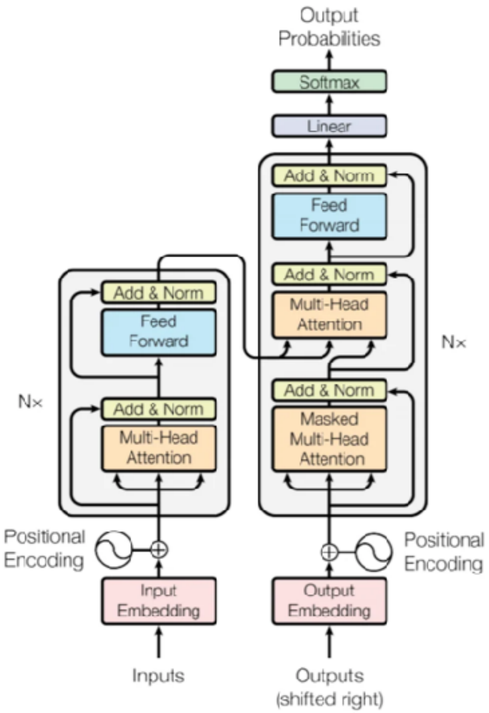






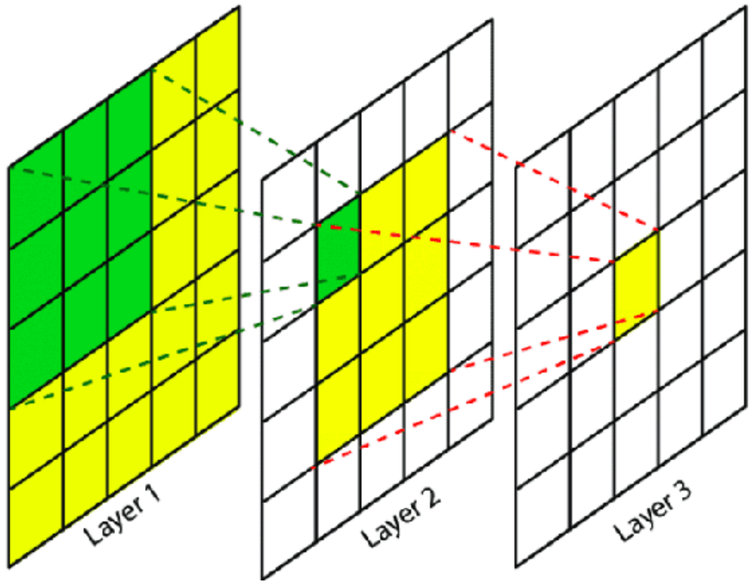

# 8. Előadás: Transformer architektúra

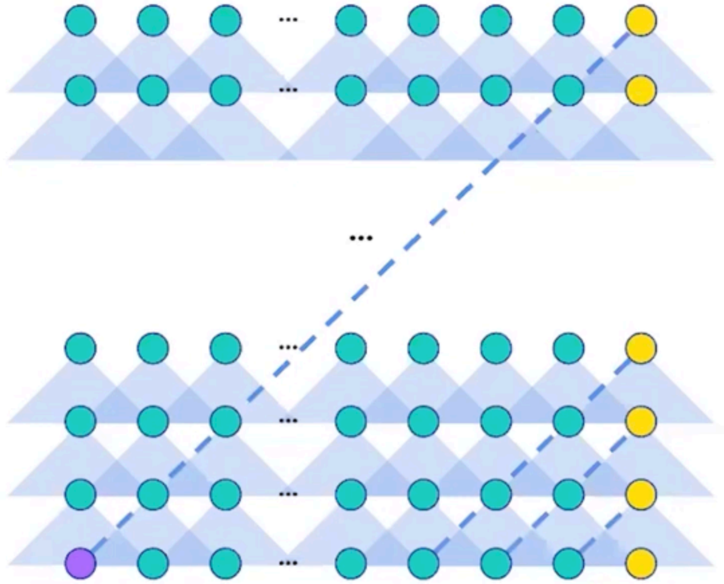
## Generatív AI és Inverz Módszerek a Képszintézisben

***BME-VIK IIT, 2026***

**Dr. Vaitkus Márton**

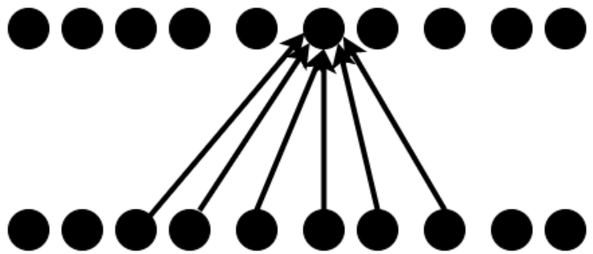
# Motiváció



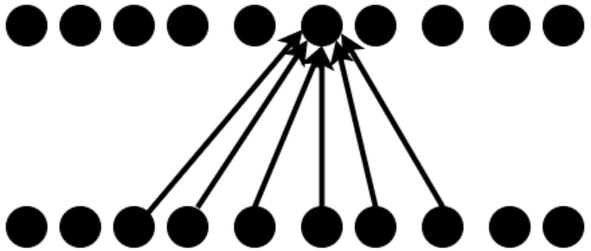
A konvolúciós rétegek *lokális* operációkat végeznek
Távoli régiók közötti kommunikáció csak több rétegen keresztül történhet!
(Pl. 224x224 kép széleit a 28.(!!!) 3x3 réteg látja egyszerre!)

# Motiváció

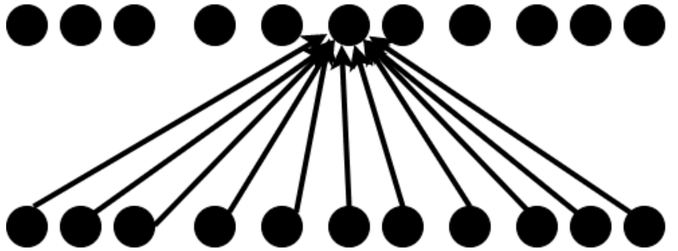


**<u>Konvolúció (CNN)</u>**:
lokális, *fix* kombináció

# Motiváció

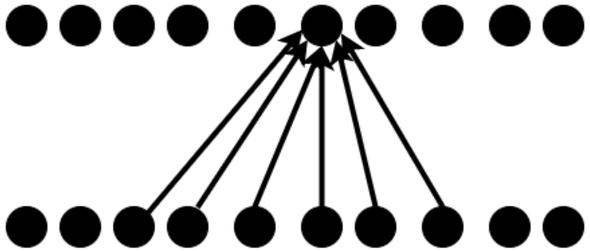


**<u>Konvolúció (CNN)</u>**:
lokális, *fix* kombináció

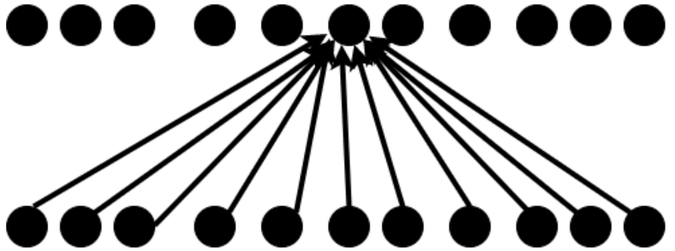


**<u>Fully Connected (MLP)</u>**:
globális, *fix* kombináció

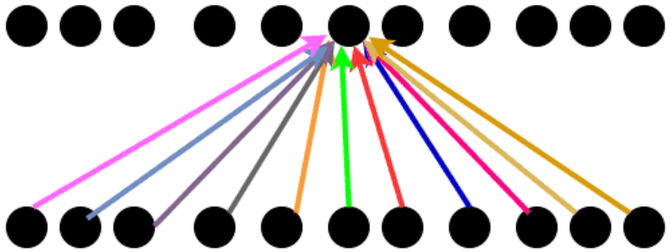
# Motiváció



**Konvolúció (CNN)**:
lokális, *fix* kombináció

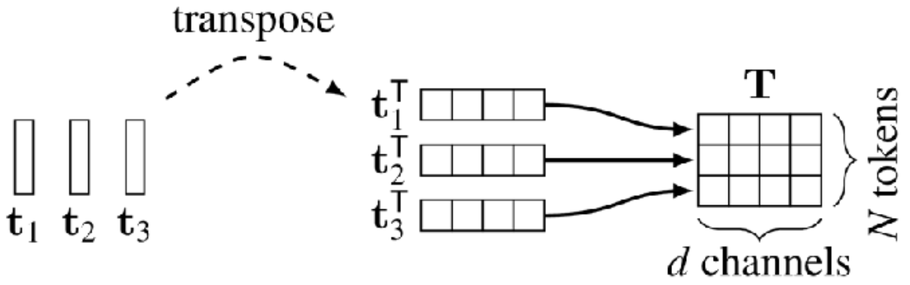


**Fully Connected (MLP)**:
globális, *fix* kombináció

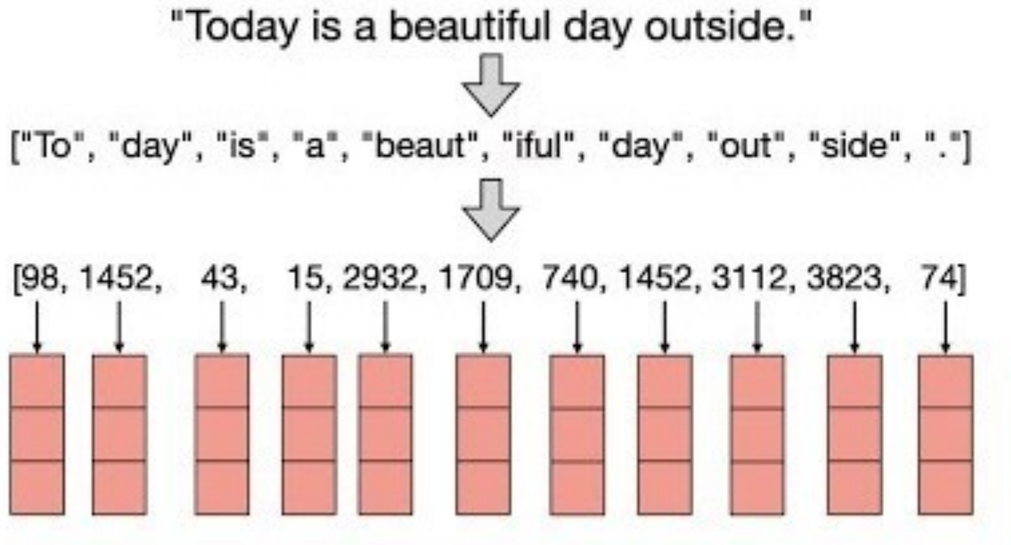


**Figyelem/Attention (Transformer):**
globális, *dinamikus* kombináció!

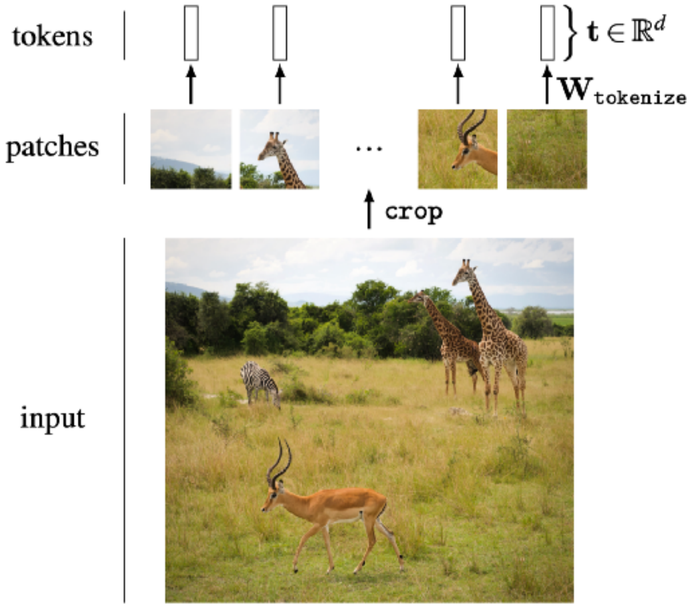
# Tokenek




Eddig a hálók bemenete/kimenete vektor vagy tenzor volt.
Új adattípus: **tokenek** — vektorok halmaza

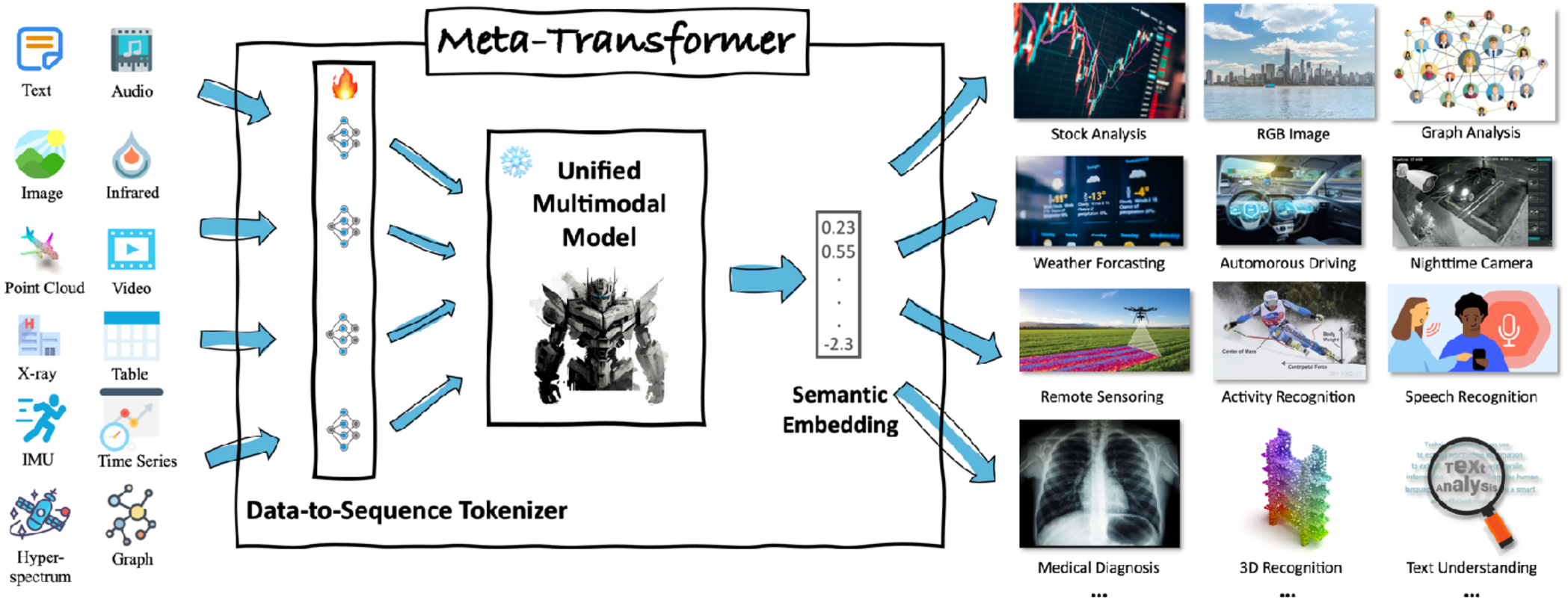



Szöveg tokenizálása




Kép tokenizálása

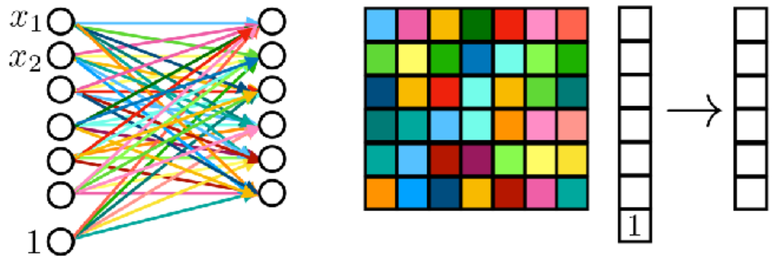
# Tokenek

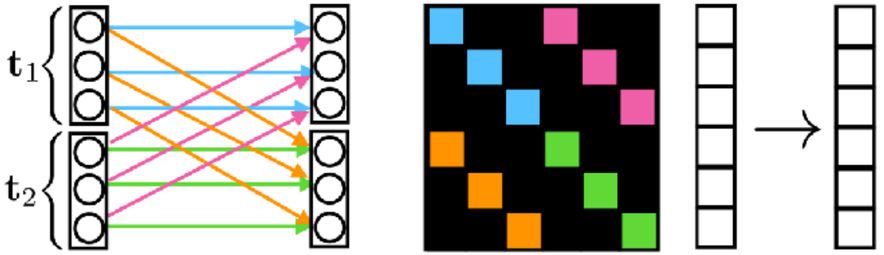


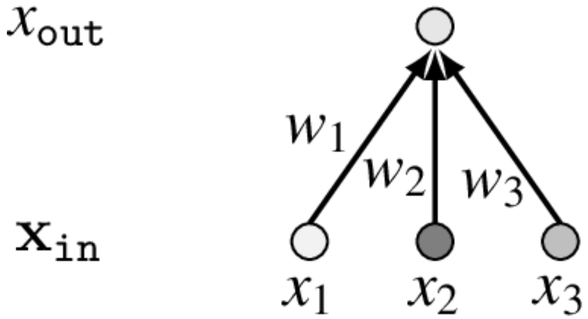

*Bármiből* lehet “token”! — **Multimodalitás**

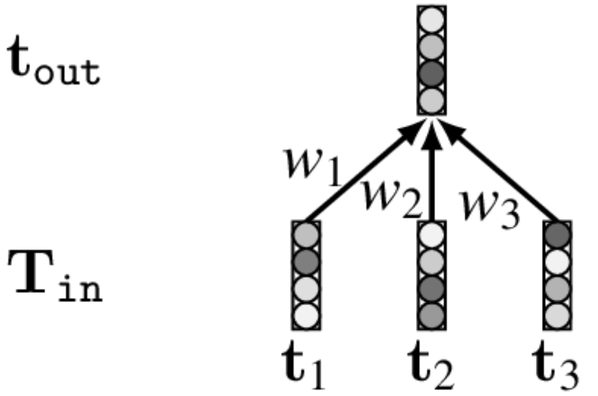
# Figyelem (Attention)

