

Doamne ajutaa. Numa bine

Care este motivul pentru care utilizăm activări neliniare în rețele neurale artificiale?

a. Activările trebuie limitate într-un domeniu ușor de gestionat (e.g. $(-1,1)$ sau $(0,1)$) pentru a evita fenomenele de *vanishing/exploding gradients*.

b. Spațiile descrise de datele de intrare sunt întotdeauna neliniare.

c.

leșirile rețelei trebuie să fie neliniare pentru a putea fi utilizate în calculul funcției de eroare.

d.

Dacă s-ar utiliza doar activări liniare, prin compoziționalitate rețeaua ar echivala cu o funcție liniară. așa cred

Care afirmație este adevărată în cazul utilizării de straturi de tip *Batchnorm* în cadrul arhitecturii unei rețele adânci.

a. Valorile activărilor de pe straturi se stabilizează prin normalizarea acestora (per-canal), accelerând astfel procesul de antrenare. Asta? (aparent da, după ce au aparut rezultatele :)))

b. Valorile gradientilor nu mai fluctuează de la un mini-batch la altul stabilizând astfel antrenarea.

c. Ponderile de pe straturi nu mai cresc la valori foarte mari ele fiind stabilizate de straturile *Batchnorm*.

d. Valorile ponderilor neimportante scad pe măsură ce se realizează mai mulți pași de învățare.

Intr-o rețea SOM, care răspunsuri sunt adevărate pentru U-matrix:

a. Reda distanța între neuroni adiacenți care este calculată și redată sub forma de diferite culori

b. O culoare închisă între neuroni corespunde unui gap între ponderi și spațiul de intrare

c. Toate răspunsurile sunt corecte (așa zic și eu) +1 +1

d. Permite vizualizarea distanței dintre neuroni

Care afirmație este adevărată despre regularizarea L2 a rețelelor adânci.

a.

Activările sunt normalizate la nivel de mini-batch evitând astfel valori mari care să influențeze puternic decizia și reducând overfit-ul.

b.

Fenomenul de overfit este estompat în timpul învățării deoarece (în medie) valorile ponderilor neimportante sunt micșorate, reducându-se astfel contribuția zgomotoasă a acestora la activările rețelei.

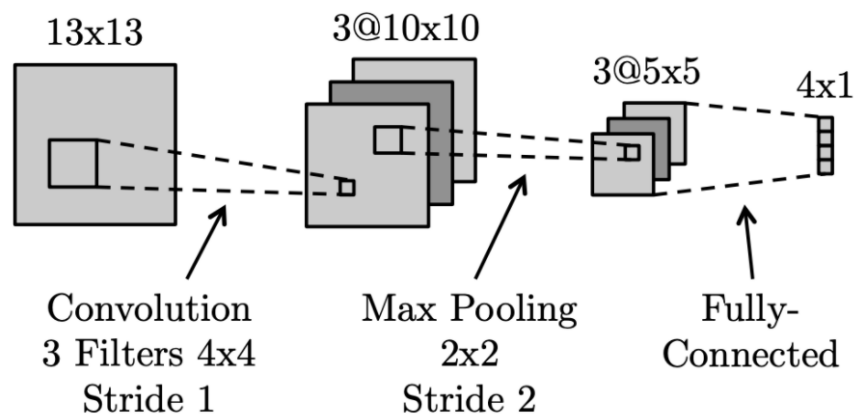
c.

Ajută la aducerea valorilor pentru ponderile rețelei mai aproape de setului unic de valori care rezolvă optim task-ul (minimizează funcția de loss).

d.

Împiedică apariția optimelor locale ducând astfel la o generalizare mai bună a rețelei.

Fie următoarea rețea adanca cu straturile: convolutional, max pooling, fully connected.



Numărul de ponderi asociate stratului convolutional este:

a.

30

b.

169

c.

48

Care este ieșirea unui neuron care are ponderile $[3, -1, 2]$ și bias 2 pentru intrarea $[7, 9, 10]$?

a.

30

b.

34

c.

37

d.

32

Care dintre următorii indivizi nu reprezintă un posibil copil obținut printr-o operație de recombinare multi-punct între părinții de mai jos?

părinte 1: 0 0 1 1 0 1 0 1 1 1

părinte 2: 1 1 0 1 0 0 1 0 0 0

a.

0 0 1 1 0 0 0 1 0 0

b.

0 0 0 1 0 1 1 0 0 0

c.

