



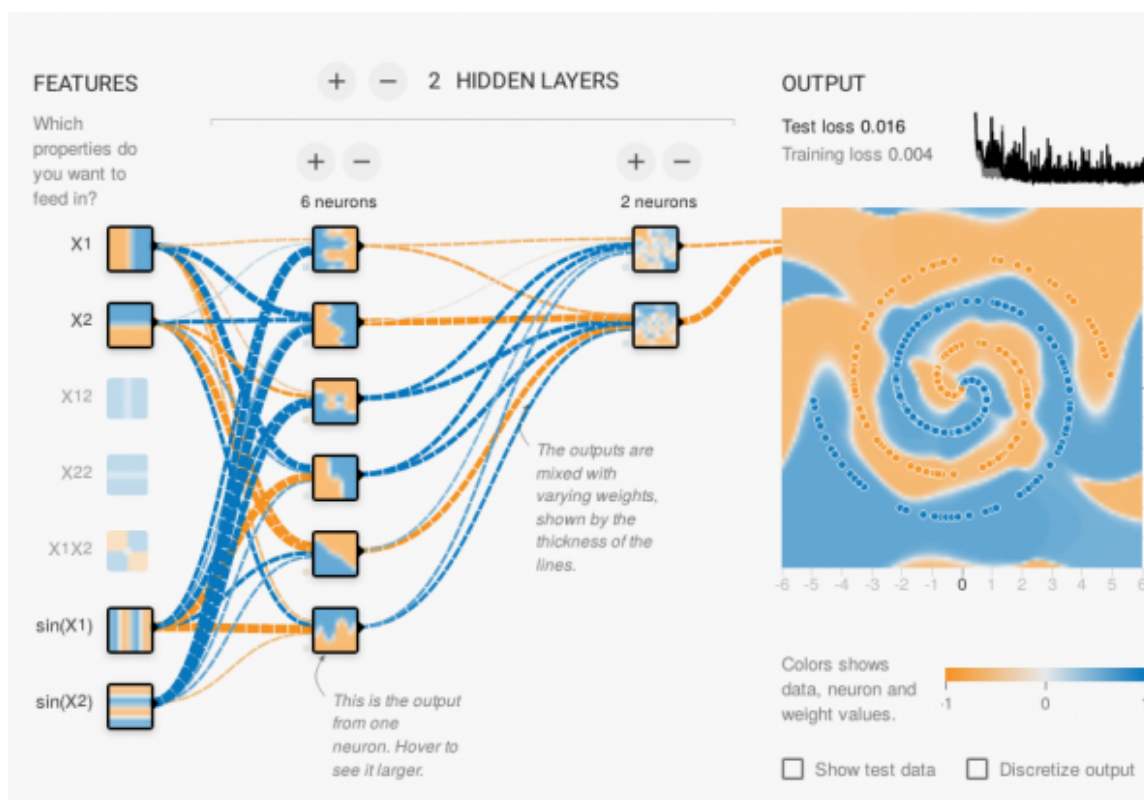
SL ▾

Copy link 🔗

i Metadata

DEJAVNOST: IGRAJTE SE Z NEVRONI STROJA

Spletna programska oprema **TensorFlow** omogoča gradnjo umetnih nevronske mreže in preizkušanje njihovih odzivov za različne vrste problemov in na različnih vrstah podatkov. Pri vrsti problema "razvrščanje" je cilj ločiti modro in oranžno obarvane točke. Uporaba te vrste problema je na primer algoritem za razvrščanje fotografij. V spodnjem primeru je en vhodni podatek (značilnost), ki ločuje točke vodoravno, in drugi, ki ločuje navpično. Z združitvijo teh dveh vhodov dobimo poševno ločevanje. Rezultat (izhod) je dobro prilagojen izbrani vrsti podatkov.



Pogled na igrišče tenzorskega toka

TENSORFLOW: NEKAJ POJASNIL, PREDEN POSKUSITE SIMULACIJO NEVRONSKE MREŽE

Izvir: prevod iz Pixejeve francoske spletne strani



Ko je nevronska omrežje in kako deluje? Nevronska omrežje je splošni mehanizem, sestavljen iz majhnih enot (psevdonevtronov), ki so med seboj povezane. Vsaka enota izvaja zelo preprosto operacijo: sprejme vhodne vrednosti, jih zelo preprosto združi (preprosto povprečenje s koeficienti) in na rezultat uporabi transformacijo (npr. ohrani samo pozitivne vrednosti).

Parametri tega algoritma so koeficienti, ki se uporabljajo za ponderiranje povprečja. Kombinacija zelo, zelo velikega števila teh enot je tista, ki omogoča izvajanje zelo zapletenih operacij. Omrežje takšnih "nevtronov" dobimo z akumulacijo več slojev takšnih enot. Vhod so podatki, ki jih želimo obdelati. Ti se preoblikujejo skozi vse plasti, zadnja plast pa kot rezultat poda napoved teh podatkov, na primer za ugotavljanje, ali je na sliki obraz. Nevronska mreža je torej parametrizirana funkcija s številnimi koeficienti (imenovanimi "uteži"), izbira teh uteži pa določa izvedeno obdelavo.

