

Strumenti pratici per il Deep Learning

Ing. Tiziana D'Alessandro, Researcher
Ing. Ciro Russo, PhD
Ing. Emanuele Nardone, PhD Student



MEMBER OF
eut+
EUROPEAN UNIVERSITY
OF TECHNOLOGY



CAMERA DI COMMERCIO
FROSINONE LATINA
AZIENDA SPECIALE
CAMERA DI COMMERCIO FR - LT
informare
internazionalizzazione formazione economia del mare



**CONSORZIO
INDUSTRIALE
DEL LAZIO**

18/03/2025

00 — Topics

■ Tensorflow playground **01**

■ Hugging Face Spaces and GenAI **03**

■ Google Teachable Machine **02**

01 — Tensorflow playground

Introduzione

Parametri

Epoch: 000,000

Learning rate: 0.03 Activation: Tanh Regularization: None Regularization rate: 0 Problem type: Classification

Input

DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE

FEATURES
Which properties do you want to feed in?
+ - 2 HIDDEN LAYERS
4 neurons 2 neurons

Architettura della rete

OUTPUT
Test loss 0.501
Training loss 0.512

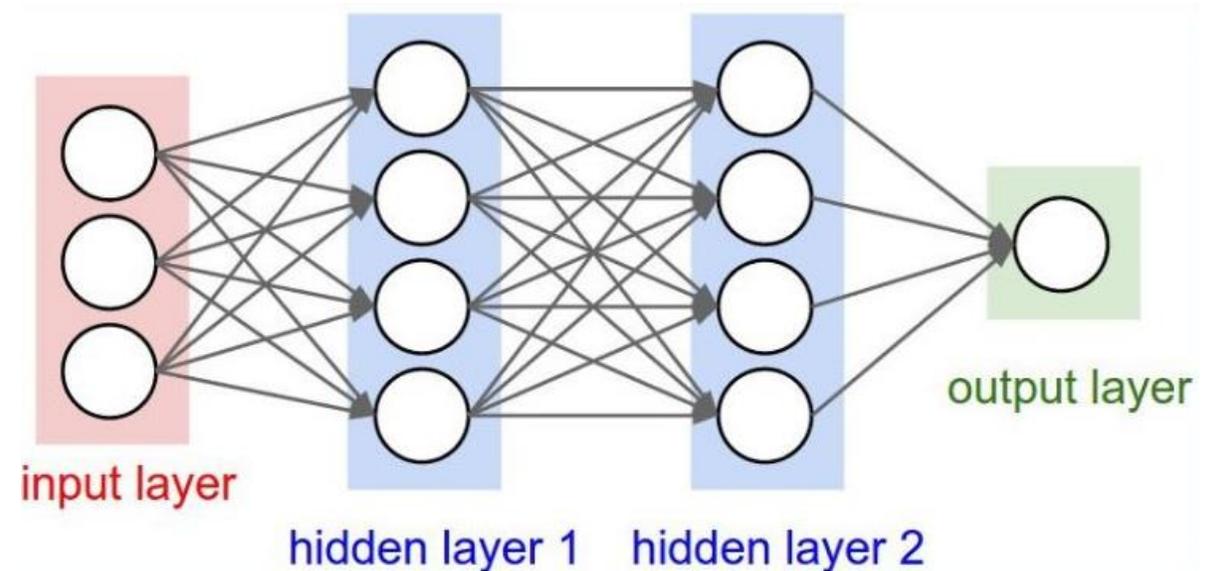
Colors shows data, neuron and weight values. -1 0 1

Show test data Discretize output

The screenshot displays the TensorFlow Playground interface. At the top, there are controls for Epochs (000,000), Learning rate (0.03), Activation (Tanh), Regularization (None), Regularization rate (0), and Problem type (Classification). The main area is divided into three sections: Input, Features, and Output. The Input section includes a 'DATA' dropdown, a 'Ratio of training to test data' slider (set to 50%), a 'Noise' slider (set to 0), and a 'Batch size' slider (set to 10). The Features section shows a neural network architecture with 2 hidden layers, 4 neurons in the first hidden layer, and 2 neurons in the second hidden layer. The network is connected to a 2D plot in the Output section, which shows the test loss (0.501) and training loss (0.512). The plot displays data points (orange and blue) and neuron outputs (blue and orange) on a coordinate system ranging from -6 to 6. A color scale at the bottom indicates values from -1 to 1. The text 'Architettura della rete' is centered below the network diagram.

