Învățare Automată în Arta Vizuală

Curs 3: Introducere În Rețele Neurale Convoluționale (CNN / ConvNet)

Studiu de caz

- dorim un model pentru detecția lebedelor în imagini
- caracteristici specifice lebedelor:
  - gât lung
  - cioc portocaliu
  - culoarea albă
  - etc
- soluție (naivă) - detectăm aceste caracteristici în imagini. Simplu, problemă rezolvată ʘ‿ʘ

Studiu de caz

Sursa foto:
https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

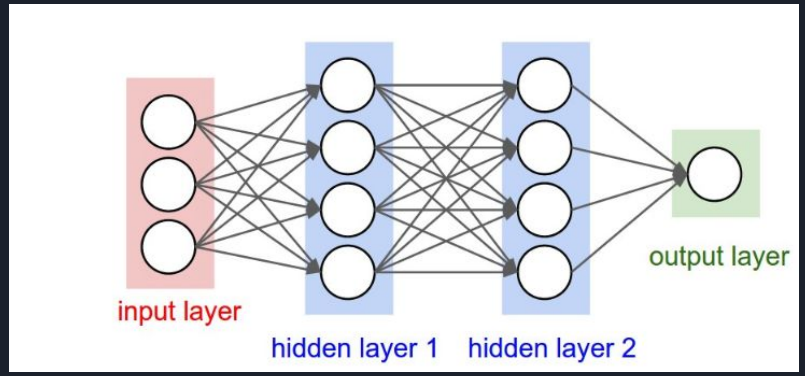
Studiu de caz

Sursă foto:
https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

Studiu de caz

- tehnici pentru rezolvarea acestor probleme - *feature detectors*: SIFT, FAST, SURF, BRIEF, etc
- problemă: descriptorii sunt fie prea simpli (prea generali), fie prea complecși (generalizează greu)
- soluție: ce ar fi dacă modelul ar învăța ce caracteristici să detecteze în imagini?
  - nesupervizat: K-means, PCA
  - supervizat: rețele neurale
    - tradiționale (fully-connected)
    - convoluționale

Rețele Neurale Tradiționale (fully-connected)



Sursă foto: http://cs231n.github.io/convolutional-networks/

- primesc ca input un vector, pe care îl transformă printr-o serie de *hidden layers*
- fiecare strat este alcătuit dintr-un set de neuroni, în care fiecare neuron este conectat la toți neuronii din stratul anterior
- neuronii dintr-un strat sunt complet independenți și nu au conexiuni comune
- ultimul strat se numește `output`, iar în problemele de clasificare reprezintă scorurile claselor

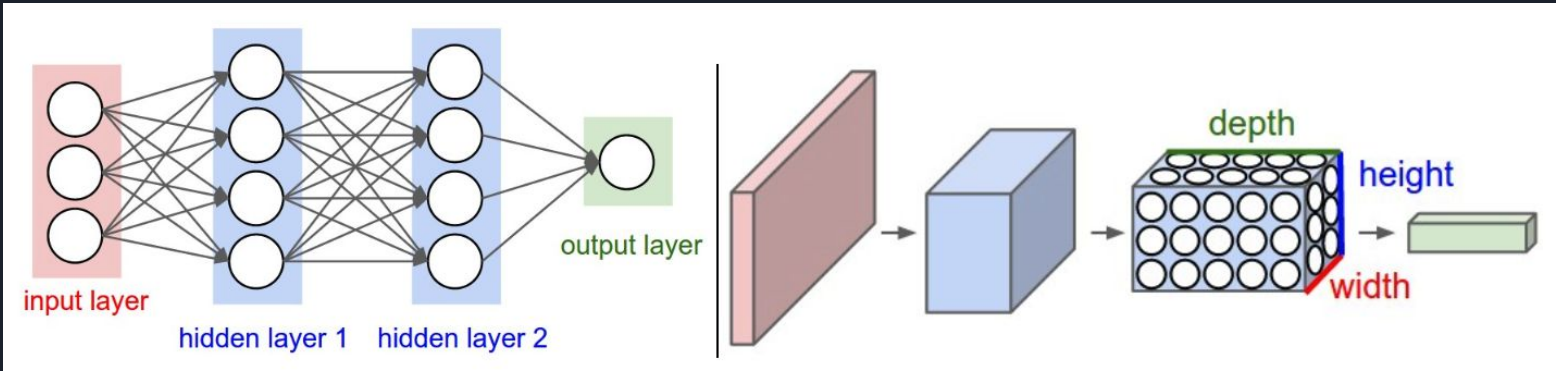
Rețele Neurale Tradiționale (fully-connected)

