ROAI - CLIP and Multimodal Models

Mihai Andrei Gherghinescu

FMI, UVT, Timisoara

My background

- **Software Engineer la Microsoft**, specializat în dezvoltarea infrastructura pentru echipele de data science.
- 4 ani de experiență în domeniul IT.
- Student masterand în ultimul an la **Universitatea de Vest din Timișoara**, programul *Artificial Intelligence and Distributed Computing*.
- Interese de cercetare::
  - Aplicații medicale ale inteligenței artificiale
  - Sisteme multi-agent

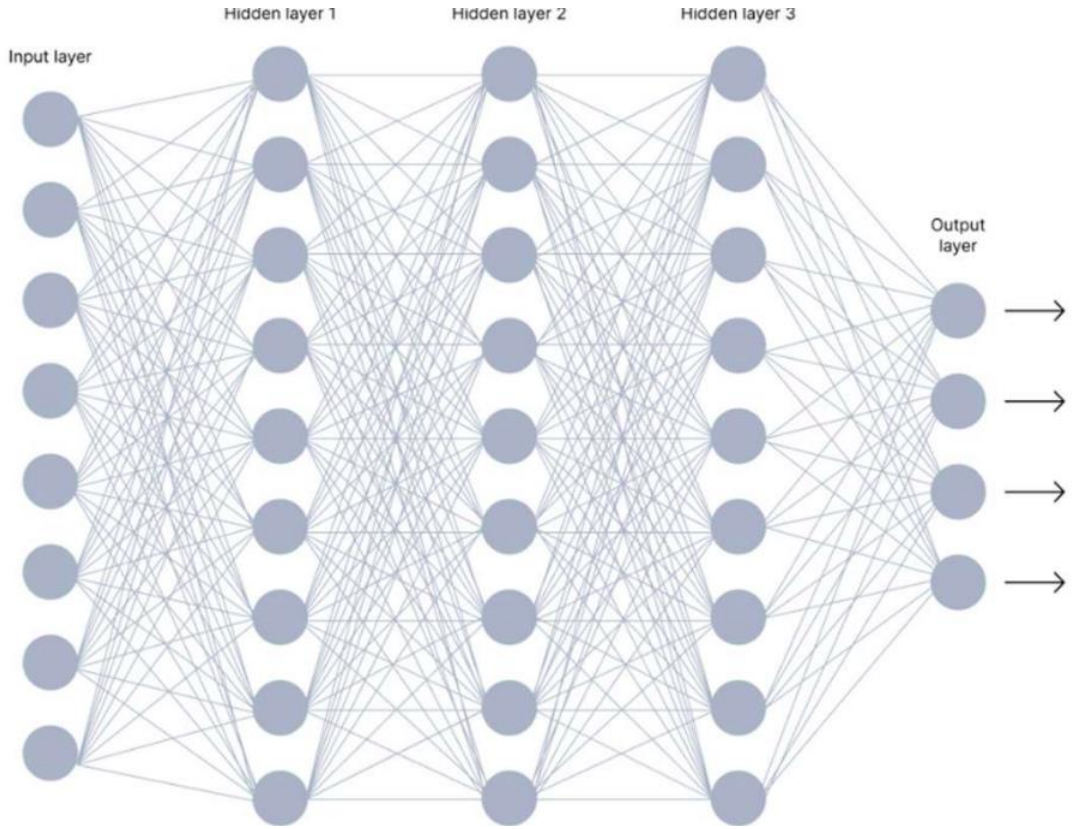
Agenda

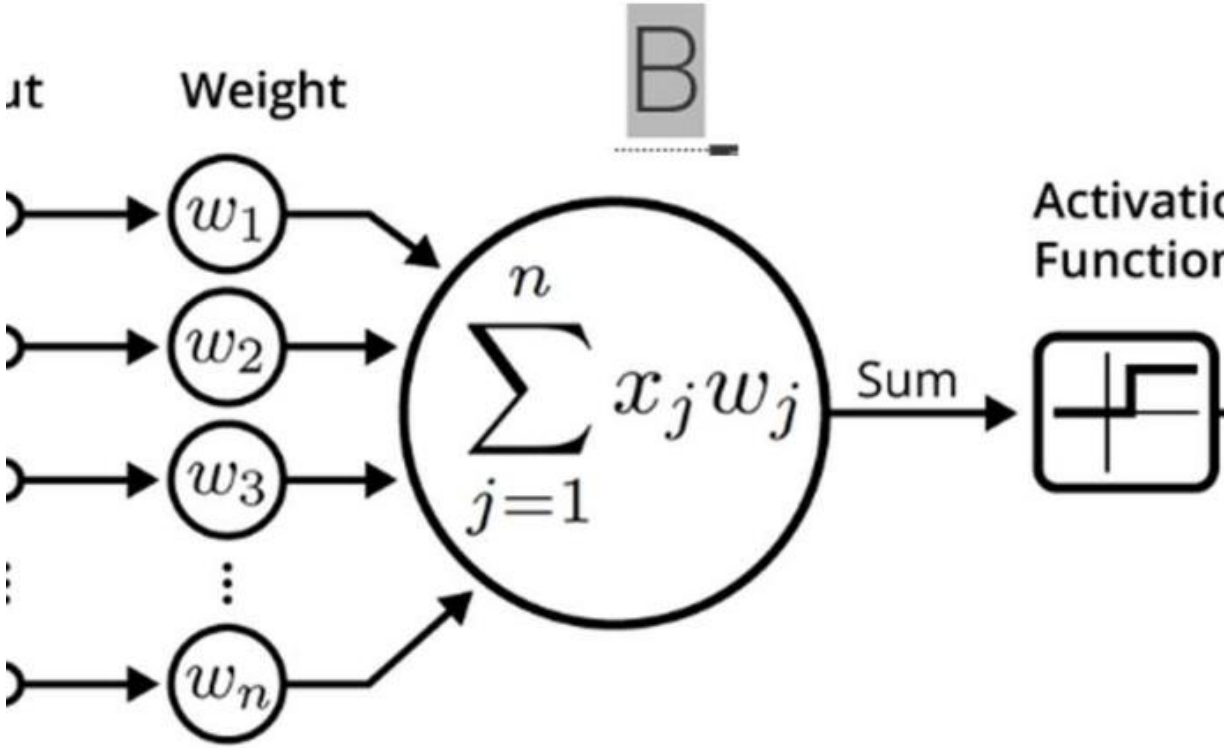
Retele neuronale
Retele neuroanle convolutionale
U-NET
Modele bazate pe difuzie
Transformeri si mecanisme de atentie
CLIP
Stable diffusion si DALL-E
Sesiune de coding

Rețele neuronale

- Inspirate din medicină.
- Alcătuite din mai multe noduri și conexiuni multistratificate.
- Fiecare conexiune reprezintă o funcție liniară ce se adaptează în timpul antrenamentului pentru a se plia pe datele de antrenare.
- În fiecare strat, există o funcție de activare ce decide cât de mult contribuie fiecare neuron și, implicit, la rezultatul final al rețelei.
- Folosite pentru:
  - Clasificare
  - Prezicere de valori