Kje so nevroni v programu TensorFlow? V spletnem vmesniku TensorFlow lahko preprosto ustvarite omrežje z ducatom nevtronov, od katerih ima vsak od 3 do 10 parametrov. Izračunani rezultat je tako poleg dveh koordinat (x,y) vhodne točke odvisen od več sto parametrov. Na vmesniku vsak kvadraterk predstavlja nevtron, barva pike s koordinatama (x,y) v kvadratu pa predstavlja izhod nevtrona, ko vnesemo (x,y) v vhod omrežja. Če je na izhodu samo en nevtron, je predstavljen z večjim kvadratom na desni strani omrežja. Parametri omrežja so inicializirani z naključnimi vrednostmi.

Ale kako se naučimo teh uteži? Nadzorno učenje je sestavljeno iz zagotavljanja primerov podatkov z rešitvijo, ki jo je treba poiskati, da se omrežje usposobi za ustrezno prilagajanje teh uteži. V primeru na zgornji sliki gre za vrsto točk v kvadratu, od katerih ima vsaka pričakovano barvo (modro ali oranžno), z namenom napovedati barvo točke na določenem mestu. Za iskanje zadevnih parametrov se uporablja klasični algoritem postopnega prilagajanja uteži. Ta algoritem se zažene z gumbom "play" v zgornjem levem kotu vmesnika, izhod nevronske mreže pa se med procesom "učenja" spreminja: barva ozadja izhodnega nevtrona prevzame barvo učnih točk, ki so narisane nad njim. Drug del nabora podatkov se nato uporabi za testiranje kakovosti funkcije, ki jo dobi omrežje. Krivulja zgoraj desno prikazuje stopnjo napake podatkov, uporabljenih za učenje (za preverjanje, ali so se uteži pravilno prilagodile), in stopnjo napake drugih testnih podatkov (za preverjanje, ali se naučeno dobro posplošuje na nove podatke). Gumbi na levi strani omogočajo prilagoditev porazdelitve podatkov med učno in testno množico ter dodajanje napak podatkom (šumni podatki), da bi preverili, ali je mehanizem odporen na te napake.

V praksi nam uspe najti zadovoljive parametre, vendar ni pravega teoretičnega okvira za formalizacijo vsega tega, temveč je to stvar eksperimentiranja: izbira pravega števila nevtronov, pravega števila slojev nevtronov, katere predhodne izračune dodati kot vhod (na primer pomnoževanje vhodov za povečanje stopenj prostosti za izračun).

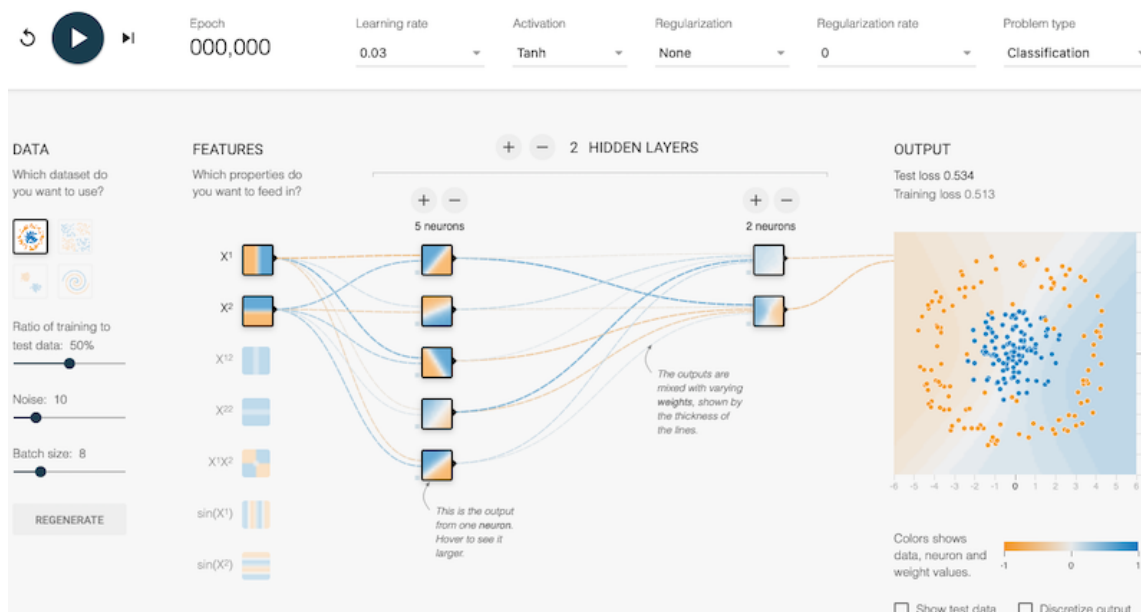
Tovrstne tehnike lahko v praksi prinesejo impresivne rezultate, na primer pri prepoznavanju glasu ali predmetov na sliki.

Vendar pa je razumevanje, zakaj (in kako) se dosegajo tako dobri rezultati, še vedno precej odprto znanstveno vprašanje.



POSKUSITE TENSORFLOW

C Klikni na spodnjo sliko, da v novem oknu odpreš aplikacijo TensorFlow



TensorFlow playground view