- probleme:
  - nu scalează cu dimensiunea imaginilor
    - în MNIST, dimensiunea imaginilor e de doar 28x28x1 (grayscale), astfel că un singur neuron de pe primul hidden layer al unei rețele neurale fully-connected are 28*28*1 = 784 parametri - fezabil, dar clar structura aceasta fully-connected nu scalează pentru imagini mai mari
    - pentru o imagine de dimensiune 224x224x3 (rgb), neuronii ar avea 224*224*3 = 150,528 parametri, iar numărul mare de parametri ar duce rapid la overfitting.

Rețele Neurale Tradiționale (fully-connected)

- probleme:
  - reacționează diferit față de o imagine și versiunea translatată a acesteia, adică nu sunt invariante la translație (de exemplu, dacă imaginea unei pisici apare în colțul din stânga într-o poză și apare în colțul din dreapta în altă poză, rețeaua va încerca să se corecteze, crezând că pisica va apărea întotdeauna în colțul din dreapta)
  - informația spațială se pierde când imaginile sunt vectorizate pentru rețea

Rețele Neurale Convoluționale



Sursă foto: http://cs231n.github.io/convolutional-networks/

- asemănări cu rețelele tradiționale:
  - sunt alcătuite din neuroni care au parametri și bias învățabili
  - fiecare neuron primește un input, efectuează un produs scalar și opțional aplică o funcție neliniară
  - au o funcție de cost pe ultimul strat și conceptele de bază ale rețelelor tradiționale se aplică în continuare

Rețele Neurale Convoluționale

- diferențe față de rețelele tradiționale:
  - fac presupunerea explicită că input-urile sunt imagini, ceea ce ne permite să introducem anumite proprietăți în arhitectura rețelei, astfel că se reduce foarte mult numărul de parametri, iar funcția de feed-forward este mai eficientă
  - neuronii straturilor sunt aranjați în 3 dimensiuni (lățime, înălțime, adâncime) și sunt conectați doar cu o mică regiune din stratul anterior (față de toți neuronii, cum e cazul în rețelele tradiționale)

Rețele Neurale Convoluționale - tipuri de straturi

- trei tipuri principale de straturi pentru a construi arhitecturi convoluționale: *Convolutional*, *Pooling* și *Fully-Connected*
- exemplu de rețea convoluțională: [INPUT - CONV - RELU - POOL - FC]
  - INPUT [32x32x3] - conține valorile pixelilor imaginilor, în cazul de față imagini de dimensiune 32x32, cu 3 canale de culori R, G, B
  - CONV - calculează outputul neuronilor conectați la regiuni locale din input, fiecare efectuând un produs scalar între proprii parametri și regiunea din input la care sunt conectați, rezultând un output de dimensiune [32x32x12], de exemplu, dacă s-a decis folosirea a 12 filtre de convoluție
  - RELU - aplică o funcție de activare, *max(0, x)*, pe fiecare element în parte, lăsând dimensiunea volumului neschimbată ([32x32x12])

Rețele Neurale Convoluționale - tipuri de straturi

- POOL - reduce dimensiunea volumului de-a lungul axelor spațiale (lățime, înălțime), rezultând, de exemplu, un volum de forma [16x16x12]
- FC - calculează scorurile claselor, rezultând un volum de forma [1x1xnum_classes], unde fiecare număr corespunde scorului unei clase, iar fiecare neuron este conectat cu toți neuronii din volumul anterior

Rețele Neurale Convoluționale - sumar

- o rețea convoluțională reprezintă, la bază, o listă de straturi care transformă un volum de input (imagine) într-un volum de output (conținând scorurile claselor)
- există câteva tipuri principale de straturi (CONV/FC/RELU/POOL sunt cele mai populare)
- fiecare strat acceptă un volum 3D ca input, pe care-l transformă într-un volum 3D de output printr-o funcție diferențiabilă
- fiecare layer poate avea, sau nu, parametri învățabili (de exemplu, CONV/FC au parametri învățabili, pe când RELU/POOL nu au)
- fiecare layer poate avea, sau nu, hiperparametri adiționali (de exemplu, CONV/FC/POOL au, RELU nu are)