INTELIGENȚĂ ARTIFICIALĂ

Sisteme inteligente

Sisteme care învaţă singure

– generative AI–

Laura Dioşan

Sumar

A. Scurtă introducere în Inteligenţa Artificială (IA)

C. Sisteme inteligente
- Sisteme care învaţă singure
  - Arbori de decizie
  - **Reţele neuronale artificiale**
  - kNN
  - Algoritmi evolutivi
  - Maşini cu suport vectorial
- Sisteme bazate pe reguli
- Sisteme hibride

B. Rezolvarea problemelor prin căutare
- Definirea problemelor de căutare
- Strategii de căutare
  - Strategii de căutare neinformate
  - Strategii de căutare informate
  - Strategii de căutare locale (Hill Climbing, Simulated Annealing, Tabu Search, Algoritmi evolutivi, PSO, ACO)
  - Strategii de căutare adversială

Materiale de citit şi legături utile

- https://www.tensorflow.org/text/tutorials/word2vec
- *https://radimrehurek.com/gensim/models/word2vec.html*
- *https://www.ruder.io/word-embeddings-1/*
- *https://aylien.com/blog/a-review-of-the-recent-history-of-natural-language-processing?ref=ruder.io*
- *https://nlp.stanford.edu/~manning/ - a lot of lectures about NLP*

De ce reprezentari vectoriale ale intelesului unui cuvant/text?

- Permit determinarea similaritatii intre cuvinte/texte
  - ***fast*** is similar to ***rapid***
  - ***tall*** is similar to ***height***

Question answering:

- *Q: "How **tall** is Mt. Everest?"*

*Candidate A: "The official **height** of Mount Everest is 29029 feet"*

Intuitia din spatele similaritatii

- Exemplu: Ce este o Sirrus X 3.0?

*The **Sirrus X 3.0** invites you to explore beyond boundaries. With confidence-inspiring tires, an upright riding position, and intuitive components, **Sirrus X 3.0** is your ticket to adventure.*

- Din context, o persoana poate ghici ca Sirrus X 3.0 este un model de bicicleta

- Pentru un algoritm, intuitia e ca **doua cuvinte sunt similar daca ele sunt folosite in context similare**

Diferite reprezentari vectoriale pentru text

- Reprezentari rare (*sparse*):
  1. Mutual-information weighted word co-occurrence matrices

- Reprezentari dense (compacte) :
  2. Singular Value Decomposition (si Latent Semantic Analysis)
  3. Neural-network-inspired models (skip-grams, CBOW)
  4. Altele (e.g. brown clusters)

Vectori rari *vs*. densi

- Vectorii → matricea de co-ocurenta a termenilor
  - **lungi** (length |V|= 20,000 -> 50,000)
  - **rari** (f multe elemente sunt 0)
- Alternativa: vectori invatati (prin AI/ML)
  - scurti (length 200-1000)
  - **densi** (multe elemente nu sunt 0)
- De ce vectori densi?
  - Vectorii scurti -> folositi mai usor ca si *features* in algoritmii de invatare (mai putini coeficienti de invatat)
  - Vectorii densi pot generaliza mai bine, captand sinonimia termenilor
    - Bike – scooter
    - Car – automobile
    - House – apartment

