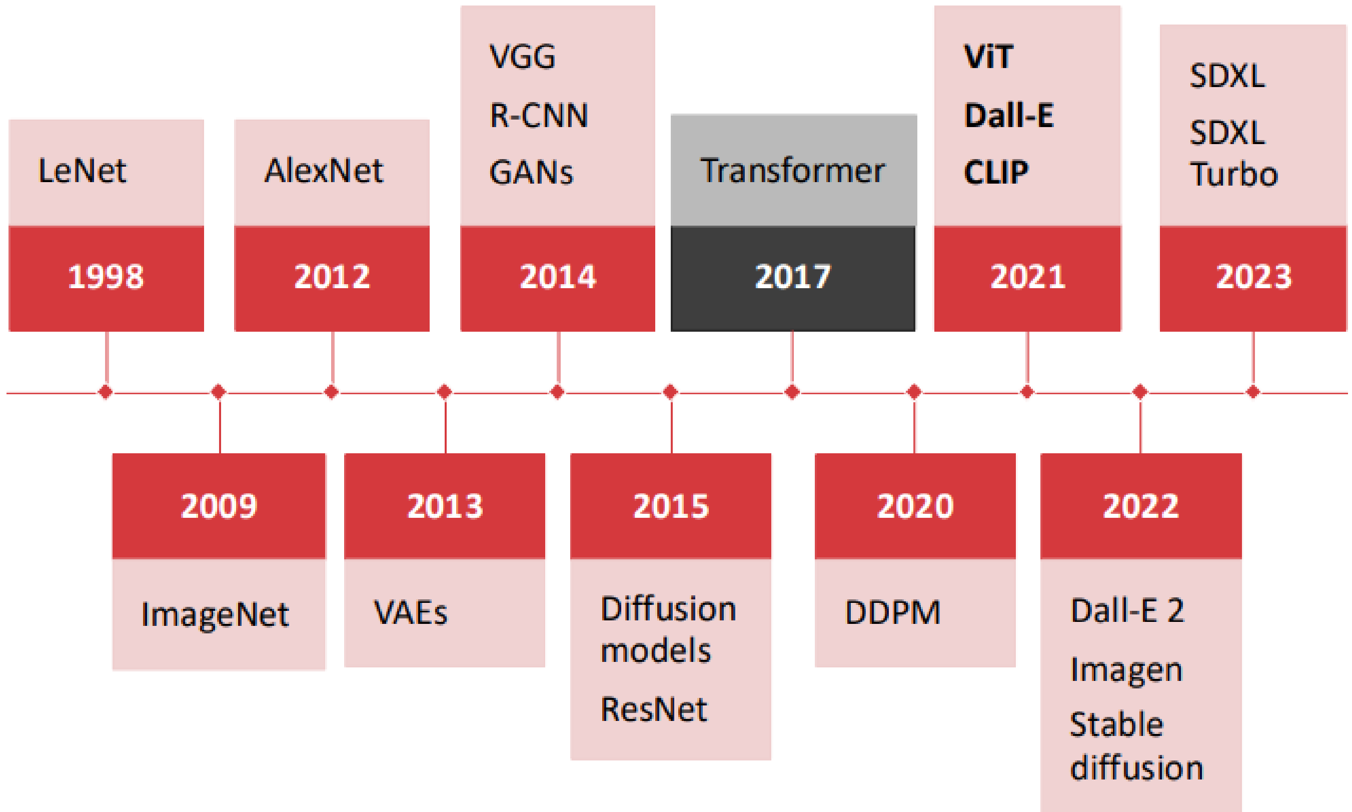
METODE INTELIGENTE DE REZOLVARE A PROBLEMELOR REALE

Laura Dioşan

Vision Transformers

Istoric al modelelor de CV

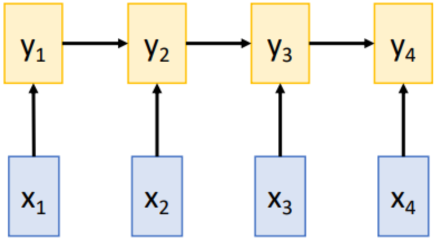


M I R P R

Procesarea unei secvente

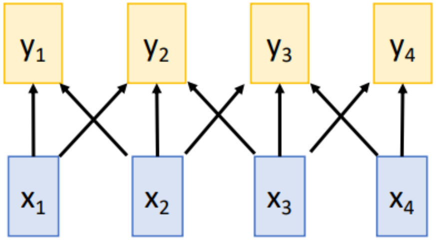
M
I
R
P
R

Recurrent Neural Network



Works on **Ordered Sequences**
(+) Good at long sequences: After one RNN layer, h_T "sees" the whole sequence
(-) Not parallelizable: need to compute hidden states sequentially

1D Convolution



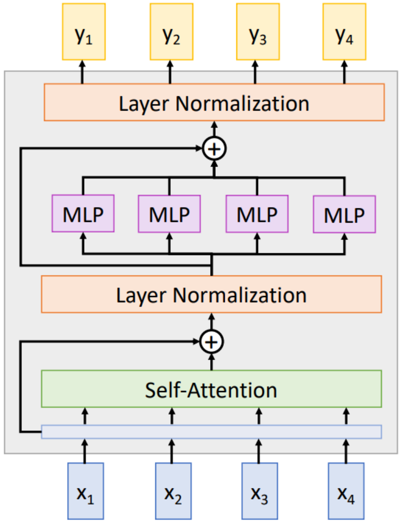
Works on **Multidimensional Grids**
(-) Bad at long sequences: Need to stack many conv layers for outputs to "see" the whole sequence
(+) Highly parallel: Each output can be computed in parallel

Self-Attention

Works on **Sets of Vectors**
(-) Good at long sequences: after one self-attention layer, each output "sees" all inputs!
(+) Highly parallel: Each output can be computed in parallel
(-) Very memory intensive

Transformer*

- Blocurile unui transformer au
  - Input – vectori (multimi de vectori)
  - Output – vectori (multimi de vectori)
  - Hyper-parametrii:
    - # bloks
    - # heads / block
    - width



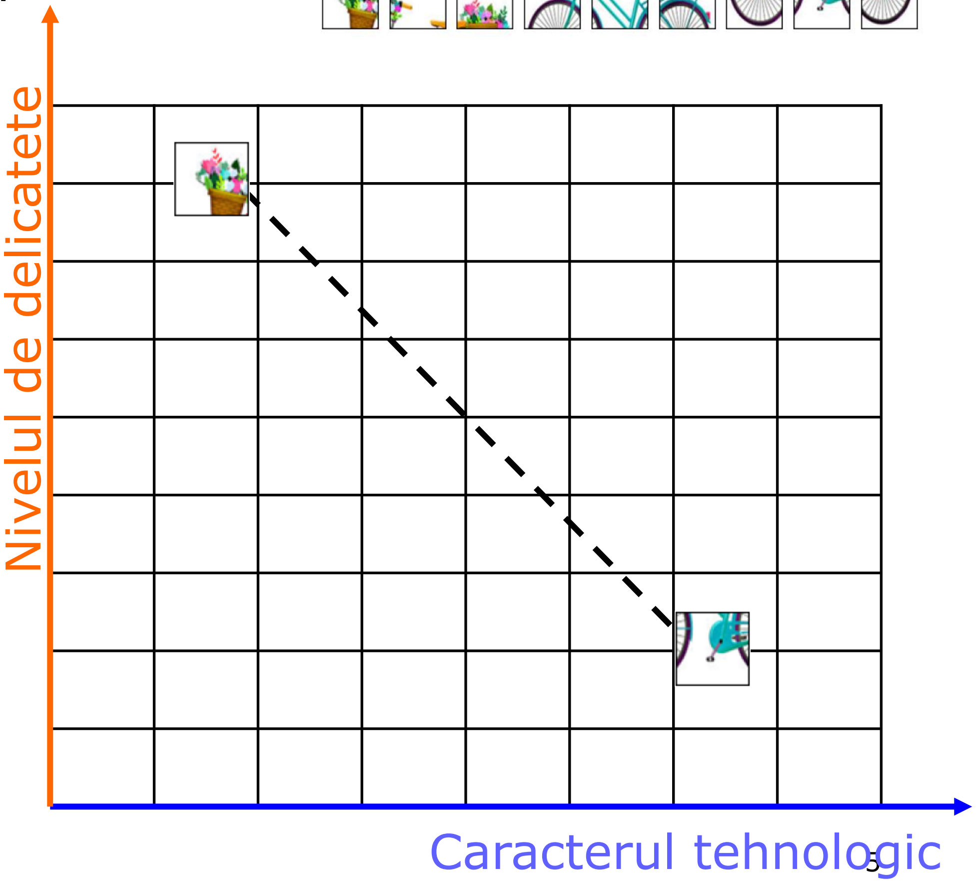
- Vectorii comunica intre ei prin mecanismul de *self-attention* al
  - Unui singur head de procesare
  - A mai multor head-uri de procesare

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems* link

M
I
R
P
R

Mecanismul de “atentie” (attention)

- Remember – embeddings
  - Imagini / patch-uri izolate precum:
  - Presupunem 2 atribute:
    - *caracterul tehnologic*
    - *nivelul de delicatete*

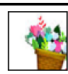


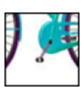
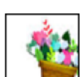
- *Attention*
  - Contextul e ca un magnet!
  - Atrage elementele care se potrivesc!

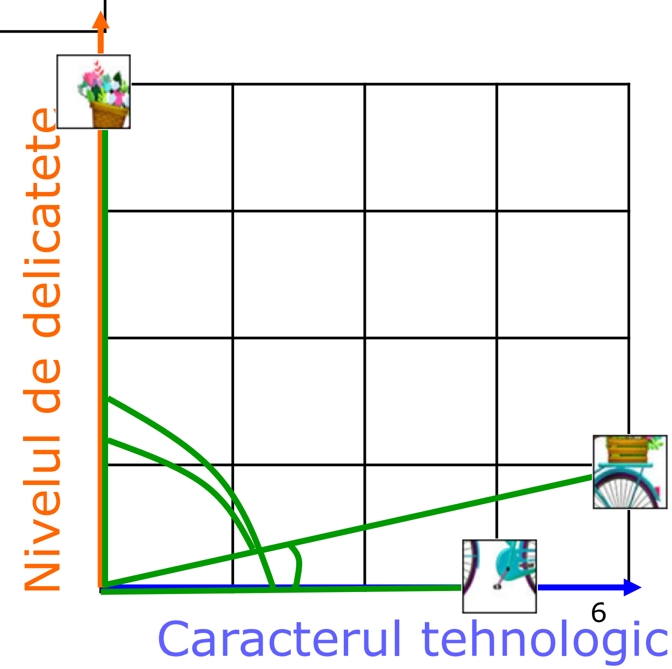
M I R P R

Mecanismul de “atentie” (attention)

- Similaritatea între patch-uri

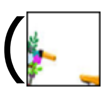
Cuvântul	***caracterul tehnologic***	***nivelul de delicatete***
	3	0
	4	1
	0	4

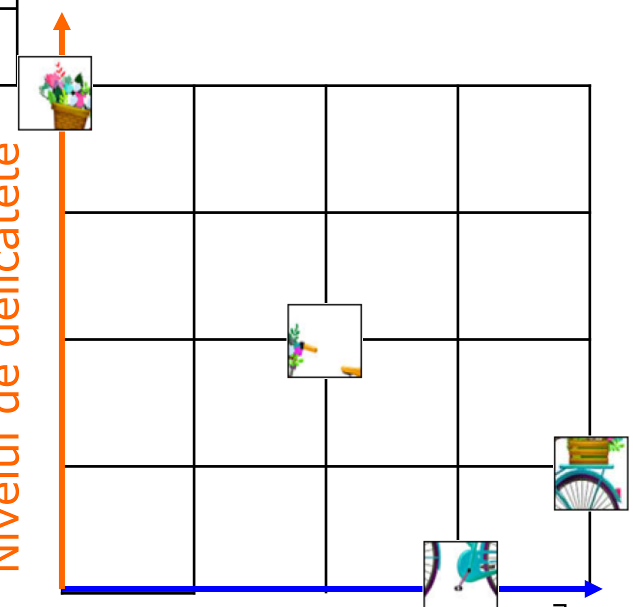
- sim ( , ) = 12
  sim([4,1], [3,0]) =
  4*3 + 1*0 = 12
- sim( , ) = 4
  sim([4,1],[0,4]) =
  4*0 + 1*4 = 4
- sim( , ) = 0
  sim([3,0],[0,4]) =
  3*0 + 0*4 = 0



