

Rețele neuronale și deep learning

Cursul 4: Rețele neuronale convoluționale (CNN) și transferul învățării

Prof. dr. Daniel Traian PELE

Academia de Studii Economice din București (ASE), Departamentul de Statistică și Econometrie
Academia Română, Institutul de Prognoză Economică (IPE)
IDA Institute Digital Assets ▪ AI4EFin ▪ MSCA Digital Finance

Obiectivele cursului

- Să scrieți un strat convoluțional ca formulă: nucleu, completarea marginilor, pas, canale, număr de parametri și cîmp receptiv.
- Să explicați de ce o rețea neuronală convoluțională (CNN) are nevoie de mult mai puțini parametri decît un perceptron multistrat (MLP) pe imagini și să verificați acest lucru pe date reale.
- Să citiți arhitecturile clasice, de la LeNet la ResNet, și conexiunea rezidual care face antrenabile rețelele adînci.
- Să folosiți transferul învățării (extragerea trăsăturilor și ajustarea fină) atunci cînd imaginile etichetate sînt scumpe și să măsurați cît economisește.
- Să explicați o predicție cu o hartă de activare a clasei (CAM) ponderată cu gradienți și să transformați predicțiile pe imagini în cifre utile în afaceri.

Abrevieri

AI inteligență artificială
ASE Academia de Studii Economice din București
BN normalizare pe loturi (batch normalisation)
CAM harta de activare a clasei (class activation map)
CIFAR Canadian Institute for Advanced Research (setul de imagini CIFAR-10)
CNN rețea neuronală convoluțională
CPU procesor central
GAP agregare prin medie globală (global average pooling)
GPT transformer generativ preantrenat
GPU procesor grafic
ICBE International Conference on Business Excellence
IPE Institutul de Prognoză Economică, Academia Română
MLP perceptron multistrat
MNIST Modified National Institute of Standards and Technology (imagini cu cifre)
MPS Metal Performance Shaders (procesorul grafic Apple, în PyTorch)
ReLU unitate liniară rectificată
RGB roșu, verde, albastru (canalele de culoare)
SGD coborîre stohastică pe gradient
UE Uniunea Europeană
VGG Visual Geometry Group, Universitatea Oxford
VLM model viziune–limbaj

Datele: trei colecții reale de imagini

- Cifrele Modified National Institute of Standards and Technology (MNIST) sînt înlocuite în Fashion-MNIST cu fotografii de îmbrăcăminte; Canadian Institute for Advanced Research (CIFAR) a publicat CIFAR-10.

Setul de date	Conținut	Împărțire (antr./valid./test)	Utilizare în afaceri
Fashion-MNIST	fotografii din catalogul Zalando, 10 tipuri de produse, 28×28, gri	50 000 / 10 000 / 10 000	etichetarea automată a produselor
CIFAR-10	fotografii, 10 clase de obiecte, 32×32, color	45 000 / 5 000 / 10 000	banc de probă pentru arhitecturi
EuroSAT	imagini satelitare Sentinel-2 din Europa, 10 clase de utilizare a terenului, 64×64, color	16 200 / 5 400 / 5 400	agricultură, asigurări, imobiliare

- Surse: Zalando Research (Xiao et al., 2017); Krizhevsky (2009); Helber et al. (2019), date Copernicus Sentinel-2; toate sînt descărcate de scripturile cursului (data/SOURCES.md).
- Seturile de validare servesc la oprirea timpurie și la reglaj; fiecare set de test este folosit o singură dată, la final.
- Ponderi preantrenate: ResNet-18 antrenat pe ImageNet (torchvision); antrenare pe procesorul grafic (GPU) Apple prin Metal Performance Shaders (MPS), sămînța 20261108.

(data/SOURCES.md)

Cuprins

- Imaginile ca date și motivarea convoluțiilor
- Stratul convoluțional
- Studiu de caz: clasificarea fotografiilor de produse cu regresie softmax și CNN
- Arhitecturi: de la LeNet la ResNet
- Transferul învățării: utilizarea terenului din imagini satelitare

O fotografie de produs este o matrice de numere

- **Pe scurt:** fiecare imagine este o matrice de 28×28 niveluri de gri; modelul trebuie să o asocieze unuia dintre cele 10 tipuri de produse.

$$\mathbf{X} \in [0,1]^{28\times 28}, \qquad \hat{\mathbf{p}} = f_{\theta}(\mathbf{X}) \in \Delta^{9}, \qquad \hat{y} = \arg\max_{k} \hat{p}_k$$



Sursa: Fashion-MNIST, Zalando Research (Xiao et al., 2017). Date: 60.000 de imagini de antrenare, 28 x 28 niveluri de gri.

- Fashion-MNIST: 50 000 de imagini de antrenare, 10 000 de validare și 10 000 de test, din catalogul Zalando.

- **În practică:** un magazin online etichetează zilnic mii de fotografii noi de produse
  - Clasificatorul stabilește categoria, iar un operator verifică doar cazurile îndoielnice.

(Xiao et al., 2017)

NN_L4_fmnist_samples

Un strat dens pe pixeli irosește parametri și ignoră spațiul

- **Pe scurt:** un MLP tratează pixelii (1,1) și (1,2) ca intrări fără legătură și are nevoie de cîte o pondere pentru fiecare pixel și fiecare unitate.

$$\mathbf{h} = \sigma(\mathbf{W}\,\mathrm{vec}(\mathbf{X}) + \mathbf{b}), \quad \mathbf{W} \in \mathbb{R}^{m \times HWC}$$

- **unde**
  - H, W înălțimea și lățimea
  - C numărul de canale
  - m unitățile ascunse
  - O imagine color de 224×224 și $m = 1000$ dau $224 \cdot 224 \cdot 3 \cdot 1000 \approx 1{,}5 \cdot 10^8$ ponderi numai în primul strat

- **Cele trei idei ale unei CNN**
  - Conectivitate locală: o unitate vede doar o mică zonă
  - Ponderi partajate: același filtru în toate pozițiile
  - Echivarianță: o deplasare a intrării deplasează harta de trăsături, $f(T_\delta \mathbf{X}) = T_\delta f(\mathbf{X})$
- Pe datele noastre: MLP cu 235 146 de parametri, CNN cu 140 778, de 1,67 ori mai puțini.