Modele de predictie (invatare) a reprezentarilor

- Evolutie
  - Word-level
    - 2003 - N-gram Neural language model (Montreal - Bengio)
      - https://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf
      - Probability of the target word based on previous k words
      - Estimation of the probability by an ANN -> prediction of the next word in a sequence
    - 2008 – multi-task model (Princeton – Collobert)
      - https://ronan.collobert.com/pub/matos/2008_nlp_icml.pdf
      - Probability of MORE target words (a sequence of words)
      - Estimation of the probability by an ANN –similar to Bengio's model,
    - 2013 – word2vec (Google – Mikolov)
      - https://papers.nips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf
      - Continous BOW model -> try to predict a word based on its context (neighbours); input = neighbour words, output = target word
      - Skip-gram model -> try to predict context (neighbours) of a word; input = target word; output = neighbour words;
      - Visualisation of embeddings https://ronxin.github.io/wevi/
      - Trained vectors https://radimrehurek.com/gensim/models/word2vec.html or https://github.com/3Top/word2vec-api#where-to-get-a-pretrained-models
  - Sentence (document) level (2014 - …)
    - 2014 - Paragraph embedding https://arxiv.org/abs/1405.4053
  - Contextual word-vectors (Word vectors compress all contexts into a *single vector) (2016 - …)*
    - 2016 context2vec https://www.aclweb.org/anthology/K16-1006.pdf
    - 2017 tagLM https://arxiv.org/abs/1705.00108
    - 2017 CoVe https://papers.nips.cc/paper/2017/hash/20c86a628232a67e7bd46f76fba7ce12-Abstract.html
    - 2018 ELMo https://www.aclweb.org/anthology/N18-1202/
    - 2018 ULMFiT https://arxiv.org/abs/1801.06146
    - 2019 BERT https://arxiv.org/abs/1810.04805

Modele de predictive (invatare) a reprezentarilor

- **Modelul Word2vec**
  - **Skip-gram** (Mikolov et al. 2013a), **CBOW** (Mikolov et al. 2013b)
  - Se invata reprezentari, numite embeddings, ca parte din procesul de predictive/generare a textului
  - Se antreneaza o retea neuronala pentru prezicerea urmatorului cuvant
  - Avantaje
    - Rapid, simplu de antrenat
    - Modele gata antrente disponibile online

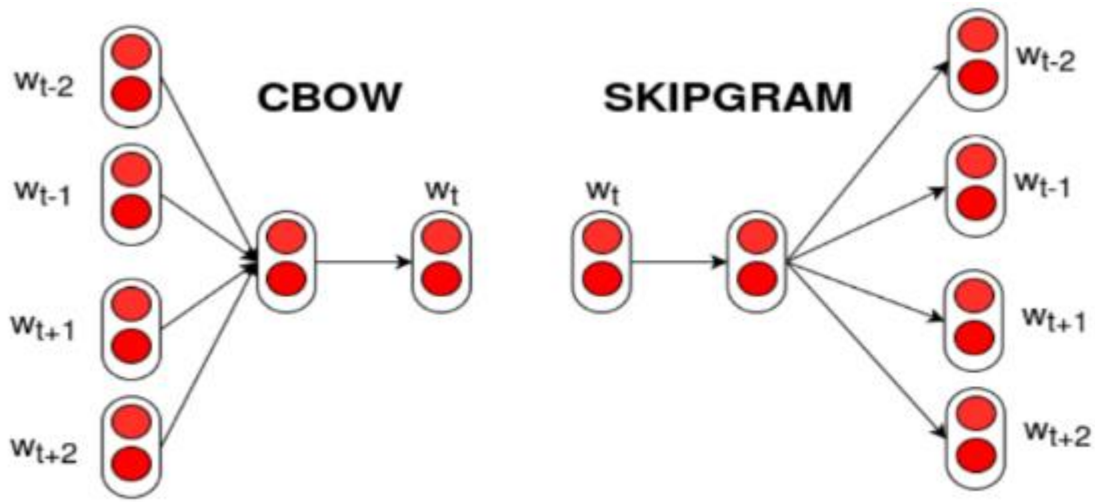
Invatare supervizata fara etichetare manuala de date

- Text: "*She rides a bike. He rides a scooter in park. My father drives a motorcycle. …*"
- Vocabular V = {*she, ride, bike, he, scooter, park, my, father, drive, motorcycle*}
- Perechi (context, cuvant)