1 0 1 1 0 1 1 0 0 0

d.

1 1 0 0 0 0 1 1 1 +1 +1

Cum se poate preveni problema *vanishing gradients* în rețelele neurale adânci?

a. Prin introducerea de conexiuni reziduale între straturi

b. Prin adăugarea mai multor straturi convoluționale în rețea

c. Prin creșterea numărului de parametri în straturi

Alegeți afirmația greșită:

a. Ieșirea unei funcții treaptă pentru intrarea $[-1, 2, 0, 4, -3]$ este $[-1, 1, 1, 1, -1]$

b. Ieșirea unei funcții sigmoid pentru intrarea $[-1, 2, 0, 4, -3]$ este $[0.26, 0.88, 0.5, 0.98, 0.04]$

c. Ieșirea unei funcții ReLU pentru intrarea $[-1, 2, 0, 4, -3]$ este $[0, 2, 0, 4, 0]$

d. Ieșirea unei funcții tanh pentru intrarea $[-1, 2, 0, 4, -3]$ este $[0.83, 0.16, 0, 0.22, -0.45]$ (+1)

Alegeți afirmația corectă:

a.

Probabilitatea operației de mutație crește odată cu lungimea indivizilor.

b.

Operația de mutație adaugă gene noi la genele existente pe care le are un individ.

c.

Operația de mutație schimbă întotdeauna valorile genelor unui individ.

d.

Operația de mutație se aplică asupra descendenților.

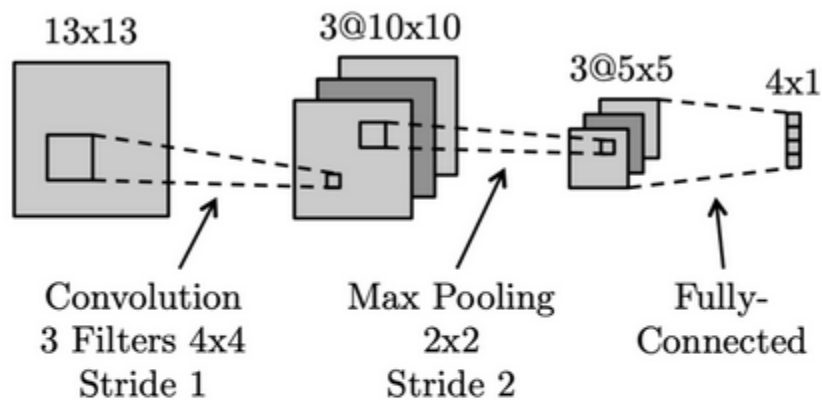
Intr-o rețea perceptron, putem renunța la bias în calculul activării dacă îl considerăm ca pondere conectată la o falsă intrare cu valoare constantă?

Select one:

True

False

Fie următoarea rețea adanca cu straturile: convolutional, max pooling, fully connected.



Numărul de ponderi asociate stratului convolutional este:

- ☐ a. 30
- ☐ b. 169
- ☐ c. 48

$$4 * 4 * 3 = 48$$

Câți parametri trebuie să învețe un layer de Average Polling cu dimensiunea kernel-ului de 2 și pas de 2?

Select one:

a.

2

b.

4

c.

3

d. 0 +1 ca ala nu invata

Straturile de tip Average Pool sunt utilizate în arhitecturile de rețele neuronale artificiale adânci cu scopul de a

Select one:

- ☐ a. Reduce zgomotul din rețea prin efectuarea mediei activărilor
- ☒ b. Reduce dimensiunea activărilor de la ieșirea unui strat înainte de a fi trimis stratului următor
- ☐ c. Îmbunătățește calitatea informației care este propagată prin rețea.
- ☐ d. Întărește corelațiile care se întâmplă la nivel local.

Fie un strat de convoluție cu intrarea de dimensiune (7x7), filtru de dimensiune (3x3), stride = 1 (se ignora a treia dimensiune, nu exista padding). Specificati dimensiunea ieșirii stratului convolutional.

Select one:

a.

3x3

b.

7x7

c.

5x5

Ce s-ar întâmpla într-o rețea neuronală artificială dacă am utiliza funcția de activare $y = 2x$, unde x este intrarea de pe un strat și y este ieșirea acestuia?

Selectați răspunsul corect:

- a. Ponderile s-ar micșora considerabil.
- b. Ar fi imposibil ca rețeaua să mai învețe să rezolve o problemă de clasificare sau regresie
- c. Valorile de la output ar deveni foarte mari
- d. Întreaga rețea s-ar putea reduce la o combinație liniară a intrărilor.