## Lineáris rétegek




$$x_{\text{out}} = w_1 x_1 + w_2 x_2 + w_3 x_3$$

Neuronok kombinációja

$$\mathbf{t}_{\text{out}} = w_1 \mathbf{t}_1 + w_2 \mathbf{t}_2 + w_3 \mathbf{t}_3$$

Tokenek kombinációja

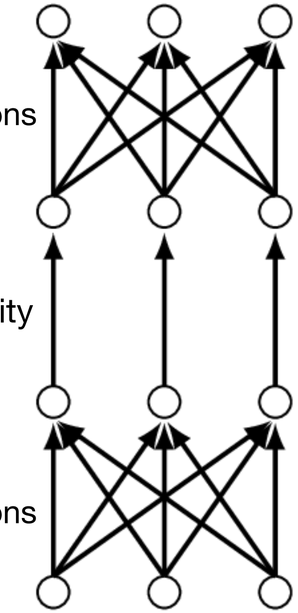
# Figyelem (Attention)

## Neurális vs. token hálózatok

$$\mathbf{x}_{\text{out}} = \mathbf{W}\mathbf{x}_{\text{in}} \triangleleft \text{linear combination of neurons}$$

$$\mathbf{x}_{\text{out}} = \begin{bmatrix} \text{relu}(x_{\text{in}}[0]) \\ \vdots \\ \text{relu}(x_{\text{in}}[N-1]) \end{bmatrix} \triangleleft \text{per-neuron nonlinearity}$$

$$\mathbf{x}_{\text{out}} = \mathbf{W}\mathbf{x}_{\text{in}} \triangleleft \text{linear combination of neurons}$$

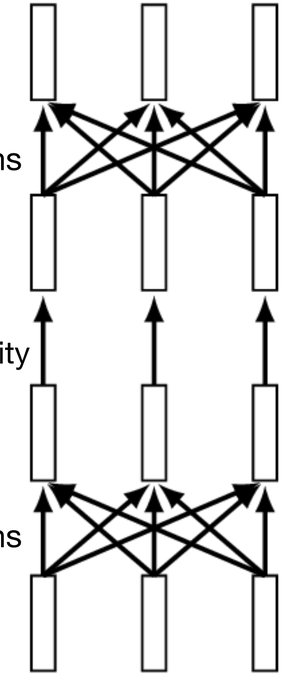


Neurális hálózat

$$\mathbf{T}_{\text{out}}[i,:] = \sum_{j=1}^{N} w_{ij}\mathbf{T}_{\text{in}}[j,:] \triangleleft \text{linear combination of tokens}$$

$$\mathbf{T}_{\text{out}} = \begin{bmatrix} F_\theta(\mathbf{T}_{\text{in}}[0,:]) \\ \vdots \\ F_\theta(\mathbf{T}_{\text{in}}[N-1,:]) \end{bmatrix} \triangleleft \text{per-token nonlinearity}$$

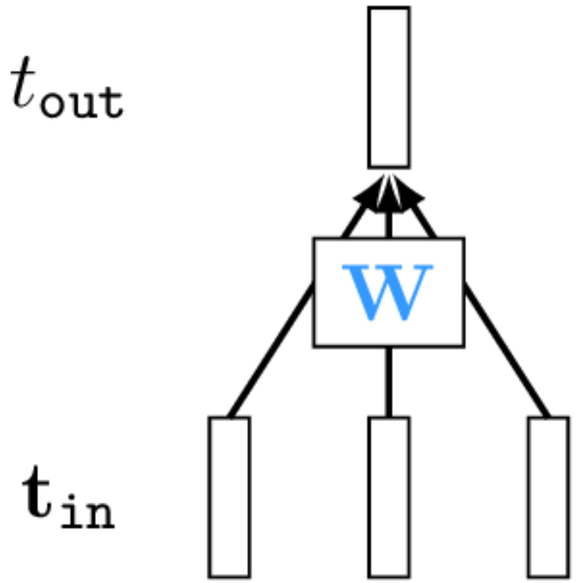
$$\mathbf{T}_{\text{out}}[i,:] = \sum_{j=1}^{N} w_{ij}\mathbf{T}_{\text{in}}[j,:] \triangleleft \text{linear combination of tokens}$$



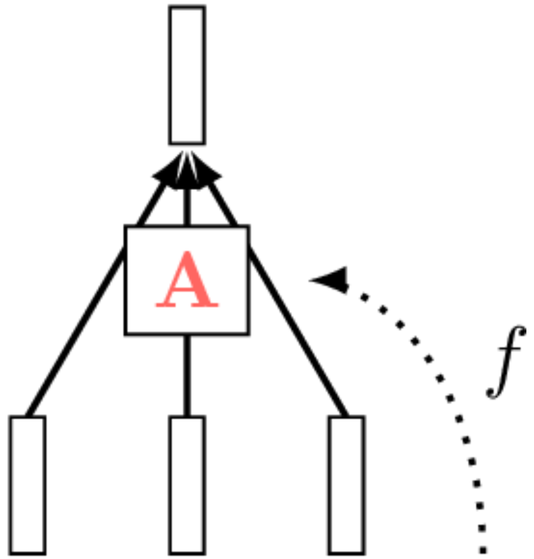
Token hálózat

# Figyelem (Attention)

## MLP vs. Attention

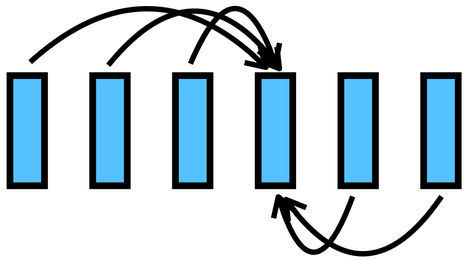



**MLP / fully-connected**
statikus súlyok



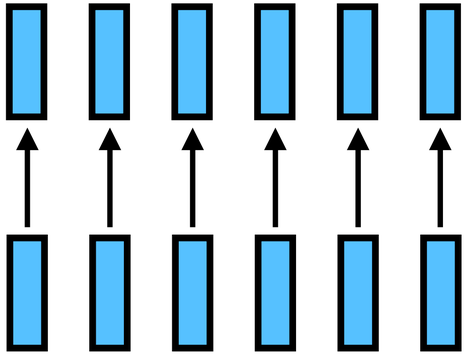

**Attention**
dinamikus (bemenet-függő) súlyok

# Tokenek



1. Tokenek közötti kommunikáció
(üzenetváltás — all-to-all / teljes gráf)

**Attention**

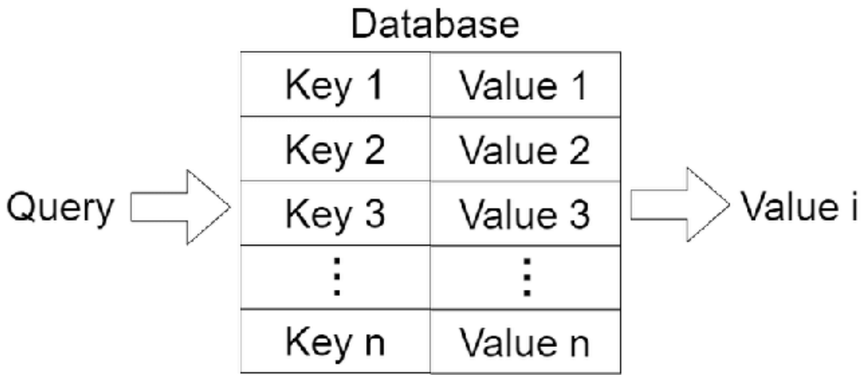


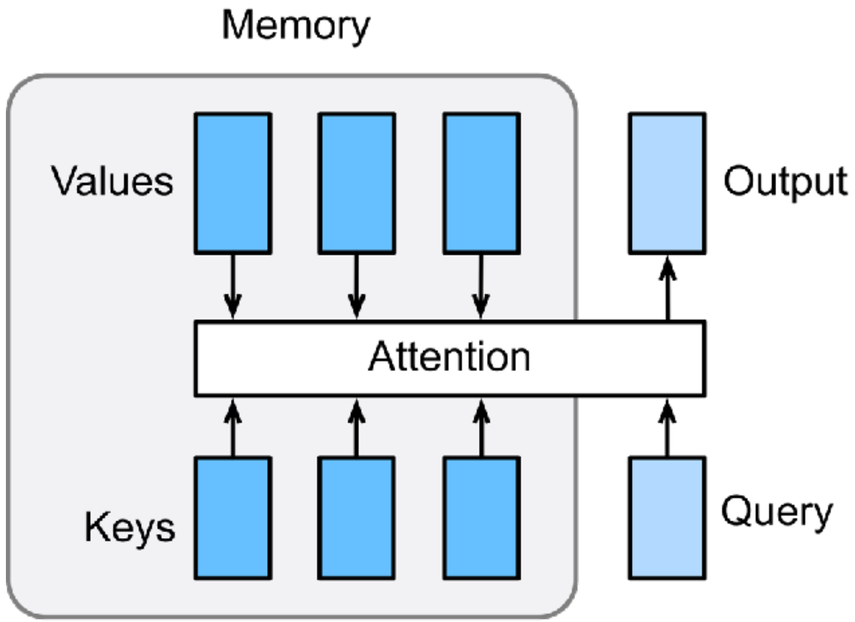
2. Tokenek feldolgozása
(per-token)

**Fully Connected / MLP**

# Figyelem (Attention)
## Query / Key / Value


- Analógia: *adatbázis / asszociatív memória*
- **Query**: mit keresünk?
- **Key**: mire reagálunk?
- **Value**: mit közlünk?



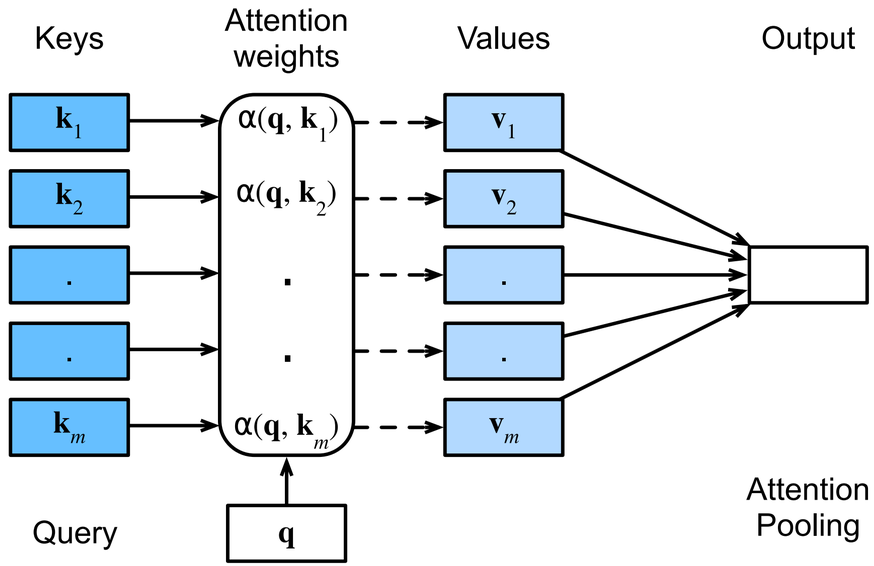



# Figyelem (Attention)
## QKV Mátrixok, Attention


- **Query**: mit keresünk? $q_i = Qx_i$
- **Key**: mire reagálunk? $k_i = Kx_i$
- **Value**: mit közlünk? $v_i = Vx_i$
- Attention / figyelem súlyok: $\alpha(q_i, k_j)$
- Value-k figyelem szerint súlyozott átlaga — *figyelem (attention) mechanizmus*:

$$x_i' = \sum_{j=1}^{n} \alpha(k_i, q_j) \cdot v_j$$

$Q, K, V$: ***lineáris*** transzformációk
**Minden tokenre azonosak!**
**Tanuljuk őket!**



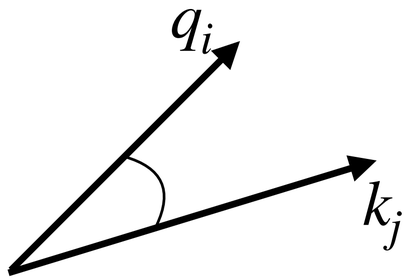
# Figyelem (Attention)

## Dot-product attention

- Skalárszorzat (**dot-product**):

$$\alpha(q_i, k_j) = q_i^T k_j$$

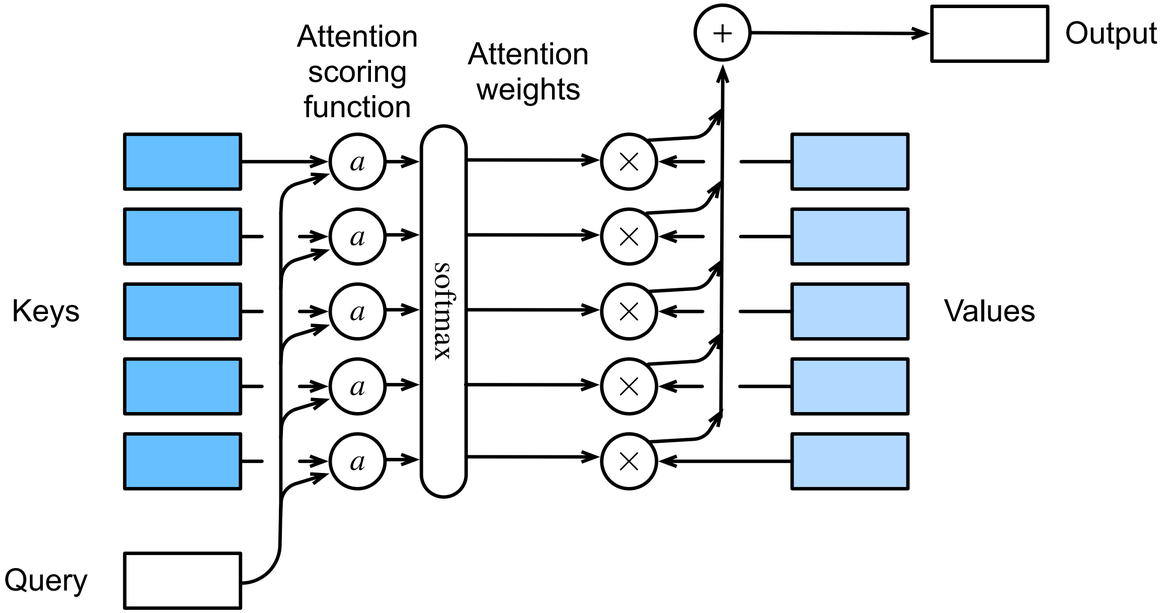
$$q_i = Qx_i$$
$$k_i = Kx_i$$
$$v_i = Vx_i$$




$$q_i^T k_j = \| q_i \| \| k_j \| \cos(\angle(q_i, k_j))$$

- **Softmax** képzés:

$$x_i' = \sum_{j=1}^{n} \mathrm{SoftMax}(\alpha(q_i, k_j)) \cdot v_j$$



# Figyelem (Attention)

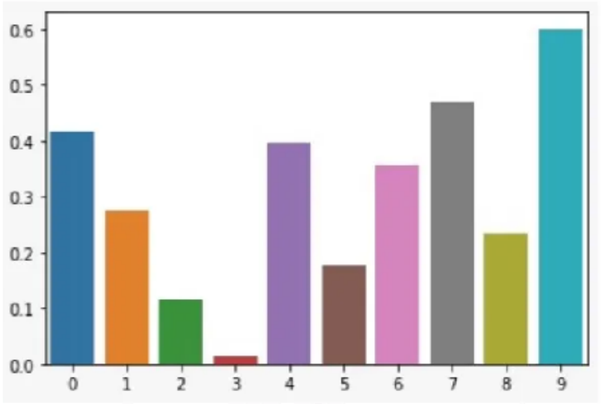
## Emlékeztető: SoftMax

- Logitok: $x_1, \ldots, x_n$
- **SoftMax**:

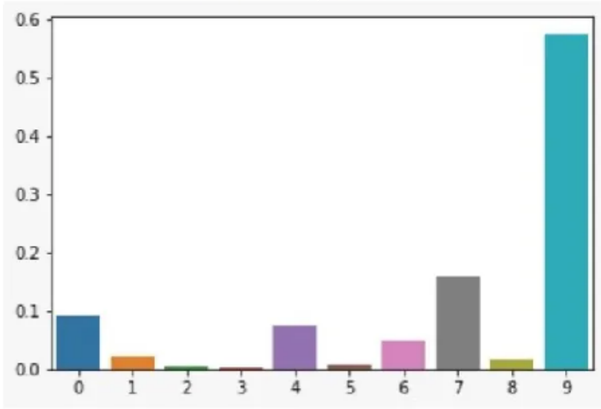
$$\mathrm{SoftMax}(x_i) = \frac{e^{x_i/T}}{\sum_{j=1}^{n} e^{x_j/T}}$$

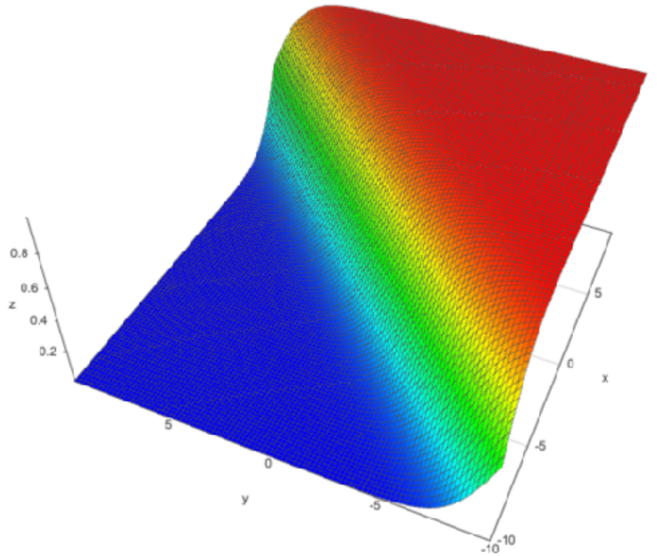
- Tetszőleges számokból valószínűségleoszlást csinál!
- Kiemeli a maximális értéket!