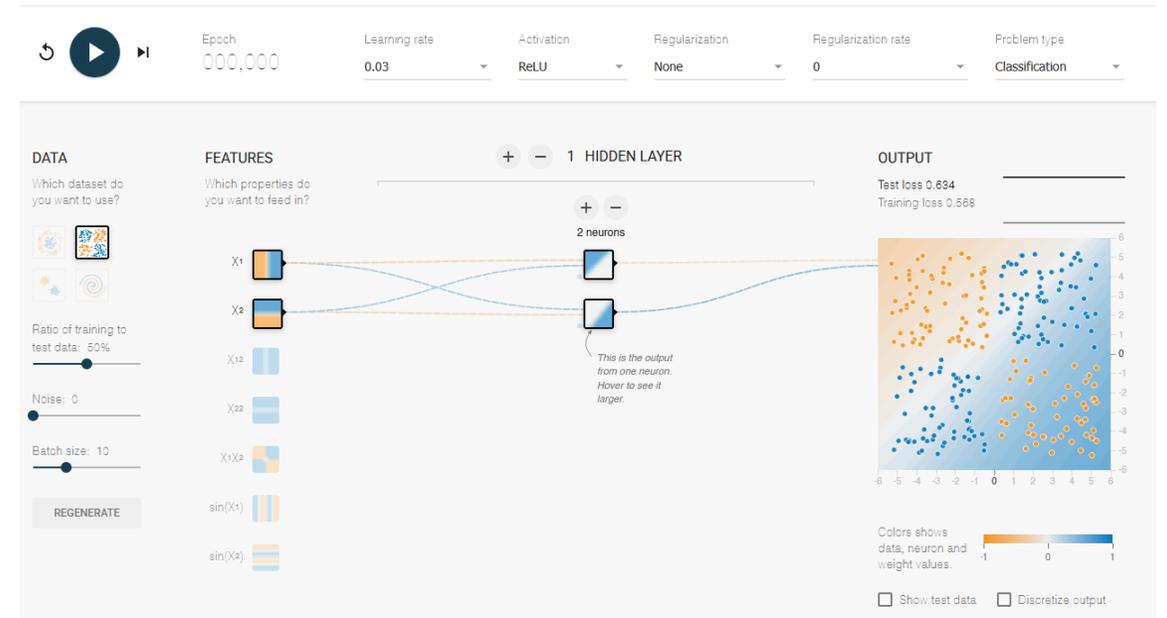
Hidden Layer: Definizione

- Sono componenti fondamentali di una rete neurale che si trovano tra lo strato di input e lo strato di output.
- **Cosa sono:**
 - Sono gruppi di neuroni artificiali che elaborano le informazioni
 - Non sono direttamente visibili dall'esterno della rete
 - Possono essere multipli in una singola rete
- **Come funzionano:**
 - Ricevono dati dallo strato precedente, li trasformano attraverso calcoli matematici e trasmettono i risultati allo strato successivo



Esercizio 1: Dati Lineari vs. Non Lineari

- **Obiettivo:** Dimostrare come le reti neurali gestiscono dati linearmente separabili e non.
- **Configurazione**
 - **Dataset:** XOR
 - **Rete Neurale:**
 - 1 Hidden Layer con 2 neuroni.
 - Funzione di attivazione: ReLU.



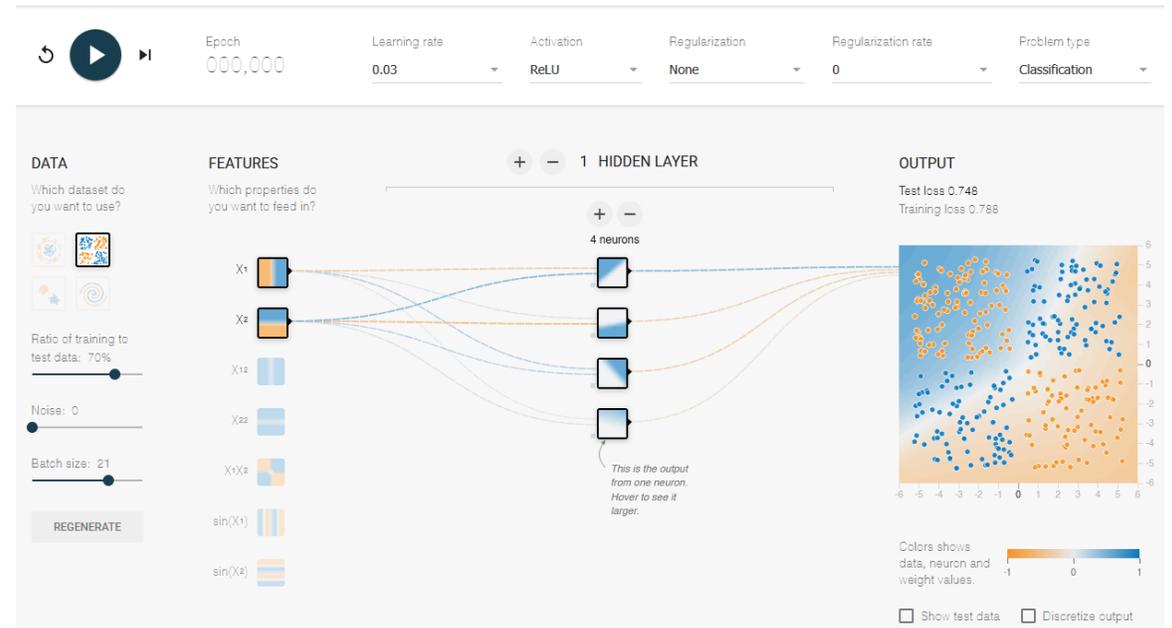
Esercizio 1: Dati Lineari vs. Non Lineari

■ Procedura

1. Addestra la rete e osserva le prestazioni iniziali.
2. Aumenta il numero di neuroni nell'Hidden Layer a 4.
3. Riaddestra e confronta i risultati.

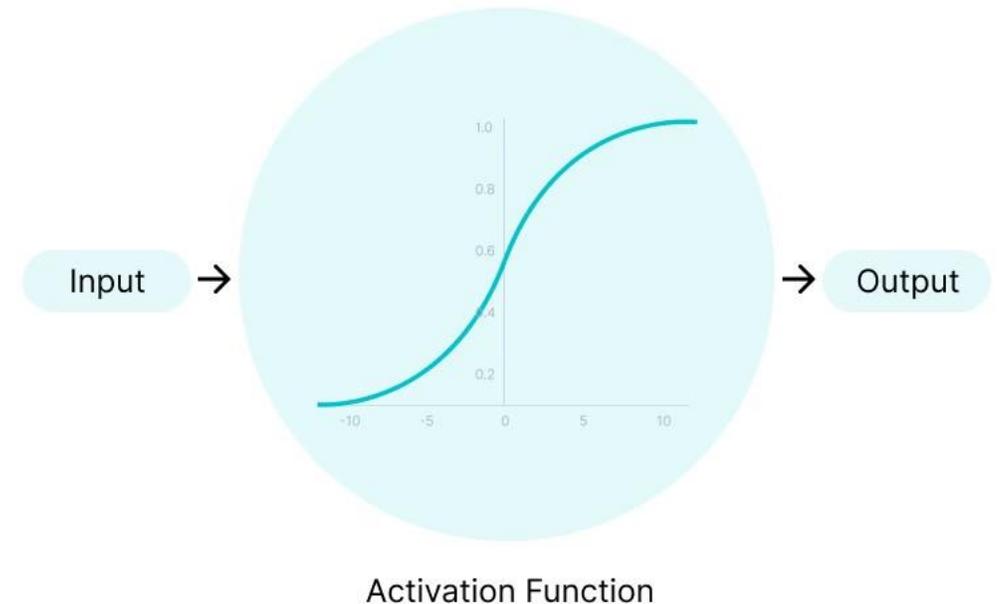
■ Discussione

1. Influenza della complessità della rete sulla capacità di apprendimento.
2. Rischio di overfitting con reti troppo complesse.



