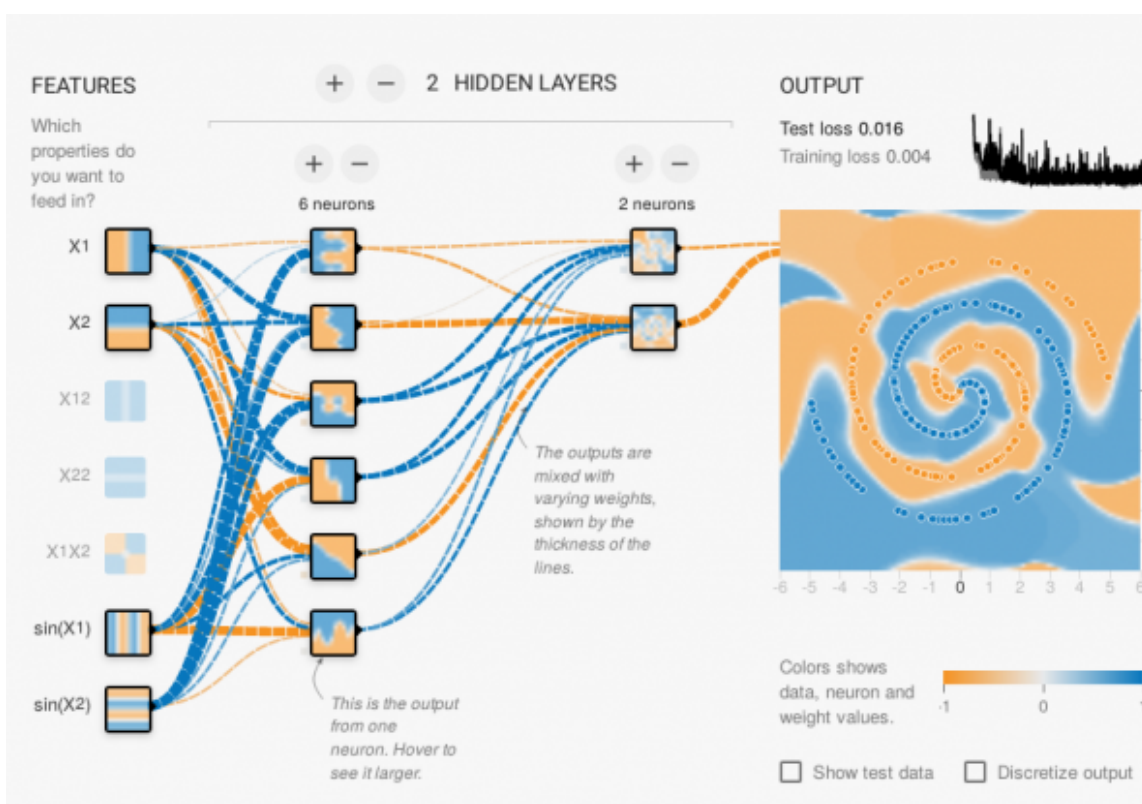


## Attività: Giocare con i neuroni della macchina

Il software online [TensorFlow](#) permette di costruire reti neurali artificiali e di testare le loro risposte per diversi tipi di problemi e su diversi tipi di dati. Nel tipo di problema "Classificazione", l'obiettivo è separare i punti di colore blu da quelli di colore arancione. Un'applicazione di questo tipo di problema è, ad esempio, un algoritmo di classificazione delle foto. Nell'esempio seguente, c'è un input (caratteristica) che separa i punti in orizzontale e un altro che li separa in verticale. Combinando questi due input, si ottiene una separazione obliqua. Il risultato (output) si adatta bene al tipo di dati scelti.



*Flusso tensoriale vista parco giochi*

## TensorFlow: alcune spiegazioni prima di provare la simulazione di una rete neurale

Fonte: traduzione da [sito web francese di Pixees](#)

**Che cos'è una rete neurale e come funziona?** Una rete neurale è un meccanismo generico composto da piccole unità (pseudo-neuroni) collegate tra loro. Ogni unità esegue un'operazione molto semplice: prende i valori in ingresso, li combina in modo molto semplice



(una semplice media con coefficienti) e applica una trasformazione al risultato (ad esempio, mantiene solo i valori positivi).

I coefficienti utilizzati per ponderare la media sono i parametri di questo algoritmo. È la combinazione di un numero molto, molto elevato di queste unità che consente di eseguire operazioni molto complesse. Una rete di tali "neuroni" si ottiene accumulando diversi strati di tali unità. L'input è costituito dai dati che vogliamo elaborare. Questi vengono trasformati attraverso tutti gli strati e l'ultimo strato fornisce come output una previsione su questi dati, ad esempio per rilevare la presenza di un volto in un'immagine. La rete neurale è quindi una funzione parametrizzata con molti coefficienti (chiamati "pesi") ed è la scelta di questi pesi che definisce l'elaborazione effettuata.

