



Rețele neuronale artificiale

În învățarea mecanică și științele cognitive, o *rețea neuronală artificială* (RNA) este o rețea inspirată de rețelele neuronale biologice (sistemul nervos central al animalelor, în special creierul) care este folosită pentru a estima sau aproxima funcții care pot depinde de un număr mare de intrări, care sunt, în general, necunoscute. Rețelele neuronale artificiale sunt specificate în mod obișnuit, folosind trei lucruri:

- *Arhitectura* specifică variabilele implicate în rețea și relațiile lor topologice – de exemplu variabilele implicate într-o rețea neuronală ar putea fi greutatea conexiunilor dintre neuroni, împreună cu activitățile neuronilor.
- *Regula activității* – majoritatea modelelor de rețea neuronală au o dinamică la scară de timp scurt: reguli locale definesc modul în care activitățile neuronilor se schimbă ca răspuns de la unul la altul. De obicei regula activității depinde de mase (parametrii) din rețea.
- *Regula de învățare* specifică modul în care greutatea rețelei neuronale se schimbă în timp. Acest tip de învățare este de obicei văzut ca având loc pe o scară de timp mai lungă decât scara de timp a dinamicii sub regula activității. De obicei, regula de învățare va depinde de activitatea neuronilor. Aceasta poate depinde și de valorile valorilor-țintă furnizate de către un cadru didactic și valoarea curentă a greutăților.

De exemplu, o rețea neuronală pentru recunoașterea scrisului de mână este definită de un set de neuroni de intrare, care pot fi activați de către pixelii unei imagini de intrare. După ce au fost ponderate și transformate de o funcție (determinată de către proiectantul rețelei), activările acestor neuroni sunt apoi transmise altor neuroni. Acest proces se repetă până când, în final, neuronul de ieșire determină ce caracter citit este activat.

La fel ca și alte metode de învățare automată – sisteme care învață din date – rețelele neuronale au fost folosite pentru a rezolva o mare varietate de sarcini, cum ar fi viziunea pe calculator și recunoașterea vocală, care sunt greu de rezolvat cu ajutorul programării obișnuite, bazată pe reguli.

Origine

Examinări ale sistemelor nervoase centrale ale oamenilor au inspirat conceptul de rețele neuronale artificiale. Într-o rețea neuronală artificială, nodurile artificiale simple, cunoscute sub numele de "neuroni", "Neurodele", "elemente de procesare" sau "unități", sunt conectate împreună pentru a forma o rețea care imită o rețea neurală biologică.

Nu există nicio singură definiție formală a ceea ce este o rețea neuronală artificială. Cu toate acestea, o clasă de modele statistice poate fi numită în mod obișnuit "neurală" în cazul în care posedă următoarele caracteristici:

1. conține seturi de greutăți adaptive, adică parametrii numerici, care sunt reglate printr-un algoritm de învățare;



2. este capabilă de a aproxima funcții de bază non-liniare pe baza intrărilor lor.

Greutățile adaptive pot fi privite ca punctele forte de conectare dintre neuroni, care sunt activate în timpul antrenamentului și predicției.

Rețelele neuronale artificiale sunt similare cu rețelele neuronale biologice în efectuarea de către unitățile sale de funcții în mod colectiv și, în paralel, mai degrabă decât printr-o delimitare clară a subactivității cărora le sunt atribuite unități individuale. Termenul "rețele neuronale", de obicei, se referă la modelele utilizate în statistică, psihologia cognitivă și inteligență artificială. Modelele de rețea neuronală, care comandă sistemul nervos central și restul creierului fac parte din neuroștiința teoretică și neuroștiința computațională.

