato inoltre possibile aggiungere rumore e quindi variare l'SNR per verificare le prestazioni del modello addestrato. Questo processo ha permesso di verificare l'accuratezza e l'affidabilità della rete neurale nella classificazione dei segnali modulati.

3
Machine Learning

3.1 Rete neurale

Per il progetto è stata utilizzata una rete neurale basata sull'architettura GoogleNet anche conosciuta come Inception v1 [7], un'architettura sviluppata dal team Google per migliorare le prestazioni dei modelli di apprendimento automatico su immagini.

L'architettura GoogleNet, presenta un design profondo e complesso che consente di apprendere le rappresentazioni ad alto livello dei dati. Questa rete neurale si distingue per l'utilizzo di moduli Inception, che sono composti da più livelli convoluzionali con kernel di dimensioni e filtri diversi. Questi livelli lavorano parallelamente e le loro uscite vengono concatenate insieme per formare l'output finale del modulo Inception. Questa struttura multi-scala consente alla rete di catturare caratteristiche a diverse scale spaziali, permettendo di discriminare dettagli sia locali che globali nei dati di input.

Inoltre, l'architettura GoogleNet utilizza diverse tecniche per migliorare le prestazioni del modello. Una di queste è la regolarizzazione Dropout, che consiste nel disattivare in modo casuale alcune unità neurali durante la fase di addestramento. Questa tecnica aiuta a prevenire l'overfitting, ossia un adattamento eccessivo ai dati di addestramento che può compromettere le prestazioni su nuovi dati forniti in inferenza. Un'altra tecnica utilizzata è l'uso di connessioni residue, in cui l'output di un livello viene sommato all'input di un livello successivo. Questo facilita il flusso di informazioni attraverso i vari livelli della rete, permettendo una migliore propagazione del gradiente durante la fase di retropropagazione dell'errore.

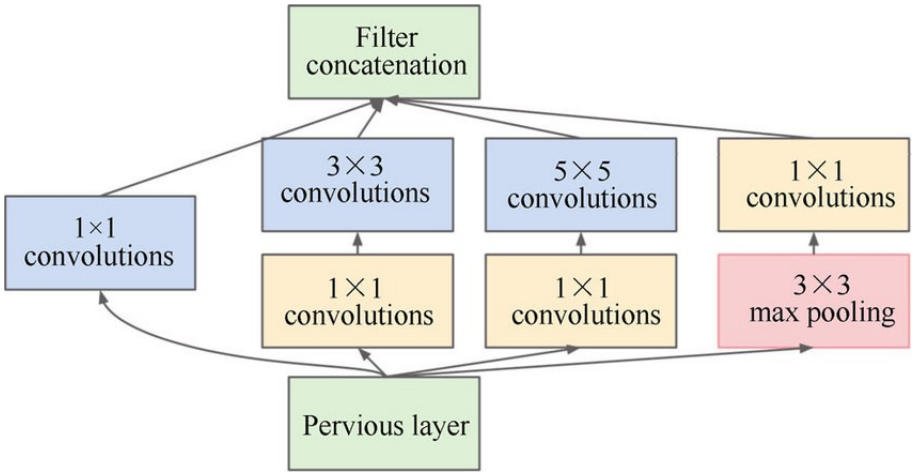


Figura 3.1: Struttura della rete neurale Inception

La struttura Inception include anche un meccanismo di ”bottleneck” (collo di bottiglia) che riduce la quantità di parametri della rete, evitando il sovraccarico di dati. Nel contesto di questo progetto, è stato adottato il transfer learning, adattando una rete inception addestrata per lavorare su immagini, ad elaborare input bidimensionali di forma (2, 1024), che rappresentano i dati IQ del segnali. Il transfer learning infatti, consente di sfruttare le conoscenze apprese dal modello preaddestrato per svolgere un task diverso e adattarle al problema corrente. Questo approccio riduce la necessità di addestrare il modello da zero riducendo il carico computazionale.

L'architettura GoogleNet è stata adattata per svolgere un compito di classificazione multi-classe, in cui l'obiettivo è assegnare una delle 23 classi ai dati radar in input. L'output del modello per una determinata istanza di input in un classificatore di modulazione basato su machine learning, è un vettore di probabilità in cui ciascun valore rappresenta la probabilità che il segnale di input possa appartenere a una delle modulazioni che il modello è stato addestrato a riconoscere.

