



UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

SPECIALIZAREA INTELIGENȚĂ ARTIFICIALĂ

Îmbunătățirea unui sistem de recomandare

---

LUCRARE DE DISERTAȚIE

COORDONATOR ȘTIINȚIFIC

CONF. DR. BOGDAN ALEXE

ABSOLVENT

ADRIAN ISPAS

BUCUREȘTI, ROMÂNIA

IUNIE 2019



# Abstract

Abstractul în limba română.

# Abstract

Abstractul în limba engleză.

# Cuprins

<b>Listă de figuri</b>	<b>6</b>
<b>Listă de tabele</b>	<b>7</b>
<b>1 Introducere</b>	<b>9</b>
1.1 Motivație . . . . .	9
1.2 Obiective propuse . . . . .	10
1.3 Structura lucrării . . . . .	10
<b>2 Fundamente teoretice</b>	<b>12</b>
2.1 Sisteme de recomandare . . . . .	12
2.1.1 Noțiuni generale . . . . .	12
2.1.2 Strategii de recomandare . . . . .	12
2.1.3 Funcții de loss . . . . .	14
2.2 Rețele neurale convoluționale . . . . .	18
2.2.1 Noțiuni generale . . . . .	18
2.2.2 VGG . . . . .	20
2.2.3 InceptionV3 . . . . .	20
2.2.4 ResNet . . . . .	20
2.2.5 NASNet . . . . .	20
2.3 Clustere . . . . .	20
2.3.1 Noțiuni generale . . . . .	20
2.3.2 K-nearest neighbors . . . . .	20
<b>Bibliografie</b>	<b>21</b>

# Listă de figuri

2.1	Filtrarea colaborativă . . . . .	13
2.2	Filtrarea bazată pe conținut . . . . .	14
2.3	Matricea de interacțiuni . . . . .	15
2.4	Setul de antrenare . . . . .	16
2.5	Procedura de învățarea BPR . . . . .	16
2.6	Online WARP Loss Optimization . . . . .	18
2.7	Exemplu rețea convoluțională . . . . .	19
2.8	Exemplu de filtru aplicat peste input . . . . .	19
2.9	Exemplu de pooling . . . . .	20

## Listă de tabele





# Capitolul 1

## Introducere

### 1.1 Motivație

Volumul de date crește semnificativ de la an la an astfel până în 2020 se estimează că pentru fiecare persoană de pe planetă vor fi creați în fiecare secundă 1.7 MB de date, ceea ce înseamnă peste 13 milioane de GB creați în fiecare secundă în lume.

În 2018 în fiecare minut se vizionau peste 97 de mii de ore de conținut pe Netflix. Peste 4.3 milioane de videoclipuri erau vizionate pe Youtube. Pe Spotify se ascultau 750 de mii de melodii, iar Amazon pregătea peste o mie de pachete [1].

În România, Netflix pune la dispoziție 575 de filme și 208 seriale. În Regatul Unit sunt disponibile 2425 de filme și 542 de seriale, iar în Statele Unite Ale Americii sunt disponibile 2942 de filme și 629 de seriale [2].

Amazon oferă cumpărătorilor o gamă cu un total de peste 119 milioane de produse, dintre care 44.2 milioane de cărți, 10.1 milioane de electronice sau 4.5 milioane de produse realizate manual [3].

În primă fază, cu cât volumul de date pus la dispoziție de o platforma este mai mare cu atât este mai mare și necesitatea unui sistem de recomandare care să vină în ajutorul utilizatorului final pentru a explora gama de produse oferită de respectiva platformă. Ulterior, acel sistem de recomandare se vrea a fi îmbunătățit astfel încât să ofere fiecărui utilizator o experiență cât mai personalizată prin care se recomande, în cazul platformelor de streaming video, conținut relevant pentru a fi consumat de utilizatorul final, sau în cazul platformelor de ecommerce, produse pe care utilizatorul ar fi dispus să le cumpere.

