



Technical University of Cluj - Napoca
Computer Science Department

Procesarea Imaginilor

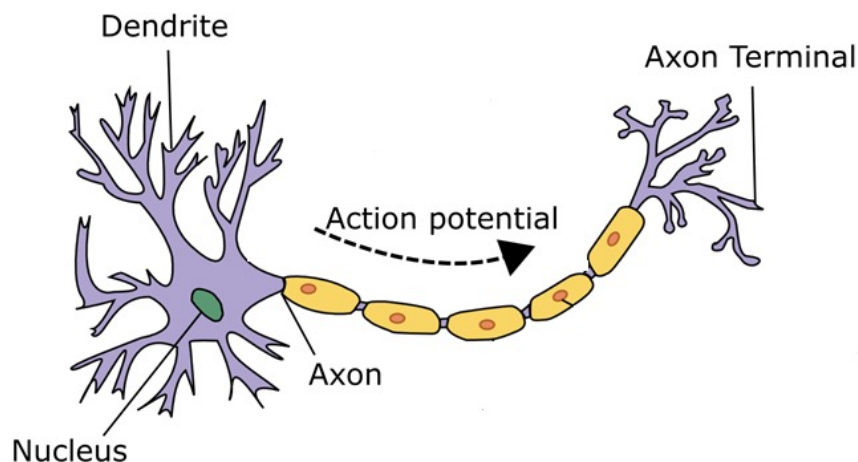
Curs 13

Procesarea imaginilor folosind rețele neuronale.



Rețele neuronale convoluționale

Analogie cu biologia



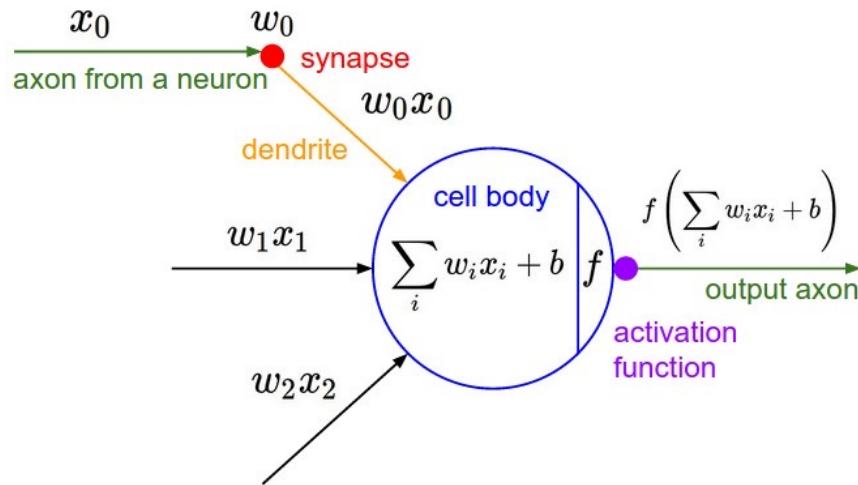
Neuronul biologic: are ca intrări semnale electrice primite pe dendrite, prin intermediul sinapselor (conexiuni cu alți neuroni). Puterea sinapselor influențează ponderea semnalului de intrare. Dacă neuronul este activat, se generează semnal de ieșire pe axon (care se poate conecta apoi la alte dendrite, prin alte sinapse).

sursa imaginii: <http://cs231n.github.io/convolutional-networks/>



Rețele neuronale convoluționale

Analogie cu biologia



Neuronul artificial: are ca intrări valori numerice, primite de la intrarea rețelei (datele de clasificat) sau de la ieșirea altor neuroni artificiali.

- Tăria sinapselor este înlocuită de ponderi ale intrărilor
- Ieșirea este o combinație liniară a intrărilor, urmată de o funcție de activare neliniară.

sursa imaginii: <http://cs231n.github.io/convolutional-networks/>

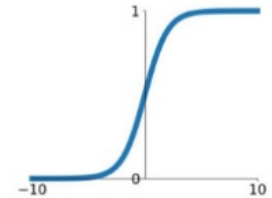


Funcții de activare

- Funcția sigmoid
 - Valori în intervalul $[0, 1]$
 - Neliniară, derivata ei e ușor de calculat
- Funcția tanh
 - Centrată în 0, valori între $[-1, 1]$
 - Utilizată pentru clasificare binară
- Funcția ReLU (Rectified Linear Unit)
 - Cea mai utilizată
 - Nu necesită operații complexe
 - Introduce o non-linearitate