Ad esempio, supponiamo di avere un classificatore di modulazione che può riconoscere quattro tipi di modulazione: AM, FM, Phase Modulation (PM) e QAM. Dato un segnale di input, l'output del modello potrebbe essere un vettore di probabilità come [0.1, 0.7, 0.2, 0.0], dove ciascun valore rappresenta la probabilità che il segnale di input appartenga rispettivamente alle modulazioni AM, FM, PM e QAM.

In questo caso, il valore più alto nel vettore (0.7) corrisponde alla modulazione FM, indicando che il modello ritiene con una probabilità del 70% che il segnale di input sia modulato in frequenza. I valori più bassi (0.1, 0.2 e 0.0) corrispondono invece alle probabilità delle altre modulazioni.

L'uso di una rete neurale come GoogleNet è particolarmente utile, in quanto questa architettura

complessa è in grado di apprendere rappresentazioni robuste e discriminative dei dati, consentendo prestazioni elevate nella classificazione. L'addestramento del modello avviene attraverso la presentazione di numerosi esempi di dati radar etichettati correttamente, permettendo alla rete di apprendere le relazioni tra le caratteristiche dei segnali e le classi di appartenenza.

3.2 Valutazione delle prestazioni

E' stata dedicata particolare attenzione al miglioramento delle prestazioni della rete neurale utilizzata per la classificazione dei segnali radio. Per ottenere risultati più affidabili e una migliore ottimizzazione dei parametri, è stato implementato un modello di apprendimento profondo utilizzando il metodo di validazione incrociata K-Fold Cross Validation.

Il K-Fold Cross Validation presenta il vantaggio di utilizzare l'intero dataset sia per l'addestramento che per la validazione del modello. Questo significa che ogni campione del dataset viene utilizzato sia come dato di addestramento che come dato di validazione in un determinato momento. Tale approccio permette di ottenere stime più accurate delle prestazioni del modello rispetto alla semplice divisione train/test.

Per la compilazione del modello, è stato utilizzata la funzione di perdita 'categorical_crossentropy', che è adatta per problemi di classificazione multi-classe. Come ottimizzatore invece è stato scelto 'adam', un algoritmo ampiamente utilizzato nell'addestramento di reti neurali.

Durante l'addestramento del modello sono state applicate diverse tecniche per migliorare le prestazioni e garantire una maggiore generalizzazione. Una di queste è stata l'utilizzo della tecnica Early Stopping, che monitora la funzione di perdita durante il training interrompendolo quando questo non migliora ulteriormente per un numero di epoche prestabilito. L'Early Stopping evita l'overfitting interrompendo l'allenamento quando la funzione di perdita smette di migliorare, evitando che la rete continui ad adattarsi ai dati di addestramento fino a diventare troppo specifica. Inoltre, è stata utilizzata la funzione callback ModelCheckpoint, la quale consente di salvare automaticamente il modello ad ogni epoca o solo quando si verifica un miglioramento in una determinata metrica, come l'accuratezza di validazione.

```
1 num_folds = 10
2 kf = KFold(n_splits=num_folds)
3 accuracy = []
4 loss = []
```

```
X = X_train
Y = Y_train models = []
for fold_index, (train, test) in enumerate(kf.split(X, Y)):
    x_train = X[train]
    y_train = Y[train]
    x_test = X[test]
    y_test = Y[test]
    in_shp = (2, 1024)
    input_img = Input(shape=in_shp)
    out = googleNet(input_img,data_format='channels_last', num_classes
        =23, num_layers = [1,1,1,1])
    model = Model(inputs=input_img, outputs=out)
    model.compile(loss='categorical_crossentropy', optimizer='adam',
        metrics=['accuracy'])
    model_path = f'/content/drive/Shareddrives/Tirocinio Salvatore/
        modelIncDRadioMaggio{fold_index}/model_{fold_index}.h5'
    # addestro il modello
    earlyStopCb = tf.keras.callbacks.EarlyStopping(monitor='loss',
        patience=3)
    modelSaveCb = tf.keras.callbacks.ModelCheckpoint(model_path,
        save_best_only=True, monitor='val_accuracy') history = model.
        fit(x_train, y_train, validation_split=0.1, batch_size=1024,
        epochs=30, callbacks=[earlyStopCb, modelSaveCb]) _,
        test_accuracy = model.evaluate(x_test, y_test)
    accuracy.append(test_accuracy)
    models.append(model)
```

