

# DEEP NEURAL NETWORK

CREATED BY :  
FAM  
THYNK  
UNLIMITED



# TABLE OF CONTENT

01

## Intro DNN

Definition and Operation

02

## Case study

California housing:  
classification problem

03

## DNN training

Activation functions  
and Hyperparameters

04

## Results

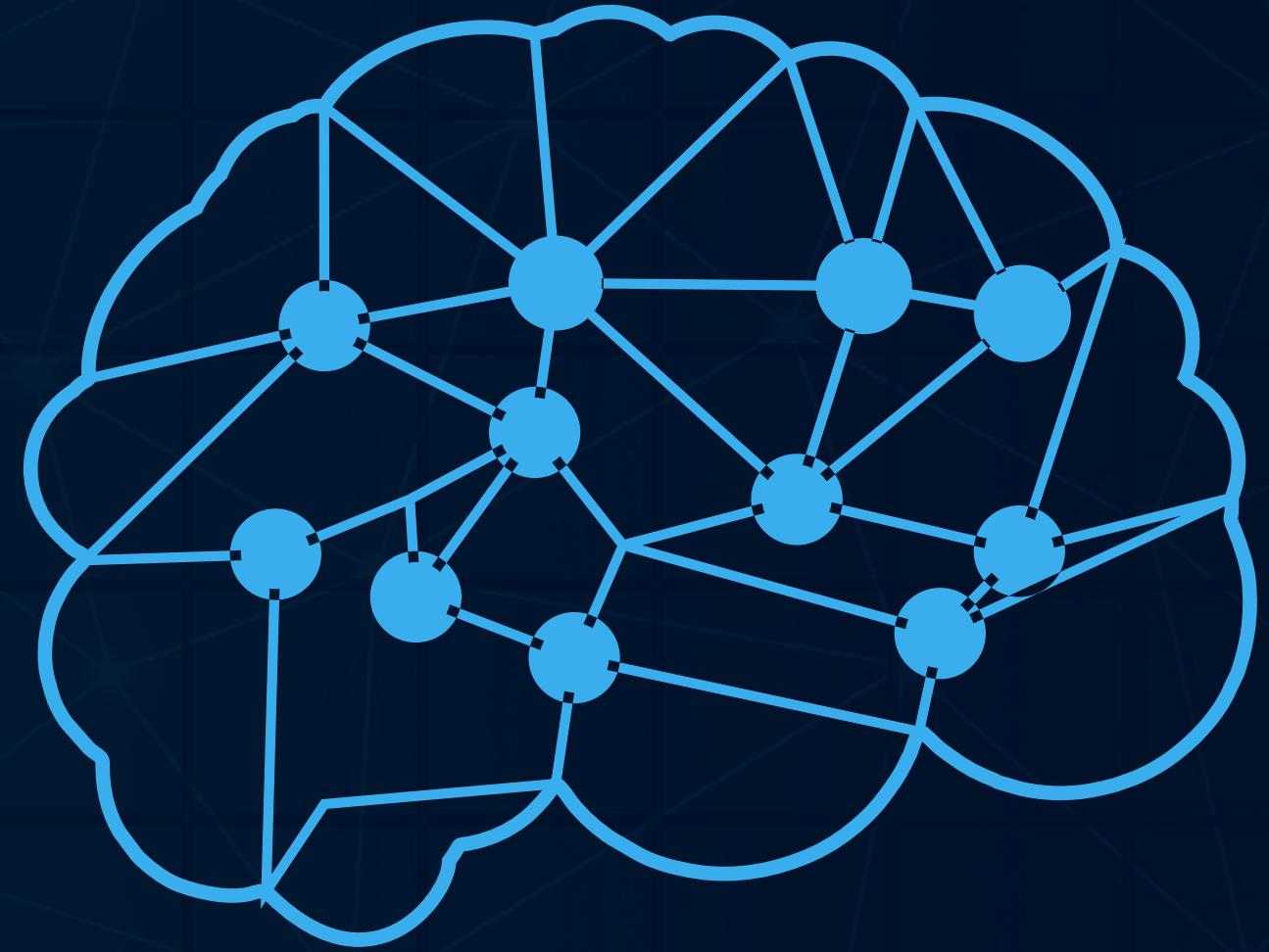
Loss and Accuracy in  
training and test sets



# DEFINITION

Le Deep Neural Network (DNN) sono una classe di reti neurali artificiali che hanno rivoluzionato il campo dell'intelligenza artificiale negli ultimi anni.