În implementări moderne de software de rețele neuronale artificiale, abordarea inspirată de biologie a fost în mare parte abandonată pentru o abordare mai practică bazată pe statistici și prelucrarea semnalului. În unele dintre aceste sisteme, rețele neuronale sau părți ale rețelelor neuronale (cum ar fi neuroni artificiali), formează componente în sisteme mai mari, care combină atât elemente adaptabile cât și neadaptabile. În timp ce abordarea mai generală a unor astfel de sisteme este mai potrivită pentru rezolvarea problemelor din lumea reală, avem puțin de a face cu modelele tradiționale, conexiunilor de inteligență artificială. Ceea ce au în comun este principiul non-liniar, distribuit, procesarea paralelă și locală și adaptarea. Din punct de vedere istoric, utilizarea modelelor de rețele neuronale a marcat o schimbare de direcție la sfârșitul anilor optzeci a inteligenței artificiale la nivel înalt (simbolic), caracterizată prin sisteme expert cu cunoștințe încorporate în reguli de *if-then*, învățare mecanică la nivel scăzut (sub-simbolic), caracterizată prin cunoașterea parametrilor unui sistem dinamic.

Istoric

Warren McCulloch și Walter Pitts (1943) au creat un model de calcul pentru rețele neuronale bazate pe matematică și algoritmi numiți logică de prag. Acest model a deschis calea pentru cercetarea rețelelor neuronale pentru a le împărți în două abordări distincte. O abordare axată pe procesele biologice în creier, iar cealaltă axată pe aplicarea rețelelor neuronale pentru inteligență artificială.

Învățarea Hebbiana

La sfârșitul anilor 1940, psihologul Donald Hebb a creat o ipoteză de învățare bazată pe mecanismul plasticității neuronale, care este acum cunoscut sub numele de învățare Hebbiană. Învățarea Hebbiană este considerată a fi o regulă de învățare nesupravegheată "tipică", iar variantele sale ulterioare au fost primele modele pentru potențarea pe termen lung. Cercetătorii au început să aplice aceste idei la modele de calcul, în 1948, cu mașinile Turing de tip B.

Farley și Wesley A. Clark (1954) au utilizat prima dată mașini de calcul, numite apoi "calculatoare", pentru a simula o rețea Hebbiană la MIT. Alte mașini de calcul de rețele neuronale au fost create de Rochester, Holland, Habit and Duda (1956).



Frank Rosenblatt (1958) a creat perceptronul, un algoritm de recunoaștere al modelului bazat pe o rețea de învățare pe calculator cu două straturi, folosind adunarea simplă și scăderea. Cu notație matematică, Rosenblatt a descris, de asemenea, circuitele nu în perceptronul de bază, cum ar fi circuitul *sau-exclusiv*, un circuit care nu putut fi procesat prin rețele neuronale până ce algoritmul de propagare nu a fost creat de Paul Werbos (1975).

Cercetarea rețelei neuronale stagnează după publicarea cercetării mașinii de învățare a lui Marvin Minsky și Seymour Papert (1969), care a descoperit două aspecte-cheie la mașinile de calcul care prelucrau rețele neuronale. Primul a fost că perceptronii de bază au fost incapabili să proceseze circuitul *sau-exclusiv*. Cea de-a doua problemă semnificativă a fost faptul că computerele nu au avut suficientă putere de procesare pentru a gestiona în mod eficient timpul de funcționare cerut de rețelele neuronale mari. Cercetarea rețelelor neuronale a încetinit până când calculatoarele au atins o putere de procesare mai mare.

Propagare și renaștere

Un avans cheie, care a venit mai târziu, a fost algoritmul de propagare, care a rezolvat în mod eficient problema de *sau-exclusiv* și, mai mult, problema formării rapide a rețelelor neuronale multi-strat (Werbos 1975).

La mijlocul anilor 1980, procesarea distribuită în paralel a devenit populară sub numele de conexiunism. Manualul lui David E. Rumelhart și James McClelland (1986) a furnizat o expunere completă a utilizării conexiunismului în computere, pentru a simula procesele neuronale.

Rețelele neuronale, așa cum sunt utilizate în domeniul inteligenței artificiale, au fost în mod tradițional privite ca modele simplificate de prelucrare neuronală în creier, chiar dacă relația dintre acest model și arhitectura biologică a creierului este dezbătută; nu este clar în ce măsură rețelele neuronale artificiale oglindesc funcția cerebrală.