Listing 3.1: Codice Python per l'addestramento del modello

Il modello è stato addestrato utilizzando un'epoca massima di 30 e una dimensione del batch di 1024. Durante l'addestramento, è stato riservato il 10% dei dati di addestramento come insieme di validazione per valutare le prestazioni del modello durante ogni epoca di addestramento. L'accuratezza ottenuta alla fine di ogni ciclo è stata registrata in un array, in modo da poter calcolare il valore minimo, massimo, medio e la deviazione standard dell'accuratezza per l'intero processo di

validazione incrociata K-Fold.

Durante ogni iterazione su tutte le partizioni di K-Fold, il modello corrispondente alla partizione i-esima è stato salvato per un'ulteriore analisi delle prestazioni e per un possibile utilizzo futuro. Questo ha permesso di valutare le performance del modello su diverse partizioni del dataset e confrontarle per individuare eventuali variazioni significative.

In questo modo, è stata ottenuta una valutazione completa delle prestazioni del modello durante l'addestramento utilizzando il metodo K-Fold Cross Validation. Salvando i modelli corrispondenti a ciascuna partizione, si è potuta valutare la differenza di performance tra le partizioni e selezionare il modello che ha ottenuto le migliori prestazioni complessive. Questo approccio ha fornito una maggiore fiducia nella validità delle prestazioni del modello sulla totalità del dataset, nonché la possibilità di utilizzare i modelli salvati per ulteriori analisi e applicazioni future.

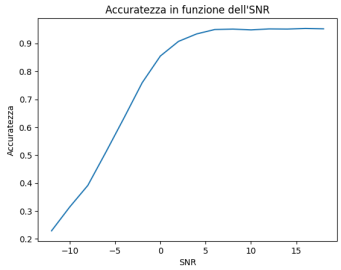
4
Risultati

In questo capitolo sono stati raccolti i risultati ottenuti durante l'esperimento utilizzando la validazione incrociata K-Fold con 10 fold. Da questo deriva l'addestramento di 10 modelli diversi e dei rispettivi valori di accuratezza per valutare l'omogeneità del dataset. Inoltre, sono stati valutati i risultati in base agli SNR, addestrando la rete sull'intero dataset, su un singolo SNR e su una parte del dataset con SNR maggiori di 0.

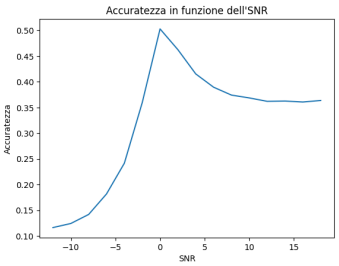
Come previsto, l'accuratezza aumenta al diminuire del rumore, ma è importante considerare che un dataset di segnali senza rumore sarebbe poco realistico per un'implementazione reale della rete neurale. Pertanto, è stata fatta una valutazione per verificare fino a che punto il rumore influisse sull'addestramento della rete, sia in termini di accuratezza che di classificazione.

La rete quindi è stata addestrata non solo sull'intero dataset, che comprendeva SNR da -12 a 20, ma anche su segnali specifici con un determinato SNR (0, 10, 20) e su segnali con SNR>0, che rappresentano le normali interferenze radio. Come risultato della sperimentazione, è stato osservato che utilizzando questi approcci più specifici di addestramento sulla presenza di rumore non ci sono stati miglioramenti significativi, sia in termini di accuratezza che di classificazione. Di seguito vengono riportate le accuratezze ottenute nei diversi casi:

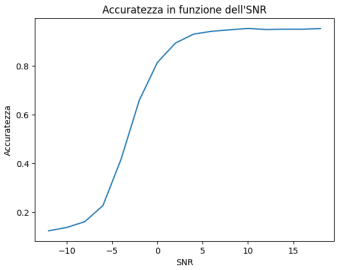
- Intero dataset (SNR da -12 a 20 con passo di 2): Media: 0.77 Minimo: 0.77 Massimo: 0.78 Deviazione standard: 0.002
- Solo SNR=0 Media: 0.53 Minimo: 0.47 Massimo: 0.58 Deviazione standard: 0.02
- Solo SNR=10 Media: 0.81 Minimo: 0.72 Massimo: 0.85 Deviazione standard: 0.036