Exemplu vizual

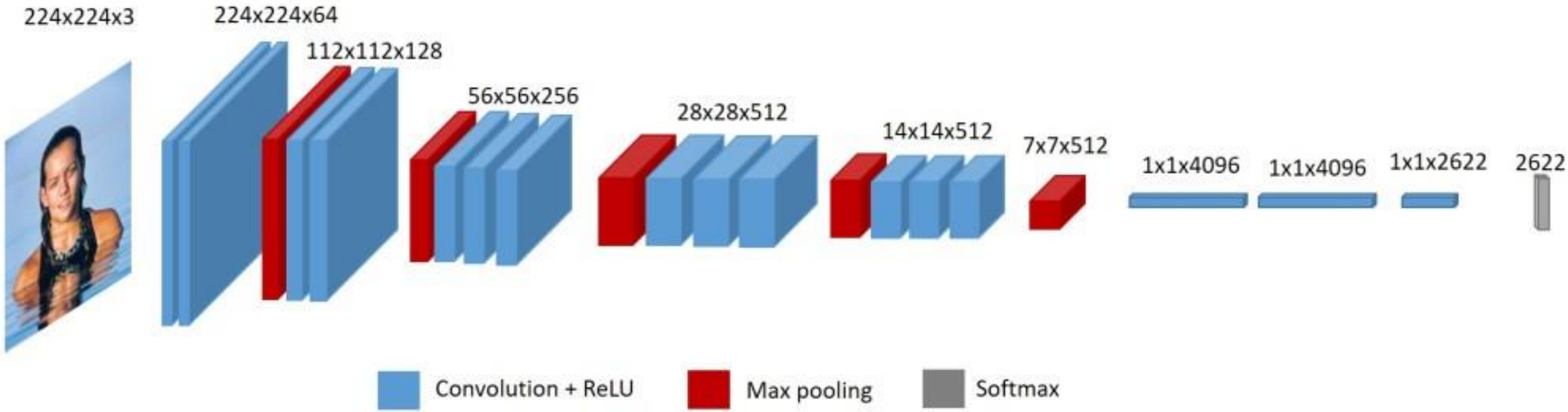




Rețele neuronale convoluționale

- Sunt un tip de rețele neuronale care primesc imagini ca input.
- Folosesc layere de convoluție si pooling pentru a extrage și comprima informația din imagini.
- Aplicații:
  - Clasificare
  - Detectarea obiectelor
  - Segmentare