Mașinile vectoriale de sprijin și alte metode, mult mai simple, cum ar fi clasificatorii liniari, au preluat treptat rețelele neuronale în popularitatea mașinii de învățare. Dar, apariția învățării aprofundate la sfârșitul anilor 2000 a stârnit un interes reînnoit în rețelele neuronale.

Îmbunătățirile din 2006

Dispozitivele de calcul au fost dezvoltate în tehnologie CMOS, atât pentru simularea biofizică cât și pentru calculul neuromorfic. Eforturile recente par promițătoare pentru crearea nano-dispozitivelor pentru analize ale componentelor principale la scară foarte mare și convoluție. Dacă ar avea succes, s-ar crea o nouă clasă de calcul neuronal, deoarece depinde de învățare, mai degrabă decât de programare și pentru că este în mod fundamental analogic, mai degrabă decât digital, chiar dacă primii instanțieri pot fi, de fapt, dispozitive digitale CMOS.

Între 2009 și 2012, rețelele neuronale recurente și rețele neuronale profunde dezvoltate în grupul de cercetare Jürgen Schmidhuber la Swiss AI Lab IDSIA au câștigat opt concursuri internaționale pentru recunoașterea modelului și învățare mecanică. De exemplu, memoria pe termen scurt bi-



directională și multi-dimensională pe termen scurt (LSTM) a lui Alex Graves și a colaboratorilor săi a câștigat trei concursuri de recunoaștere a scrierii de mână conectate la Conferința Internațională din 2009 privind Analiza Documentelor și Recunoaștere (ICDAR), fără nicio cunoaștere prealabilă cu privire la cele trei limbi diferite care urmează a fi învățate.

Implementări rapide bazate pe GPU ale acestei abordări de Dan Cireșan și colegii săi de la IDSIA au câștigat mai multe concursuri pentru recunoaștere modelului, inclusiv IJCNN 2011 Concursul Trafic Sign Recognition, ISBI 2012 Segmentarea structurilor neuronale în Electron Microscopy Stacks, și alții. Rețelele lor neuronale au fost, de asemenea, primele modele de recunoaștere artificială care au atins chiar și performanțe supraumane privind repere importante, cum ar fi recunoașterea semnelor de circulație (IJCNN 2012), sau problema cifrelor din scrisul de mână MNIST a lui Yann LeCun la NYU.

Arhitecturi profunde, extrem de neliniare, neuronale similare cu cele din 1980 neocognitron de Kunihiko Fukushima și "arhitectura standard a viziunii", inspirate de celule simple și complexe identificate de David H. Hubel și Torsten Wiesel în cortexul vizual primar, pot fi de asemenea pre-instruite prin metode nesupravegheate din laboratorul lui Geoff Hinton la Universitatea din Toronto. O echipă de la acest laborator a câștigat un concurs în 2012, sponsorizat de Merck pentru a proiecta software-ul de design în scopul ajutării la găsirea de molecule care ar putea duce la noi medicamente.

Modele

Modelele de rețea neuronală în inteligență artificială sunt de obicei denumite rețele neuronale artificiale (RNA); acestea sunt modele matematice simple, în esență, care definesc o funcție $f: X \rightarrow Y$ sau o distribuție peste X sau ambele X și Y , dar modelele sunt intim asociate cu un anumit algoritm de învățare sau regulă de învățare. O utilizare obișnuită a expresiei "modelul RNA" este într-adevăr definiția unei clase de astfel de funcții (în care membrii clasei sunt obținuți prin parametri diferiți, greutatea de conectare, sau specificul arhitecturii, cum ar fi numărul de neuroni sau conectivitatea).