(a) Intero dataset (SNR da -12 a 20 con passo di 2)



(b) SNR=0



(c) SNR da 0 a 20 con passo di 2)

Figura 4.1: Accuratezze ottenute utilizzando diversi valori di SNR

- Solo SNR=20 Media: 0.90 Minimo: 0.88 Massimo: 0.91 Deviazione standard: 0.009
- dataset con SNR maggiore o uguale a 0 (da 0 a 20 con passo di 2): Media: 0.93 Minimo: 0.93 Massimo: 0.94 Deviazione: 0.003

È importante notare che, anche se l'accuratezza più alta si ottiene con il dataset che ha un SNR maggiore o uguale a 0, durante l'addestramento del modello vengono eseguiti test solo sui campioni con lo stesso SNR. Pertanto, è stato valutato l'effetto di addestrare modelli sull'intero dataset. Nelle figure seguenti sono riportati i grafici dell'accuratezza in funzione dell'SNR, dai quali grafici emerge chiaramente che addestrare la rete con un singolo SNR non garantisce buoni risultati 4.1b. Inoltre escludere dai dati di addestramento quelli con SNR troppo bassi 4.1c non offre vantaggi significativi in termini di accuratezza.

Inoltre, è stata calcolata la matrice di confusione normalizzata al fine di valutare le prestazioni del modello nella classificazione delle diverse classi di segnali permettendo di identificare eventuali limitazioni del modello nella classificazione di classi specifiche. Nelle figure 4.2 e 4.3 sono riportate le matrici di confusione nel caso di addestramento sull'intero dataset e nel caso di addestramento con SNR>0.

In conclusione è stata effettuata la valutazione della rete su segnali radar reali registrati. Nel codice Python mostrato in Figura 4.4, viene illustrato il processo di caricamento di un file audio in formato mp3 contenente un segnale radar. Il segnale viene successivamente elaborato in formato IQ utilizzando una funzione apposita. Successivamente, il modello di deep learning pre-addestrato viene utilizzato per classificare il segnale in una delle categorie di segnale radar predefinite.

Il processo di classificazione viene eseguito iterativamente su sottoinsiemi di 1024 campioni del segnale, e per ciascun sottoinsieme il modello restituisce la classe prevista. L'elenco delle classi