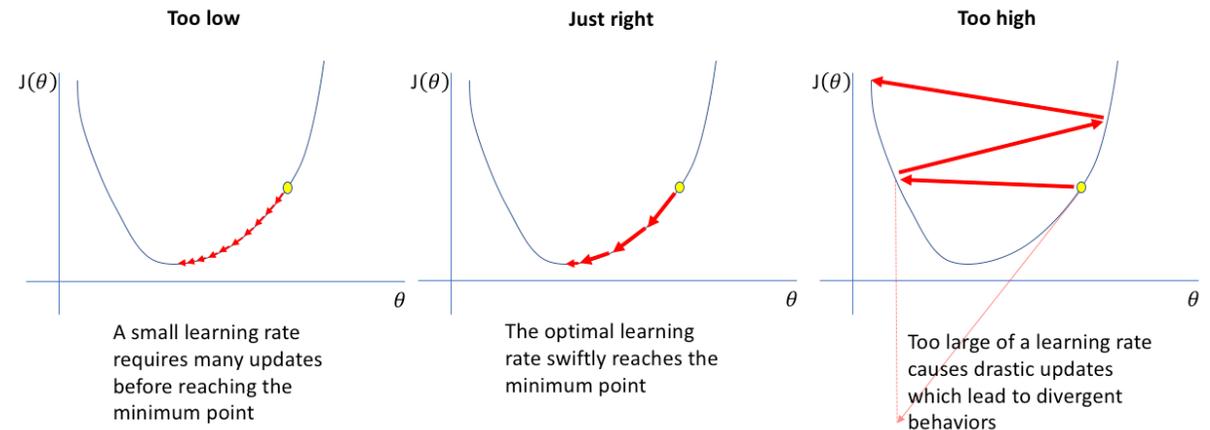
Funzione di attivazione: Definizione

- La funzione di attivazione introduce non-linearità nella rete neurale, permettendole di apprendere relazioni complesse.
- **Cos'è:**
 - Una funzione matematica applicata all'output di ogni neurone
 - Trasforma il segnale in uscita in un modo specifico
 - Esempi comuni: ReLU, Sigmoid, Tanh
- **Perché è necessaria:**
 - Senza funzioni di attivazione, la rete si comporterebbe come una singola equazione lineare



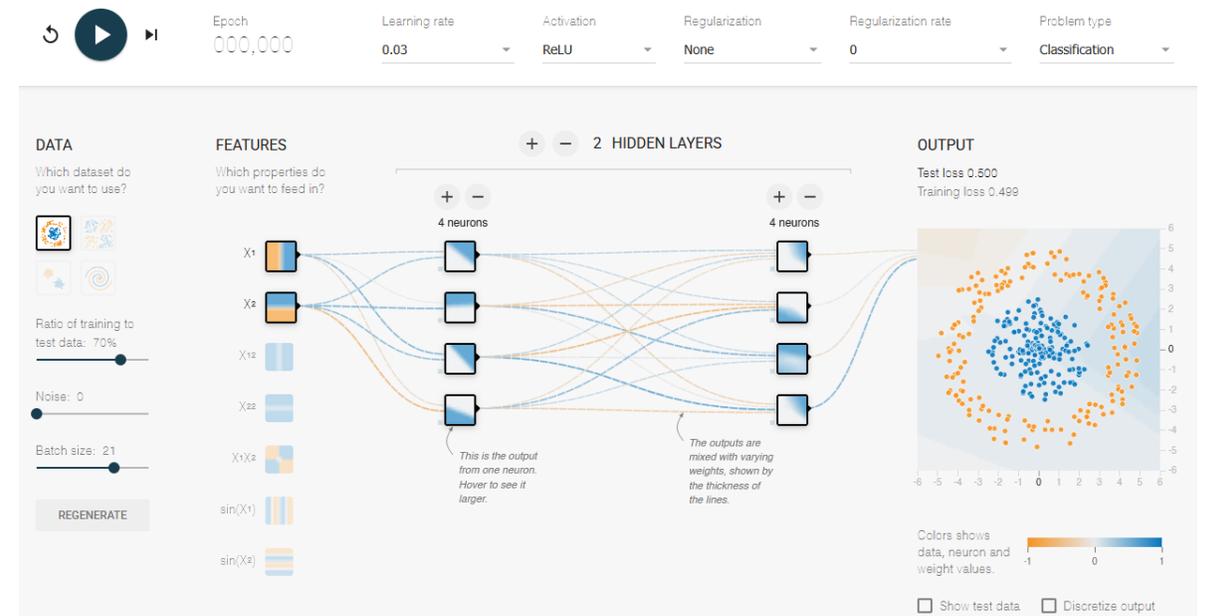
Learning Rate: Definizione

- Il Learning Rate determina quanto velocemente una rete neurale modifica i suoi parametri interni durante l'addestramento.
- Cos'è:**
 - Un numero che controlla la dimensione dei passi durante l'apprendimento
 - Di solito è un valore piccolo
- Learning Rate adeguato permette alla rete di imparare efficacemente



Esercizio 2: Effetto Funzioni di Attivazione

- **Obiettivo:** Mostrare l'impatto delle diverse funzioni di attivazione sulle prestazioni della rete.
- **Configurazione**
 - **Dataset:** Circle
 - **Rete Neurale:**
 - 2 Hidden Layer con 4 neuroni ciascuno.
 - Funzione di attivazione: ReLU.