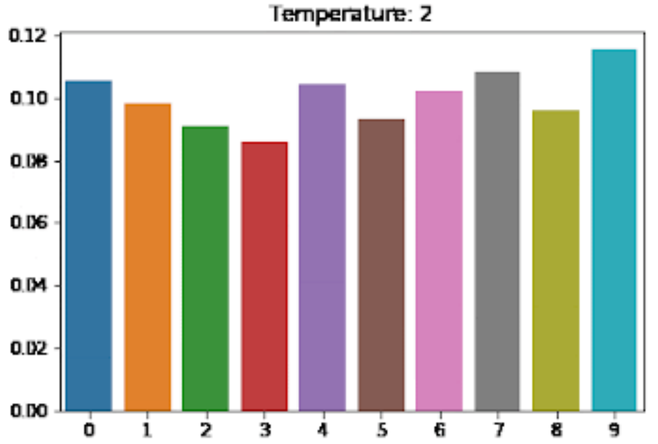
(Helyesebb volna SoftArgMax-nak nevezni…)



SoftMax








$T$: “**hőmérséklet**”

# Figyelem (Attention)

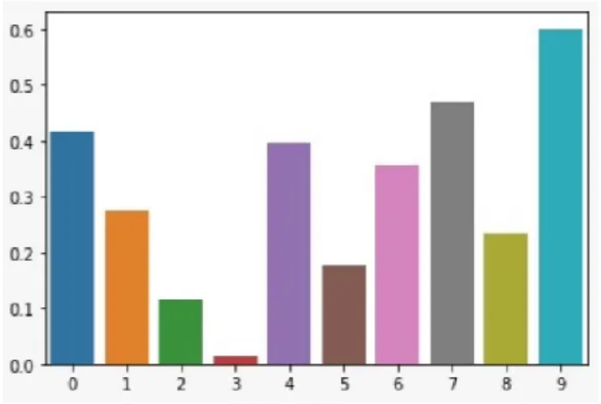
## Emlékeztető: SoftMax

- Logitok: $x_1, \ldots, x_n$
- **SoftMax**:

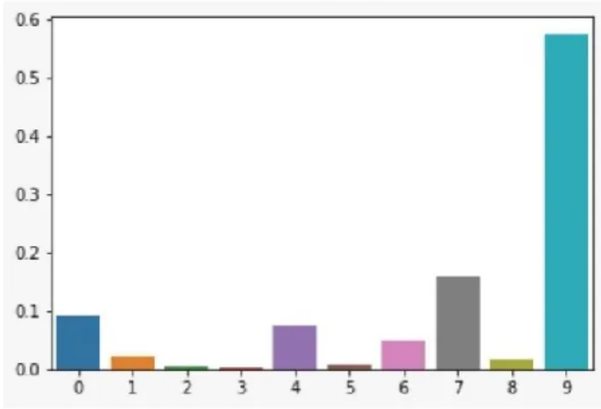
$$\mathrm{SoftMax}(x_i) = \frac{e^{x_i/T}}{\sum_{j=1}^{n} e^{x_j/T}}$$

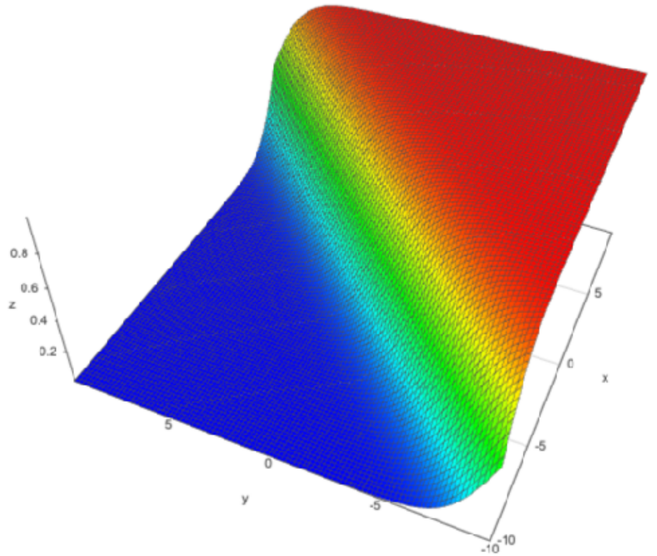
- Tetszőleges számokból valószínűségleoszlást csinál!
- Kiemeli a maximális értéket!

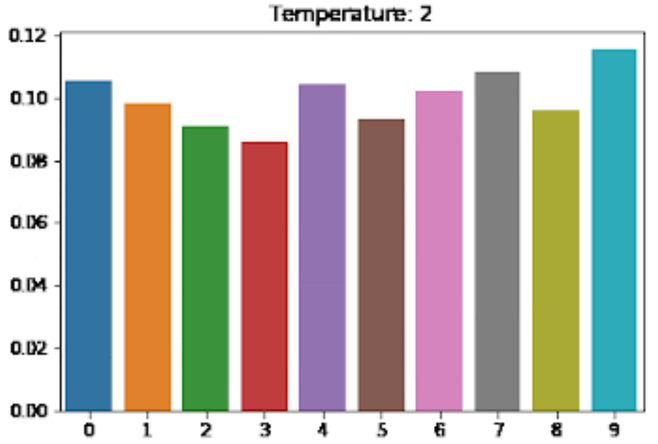
(Helyesebb volna SoftArgMax-nak nevezni…)



SoftMax







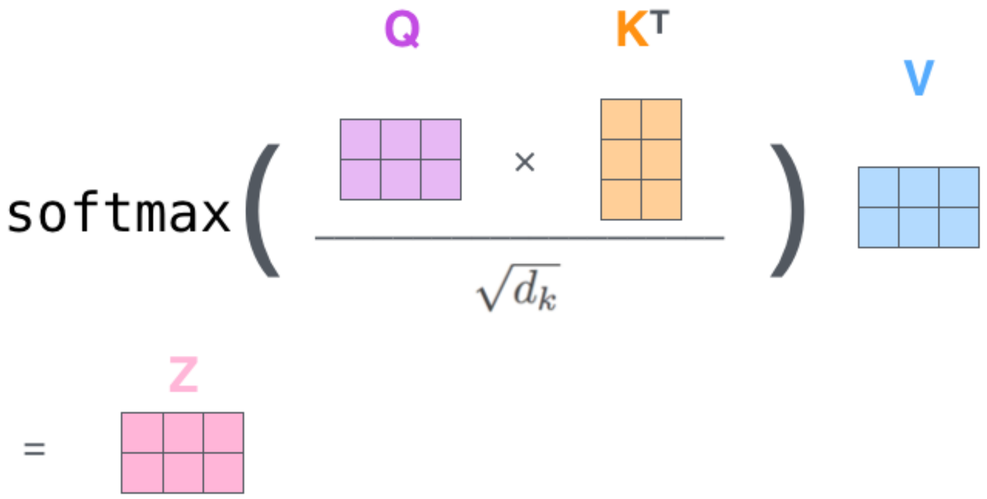

$T$: “**hőmérséklet**”

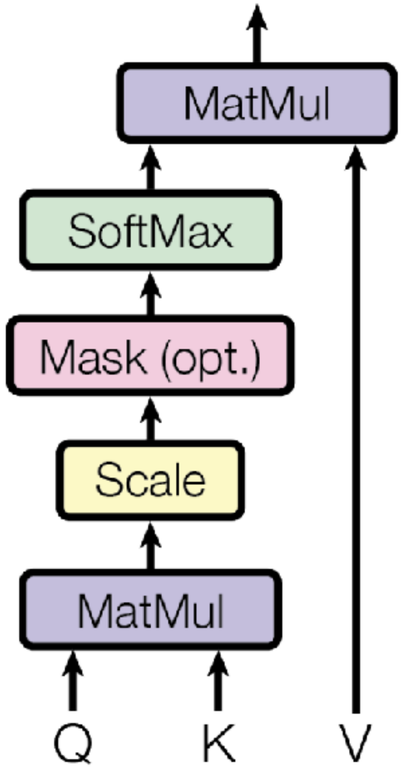
# Figyelem (Attention)

## Scaled dot-product attention

$$q_i = Qx_i$$
$$k_i = Kx_i$$
$$v_i = Vx_i$$

$q^T k \gg 1 - \mathrm{SoftMax}(q^T k) \approx \mathrm{const}$ — megáll a tanítás…
Megoldás: leskálázás — **scaled dot-product attention**



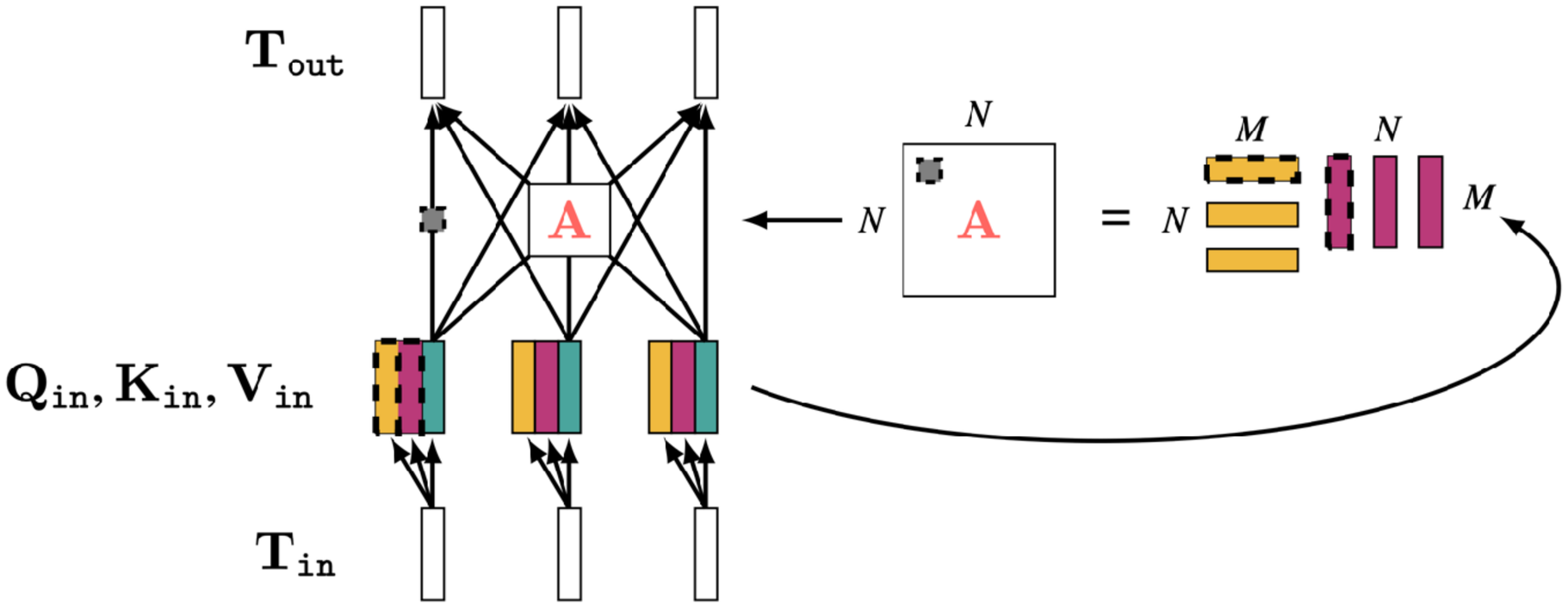




**“Attention Head”**

# Figyelem (Attention)

## Attention layer

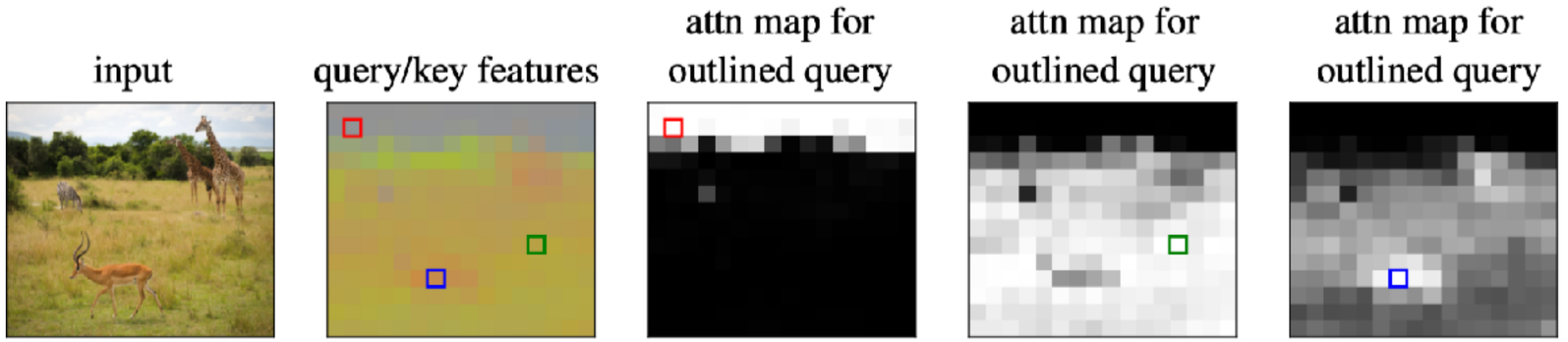
$$q_i = Qx_i$$
$$k_i = Kx_i$$
$$v_i = Vx_i$$

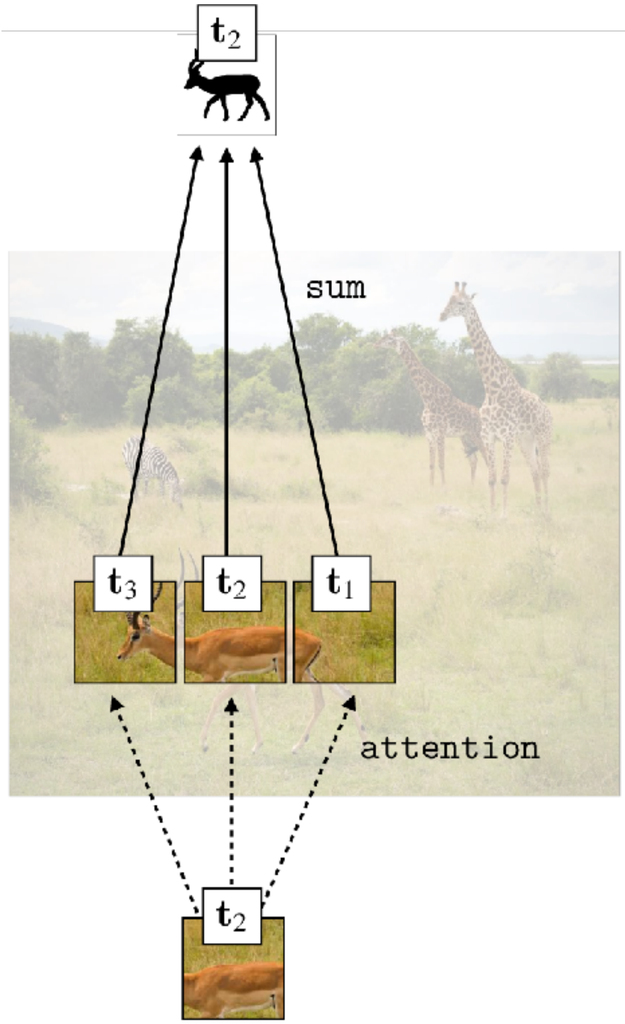



Attention layer = fully connected háló, adatfüggő súlyokkal (“hiperháló”)