Le DNN sono state ispirate dal funzionamento del cervello umano e sono in grado di apprendere rappresentazioni di dati complesse, senza che venga fornita loro una descrizione esplicita delle caratteristiche da cercare.



# OPERATION

Durante la fase di addestramento, le DNN vengono alimentate con un insieme di dati di input e i parametri della rete neurale vengono aggiornati in modo iterativo, al fine di minimizzare la funzione di costo associata alla rete neurale.

Una volta addestrate, le DNN possono essere utilizzate per elaborare nuovi dati di input e produrre i corrispondenti output.

# 2

# CASE STUDY

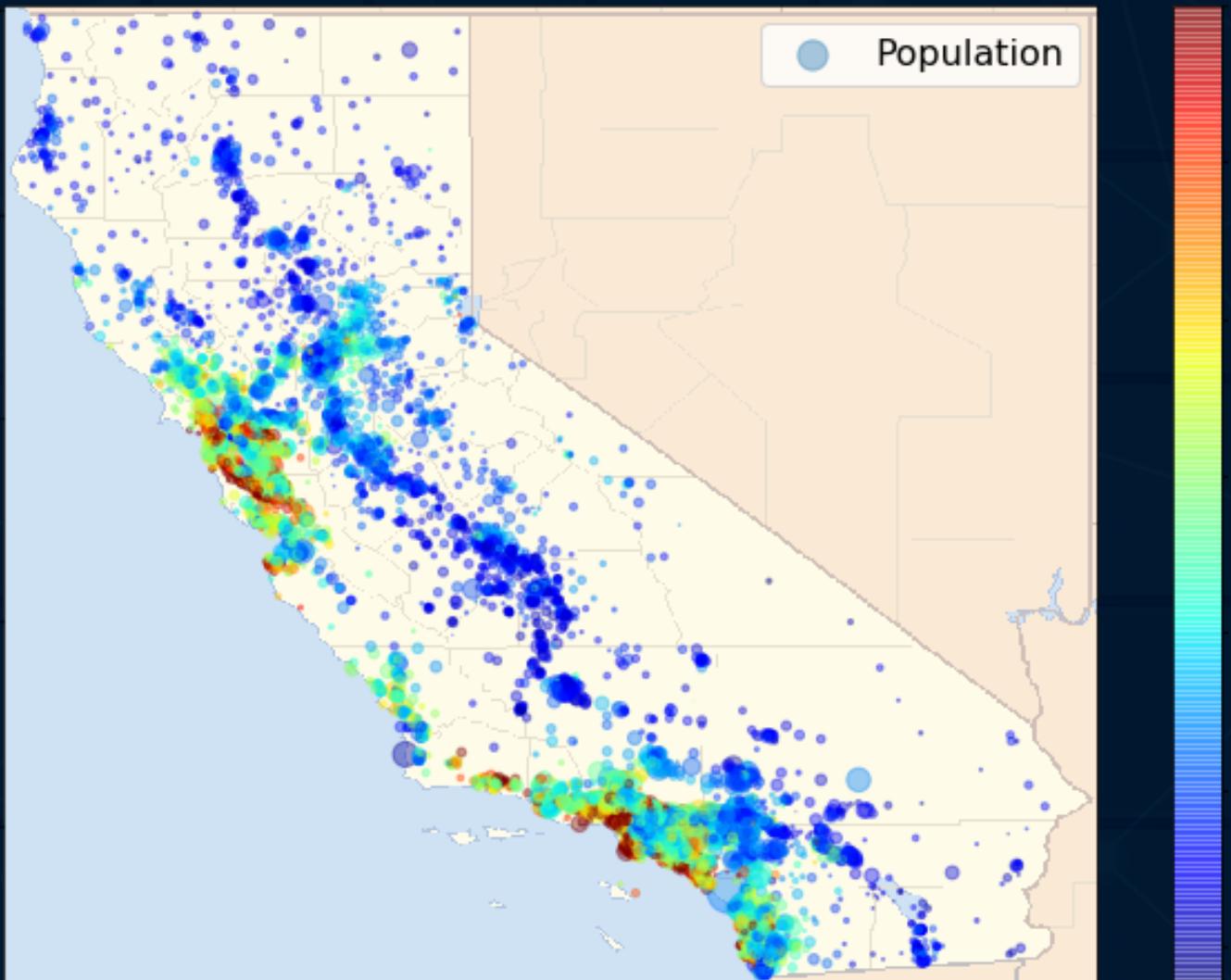
CALIFORNIA HOUSING  
DATASET MEDIAN  
HOUSE:  
CLASSIFICATION  
PROBLEM

## FEATURES

Il California Housing Dataset è un problema di classificazione in cui l'obiettivo è classificare il valore mediano delle abitazioni in 3 categorie:

Economico, Medio e Costoso.