Exemplu rețea neuronala convoluționala: arhitectura VGG16



Exemplu aplicații

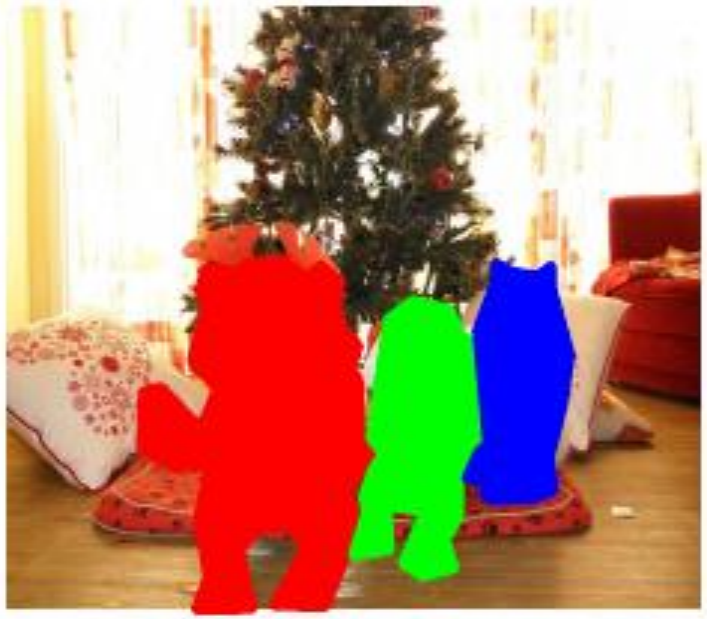
Classification + Localization

CAT

Object Detection

DOG, DOG, CAT

Instance Segmentation



DOG, DOG, CAT

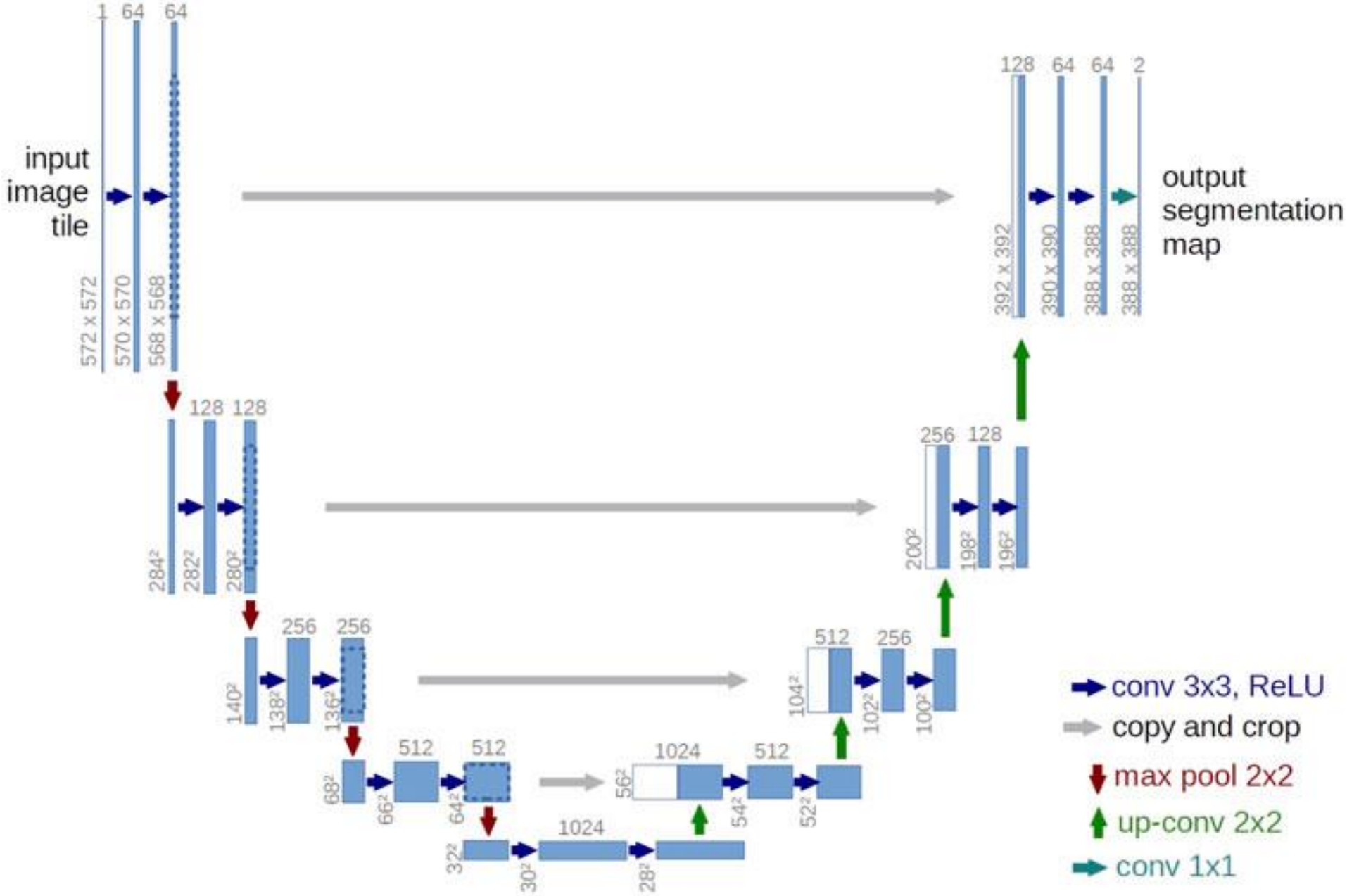
Single Object

Multiple Object

U-NET

- Rețea convolutionala care are au ca scop sprijinirea procesului de segmentare a imaginilor medicale.
- Primește ca input o imagine și o procesează pentru a evidenția regiunile de interes.
- Se bazează pe o arhitectură de tip **encoder-decoder**.
- **Structură:**
- **Encoder**: Format din straturi convoluționale care reduc treptat dimensiunea imaginii, extrăgând caracteristici esențiale.
- **Decoder**: Conține straturi de upsampling care reconstruiesc imaginea segmentată, plecând de la reprezentarea comprimată. În acest proces, sunt folosite și conexiuni directe cu straturile encoder-ului (skip connections), pentru a păstra detalii importante din imaginea originală.

Arhitectura U-NET



Modele bazate pe difuzie

- Se bazează pe conceptul de difuzie din termodinamică.
- Primesc ca și input o imagine la care se adaugă treptat „noise”.
- Învață să reconstruiască imaginea prin prezicerea „noise”-ului adăugat.
- Folosesc la bază arhitectura UNET.
- Aplicații:
  - Generare de imagini
  - Augmentare de imagini

Mecanism de difuzie

Forward SDE (data → noise)

$\mathbf{x}(0)$ — $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$ → $\mathbf{x}(T)$

score function

$\mathbf{x}(0)$ ← $\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$ — $\mathbf{x}(T)$

Reverse SDE (noise → data)