În majoritatea cazurilor sistemele de recomandare se bazează pe metadatele utilizatorilor, precum: regiunea, vârsta, genul, ce alte produse a accesat sau cumpărat și metadatele produselor: categoria din care face parte, ratingul acestuia. La acestea se pot adăuga și alte informații precum: ce alte produse a apreciat un alt user cu un profil asemănător.

## 1.2 Obiective propuse

În majoritatea cazurilor primul contact pe care îl avem cu un clip de pe Youtube, cu un film sau serial de pe Netflix sau un produs de pe Amazon este contactul vizual cu imaginea de prezentare a acelui produs.

Astfel, prezenta lucrare de disertație are drept obiectiv principal introducerea în sistemul de recomandare de informații vizuale extrase din imaginile de prezentare ale produselor. Informațiile vizuale sunt reprezentate de clusterelor create peste imaginile asociate produselor. Fiecare produs are o imagine de prezentare, iar fiecare imagine are un cluster caruia îi aparține din intervalul  $[1, N]$  unde  $N$  este corelat cu numărul de categorii de produse din baza de date pe care se execută optimizarea.  $N$  poate fi ales și pe baza altor raționamente.

Scopul final al acestei abordări fiind acela de a observa evoluția metricilor de evaluare, în cazul nostru acuratețea și precizia@k, atunci când informația vizuală este introdusă într-un sistem de recomandare, fiind singura informație prezentă exceptând matricea de interacțiuni, dar și cum se comportă un sistem de recomandare când primește această informație împreună cu alte informații, spre exemplu categoria unui articol.

## 1.3 Structura lucrării

Prezenta lucrare de disertație începe prin detalierea fundamentelor teoretice în capitolul II unde sunt prezentate teoriile ce stau la baza realizării acestei lucrări.



# Capitolul 2

## Fundamente teoretice

### 2.1 Sisteme de recomandare

#### 2.1.1 Noțiuni generale

Sistemele de recomandare au scopul de oferi sugestii de articole utilizatorilor unei platforme pe baza unor strategii. Un sistem de recomandare poate folosi una sau mai multe strategii de recomandare.

În cazul în care se folosesc cel puțin două strategii, sistemul de recomandare devine un sistem de recomandare hibrid. Prin folosirea mai multor strategii se urmărește ca fiecare strategie să vină în completarea celorlalte cu avantajele sale.

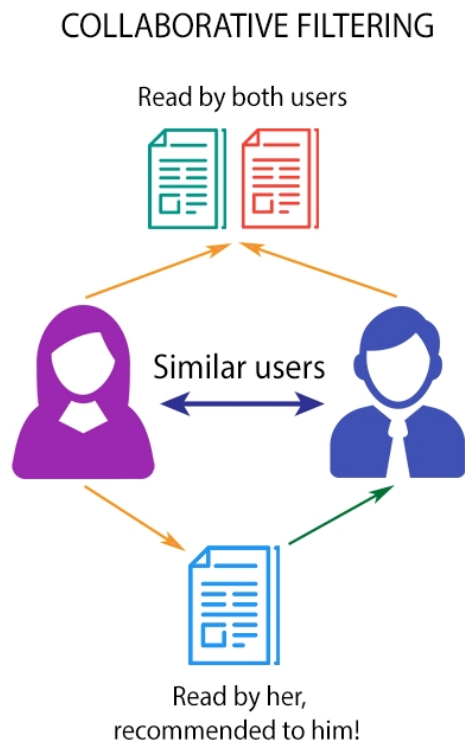
De cele mai multe ori, în implementarea unui sistem de recomandare, se folosește tehnica de filtrare colaborativă împreună cu o altă strategie de recomandare [4].

#### 2.1.2 Strategii de recomandare

##### **Filtrarea colaborativă**

Filtrarea colaborativă se bazează pe faptul că utilizatorii care au în prezent preferințe similare vor avea și în viitor preferințe destul de similare. Această abordare folosește ratingurile pe care le dau utilizatorii sau oricare altă formă de a da un feedback, îmi place/nu îmi place, pentru a identifica preferințele comune dintre grupurile de utilizatori. Odată identificate preferințele se generează recomandări pe baza similarităților dintre utilizatori.