# Figyelem (Attention)

## Önfigyelem (self-attention)

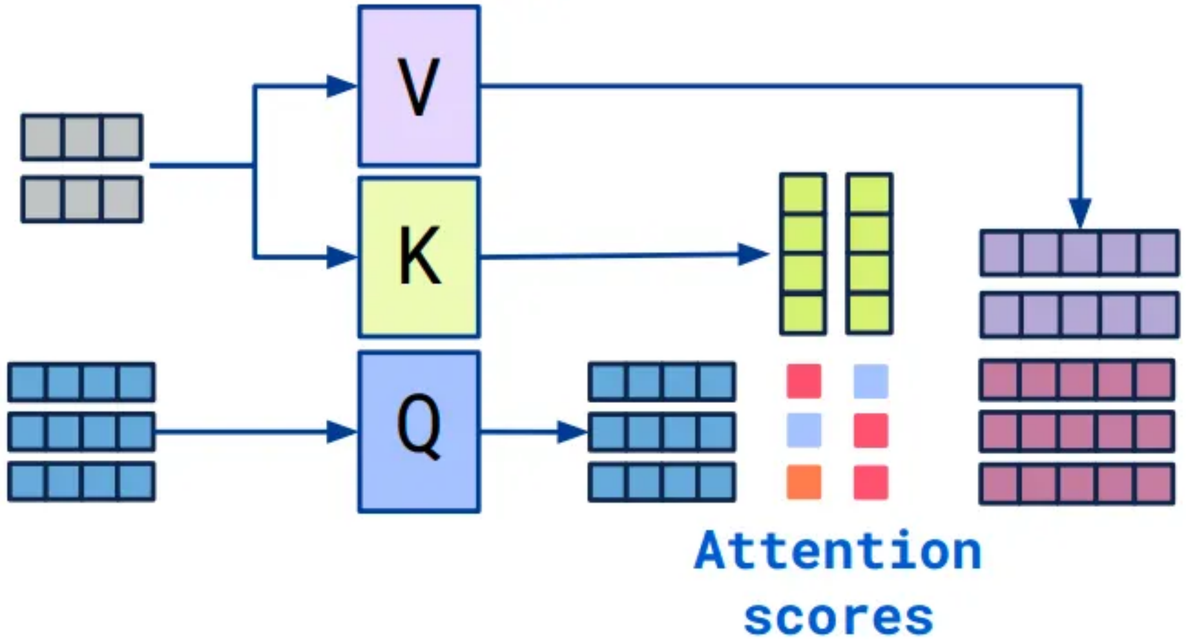


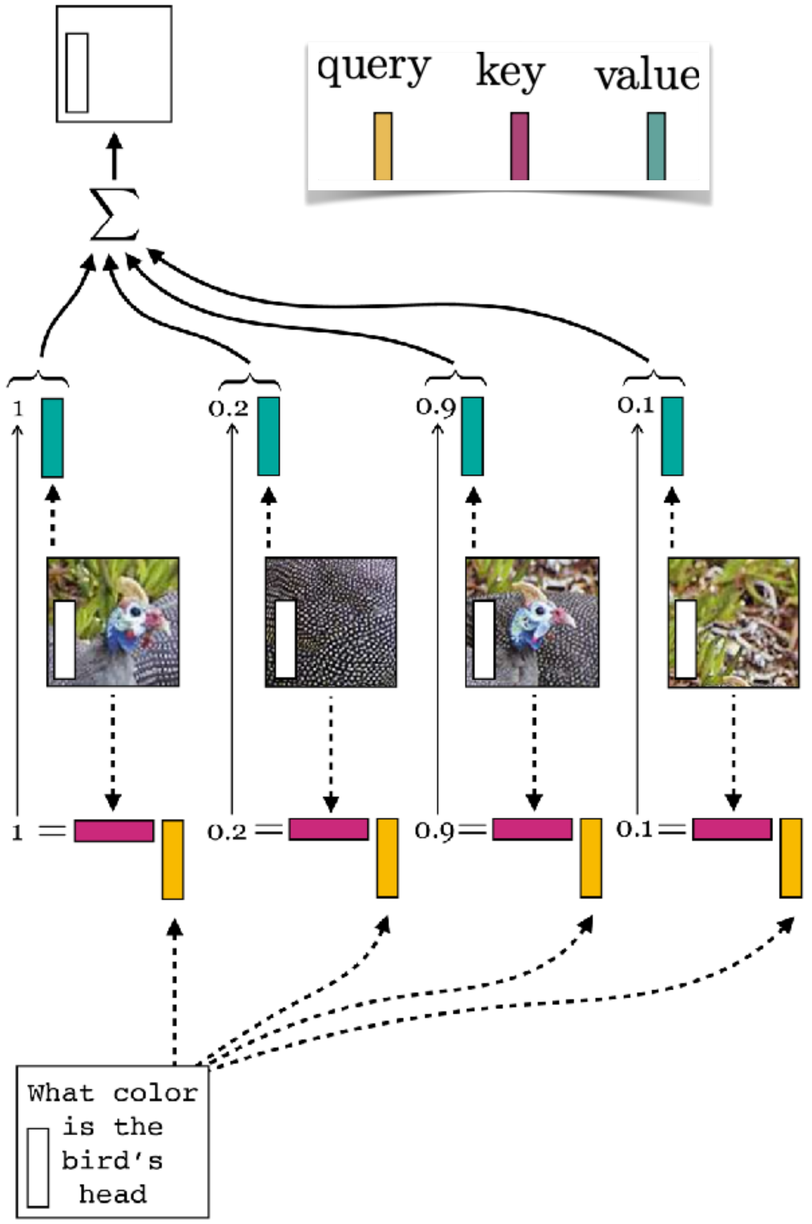




**<u>Önfigyelem</u>**: query/key/value tokenek forrása megegyezik
(Tokenek transzformációja a releváns kontextus alapján)

# Figyelem (Attention)

## Keresztfigyelem (cross-attention)




A query/key/value származhat különböző token-halmazokból is!

**Keresztfigyelem (cross-attention)**

Pl. nyelvfordítás, képleírás, szövegből kép, stb.

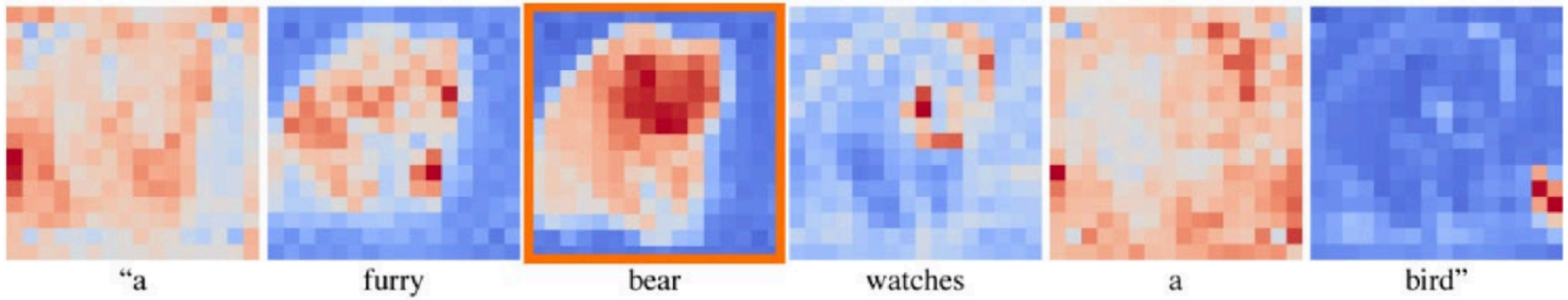
# Figyelem (Attention)

## Keresztfigyelem (cross-attention)

$P^2$

$Q$


$K$

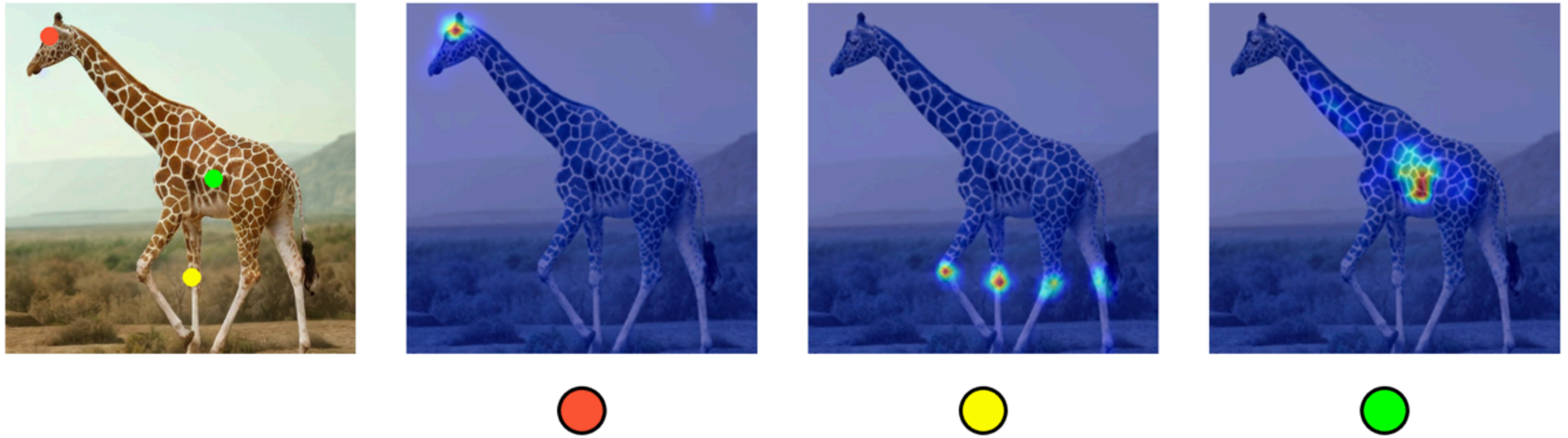



$$A_t = Q \times K$$

# Figyelem (Attention)

## Önfigyelem vs Keresztfigyelem

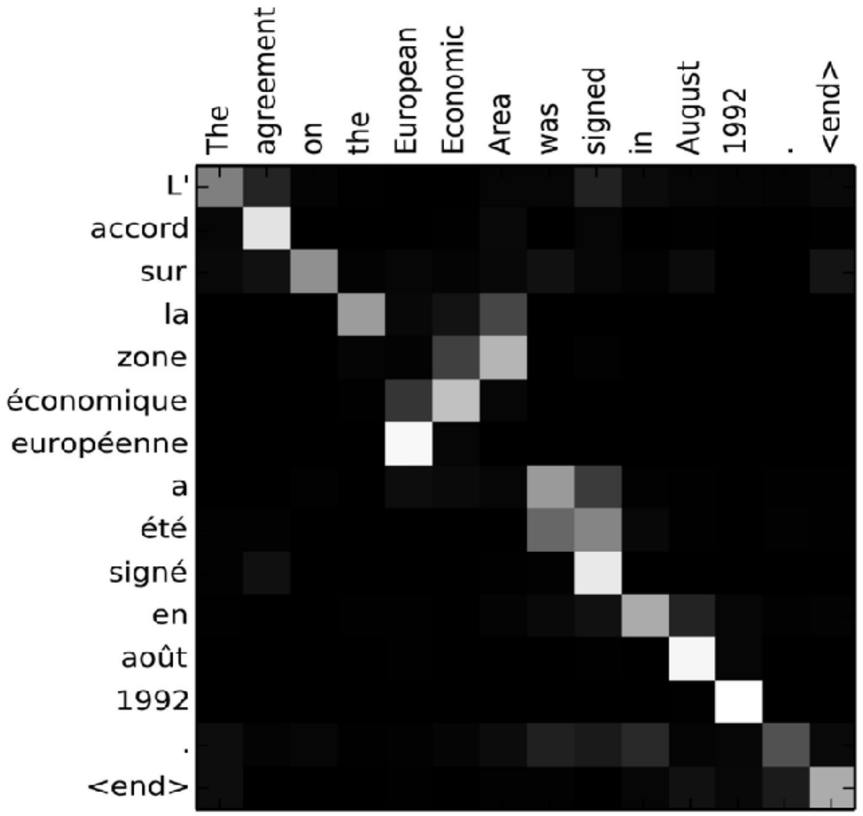
**Self-Attention**

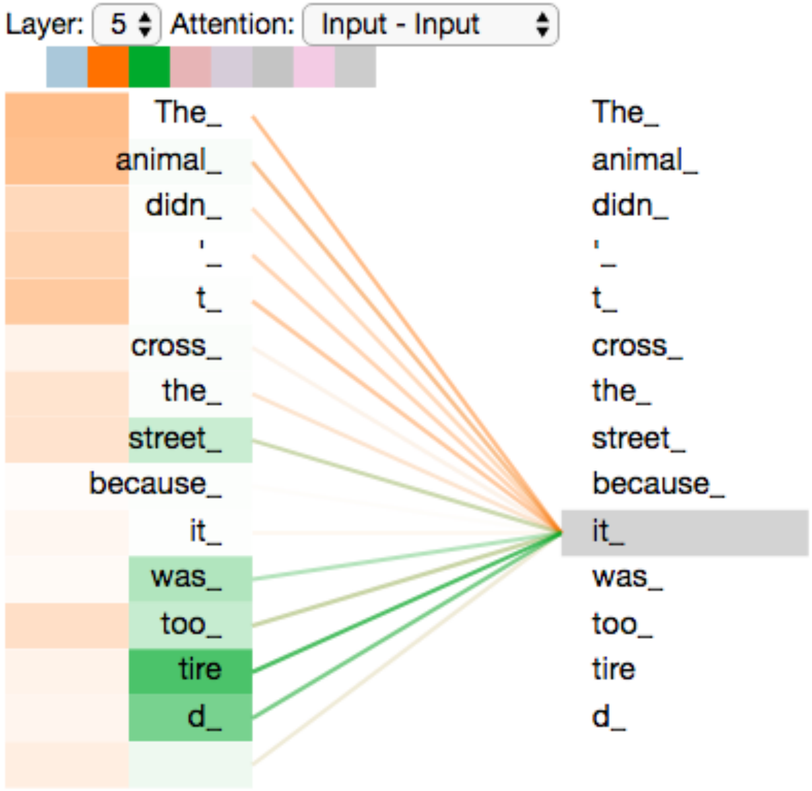


**Cross-Attention**

# Figyelem (Attention)

## Szövegről szövegre



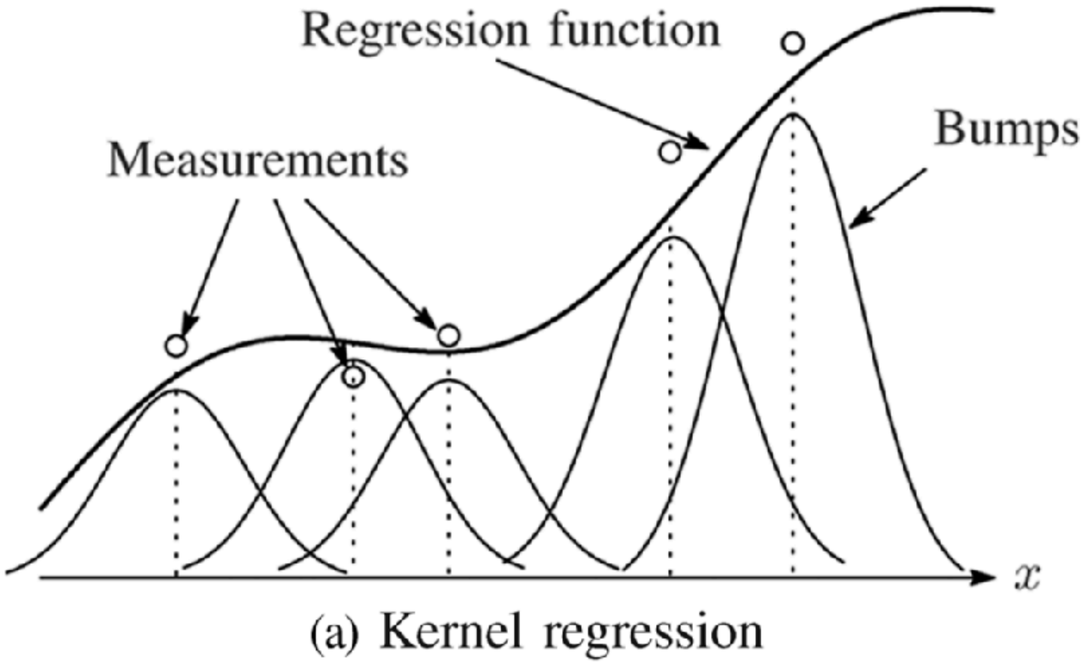



# Figyelem (Attention)
## Kernel analógia

- Adatpontok/mérések: $(k_i, v_i)$
- **Kernel szűrés** — adatpontok súlyozott átlagolás:

$$v(q) = \sum_i \mathcal{K}(k_i, q) \cdot v_i$$

- $\mathcal{K}(k_i, \cdot)$: **kernel függvény**, egy adatponthoz mért “hasonlóságot”, “távolságot” mér
  - Pl. Gauss függvény: $e^{-\| \cdot - k_i \|^2}$