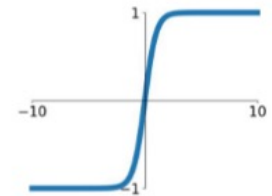
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



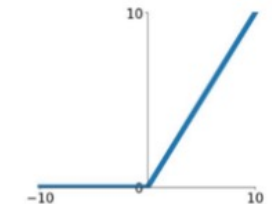
tanh

$$\tanh(x)$$



ReLU

$$\max(0, x)$$





Rețele neuronale convoluționale

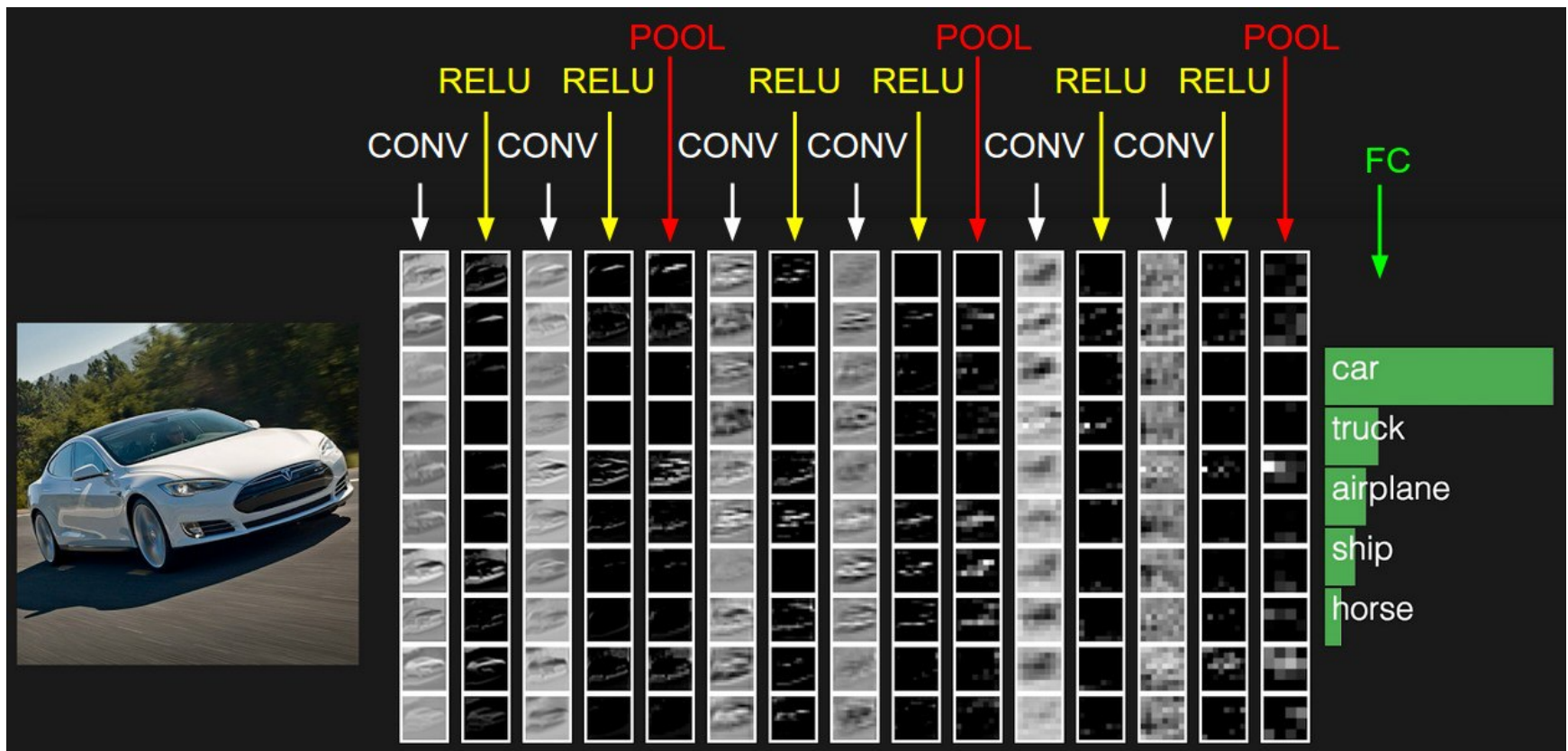
Nivelurile rețelei

- Date 3 dimensionale: în cazul imaginilor: înălțime, lățime și număr de canale (image grayscale – 1 canal, imagine color 3 canale)
- Trei tipuri de niveluri (*layer*)
 - nivel *convoluțional*
 - nivel de *pooling*
 - nivel complet conectat (*fully connected*)
- Dimensiunea spațială a nivelurilor (*width, height*) scade pe măsură ce avansăm în rețea, dar adâncimea crește odată cu avansarea în rețea
 - primele niveluri din rețea extrag trăsături cu informație semantică redusă (muchii, colțuri), iar nivelurile mai mari codifică trăsături mult mai complexe (de exemplu, roată, parbriz etc.)