Negative sampling

[she ride] bike	1		[she ride] he	0
[ride bike] he	1		[she ride] scooter	0
[he ride] scooter	1		[she ride] park	0
[ride scooter] park	1		[she ride] my	0
[my father] drive	1		[she ride] father	0
[father drive] motorcycle	1		[she ride] drive	0
			[ride bike] scooter	0
			[ride bike] park	0
			…	
			[father drive] she	0

Arhitecturi

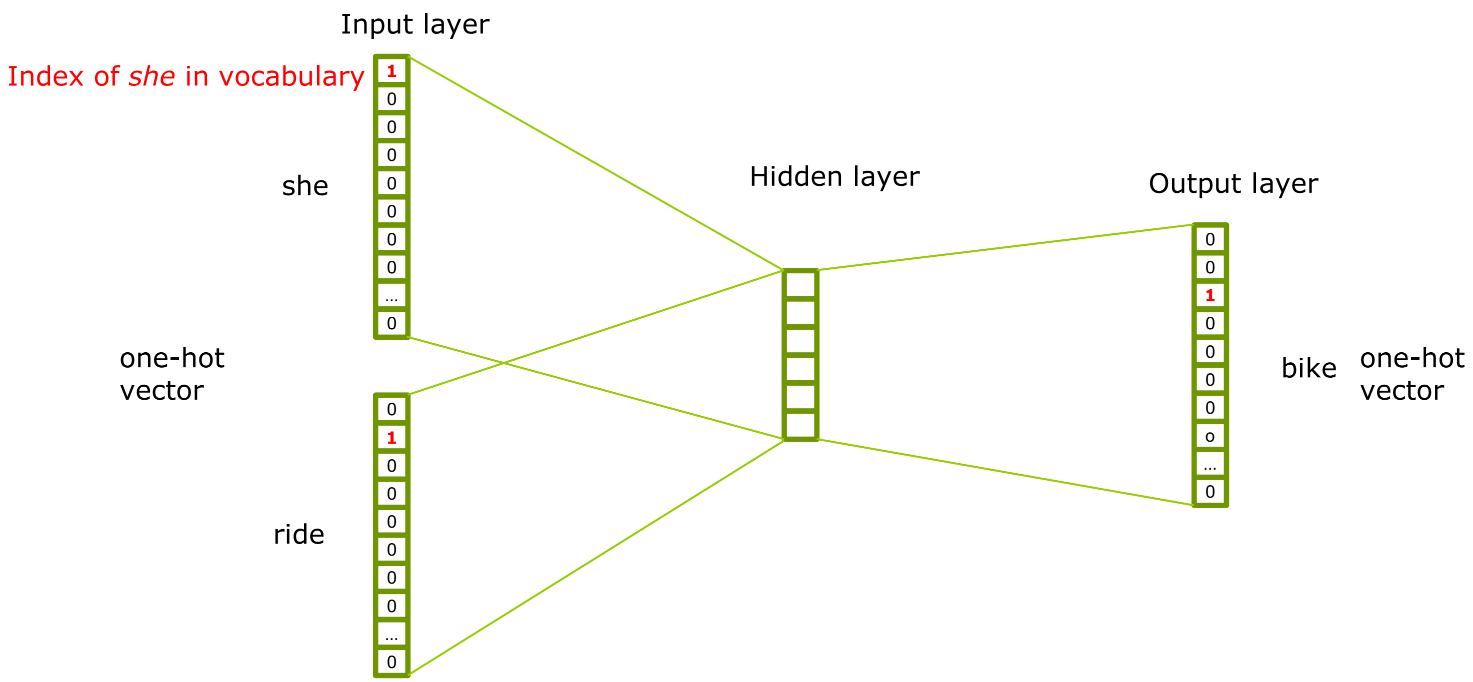