Funcția de rețea

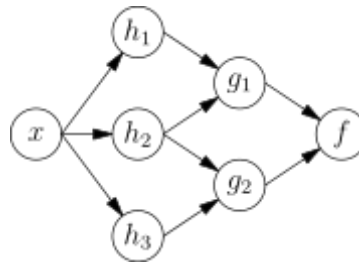
Cuvântul rețea în termenul "rețea neuronală artificială" se referă la interconexiunile dintre neuronii din diferitele straturi ale fiecărui sistem. Un exemplu de sistem are trei straturi. Primul strat are neuroni de intrare, care transmit date prin sinapse către al doilea strat de neuroni, iar apoi prin mai multe sinapse la al treilea strat de neuroni de ieșire. Sistemele mai complexe vor avea mai multe straturi de neuroni, unele având straturi de neuroni de intrare și de neuroni de ieșire crescute. Sinapsele stochează parametrii numiți "greutăți" care manipulează datele din calcule.

O RNA este definită de obicei de trei tipuri de parametri:

1. Modelul de interconectare între diferitele straturi de neuroni;
2. Procesul de învățare pentru actualizarea greutăților interconexiunilor;
3. Funcția de activare care convertește intrarea ponderată a unui neuron la activarea ieșirii sale.



Matematic, funcția rețelei de neuroni $f(x)$ este definită ca o compoziție de alte funcții $g_i(x)$, care poate fi definită mai detaliat ca o compoziție a altor funcții. Acest lucru poate fi în mod convenabil reprezentat ca o structură de rețea, cu săgeți, care descriu dependențele dintre variabile. Un tip utilizat pe scară largă de compoziție este suma ponderată neliniară, unde $f(x) = K(\sum_i w_i g_i(x))$, unde K (denumit în mod obișnuit funcția de activare) este o funcție predefinită, cum ar fi tangenta hiperbolică. Ar fi convenabil pentru următoarele să se facă referire la o colecție de funcții g_i ca un simplu vector $g = (g_1 g_2 \dots g_n)$.



Grafic de dependență RNA

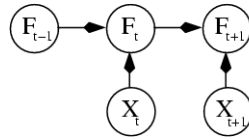
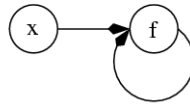
Această cifră reprezintă o astfel de descompunere a f , cu dependențe între variabilele indicate prin săgeți. Acestea pot fi interpretate în două moduri.

Primul punct de vedere este funcțional: intrarea x este transformată într-un vector tridimensional h , care este apoi transformat într-un vector bidimensional g , care este în final transformat în f . Acest punct de vedere este cel mai frecvent întâlnit în contextul optimizării.

Al doilea punct de vedere este probabilist: variabila aleatoare $F = f(G)$ depinde de variabila aleatoare $G = g(H)$, care depinde de $H = h(X)$, care depinde de variabila aleatoare X . Acest punct de vedere este cel mai frecvent întâlnit în contextul modelelor grafice.

Cele două puncte de vedere sunt în mare măsură echivalente. În ambele cazuri, pentru această arhitectură de rețea, componentele straturilor individuale sunt independente unele de altele (de exemplu, componentele g sunt independente una față de cealaltă dată de intrarea lui h). Acest lucru permite în mod natural un grad de paralelism în punerea în aplicare.

Rețele precum cea anterioară sunt numite în mod obișnuit *feedforward*, deoarece graficul lor este un grafic aciclic direcționat. Rețelele cu cicluri sunt numite de obicei recurente. Aceste rețele sunt, de obicei, reprezentate așa cum se indică în partea superioară a figurii, unde este prezentată ca fiind dependentă de ea însăși. Cu toate acestea, o dependență temporală implicită nu este afișată.



Învățare

Ceea ce a atras cel mai mare interes în rețelele neuronale este posibilitatea de a învăța. Având în vedere o sarcină specifică de rezolvat, și o clasă de funcții F , învățarea înseamnă utilizarea unui set de observații pentru a găsi $f^* \in F$ care rezolvă sarcina într-un anumit sens optim.

Acest lucru presupune definirea unei funcții de cost $C: F \rightarrow R$ astfel încât, pentru soluția optimă f^* , $C(f^*) \leq C(f), \forall f \in F$ – i.e., nici o soluție nu are un cost mai mic decât costul soluției optime.