Dezavantajul acestei strategii apare în momentul în care în sistem intră un nou utilizator. Datorită faptului că utilizatorul este nou, sistemul nu are un istoric al preferințelor lui, iar în consecință nu îl poate asigura unui grup de utilizatori pe baza preferințelor [4].



**Figura 2.1:** Filtrarea colaborativă. Imagine preluată din [5].

### Filtrarea bazată pe conținut

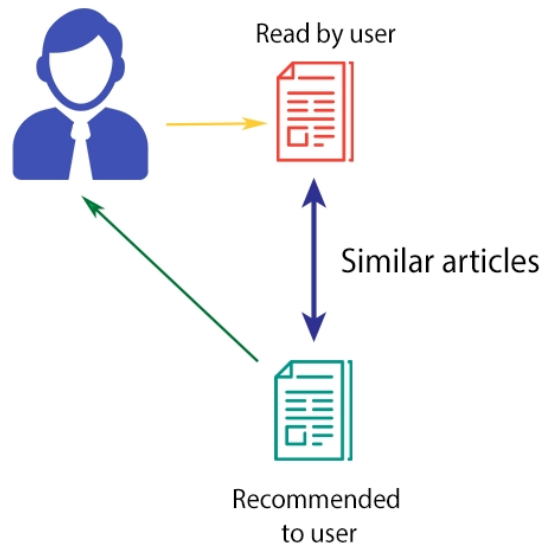
Filtrarea bazată pe conținut pleacă de la premisa că utilizatorii cărora le-au plăcut articole definite de anumite atribute în trecut, vor aprecia aceleași tip de articole și în viitor. Această abordare folosește atributele articolelor pentru a le compara cu profilul utilizatorilor și a oferi recomandări. Calitatea recomandărilor create folosind această strategie este influențată de setul de atribute ales pentru articole.

Similar cu filtrarea colaborativă și filtrarea bazată pe conținut prezintă dezavantaje în momentul în care în sistem intră un nou utilizator fără istoric [4].

### Filtrarea demografică

Filtrarea demografică folosește atribute precum vârsta, genul, educația, etc. pentru a identifica categoriile de utilizatori. Nu prezintă dezavantaje atunci când apar noi utili-

## CONTENT-BASED FILTERING



**Figura 2.2:** Filtrarea bazată pe conținut. Imagine preluată din [5].

zatori în sistem și nu se folosește de ratinguri, sau alt sistem de feedback, pentru a face recomandări.

Dezavantajul este reprezentat de faptul că procesul de colectare al datelor demografice poate fi îngreunat de legislație fapt ce reprezintă o limitare a acestei metode [4].

### Filtrarea bazată pe cunoștințe

Filtrarea bazată pe cunoștințe folosește cunoștințele despre utilizatori și articole pentru a spune ce articole îndeplinesc cerințele utilizatorilor și generează recomandări în consecință. Filtrare bazată pe cunoștințe are la bază constrângeri și este capabilă să recomande chiar și articole complexe care nu sunt cumpărate atât de des, precum mașini sau case [4].

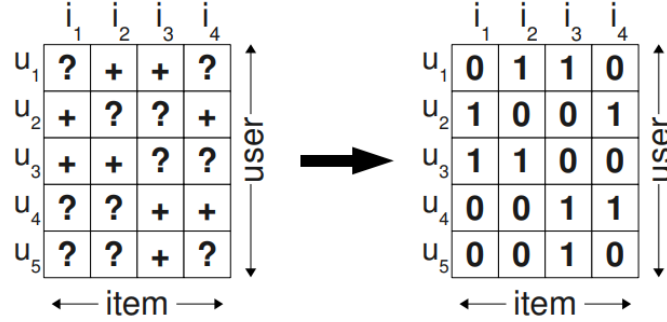
### 2.1.3 Funcții de loss

#### BPR: Bayesian Personalised Ranking

Este o metodă ce se bazează pe feedback implicit (click-uri, ratinguri, achiziții). Există multe metode ce se bazează pe acest feedback implicit, precum matrix factorization (MF),

k-nearest-neighbor (kNN), însă acestea nu sunt optimizate pentru ranguri. Metoda de învățare este bazată pe gradientul descendent. Metoda este recomandată atunci când se dorește optimizarea acurateții.