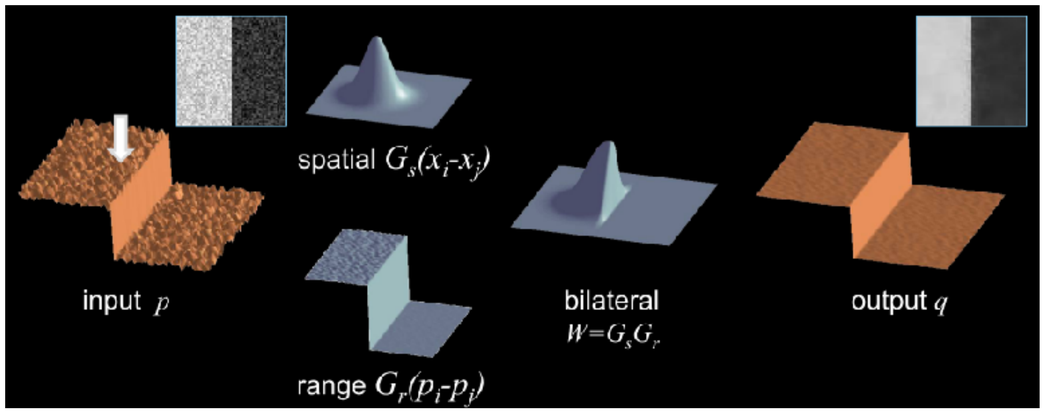


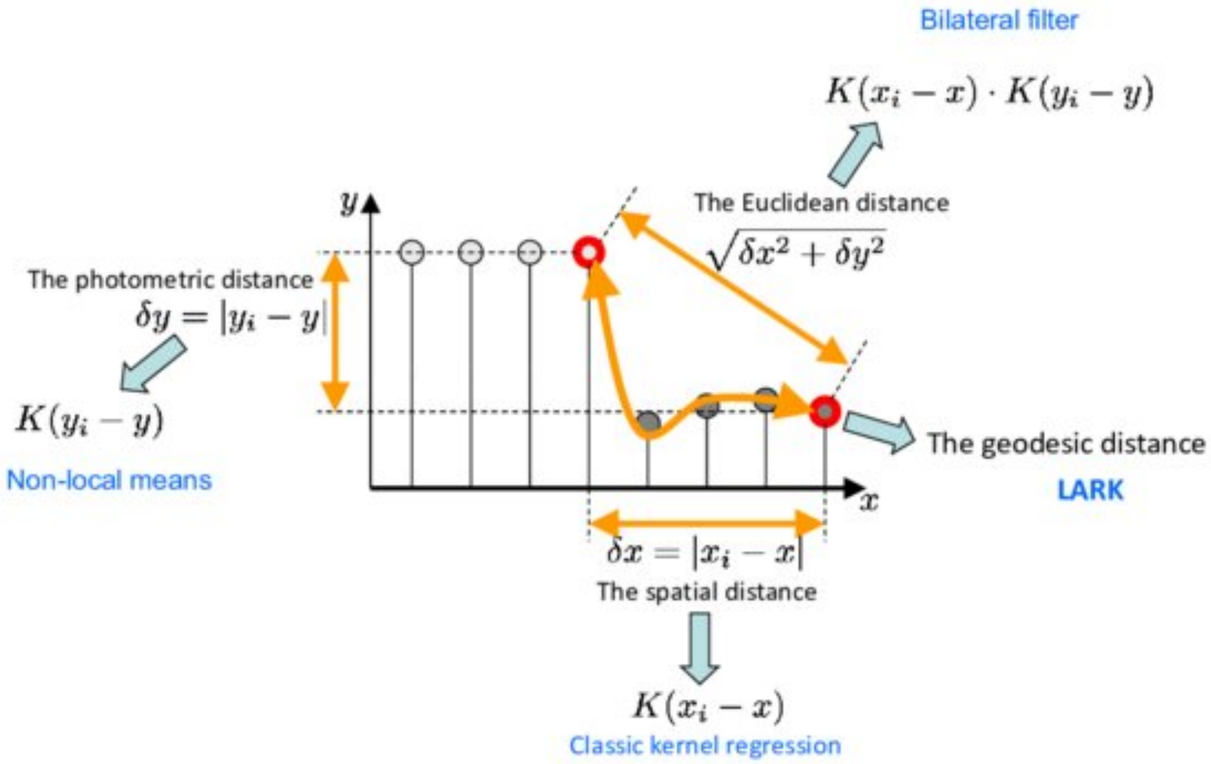
(a) Kernel regression

# Figyelem (Attention)

## Kernel analógia

- Hasonlóságot nem csak “térben” lehet mérni — “adatfüggő” kernel!
- Pl. **bilaterális szűrés**: összehasonlítás értékek szerint is (grafikon távolság)



# Figyelem (Attention)

## Kernel analógia

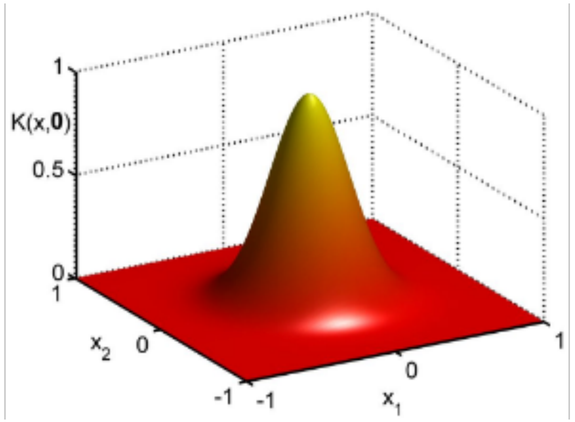
$$y(x) \propto \sum_i e^{-\| q - k_i \|^2} \cdot v_i$$

- Gauss-kernel átalakítása:

$$e^{-\| q - k_i \|^2} = e^{-(q-k_i)^T(q_i-k_j)} = e^{-(q^Tq-2q^Tk_i+k_i^Tk_i)} = e^{-\| q \|^2} \cdot e^{2q^Tk_i} \cdot e^{-\| k_i \|^2}$$

(Normalizált vektorok)

$$\propto e^{q^Tk_i}$$

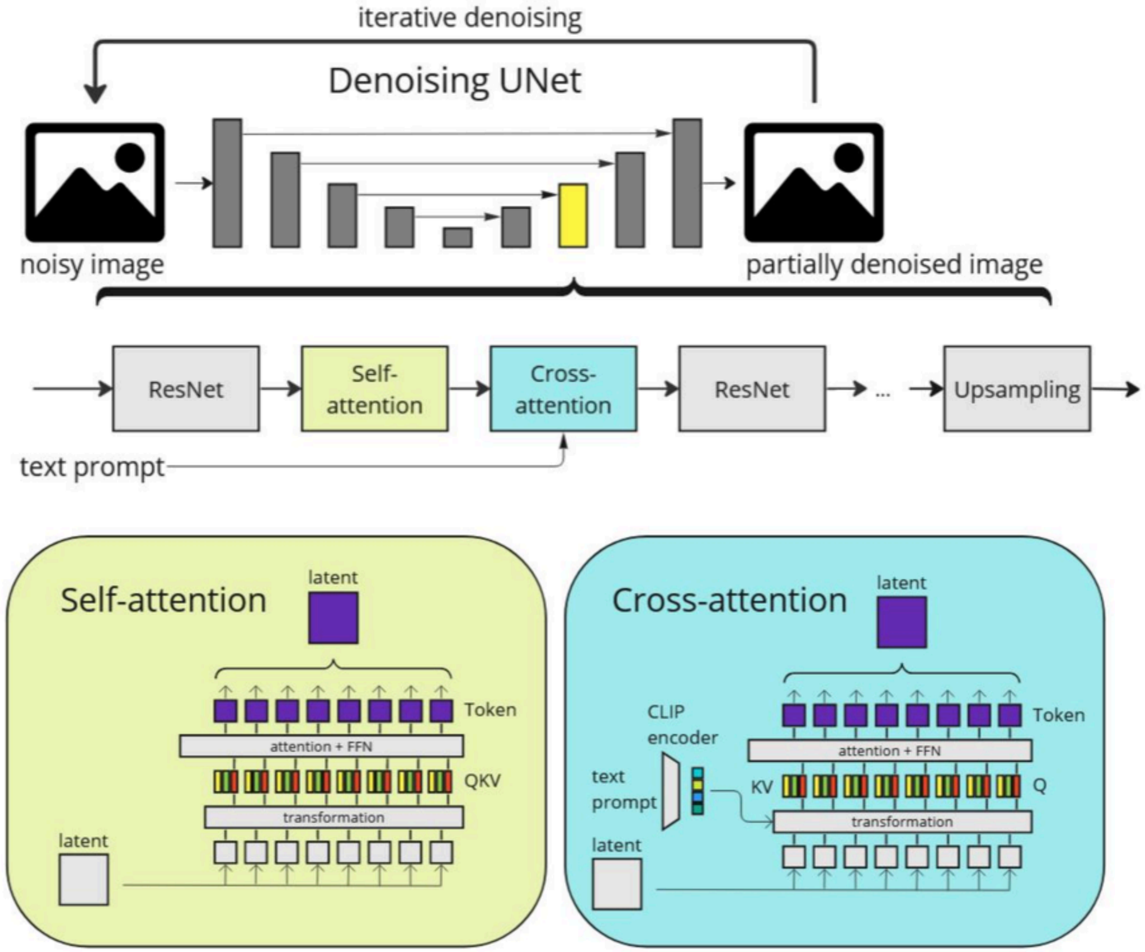



SoftMax dot-product attention
=
adatok szűrése (fix) Gauss kernellel
(tanult) lineáris transzformáció után!

$$q = Qx$$
$$k_i = Kx_i$$
$$v_i = Vx_i$$

# Figyelem (Attention)

## Alkalmazás — Diffúziós generálás kondicionálása



# Figyelem (Attention)

## Alkalmazás — Diffúziós generálás kondicionálása

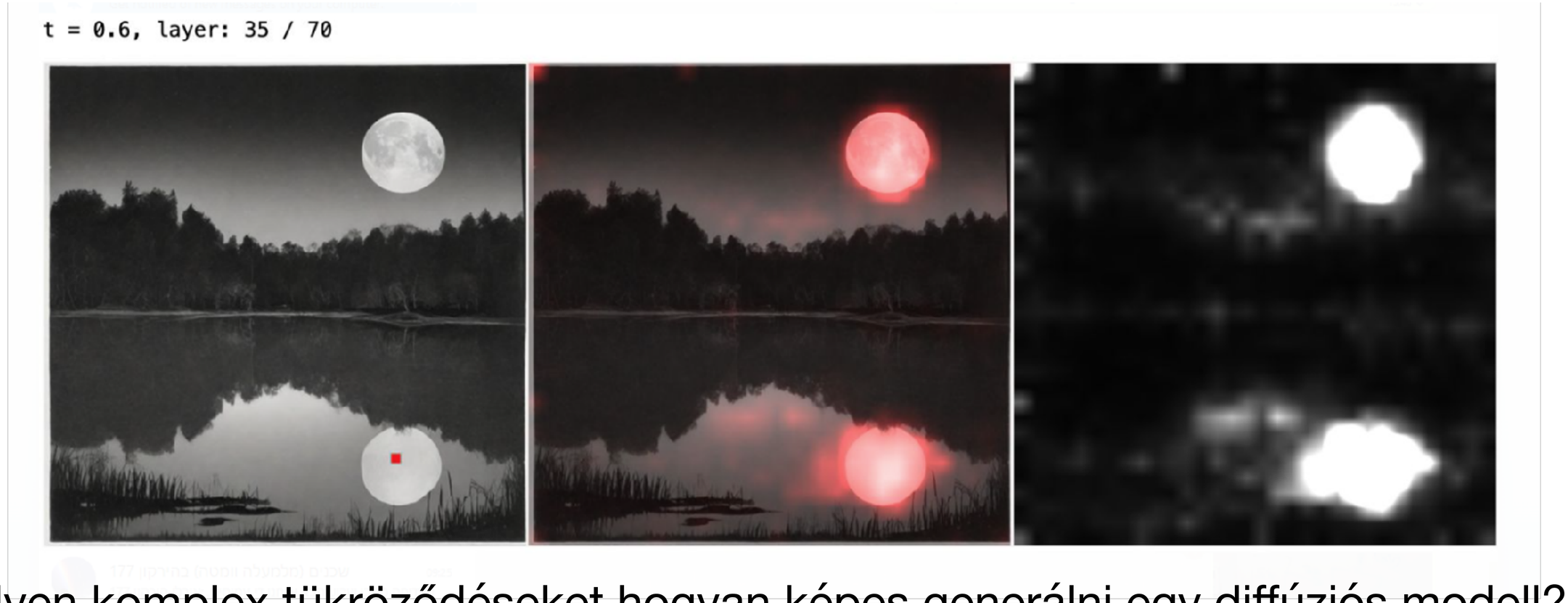
Forrás: Midjourney

Forrás: Google Gemini (“Nano Banana”)

Ilyen komplex tükröződéseket hogyan képes generálni egy diffúziós modell?

# Figyelem (Attention)

## Alkalmazás — Diffúziós generálás kondicionálása

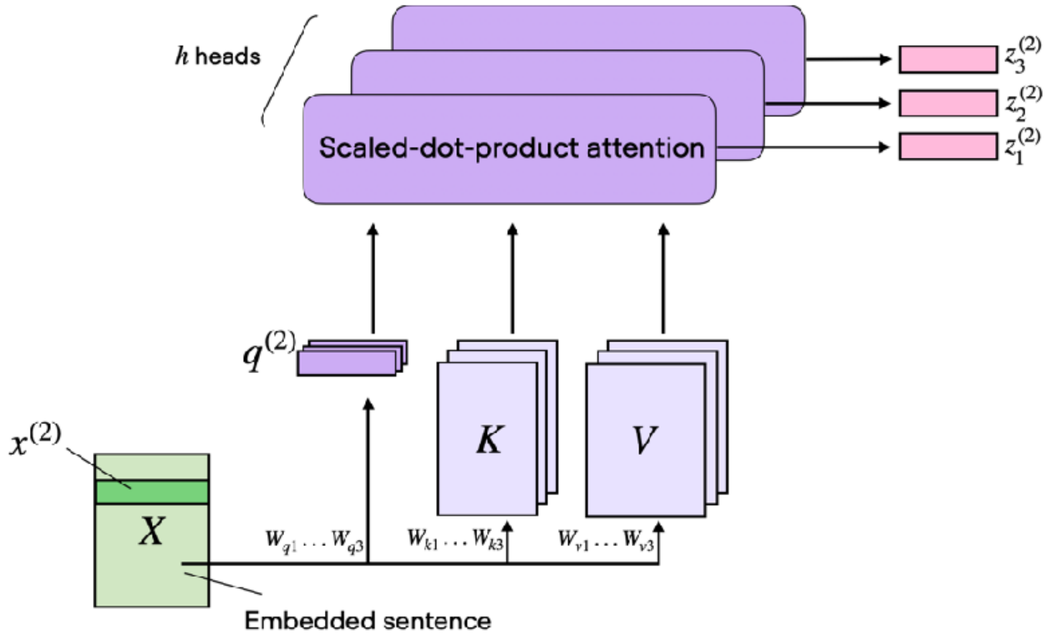


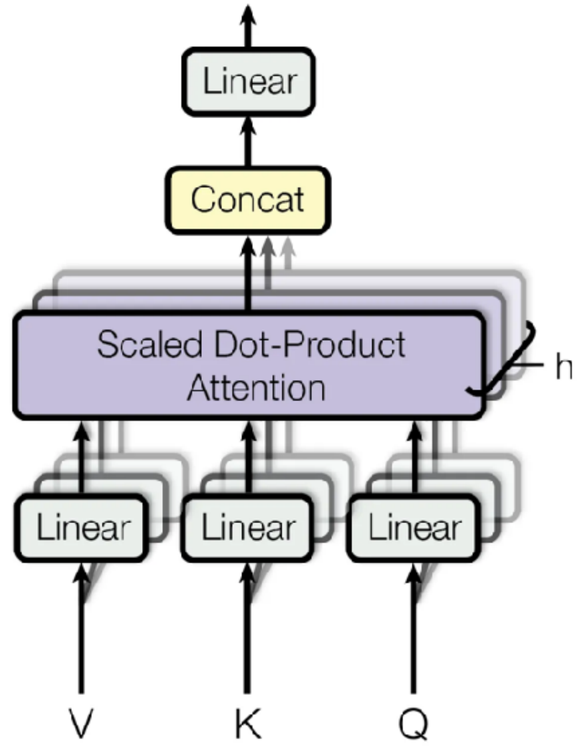

Ilyen komplex tükröződéseket hogyan képes generálni egy diffúziós modell?