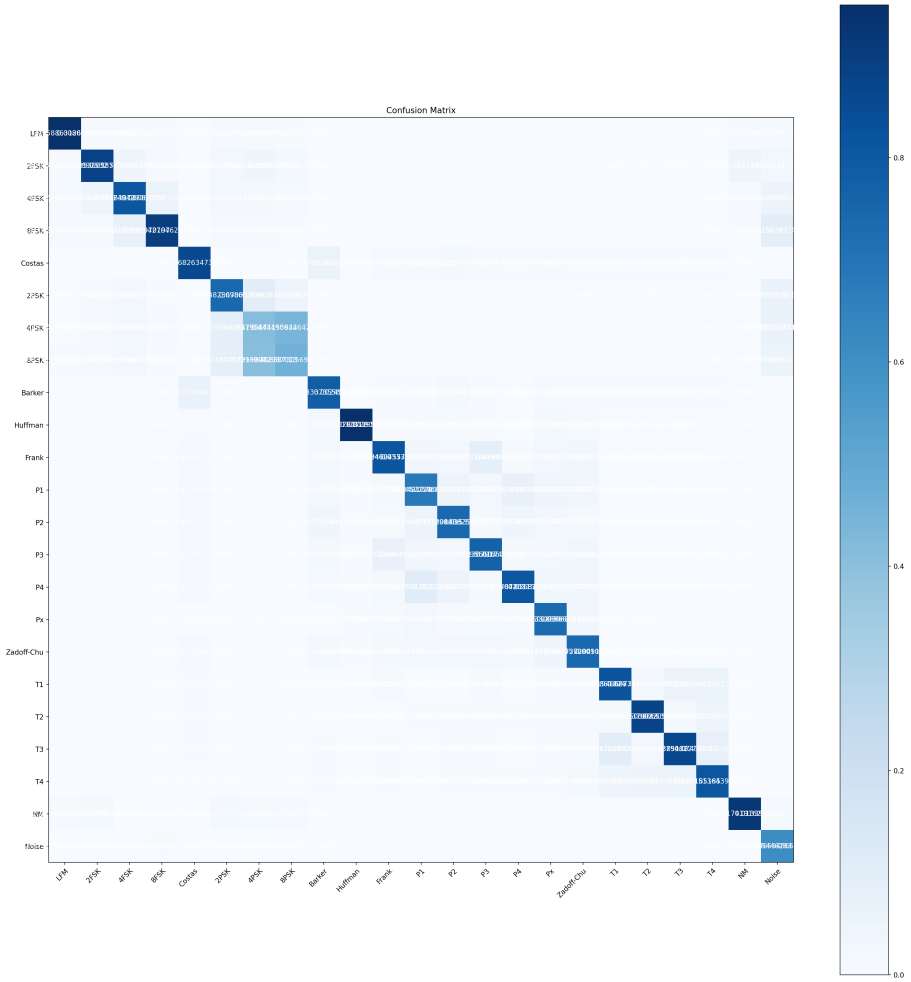


Figura 4.2: Confusion Matrix con addestramento intero dataset (SNR da -12 a 20 con passo di 2)

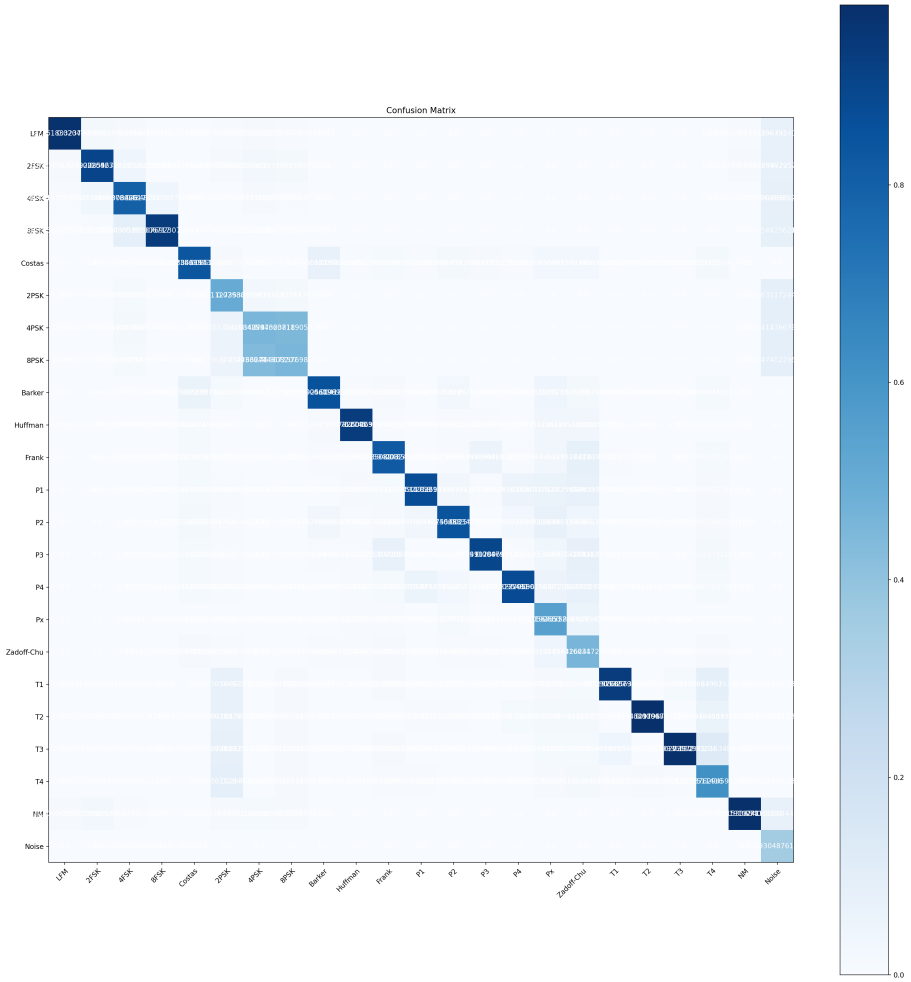


Figura 4.3: Confusion Matrix con addestramento di parte del dataset (SNR da 0 a 20 con passo di 2)