Mecanismul de “atentie” (attention)

- Contextul patch-urilor – matricea de afinitate

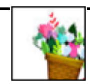
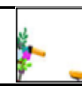
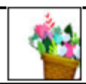
- sim( ,) = 1 / √2 = 0.7
- sim( ,) =
  sim([2,2], [0,4]) =
  (2*0 + 2*4) / √2 =
  8 /√2 = 5.65

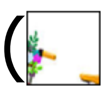


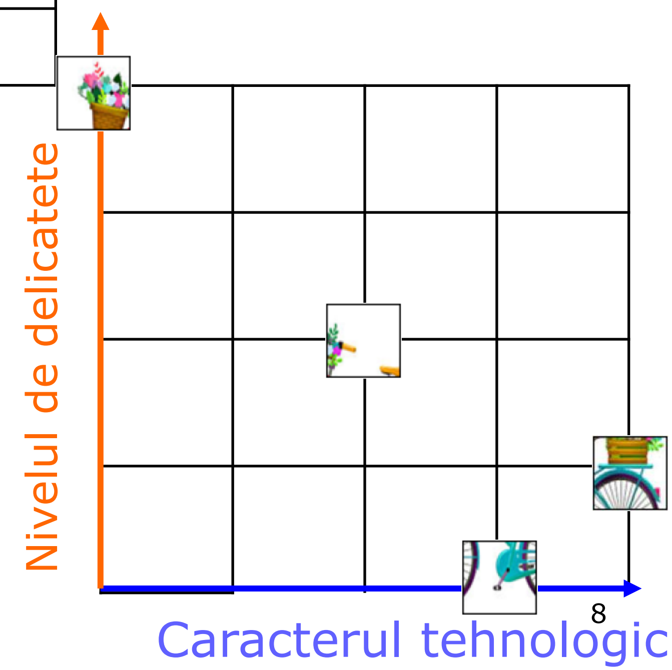
M
I
R
P
R

Mecanismul de “atentie” (attention)

- Contextul patch-urilor – matricea de afinitate

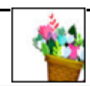
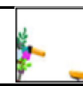
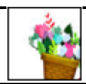
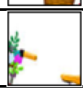
		
	0.70	5.65
	5.65	0.70

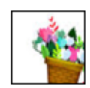
- sim( ,) = 1 / √2 = 0.7
- sim( ,) =
  sim([2,2], [0,4]) =
  (2*0 + 2*4) / √2 =
  8 /√2 = 5.65

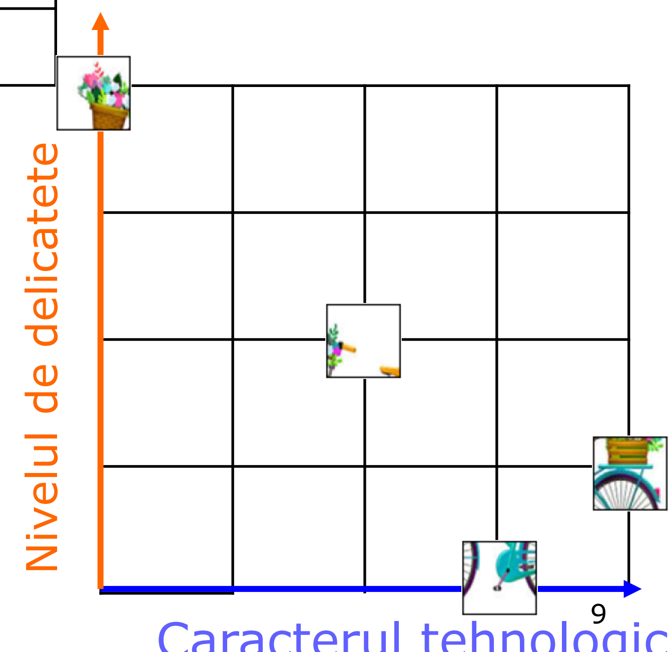


Mecanismul de “atentie” (attention)

- Contextul patch-urilor – matricea de afinitate 

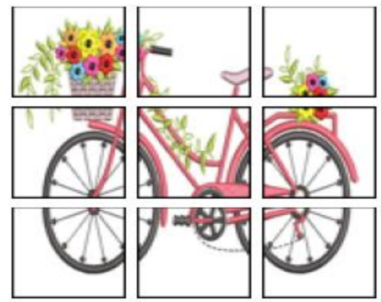
		
	0.70	5.65
	5.65	0.70

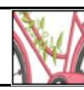
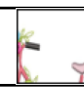
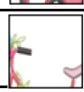
 = 0.7 *  +5.65 * 



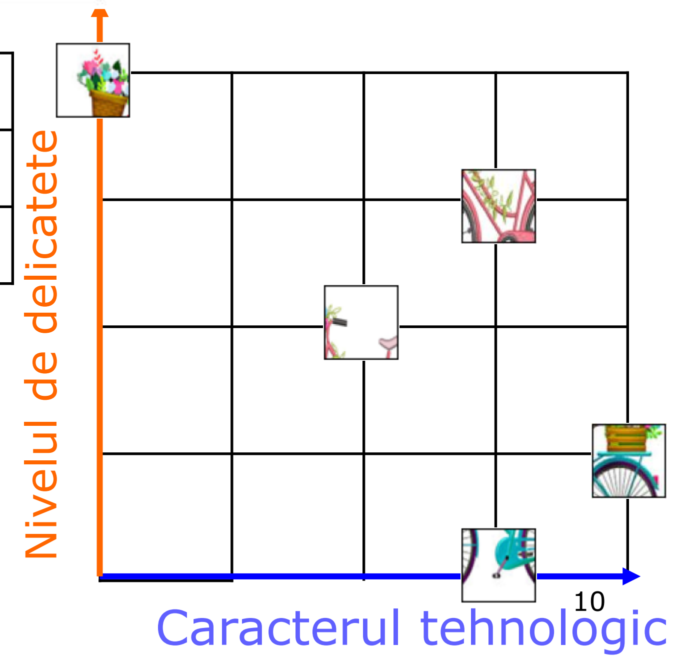
Mecanismul de “atentie” (attention)

- Contextul patch-urilor – matricea de afinitate



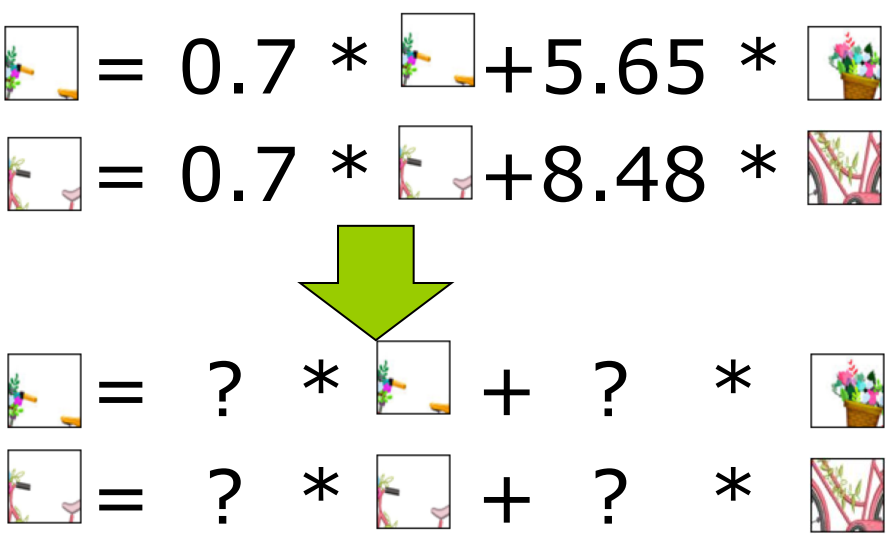
		
	0.70	8.48
	8.48	0.70

 = 0.7 *  +8.48 * 



Mecanismul de “atentie” (attention)

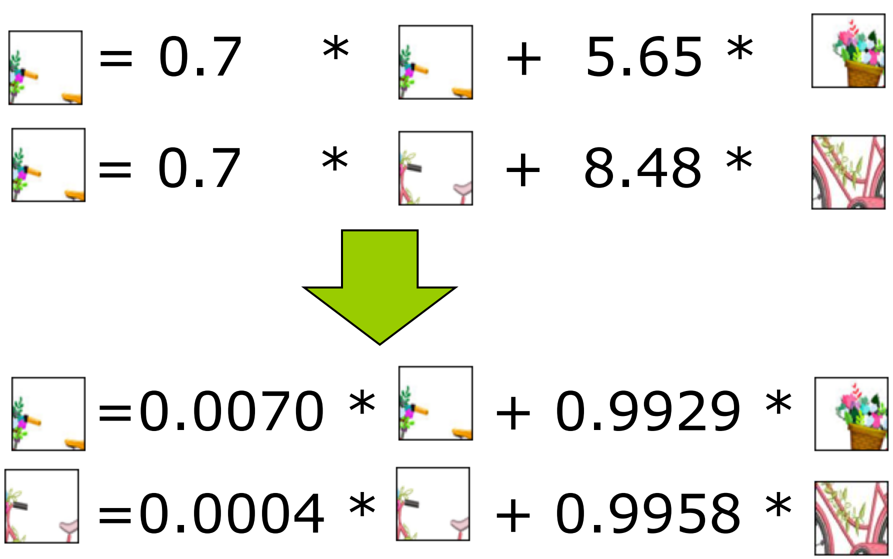
- Contextul patch-urilor – matricea de afinitate
  -
  -



Normalizare – de care?

Mecanismul de “atentie” (attention)

- Contextul patch-urilor – matricea de afinitate
  - 
  - 



Normalizare – softmax

Mecanismul de “atentie” (attention)

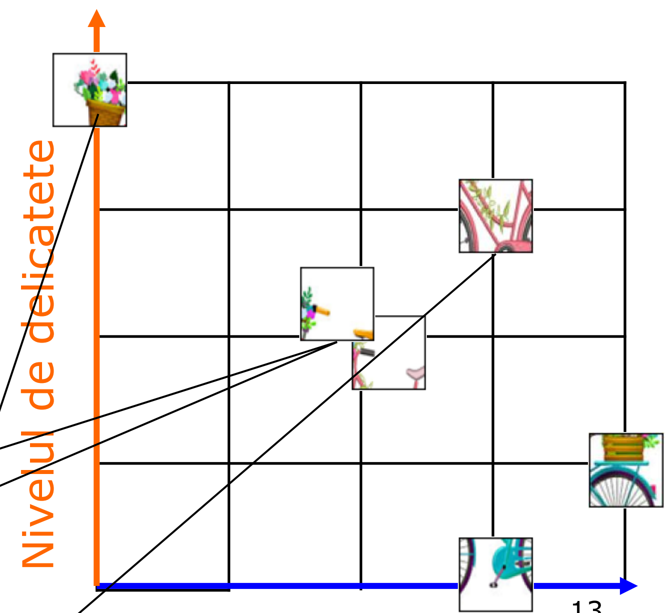
- Contextul patch-urilor – matricea de afinitate
  -
  -

 = 0.7 *  + 5.65 * 

 = 0.7 *  + 8.48 * 

 =0.0070 * [2,2] + 0.9929 * [0,4]

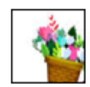
 =0.0004 *[2,2] + 0.9958 * [3,3]

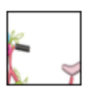


M
I
R
P
R

Mecanismul de “atentie” (attention)