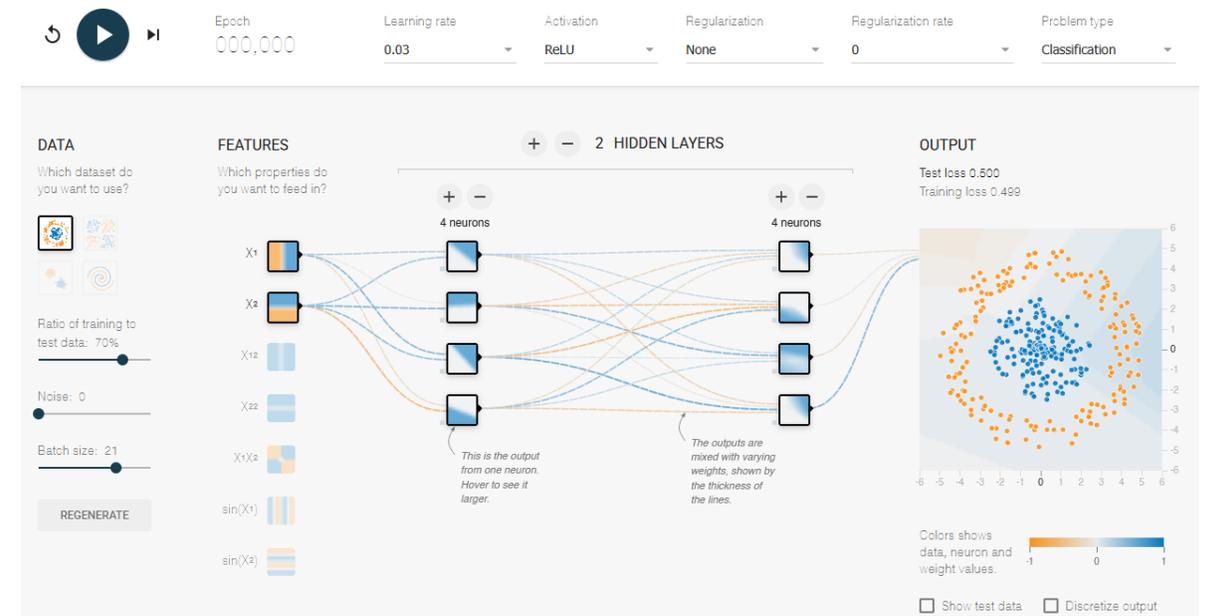
Esercizio 2: Effetto Funzioni di Attivazione

■ Procedura

1. Addestra la rete con ReLU e valuta le prestazioni.
2. Cambia la funzione di attivazione in Tanh.
3. Riaddestra e confronta i risultati.

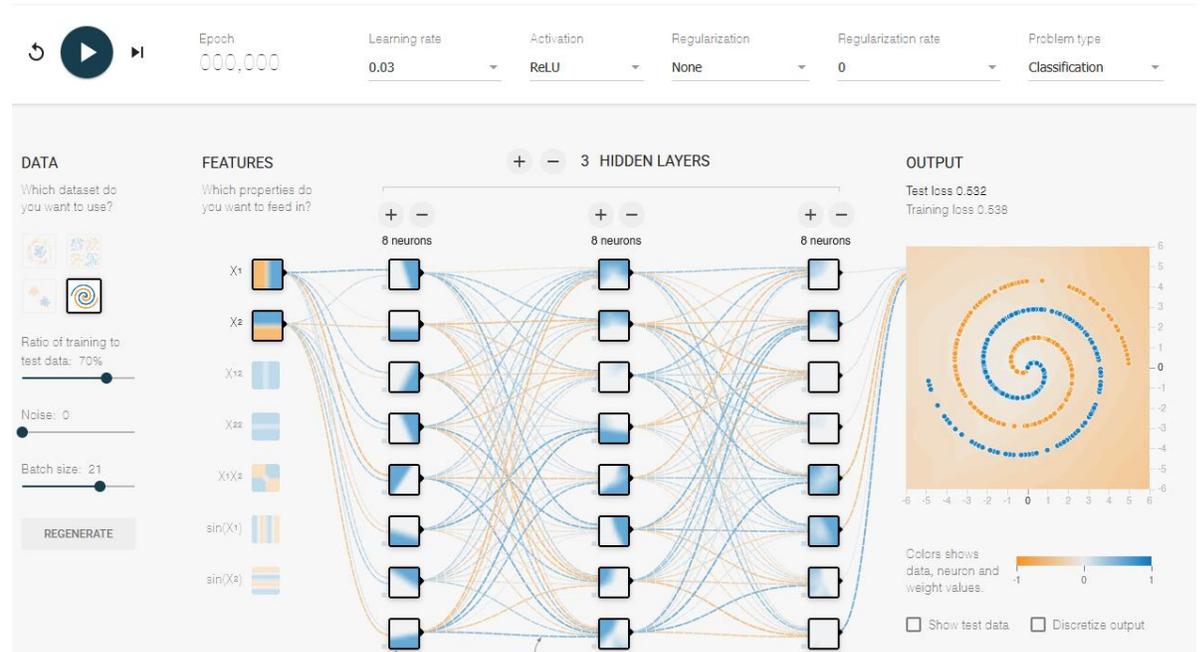
■ Discussione

1. Come le funzioni di attivazione influenzano l'apprendimento.
2. Vantaggi e svantaggi di ReLU vs. Tanh.



Esercizio 3: Impatto del Learning Rate

- **Obiettivo:** Illustrare come il learning rate influisce sulla convergenza dell'addestramento.
- **Configurazione**
 - Dataset: Spiral
 - Rete Neurale:
 1. 3 Hidden Layer con 8 neuroni ciascuno.
 2. Funzione di attivazione: ReLU.