Rețele neuronale convoluționale

Nivelurile rețelei



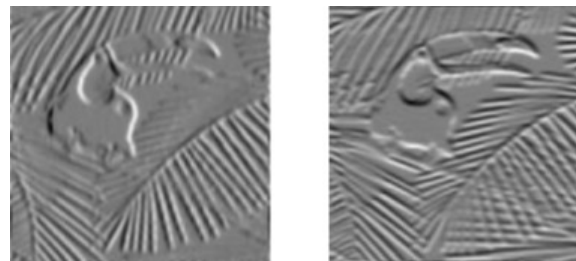
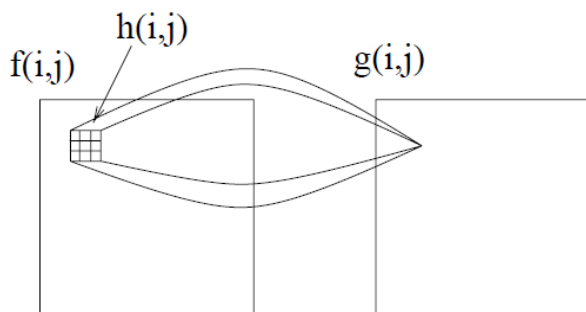
sursa imaginii: <http://cs231n.github.io/convolutional-networks/>



Rețele neuronale convoluționale

Neuronii și convoluția

- Convoluția produce o imagine rezultat unde fiecare pixel este o combinație liniară a unei vecinătăți din imaginea sursă
- În funcție de coeficienții nucleului de convoluție, putem obține filtre care au efect de netezire, sau pot pune în evidență anumite trăsături
- Neuronul artificial se bazează tot pe o combinație liniară a intrărilor





Rețele neuronale convoluționale

Neuronii și convoluția

- Convoluția se aplică în trei dimensiuni (width, height, depth)
- Este operația clasică de convoluție, dar rețeaua **ÎNVAȚĂ** ponderile pentru fiecare filtru
- Parametri:
 - Dimensiunea filtrului
 - Pasul (Stride)
 - Umplere (Padding)
 - Numărul de canale de ieșire



Exemple de filtre învățate de primul nivel al unei rețele neuronale convoluționale



Nivelul convoluțional

- Dimensiunea ieșirii unui filtru convoluțional:

$$sz_{n+1} = \frac{sz_n - f + 2p}{s} + 1$$

unde s – stride-ul operației de convoluție, f – dimensiunea filtrului de convoluție, p – valoarea pentru padding, sz_i – dimensiunea intrării de la nivelul i

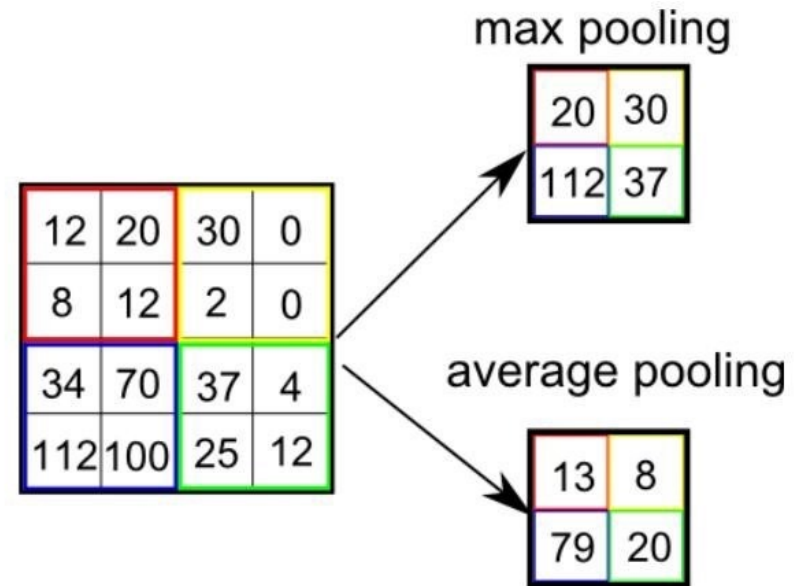
- În practică se utilizează două tipuri de *padding*:
 - **same** – se obține aceeași dimensiune a ieșirii ca a intrării ($p = ?$)
 - **valid** - *padding* = 0, dimensiunea ieșirii = ?