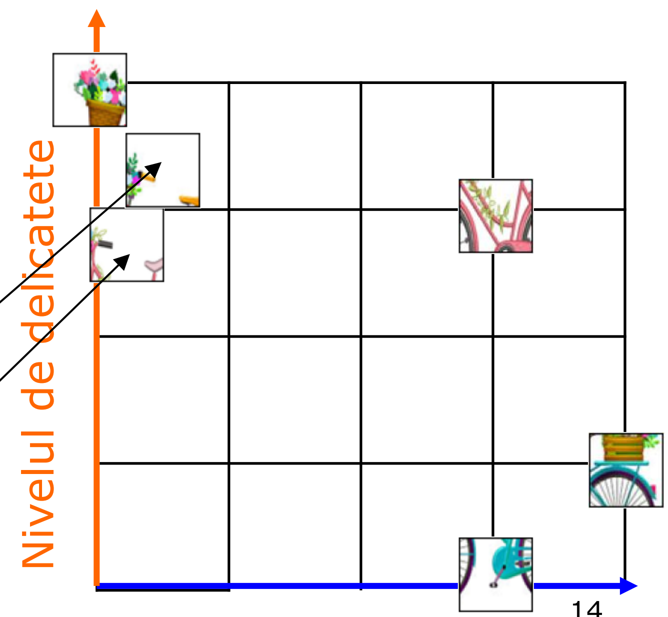
- Contextul patch-urilor – matricea de afinitate

 = 0.7 *  + 5.65 * 

 = 0.7 *  + 8.48 * 

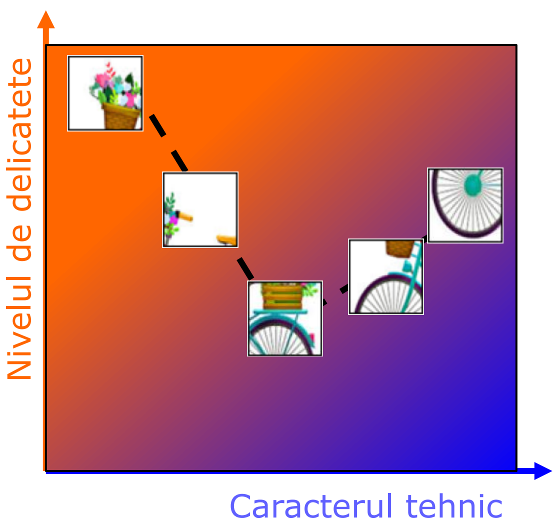
 =[0.1400, 3.9850]

 =[0.0008, 3.9840]

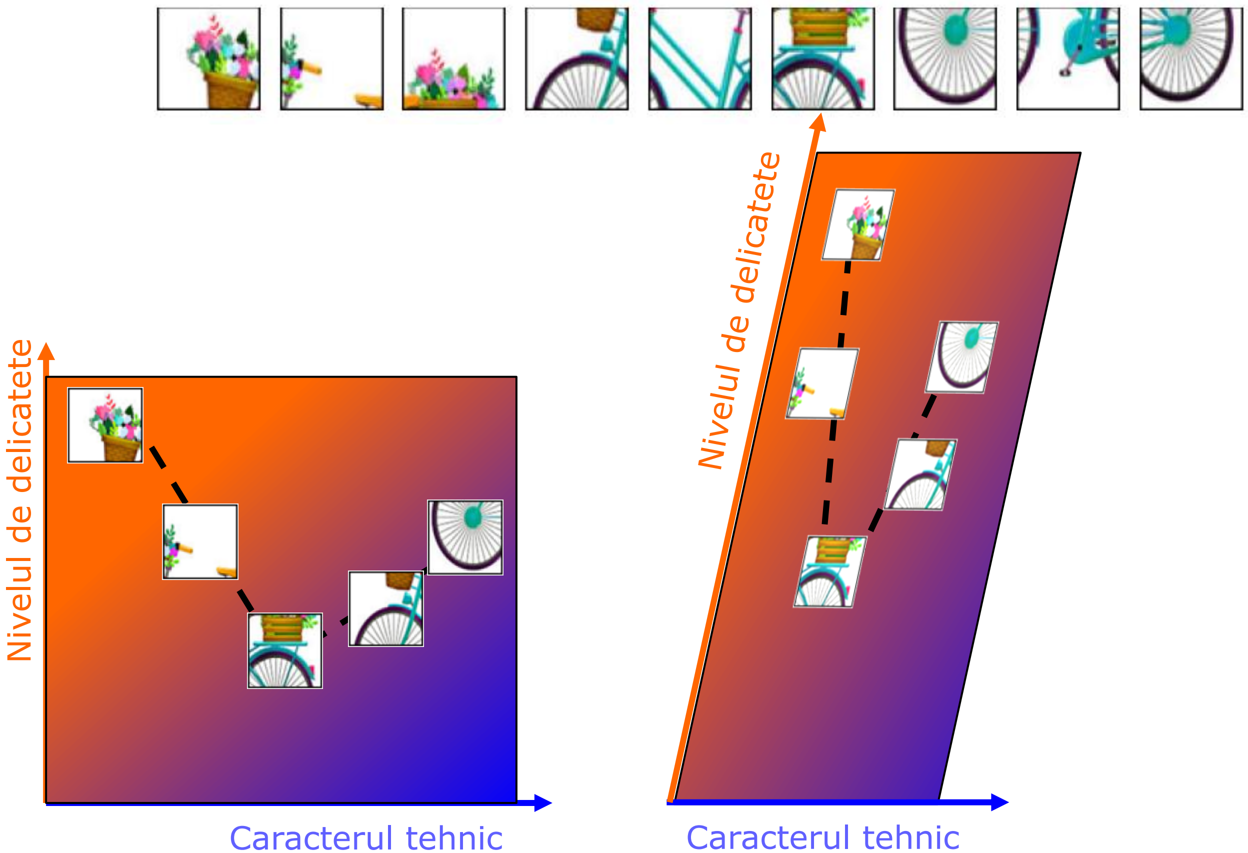


M
I
R
P
R

Mecanismul de “atentie” (attention)



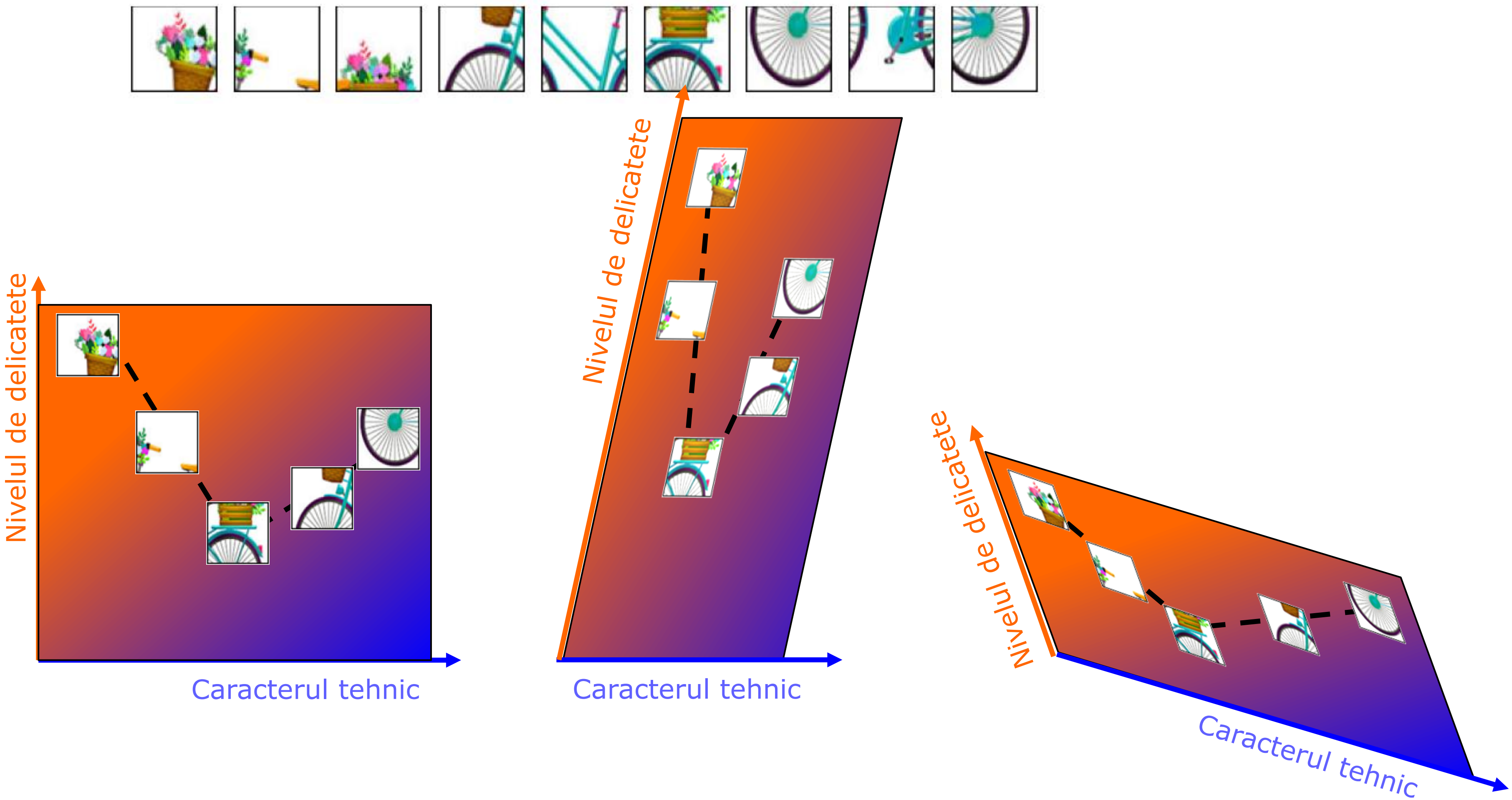
Mecanismul de “atentie” (attention)



a) reprez1

b) reprez2

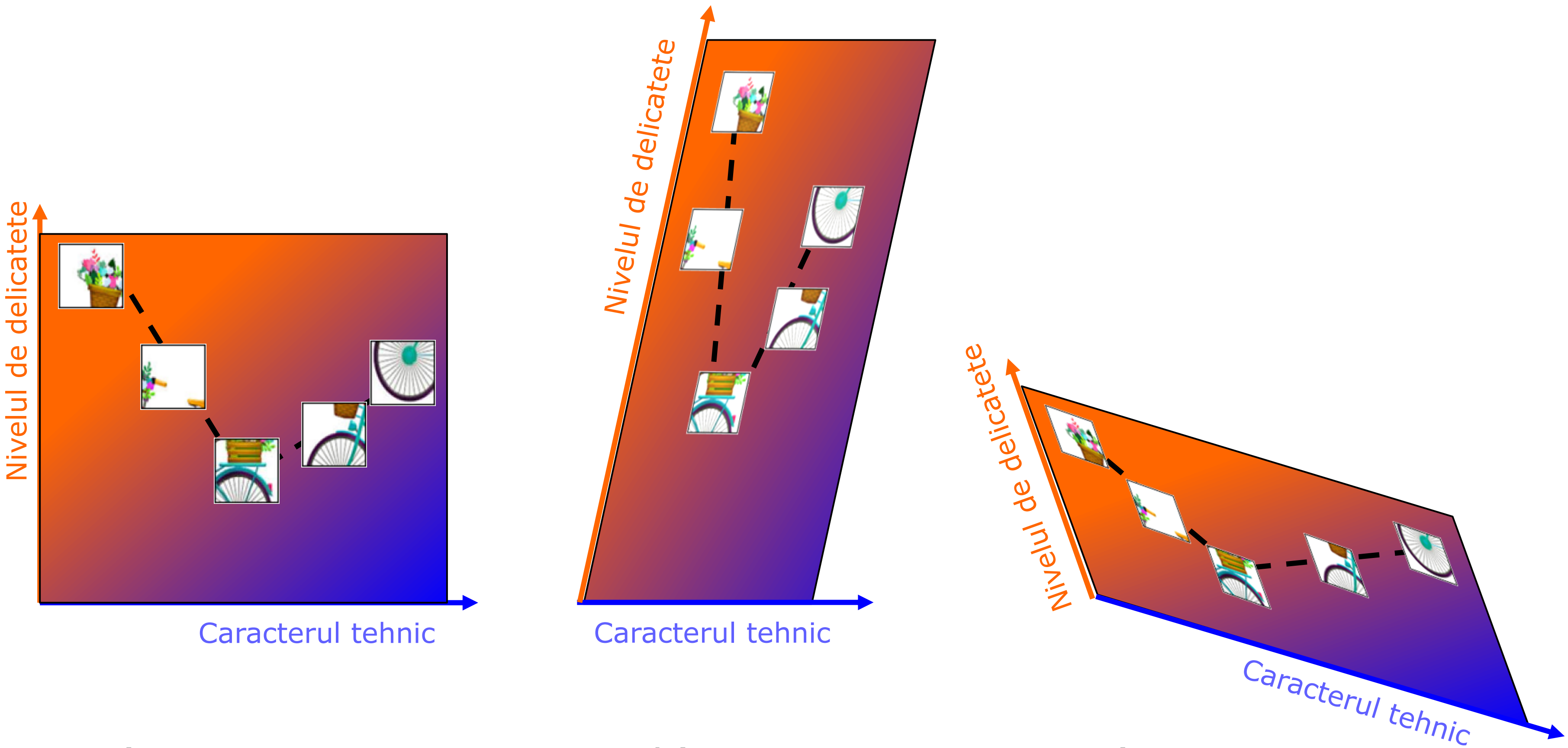
Mecanismul de “atentie” (attention)



a) reprez1 b) reprez2 c) reprez3

Mecanismul de “atentie” (attention)