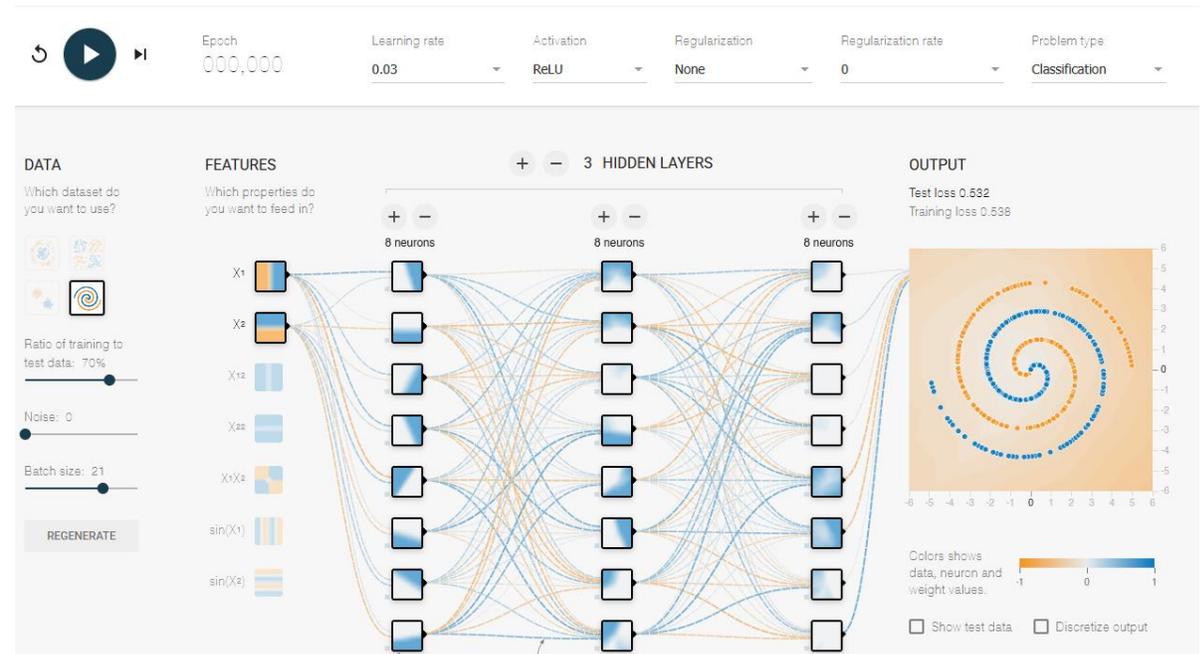
Esercizio 3: Impatto del Learning Rate

■ Procedura

1. Imposta learning rate alto (es. 1.0) e addestra la rete.
2. Osserva la convergenza e le prestazioni.
3. Riduci il learning rate a 0.01.
4. Riaddestra e confronta i risultati.

■ Discussione

- Effetti del Learning rate elevati vs. bassi.
- Importanza della scelta appropriata del learning rate



02 — Google Teachable Machine

Come si usa?

Class 1



Class 2



1 Raccogli

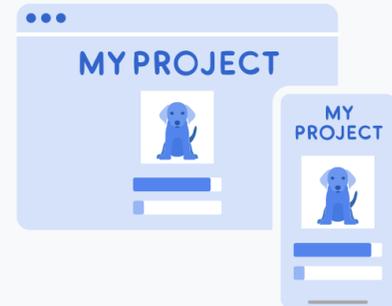
Riunisci e raggruppa i tuoi esempi in classi, o categorie, che vuoi che il computer apprenda.

TRAIN MODEL



2 Addestra

Addestra il tuo modello, poi testalo subito per vedere se è in grado di classificare correttamente nuovi esempi.

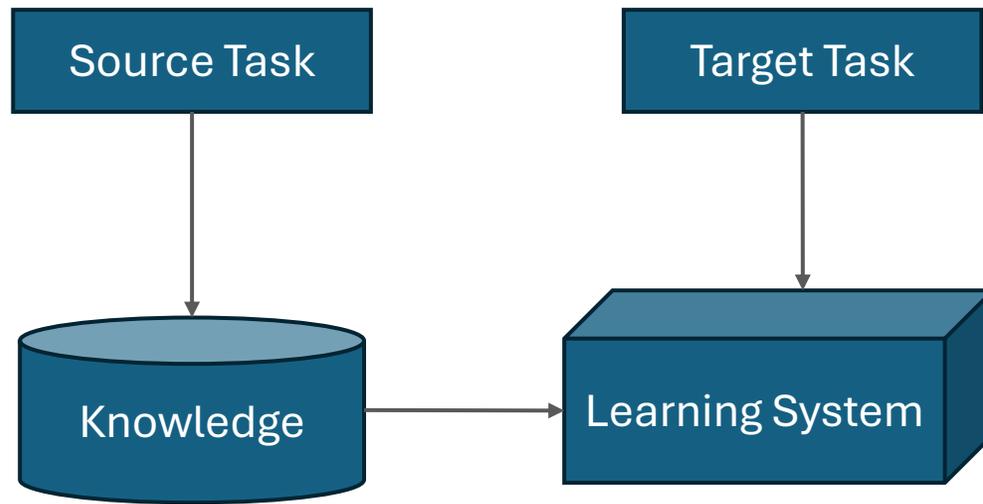


3 Esporta

Esporta il modello per i tuoi progetti: siti, app e molto altro. Puoi scaricarlo o ospitarlo online.