Funcția de cost C este un concept important în procesul de învățare, deoarece este o măsură a cât de departe o anumită soluție este față de o soluție optimă, pentru problema care trebuie rezolvată. Algoritmii de învățare caută prin spațiul de soluție pentru a găsi o funcție care are cel mai mic cost posibil.

Pentru aplicații în care soluția este dependentă de anumite date, costul trebuie să fie în mod necesar în funcție de observații, altfel nu ar modela nimic legat de date. Aceasta este frecvent definită ca o statistică la care se pot face numai aproximări. Ca un exemplu simplu, luați în considerare problema găsirii modelului f , care minimizează $C = E[(f(x) - y)^2]$, pentru perechile de date (x, y) trase din aceeași distribuție D . În situații practice am avea numai N probe din D și, astfel, pentru exemplul de mai sus, vom minimiza numai $C = \frac{1}{N} \sum_{i=1}^N (f(x_i) - y_i)^2$. Astfel, costul este redus la minim, pe o mostră de date, mai degrabă decât întreaga distribuție care generează datele.

Când $N \rightarrow \infty$ o anumită formă de memorare automată on-line trebuie să fie utilizată, în cazul în care costul este redus la minimum parțial, deoarece fiecare nou exemplu este văzut. În timp ce memorarea automată on-line este adesea folosită atunci când D este fix, este mai utilă în cazul în care se schimbă încet distribuția în timp. În metodele de rețele neuronale, o anumită formă de memorare automată on-line este frecvent utilizată pentru seturi de date finite.

Deși este posibil să se definească o funcție arbitrară ad-hoc a costurilor, în mod frecvent un cost particular va fi utilizat, fie pentru că are proprietățile dorite (cum ar fi convexitate), fie pentru că ea apare în mod natural dintr-o anumită formulare a problemei (de exemplu, într-o formulare probabilistică de probabilitate posterioară a modelului poate fi folosită ca un cost invers). În cele din urmă, funcția de cost va depinde de sarcina dorită. O trecere în revistă a celor trei categorii principale de activități de învățare este prezentată mai jos:



Paradigme de învățare

Există trei paradigme majore de învățare, fiecare corespunzând unei anumite sarcini de învățare abstractă. Acestea sunt învățarea supravegheată, învățarea nesupravegheată și învățarea de consolidare.

În procesul de *învățare supravegheată*, ni se dă un set de exemple de perechi (x, y) , $x \in X, y \in Y$, iar scopul este acela de a găsi o funcție $f: X \rightarrow Y$ în clasa permisă a funcțiilor, care se potrivește cu exemplele. Cu alte cuvinte, dorim să deducem maparea implicită de date; funcția de cost este legată de nepotrivirea între cartografierea noastră și date și conține, implicit, cunoștințele anterioare despre domeniul problemei.

Un cost frecvent utilizat este eroarea medie-pătrat, care încearcă să minimizeze eroarea medie la pătrat între ieșirea rețelei, $f(x)$, și valoarea țintă y peste toate exemplele de perechi. Atunci când cineva încearcă să reducă la minimum acest cost, folosind gradientul de coborâre pentru clasa de rețele neuronale numite perceptroni multistrat (MLP), se obține bine cunoscutul algoritm de propagare pentru formarea rețelelor neuronale.

Sarcinile care se încadrează în paradigma învățării supravegheate sunt recunoașterea tiparului (de asemenea, cunoscută sub numele de clasificare) și regresie (de asemenea, cunoscută sub numele de aproximare funcție). Învățarea supravegheată se aplică datelor secvențiale (de exemplu, pentru vorbire și recunoașterea gesturilor). Acest lucru poate fi considerat ca învățare cu un "profesor", sub forma unei funcții care oferă un feedback continuu asupra calității soluțiilor obținute până în prezent.

În procesul de *învățare nesupravegheată*, unele date x sunt furnizate și funcția de cost trebuie să fie redusă la minimum, putând fi orice funcție a datelor x și ieșirea rețelei neuronale f .