M
I
R
P
R

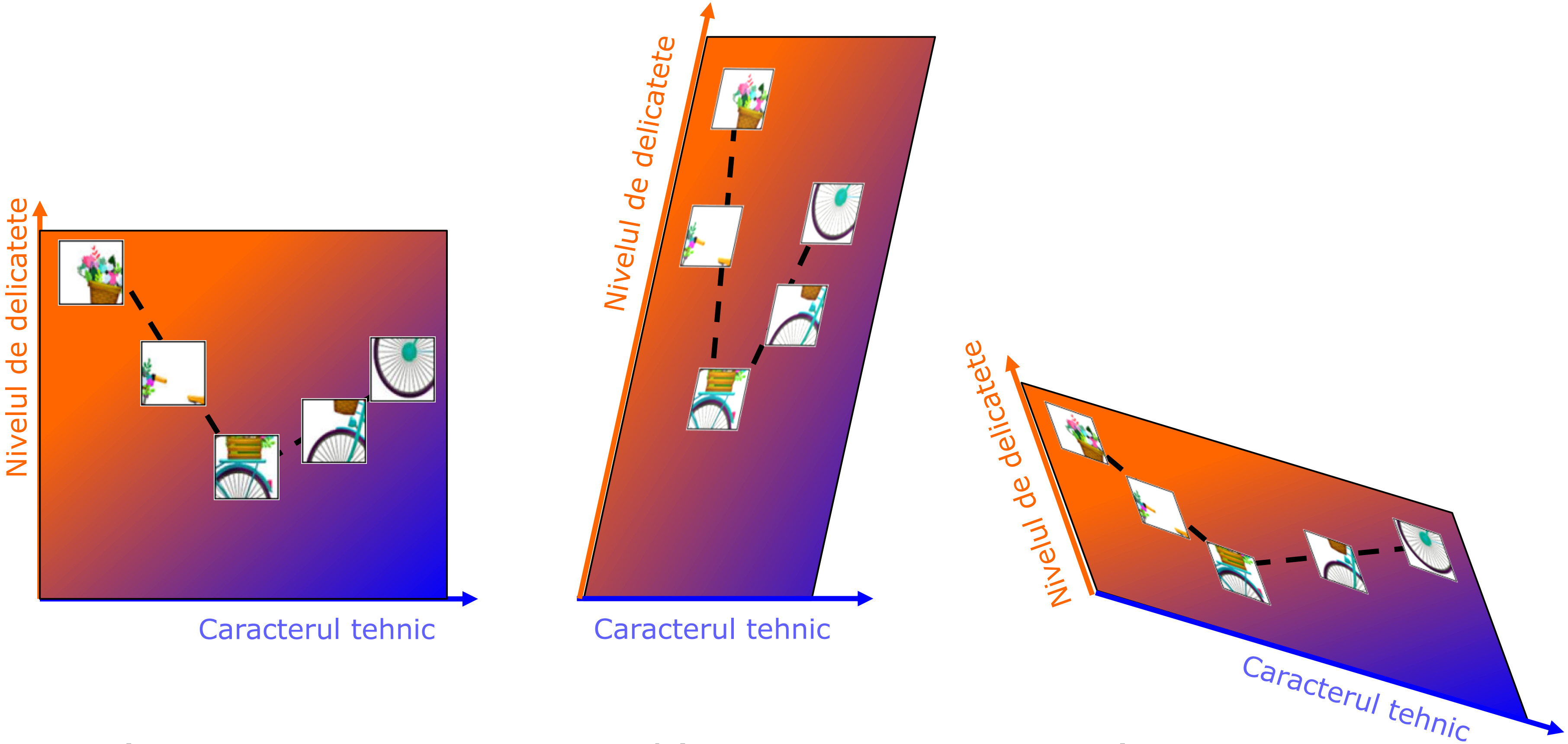


a) reprez1 b) reprez2 c) reprez3

Q1: reprez1 = f(reprez2) = g(reprez3) ???

Da!!! Tranformare liniară!!!

Mecanismul de “atentie” (attention)



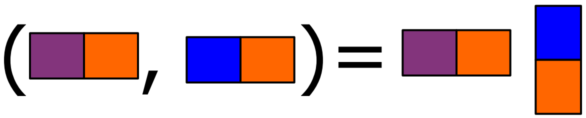
a) reprez1 b) reprez2 c) reprez3

Q2: Care reprezentare e mai bună pentru a stabili similaritatea (deci a face diferența între caracteristicile patch-lui  **)?**

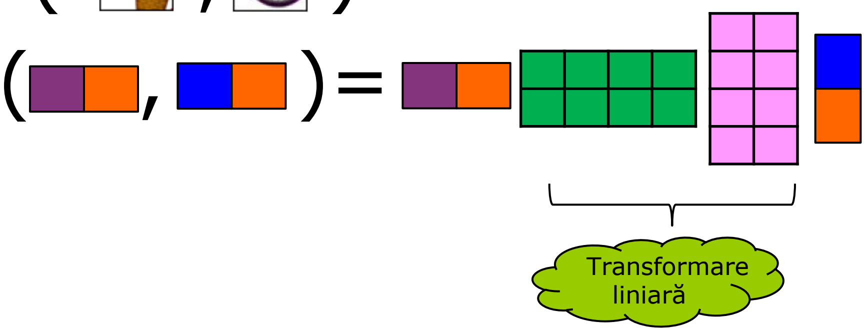
Reprezentarea 1 sau 3!

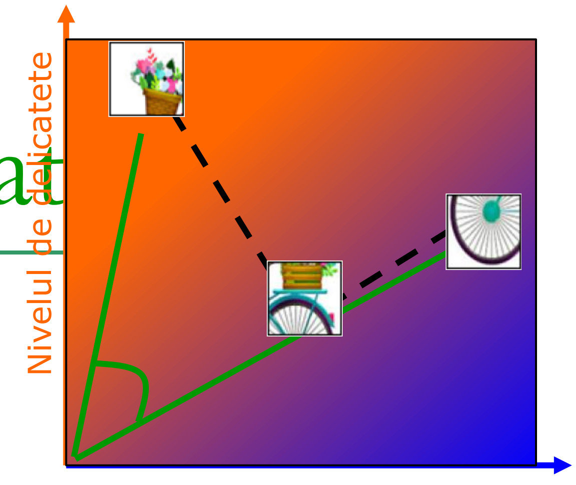
Mecanismul de “atentie” (at

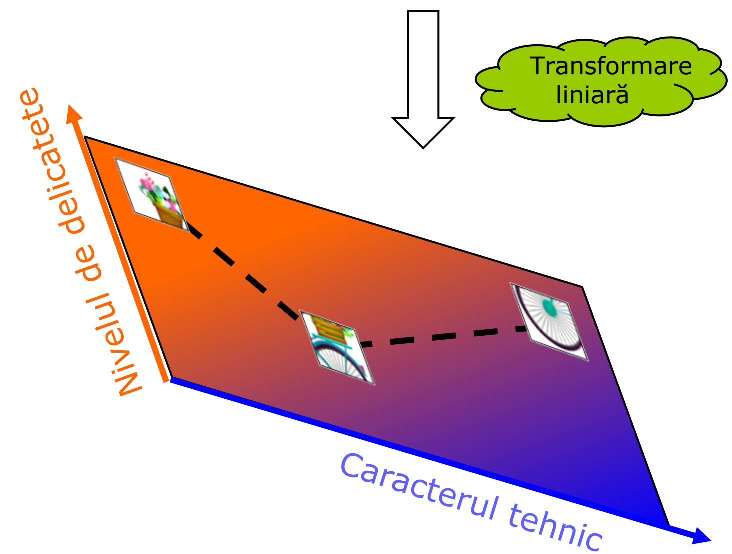
- Sim() =

  sim(

- Sim() =

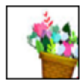
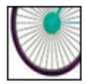
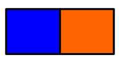
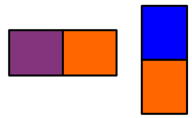
  sim(

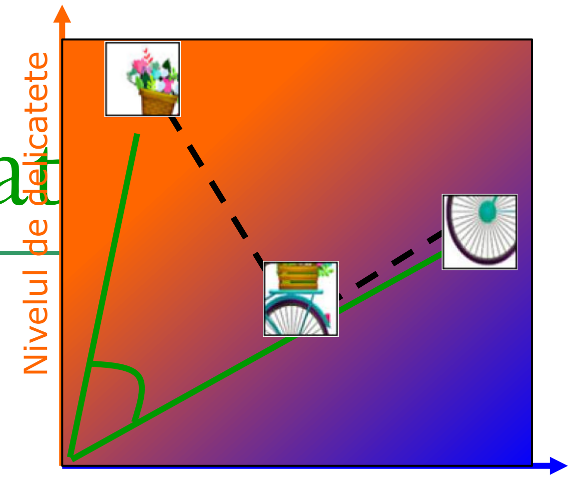


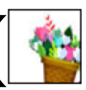
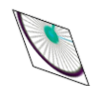
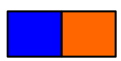
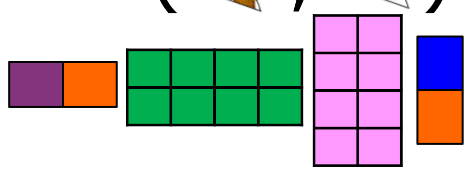


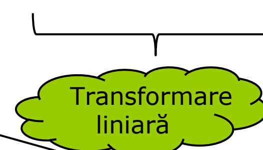
M I R P R

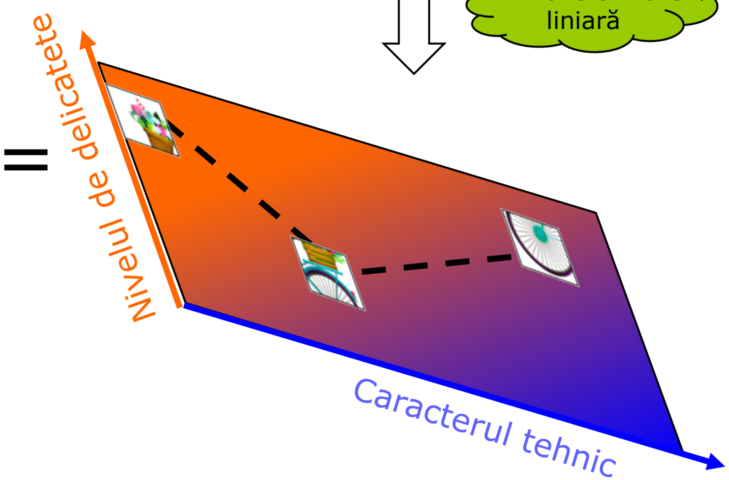
Mecanismul de “atentie” (at

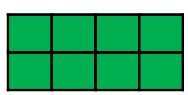
- sim(, ) =
  sim(, )= 

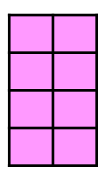


- sim(,) = sim( , ) =
  sim(, )= 





K (Key) = 

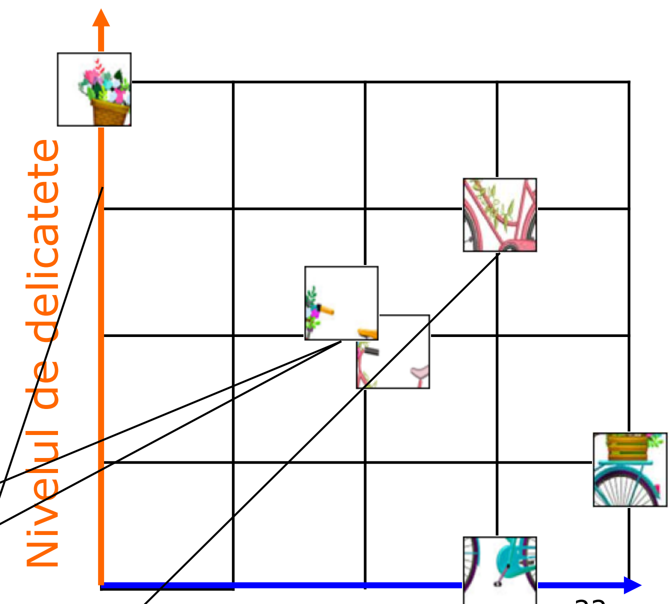
Q (Queries) = 

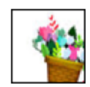
M
I
R
P
R

MIRPR

Mecanismul de “atentie” (attention)

