

# Convolutional Neural Network for epileptic seizure recognition

1 luglio 2019

## Introduzione

Questo progetto è stato realizzato.....

## 1 Epilessia

L'epilessia è un disturbo...

spiegazione  
EEG

## 2 Reti Neurali

Le reti neurali sono modelli matematici basati su neuroni artificiali ispirati al funzionamento biologico del cervello umano. Inventate intorno alla metà degli anni 80, sono tornate in auge nell'ultimo decennio come strumento di risoluzione in svariati campi, dall'informatica, all'elettronica, alla simulazione. Prendendo spunto dal cervello animale, una rete neurale (ANN) è composta da numerosi neuroni connessi fra loro. Ogni connessione, come le sinapsi cerebrali, trasmette un segnale da un neurone all'altro. Il neurone che riceve un segnale può processarlo e inviarlo ai neuroni a cui è a sua volta connesso. Tradizionalmente, un segnale trasmesso tra due neuroni è rappresentato da un numero reale, moltiplicato per un certo peso che indica la forza della connessione. Ogni neurone esegue una somma di tutti i segnali in ingresso, moltiplicandoli per il suo set di pesi. Tale somma viene poi modellata da una funzione di attivazione, tipicamente la sigmoide  $\sigma(x) = \frac{1}{1+e^{-x}}$ .

In una rete neurale i neuroni sono solitamente distribuiti su più layer, che applicano diverse trasformazioni agli input. I segnali in ingresso viaggiano sequenzialmente dal primo layer (input layer) all'ultimo (Output layer), attraversando un numero  $n$  di layer intermedi (Hidden layer).

La peculiarità delle reti neurali rispetto all'utilizzo di sistemi di risoluzione tradizionali è l'apprendimento. Le ANN vengono addestrate per capire come dovranno comportarsi nel momento in cui andranno a risolvere un problema ingegneristico. L'approccio orientato all'apprendimento si basa sui principi

del Machine Learning, la disciplina che studia i diversi metodi matematico-computazionali per apprendere informazioni dall'esperienza. Possiamo distinguere quattro principali metodologie di apprendimento:

- Supervised Learning: la rete riceve un input e il relativo risultato atteso. L'obiettivo è quello di identificare una regola generale che colleghi i dati in ingresso con quelli in uscita.
- Unsupervised Learning: la rete riceve solo set di dati in input, senza alcuna indicazione del risultato desiderato. Lo scopo è quello di risalire a schemi nascosti tra gli input, in modo da identificarne una struttura logica.
- Apprendimento per rinforzo: il comportamento del sistema è determinato da una routine di apprendimento basata su ricompensa se l'obiettivo è raggiunto, o punizione se viene commesso un errore.
- Apprendimento semi-supervised: modello ibrido basato sui primi due, in cui solo ad una parte dei dati in input è associato il rispettivo output atteso.

Nelle reti ad apprendimento supervisionato viene tipicamente usato l'algoritmo di retropropagazione dell'errore

feedforwarding  
e backpropagation

### 3 Reti Neurali Convoluzionali

Il funzionamento delle reti neurali tradizionali è poco efficiente quando deve essere analizzato un numero molto grande di dati. Ogni neurone prevede una connessione con ogni neurone del layer successivo (per questo sono chiamati fully-connected layers). Ciò implica che al crescere della dimensione della rete, cresce inesorabilmente il numero di parametri di cui tener traccia, portando a casi di sovradattamento della rete.

Già dalla fine degli anni 90 i progettisti di reti neurali hanno iniziato ad introdurre dei modelli di reti convoluzionali, che permettono di ridurre di molto la grandezza della rete. Queste sono molto simili alle reti tradizionali: sono anch'esse formate da neuroni e tengono traccia di parametri come pesi e funzioni di attivazione che vengono modificati durante l'apprendimento. La caratteristica differente risiede nei layer convoluzionali di queste reti. Essi sono particolarmente indicati per il riconoscimento di proprietà nell'architettura dei dati in ingresso, che sono trattati come fossero immagini. La convoluzione prende infatti spunto dal funzionamento della corteccia visiva animale. Intuitivamente, un layer convoluzionale ha il compito di riconoscere alcune caratteristiche visive dell'immagine, come contorni, linee, colori ecc.. concentrandosi su piccole porzioni dell'immagine, e non prendendola nel suo complesso come accade nelle reti fully-connected. Una volta individuata una caratteristica in una certa sezione dell'immagine, la rete sarà in grado di riconoscerla se dovesse presentarsi in altri punti. In una successione di layer convoluzionali, inoltre, un layer più