$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} logp(w_t|w_{t+j})$$

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} logp(w_{t+j}|w_t)$$

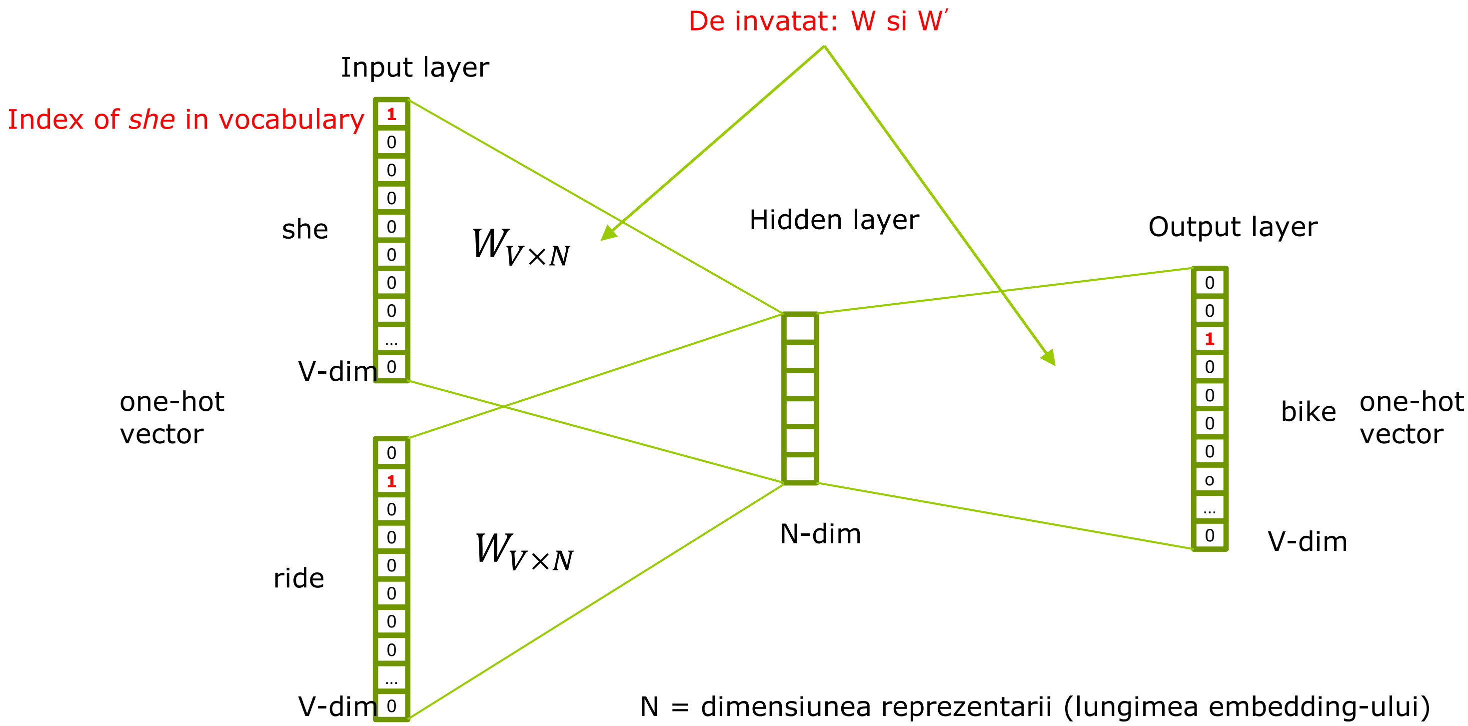
CBOW (Continuous Bag of Words)

	she	ride	bike	he	scooter	park	my	father	drive	motorcycle
She	1	0	0	0	0	0	0	0	0	0
Ride	0	1	0	0	0	0	0	0	0	0
Bike	0	0	1	0	0	0	0	0	0	0