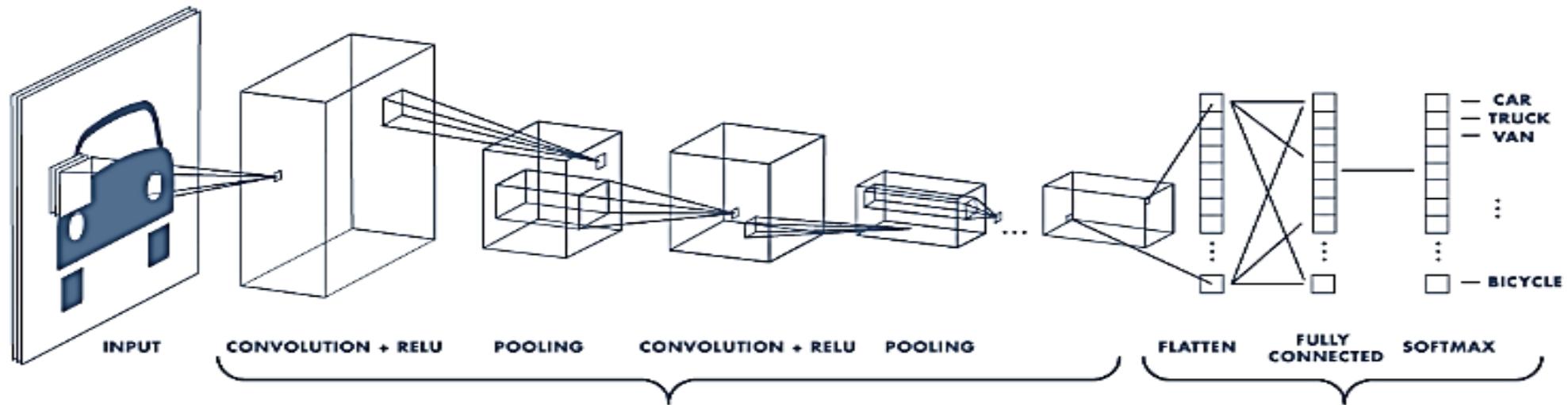
Come si usa?

Teachable Machine di Google consente di addestrare modelli di **rete neurale** in modo semplice e intuitivo, senza necessità di programmare.



Utilizza modelli di **machine learning** pre-addestrati e li riaddestra (transfer learning) con i dati forniti dall'utente.

Deep Neural Networks



FEATURE EXTRACTOR (FE)

Composto da layer convoluzionali e di pooling che analizzano i dati in ingresso, individuando pattern e caratteristiche rilevanti

CLASSIFICATORE (C)

composto da strati completamente connessi e funzioni di attivazione che trasformano le feature estratte in previsioni finali, assegnando una probabilità alle diverse classi

Dataset Split

TRAIN

- utilizzato per addestrare la rete neurale, permettendole di aggiornare i pesi attraverso l'ottimizzazione iterativa basata sull'errore

VALIDATION

- serve a monitorare le prestazioni della rete durante l'addestramento, aiutando a regolare gli iperparametri e a prevenire il sovradattamento

TEST

- utilizzato esclusivamente dopo l'addestramento per valutare le prestazioni finali della rete su dati mai visti, fornendo una stima dell'accuratezza generalizzabile

Dettagli sul Training

Epoca (Epoch)

Un ciclo completo attraverso tutto il dataset di addestramento.

Es. Se il dataset ha 1.000 campioni e addestriamo per 5 epoche, il modello vedrà ogni campione 5 volte.

Batch Size

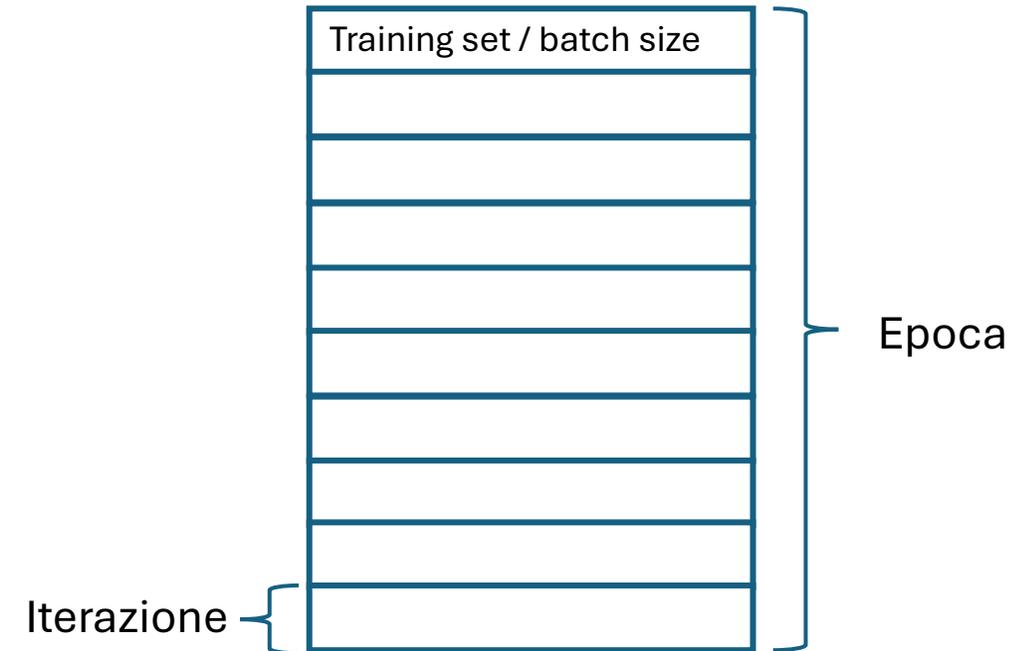
Numero di campioni processati prima di aggiornare i pesi.

Es. Con batch size = 100, il dataset viene diviso in batch da 100 campioni.

Iterazione (Iteration)

Un aggiornamento dei pesi dopo l'elaborazione di una batch.

Es. Se il dataset ha 1.000 campioni e batch size = 100, ci saranno 10 iterazioni per epoca.



$$\text{num. iterazioni} = \frac{\text{Training set samples}}{\text{batch size}}$$

Output di una rete neurale

Confidence Degree

Il confidence degree rappresenta il livello di sicurezza della rete neurale nella sua previsione, espresso generalmente come la probabilità associata alla classe più probabile nell'output del modello.