- **În practică:** mai puțini parametri înseamnă mai puține date necesare pentru aceeași acuratețe, iar imaginile etichetate costă bani.

(LeCun et al., 1998; Bishop & Bishop, 2024, cap. 10)

Cuprins

- Imaginile ca date și motivarea convoluțiilor
- Stratul convoluțional
- Studiu de caz: clasificarea fotografiilor de produse cu regresie softmax și CNN
- Arhitecturi: de la LeNet la ResNet
- Transferul învățării: utilizarea terenului din imagini satelitare

Convoluția: același filtru mic glisează peste imagine

- **Pe scurt:** fiecare ieșire este o sumă ponderată a unei vecinătăți de $k \times k$, cu aceleași ponderi peste tot.

$$Z_o(i,j) = b_o + \sum_{c=1}^{C_{\text{in}}} \sum_{u=0}^{k-1} \sum_{v=0}^{k-1} K_{o,c}(u,v)\, X_c(is + u - p,\ js + v - p)$$

- Harta de trăsături a filtrului o: $A_o = \text{ReLU}(Z_o)$, cu unitatea liniară rectificată (ReLU) $\max(0, z)$
  - K este învățat prin propagarea înapoi a erorii.

- **unde**
  - X_c canalul de intrare c
  - $K_{o,c}$ nucleul de $k \times k$ de la canalul c la harta de ieșire o
  - b_o termenul liber
  - s pasul (stride)
  - p completarea cu zerouri (padding)

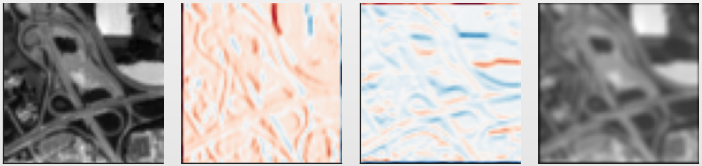
- **În practică:** un filtru este un detector de tipare testat în fiecare poziție a imaginii, cu un cost de doar $k^2 C_{\text{in}}$ ponderi.

(Goodfellow et al., 2016, cap. 9)

Filtrele construite de mînă găsesc deja drumurile și marginile parcelelor

- **Pe scurt:** un nucleu Sobel răspunde acolo unde luminozitatea se schimbă pe o direcție; un nucleu de medie estompează imaginea.

$$K_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad K_y = K_x^\top, \quad K_{\text{mean}} = \tfrac{1}{9}\mathbf{1}_{3\times 3}$$



Intrare (autostradă) | Muchii verticale | Muchii orizontale | Media 3×3 (estompare)

Sursa: EuroSAT, imagini Sentinel-2 (Helber et al., 2019). Date: o imagine Sentinel-2 de 64 × 64 (clasa Autostradă), niveluri de gri.

- Completarea $p = 1$ păstrează dimensiunea de 64×64
  - O CNN învață zeci de astfel de nuclee în fiecare strat, în loc să le fixeze.

(calcul propriu)

NN_L4_conv_edges

Dimensiunea ieșirii, completarea și pasul

- **Pe scurt:** trei numere întregi decid cît de mare este fiecare hartă de trăsături.

$$H_{\text{out}} = \left\lfloor \frac{H_{\text{in}} + 2p - k}{s} \right\rfloor + 1$$

- **unde**
  - H_{in} înălțimea intrării (aceeași formulă pentru lățime)
  - k dimensiunea nucleului
  - p completarea
  - s pasul

- **Exemple**, $H_{\text{in}} = 64$, $k = 3$
  - $p = 1, s = 1$: $H_{\text{out}} = 64$ (“same”)
  - $p = 0, s = 1$: $H_{\text{out}} = 62$ (“valid”)
  - $p = 1, s = 2$: $H_{\text{out}} = 32$ (subeșantionare)
- Parametrii stratului: $(k^2 C_{\text{in}} + 1)\, C_{\text{out}}$, independenți de dimensiunea imaginii.

- **În practică:** aceeași rețea rulează pe imagini de orice dimensiune
  - Crește doar costul în operații, $H_{\text{out}} W_{\text{out}} k^2 C_{\text{in}} C_{\text{out}}$.

(calcul propriu)

Agregarea rezumă o vecinătate; agregarea globală rezumă imaginea

- **Pe scurt:** agregarea prin maxim păstrează cel mai puternic răspuns din fiecare fereastră, micșorează harta și o face ușor invariantă la deplasări.

$$P_o(i,j) = \max_{0 \le u,v < 2} A_o(2i+u, 2j+v)$$

$$g_o = \frac{1}{HW} \sum_{i,j} A_o(i,j)$$

- **unde**
  - P_o agregarea prin maxim 2×2 cu pasul 2 (înjumătățește înălțimea și lățimea)
  - g_o agregarea prin medie globală (GAP), cîte un număr pentru fiecare canal

- CNN-ul nostru pentru Fashion-MNIST: hărți $28 \times 28 \to 14 \times 14 \to 7 \times 7$, apoi GAP dă 128 de numere, iar un strat liniar dă cele 10 scoruri ale claselor.
- Gradientul agregării prin maxim: trece doar prin poziția maximului din fiecare fereastră.
- GAP înlocuiește straturile dense mari (mii de ponderi) cu o medie (zero ponderi).

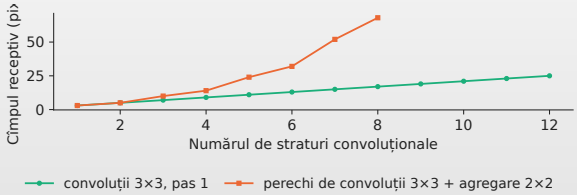
- **În practică:** GAP este motivul pentru care o CNN modernă are puțini parametri în partea finală și poate fi explicată cu hărți de activare ale claselor.