L'obiettivo è costruire un modello di apprendimento automatico che possa classificare in modo accurato le abitazioni in una delle tre categorie, per supportare le decisioni di acquisto di casa e fornire informazioni utili ai professionisti del settore immobiliare.



# VARIABILI DATASET

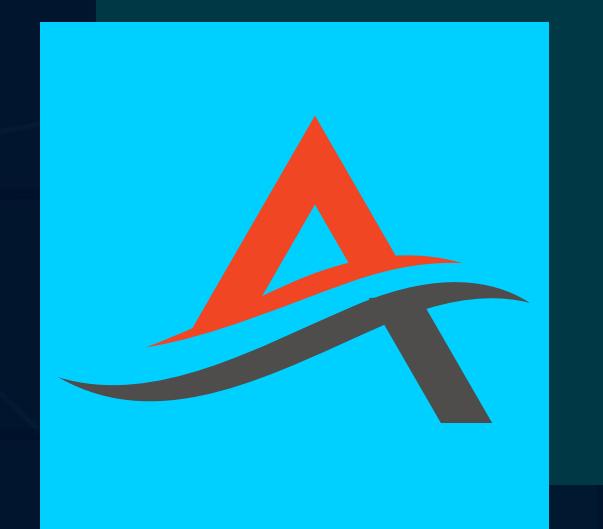
Il California Housing Dataset contiene le seguenti variabili:

- |                              |  |
|------------------------------|--|
| 1. <u>LONGITUDE</u>          | longitudine del quartiere                      |
| 2. <u>LATITUDE</u>           | latitudine del quartiere                       |
| 3. <u>HOUSING_MEDIAN_AGE</u> | età mediana delle abitazioni nel quartiere     |
| 4. <u>TOTAL_ROOMS</u>        | numero totale di stanze nel quartiere          |
| 5. <u>TOTAL_BEDROOMS</u>     | numero totale di camere da letto nel quartiere |
| 6. <u>POPULATION</u>         | popolazione totale del quartiere               |
| 7. <u>HOUSEHOLDS</u>         | numero totale di famiglie nel quartiere        |
| 8. <u>MEDIAN_INCOME</u>      | reddito mediano delle famiglie nel quartiere   |
| 9. <u>OCEAN_PROXIMITY</u>    | distanza del quartiere dal mare                |



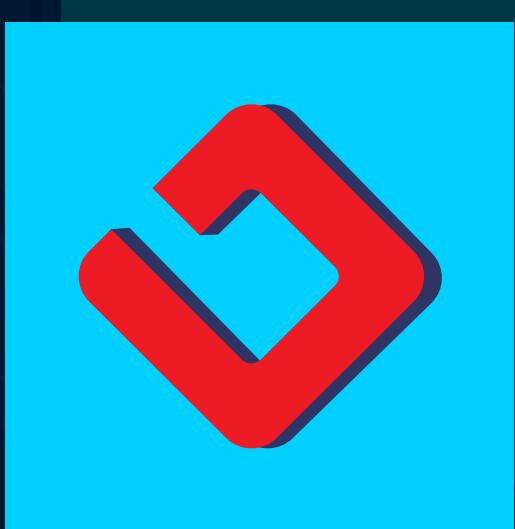
# DNN TRAINING

## (ACTIVATION FUNCTIONS)



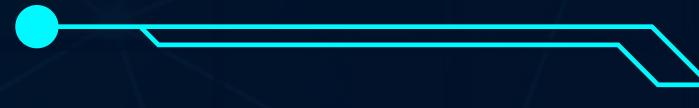
### RELU

La funzione di attivazione "ReLU" ("Rectified Linear Unit") è una funzione non lineare che viene utilizzata comunemente come funzione di attivazione per gli strati nascosti delle reti neurali.



### SOFTMAX

La funzione di attivazione "softmax" viene utilizzata in genere per l'ultimo strato di output delle reti neurali che hanno come obiettivo la classificazione multiclasse, stimando la probabilità che un'istanza appartenga a ciascuna delle classi di output possibili.



## EPOCHS

Le epoches rappresentano il numero di volte che l'intero dataset di addestramento viene utilizzato per addestrare un modello di apprendimento automatico.

## LEARNING RATE

Il learning rate è un parametro di un algoritmo di ottimizzazione utilizzato per regolare l'ampiezza delle modifiche apportate ai pesi della rete neurale durante l'addestramento.