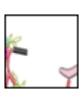
- Contextul patch-urilor – matricea de afinitate
  -
  -



 = 0.7 *  + 5.65 * 

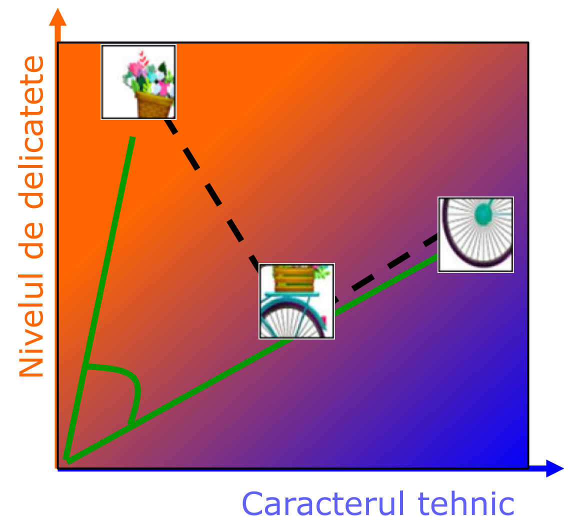
 = 0.7 *  + 8.48 * 

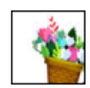
 =0.0070 * [2,2] + 0.9929 * [0,4] = [0.1400, 3.9850]

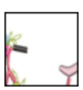
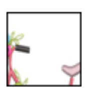
 =0.0004 *[2,2] + 0.9958 * [3,3] = [0.0008, 3.9840]

Mecanismul de “atentie” (attention)

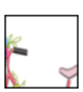
- Contextul patch-urilor – matricea de afinitate
  - 



 = 0.7 *  + 5.65 * 

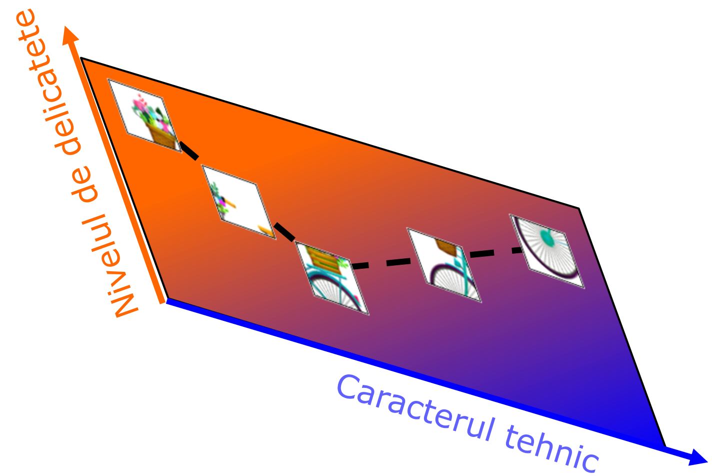
 = 0.7 *  + 8.48 * 

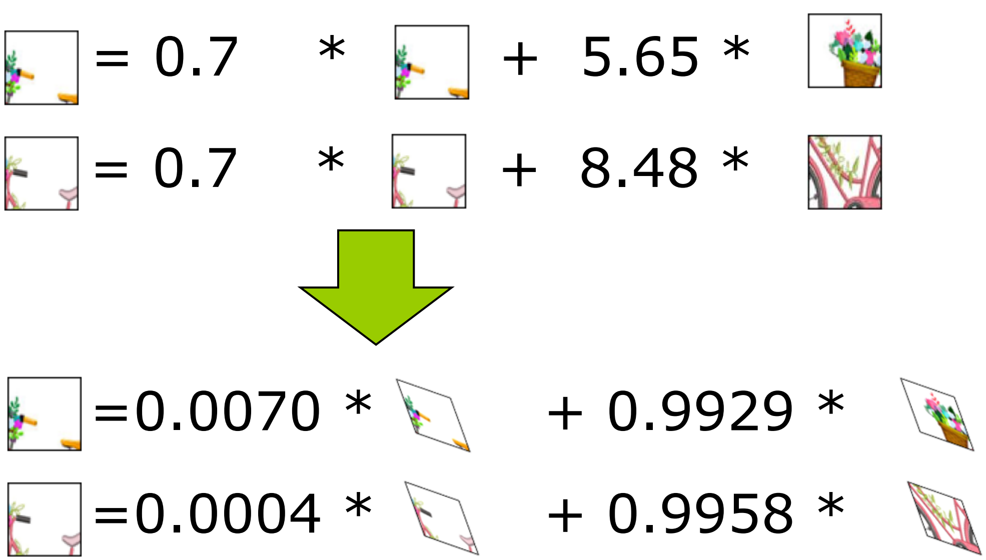
 =0.0070 * [2,2] + 0.9929 * [0,4] = [0.1400, 3.9850]

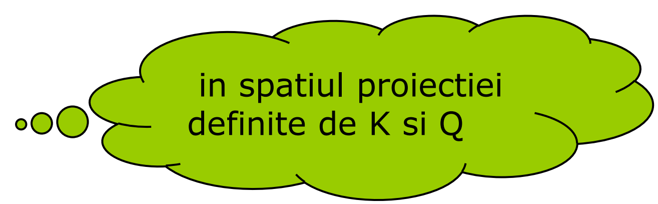
 =0.0004 *[2,2] + 0.9958 * [3,3] = [0.0008, 3.9840]

Mecanismul de "atentie" (attention)

- Contextul patch-urilor – matricea de afinitate
  -
  -

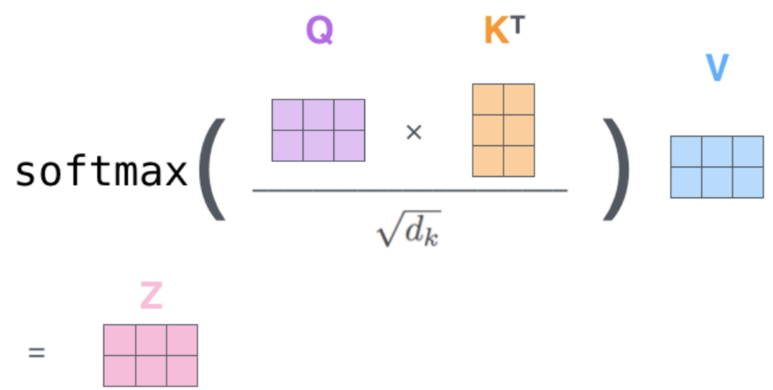






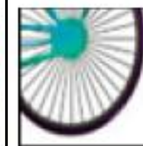
Mecanismul de “atentie” (attention)

- Input – x - vectori de dimensiune (N, #patches x #patches + 1, De)
- Output - vectori de dimensiune (N, #patches x #patches + 1, D)
  - Q = x * W^q
  - K = x * W^k
  - V = x * W^v