(Bishop & Bishop, 2024, sect. 10.2)

Adîncimea, agregarea și cîmpul receptiv

- **Pe scurt:** cîmpul receptiv este zona din intrare care poate influența o unitate.

$$r_l = r_{l-1} + (k_l - 1) \prod_{i<l} s_i, \qquad r_0 = 1$$



Sursa: calcul propriu din recurența cîmpului receptiv. Date: recurența cîmpului receptiv, fără date.

- Douăsprezece straturi 3×3 cu pasul 1 văd doar 25×25 pixeli
  - Opt straturi cu agregare 2×2 după fiecare pereche văd 68×68.

(calcul propriu)

NN_L4_receptive_field

Propagarea înapoi printr-o convoluție este tot o convoluție

- **Pe scurt:** pentru că nucleul este partajat, gradientul lui adună contribuțiile tuturor pozițiilor.

$$\frac{\partial \mathcal{L}}{\partial K_{o,c}(u,v)} = \sum_{i,j} \delta_o(i,j)\, X_c(is+u-p,\, js+v-p)$$

$$\delta_o(i,j) = \frac{\partial \mathcal{L}}{\partial Z_o(i,j)}$$

- **unde**
  - δ_o harta erorilor pentru ieșirea o, calculată prin propagarea înapoi din stratul superior
  - Pentru $s = 1$, $\partial \mathcal{L}/\partial X_c = \sum_o \delta_o \star \mathrm{rot}_{180}(K_{o,c})$, o convoluție completă

- Partajarea ponderilor transformă milioane de gradienți specifici fiecărei poziții într-o singură sumă pentru fiecare pondere.
- În cod, diferențierea automată face totul: `loss.backward()` pe un `nn.Conv2d`.
- Derivarea completă este în anexă (convoluția ca produs de matrice).

- **În practică:** o trecere înainte și una înapoi costă aproximativ același număr de operații, deci timpul de antrenare crește odată cu costul trecerii înainte.

(Goodfellow et al., 2016, secț. 9.5)

Normalizarea pe loturi menține fiecare canal pe aceeași scală

- **Pe scurt:** fiecare canal este standardizat pe lot și pe poziții, apoi rescalat cu doi parametri învățați.

$$\hat{a} = \frac{a - \mu_{\mathcal{B},o}}{\sqrt{\sigma^2_{\mathcal{B},o} + \epsilon}}, \qquad \mathrm{BN}(a) = \gamma_o \hat{a} + \beta_o$$

- **unde**
  - $\mu_{\mathcal{B},o}, \sigma^2_{\mathcal{B},o}$ media și varianța canalului o pe lotul $\mathcal{B}$ și pe toate pozițiile
  - γ_o, β_o învățați
  - La testare, mediile mobile înlocuiesc statisticile lotului
- Toate rețelele din acest curs folosesc normalizarea pe loturi (BN) după fiecare convoluție: Conv $\rightarrow$ BN $\rightarrow$ ReLU.
- Permite rate de învățare mai mari
  - Antrenările noastre pe CIFAR folosesc o programare one-cycle pînă la 0,1 (aleasă a priori).
- Nu uitați `model.eval()` înainte de predicție: altfel BN folosește statisticile lotului de test.

- **În practică:** un model care merge bine în notebook, dar prost în producție, a uitat adesea modul eval sau scalarea intrărilor din antrenare.

(Ioffe & Szegedy, 2015)

Cuprins

- Imaginile ca date și motivarea convoluțiilor
- Stratul convoluțional
- Studiu de caz: clasificarea fotografiilor de produse cu regresie softmax și CNN
- Arhitecturi: de la LeNet la ResNet
- Transferul învățării: utilizarea terenului din imagini satelitare

Protocolul experimental pentru cele trei modele

- **Reper**: regresia softmax (logistică multinomială)
  - $\hat{\mathbf{p}} = \text{softmax}(\mathbf{W}\text{vec}(\mathbf{X}) + \mathbf{b})$, 7 850 de parametri
- **MLP**: 784–256–128–10, ReLU, dropout 0,2
  - 235 146 de parametri
- **CNN**: 2 blocuri [Conv3×3–BN–ReLU]×2 + agregare maxim, Conv 128, GAP, liniar
  - 140 778 de parametri
- **Aceeași antrenare pentru toate**
  - AdamW, loturi de 128, 12 epoci, rată de învățare one-cycle
  - Entropia încrucișată $\mathcal{L} = -\frac{1}{n}\sum_i \log \hat{p}_{i,y_i}$
  - Pixeli standardizați cu media de antrenare 0,2859 și abaterea standard 0,3530
- **Comparație corectă**
  - Oprire timpurie pe cele 10 000 de imagini de validare
  - 3 semințe pentru fiecare model; imaginile de test sînt folosite o singură dată

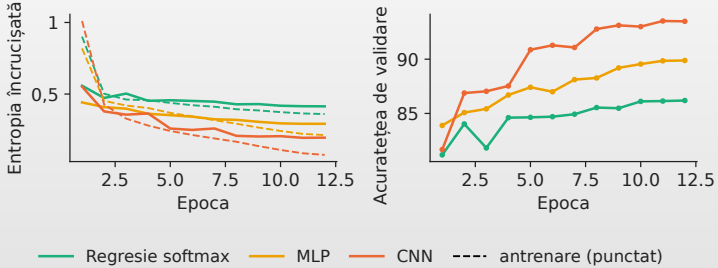
- **În practică:** timpul total de antrenare al celor nouă rulări: 299 de secunde pe procesorul grafic al unui laptop.

(calcul propriu)

Exercițiu practic: §3

CNN învață mai repede și generalizează mai bine

- **Pe scurt:** pierderea de validare scade cel mai mult pentru CNN; MLP se potrivește datelor de antrenare, dar nu mai progresează pe imagini noi.



Sursa: Fashion-MNIST, Zalando Research (Xiao et al., 2017). Date: 50.000 de imagini de antrenare și 10.000 de validare; sămînța 20261108.