O rată de învățare foarte mare nu avantajează antrenarea unei rețele neuronale artificiale deoarece:

Select one:

a.

Reducem prea mult numărul de iterații necesari antrenării rețelei.

b.

Ieșirile de pe fiecare strat vor avea valori foarte mari.

c.

Derivatele în raport cu eroarea vor avea valori foarte mici.

d.

La fiecare pas de optimizare ponderile vor fi modificate prea mult corespunzător cu batch-ul curent de exemple de învățare (în raport cu restul setului de date). ++

Straturile de tip Average Pool sunt utilizate în arhitecturile de rețele neuronale artificiale adânci cu scopul de a

Select one:

a.

Reduce zgomotul din rețea prin efectuarea mediei activărilor

b.

Îmbunătățește calitatea informației care este propagată prin rețea.

c.

Întărește corelațiile care se întâmplă la nivel local.

d.

Reduce dimensiunea activărilor de la ieșirea unui strat înainte de a fi trimis stratului următor

Partea 2

Wtf coaie? PULA

Explicați care este legătura între utilizarea legăturilor reziduale în rețele neurale adânci și tehnicile de boosting.

Legaturile reziduale iau informată din urma, prin intermediul skip-connections. Astfel, antrenarea de la blocul k este adusă și combinată cu antrenarea de la blocul $k + 1$. Într-un mod similar funcționează și boostingul, în care modelul i continuă antrenarea din punctul în care "ramăsese" modelul $i - 1$.

<https://stats.stackexchange.com/questions/214273/are-residual-networks-related-to-gradient-boosting>

Explicați legătura dintre presiunea de selecție din algoritmi genetici și balanța *explorare* vs. *exploatare* a acestora. Dați exemple de 2 mecanisme de selecție și felul în care acestea o influențează.

Presiunea de selecție înseamnă probabilitatea de selecție a celui mai bun individ comparată cu probabilitatea medie de selecție a tuturor indivizilor. Astfel, când presiunea este mare, probabilitatea de selecție a celui mai bun individ este mare, deci este vorba de exploatare. În caz contrar, toți indivizii au probabilități de selecție apropiate, deci se preferă explorarea.

Două mecanisme de selecție: selecția pe baza de ruleta, esantionare universală stohastică

Care este ieșirea unui strat softmax care primește la intrare vectorul de mai jos?

4 8 1 2 5

$$P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

$e^4 / (e^4 + e^8 + e^1 + e^2 + e^5)$. și la fel și pentru restul nu ?

P1 = 1.72%? :)))

Etc...

$\text{softmax}_j = e^{z_j} / \sum(e^{z_k})$ cu $j = 1, k$

$\text{softmax}_1 = e^4 / (e^4 + e^8 + e^1 + e^2 + e^5) = 0.017093562956559$

$\text{softmax}_2 = e^8 / (e^4 + e^8 + e^1 + e^2 + e^5) = 0.93327691490319$

$\text{softmax}_3 = e^1 / (e^4 + e^8 + e^1 + e^2 + e^5) = 8.5103838756858E-4$

$\text{softmax}_4 = e^2 / (e^4 + e^8 + e^1 + e^2 + e^5) = 0.0023133621842487$

$\text{softmax}_5 = e^5 / (e^4 + e^8 + e^1 + e^2 + e^5) = 0.046465121568434$

Verificati si voi

N experti trebuie sa decida daca un mesaj de email este SPAM (1) sau EMAIL (0). Obiectivul este să se determine o singură predicție per mesaj. In acest scop, doriți să implementați un mecanism de vot majoritar, adică dacă mai mult de jumătate dintre experți prezic SPAM, predicția finală trebuie să fie SPAM pentru acea instanță, in caz contrar, predicția finală trebuie să fie EMAIL.

(1) Construiti o retea neurala care implementeaza un astfel de vot majoritar pentru cazul a 4 experti, A, B, C si D. Indicati structura retelei, intari, iesiri, ponderi, valori de prag si functia de activare.

(2) Explicati pe scurt cum se poate generaliza retea la N experti

(3) Care model dintre o retea neurala si un arbore de decizie este mai potrivit pentru rezolvarea acestei probleme in cazul general in care N este mare ? Justificati raspunsul.