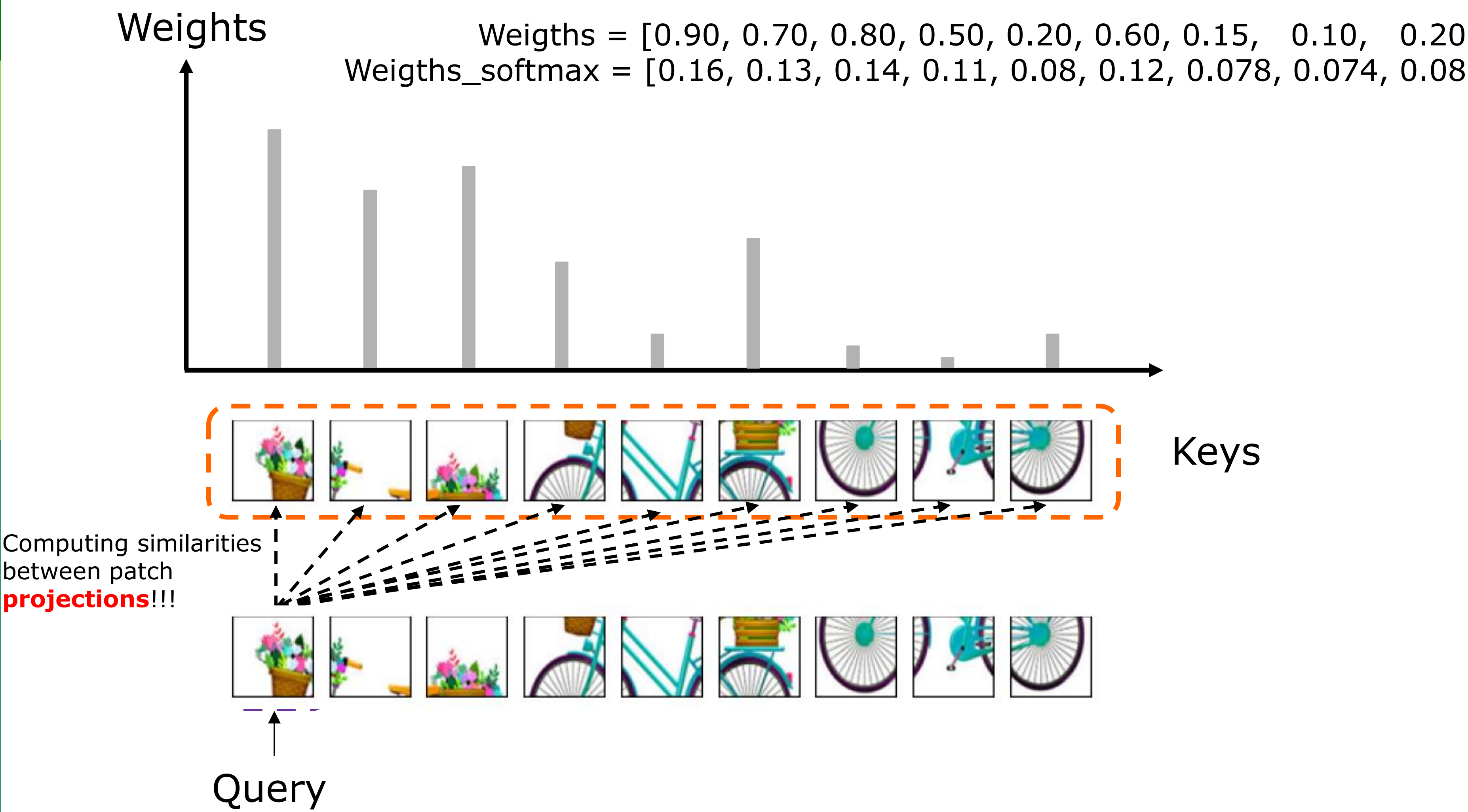


MIRPR

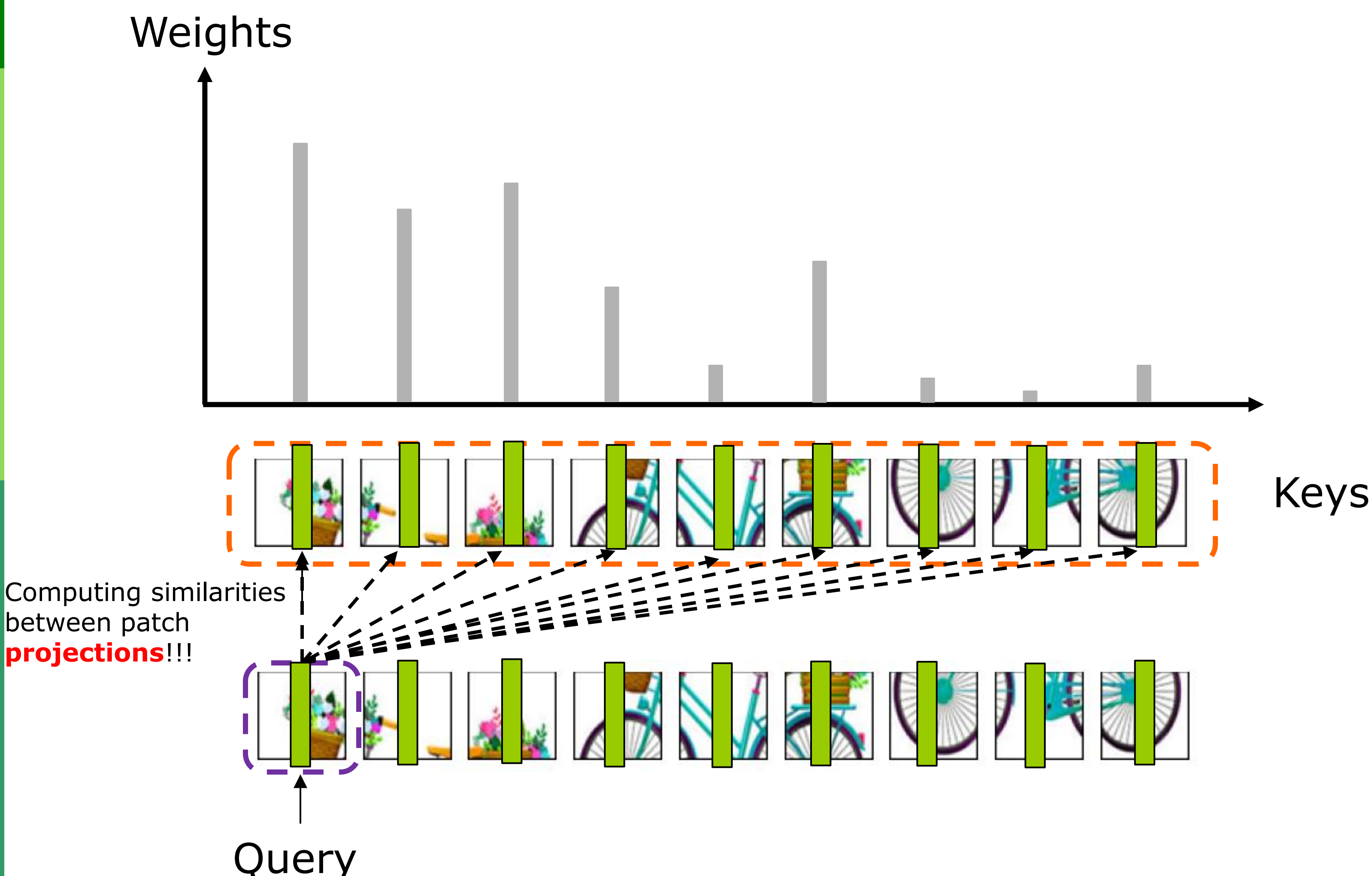
Mecanismul de “atentie” (attention)

Input									
representation	x_1								
Queries	q_1								
Keys	k_1								
Values	v_1								
Similarities	$q_1 \bullet k_1$	$q_1 \bullet k_2$	$q_1 \bullet k_3$	$q_1 \bullet k_4$	$q_1 \bullet k_5$	$q_1 \bullet k_6$	$q_1 \bullet k_7$	$q_1 \bullet k_8$	$q_1 \bullet k_9$
Scores s= [s1, s2, .., s9]	$q_1 \bullet k_1 / \sqrt{D}$	$q_1 \bullet k_2 / \sqrt{D}$	$q_1 \bullet k_3 / \sqrt{D}$	$q_1 \bullet k_4 / \sqrt{D}$	$q_1 \bullet k_5 / \sqrt{D}$	$q_1 \bullet k_6 / \sqrt{D}$	$q_1 \bullet k_7 / \sqrt{D}$	$q_1 \bullet k_8 / \sqrt{D}$	$q_1 \bullet k_9 / \sqrt{D}$
w = Softmax(s)									
Weightening the values	$z_1 = w_1 * v_1 + w_2 * v_2 + \ldots + w_9 * v_9$								

Mecanismul de “atentie” (attention)



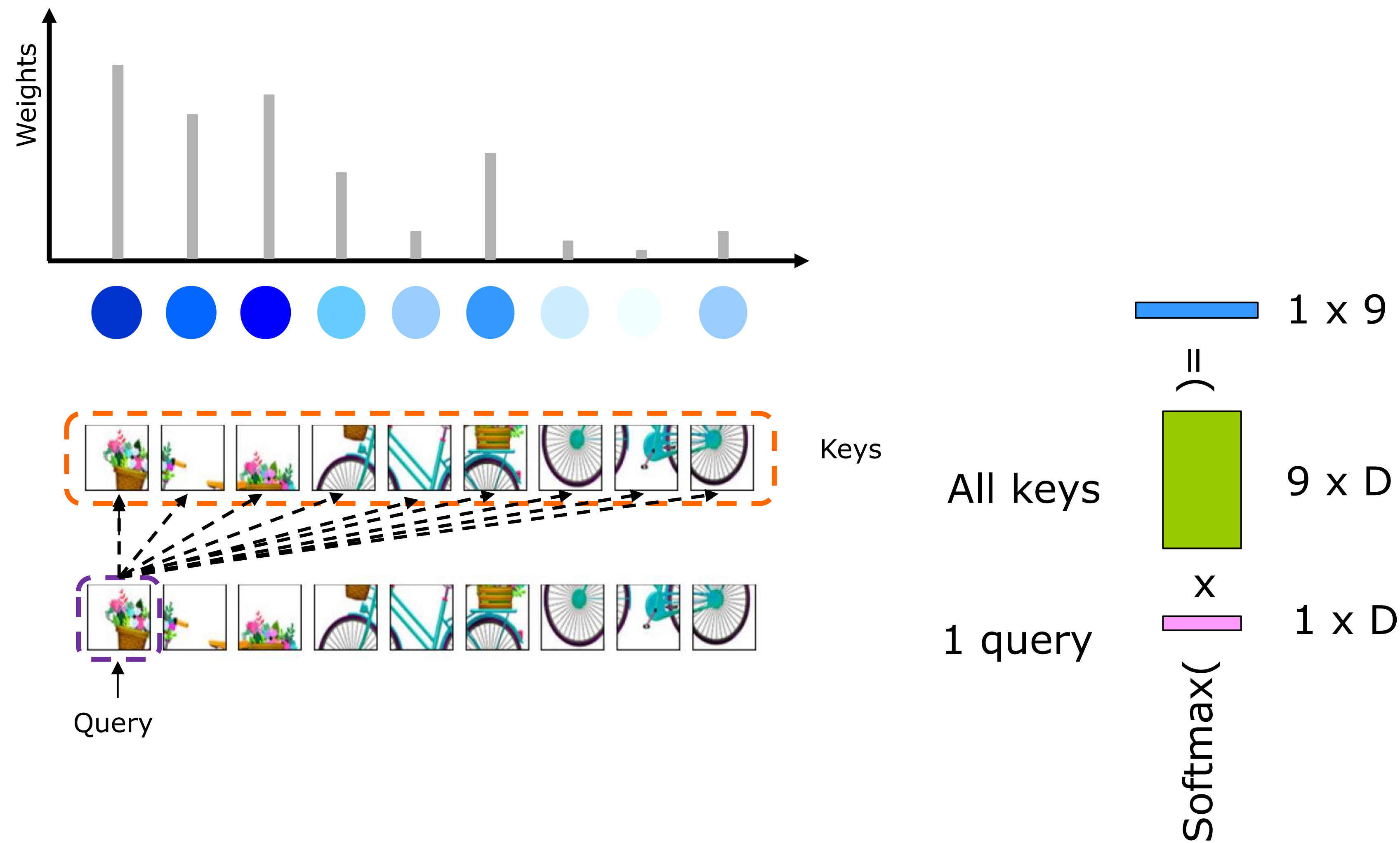
Mecanismul de “atentie” (attention)



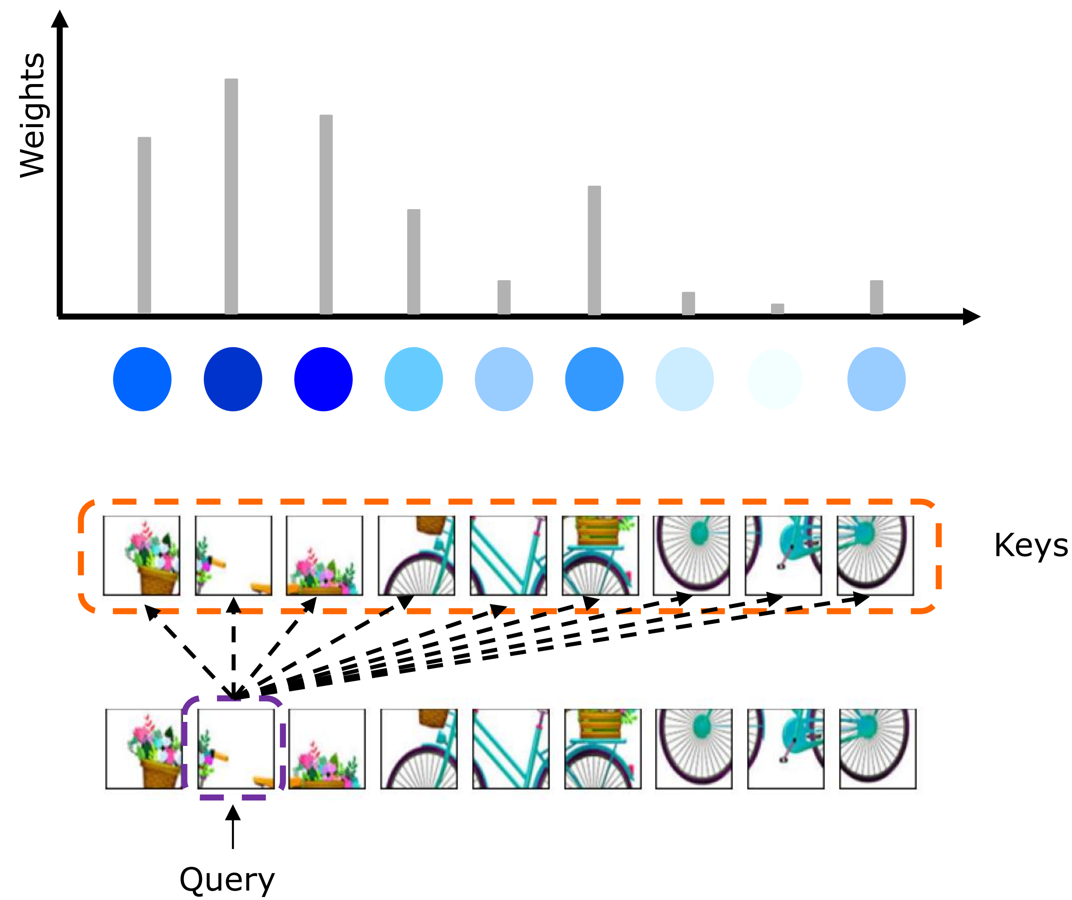
M I R P R

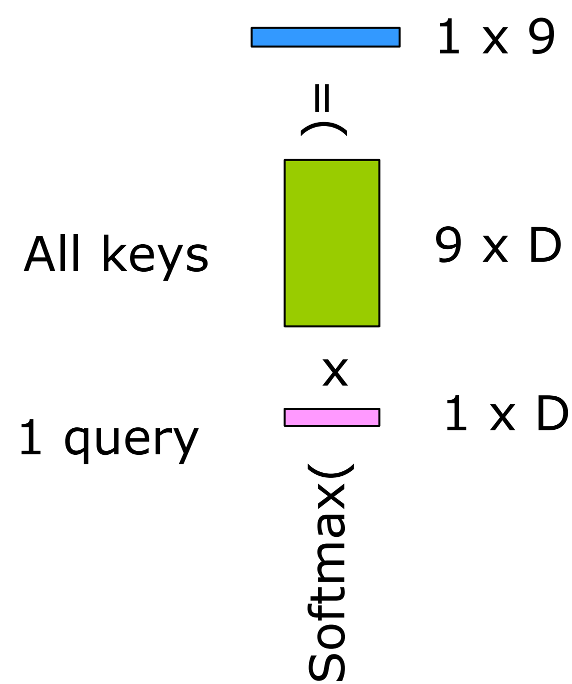
MIRPR

Mecanismul de “atentie” (attention)