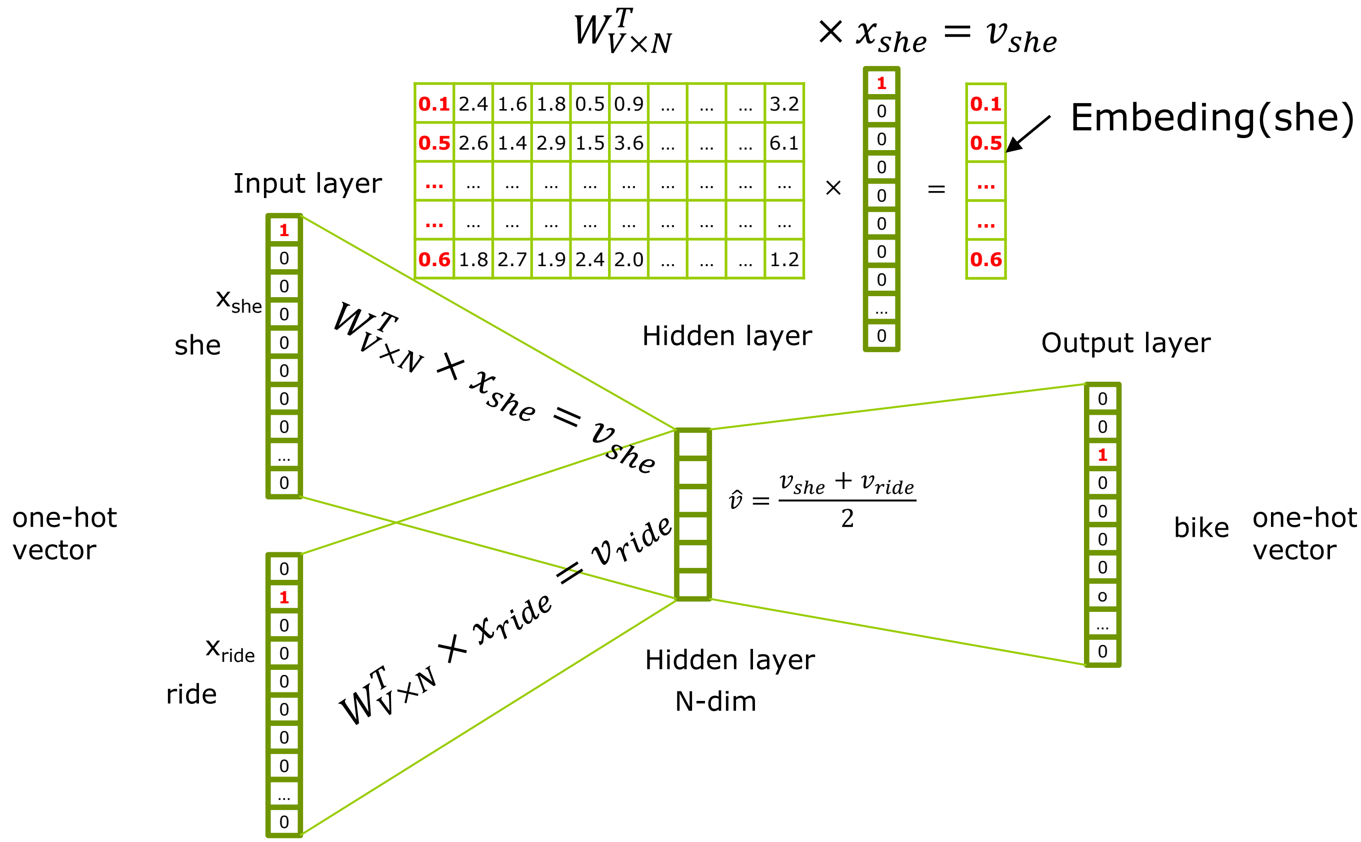
CBOW (Continuous Bag of Words)

	she	ride	bike	he	scooter	park	my	father	drive	motorcycle
She	1	0	0	0	0	0	0	0	0	0
Ride	0	1	0	0	0	0	0	0	0	0
Bike	0	0	1	0	0	0	0	0	0	0