Transformeri

- Se bazează pe o arhitectură de tip encoder-decoder.
- Primește ca și input tokeni pe post de „cuvinte” și încearcă să prezică următorul token/„cuvânt”.
- Introduc un mecanism de atenție care se bazează pe calcularea similarității dintre tokeni pentru a determina contextul curent.
- Aplicații:
  - Generare de text
  - Traducere de text
  - Sumarizare de text

Self-attention

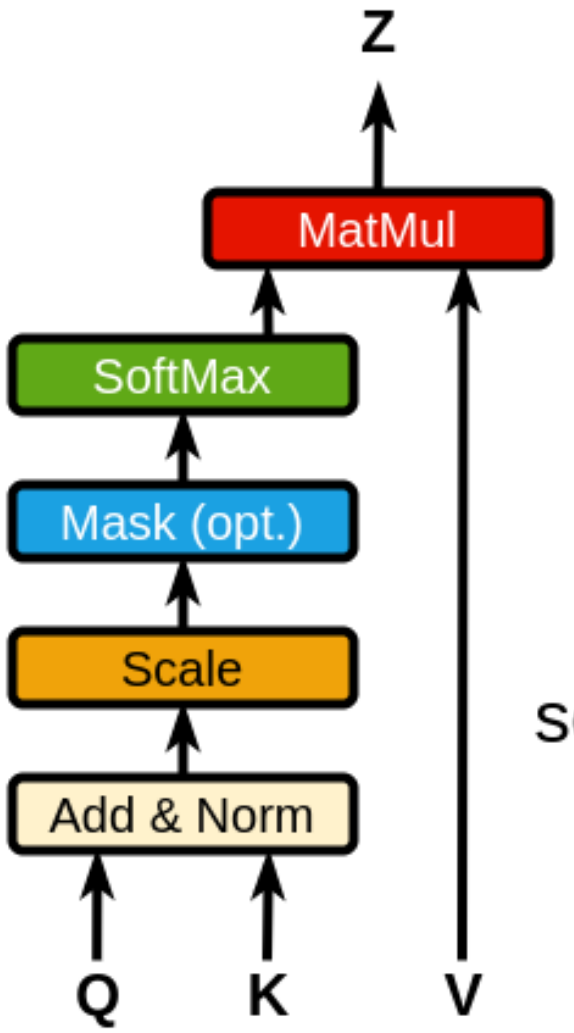
Atenție de tip single-head:

- Primește ca și input o fereastră de token-uri.
- Are la bază trei matrici antrenabile: Query (Q), Valoare (V) și Cheie (K).
- Pe lângă embedding-urile token-urilor, ia în considerare și poziția acestora.
- Încearcă să adauge context la token-ul curent, luând în considerare similaritatea față de token-urile precedente.

Atenție de tip multi-head:

- Echivalent cu mai multe layere de tip single-head ce rulează în paralel.
- Adaugă context al mai multor ferestre la cuvântul curent.