**Dove sono i neuroni in TensorFlow?** Nell'interfaccia web di TensorFlow è possibile creare facilmente una rete di una dozzina di neuroni, ciascuno con un numero di parametri compreso tra 3 e 10. Il risultato calcolato dipende quindi da centinaia di parametri. L'output calcolato dipende quindi da centinaia di parametri oltre alle due coordinate (x,y) del punto di ingresso. Nell'interfaccia, ogni quadrato rappresenta un neurone e il colore del pixel di coordinate (x,y) nel quadrato rappresenta l'uscita del neurone quando si mette (x,y) in ingresso alla rete. Se c'è un solo neurone in uscita, viene rappresentato con un quadrato più grande a destra della rete. I parametri della rete sono inizializzati con valori casuali.

**Ma come si imparano i pesi?** L'apprendimento supervisionato consiste nel fornire esempi di dati con la soluzione da trovare, in modo da addestrare la rete a regolare i pesi come richiesto. Nell'esempio della figura precedente, si tratta di una serie di punti in un quadrato, ciascuno con un colore previsto (blu o arancione), con l'obiettivo di prevedere il colore del punto in una determinata posizione. Per trovare i parametri in questione si utilizza un algoritmo classico di aggiustamento progressivo dei pesi.

Il pulsante "play" in alto a sinistra dell'interfaccia permette di lanciare l'algoritmo e di osservare l'evoluzione dell'output della rete neurale durante il processo di "apprendimento": il colore dello sfondo del neurone di uscita tende ad assumere il colore dei punti di addestramento disegnati sopra di esso. Un'altra parte del set di dati viene poi utilizzata per testare la qualità della funzione risultante della rete. Una curva in alto a destra mostra il tasso di errore dei dati utilizzati per l'apprendimento (per verificare che i pesi siano stati regolati correttamente) e il tasso di errore degli altri dati di prova (per verificare che ciò che è stato appreso si generalizza bene a nuovi dati). I pulsanti sul lato sinistro consentono di regolare la distribuzione dei dati tra l'insieme di addestramento e quello di test e anche di aggiungere errori ai dati (dati rumorosi) per vedere se il meccanismo è robusto a questi errori.

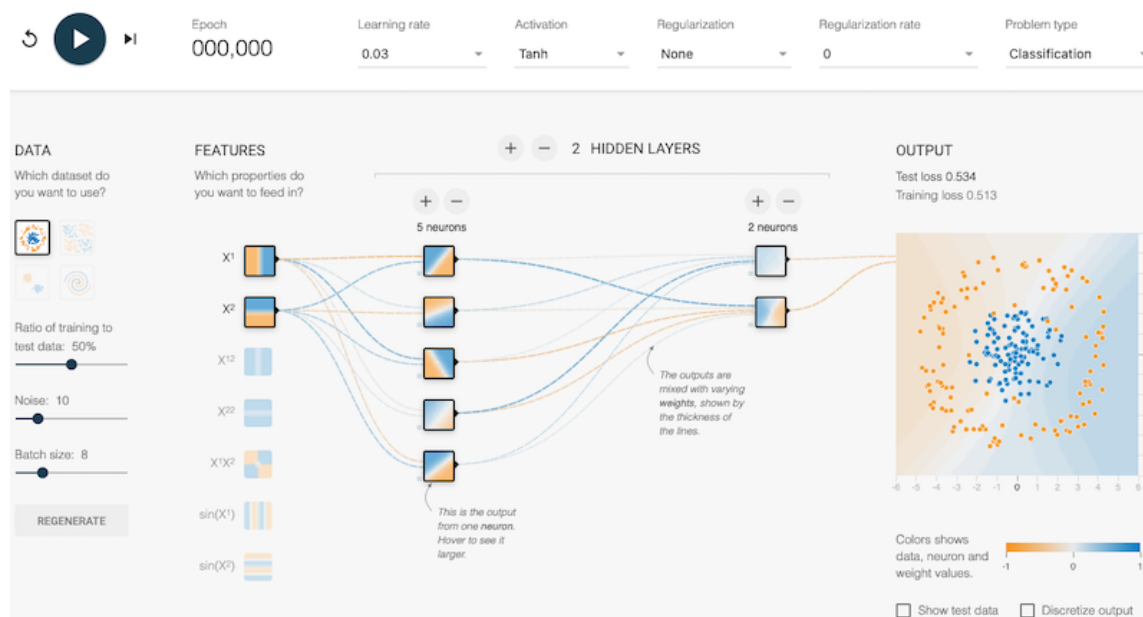
In pratica, riusciamo a trovare parametri soddisfacenti, ma non esiste un vero e proprio quadro teorico per formalizzare tutto questo, è una questione di sperimentazione: scegliere il giusto numero di neuroni, il giusto numero di strati di neuroni, quali calcoli preliminari aggiungere come input (ad esempio moltiplicando gli input per aumentare i gradi di libertà del calcolo). Questo tipo di tecniche può produrre risultati impressionanti nella pratica, come nel riconoscimento vocale o di oggetti in un'immagine.



Tuttavia, capire perché (e come) si ottengono questi buoni risultati è ancora una questione scientifica piuttosto aperta.

## Prova TensorFlow

Cliccate sull'immagine sottostante per accedere all'applicazione TensorFlow in una nuova finestra.



*TensorFlow playground view*