Predizione

La predizione è il risultato finale prodotto dalla rete neurale dopo l'elaborazione dei dati in ingresso, consistente nell'assegnazione di una classe o di un valore continuo in base al task affrontato, come classificazione o regressione.

Es. Problema di classificazione binario

Specifiche:

- 2 Classi (0 ed 1)
- 2 neuroni di uscita, uno per ogni classe
- Per ogni input i neuroni di uscita forniscono un confidence degree ($cd \in [0, 1]$)

$$\text{Neurone}_0 \rightarrow cd_0 = 0.8$$

$$\text{Neurone}_1 \rightarrow cd_1 = 0.2$$

$$\text{prediction}_{\text{sample}} = 0$$

Esercizio 1: Creazione di un modello di classificazione Binario

Obiettivo:

1. Creare un dataset binario utilizzando Teachable Machine.
 2. Addestrare un modello di classificazione binaria.
 3. Testare il modello e calcolare metriche di valutazione (accuratezza, precisione, recall, F1-score, sensibilità e specificità).
- **Configurazione**
 - **Dataset:** Due classi di immagini catturate con la webcam o caricate (es. “Oggetto Presente” vs. “Oggetto Assente”).
 - **Rete Neurale:** Definisci periodo, batch e lr.
 - **Procedura**
 1. Raggiungi i tre obiettivi richiesti
 2. Modifica il dataset o il modello e valuta i cambiamenti nelle prestazioni
 - **Discussione**
 - Spunti di riflessione

Obiettivo 1: Creazione di un modello di classificazione Binario

- Vai su Teachable Machine, seleziona Image Project per un modello basato sulle immagini.
- Crea due classi (esempio: "Oggetto Presente" e "Oggetto Assente", oppure "Volto Felice" e "Volto Triste").
- Acquisisci almeno 20 immagini per classe utilizzando la webcam o caricando immagini dal tuo dispositivo.
- Controlla il bilanciamento delle classi: cerca di avere un numero simile di immagini in entrambe le categorie.

Obiettivo 2: Addestrare un modello di classificazione Binario.

- Clicca su Train Model e attendi il completamento dell'addestramento.
- Dopo l'addestramento, testa il modello in tempo reale con la webcam o caricando nuove immagini.
- Verifica il comportamento del modello e annota eventuali errori di classificazione.

Obiettivo 3: Testare il modello e calcolare metriche di valutazione

- Raccolta dei dati di test: utilizza almeno 10 immagini per classe che non siano state usate nell'addestramento.
- Crea una matrice di confusione, calcola:
 - **True Positive (TP)**: immagini correttamente classificate nella classe positiva.
 - **False Positive (FP)**: immagini erroneamente classificate come positive.
 - **True Negative (TN)**: immagini correttamente classificate come negative.
 - **False Negative (FN)**: immagini erroneamente classificate come negative.

		Predicted	
		PN	PP
Actual	N	TN	FP
	P	FN	TP

Obiettivo 3: Testare il modello e calcolare metriche di valutazione

- Calcola le seguenti metriche:
 - Accuratezza: misura della performance di un modello di classificazione, calcolata come il rapporto tra il numero di predizioni corrette e il numero totale di campioni valutati.
 - Sensitività (TPR): misura la capacità del modello di identificare correttamente i campioni positivi, ed è calcolata come il rapporto tra i veri positivi e la somma di veri positivi e falsi negativi.
 - Specificità (TNR): indica la capacità del modello di riconoscere correttamente i campioni negativi, ed è calcolata come il rapporto tra i veri negativi e la somma di veri negativi e falsi positivi.

$$Acc = \frac{TP + TN}{P + N}$$

$$TPR = \frac{TP}{P}$$

$$TNR = \frac{TN}{N}$$

Discussione: Spunti di riflessione

- Il dataset è bilanciato? Se no, come potrebbe influenzare i risultati?
- Il modello ha mostrato overfitting o underfitting? Come potresti migliorarlo?
- Quale metrica ritieni più importante per questo problema? Perché?