Funcția de cost depinde de sarcină (ceea ce încercăm să modelăm) și de ipoteza apriori (proprietățile implicite ale modelului nostru, parametrii săi și variabilele observate).

Ca un exemplu banal, luați în considerare modelul $f(x) = a$ unde a este o constantă și costul $C = E[(x - f(x))^2]$. Minimizând acest cost obținem o valoare de a care este egală cu media datelor. Funcția de cost poate fi mult mai complicată. Forma sa depinde de aplicare: de exemplu, prin compresie ar putea fi legată de informația reciprocă între x și $f(x)$, în timp ce în modelarea statistică, ar putea avea legătură cu probabilitatea posterioară a modelului de date (notați că în ambele exemple aceste cantități ar fi maximizate mai degrabă decât reduse la minimum).

Sarcinile care se încadrează în paradigma învățării nesupravegheate sunt în general probleme de estimare; aplicațiile includ gruparea, estimarea distribuțiilor statistice, compresie și filtrare.

La *învățarea de consolidare*, datele x de obicei nu sunt date, dar sunt generate de interacțiunile agentului cu mediul. În fiecare punct în timp t , agentul efectuează o acțiune y_t și mediul generează o observație x_t și un cost instantaneu c_t , potrivit unor dinamici (de obicei, necunoscute). Scopul este de a descoperi o politică de selectare a acțiunilor care minimizează unele măsuri de costuri pe



termen lung, de exemplu, costul cumulat așteptat. Dinamica mediului ambiant și costurile pe termen lung pentru fiecare politică sunt, de obicei necunoscute, dar pot fi estimate.

Mai mult formal, mediul este modelat ca un proces de decizie Markov (MDP) cu stările $s_1, s_2, \dots, s_n \in S$ și acțiunile $a_1, a_2, \dots, a_m \in A$ cu următoarele distribuții de probabilitate: distribuția instantanee a costurilor $P(c_t | s_t)$, distribuția observației $P(x_t | s_t)$ și tranziția $P(s_{t+1} | s_t, a_t)$, în timp ce o politică este definită ca distribuția condiționată asupra acțiunilor având în vedere observațiile. Luată împreună, cele două definesc apoi un lanț Markov (MC). Scopul este de a descoperi politica (adică, MC) care minimizează costurile.

RNA-urile sunt frecvent utilizate în procesul de învățare de consolidare, ca parte a algoritmului de ansamblu. Programarea dinamică a fost cuplată cu RNA (dând programare neurodinamică) prin Bertsekas și Tsitsiklis și aplicate problemelor multidimensionale neliniare, cum ar fi cele implicate în dirijarea vehiculului, gestionarea resurselor naturale sau medicină, din cauza capacității RNA de a atenua pierderile de precizie chiar și atunci când are loc reducerea densității rețelei de discretizare pentru aproximarea numerică a soluției problemelor de control originale.

Sarcinile care se încadrează în paradigma învățării de consolidare sunt probleme de control, jocuri și alte sarcini de luare a deciziilor secvențiale.

Algoritmi de învățare

Formarea unui model de rețele neuronale înseamnă, în esență, selectarea unui model din setul de modele permise (sau, într-un cadru Bayesian, determinând o distribuție peste setul de modele permise), care minimizează criteriul de cost. Există numeroși algoritmi disponibili pentru modelele de rețele neuronale de formare; cei mai mulți dintre aceștia pot fi văzuți ca o aplicație simplă a teoriei optimizării și estimare a statisticii.

Majoritatea algoritmilor utilizați în formarea de rețele neuronale artificiale folosesc o anumită formă de gradient descendent, folosind propagarea pentru a calcula gradientii reali. Acest lucru se face prin simpla luare a derivatei funcției de cost în raport cu parametrii rețelei și apoi schimbarea acestor parametri într-o direcție legată de pantă. Algoritmii de formare a propagării sunt de obicei clasificați în trei categorii: cea mai abruptă coborâre (cu rată de învățare variabilă, cu rată variabilă de învățare și de impuls, propagare elastică), cvasi-Newton (Broyden-Fletcher-Goldfarb-Shanno, secant cu un pas), Levenberg-Marquardt și gradient conjugat (update Fletcher-Reeves, update Polak Ribiere, repornire Powell-Beale, gradient conjugat scalat).