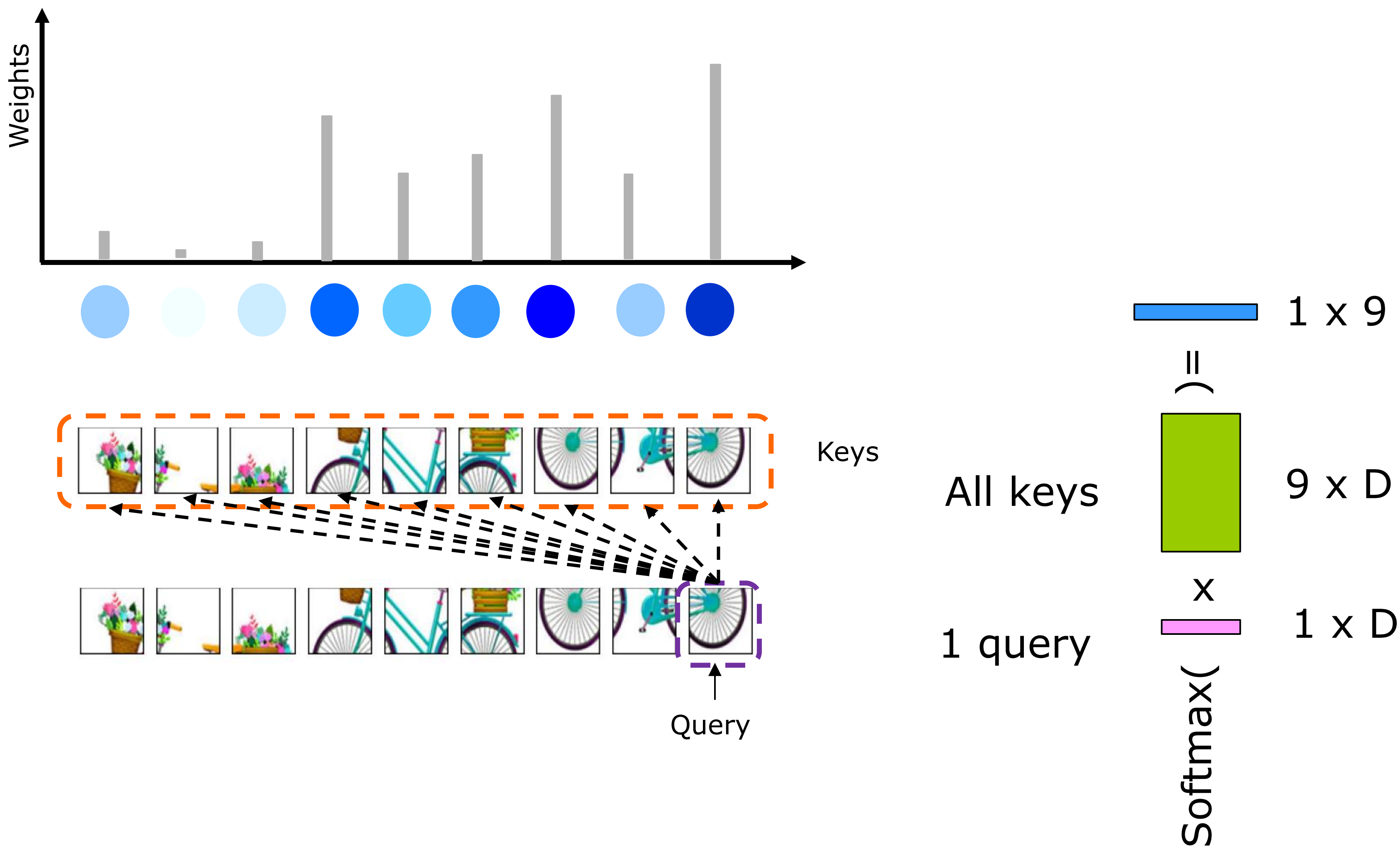


Mecanismul de “atentie” (attention)



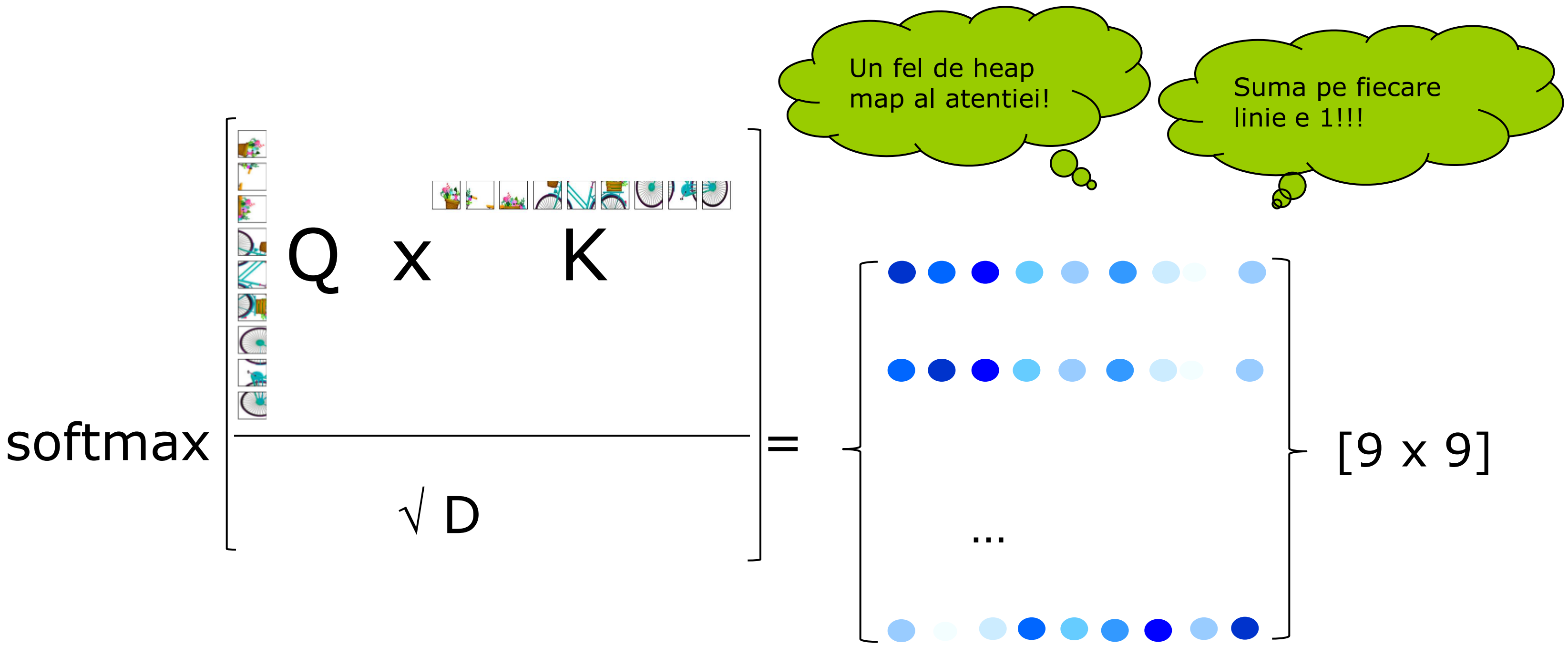


Mecanismul de “atentie” (attention)

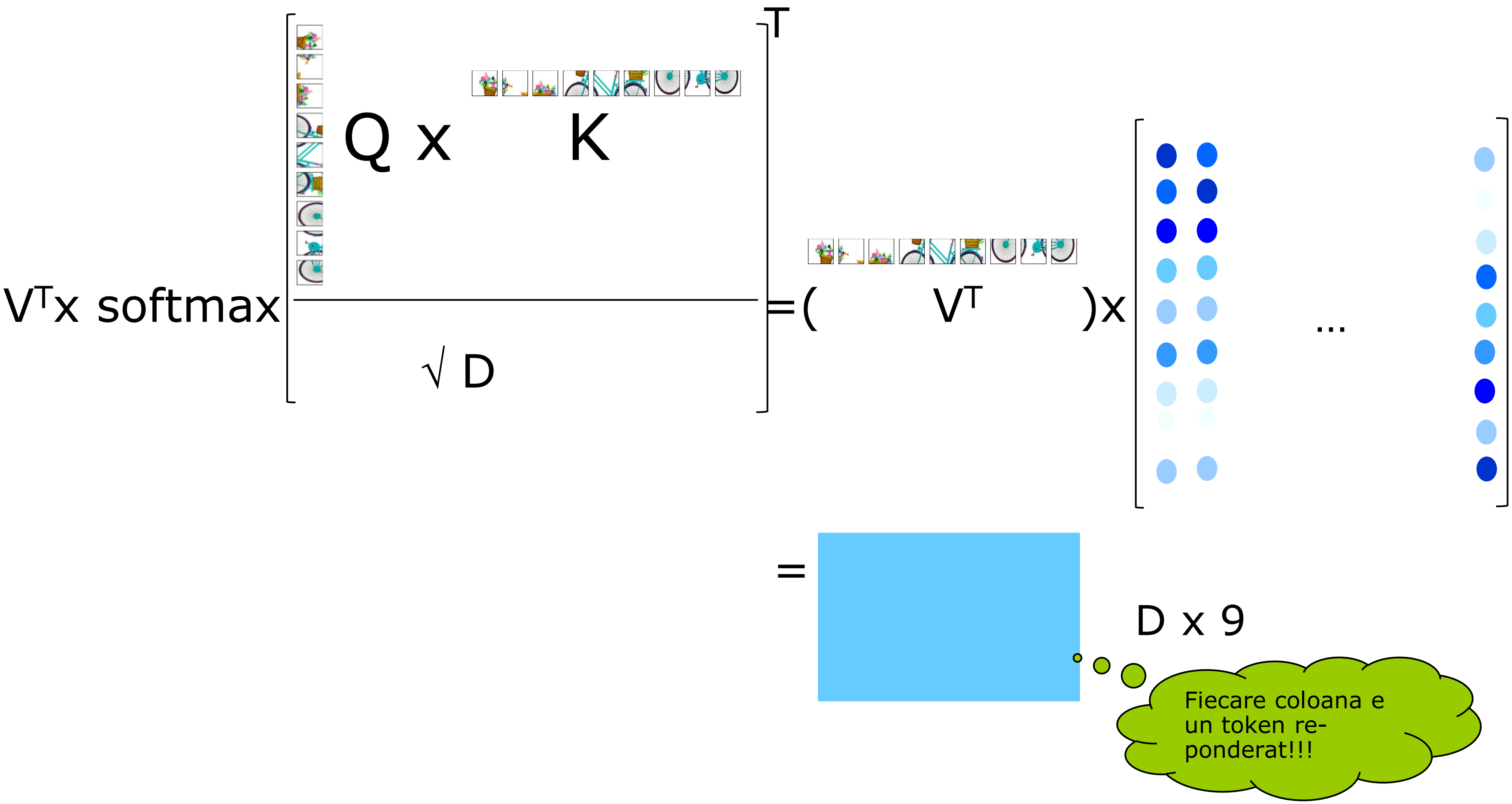


M
I
R
P
R

Mecanismul de “atentie” (attention)