Rețele neuronale convoluționale

Nivelul de *pooling* (integrare, încorporare)

- se aplică individual pe fiecare canal al datelor de intrare
- **NU** are parametri care trebuie să fie învățați
- reduce dimensiunea spațială, deci și riscul de overfitting
- Cele mai comune filtre de tip *pooling*:
 - *max pooling* – selectează maximul
 - *average pooling* – selectează media

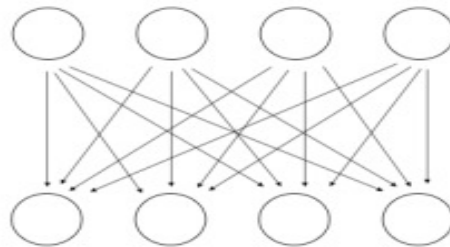




Rețele neuronale convoluționale

Nivelul complet conectat (fully connected - FC)

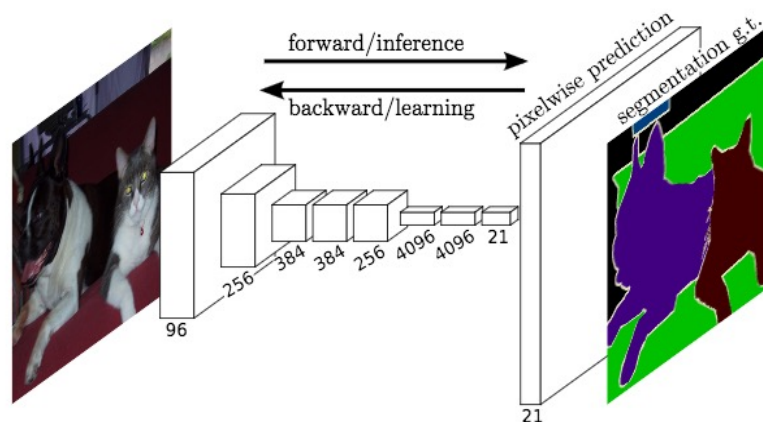
- Similar cu rețelele neuronale clasice, fiecare neuron din nivelul k este conectat la toți neuronii din nivelul $k+1$
- Folosit în ultimele niveluri din rețea pentru clasificare
- Problemă: sunt foarte multe ponderi de învățat, astfel încât sunt practice doar după de numărul de trăsături este redus
- Există rețele care nu mai folosesc niveluri FC, ci doar niveluri convoluționale.





Segmentare semantică

- Fiecărui pixel din imagine i se asociază o clasă
- Rețele neuronale complet convoluționale (*fully convolutional neural networks*)