imparare a riconoscere combinazioni di caratteristiche base individuate nei precedenti strati. Ciò le rende particolarmente adatte alla comprensione di pattern complessi.

Un convolutional layer è generalmente formato da un set di filtri (chiamati kernel) della stessa dimensione. Durante la computazione tali filtri vengono applicati ai dati in ingresso, attraverso una moltiplicazione element-wise

conv1d in  
pytorch

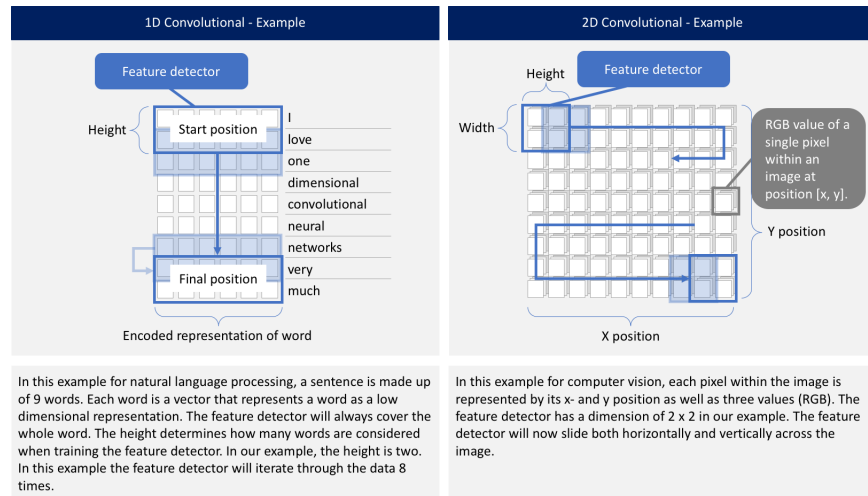


Figura 1: Differenza convoluzione 1D e 2D

## 4 Dataset

Il dataset chb mit...

Provenienza,  
composizio-  
ne, prepro-  
cessing

## 5 Structure Application

L'applicazione creata è suddivisa in due grandi componenti, da una parte il backend che offre servizi e modella la struttura dei dati, dall'altra il frontend che richiede i servizi e ne implementa l'interfaccia grafica, in modo da garantire un'interazione user friendly con l'utente. Questa garantisce anche una più facile manutenzione del codice, visto che ognuna delle due parti è divisa in diversi componenti. L' applicazione è una web app, essa infatti è capace di girare sia nell'ambito web che mobile grazie ai vari framework utilizzati. Essa è una single page application, cioè un'applicazione web che è fruibile su una sola pagina web garantendo una fluidità maggiore nell'esperienza dell'utente.

## 6 Backend

L'applicazione è strutturata come una web application. La parte di backend è utilizzabile indipendentemente dall'interfaccia fornita.

### 6.1 Django Framework

Django è un server-side web framework Python estremamente popolare. Un web framework è un insieme di componenti che rendono lo sviluppo di siti web più facile e veloce. Django aiuta a eliminare tutte quelle attività che vengono ripetute durante lo sviluppo dell'applicazione, rendendo il processo di sviluppo un'attività facile e rapida. Si tratta di un web framework di alto livello scritto in linguaggio Python, sviluppato dalla Django Software Foundation e distribuito open source. Essendo un framework di sviluppo web, quindi con un interfaccia web, si parla di un pattern alla base chiamato MVC, che corrisponde a tre componenti: Model, View e Controller. Anche Django usa questo pattern, però nel suo caso il pattern è chiamato MVT, che corrispondono a tre componenti quali: Model, View, Template.