Calculul contextului relativ la token-ul curent



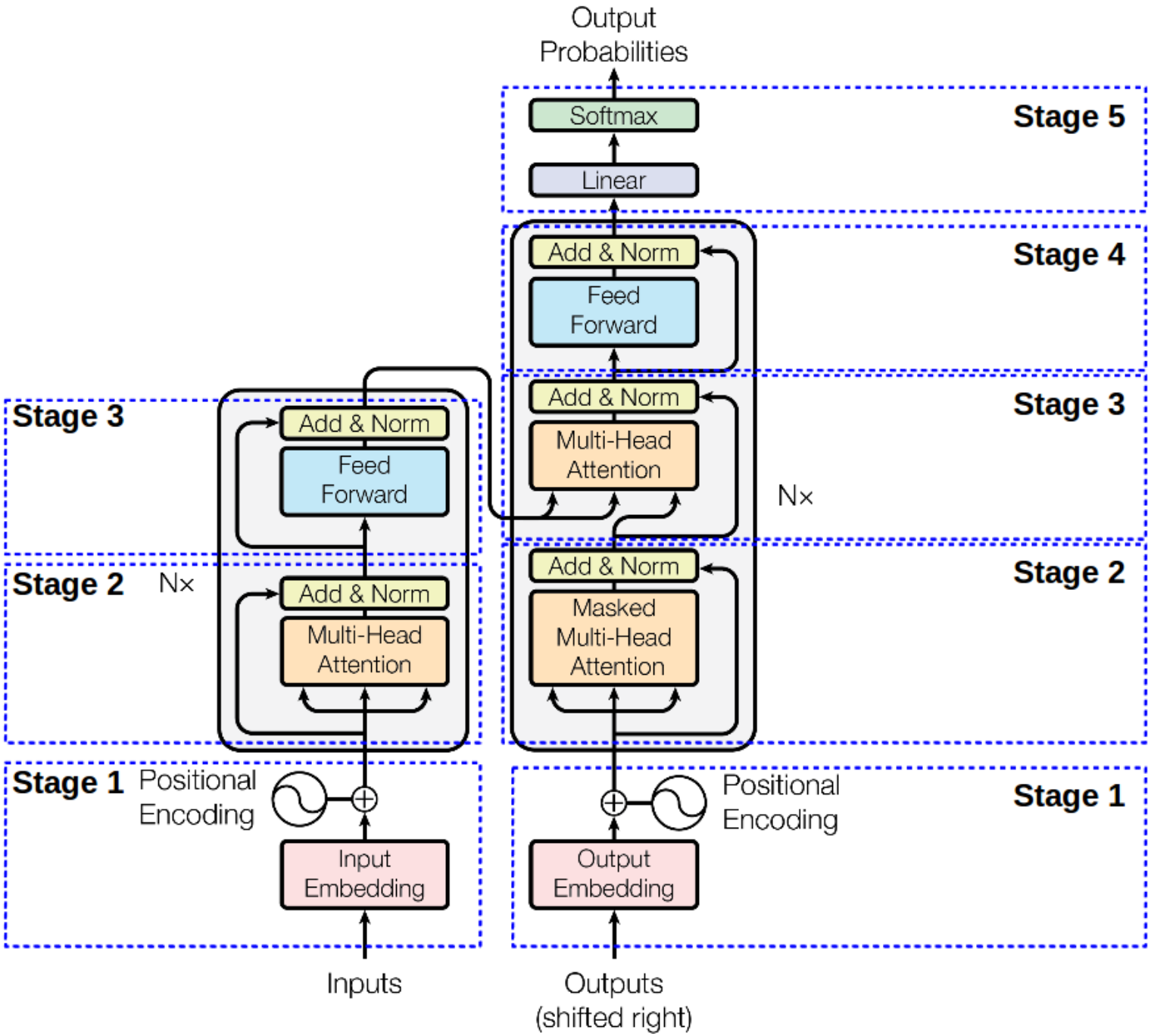
SCALED DOT-PRODUCT

```
from transformerx.layers import DotProductAttention
```