**Self-attention rétegekkel!**

# Figyelem (Attention)

## Multi-head Attention




Alkalmazzunk több attention head-et párhuzamosan — **multi-head attention**

# Transformer

**Attention Is All You Need**


**Ashish Vaswani*** Google Brain avaswani@google.com

**Noam Shazeer*** Google Brain noam@google.com

**Niki Parmar*** Google Research nikip@google.com

**Jakob Uszkoreit*** Google Research usz@google.com

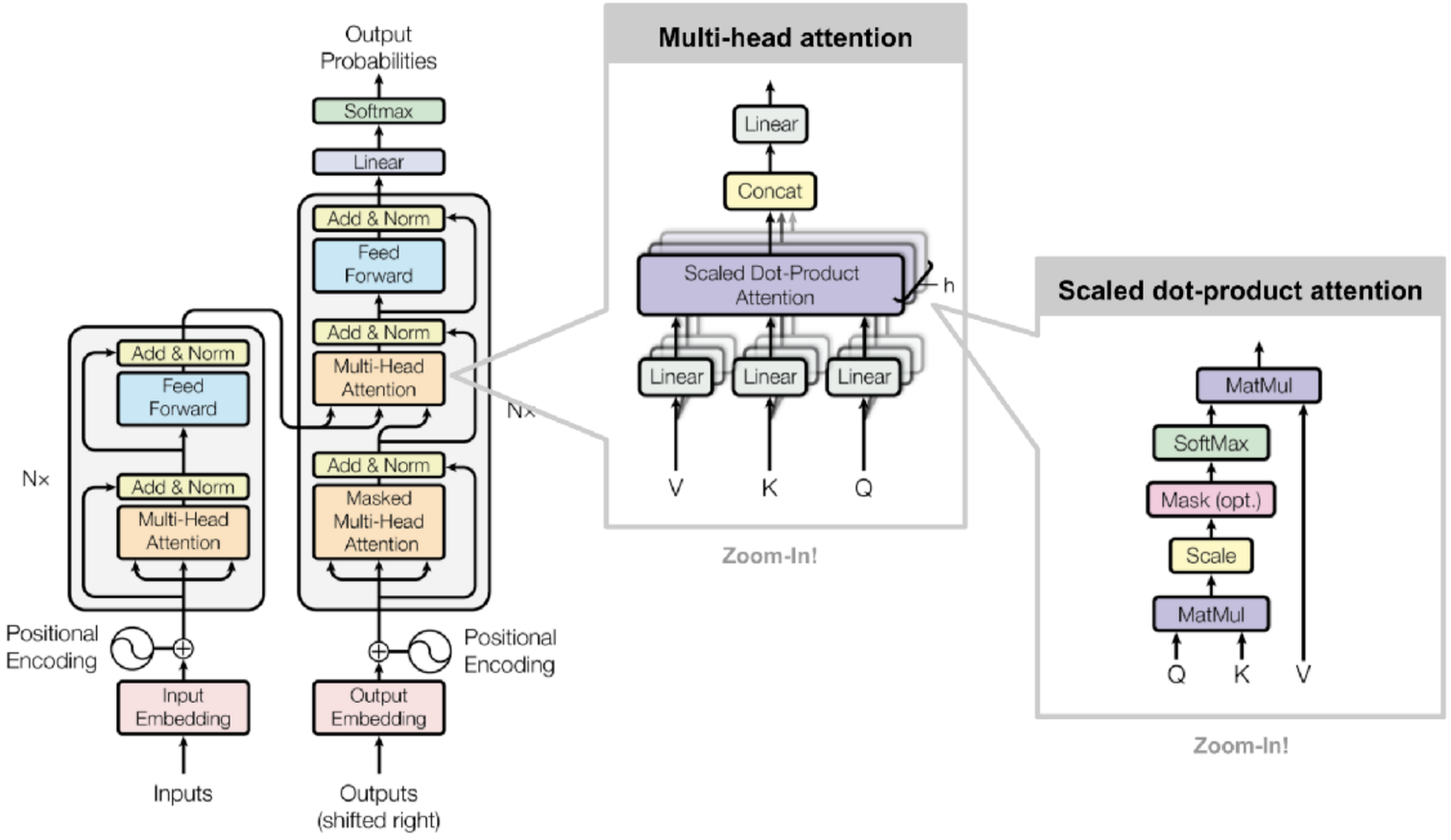
**Llion Jones*** Google Research llion@google.com

**Aidan N. Gomez*** † University of Toronto aidan@cs.toronto.edu

**Łukasz Kaiser*** Google Brain lukaszkaiser@google.com

**Illia Polosukhin*** ‡ illia.polosukhin@gmail.com

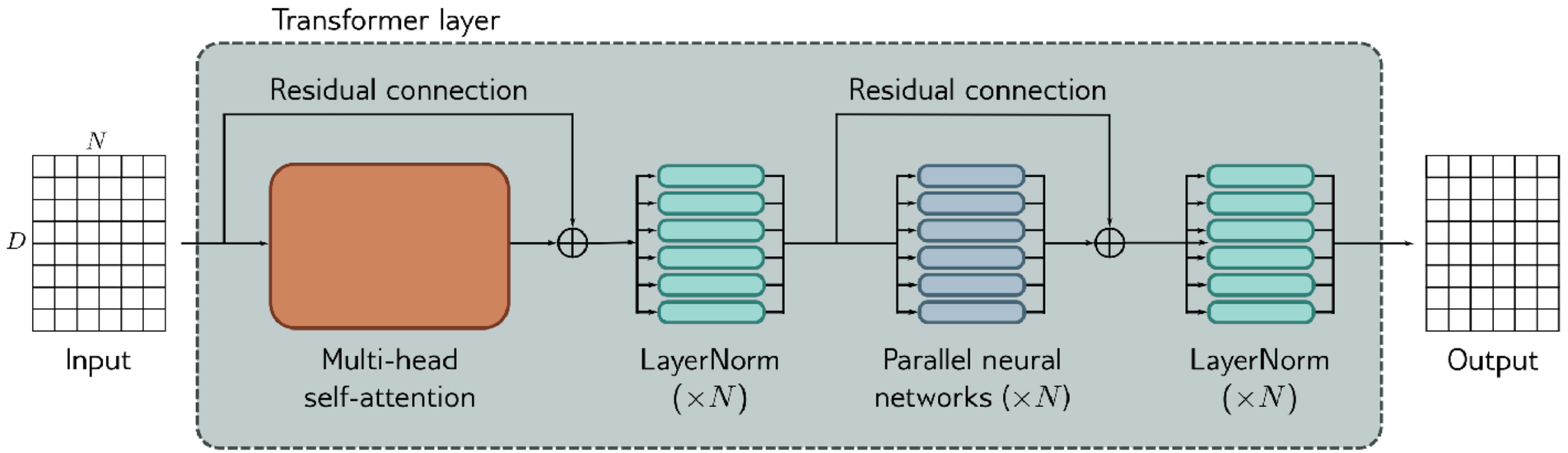

https://arxiv.org/abs/1706.03762




**Transformer architektúra**

# Transformer

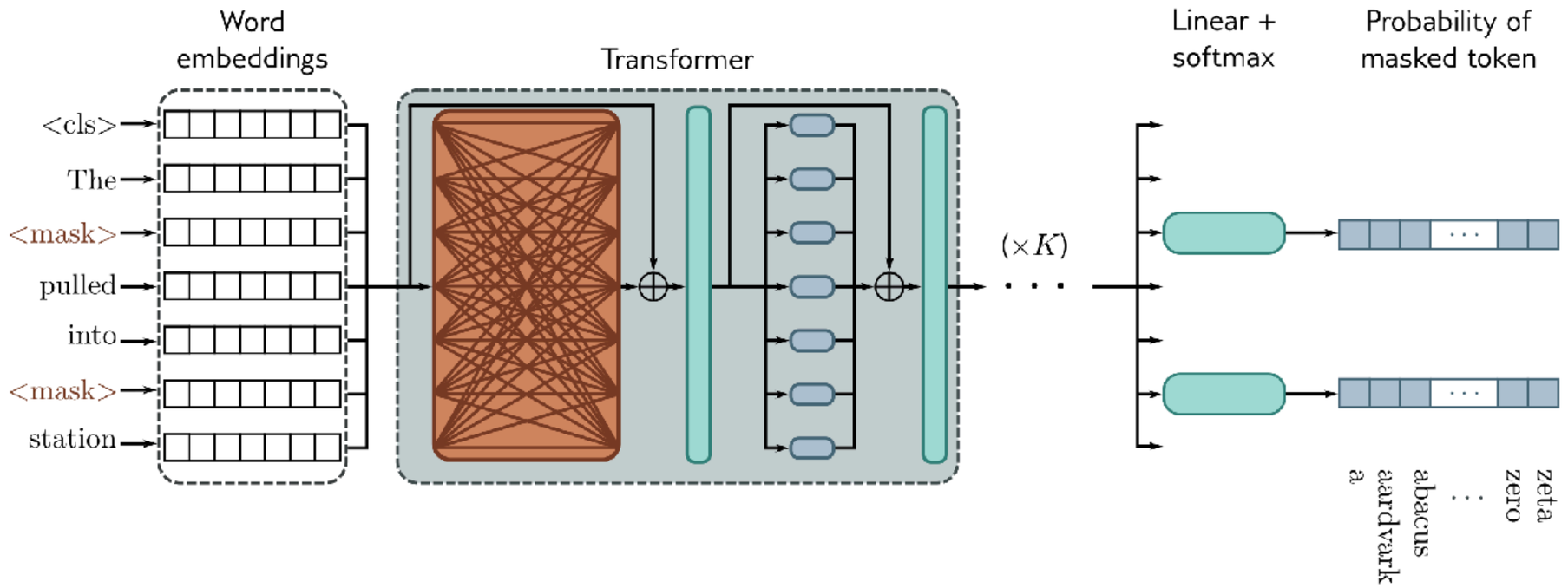
## Transformer réteg




**Transformer réteg**: attention + per-token MLP (+ Layer Normalizáció)

# Transformer

## Enkóder



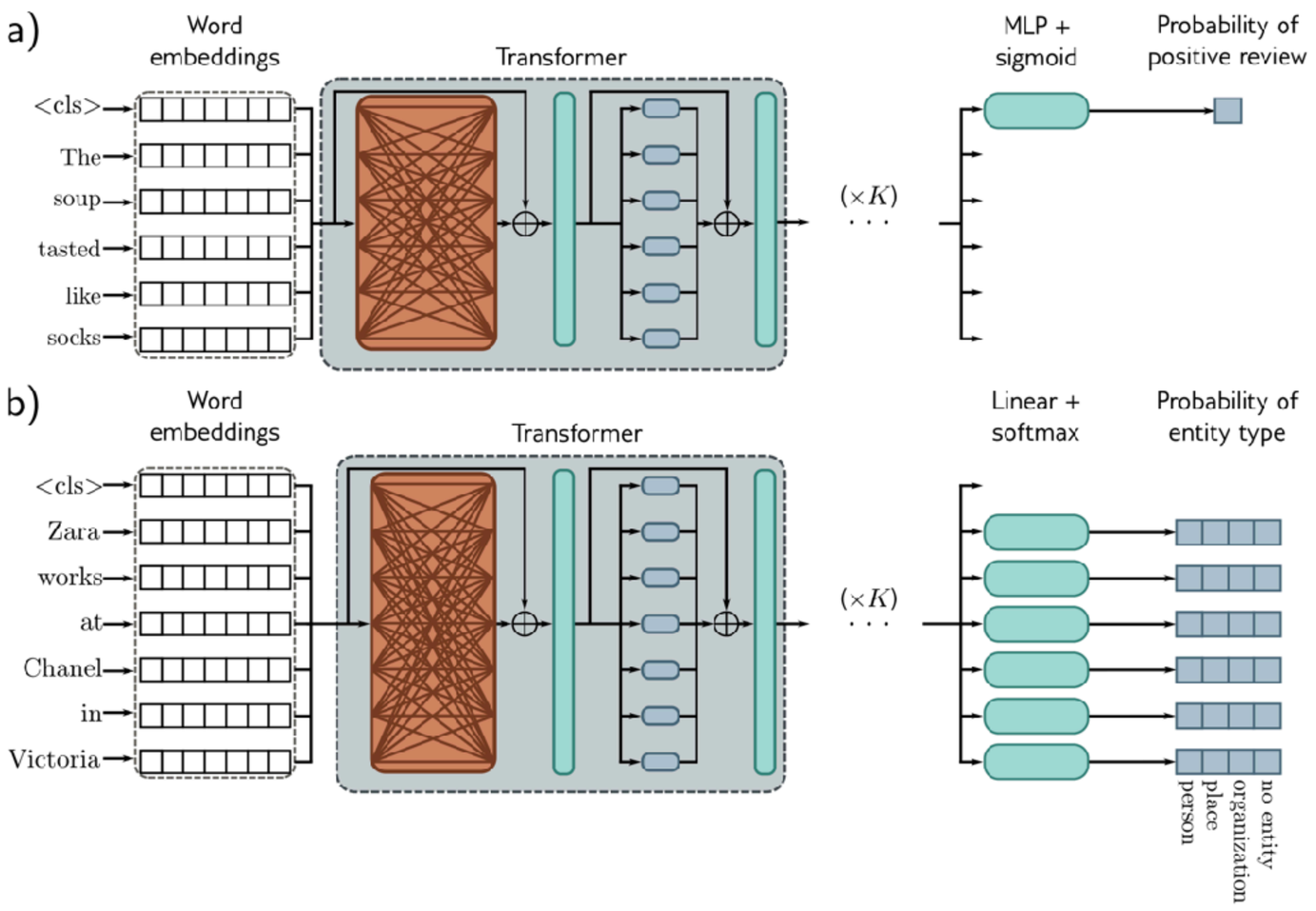


**<u>Transformer Enkóder</u>**:
önfelügyelt előtanítás (self-supervised pre-training)
önfigyelem

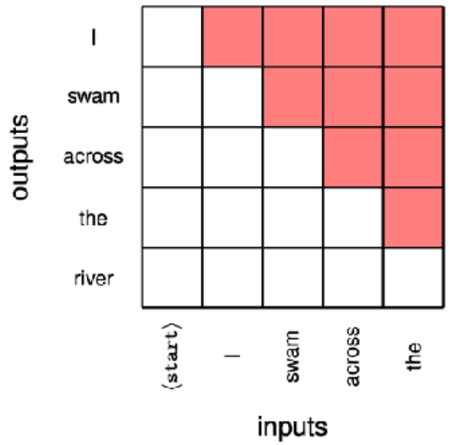
# Transformer

## Enkóder — Finomhangolás

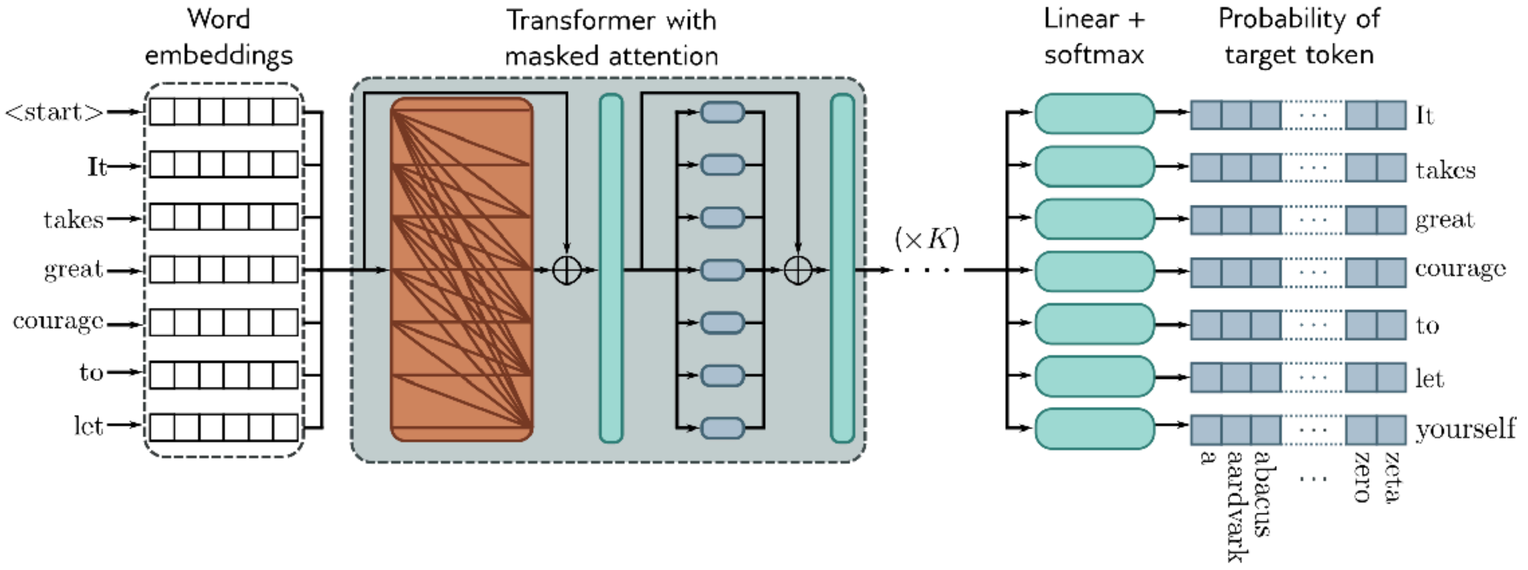
- Egy előtanított transformer enkódert később **finomhangolni** lehet!