- Model: corrisponde a un insieme di classi python che descrivono il modello dei dati. Esse forniscono una rappresentazione delle tabelle del database, consentendo di sfruttare gli oggetti per effettuare operazioni CRUD sui dati. Le informazioni delle classi sono contenute nel file *models.py*.

è la parte del programma che si occupa del database. Django fornisce una serie di comandi rapidi per la gestione delle operazioni più comuni.

- Template: corrisponde nella vista dell'applicazione, consente di effettuare l'interazione con l'app web essa descrive tutti i componenti visuali facenti parte della GUI con cui l'utente poi andrà ad interagire. Il Template consiste di file html che descrivono la presentazione dei dati implementando la *View* del pattern.

consiste in una serie di documenti per la generazione di codice HTML, combinando l'utilizzo di parti statiche e tag dinamici. In questo progetto non è stato tuttavia utilizzato alcun template, dato che la parte visuale è gestita esternamente dal framework Ionic.

- View: funzioni python che gestiscono il flusso di esecuzione dell'applicazione, implementando la parte del controller del pattern MVC. Garantisce la possibilità di creare delle pagine e di descriverne il comportamento che esse avranno in funzione dell'iterazione con l'utente. La View descrive e modella il comportamento dell'intera applicazione, rispondendo a eventi scaturiti dall'interazione con l'utente e modellando i dati facente parti del modello

gestisce l'interazione tra utente, sistema e database. Si occupa di ricezione, elaborazione e risposta alle richieste dell'utente.

immagine  
mtv

## 6.2 Python

La sezione di backend del programma è stata scritta in Python, linguaggio di programmazione interpretato, di alto livello e generico. Esso supporta diversi paradgmi di programmazione, come quella procedurale, orientata agli oggetti e funzionale. Di seguito verranno descritte le più importanti librerie utilizzate nel progetto.

## 6.3 PyEDFlib

Il funzionamento base del sistema prevede la lettura di tracciati EEG. Questi vengono comunemente distribuiti nel formato EDF (European Data Format), ideato per lo scambio di segnali multicanali fisici e biologici. La libreria che si occupa dell'interazione con tali file è PyEDFlib, basata a sua volta su edflib e Numpy. Essa mette a disposizione funzioni per la lettura dei segnali del file e dei suoi metadati, tra cui il numero di canali, la lunghezza e la frequenza del campionamento, l'inizio e la fine della rilevazione.

## 6.4 PyTorch

PyTorch è una libreria del linguaggio Python basata sul framework torch. Essa contiene diverse funzioni e metodi scientifici per le applicazioni dedicate al machine learning, al deep learning e al natural processing language. PyTorch offre due funzionalità di alto livello:

- Tensor Computing(come NumPy) con forte accelerazione attraverso unità di elaborazione grafica(GPU)
- Piattaforma di ricerca del deep learning che offre la massima flessibilità e velocità (DA VEDERE) La parte di neural network e deep learning è stata realizzata mediante l'utilizzo del framework PyTorch

## 6.5 API

L'utente ha la possibilità di interagire con il sistema attraverso richieste HTTP

views singolo  
file, view  
traininig

# 7 Frontend

La parte di frontend è stata implementata per garantire la gestione dell'interfaccia grafica che poi andrà ad interagire con l'utente, essa rappresenta il confine del sistema.

## 7.1 Angular

## 7.2 Ionic

### Riferimenti bibliografici

- [1] Nicoletta Boldrini. *Reti neurali: cosa sono e a cosa servono*. 2019. URL: <https://www.ai4business.it/intelligenza-artificiale/deep-learning/reti-neurali/>.
- [2] Jordi Torres. *Convolutional neural networks for beginners*. URL: <https://towardsdatascience.com/convolutional-neural-networks-for-beginners-practical-guide-with-python-and-keras-dc688ea90dca>.
- [3] Wikipedia. *Artificial neural network*. URL: [https://en.wikipedia.org/wiki/Artificial\\_neural\\_network](https://en.wikipedia.org/wiki/Artificial_neural_network).
- [4] Wikipedia. *Convolutional neural network*. URL: [https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network).