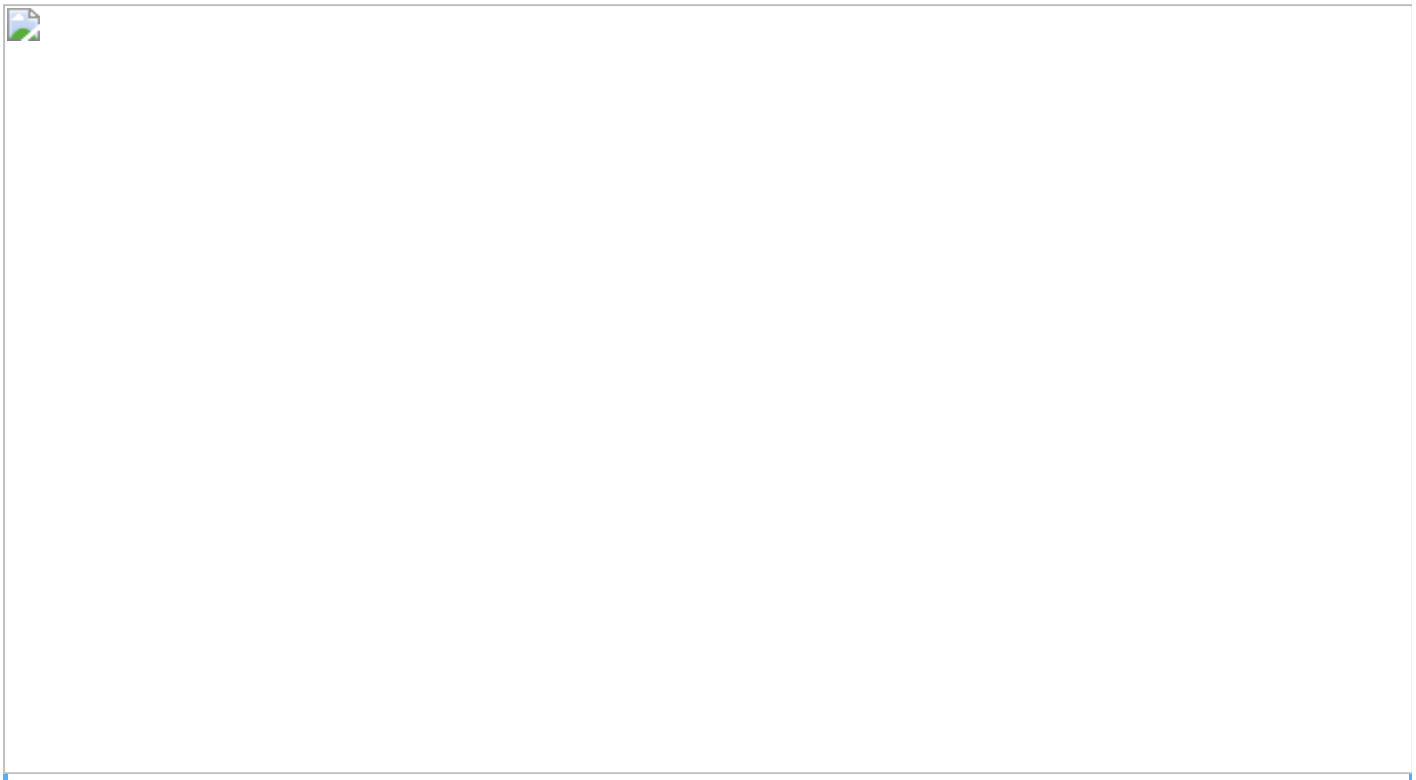
Oltre il codice: comprendere il Machine Learning con Google Teachable Machine (II)

I CONSIGLI DELLO YOUTUBER

DI THEFABLAB

19/03/2026

MACHINE LEARNING GOOGLE TEACHABLE MACHINE



TheFablab è attivo dal 2014 nell'ambito dell'educazione tecnologica e della fabbricazione digitale, progettando percorsi formativi per i giovani e le comunità. Collabora a progetti nazionali con aziende, enti pubblici, scuole e istituzioni culturali per promuovere le competenze STEM, l'alfabetizzazione digitale e la cultura scientifica. Attraverso programmi educativi hands-on, contribuisce a rafforzare il capitale scientifico delle nuove generazioni e l'approccio critico verso le tecnologie emergenti, sostenendo l'empowerment, l'inclusione e il benessere educativo.