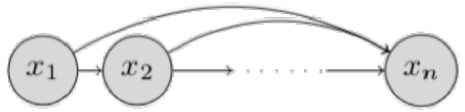
# Transformer
## Dekóder




Maszkolt / “kauzális” figyelem:
csak a korábbi tokenekre figyelünk




$$p(x_1, x_2, \ldots x_n) = p(x_1) \cdot p(x_2 \mid x_1) \cdot \cdots \cdot p(x_n \mid x_1, x_2, \ldots x_{n-1})$$




Autoregresszió

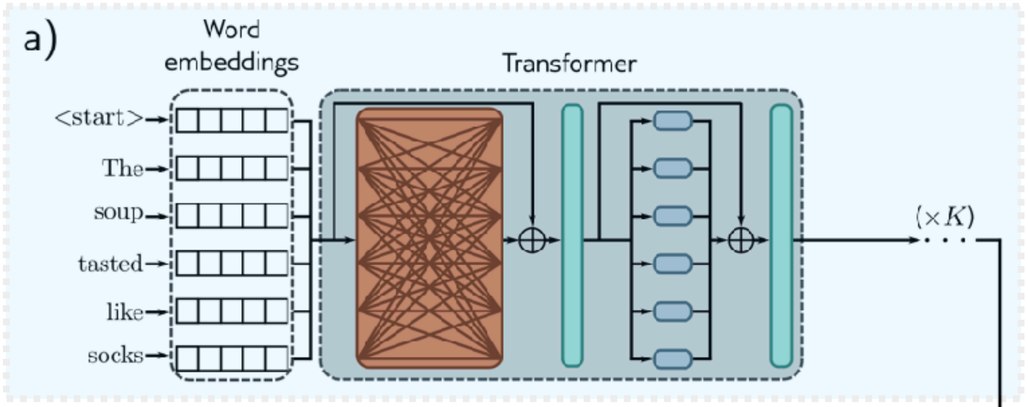
**Transformer Dekóder**:
tokenek generálására tanítjuk
(autoregresszív módon, shiftelve a bemenetet)

# Transformer

## Enkóder—Dekóder

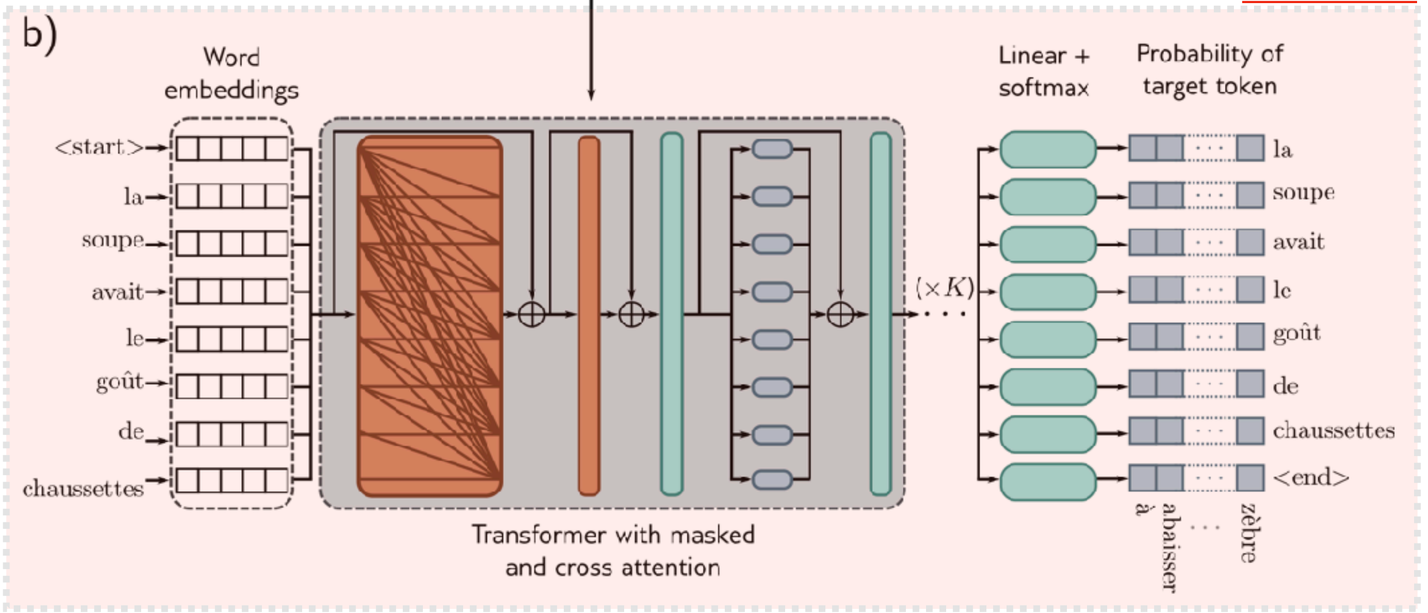
- **Enkóder-dekóder:** az “OG” Transformer architektúra
- Eredeti alkalmazás: nyelvfordítás
  - Enkóder feldolgozza a forrás szöveget (self-attention)
  - Dekóder auto-regresszíven generálja a fordítást az enkódolt forrás-tokenek alapján (cross-attention)

Enkóder




T5

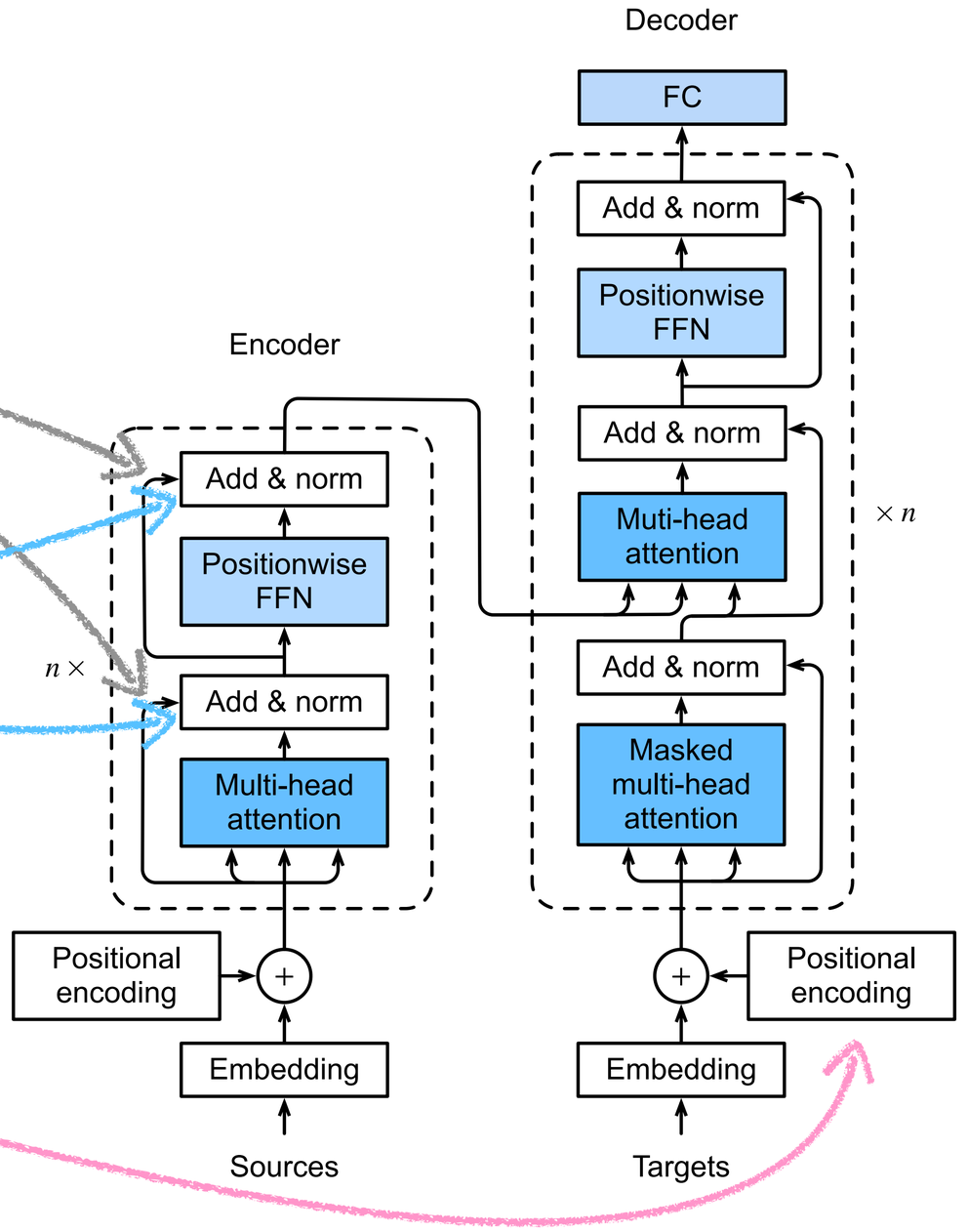
Dekóder



# Transformer

## Architektúrális részletek

- **Reziduális kapcsolatok**
  - Enélkül reménytelen tanítani…
- **Layer Normalizáció**
  - Manapság inkább az attention / MLP blokkok *előtt* csináljuk (pre-norm)!
- **Pozícionális kódolás**
  - Az micsoda?

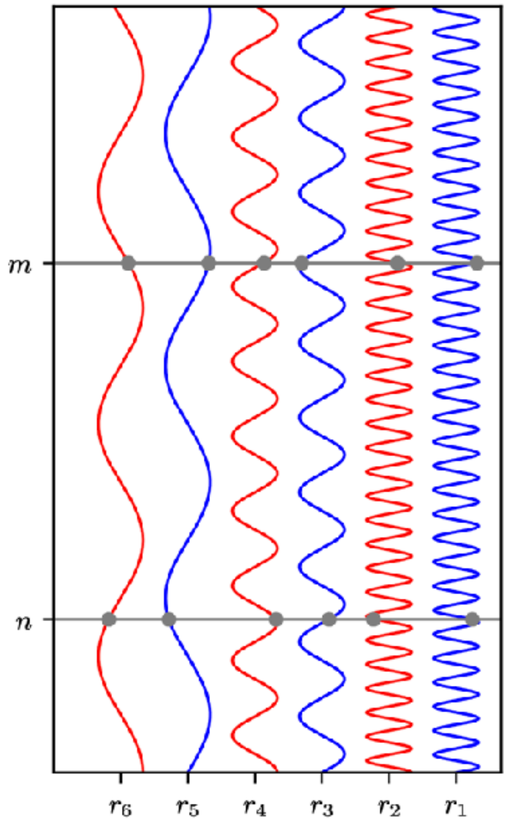


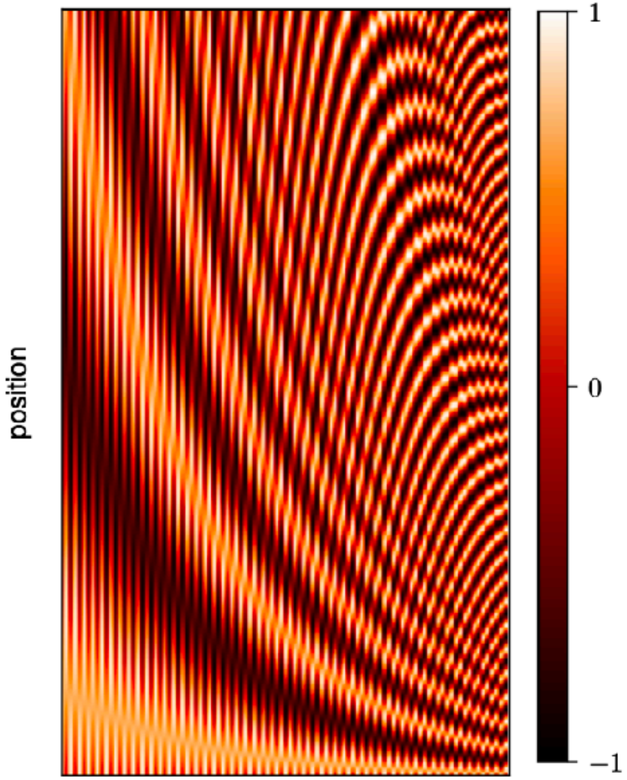
# Transformer

## Pozícionális kódolás

- Az attention blokk eredménye csak a tokenek értékétől függ, a pozíciójuktól nem!
- Egészítsük ki a tokeneket valamilyen pozíciót kódoló információval — **positional encoding**
  - Figyelem: főleg a *relatív* pozíciókra szeretnénk figyelni!
- Klasszikus megoldás: használjuk a token pozíció különböző frekvenciájú sin/cos függvényeit — a relatív pozíció (fázistolás) kinyerhető:

$$\begin{bmatrix}\cos(\omega(n-m))\\ \sin(\omega(n-m))\end{bmatrix} = \begin{bmatrix}\cos(\omega n) & -\sin(\omega n)\\ \sin(\omega n) & \cos(\omega n)\end{bmatrix}\begin{bmatrix}\cos(\omega m)\\ \sin(\omega m)\end{bmatrix}$$

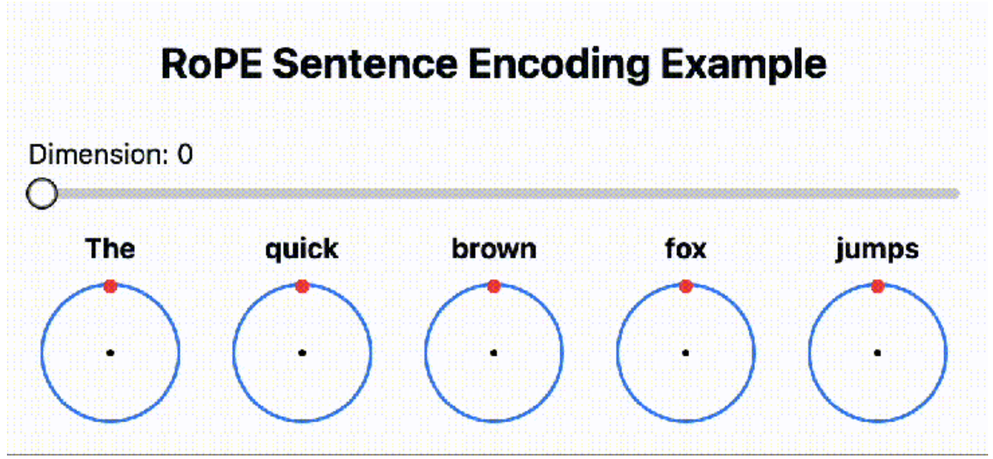


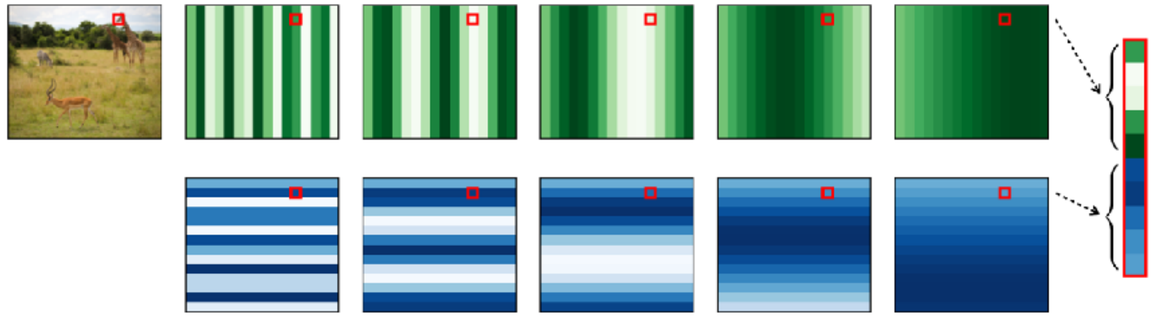
# Transformer

## Pozícionális kódolás

(“koordináta” MLP-kben is használatos — pl. Neural Radiance / Distance Field)




**Rotary Positional Encoding (RoPE):**
QK vektorok (!!) forgatása 2D síkokban



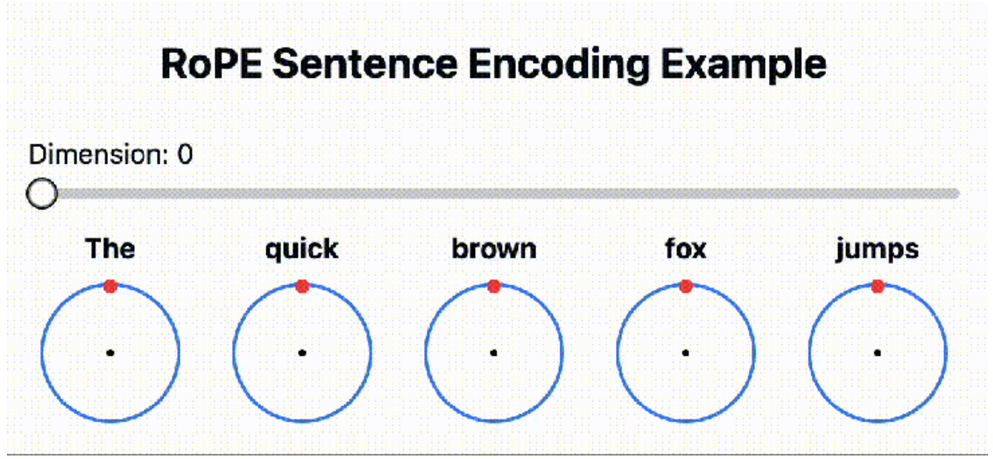
**2D pozícionális kódolás**

Előfordul tanult pozícionális kódolás is!