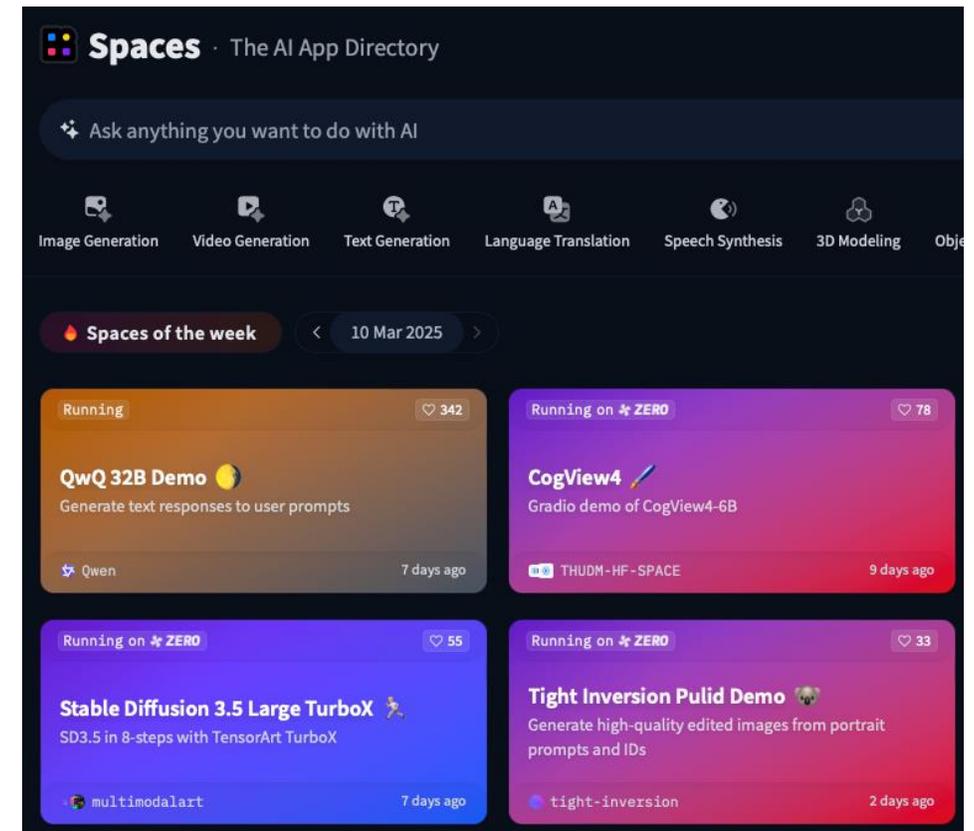
03 — Hugging Face Spaces Generative AI

Hugging Face

Hugging Face è una piattaforma nel settore dell'AI, specializzata in modelli open-source.

Attraverso gli **Spaces**, offre *ambienti cloud* per la *condivisione* e *l'esecuzione* interattiva di modelli AI

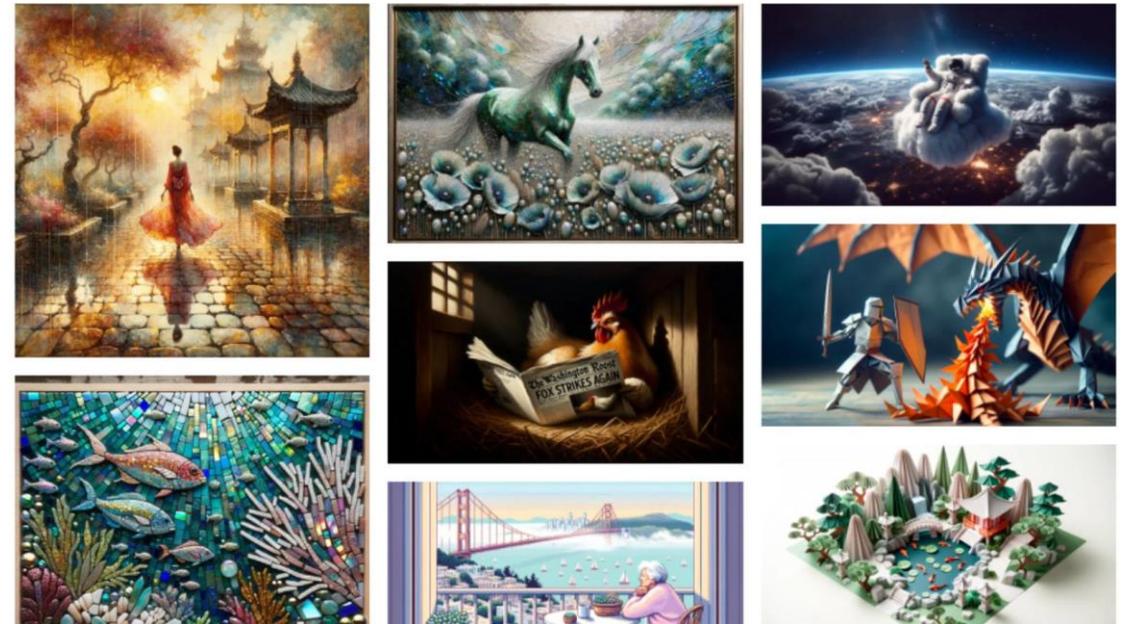
Gli Spaces favoriscono la collaborazione tra sviluppatori, ricercatori e aziende, accelerando l'adozione dell'AI in diversi settori



Generative AI

La Generative AI utilizza modelli per creare contenuti nuovi, come:

- **Testo:** chatbot, scrittura assistita, traduzione automatica (es. [ChatGPT](#))
- **Immagini:** creazione artistica, design, fotorealismo (es. [DALL·E](#)).
- **Audio e Video:** sintesi vocale, musica AI, editing video intelligente (es. [PikaArt](#), [ElevenLabs](#), [Whisper](#))



Dall-E



Tecniche di Image Generation

- **Trasferimento di Stile:** applica caratteristiche stilistiche di un'opera d'arte (es. stile di Van Gogh) ad un'immagine, mantenendo la struttura originale.
- **Outpainting:** espansione di un'immagine oltre i suoi confini originali, aggiungendo nuovi dettagli e mantenendo coerenza visiva.
- **Inpainting:** modifica o ricostruzione di parti di un'immagine, utile per rimuovere oggetti indesiderati o riparare immagini incomplete in modo realistico.

Challenge 3.1

Confronto tra DALL-E e Stable Diffusion
in Hugging Face Spaces nella
generazione di immagini

"A medieval castle on a hill"

*"A medieval castle on a hill at sunset, in
the style of Van Gogh"*

*"A futuristic castle on a crystal hill at
sunset"*



Challenge 3.2

Utilizzo di Stable Diffusion – Inpainting per aggiungere o modificare dettagli nelle immagini:

1. Caricare l'immagine
2. Mascherare l'area da modificare
3. Scrivere un prompt descrittivo
4. Generare l'immagine



Challenge 3.3

Utilizzo di Stable Diffusion – Outpainting per estendere i confini originale delle immagini

1. Caricare l'immagine
2. Scrivere un prompt descrittivo (opzionale)
3. Generare l'immagine



Grazie per l'attenzione!

Ing. Tiziana D'Alessandro, Researcher
Ing. Ciro Russo, PhD
Ing. Emanuele Nardone, PhD Student



MEMBER OF
eut+
EUROPEAN UNIVERSITY
OF TECHNOLOGY



CAMERA DI COMMERCIO
FROSINONE LATINA
AZIENDA SPECIALE
CAMERA DI COMMERCIO FR - LT
informare
internazionalizzazione formazione economia del mare



**CONSORZIO
INDUSTRIALE
DEL LAZIO**

18/03/2025