Definim în continuare  $U$  ca fiind mulțimea de utilizatori și  $I$  ca fiind mulțimea de articole. Feedback-ul implicit este reprezentat de mulțimea  $S \subseteq U \times I$ . De asemenea, definim  $I_u^+ := i \in I : (u, i) \in S$  și  $U_i^+ := u \in U : (u, i) \in S$ .



**Figura 2.3:** Matricea de interacțiuni, mulțimea  $S$ . Imagine preluată din [9].

O abordare uzuală pentru recomandarea de articole este să fie prezis scorul  $\hat{x}_{ui}$  care să reflecte preferința utilizatorului  $u$  pentru articolul  $i$ . Apoi fiecare articol primește un rang după sortarea scorurilor.

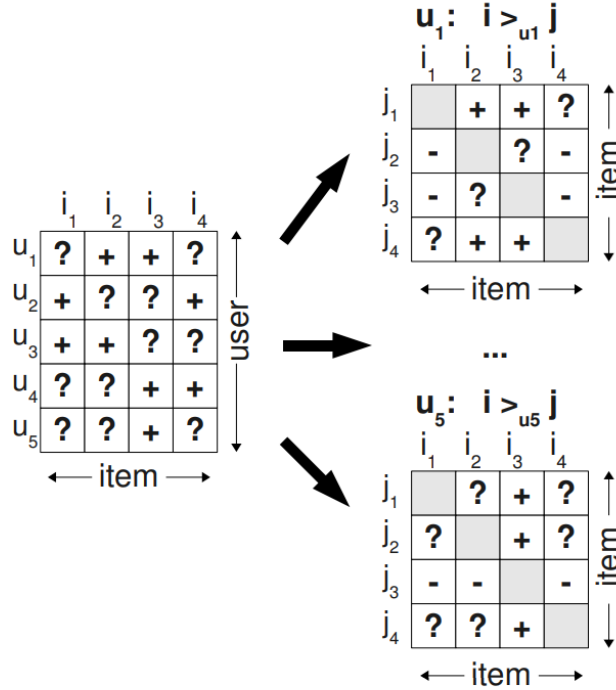
Setul de antrenare este definit de mulțimea  $D_S := \{(u, i, j) | i \in I_u^+ \wedge j \in I \setminus I_u^+\}$  unde  $(u, i, j)$  înseamnă că utilizatorul  $u$  preferă articolul  $i$  în detrimentul articolului  $j$ .

Criteriul de optimizare pentru pentru rangurile personalizate este definit după cum urmează:

$$BPR - OPT := \sum_{(u, i, j) \in D_S} \ln \sigma(\hat{x}_{uij}) - \lambda_{\Theta} \|\Theta\|^2 \quad (2.1)$$

unde  $\sigma$  este funcția sigmoid,  $\sigma(x) := \frac{1}{1+e^{-x}}$ ,  $\Theta$  reprezintă vectorul parametru al modelului care definește interacțiunea dintre utilizatorul  $u$ , articolul  $i$  și articolul  $j$ , iar  $\lambda_{\Theta}$  reprezintă parametrii de regularizare.

Cu aceste definiții putem defini și procedura de învățare a  $BPR$  după cum urmează.



**Figura 2.4:** Setul de antrenare. + reprezintă articolele  $i$  pe care utilizatorul le preferă în locul articolelor  $j$ , – utilizatorul preferă articolele  $j$  în loc de  $i$ , iar ? reprezintă lipsa informației despre acea interacțiune. Imagine preluată din [9].

```

1: procedure LEARNBPR( $D_S, \Theta$ )
2:   initialize  $\Theta$ 
3:   repeat
4:     draw  $(u, i, j)$  from  $D_S$ 
5:      $\Theta \leftarrow \Theta + \alpha \left( \frac{e^{-\hat{x}_{uij}}}{1+e^{-\hat{x}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{x}_{uij} + \lambda_{\Theta} \cdot \Theta \right)$ 
6:   until convergence
7:   return  $\hat{\Theta}$ 
8: end procedure

```

**Figura 2.5:** Optimizarea modelului bazată metoda gradientului descendent cu parametrul de învățare  $\alpha$  și regularizarea  $\lambda_{\Theta}$ . Imagine preluată din [9].