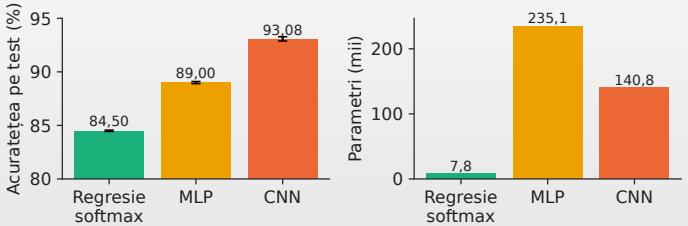
- Cea mai bună epocă pe validare (sămînța 20261108): CNN 11, MLP 12, softmax 12, din 12.

(calcul propriu)

NN_L4_fmnist_curves

Acuratețea și numărul de parametri ai celor trei modele

- **Pe scurt:** partajarea ponderilor aduce acuratețe și economisește parametri în același timp.



Sursa: Fashion-MNIST, Zalando Research (Xiao et al., 2017). Date: 10.000 de imagini de test; media și abaterea standard pe 3 semințe.

- Acuratețea pe test (medie $\pm$ abatere standard, 3 semințe): softmax $84{,}50 \pm 0{,}06$ %, MLP $89{,}00 \pm 0{,}10$ %, CNN $93{,}08 \pm 0{,}20$ %.
- CNN elimină 37,1 % din erorile MLP, cu de 1,67 ori mai puțini parametri.

(calcul propriu)

NN_L4_fmnist_compare

Testul McNemar pentru cîștigul de acuratețe pe aceleași imagini de test

- **Pe scurt:** două clasificatoare evaluate pe aceleași imagini diferă doar prin imaginile pe care exact unul dintre ele le clasifică bine.

$$\chi^2 = \frac{(|b-c|-1)^2}{b+c} \overset{H_0}{\sim} \chi^2_1$$

- **unde**
  - b imaginile corecte doar pentru CNN
  - c corecte doar pentru MLP
  - H_0: ambele au aceeași rată de eroare
  - Valoarea critică la 5 % este 3,84

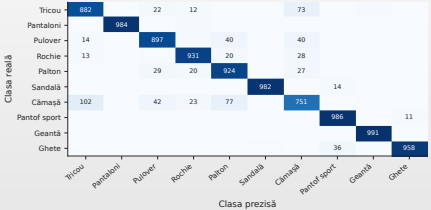
- Sămînța 20261108: $b = 614$, $c = 226$, $\chi^2 = 178{,}3$: ipoteza nulă este respinsă.
- Pe cele 3 semințe, statistica t pereche a diferenței de acuratețe este 31,38.
- Semințele măsoară zgomotul antrenării
  - Testul McNemar măsoară zgomotul setului de test
  - Raportați-le pe amîndouă.

- **În practică:** înainte de a înlocui un model aflat în producție, arătați că avantajul rezistă unui test pe aceleași cazuri, nu doar o medie mai mare.

(McNemar, 1947)

Matricea de confuzie: erori între produse cu forme asemănătoare

- **Pe scurt:** cele mai multe greșeli apar între articolele pentru partea de sus a corpului, cu forme asemănătoare.



Sursa: Fashion-MNIST, Zalando Research (Xiao et al., 2017). Date: 10.000 de imagini de test, CNN cu sămînța 20261108.

- Clasa cea mai slabă: Cămașă (sensibilitate 75,1 %)
  - Cea mai mare confuzie: Cămașă clasificat drept Tricou, 102 de imagini.

- **În practică:** trimiteți la verificare umană doar fotografiile cu încredere scăzută din clasele slabe: costul erorilor scade acolo unde este cel mai mare.

(calcul propriu)

NN_L4_fmnist_confusion

Filtrele din primul strat detectează muchii și contraste

- **Pe scurt:** nimeni nu i-a cerut rețelei să caute muchii; coborîrea pe gradient le-a găsit singură.



Sursa: Fashion-MNIST, Zalando Research (Xiao et al., 2017). Date: primul strat convoluțional al CNN (32 de filtre de 3 × 3), sămînța 20261108.

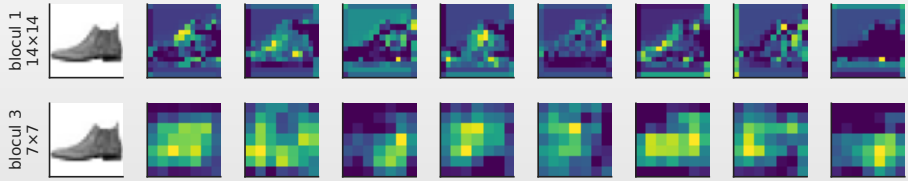
- Roșu = pondere pozitivă, albastru = negativă: multe nuclee sînt diferențe orientate, ca nucleele Sobel alese de mînă.
- Straturile mai adînci le combină în părți (gulere, tălpi, mînere) (Zeiler & Fergus, 2014).

(calcul propriu; Zeiler & Fergus, 2014)

NN_L4_fmnist_filters

Hărțile de trăsături la adîncimi crescătoare

- **Pe scurt:** hărțile timpurii păstrează conturul ghetei; cele adînci sînt mici, dar arată unde se află părțile informative.



Sursa: Fashion-MNIST, Zalando Research (Xiao et al., 2017). Date: o imagine de test (ghete); cele mai active 8 canale din fiecare bloc.

- Blocul 1: 32 de hărți de 14×14
  - Blocul 3: 128 de hărți de 7×7, apoi mediate de GAP.

(calcul propriu)

NN_L4_fmnist_featuremaps

Cuprins

- Imaginile ca date și motivarea convoluțiilor
- Stratul convoluțional
- Studiu de caz: clasificarea fotografiilor de produse cu regresie softmax și CNN
- Arhitecturi: de la LeNet la ResNet
- Transferul învățării: utilizarea terenului din imagini satelitare

Arhitecturi pentru recunoașterea imaginilor, 1998–2021

Rețeaua	Anul	Ideea principală
LeNet-5	1998	convoluție + subeșantionare pentru cifre scrise de mînă (banc de probă MNIST); antrenată prin propagarea înapoi a erorii
AlexNet	2012	CNN adîncă pe ImageNet antrenată pe GPU; ReLU, dropout, augmentarea datelor
Visual Geometry Group (VGG)	2015	numai nuclee 3×3: două nuclee 3×3 suprapuse văd 5×5 cu $18C^2$ în loc de $25C^2$ ponderi
ResNet	2016	conexiuni reziduale $\mathbf{y} = \mathbf{x} + F(\mathbf{x})$: rețelele cu peste 100 de straturi devin antrenabile
Transformer vizual	2021	fragmente de imagine tratate ca unități lexicale, cu atenție (cursul 7)

⊡ ResNet-18, folosit mai jos pentru transferul învățării, are 11 181 642 de parametri (numărați de noi, cu un strat final de 10 clase).