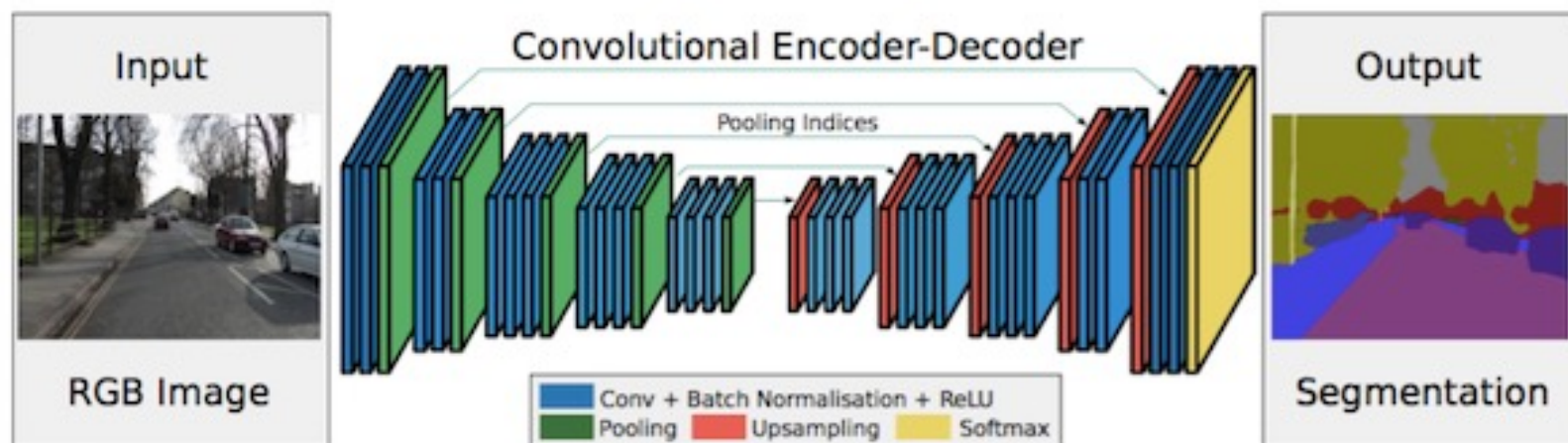


Segmentare semantică

- Segmentarea implică obținerea unei imagini de aceeași dimensiune cu imaginea de intrare în care fiecare pixel are asignată o anumită clasă
- Rețele convoluționale neuronale clasice scad dimensiunea spațială a hărților de trăsături pe măsură ce ne apropiem de rezultat
- Este necesar un mecanism de refacere a rezoluției inițiale (up-sampling)



Segmentare semantică



Sursa: Sasank Chilamkurthy, “A 2017 Guide to Semantic Segmentation with Deep Learning”,

<http://blog.qure.ai/notes/semantic-segmentation-deep-learning-review>



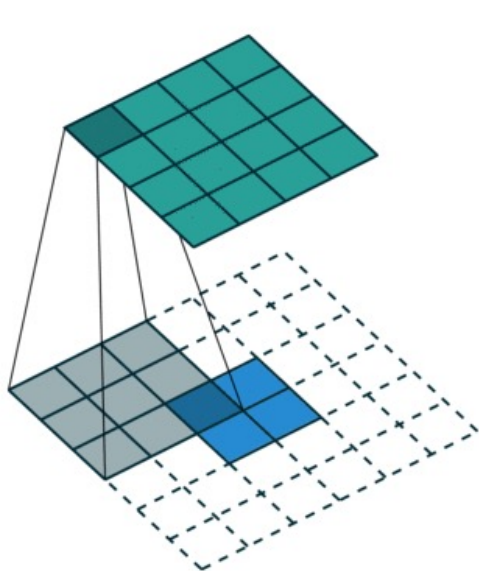
“Deconvoluție” – convoluție transpusă

- Se utilizează nuclee pentru operația de deconvoluție transpusă, cu parametri variabili, supuși procesului de învățare
- Din punct de vedere intuitiv, se proiectează nucleul în imaginea rezultat, ponderat cu valorile din imaginea sursă, și se însumează aceste proiecții
- Pasul de proiecție “stride” poate fi mai mare de 1, rezultând o imagine rezultat mai mare