### WARP: Weighted Approximate-Rank

Această metodă își are originile în procesarea imaginilor și anume pentru un set de reprezentări ale unor imagini  $x \in R^d$  și pentru un set de reprezentări ale unor adnotări  $i \in \Upsilon = \{1, \dots, Y\}$  - indici într-un dicționar cu posibile adnotări, metoda învață să mapeze imagini din spațiul reprezentărilor într-un spațiu comun  $R^D$



$$\Phi_I(x) : R^d \rightarrow R^D \quad (2.2)$$

în același timp învățând și mapări pentru adnotări în același spațiu

$$\Phi_W(i) : 1, \dots, Y \rightarrow R^D \quad (2.3)$$

Scopul principal fiind acela de a oferi ranguri posibilelor adnotări pentru o imagine dată astfel încât cel mai mare rang să descrie cel mai bine conținutul semnificativ al imaginii.

Modelul folosit este următorul:

$$f_i(x) = \Phi_W(i)^T \Phi_I(x) \quad (2.4)$$

Metoda învață să producă ranguri optimizate pentru primele adnotări din listă, ceea ce înseamnă că optimizează precizia@k.

În ceea ce privește funcția de eroare definim:  $f(x) \in R^Y$  ce produce un scor pentru fiecare etichetă și unde  $f_i(x)$  este valoarea etichetei  $i$ . Definim funcția de eroare pentru ranguri ca fiind:

$$err(f(x), y) = L(rank_y(f(x))) \quad (2.5)$$

unde  $rank_y(f(x))$  este rangul etichetei corecte data de  $f(x)$ :

$$rank_y(f(x)) = \sum_{i \neq y} I(f_i(x) \geq f_y(x)) \quad (2.6)$$

unde  $I$  este funcția indicator, iar  $L(\cdot)$  transformă rangul în penalizare

$$L(k) = \sum_{j=1}^k \alpha_j, \quad cu \quad \alpha_1 \geq \alpha_2 \geq \dots \geq 0. \quad (2.7)$$

$L(\cdot)$  poate lua diferite forme în funcție de ce se dorește a se optimiza:  $\alpha_j = \frac{1}{Y-1}$  optimizează rangul mediu,  $\alpha_j = 1$  și  $\alpha_{j>1} = 0$  optimizează proporția de ranguri corecte aflate în top, iar valorile mari ale lui  $\alpha$  optimizează primele  $k$  în lista de ranguri[8].

Cu definițiile prezentate mai sus putem descrie algoritmul acestei metode după cum urmează.

---

**Algorithm 1** Online WARP Loss Optimization

---

**Input:** labeled data  $(x_i, y_i), y_i \in \{1, \dots, Y\}$ .  
**repeat**  
  Pick a random labeled example  $(x_i, y_i)$   
  Let  $f_{y_i}(x_i) = \Phi_W(y_i)^\top \Phi_I(x_i)$   
  Set  $N = 0$ .  
  **repeat**  
    Pick a random annotation  $\bar{y} \in \{1, \dots, Y\} \setminus y_i$ .  
    Let  $f_{\bar{y}}(x_i) = \Phi_W(\bar{y})^\top \Phi_I(x_i)$   
     $N = N + 1$ .  
  **until**  $f_{\bar{y}}(x_i) > f_{y_i}(x_i) - 1$  or  $N \geq Y - 1$   
  **if**  $f_{\bar{y}}(x_i) > f_{y_i}(x_i) - 1$  **then**  
    Make a gradient step to minimize:  
       $L(\lfloor \frac{Y-1}{N} \rfloor) |1 - f_{y_i}(x_i) + f_{\bar{y}}(x_i)|_+$   
    Project weights to enforce constraints (2)-(3).  
  **end if**  
**until** validation error does not improve.