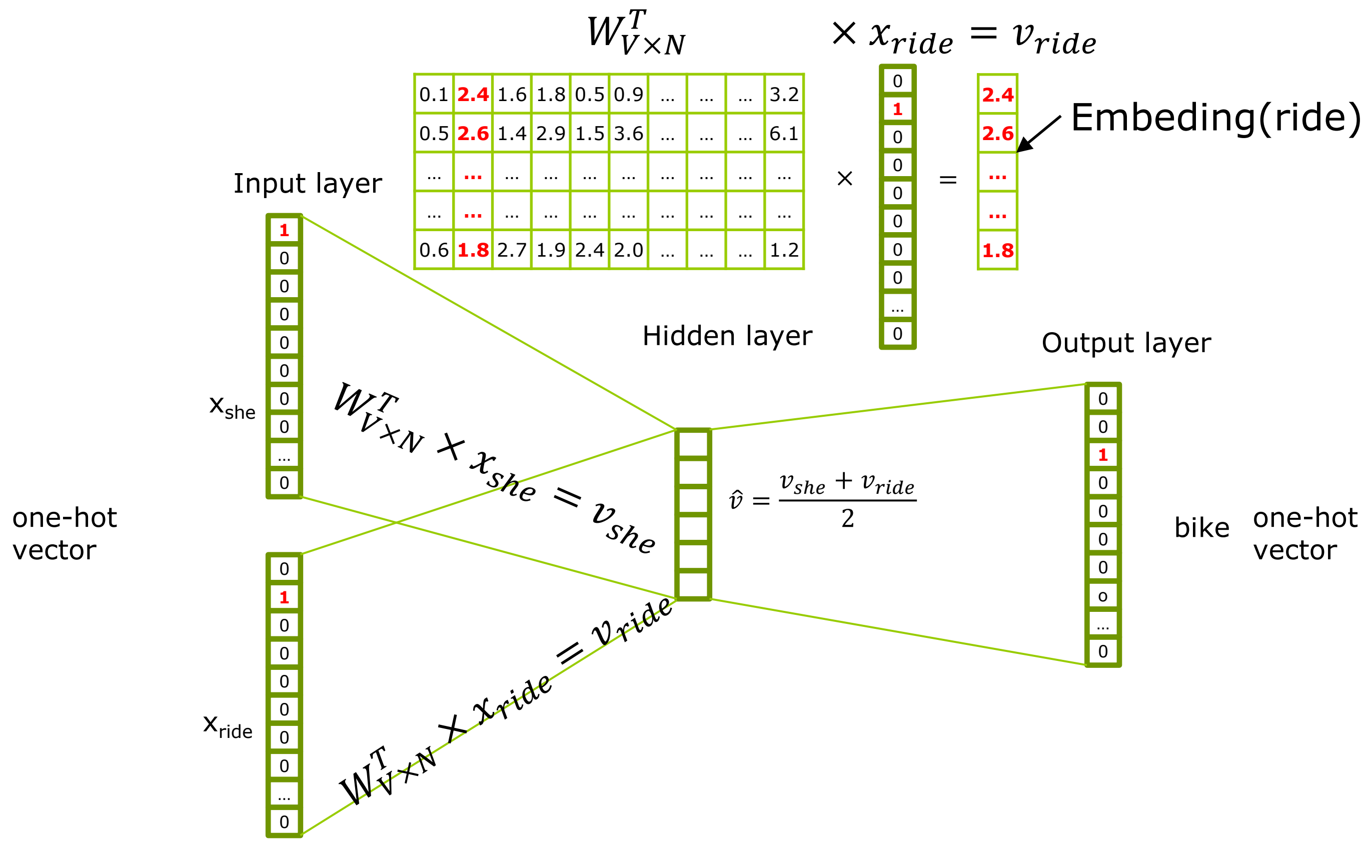


N = dimensiunea reprezentarii (lungimea embedding-ului)