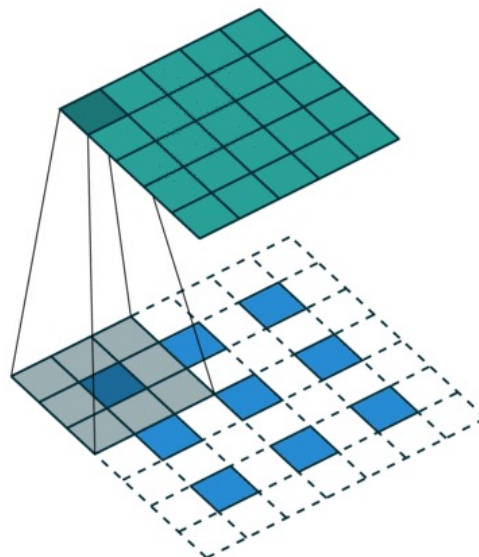


Segmentare semantică

“Deconvoluție” – convoluție transpusă



Pas (stride) = 1



Pas (stride) = 2

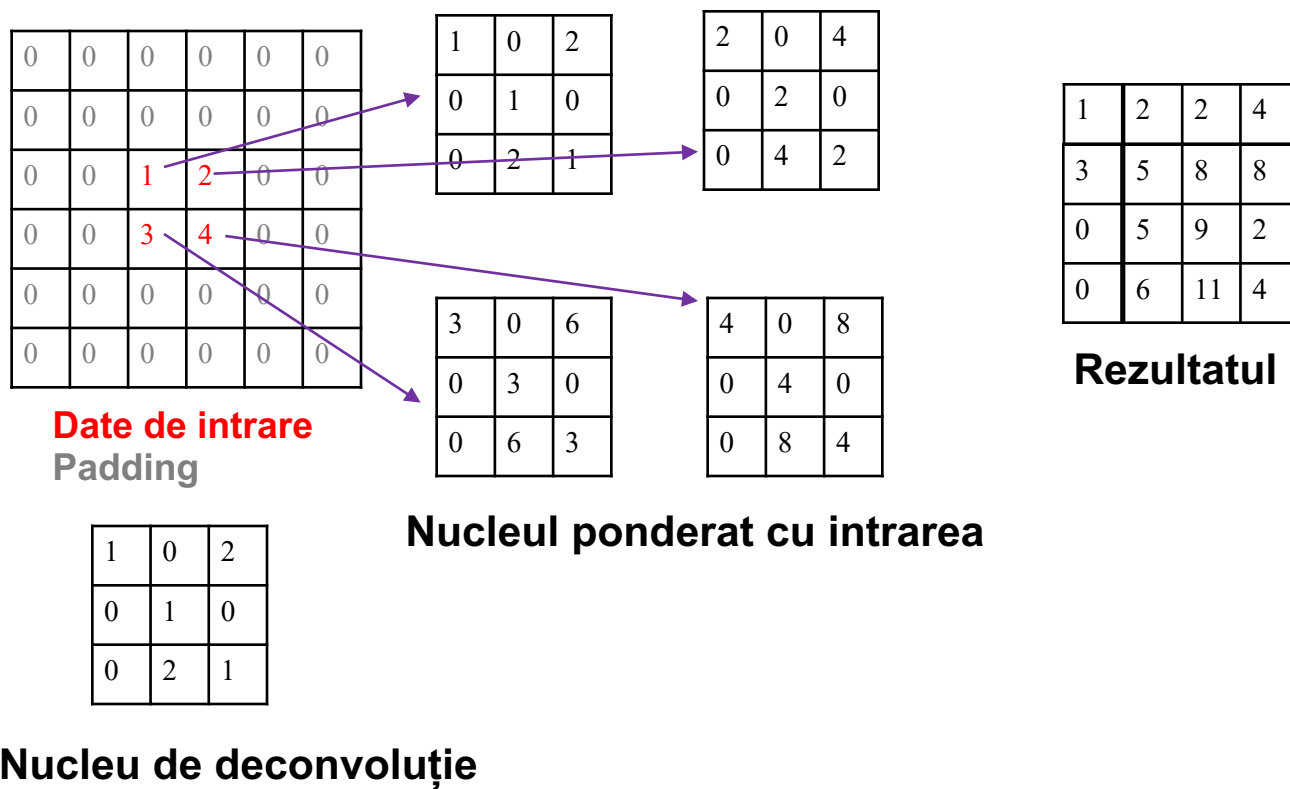
Sursa imaginii:

<https://datascience.stackexchange.com/questions/6107/what-are-deconvolutional-layers>



Segmentare sematică

“Deconvoluție” – convoluție transpusă





Deconvoluția prin convoluție

- Același rezultat se obține prin folosirea unei convoluții cu nucleul oglindit pe ambele axe:

0	0	0	0	0	0
0	0	0	0	0	0
0	0	1	2	0	0
0	0	3	4	0	0
0	0	0	0	0	0
0	0	0	0	0	0

*

1	2	0
0	1	0
2	0	1

=

0	0	0	0	0	0
0	1	2	2	4	0
0	3	5	8	8	0
0	0	5	9	2	0
0	0	6	11	4	0
0	0	0	0	0	0

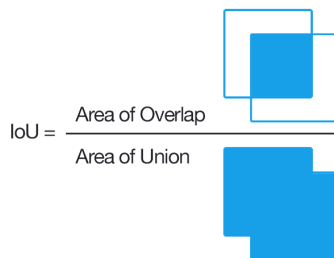
- Termeni improprii: deconvoluția CNN nu e nici deconvoluție (operația inversă convoluției), nici transpusă (e oglindită)



Segmentare semantică

Funcția de pierdere (loss function)

- Cum se evaluează numeric segmentarea calculată de rețea?
 - Indexul Jaccard sau IoU (Intersection over Union)