---

**Figura 2.6:** Online WARP Loss Optimization. Imagine preluată din [8].

## 2.2 Rețele neurale convoluționale

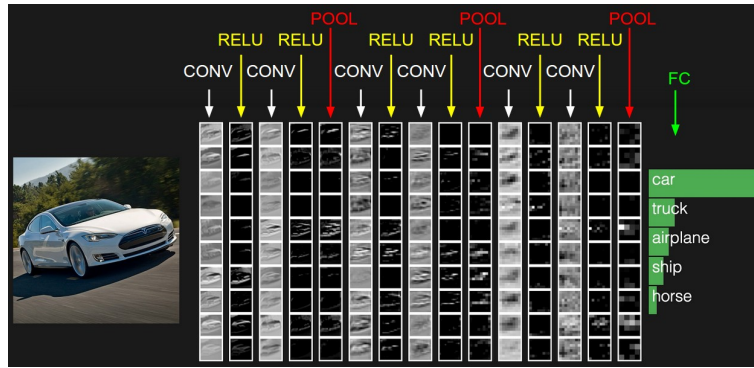
### 2.2.1 Noțiuni generale

Rețelele neurale convoluționale sunt foarte similare cu rețelele neurale fiind formate din neuroni ce învață ponderi ( $w$ ) și biasuri ( $b$ ). Scopul rețelei convoluționale este de a primi o imagine la input și de a scoate la output un scor pentru fiecare clasă ce corespunde imaginii.

Spre exemplu, la input se dă o imagine cu un autovehicul, iar rețeaua convoluțională poate spune că în imagine este o mașină în proporție de 80%, un camion în proporție de 10%, un avion în proporție de 6%, o barcă în proporție de 3% sau un cal în proporție de 1%.

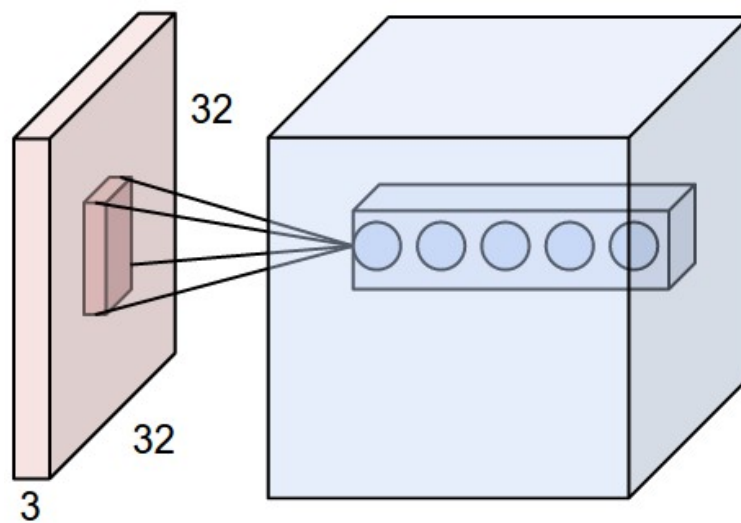
Rețelele convoluționale sunt compuse dintr-o secvență de layere ce poate fi împărțită în trei tipuri principale de layere [7]:

1. Convolutional Layer este layerul de bază într-o rețea convoluțională. Parametrii acestui layer sunt reprezentați de filtre învățabile, unde fiecare filtru reprezintă o mică bucată din imaginea de input. De exemplu, un filtru pentru acest layer poate avea dimensiunea de  $5 \times 5 \times 3$ , dimensiune ce reprezintă faptul că se iau 5 pixeli pe lățime și înălțime cu o adâncime de 3, unde adâncimea reprezintă canalele RGB. În continuare se glisează fiecare filtru peste input și se compune produsul