## HYPER PARAM ETERS

### BATCH SIZE

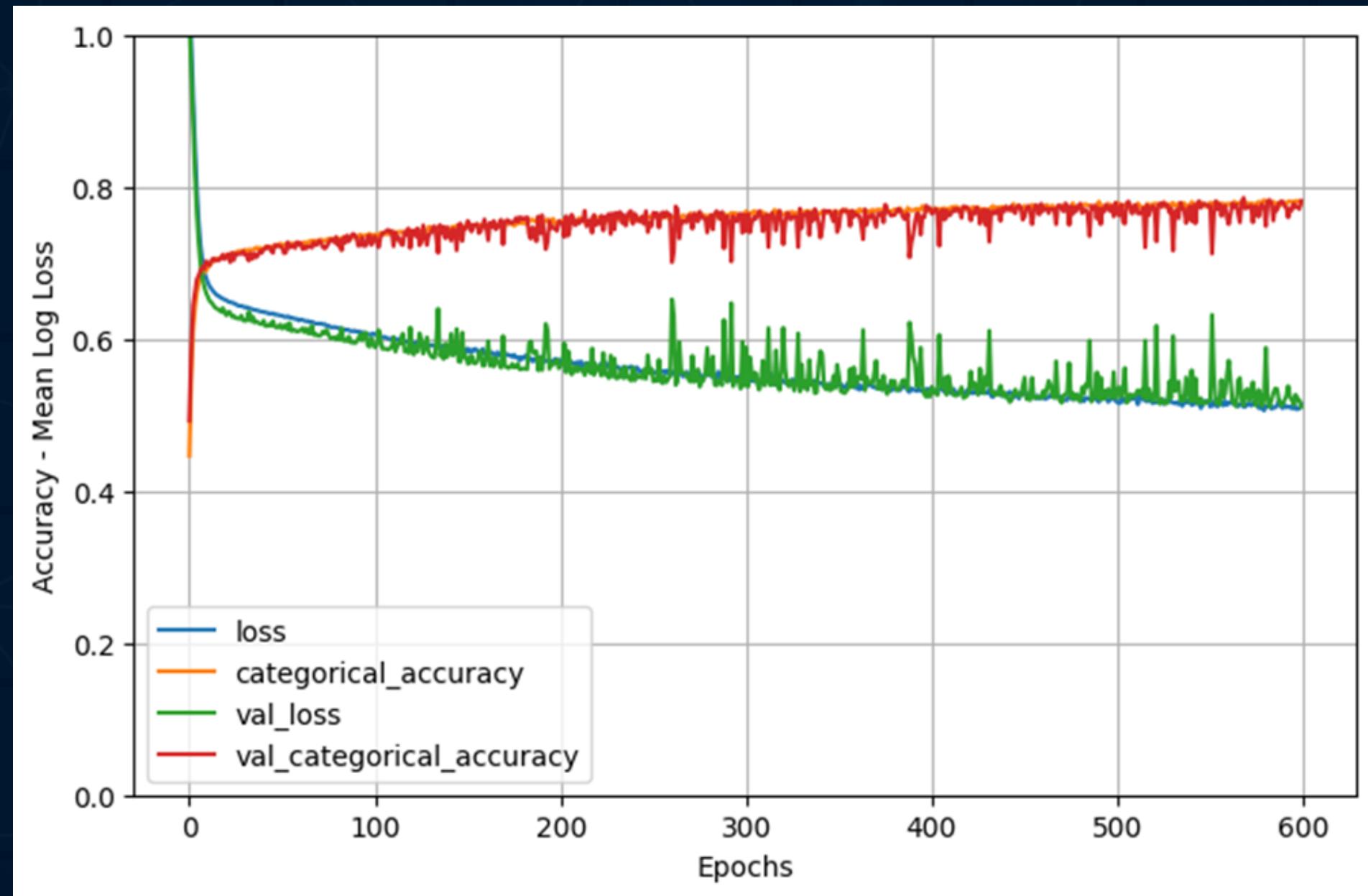
Il batch size rappresenta il numero di campioni di addestramento utilizzati in una singola iterazione durante l'addestramento di una rete neurale.

### NEURONS

I neurons per layer si riferiscono al numero di unità di elaborazione o nodi presenti in uno specifico strato di una rete neurale artificiale.