Metodele evolutive, programarea expresiei genelor, alipirea simulată, maximizarea așteptării, metodele non-parametrice și optimizarea grupului de particule sunt câteva alte metode pentru rețele neuronale de formare.



Folosirea rețelelor neuronale artificiale

Cel mai mare avantaj al RNA este capacitatea de a putea fi folosite ca un mecanism de aproximare a funcției arbitrare care "învăță" din datele observate. Cu toate acestea, utilizarea lor nu este atât de simplă, și o relativ bună înțelegere a teoriei de bază este esențială.

- Alegerea modelului: aceasta va depinde de reprezentarea datelor și aplicație; modelele excesiv de complexe tind să conducă la provocări în procesul de învățare.
- Algoritmii de învățare: există numeroase compromisuri între algoritmii de învățare. Aproape orice algoritm va lucra bine cu hiper-parametri corecți pentru formarea unui anumit set de date fixe. Cu toate acestea, selectarea și reglarea unui algoritm de formare pe date nevăzute necesită o cantitate semnificativă de experimentare.
- Robustețe: în cazul în care modelul, funcția de cost și algoritmul de învățare sunt selectate în mod corespunzător, RNA rezultată poate fi extrem de robustă.

Odată cu punerea corectă în aplicare, RNA poate fi utilizată în mod natural în procesul de învățare on-line și în aplicații de seturi mari de date. Implementarea lor simplă și existența unor dependențe preponderent locale, expuse în structură permit implementări rapide și paralele în hardware.

Aplicații

Utilitatea modelelor de rețele neuronale artificiale constă în faptul că acestea pot fi utilizate pentru a deduce o funcție din observații. Acest lucru este util mai ales în aplicațiile în care complexitatea datelor sau sarcina face proiectarea unei astfel de funcții de mână impracticabilă.

Aplicații în viața reală

Sarcinile cărora le sunt aplicate rețelele neuronale artificiale tind să se încadreze în următoarele categorii generale:

- aproximarea funcției sau analiza de regresie, inclusiv predicția seriei de timp, aproximarea capacității și modelare;
- Clasificarea, inclusiv recunoașterea modelului și secvenței, detecție noutate și luarea deciziilor secvențiale;
- Prelucrarea datelor, inclusiv filtrarea, gruparea, separarea sursei oarbe și compresie;
- Robotică, inclusiv manipulare directe, proteză;
- Control, inclusiv controlul numeric de calculator.

Domeniile de aplicare includ identificarea și controlul sistemului (controlul vehiculului, predicția traiectoriei, controlul procesului, gestionarea resurselor naturale), chimia cuantică, jocul și luarea deciziilor (table, șah, poker), recunoașterea modelului (sisteme de radar, de identificare a feței, recunoaștere obiect și multe altele), recunoaștere secvență (gest, vorbire, recunoaștere a textului scris de mână), diagnostic medical, aplicații financiare (de exemplu, sisteme automate de



tranzacționare), mimare date (sau descoperirea de cunoștințe în baze de date, "KDD"), vizualizare și filtrare a spam-ului în e-mail.

Rețelele neuronale artificiale au fost folosite pentru a diagnostica mai multe tipuri de cancer. Un sistem hibrid de detectare a cancerului pulmonar RNA pe nume HLND îmbunătățește precizia diagnosticului și viteza radiologiei la cancerul pulmonar. Aceste rețele au fost de asemenea utilizate pentru diagnosticarea cancerului de prostată. Diagnosticul poate fi folosit pentru a face modele specifice luate dintr-un grup mare de pacienți, comparativ cu informații de la un pacient dat. Modelele nu depind de ipoteze cu privire la corelarea diferitelor variabile. Cancerul colorectal a fost, de asemenea, prezis cu ajutorul rețelelor neuronale. Rețelele neuronale ar putea prezice rezultatul pentru un pacient cu cancer colorectal, cu mai multă precizie decât metodele clinice curente. După formare, rețelele ar putea prezice mai multe rezultate pentru pacient din instituții care nu au legătură unele cu altele.