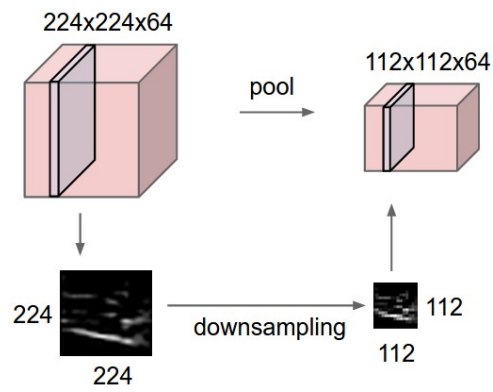
**Figura 2.7:** Exemplu de rețea convoluțională care primește la input o imagine și produce la output o listă de clase ce pot descrie imaginea de input. Imagine preluată din [7].

dintre filtre și input la fiecare poziție. În urma acestei operații se produce un vector de activare 2-dimensional care reprezintă răspunsul filtrului la fiecare poziție. Altfel spus, rețeaua va învăța filtre care se activează atunci când sunt prezente anumite tipuri de caracteristici, precum culoarea sau orientarea.

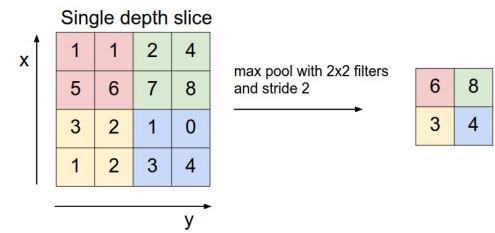


**Figura 2.8:** Exemplu de filtru aplicat peste input într-un layer convoluțional. Imagine preluată din [7].

2. Pooling Layer reprezintă o practică des folosită între mai multe layere convoluționale succesive. Această operație reduce numărul de parametri (dimensiunea), computațiile din rețea și controlează overfittingul. Se execută independent pe fiecare nivel al adâncimii unui input și pastrează valoarea maximă a acelei zone. Rezultatul este o zonă de caracteristici mai mică dar care păstrează cea mai relevantă statistică.
3. Fully-Connected Layer este stratul în care caracteristicile sunt vectorizate pentru a



(a) Reducerea dimensiunii.



(b) Filtrul de  $2 \times 2$  aplicat ce păstrează valoarea maximă.

**Figura 2.9:** Exemplu de pooling. Imagine preluată din [7].

putea fi folosite.

## 2.2.2 VGG

## 2.2.3 InceptionV3

## 2.2.4 ResNet

## 2.2.5 NASNet

## 2.3 Clustere

### 2.3.1 Noțiuni generale

### 2.3.2 K-nearest neighbors

# Bibliografie

- [1] Data never sleeps 6.0  
<https://www.domo.com/learn/data-never-sleeps-6>
- [2] Netflix International: What movies and TV shows can I watch, and where can I watch them?  
<https://www.finder.com/global-netflix-library-totals>
- [3] How Many Products Does Amazon Sell? – April 2019  
<https://www.scrapehero.com/number-of-products-on-amazon-april-2019/>
- [4] Erion Çano, Maurizio Morisio. *Hybrid Recommender Systems: A Systematic Literature Review*. Intelligent Data Analysis, vol. 21, no. 6, pp. 1487-1524, 2017
- [5] An Overview of Recommendation Systems  
<http://datameetsmedia.com/an-overview-of-recommendation-systems/>
- [6] LightFM 1.15 - documentation  
<http://lyst.github.io/lightfm/docs/lightfm.html>
- [7] CS231n: Convolutional Neural Networks for Visual Recognition  
<http://cs231n.stanford.edu/2018/syllabus.html>
- [8] Jason Weston, Samy Bengio, Nicolas Usunier. *Wsabie: Scaling up to large vocabulary image annotation*. IJCAI. Vol. 11. 2011.
- [9] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner and Lars Schmidt-Thieme. *BPR: Bayesian personalized ranking from implicit feedback*. Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press, 2009.
- [10] Karen Simonyan, Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. ICLR, 2015.

- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. *Deep Residual Learning for Image Recognition*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p770-778, 2016.
- [12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna. *Rethinking the Inception Architecture for Computer Vision*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p2818-2826, 2016.
- [13] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, Quoc V. Le. *Learning Transferable Architectures for Scalable Image Recognition*. IEEE/CVF Conference on Computer Vision and Pattern Recognition, p8697-8710, 2018.