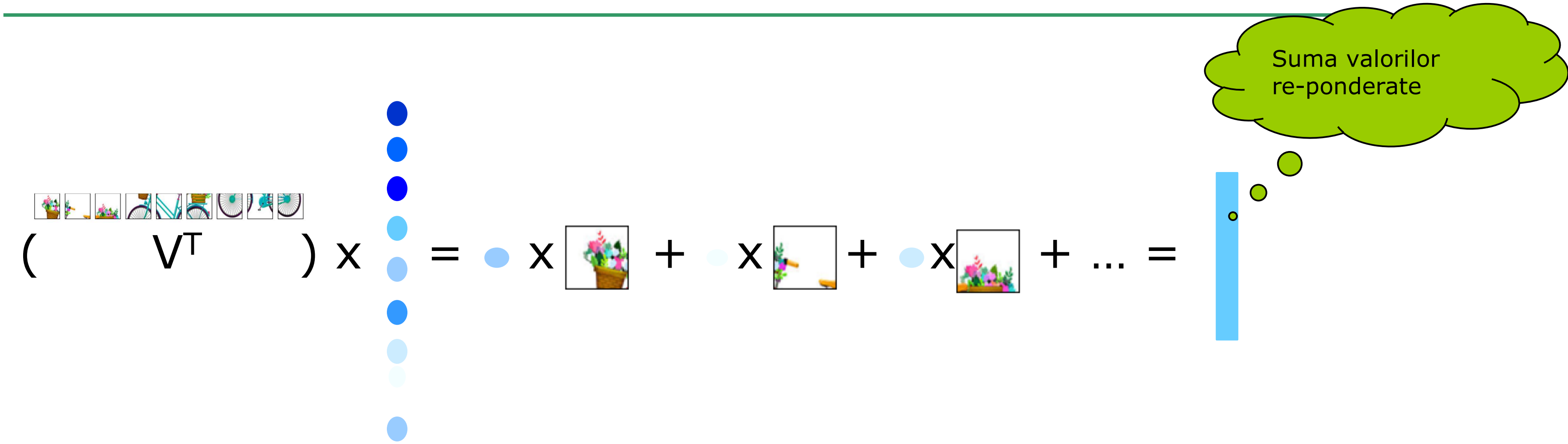


Mecanismul de “atentie” (attention)



M
I
R
P
R

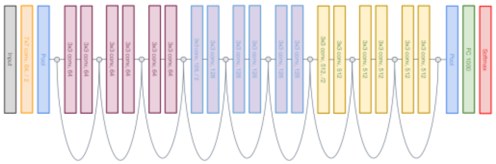
Mecanismul de “atentie” (attention)

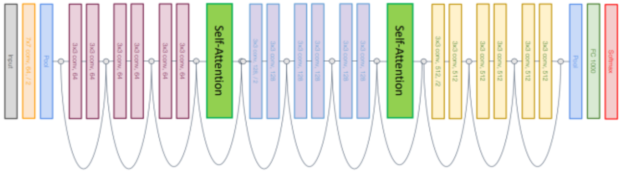


M
I
R
P
R

Cum se poate integra mecanismul de atentie intr-o retea neuronala pentru procesarea imaginilor?

- V1: Integrarea atentiei intr-o CNN – de ex ResNet[1]
  - Cum?
    - Straturi suplimentare
  - Pro
    - Integrare facila
  - Contra
    - Modelul (arhitectura) e dominate de convolutii

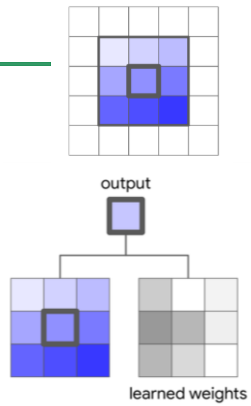


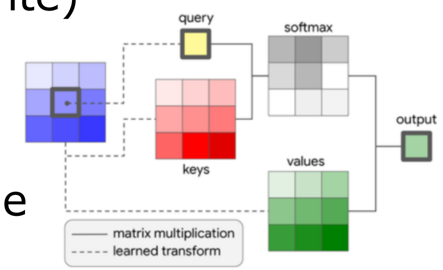


Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7794-7803) link

Cum se poate integra mecanismul de atentie intr-o retea neuronala pentru procesarea imaginilor?

- V2: Inlocuirea convolutiei cu o “atentie locala” (“local relation”)
  - Cum?
    - Convolutia = produs scalar intre filtru si patch
    - Local attention
      - Centrul patch-ului -> query (vector cu D elemente)
      - Fiecare element din patch ->
        - Keys (vector R x R x D)
        - Values (vector R x R x C’)
      - Outputul e calculate cu ajutorul mecanismului de atentie
  - Pro
    - Numar mai mic de parametrii (in Hu et al.: ResNet-50 are $25.5x10^6$ param, iar LR-Net-50 are $23.3x10^6$ params)
  - Contra
    - Implementare dificila (multe detalii tricky)
    - Doar putin mai buna ca ResNet

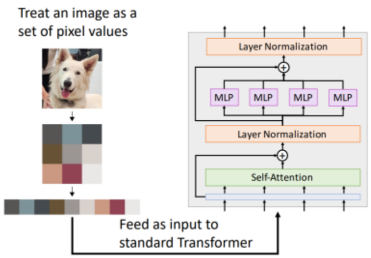




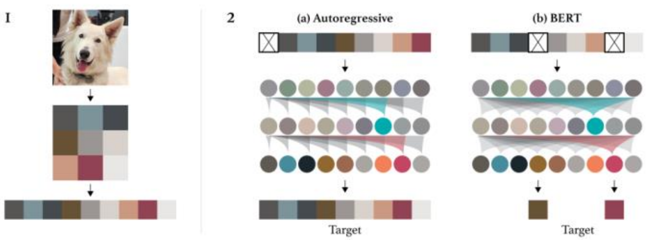
Hu, H., Zhang, Z., Xie, Z., & Lin, S. (2019). Local relation networks for image recognition. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3464-3473) link

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., & Shlens, J. (2019). Stand-alone self-attention in vision models. *Advances in neural information processing systems*, *32* link

Cum se poate integra mecanismul de atentie intr-o retea neuronala pentru procesarea imaginilor?



- V3: transformer aplicat direct pe pixeli
  - Cum?
    - Input-ul pt transformer este imaginea (redimensionata) si aplatizata

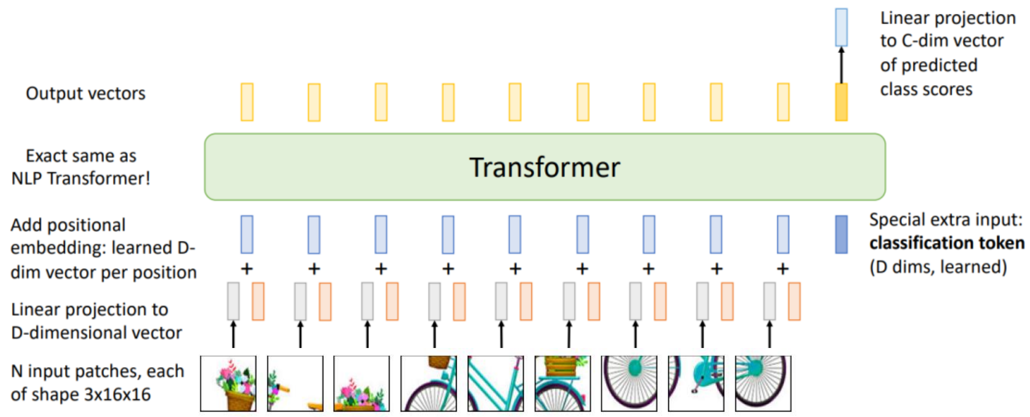


- Pro
  - Simplu (conceptual)
- Contra
  - O imagine de dimensiune n x n necesita n^4 elemente in fiecare matrice de atentie -> multa memorie

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., & Sutskever, I. (2020, November). Generative pretraining from pixels. In *International conference on machine learning* (pp. 1691-1703). PMLR link

Cum se poate integra mecanismul de atentie intr-o retea neuronala pentru procesarea imaginilor?

- V4: transformer aplicat pe patch-uri
  - Cum?



  - Pro
    - Mai putine convolutii
  - Contra
    - Numar mare de parametrii
    - Nevoie de multe date de antrenament

Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. link

MIRPR

Cum se poate imbunatati performanta unui ViT?

- Regularizare
  - Weight decay, stochastic depth, dropout
- Data augmentation
  - MixUp, RandAugment

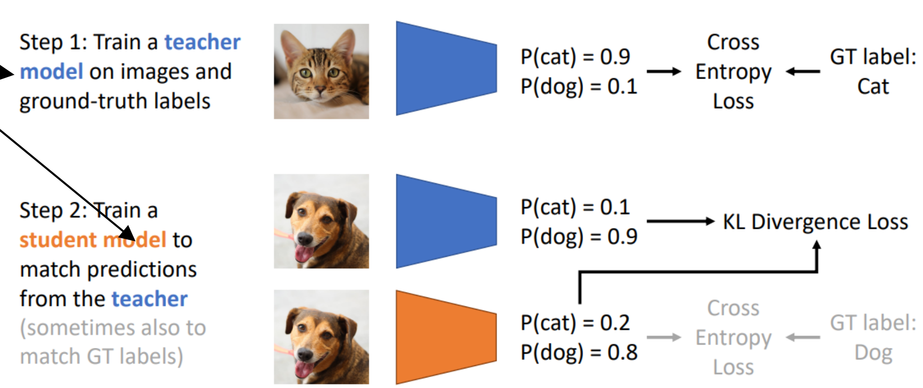
M
I
R
P
R

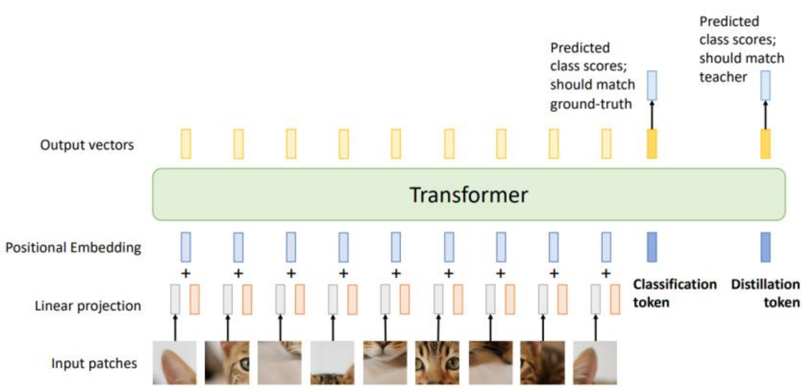
Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., & Beyer, L. (2021). How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*. link

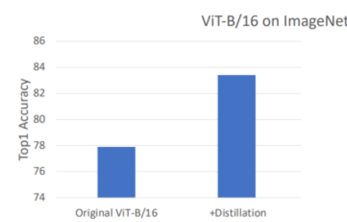
Cum se poate imbunatati performanta unui ViT?

- Distillation

CNN







Hinton, G. (2015). Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531* link
Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021, July). Training data-efficient image transformers & distillation through attention. In *International conference on machine learning* (pp. 10347-10357). PMLR link

Cum se poate imbunatati performanta unui ViT?

ViT vs CNN

In most CNNs (including ResNets), **decrease** resolution and **increase** channels as you go deeper in the network (Hierarchical architecture)

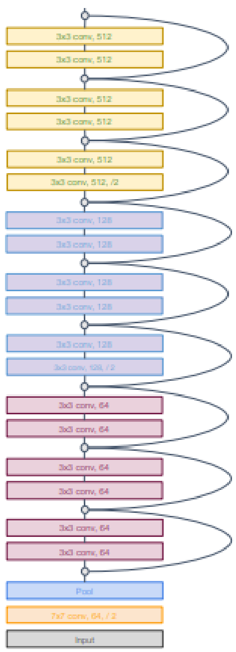
Useful since objects in images can occur at various scales

Stage 3:
256 x 14 x 14

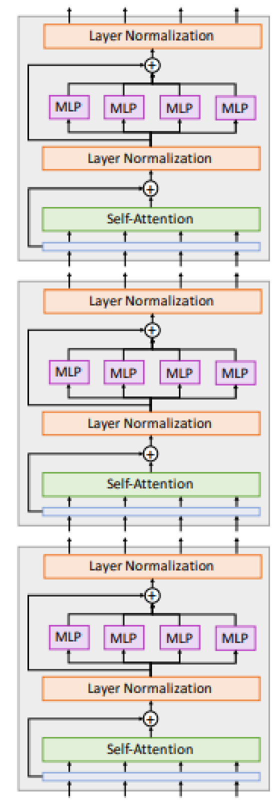
Stage 2:
128 x 28 x 28

Stage 1:
64 x 56 x 56

Input:
3 x 224 x 224



In a ViT, all blocks have same resolution and number of channels (Isotropic architecture)



3rd block:
768 x 14 x 14

2nd block:
768 x 14 x 14

1st block:
768 x 14 x 14

Input:
3 x 224 x 224

More details

- https://github.com/google-research/vision_transformer
- https://huggingface.co/docs/transformers/model_doc/vit

M I R P R