Di cosa hanno bisogno gli studenti per diventare utenti consapevoli della tecnologia? Nella prima parte di questa serie di articoli dedicata allo strumento Teachable Machine di Google abbiamo esplorato come questo strumento *no-code* permette di comprendere il ruolo cruciale della **qualità dei dati** in ogni processo di *machine learning*. Le semplici attività didattiche che fanno uso di Google Teachable Machine permettono di **fare luce sui processi che influenzano la qualità del funzionamento di un algoritmo** e sull'importanza di usare dataset ricchi e completi per ottenere un output adeguato.

In questa seconda parte della serie, esploreremo come Teachable Machine può supportare lo sviluppo di un punto di vista più approfondito anche sulle funzioni tecniche che influenzano l'addestramento dei sistemi di machine learning. L'azione sui **parametri avanzati**, infatti, permette di comprendere non solo come imparano gli algoritmi, ma anche come questo processo di apprendimento può essere attivamente guidato e corretto. Manipolare questi valori offre uno sguardo dettagliato sui modi in cui la macchina impara, su come può ottimizzare i propri risultati o, al contrario, fallire a causa di un training approssimativo.

Comprendere la logica profonda dietro il funzionamento degli algoritmi aiuta gli studenti a **decodificare con spirito critico la realtà digitale** che li circonda e ad affrontare sfide etiche attuali. La tecnologia diventa così un territorio di indagine critica e sperimentazione.

I parametri avanzati di Google Teachable Machine

Nel primo articolo della serie abbiamo esplorato le funzioni di base di Google Teachable Machine, tra cui le tre tipologie di addestramento possibili (riconoscimento di immagini, suoni, pose) e gli step necessari per creare un semplice modello. In questo episodio, approfondiremo il ruolo delle **funzioni avanzate**, su cui è possibile agire attraverso il menu a tendina sotto il pulsante *Addestra modello*.



Il pannello **Avanzate** darà accesso a tre parametri modificabili.

- **Periodi (*Epochs*)**. Un periodo è un ciclo completo di addestramento in cui l'algoritmo passa in rassegna tutti i campioni che sono stati caricati nelle classi. La sua **impostazione di default** è 50: questo vuol dire che, se sono state caricate 100 immagini come dataset, il modello vedrà ogni singola foto 50 volte. Possiamo paragonare questo processo al metodo di studio: se leggiamo dalla prima all'ultima pagina un libro di testo per cinque volte, la nostra preparazione sarà più profonda rispetto a quella di chi l'ha letto una sola volta. Aumentare il numero di Periodi, quindi, aiuta a consolidare l'apprendimento del modello. Un numero di Periodi troppo alto, tuttavia, può innescare il fenomeno dell'***overfitting***: questo accade quando il modello impara "*troppo bene*" a riconoscere i dati di addestramento, finendo per attenersi a singoli dettagli specifici invece di astrarre le caratteristiche generali. Nel caso training per immagini, il modello impara quindi a riconoscere le specifiche foto su cui è stato addestrato, ma può fallire nel classificare immagini leggermente diverse. Proseguendo l'analogia con il metodo di studio, l'overfitting è simile a ciò che accade quando adottiamo un apprendimento mnemonico, ma non siamo in grado di traslare i concetti in contesti differenti.



- **Dimensioni del batch**. Questo parametro indica quanti dati vengono mostrati alla macchina prima che **la sua conoscenza sia aggiornata**. Supponiamo per esempio di avere 160 immagini campione e che il parametro *Dimensioni del batch* sia impostato a 16. Le immagini vengono mostrate al modello in gruppi da 16, quindi i dati sono suddivisi in 160/16 = 10 batch. Quando il modello ha memorizzato le immagini contenute in tutti e 10 i batch, ha completato esattamente un periodo. L'impostazione di default su Google Teachable Machine per questo parametro è 16. Per convenzione, **le dimensioni del batch sono solitamente multipli di 16** (32, 64 e così via), in modo da massimizzare l'efficienza di calcolo su GPU. Batch di dimensioni ridotte rendono l'apprendimento più sensibile ai dettagli, mentre aumentare il valore del parametro garantisce stabilità e velocità a scapito dell'attenzione ai particolari.