- Acuratețea bazată pe pixeli (*pixel accuracy*)

$$acc = \frac{TP + TN}{TP + TN + FP + FN}$$

TP, TN, FP, FN se calculează pe baza pixelilor segmentați

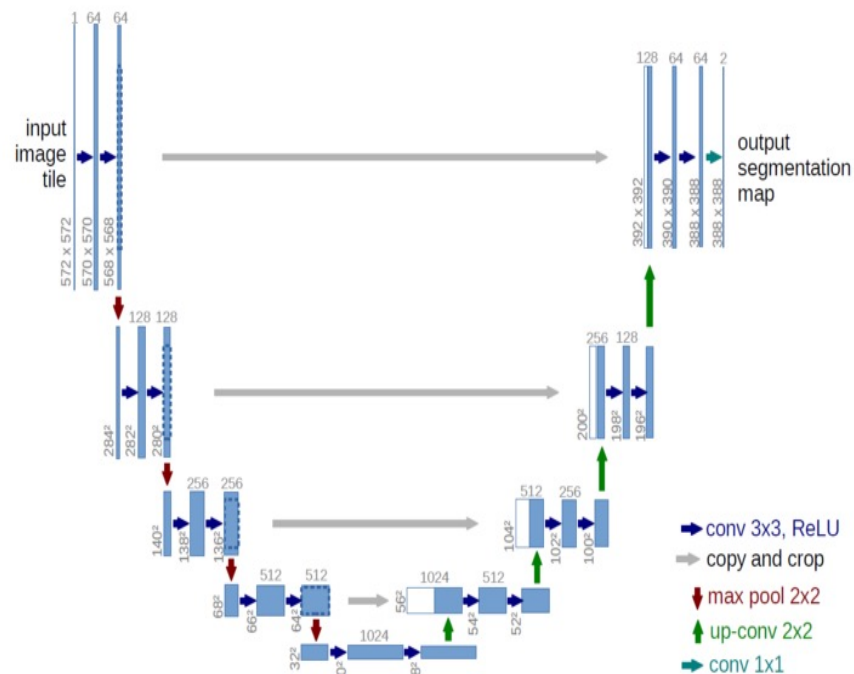


Segmentare semantică

Rețeaua U-Net

- Dezvoltată pentru segmentarea imaginilor în domeniul medical
- Există două căi în rețea:
 - Calea de contracție – extrage **contextul** din imaginea de intrare, exact ca o rețea convoluțională clasică
 - Calea de expansiune – simetrică cu cea de contracție, **extinde** trăsăturile prin deconvoluții pentru a clasifica precis fiecare pixel
- Legături între cele două căi (*skip connections*) – simplă concatenare a hărților de trăsături
 - Astfel se adaugă informație de localizare în ieșirea rețelei

2



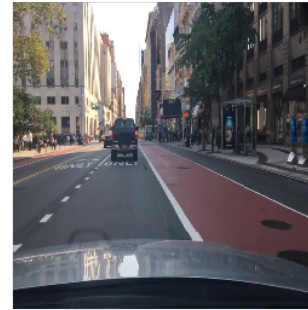
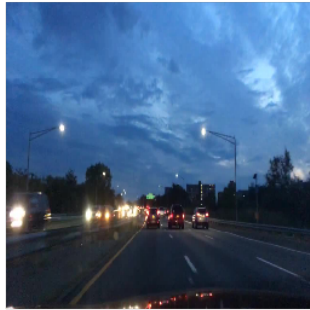
Olaf Ronneberger, Philipp Fischer, Thomas Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, LNCS, Vol.9351: 234--241, 2015



Segmentare semantică

Antrenarea rețelei

- Se utilizează perechi de imagini (image color sursa + image etichetată (adnotată))



- Deoarece adnotarea imaginilor pentru segmentare este mult mai dificilă decât pentru clasificare, se folosesc de obicei baze de date publice.

F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, T. Darrell, “BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling”, arXiv: 1805.04687, 2018.



Antrenarea rețelei

- Se calculează ieșirile rețelei pentru imaginile de antrenare, și se compară cu rezultatele dorite (imaginile adnotate, “Ground Truth”).
- Se calculează funcții de cost (de pierdere, “loss”), cum ar fi “Intersection over Union”, sau eroarea pătratică medie
- Ponderile neuronilor sunt ajustate în funcție de eroare și de impactul ponderilor asupra ei, folosind metoda Backpropagation, o particularizare a metodei generale Gradient Descent.