# RESULTS



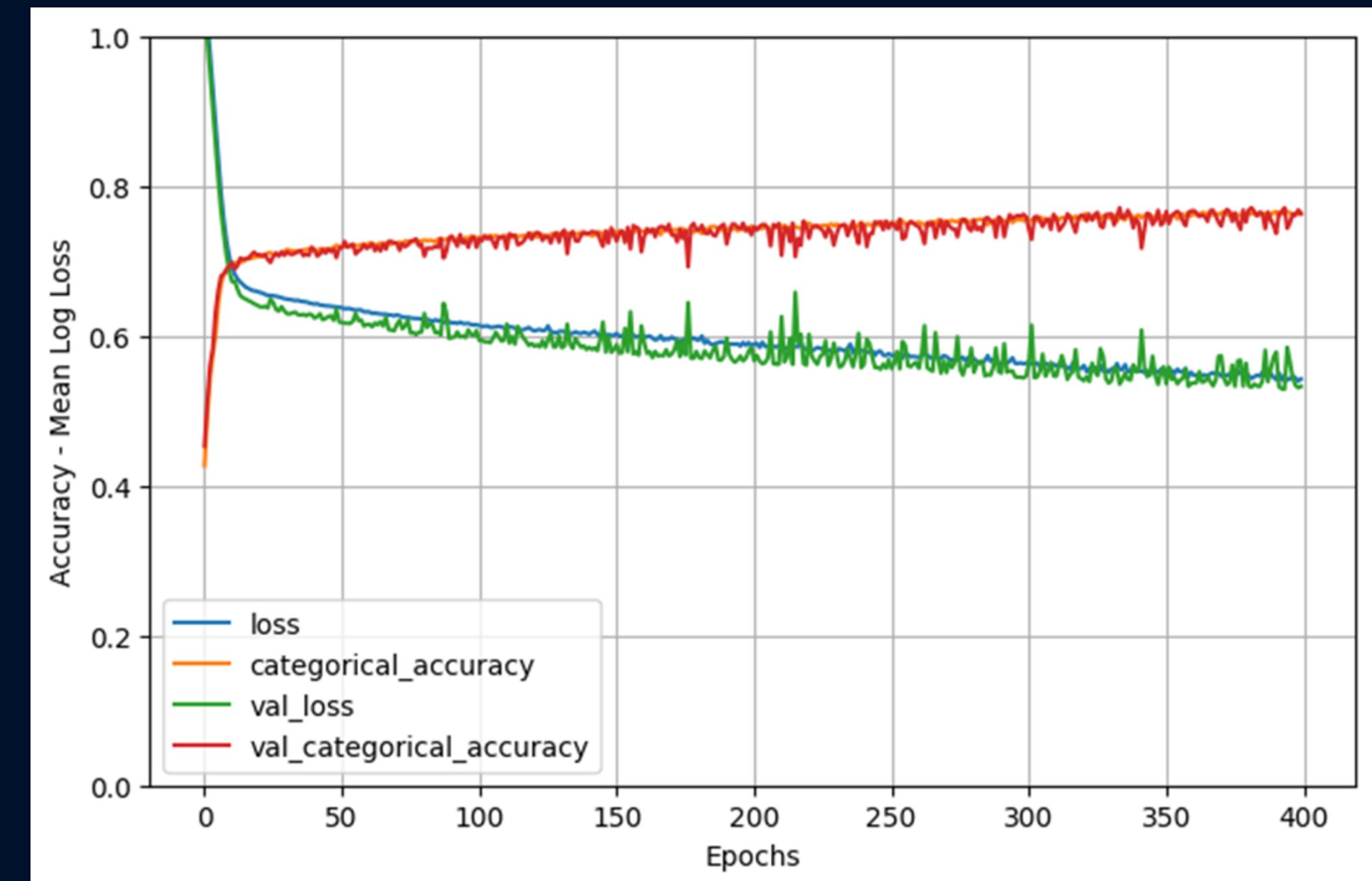
# ATTEMPT 1

N\_EPOCHS = 400

LEARNING\_RATE = 0.003

BATCH\_SIZE = 128

N\_NEURONS\_PER\_HLAYER = [128,16,8]