- **Tasso di apprendimento**. In Google Teachable Machine, il tasso di apprendimento rappresenta il passo con cui il modello corregge gli errori commessi durante il training.Possiamo immaginare la fase di addestramento del modello come un tentativo di raggiungere il punto in cui l'errore è minimo. Per valori troppo alti del tasso di apprendimento, il passo con cui il modello si muove da uno stato a un altro è troppo lungo, incorrendo nel rischio di **overshooting**: in questo caso il modello "salta" oltre il punto di errore minimo, oscillando senza arrivare a una soluzione precisa. Per valori troppo bassi, invece, il modello compie passi molto piccoli e il processo è dunque molto lento. Google Teachable Machine imposta di default questo parametro a 0,001. Poiché si tratta di un valore molto delicato, l'applicazione sconsiglia di alterarlo. In ogni caso, è sempre possibile riportarlo alle impostazioni predefinite attraverso l'apposito pulsante.



Dopo l'addestramento del modello diventerà disponibile il tasto **Funzioni avanzate**, che permette di accedere a grafici riassuntivi sul funzionamento del modello.

La **matrice di confusione**, per esempio, mette a confronto le etichette reali delle classi con quelle predette dal modello: una matrice perfetta ha numeri solo sulla diagonale, mentre la presenza di numeri nei quadranti in cui si incrociano classi diverse può indicare un errore specifico.

Il grafico dell'**Accuratezza per Periodi** mostra quanto bene il modello sta classificando i dati al trascorrere dei Periodi. In un training ideale, la curva sale rapidamente e poi si stabilizza verso valori elevati. Se la curva raggiunge troppo rapidamente il 100%, potrebbe essere un indicatore della presenza di overfitting o delle piccole dimensioni del dataset.

Infine, il grafico della **Perdita per periodo** mostra quanto sono distanti le previsioni del modello rispetto alla realtà. La curva dovrebbe scendere verso lo zero con il procedere dell'addestramento. Se questo grafico oscilla su e giù, potrebbe essere il segno che il Tasso di apprendimento è troppo alto.

L'impatto dei parametri avanzati sull'apprendimento: un esempio di attività pratica

I **Deepfake**, ovvero immagini e video generati artificialmente che risultano quasi indistinguibili dalla realtà, rappresentano una delle frontiere più insidiose dell'IA generativa. In questa attività la classe testerà le possibilità di influenzare il riconoscimento di Deepfake da parte di Google Teachable Machines agendo sulle impostazioni avanzate.

Per prima cosa, iniziate un progetto **Immagini** su Google Teachable Machine. L'attività può essere divisa in quattro step:

1. **Creazione del dataset**.
Create due classi distinte: la classe **Persone reali** conterrà immagini di persone reali (per esempio studenti e docenti) raccolte tramite webcam. La classe **Deepfake** conterrà immagini di **volti sintetici** di persone generate tramite IA.

2. **Addestramento e primo test**.
Addestrate il modello con i dati raccolti usando le impostazioni di default e testate il suo funzionamento. Cosa osservate? Se la barra di confidenza dell'output raggiunge istantaneamente il 100%, sospettate un overfitting: il modello potrebbe aver memorizzato le immagini anziché i tratti distintivi. Salvate i grafici prodotti in questa prima fase nella sezione **Funzioni avanzate**.



3. **Modifica dei parametri**. Intervenite sui parametri avanzati per tenere traccia dei cambiamenti nella reattività del modello. Per esempio, se sospettate un overfitting, potete provare a ridurre i Periodi o ad aumentare le dimensioni del batch. Se il modello ha funzionato bene con i parametri di default, provate comunque a modificare le impostazioni per osservare come cambia la sua risposta.

4. **Secondo test e confronto**. Addestrate nuovamente il modello e confrontate la risposta con l'output della prima fase. Attraverso le Funzioni avanzate, osservate i nuovi grafici e confrontate queste prestazioni con quelle ottenute nella fase precedente.