(LeCun et al., 1998; Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; He et al., 2016)

Conexiunile reziduale și calea identității

- **Pe scurt:** un bloc își adună ieșirea la intrare, astfel încît gradientul are mereu un drum direct înapoi.

$$\mathbf{y} = \mathrm{ReLU}\left(\mathbf{x} + F(\mathbf{x}; \mathbf{W})\right), \qquad \frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial \mathcal{L}}{\partial \mathbf{y}}\left(\mathbf{I} + \frac{\partial F}{\partial \mathbf{x}}\right)$$

- **unde**
  - F două straturi Conv–BN
  - Cînd forma se schimbă, $\mathbf{x}$ trece printr-o convoluție 1×1 cu pasul 2
  - Derivata este scrisă pentru suma dinaintea activării

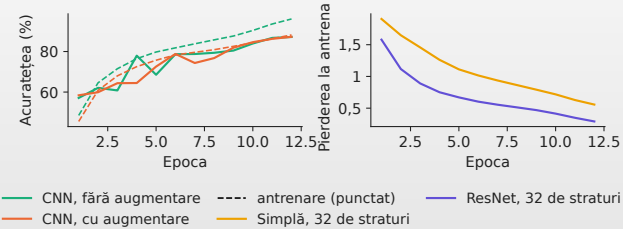
- Rețelele adînci simple suferă de degradare: cele mai adînci au eroare de antrenare mai mare, o problemă de optimizare, nu de supraînvățare.
- Dacă cel mai bun bloc este identitatea, este suficient ca $F = 0$.
- Pe CIFAR-10 (diapozitivul următor): aceleași 32 de straturi, cu și fără scurtătură.

- **În practică:** blocurile reziduale sînt alegerea implicită în toate modelele moderne de viziune și de limbaj, inclusiv în transformere.

(He et al., 2016)

Efectele augmentării datelor și ale conexiunilor reziduale

- **Pe scurt:** stînga: aceeași CNN pe fotografiile Canadian Institute for Advanced Research (CIFAR-10); dreapta: rețele de 32 de straturi care diferă doar prin scurtătură.



Sursa: CIFAR-10 (Krizhevsky, 2009). Date: 45.000 de imagini de antrenare și 5.000 de validare; sămînța 20261108.

- Acuratețea pe test: CNN 86,56 % fără și 86,86 % cu augmentare (decalaj antrenare–validare 8,9 față de 0,9 puncte).
- Rețeaua simplă cu 32 de straturi 79,96 %, ResNet-32 87,81 %
  - Pierderea finală la antrenare 0,557 față de 0,291.

(calcul propriu)

NN_L4_cifar_curves

Augmentarea datelor ca informație a priori

- **Pe scurt:** antrenăm pe transformări aleatoare care nu ar trebui să schimbe eticheta, adică mediem pierderea peste ele.

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{T \sim \mathcal{T}} \Big[\ell\big(f_\theta(T(\mathbf{X}_i)), y_i\big) \Big]$$

- **unde**
  - $\mathcal{T}$ mulțimea transformărilor care păstrează eticheta
  - CIFAR-10: decupări aleatoare după o completare de 4 pixeli și oglindiri orizontale
  - EuroSAT: oglindiri și rotații cu 90°

- Cîștigul pe CIFAR-10, pentru aceeași rețea și același buget: +0,30 puncte procentuale de acuratețe pe test.
- Alegeți $\mathcal{T}$ după domeniu: o cifră 6 oglindită nu mai este 6, dar o imagine satelitară nu are "sus".
- Sensibilitatea pe clase a ResNet-32: cea mai bună automobil 95,7 %, cea mai slabă pisică 71,9 %.

- **În practică:** augmentarea înseamnă date suplimentare gratuite
  - Este prima pîrghie de folosit înainte de a cumpăra mai multe etichete.

(calcul propriu)

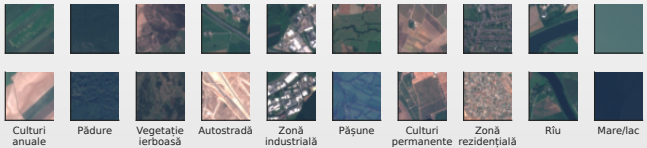
Cuprins

- Imaginile ca date și motivarea convoluțiilor
- Stratul convoluțional
- Studiu de caz: clasificarea fotografiilor de produse cu regresie softmax și CNN
- Arhitecturi: de la LeNet la ResNet
- Transferul învățării: utilizarea terenului din imagini satelitare

Imagini satelitare: utilizarea terenului pentru creditori, asigurători și investitori

- **Pe scurt:** zece clase de utilizare a terenului din imagini Sentinel-2 ale unor țări din Uniunea Europeană (UE); întrebarea este cu cît de puține etichete ne descurcăm.

$$\mathbf{X} \in [0,1]^{64\times 64\times 3}, \qquad y \in \{\text{culturi}, \ldots, \text{mare/lac}\}$$



Sursa: EuroSAT, imagini Sentinel-2 (Helber et al., 2019). Date: 27.000 de imagini de 64 x 64 pixeli, 10 clase.

- 27 000 de imagini roșu, verde, albastru (RGB) de 64×64 pixeli
  - Între 2 000 și 3 000 pe clasă.

- **În practică:** un creditor agricol verifică dacă un împrumut finanțează culturi
  - Un asigurător cartografiază zonele rezidențiale expuse inundațiilor.