# TEMPO ESECUZIONE E RISULTATI

2 MINUTI, 27 SECONDI

INDEX	LOSS	CATEGORICAL_AC CURACY	VAL_LOSS	VAL_CATEGORICAL_AC CURACY
399	0.5424	0.7681	0.5325	0.7701

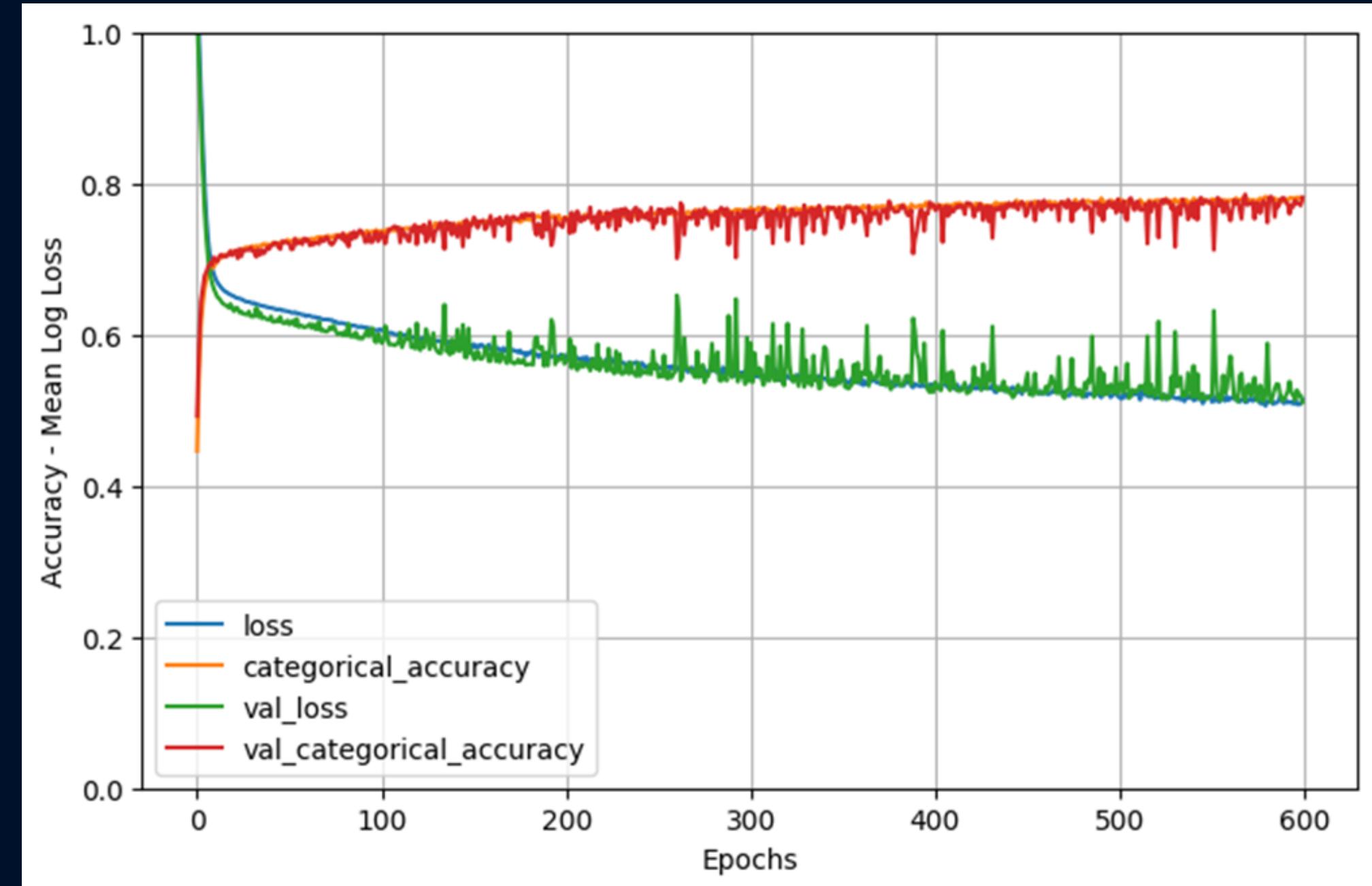
# ATTEMPT 2

N\_EPOCHS = 600

LEARNING\_RATE = 0.01

BATCH\_SIZE = 128

N\_NEURONS\_PER\_HLAYER = [128,32,8]



