

# UNIVERSITÀ DI PISA



Dipartimento di Informatica  
Corso di Laurea Triennale in Informatica

## **Localizzazione Indoor Basata su Beacon Bluetooth a Bassa Potenza Attraverso Tecniche di Deep Learning**

un progetto realizzato per Consorzio Metis e ASL Toscana

**Relatori:**  
**Prof. Gianluigi Ferrari**

**Presentata da:**  
**Marco Pampaloni**

**Sessione Estiva**  
**Anno Accademico 2019/2020**

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Localizzazione Indoor</b>	<b>2</b>
2.1	Introduzione al problema . . . . .	2
2.2	Possibili soluzioni . . . . .	2
2.3	Beacon BLE . . . . .	2
2.4	RSSI e propagazione del segnale . . . . .	2
2.5	Variabilità e rumore di fondo: requisiti di usabilità . . . . .	2
2.6	Installazione dei Beacon e Acquisizione dei Dati . . . . .	2
<b>3</b>	<b>Deep Learning</b>	<b>3</b>
3.1	Neurone Artificiale: Perceptron . . . . .	4
3.2	Multi Layer Perceptron . . . . .	4
3.3	Attivazione: ReLU e Funzioni Sigmoidee . . . . .	4
3.4	Apprendimento: Metodo del Gradiente e BackPropagation . . . . .	4
3.5	Reti Neurali Convoluzionali . . . . .	4
3.6	Regolarizzazione . . . . .	4
3.6.1	Overfitting e Underfitting . . . . .	4
3.6.2	Regolarizzazione L2 . . . . .	4
3.6.3	Dropout . . . . .	4
3.7	Dataset Augmentation e Preprocessing . . . . .	4
3.7.1	Jittering . . . . .	4
3.7.2	Ridimensionamento (Scaling) . . . . .	4

3.7.3	Magnitude Warping . . . . .	4
3.7.4	Permutazione di Sottoinsiemi (Subset Shuffling) . . . . .	4
3.7.5	Deattivazione Selettiva . . . . .	4
<b>4</b>	<b>Architettura Software</b>	<b>5</b>
4.1	TensorFlow . . . . .	6
4.2	Keras . . . . .	6
4.3	Google Colab . . . . .	6
4.4	Weights & Biases . . . . .	6
4.5	Modello di Apprendimento . . . . .	6
4.5.1	Input del Modello . . . . .	6
4.5.2	Blocco Convolutionale . . . . .	6
4.5.3	Uso della Bussola e Output Ausiliario . . . . .	6
4.5.4	Coefficiente di Memoria Residua e Input Ausiliario . . . . .	6
4.5.5	Output del Modello . . . . .	6
4.6	Addestramento del Modello . . . . .	6
4.7	Ensembling . . . . .	6
4.8	Compilazione e Deploy del Modello . . . . .	6
<b>5</b>	<b>Applicazione Mobile</b>	<b>7</b>
5.1	Flutter . . . . .	7
5.2	Planimetrie e Poligoni . . . . .	7
5.3	Backend TensorFlow . . . . .	7
5.3.1	TensorFlow Lite . . . . .	7
5.3.2	Implementazione del Bridge di Comunicazione . . . . .	7
5.4	Stabilizzazione del Modello . . . . .	7
5.4.1	Utilizzo di Sensori Inerziali . . . . .	7
5.4.2	Filtro di Kalman . . . . .	7
<b>6</b>	<b>Conclusioni</b>	<b>8</b>
6.1	Risultati Sperimentali . . . . .	8

6.1.1	Metriche di Errore: MSE, MAE, MaxAE . . . . .	8
6.2	Lavori futuri . . . . .	8
6.2.1	Input a Lunghezza Variabile . . . . .	8
6.2.2	Reti Neurali Residuali . . . . .	8
6.2.3	Variational Autoencoder: Generazione di nuovi dati . . . . .	8
6.2.4	Transfer Learning . . . . .	8
6.2.5	Input Masking e Ricostruzione dei Segnali . . . . .	8
6.2.6	Transformers per Problemi di Regressione . . . . .	8
6.2.7	Simulatore BLE . . . . .	8
6.2.8	Posizionamento Magnetico . . . . .	8