(Helber et al., 2019)

NN_L4_eurosat_samples

Transferul învățării: trăsături învățate pe un milion de fotografii

- **Pe scurt:** primele straturi ale unei rețele antrenate pe ImageNet detectează muchii și texturi utile pentru orice imagine.

$$f_\theta(\mathbf{X}) = g_\psi\left(\phi_\omega(\mathbf{X})\right), \quad \omega^{(0)} = \omega_{\text{ImageNet}}$$

- **unde**
  - ϕ_ω corpul convoluțional (512 trăsături după GAP)
  - g_ψ un nou strat final liniar cu 10 ieșiri

- **Extragerea trăsăturilor**
  - ω înghețat; se estimează doar ψ (aici o regresie logistică, cu C ales pe validare)
- **Ajustarea fină**
  - Toate ponderile sînt antrenate, cu rată de învățare mai mică pentru corp ($3 \cdot 10^{-4}$) decît pentru stratul final ($3 \cdot 10^{-3}$), alese a priori
- **De la zero**
  - O CNN cu 4 blocuri, 1 176 746 de parametri, inițializare aleatoare

- **În practică:** extragerea trăsăturilor este cea mai ieftină: o singură trecere prin imagini și o regresie logistică.

(Yosinski et al., 2014)

Protocolul experimental pentru cele trei strategii de transfer

- **Împărțirea**, stratificată pe clase
  - 16 200 de imagini de antrenare / 5 400 de validare / 5 400 de test
- **Intrarea**
  - Imaginile sînt redimensionate la 112×112 pentru ResNet-18 și scalate cu media și abaterea standard ImageNet ale fiecărui canal
  - $\tilde{x}_c = (x_c - \mu_c)/s_c$
- **Antrenarea**
  - AdamW, loturi de 64, programare one-cycle; oglindiri și rotații cu 90°
  - De la zero 18 epoci; ajustare fină 6 epoci
  - Se păstrează modelul cu cea mai bună acuratețe de validare
- **Costul**
  - 1225 de secunde pentru cele trei rulări principale, pe procesorul grafic al unui laptop

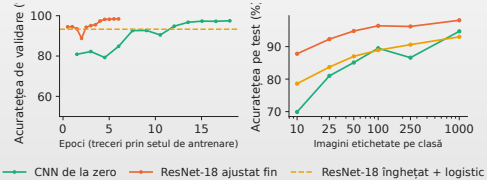
În practică: fixarea imaginilor de test înaintea oricărei modelări face comparabile cele trei strategii.

(calcul propriu)

Exercițiu practic: §5

Ajustarea fină este cea mai precisă, iar transferul contează cel mai mult cînd etichetele sînt puține

- **Pe scurt:** stînga: învățarea cu toate etichetele; dreapta: acuratețea pe test cînd sînt disponibile doar n imagini etichetate pe clasă.



Sursa: EuroSAT, imagini Sentinel-2 (Helber et al., 2019). Date: 16.200 de imagini de antrenare, 5.400 de validare, 5.400 de test; dreapta: media pe 2 semințe.

- Acuratețea pe test, cu toate etichetele: de la zero 97,78 %, înghețat + logistic 93,48 %, ajustat fin 98,76 %.
- Cu 10 imagini pe clasă: de la zero 69,9 %, înghețat 78,6 %, ajustat fin 87,8 %.

(calcul propriu)

NN_L4_transfer_curves

Bugetul de etichete și economia obținută prin transferul învățării

Imagini/clasă	De la zero	Înghețat	Ajustat fin
10	69,9	78,6	87,8
25	81,0	83,7	92,3
50	85,1	87,0	94,8
100	89,5	88,9	96,4
250	86,6	90,6	96,2
1000	94,7	93,0	98,1

- Acuratețea pe test (%), media a 2 extrageri aleatoare ale imaginilor etichetate; aceleași 5.400 de imagini de test.

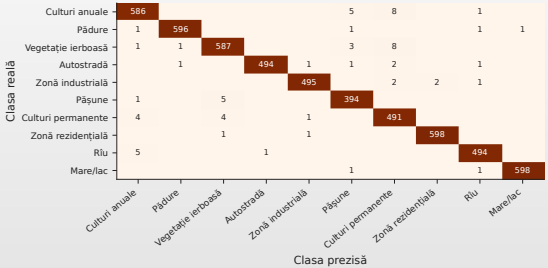
- Cu doar 100 de imagini etichetate în total, rețeaua preantrenată depășește clar o CNN antrenată de la zero pe aceleași etichete.
- CNN-ul de la zero are nevoie de întregul buget pentru a se apropia de rețeaua ajustată fin.
- Rularea întregului tabel a durat 1857 de secunde.

- **În practică:** etichetarea unei imagini costă un minut de lucru al unui expert
  - Transferul învățării transformă săptămâni de etichetare într-o după-amiază.

(calcul propriu)

Erorile rămase: culturi anuale față de culturi permanente

- **Pe scurt:** rețeaua ajustată fin confundă clase care arată la fel din spațiu.



Sursa: EuroSAT, imagini Sentinel-2 (Helber et al., 2019). Date: 5.400 de imagini de test, ResNet-18 ajustat fin, sămînța 20261108.

- 67 de erori la 5 400 de imagini
  - Cea mai mare: Culturi anuale clasificat drept Culturi permanente (8)
  - Macro-F1 98,72 %.

(calcul propriu)

NN_L4_eurosat_confusion

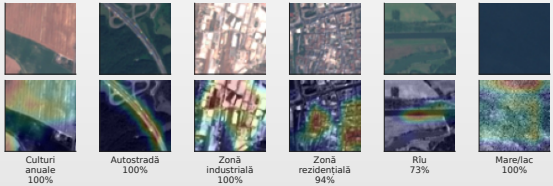
Grad-CAM: regiunile imaginii care determină clasificarea

- **Pe scurt:** ponderăm fiecare hartă de trăsături după cît reacționează scorul clasei la ea, adunăm și păstrăm partea pozitivă.

$$\alpha_o^c = \frac{1}{HW} \sum_{i,j} \frac{\partial z_c}{\partial A_o(i,j)}$$

$$M^c = \mathrm{ReLU}\left(\sum_o \alpha_o^c A_o \right)$$

- **unde**
  - z_c scorul (logitul) clasei c
  - A_o harta de trăsături o din stratul ales
  - M^c harta de activare a clasei (CAM), mărită la dimensiunea imaginii



Sursa: EuroSAT, imagini Sentinel-2 (Helber et al., 2019). Date: șase imagini de test; Grad-CAM pe hărțile de 7 × 7 ale blocului 3 din ResNet-18 ajustat fin.