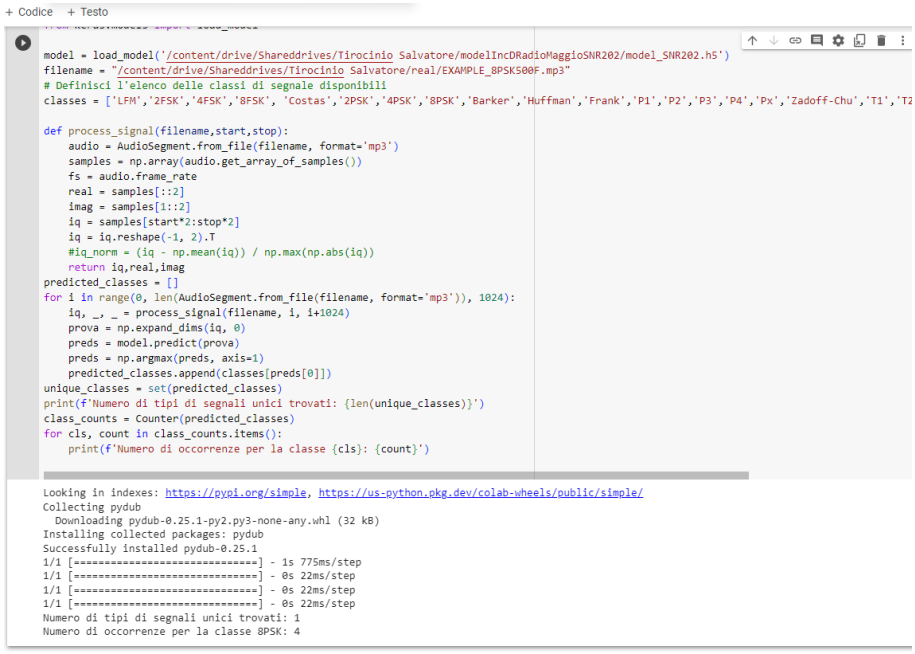


Figura 4.4: Classificazione di un segnale reale registrato

previste viene quindi utilizzato per contare il numero di occorrenze di ciascuna classe e stampare i risultati.

Nel caso specifico descritto, il modello ha correttamente previsto la classe 8PSK per tutti i sottoinsiemi di campioni del segnale.

Conclusioni e sviluppi futuri

Le conclusioni del presente progetto di ricerca evidenziano che l'utilizzo di algoritmi di machine learning, in particolare reti neurali, permette di sviluppare un sistema automatizzato efficace per il rilevamento e la classificazione delle interferenze RFI. Ciò contribuisce a mitigare le interferenze durante le osservazioni scientifiche con il Sardinia Radio Telescope, migliorando la qualità delle osservazioni astronomiche. L'implementazione di tale sistema riduce la dipendenza da approcci manuali, consentendo un rilevamento e un'analisi più rapidi ed efficienti delle interferenze RFI. Ciò si traduce in un'ottimizzazione delle prestazioni complessive del telescopio, promuovendo l'avanzamento delle ricerche scientifiche.

Bibliografia

[1] S. Scholl, *Classification of Radio Signals and HF Transmission Modes with Deep Learning*, 2019. arXiv: 1906.04459 [eess.SP].

[2] T. J. O'Shea, J. Corgan e T. C. Clancy, "Convolutional Radio Modulation Recognition Networks", in *Engineering Applications of Neural Networks*, C. Jayne e L. Iliadis, cur., Cham: Springer International Publishing, 2016, pp. 213–226, ISBN: 978-3-319-44188-7.

[3] S. Peng, H. Jiang, H. Wang et al., "Modulation Classification Based on Signal Constellation Diagrams and Deep Learning", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, n. 3, pp. 718–727, 2019. DOI: 10.1109/TNNLS.2018.2850703.

[4] K. O'Shea e R. Nash, *An Introduction to Convolutional Neural Networks*, 2015. arXiv: 1511.08458 [cs.NE].

[5] C. Lin, *OFDM modulation classification dataset*, 2021. DOI: 10.21227/xwk3-t431. indirizzo: https://dx.doi.org/10.21227/xwk3-t431.

[6] V. Clerico, J. Gonzalez-Lopez, G. Agam e J. Grajal, *LSTM Framework for Classification of Radar and Communications Signals*, 2023. arXiv: 2305.03192 [eess.SP].

[7] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions", in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.