$$\text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) V = Z$$

soran-ghaderi.github.io
linkedin.com/in/soran-ghaderi

Arhitectura de tip transfomer



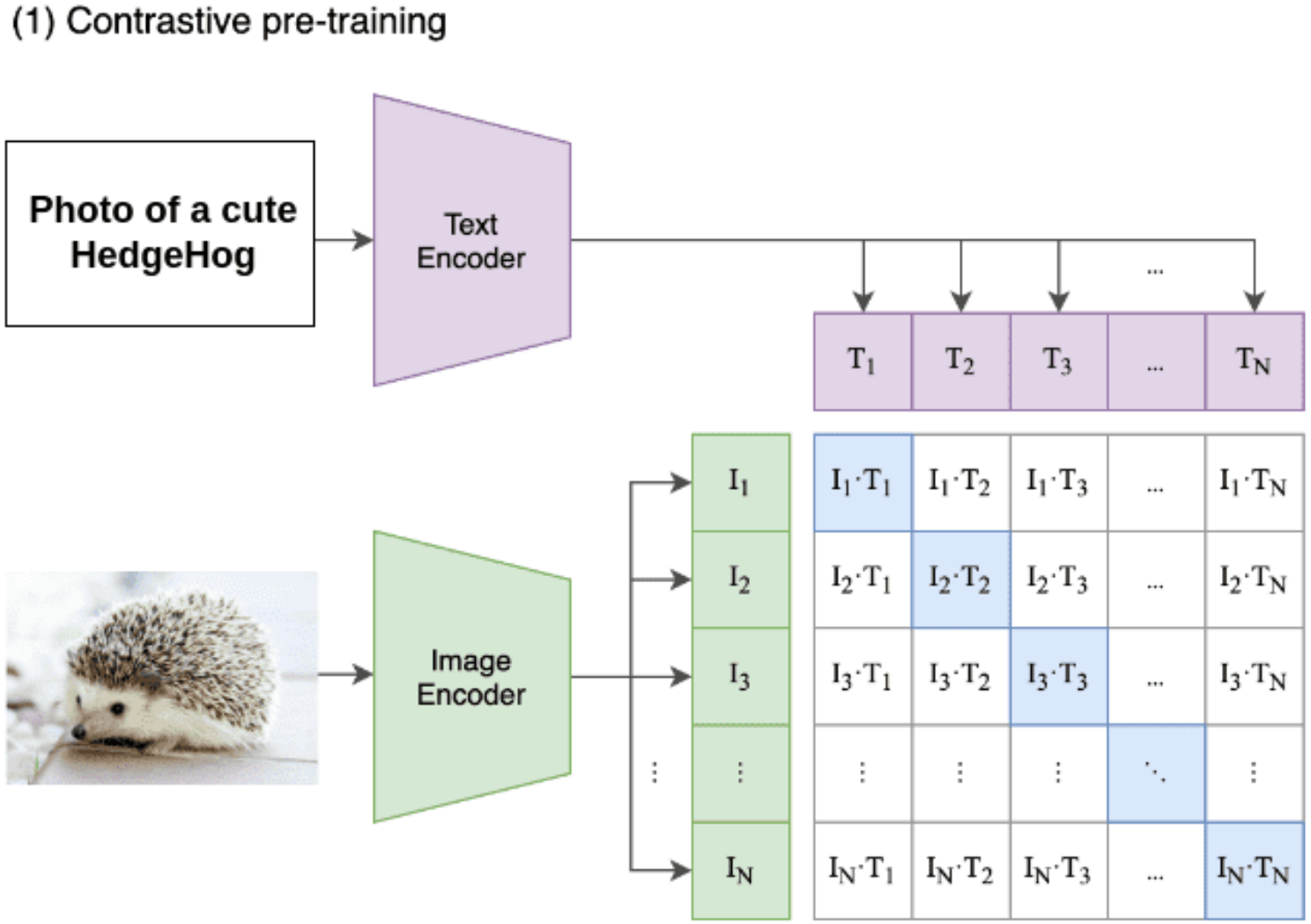
Cross-attention

- Bazat pe aceleași concepte ca și mecanismul de self-attention.
- Adaugă atenție față de un alt tip de date.
- Similaritatea se calculează între tipuri de date eterogene.
- Deschide conceptul de modele multimodale:
  - Text to image
  - Text to audio

Contrastive Language–Image Pre-training (CLIP)

- Permite asocierea dintre text și imagini
- Model antrenat pe un dataset de 400M de perechi de tipul imagine-text
- Model bazat pe 2 tipuri de encoderi:
  - VIT/CNN ca și encoder pentru imagini
  - Transformer ca și encoder pentru text
- Încearcă să minimizeze similaritatea (cosine similarity) dintre embedding-urile textului și embedding-urile imaginii corespunzătoare.
- Aplicații:
  - Clasificare Zero-Shot
  - Căutare de imagini pe baza unui prompt
  - Generare de imagini ghidată de text