$$x^{(k+1)} = x^{(k)} - \gamma \nabla f(x^{(k)})$$



Segmentare semantică

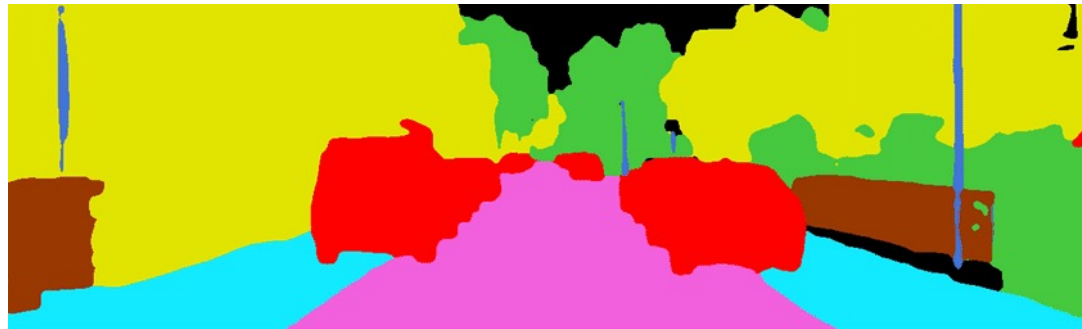
Comparație cu segmentarea clasică



Imagine inițială



Segmentare color



Segmentare semantică

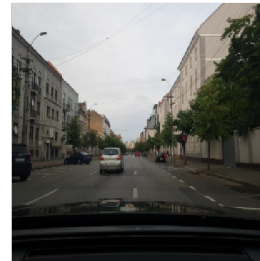
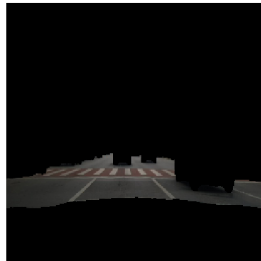
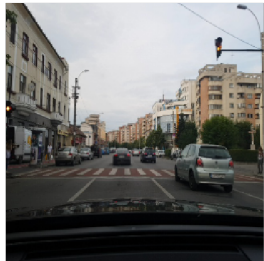
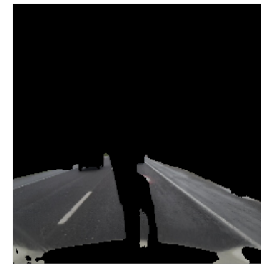
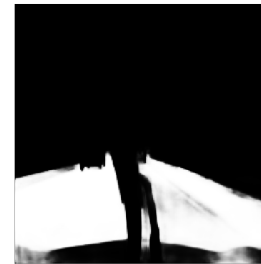
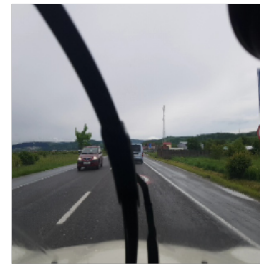
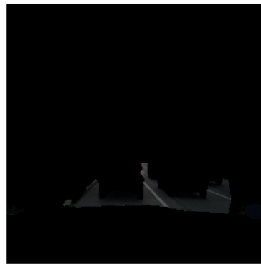
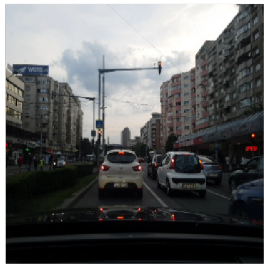
https://www.researchgate.net/publication/326875064_Towards_a_Meaningful_3D_Map_Using_a_3D_Lidar_and_a_Camera



Segmentare semantică

Alte exemple

- Segmentare drum, folosind rețeaua U-Net antrenată pe baze de date publice, și testată pe imagini din Cluj-Napoca
- Capacitate de generalizare, dovada unei învățări corecte!





- Performanța unei rețele este dată de arhitectura ei, dar și de cantitatea și calitatea datelor de antrenare.
- De exemplu, rețeaua folosită pentru segmentarea imaginilor din trafic are **93000000** de parametri care trebuie reglați
- Au fost folosite 10000 de imagini pentru antrenare
- Timp de antrenare: 1h 30 min, pe un sistem echipat cu două plăci grafice nVidia 1080 Ti
- Timp de predicție (segmentare):
 - 15 ms folosind plăcile grafice (GPU)
 - 410 milisecunde folosind doar CPU (Intel i7 6700K)



Resurse suplimentare

- TensorFlow – Platformă open source pentru inteligență artificială <https://www.tensorflow.org/>
- Keras: The Python Deep Learning library – API de nivel înalt pentru dezvoltarea rețelelor neuronale artificiale (folosește engine-ul TensorFlow) <https://keras.io/>
- Dlib – pachet de unelte C++ pentru inteligență artificială și viziune computerizată <http://dlib.net/>