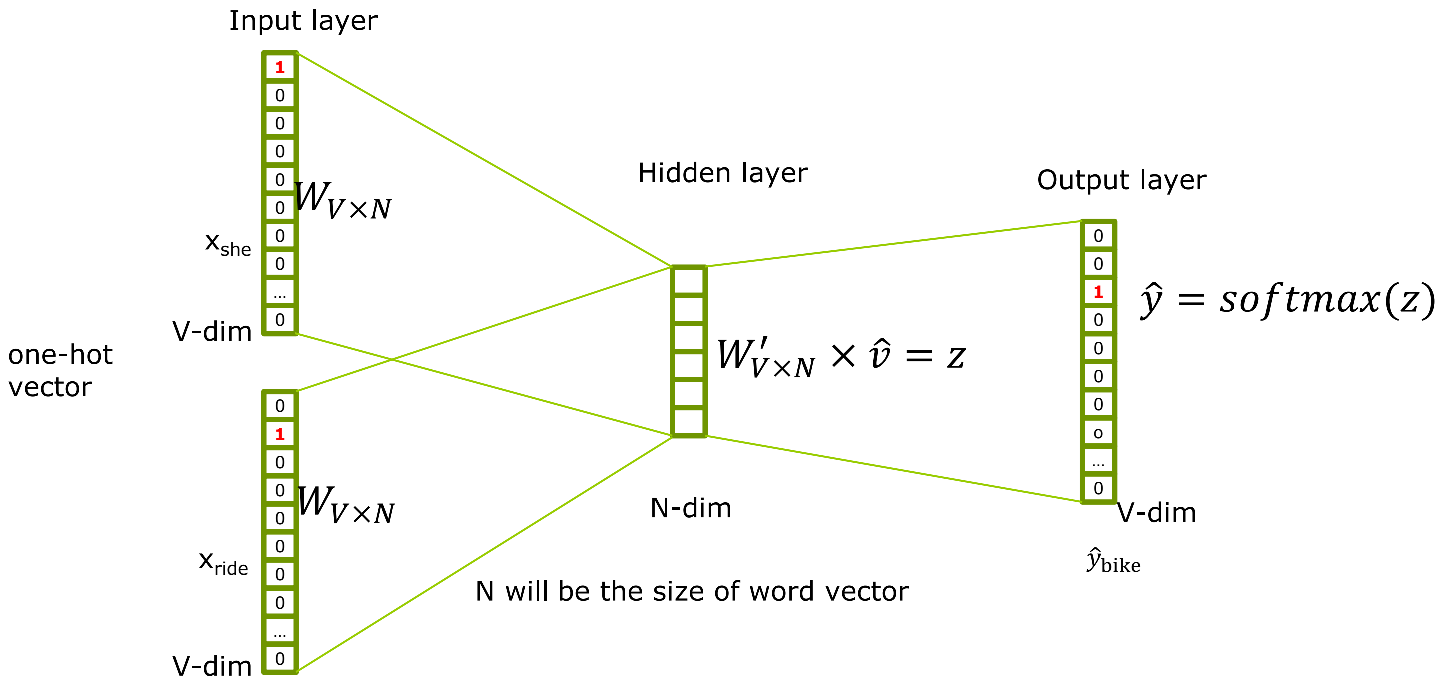
CBOW (Continuous Bag of Words)