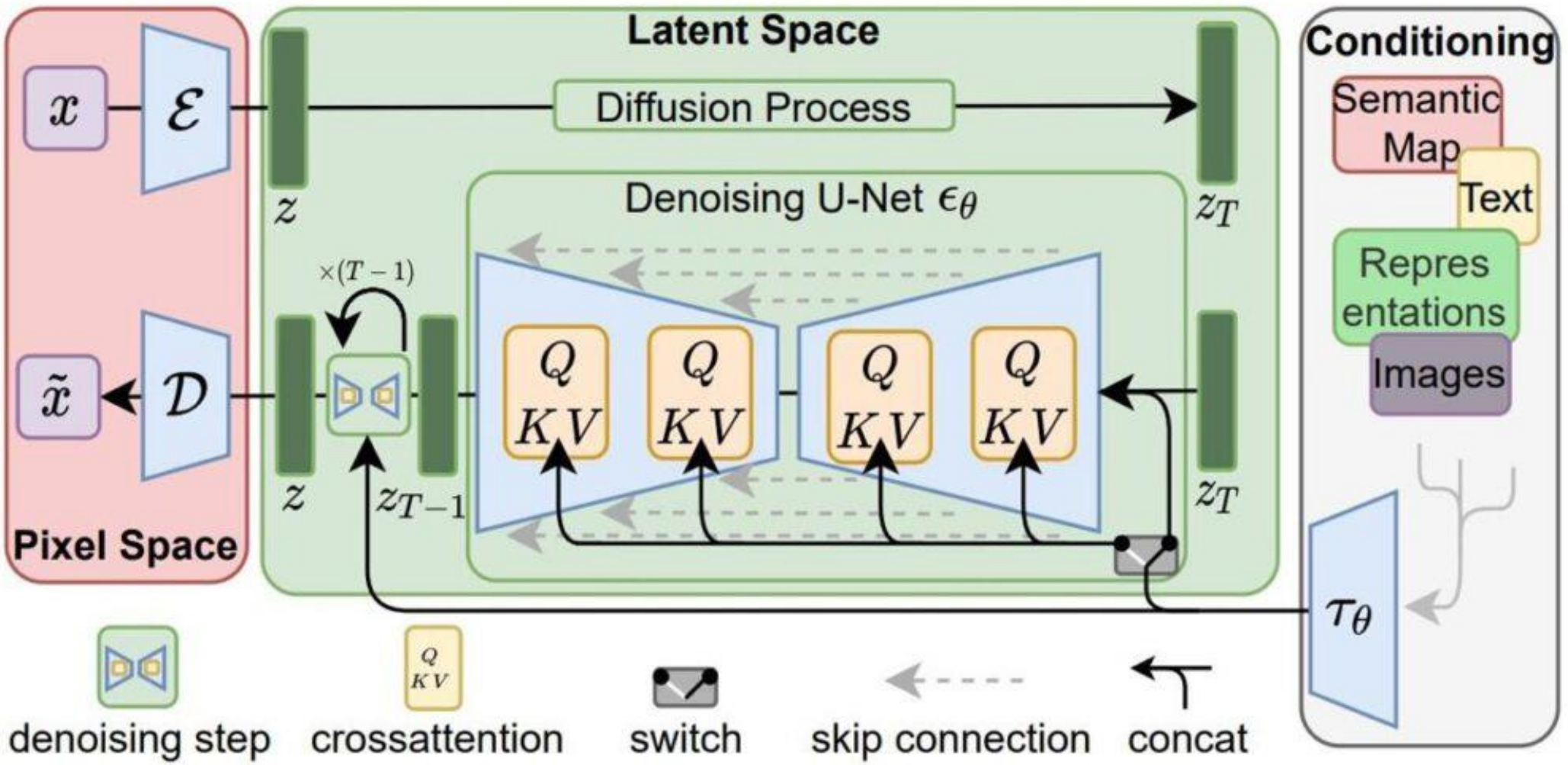
Exemplu de antrenament al arhitecturi CLIP



Stable diffusion si DALL-E

- Bazat pe modele bazate pe difuzie care se folosesc de arhitectura CLIP.
- Adaugă un strat suplimentar de atentie (cross-attention) în interiorul UNET-ului.
- Straturile de atenție permit modelului să alinieze trăsăturile vizuale cu semnificația semantică din text.
- Fiecarui pas ce adauga "noise" ii este concatenat embedding-urile textului folosit pentru ghidare.
- Stable Diffusion este open source, disponibil pe Hugging Face.
- DALL·E este deținut de OpenAI și include un modul suplimentar de Speech-to-Image. Acest lucru este posibil prin integrarea unui model pretrained de Speech-to-Text în fluxul său.

Vizualizare a arhitecturi Stable Diffusion



Coding starts now

Link google collab:
https://colab.research.google.com/drive/1GzqusXWC585qXjOGOP0hhZ6IWgXMY9EM?usp=sharing

Contact

- Email: mihai.gherghinescu00@e-uvt.ro
- GitHub: https://github.com/GMihai00
- LinkedIn: https://www.linkedin.com/in/mihai-gherghinescu