# TEMPO ESECUZIONE E RISULTATI

5 MINUTI, 54 SECONDI

INDEX	LOSS	CATEGORICAL_AC CURACY	VAL_LOSS	VAL_CATEGORICAL_AC CURACY
599	0.5145	0.7776	0.5104	0.7873

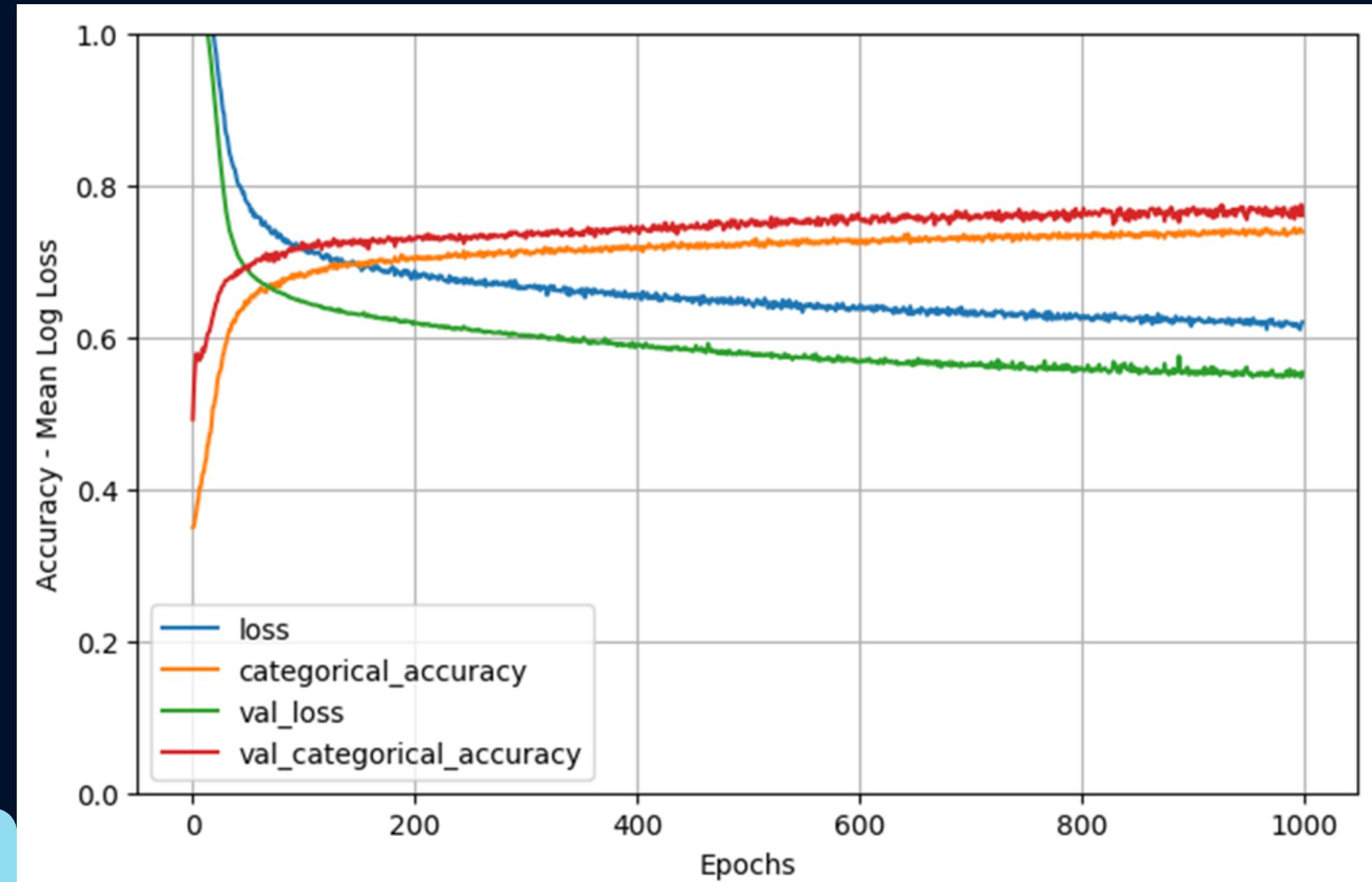
# ATTEMPT 3

N\_EPOCHS = 1000  
LEARNING\_RATE = 0.1  
BATCH\_SIZE = 256  
N\_NEURONS\_PER\_HLAYER = [128,128,64]

DROP OUT = 0.5

## FOCUS DROPOUT

Dropout è una tecnica di regolarizzazione utilizzata per prevenire l'overfitting.  
Essa consiste nel disattivare, in modo casuale, un certo numero di neuroni durante l'addestramento della rete neurale.