(4) Sa presupunem ca fiecare expert individual foloseste informatii complet diferite si fiecare expert are acelasi nivel de acuratete (85%) in clasificare pentru un set de exemple dat. Se poate imbunatati acuratetea in acest caz pentru setul de exemple folosind un mecanism de vot majoritar ? Justificati raspunsul.

Fie un strat de convoluție care are cele două filtre de mai jos:

1	2	-1	4	0	7
0	9	4	0	5	0
5	8	2	3	0	6

Care este ieșirea stratului de convoluție având ca intrare imaginea de mai jos, cu stride 2 și padding 1? Explicați pașii urmați pentru a obține răspunsul.

2	4	1	3	-2	5	9
1	1	1	3	2	1	7
6	0	5	-1	1	-1	4
3	1	1	2	7	8	6
2	5	9	9	8	4	6
0	0	0	0	0	0	0
-1	1	2	5	9	2	4

E ceva online care calculeaza asta? Nu cred ca vor ceva direct

Revin cu rezultat :D Hai ca e nesimtita Foarte

Mirel

Avand padding 1, intrarea va fi bordata pe fiecare latura cu o coloana/ un rand plin de elemente egale cu 0.

Avand stride 2, fiecare filtru va avansa in input din 2 in 2 pozitii (atat pe verticala, cat si pe orizontala)

Pentru fiecare filtru, folosim urmatoarele formule

$$H_{out} = (H_{in} + 2 * padding - dilatation * (kernel_size - 1) - 1) / stride + 1$$

Analog pentru Wout

Astfel, imaginile rezultate vor avea fiecare dimensiunea 4x4. Astfel, avand 2 canale (2 filtre), vom avea ca iesire 4x4x2 (H x W x C).

Canalul 1:

- primul element: $1*0 + 2*0 + (-1)*0 + 0*0 + 9*2 + 4*4 + 5*0 + 8*1 + 2*1 = 0 + 0 + 0 + 0 + 18 + 16 + 0 + 8 + 2 = 44$

- al doilea element: $1*0 + 2*0 + (-1)*0 + 0*4 + 9*1 + 4*3 + 5*1 + 8*1 + 2*3 = 0 + 0 + 0 + 0 + 9 + 12 + 5 + 8 + 6 = 40$

Canalul 2:

- primul element: $4*0 + 0*0 + 7*0 + 0*0 + 5*2 + 0*4 + 3*0 + 0*1 + 6*1 = 0 + 0 + 0 + 0 + 10 + 0 + 0 + 0 + 6 = 16$

- al doilea element: $4*0 + 0*0 + 7*0 + 0*4 + 5*1 + 0*3 + 3*1 + 0*1 + 6*3 = 0 + 0 + 0 + 0 + 5 + 0 + 3 + 0 + 18 = 26$

Explicati rolul regularizarii in retelele adanci. Specificati asupra caror parametri se aplica regularizarea. (SCRIETI SI VOI ALTFEL, ATAT M-A DUS PE MINE CAPUL)

Regularizarea are rolul de a constrange influenta ponderilor cu valori foarte mari asupra iesirii.

Regularizarea se aplica asupra ponderilor // e bine(reformulati)

Intr-un algoritm genetic se considera urmatoorii doi parinti:

p1: 128 255 50 17 2 si

p2: 123 4 344 280 12.

Construiti cei doi copii obtinuti prin recombinare reala discreta avand

sample1: 2 2 1 1 2 si

sample 2: 1 2 1 1 2.

c1: 123 4 50 17 12

c2: 128 4 50 17 12

Check this. E bun yap

Explicați de ce rețelele convoluționale sunt invariante la translație dar nu și la rotație sau scalare. Cum putem ameliora aceste neajunsuri?

Invarianta la trnaslatie presupune ca indiferent de pozitia unui obiect intr-o imagine (presupunand o problema de computer vision), acesta va fi tratat (de exemplu detectat) la fel de o retea convolutionala. Totusi, daca am roti obiectul sau i-am modifica dimensiunea, rezultatele retelei s-ar schimba din cauza ca filtrele care se aplica unei retele gasesc patternuri intre pixeli apropiati, indiferent de pozitia acestora in imagine, dar nu si de orientarea lor. De exemplu, un filtru care detecteaza linii drepte, precum cel de jos:

0 10

0 10

nu ar detecta nimic daca am intoarce o imagine cu linii drepte la 90 de grade. Cat despre invarianta la scalare, o imagine mai mica, de exemplu, ar avea un numar mai mic de pixeli si, implicit, mai putine date pentru retea.