- **În practică:** un auditor poate verifica dacă "zonă industrială" este decisă de acoperișuri, nu de un nor sau de un artefact de margine.

(Selvaraju et al., 2017; calcul propriu)

NN_L4_gradcam

Ponderile utilizării terenului estimate din imagini satelitare

- **Pe scurt:** o întrebare de portofoliu privește ponderi și totaluri, deci erorile care se compensează contează mai puțin decît cele care nu se compensează.

$$\hat{\pi}_k = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\hat{y}_i = k\}, \qquad \mathbb{E}[\hat{\pi}_k] = \sum_j \pi_j \, P(\hat{y} = k \mid y = j)$$

- **unde**
  - π_k ponderea reală a clasei k într-o regiune
  - $P(\hat{y} = k \mid y = j)$ matricea de confuzie, estimată pe imaginile de test

- Imaginile de test: ponderea culturilor (anuale + permanente) este 20,37 % în realitate și 20,54 % prezisă.
- Cea mai mare eroare a ponderii oricărei clase: 0,20 puncte procentuale.
- Inversarea matricei de confuzie corectează ponderile atunci cînd ratele de eroare sînt stabile (schimbarea domeniului strică acest lucru).

- **În practică:** pentru un asigurător sau un creditor agricol, rezultatul este ponderea hectarelor pe regiune, cu o marjă de eroare, nu o etichetă pentru fiecare imagine.

(calcul propriu)

Avantajele și riscurile transferului învățării

- **Ajută cel mai mult**
  - Cînd sînt puține etichete (panoul din dreapta al graficului de transfer)
  - Cînd domeniul sursă are statistici de nivel jos similare (muchii, texturi)
  - Cînd puterea de calcul este limitată: trăsăturile înghețate costă o singură trecere înainte
- **Poate induce în eroare**
  - Schimbarea domeniului: alt anotimp, alt senzor, altă țară decît imaginile de antrenare
  - Indicii false învățate din sursă (filigrane, margini)
  - Setul de test trebuie să provină din populația în care va fi folosit modelul

- **În practică:** înainte de utilizare, etichetați un mic eșantion din regiunea țintă și măsurați din nou
  - Modelul ajustat fin se actualizează ieftin.

(Yosinski et al., 2014; Helber et al., 2019)

Convoluții pentru secvențe și grafice

- **Pe scurt:** o convoluție unidimensională glisează în timp, la fel cum una bidimensională glisează în spațiu.

$$z_t = b + \sum_{c} \sum_{u=0}^{k-1} K_c(u)\, x_{c,\,t-u}$$

- **unde**
  - $x_{c,t}$ seria de intrare c la momentul t (randamente, volum, consum)
  - Cauzală: doar valori trecute

- O CNN unidimensională este un set învățat de filtre de tip medie mobilă
  - Cursul 5 o compară cu rețelele recurente.
- Graficele piețelor pot fi citite ca imagini de un model viziune–limbaj (VLM): cercetarea grupului nostru, în secțiunea următoare.
- Aceleași reguli: date reale, împărțire în ordine cronologică, întîi un model de referință simplu.

- **În practică:** un departament de risc poate examina zilnic sute de grafice cu un model, iar analistul se ocupă doar de cele semnalate.

(Bishop & Bishop, 2024, cap. 10)

Test 1: dimensiunea ieșirii

- O imagine de 64×64 trece printr-o convoluție 3×3 cu completare 1 și pasul 2
  - Cît de mare este harta de ieșire?

Răspuns

- $\lfloor (64 + 2 - 3)/2 \rfloor + 1 = 32$, deci 32×32.

Test 2: parametrii

- De ce depășește CNN-ul nostru MLP pe Fashion-MNIST, deși are mai puțini parametri?

Răspuns

- Partajarea ponderilor și conectivitatea locală codifică structura imaginilor
  - CNN atinge 93,08 % cu 140 778 de parametri, față de 89,00 % cu 235 146.

Test 3: adîncimea

- O rețea simplă de 32 de straturi are o pierdere la antrenare mai mare decît una rezidual de aceeași adîncime
  - Este supraînvățare?

Răspuns

- Nu: supraînvățarea scade pierderea la antrenare
  - Aceasta este problema degradării (0,557 față de 0,291), rezolvată de scurtătura identitate.

Test 4: un buget de etichete

- Puteți eticheta în total doar 100 de imagini satelitare
  - Ce strategie alegeți?

Răspuns

- O rețea preantrenată: cu 10 imagini pe clasă, ajustarea fină dă 87,8 %, iar trăsăturile înghețate 78,6 %, față de 69,9 % de la zero.

Test 5: explicația

- Harta Grad-CAM a unei imagini "zonă industrială" evidențiază un nor
  - Ce faceți?

Răspuns

- O tratați ca pe un semnal de indiciu fals: verificați imagini asemănătoare, adăugați exemple cu nori sau o mască de nori și testați din nou pe regiunea țintă.

Recapitulare

- Imaginile ca date și motivarea convoluțiilor
- Stratul convoluțional
- Studiu de caz: clasificarea fotografiilor de produse cu regresie softmax și CNN
- Arhitecturi: de la LeNet la ResNet
- Transferul învățării: utilizarea terenului din imagini satelitare

- O convoluție este un filtru local partajat
  - Completarea, pasul și agregarea stabilesc dimensiunea hărții, iar adîncimea și agregarea cîmpul receptiv.
- Pe imagini reale, CNN depășește MLP cu mai puțini parametri
  - Blocurile reziduale fac adîncimea antrenabilă
  - Augmentarea înseamnă date gratuite.
- Transferul învățării cîștigă atunci cînd etichetele sînt rare
  - Urmează secvențele și rețelele recurente (cursul 5).