Retele neuronale si Neuroștiință

Neuroștiința teoretică și de calcul este domeniul care se ocupă cu analiza teoretică și modelarea computațională a sistemelor neuronale biologice. Din moment ce sistemele neuronale sunt strâns legate de procesele cognitive și de comportament, domeniul este strâns legat de modelarea cognitivă și comportamentală.

Scopul domeniului este de a crea modele de sisteme neuronale biologice, pentru a înțelege modul în care funcționează sistemele biologice. Pentru a obține această înțelegere, neurologii depun eforturi pentru a face o legătură între procesele biologice observate (date), mecanisme plauzibile din punct de vedere biologic pentru prelucrarea neuronală și învățare (modele de rețele neuronale biologice) și teorie (teoria învățării statistice și teoria informației).

Tipuri de modele

Multe modele sunt utilizate în domeniu, definite la diferite niveluri de abstractizare și modelează diferite aspecte ale sistemelor neuronale. Acestea variază de la modelele comportamentului pe termen scurt al neuronilor individuali, modele referitor la modul în care apare dinamica circuitelor neuronale din interacțiuni între neuroni individuali și, în final, la modele despre modul în care comportamentul poate apărea din module neuronale abstracte care reprezintă subsisteme complete. Acestea includ modelele pe termen lung, și plasticitatea pe termen scurt, a sistemelor neuronale și relațiile lor de învățare și de memorie din neuronul individual la nivel de sistem.

Retele de memorie

Integrarea componentelor externe de memorie cu rețele neuronale artificiale are o lungă istorie care datează din primele cercetări în reprezentările distribuite și hărți cu auto-organizare. De exemplu, în memoria distribuită tiparele codificate de rețele neuronale sunt folosite ca adrese de memorie pentru memorie cu conținut adresabil, cu "neuroni", în esență, servind drept codoare de adrese și decodoare.



Mai recent, învățarea profundă a fost dovedită a fi utilă în complicarea semantică, în cazul în care un model grafic profund al vectorilor cuvânt – numărare se obține dintr-un set mare de documente. Documentele sunt mapate la adresele de memorie în așa fel încât documentele similare sunt situate semantic la adresele din apropiere. Documente similare unui document de interogare pot fi găsite prin simpla accesare a tuturor adreselor care diferă cu doar câțiva biți față de adresa documentului de interogare.

Mașinile Neuronale Turing dezvoltate de Google DeepMind extind capacitățile rețelelor neuronale profunde prin cuplarea lor la memoria de resurse externe, cu care pot interacționa prin procese atenționale. Sistemul combinat este analog cu o mașină Turing, dar este diferențibil end-to-end, permițându-i să fie antrenat în mod eficient cu gradient descendent. Rezultatele preliminare demonstrează că Echipamentele Neuronale Turing pot deduce algoritmi simpli, cum ar fi copierea, sortarea și rechemarea asociativă din exemplele de intrare și de ieșire.

Rețele de memorie sunt o altă extensie a rețelelor neuronale care încorporează memoria pe termen lung, care a fost dezvoltată de cercetarea Facebook. Memoria pe termen lung poate fi citită și scrisă, cu scopul de a fi utilizată pentru predicție. Aceste modele au fost aplicate în contextul răspunsului la întrebare, în care memoria pe termen lung acționează în mod eficient ca o bază de cunoștințe (dinamic), iar rezultatul este un răspuns textual.

Bibliografie:

- https://en.wikipedia.org/wiki/Artificial_neural_network
- https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html
- <http://natureofcode.com/book/chapter-10-neural-networks/>
- <http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html>
- <http://www.journals.elsevier.com/neural-networks>