# TEMPO ESECUZIONE E RISULTATI

10 MINUTI, 46 SECONDI

INDEX	LOSS	CATEGORICAL_AC CURACY	VAL_LOSS	VAL_CATEGORICAL_AC CURACY
999	0.6195	0.7383	0.5536	0.7744

# ATTEMPT 4

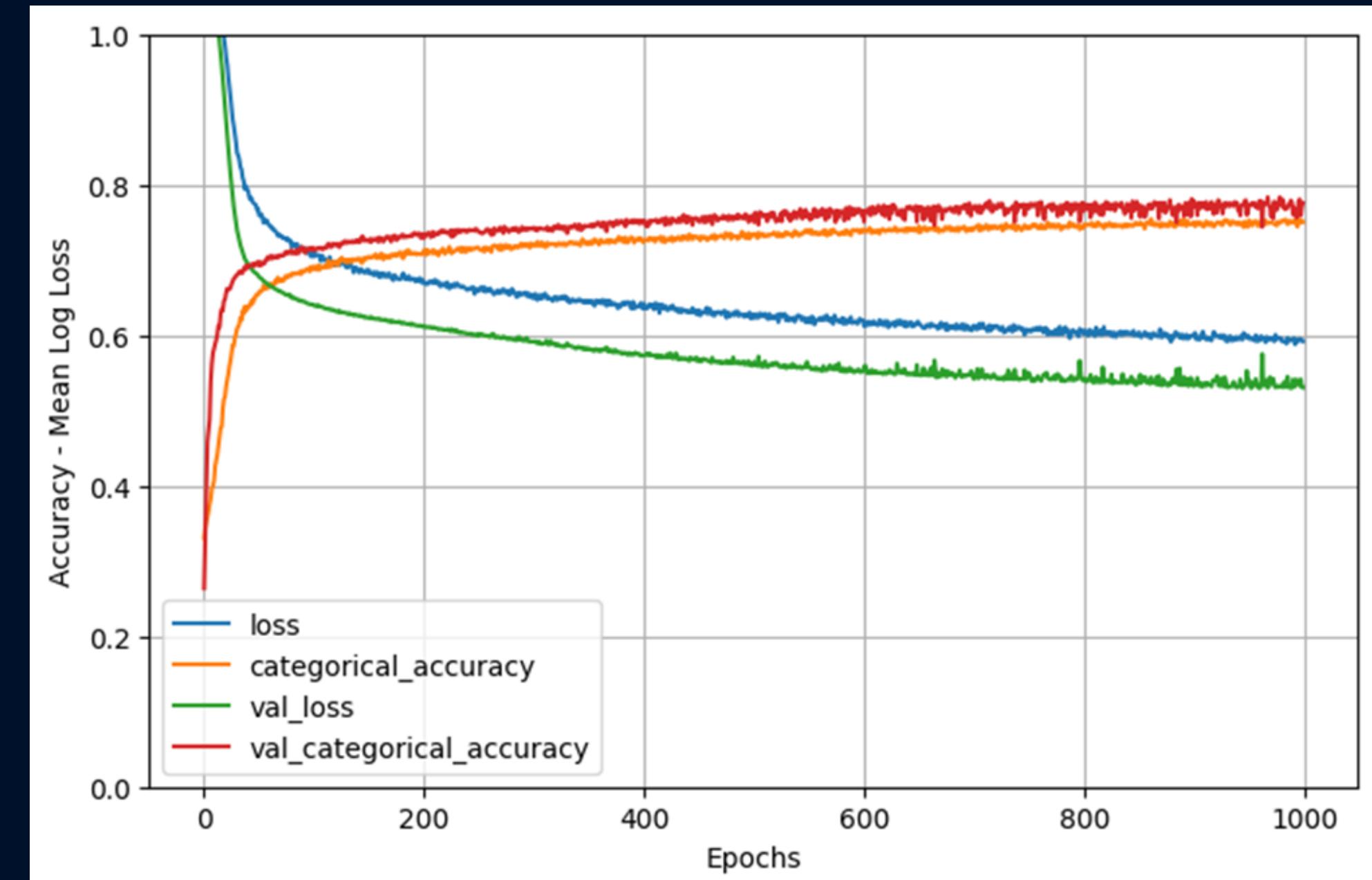
N\_EPOCHS = 1000

LEARNING\_RATE = 0.1

BATCH\_SIZE = 256

N\_NEURONS\_PER\_HLAYER = [256,128,64]

DROP OUT = 0.5



# TEMPO ESECUZIONE E RISULTATI

5 MINUTI, 13 SECONDI

INDEX	LOSS	CATEGORICAL_AC CURACY	VAL_LOSS	VAL_CATEGORICAL_AC CURACY
999	0.5921	0.7506	0.5307	0.7806

# THANKS!

FRANCESCO PALMISANO  
[kekkopalmi918@icloud.com](mailto:kekkopalmi918@icloud.com)