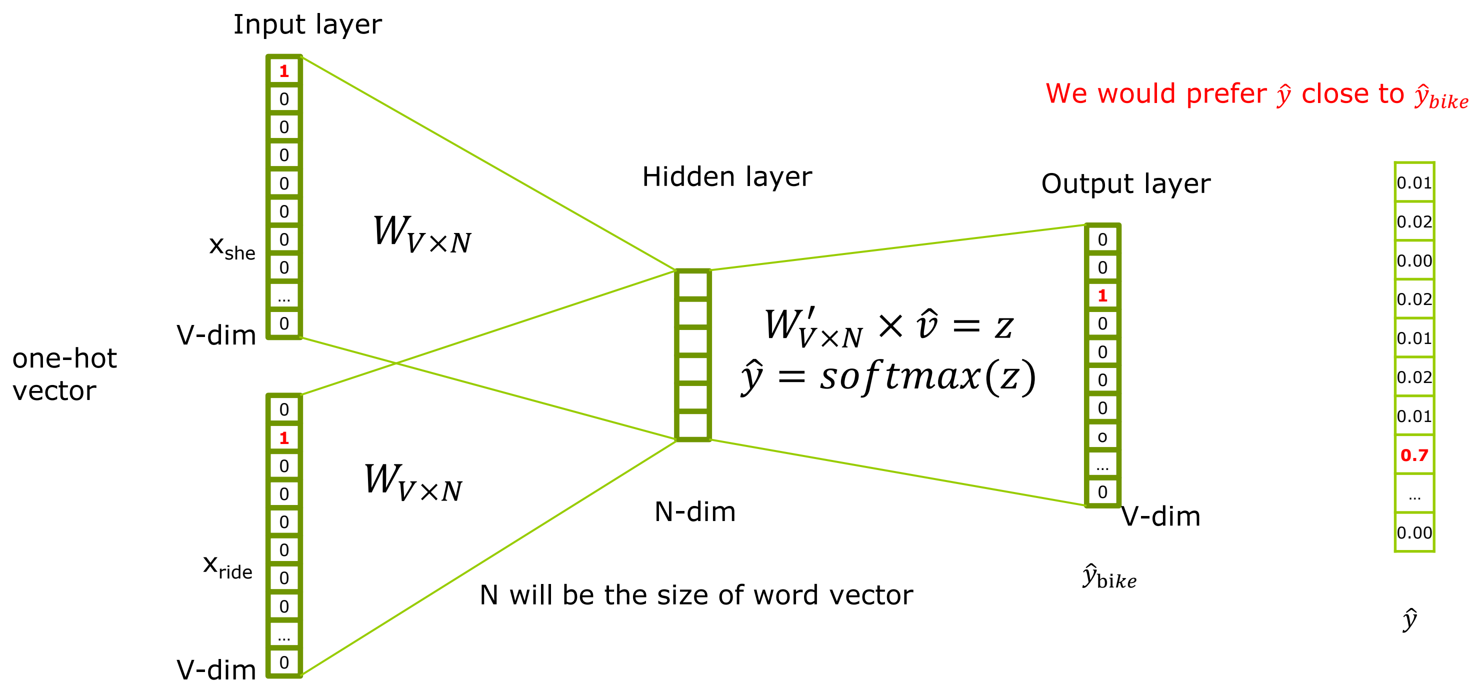
CBOW (Continuous Bag of Words)



CBOW (Continuous Bag of Words)



CBOW (Continuous Bag of Words)



Cursul următor

A. Scurtă introducere în Inteligenţa Artificială (IA)

B. Sisteme inteligente
- **Sisteme care învaţă singure**
  - Arbori de decizie
  - Reţele neuronale artificiale - transformers
  - Maşini cu suport vectorial
  - Algoritmi evolutivi
- Sisteme bazate pe reguli
- Sisteme hibride

C. Rezolvarea problemelor prin căutare
- Definirea problemelor de căutare
- Strategii de căutare
  - Strategii de căutare neinformate
  - Strategii de căutare informate
  - Strategii de căutare locale (Hill Climbing, Simulated Annealing, Tabu Search, Algoritmi evolutivi, PSO, ACO)
  - Strategii de căutare adversială

- Informaţiile prezentate au fost colectate din diferite surse de pe internet, precum şi din cursurile de inteligenţă artificială ţinute în anii anteriori de către:
  - Conf. Dr. Mihai Oltean – www.cs.ubbcluj.ro/~moltean
  - Lect. Dr. Crina Groşan - www.cs.ubbcluj.ro/~cgrosan
  - Prof. Dr. Horia F. Pop - www.cs.ubbcluj.ro/~hfpop
  - Prof. Dr. Radu Ionescu - https://raduionescu.herokuapp.com/