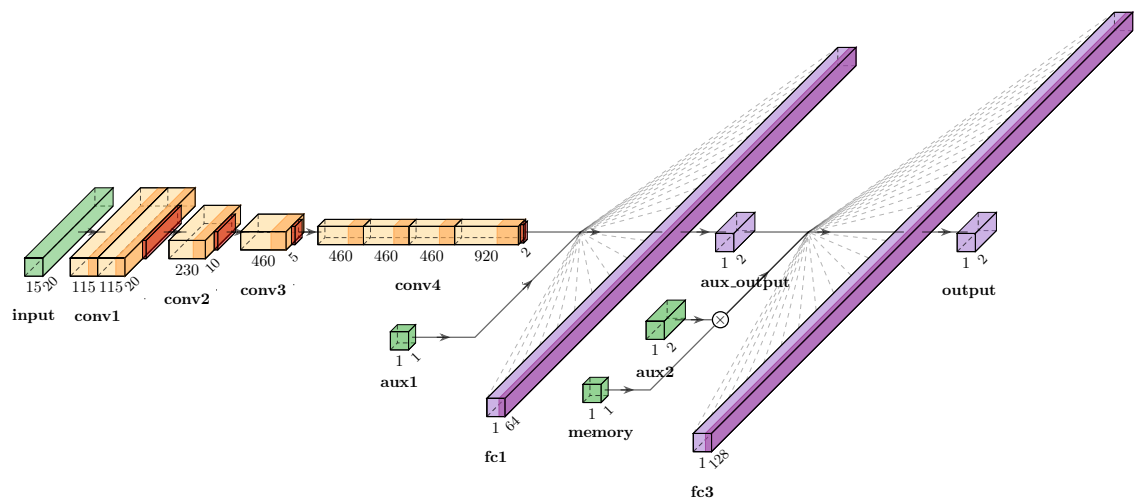


Figura 1: Architettura Neurale

## Sommario

prova abstract

# **Capitolo 1**

## **Introduzione**

# **Capitolo 2**

## **Localizzazione Indoor**

### **2.1 Introduzione al problema**

### **2.2 Possibili soluzioni**

### **2.3 Beacon BLE**

### **2.4 RSSI e propagazione del segnale**

### **2.5 Variabilità e rumore di fondo: requisiti di usabilità**

### **2.6 Installazione dei Beacon e Acquisizione dei Dati**





# Capitolo 3

## Deep Learning

### 3.1 Neurone Artificiale: Perceptron

### 3.2 Multi Layer Perceptron

### 3.3 Attivazione: ReLU e Funzioni Sigmoidi

### 3.4 Apprendimento: Metodo del Gradiente e BackPropagation

### 3.5 Reti Neurali Convoluzionali

### 3.6 Regularizzazione

#### 3.6.1 Overfitting e Underfitting

#### 3.6.2 Regularizzazione L2

#### 3.6.3 Dropout

### 3.7 Dataset Augmentation e Preprocessing

#### 3.7.1 Jittering

#### 3.7.2 Ridimensionamento (Scaling)



# **Capitolo 4**

## **Architettura Software**

### **4.1 TensorFlow**

### **4.2 Keras**

### **4.3 Google Colab**

### **4.4 Weights & Biases**

### **4.5 Modello di Apprendimento**

#### **4.5.1 Input del Modello**

#### **4.5.2 Blocco Convoluzionale**

#### **4.5.3 Uso della Bussola e Output Ausiliario**

#### **4.5.4 Coefficiente di Memoria Residua e Input Ausiliario**

#### **4.5.5 Output del Modello**

### **4.6 Addestramento del Modello**

### **4.7 Ensembling**

# **Capitolo 5**

## **Applicazione Mobile**

### **5.1 Flutter**

### **5.2 Planimetrie e Poligoni**

### **5.3 Backend TensorFlow**

#### **5.3.1 TensorFlow Lite**

#### **5.3.2 Implementazione del Bridge di Comunicazione**

### **5.4 Stabilizzazione del Modello**

#### **5.4.1 Utilizzo di Sensori Inerziali**

#### **5.4.2 Filtro di Kalman**

# Capitolo 6

## Conclusioni

### 6.1 Risultati Sperimentali

#### 6.1.1 Metriche di Errore: MSE, MAE, MaxAE

### 6.2 Lavori futuri

#### 6.2.1 Input a Lunghezza Variabile

#### 6.2.2 Reti Neurali Residuali

#### 6.2.3 Variational Autoencoder: Generazione di nuovi dati

#### 6.2.4 Transfer Learning

#### 6.2.5 Input Masking e Ricostruzione dei Segnali

#### 6.2.6 Transformers per Problemi di Regressione

#### 6.2.7 Simulatore BLE

#### 6.2.8 Posizionamento Magnetico