Cercetări ale grupului nostru: modele de viziune în finanțe

- **Un cadru multimodal viziune–limbaj pentru detectarea anomaliilor financiare** (Jheng, Pele, Tak & Găman, 2026, PICBE), premiul Best Paper la International Conference on Business Excellence (ICBE) 2026
  - Indicatorii de piață sînt proiectați într-o imagine standardizată, cu mai multe panouri; clasificarea regimurilor de piață devine o problemă de recunoaștere vizuală
  - Model viziune–limbaj folosit fără antrenare suplimentară (GPT-4.1, transformer generativ preantrenat), cu etichete din cuantile glisante și justificări structurate în limbaj natural
  - Teste în afara eșantionului pe S&P 500: detectarea anomaliilor sistemice cu o fidelitate promițătoare și cu explicații interpretabile
- **Quantlet: platforma de cunoștințe pentru fragmente de cod** (Găman, Winkel, Zuo, Härdle & Pele, 2026, Computational Statistics)
  - Platforma pe care fiecare grafic din acest curs este publicat ca Quantlet reproductibil
- Rezultatele de pe acest diapozitiv sînt cele raportate în lucrări; celelalte cifre din curs provin din calculele noastre.

(Academia de Studii Economice din București; Academia Română, Institutul de Prognoză Economică)

Lecturi și resurse suplimentare

- **Bishop & Bishop (2024)**, cap. 10: rețele convoluționale, cu derivările din acest curs.
- **Goodfellow, Bengio & Courville (2016)**, cap. 9: convoluție, agregare, variante.
- **He et al. (2016)**: articolul ResNet, scurt și ușor de citit.
- **Acest curs**: cod, date, notebook-uri Jupyter — github.com/danpele/NEURAL_BIZ.
- **Exercițiu practic**: notebook-ul reproduce graficele acestui curs — notebooks/.
- **Seminarul 4**: transferul învățării pe EuroSAT, cu propriul buget de etichete.

Bibliografie (1/2)

- Bishop, C. M. & Bishop, H. (2024). *Deep Learning: Foundations and Concepts*, ch. 10. Springer. doi:10.1007/978-3-031-45468-4
- Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*, ch. 9. MIT Press. deeplearningbook.org
- He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. *CVPR*, 770–778. doi:10.1109/CVPR.2016.90
- Helber, P., Bischke, B., Dengel, A. & Borth, D. (2019). EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE JSTARS*, 12(7), 2217–2226. doi:10.1109/JSTARS.2019.2918242
- Ioffe, S. & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML*, PMLR 37, 448–456. arXiv:1502.03167
- Jheng, S.-L., Pele, D. T., Tak, R. & Găman, Ș. (2026). A multimodal vision-language framework for financial anomaly detection. *Proceedings of the International Conference on Business Excellence*, 20(1), 191–201. doi:10.2478/picbe-2026-0020
- Găman, Ș., Winkel, J., Zuo, X., Härdle, W. K. & Pele, D. T. (2026). Quantlet: the code snippet knowledge platform. *Computational Statistics*, 41(3), 66. doi:10.1007/s00180-026-01742-6
- Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto. cs.toronto.edu
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *NIPS*, 25, 1097–1105. doi:10.1145/3065386

Bibliografie (2/2)

- LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. doi:10.1109/5.726791
- McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2), 153–157. doi:10.1007/BF02295996
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *ICCV*, 618–626. doi:10.1109/ICCV.2017.74
- Simonyan, K. & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *ICLR*. arXiv:1409.1556
- Xiao, H., Rasul, K. & Vollgraf, R. (2017). Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747
- Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. (2014). How transferable are features in deep neural networks? *NIPS*, 27, 3320–3328. arXiv:1411.1792
- Zeiler, M. D. & Fergus, R. (2014). Visualizing and understanding convolutional networks. *ECCV*, 818–833. doi:10.1007/978-3-319-10590-1_53

Anexă: convoluția ca produs de matrice

- Fiecare zonă de $k \times k \times C_{\text{in}}$ a intrării devine un rînd al matricei $\mathbf{P} \in \mathbb{R}^{H_{\text{out}} W_{\text{out}} \times k^2 C_{\text{in}}}$ ("im2col").
- Nucleele devin coloanele matricei $\mathbf{K} \in \mathbb{R}^{k^2 C_{\text{in}} \times C_{\text{out}}}$
  - Atunci $\mathbf{Z} = \mathbf{P}\mathbf{K} + \mathbf{1}\mathbf{b}^\top$.
- Gradienții rezultă din regulile matriceale din cursul 2: $\partial \mathcal{L}/\partial \mathbf{K} = \mathbf{P}^\top \mathbf{\Delta}$ și $\partial \mathcal{L}/\partial \mathbf{P} = \mathbf{\Delta}\mathbf{K}^\top$, cu $\mathbf{\Delta} = \partial \mathcal{L}/\partial \mathbf{Z}$.
- Readunarea rîndurilor din $\partial \mathcal{L}/\partial \mathbf{P}$ la pozițiile pixelilor lor ("col2im") dă $\partial \mathcal{L}/\partial \mathbf{X}$: convoluția completă din textul principal.
- O convoluție 1×1 este cazul particular $k = 1$: un strat dens aplicat canalelor fiecărui pixel.

Anexă: CNN-ul din studiul de caz, în PyTorch

```
block = lambda a, b:  [nn.Conv2d(a, b, 3, padding=1), nn.BatchNorm2d(b), nn.ReLU()]
cnn = nn.Sequential(*block(1, 32), *block(32, 32), nn.MaxPool2d(2),    # 28 -> 14
                    *block(32, 64), *block(64, 64), nn.MaxPool2d(2),    # 14 -> 7
                    *block(64, 128), nn.AdaptiveAvgPool2d(1), nn.Flatten(),
                    nn.Dropout(0.3), nn.Linear(128, 10))
resnet = torchvision.models.resnet18(weights='IMAGENET1K_V1')
resnet.fc = nn.Linear(512, 10)               # new head for the 10 land-use classes
```

- Codul complet: lectures/L4/analysis_L4.py și notebook-ul acestui curs.