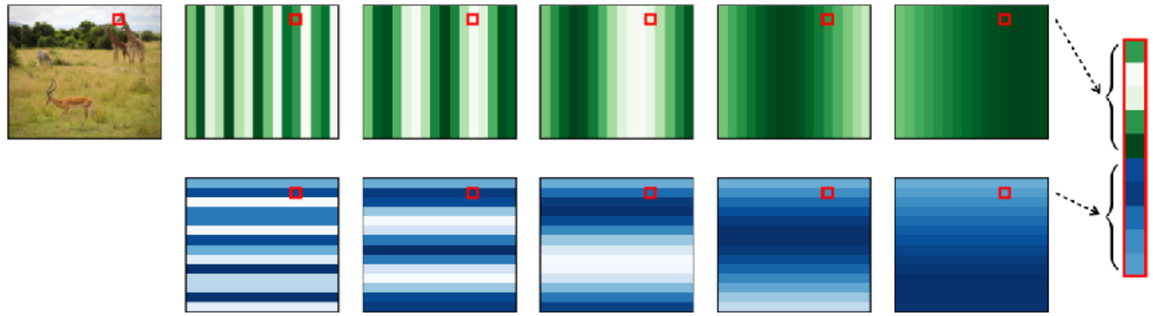
# Transformer

## Pozícionális kódolás

(“koordináta” MLP-kben is használatos — pl. Neural Radiance / Distance Field)




**Rotary Positional Encoding (RoPE):**
QK vektorok (!!) forgatása 2D síkokban



**2D pozícionális kódolás**

Előfordul tanult pozícionális kódolás is!

# Transformer

## Nagy nyelvi modellek (LLM)

- **Large Language Model (LLM)**: nagy (akár 100+ GB méretű) szöveggenerátor modell óriási (“internet-méretű”) szöveges corpuson (elő)tanítva
- Dekóder architektúrájú Transformer
  - Kisebb hatékonyság-növelő módosításokkal még ma is az eredeti transformer architektúra…
- Utólagos *finomhangolás* konkrét feladatokra (pl. beszélgető chatbot, kódgenerálás, stb.) — Reinforcement Learning from Human Feedback (RLHF)

GPT-3

Claude

Gemini

LLAMA 3

https://sebastianraschka.com/llm-architecture-gallery/

Gemma

NVIDIA Nemotron

Mistral

Qwen

OLMo

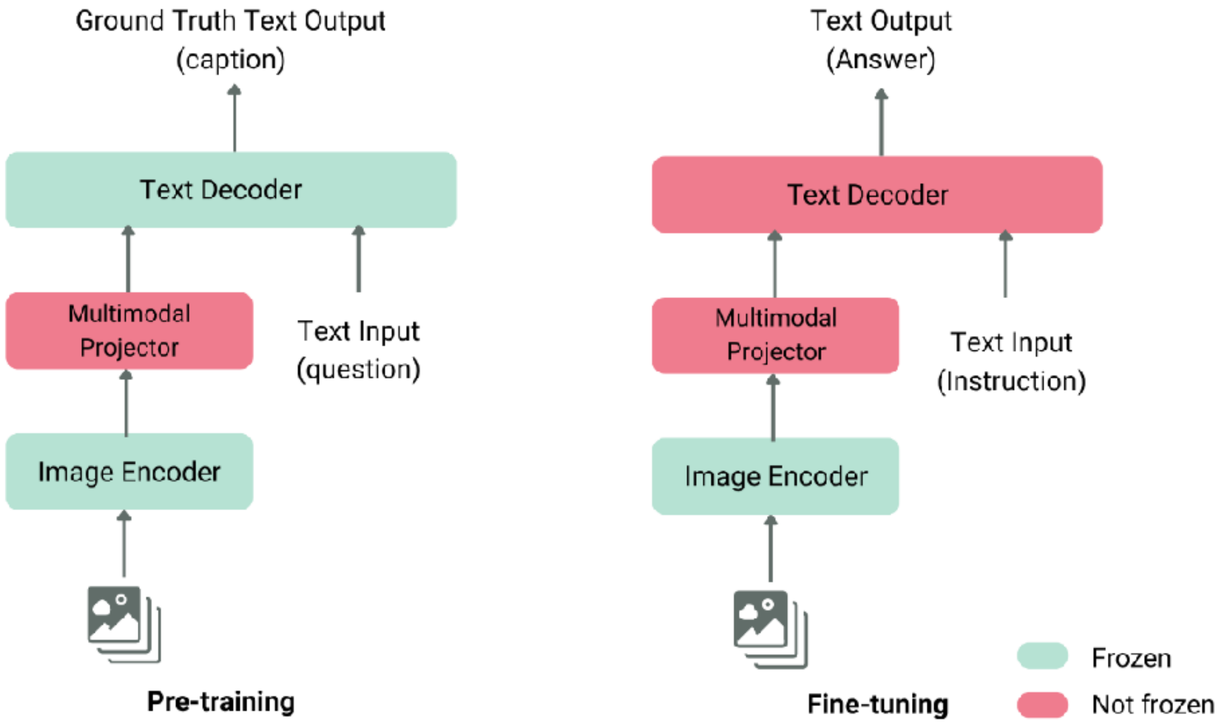
deepseek

Kimi

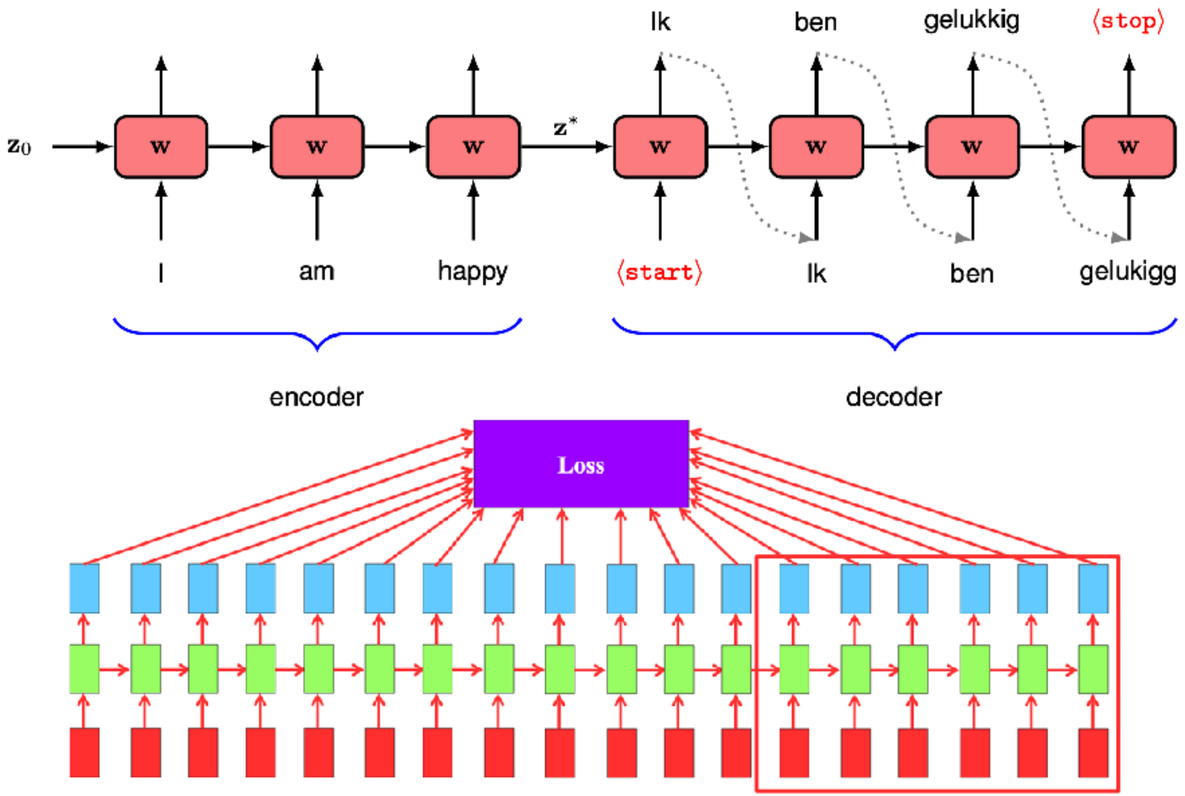
# Transformer

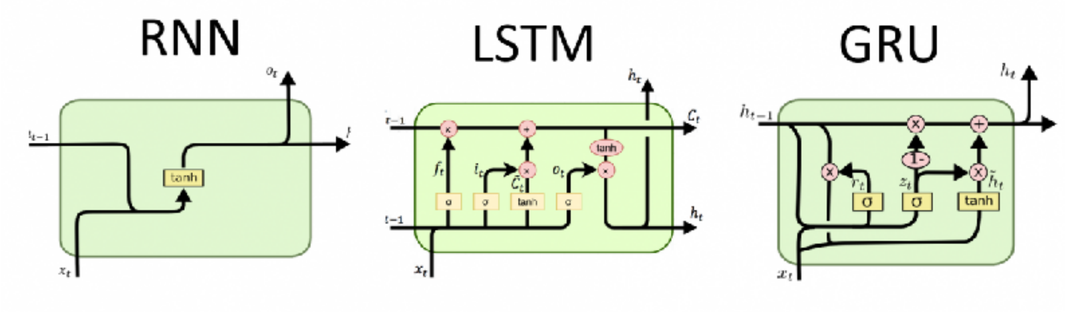
## Vision-Language Model (VLM)

- **Vision-Language Model (VLM)**: képeket is értelmezni képes nyelvi modell
- Általában egy dekóder transformer, ami egy projektor hálón keresztül képi tokeneket is feldolgoz
- A legtöbb LLM ma már valójában VLM!



# Kitérő: Rekurrens Neurális Hálók



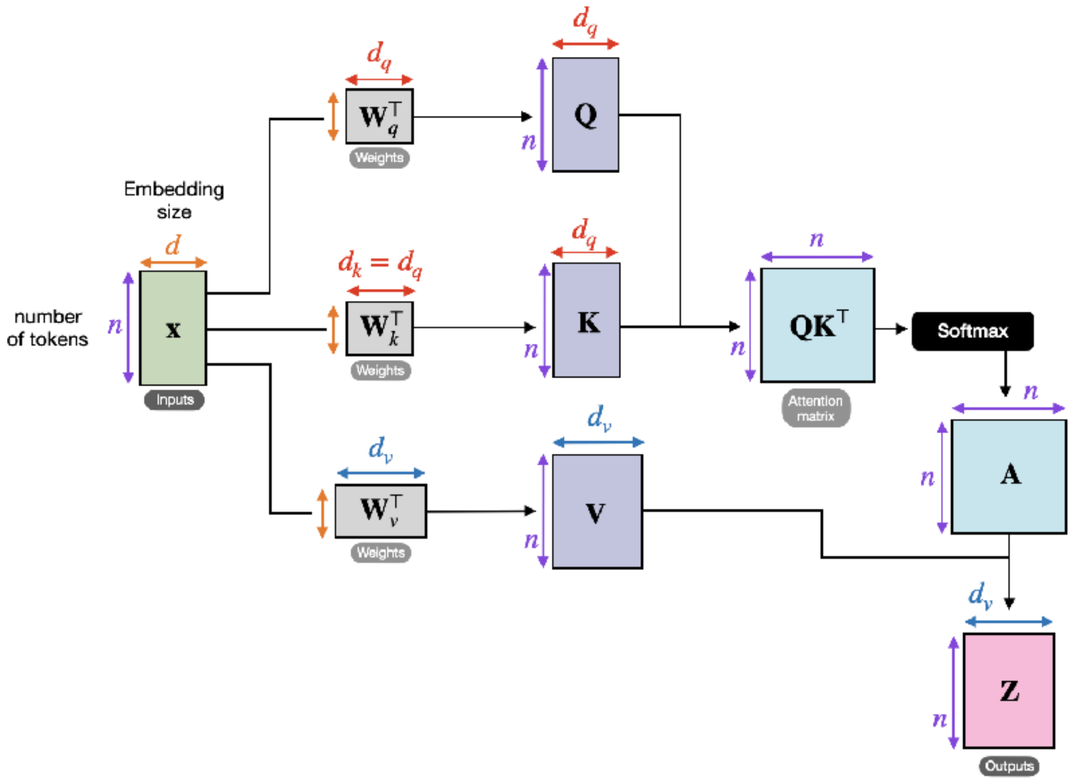

A transzformerek térhódítása előtt:
**Rekurrens Neurális Hálók (RNN)**

A kontextust csak egy "rejtett" látens állapotba tömörítve látják — előbb-utóbb “felejtenek”…
(modern, memóriával rendelkező variánsok: LSTM/GRU)

Nehéz volt őket tanítani (backprop-in-time)…

# Transformer

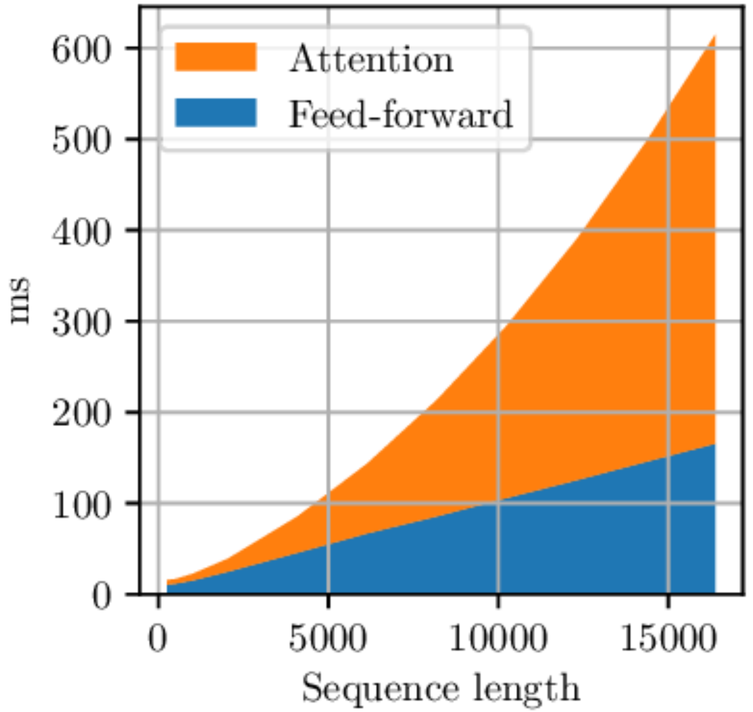
## GPU implementáció




Transformer kiértékelés = nagy mátrixok szorzása és összeadása
SIMD párhuzamos hardver (GPU) számára ideális feladat!

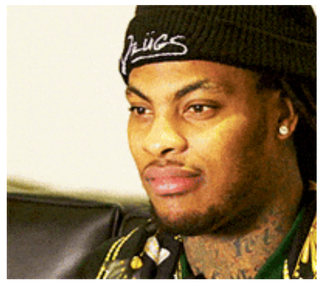
# Transformer

## A feketeleves…

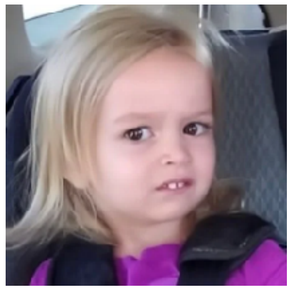



**A tokenek számával (kontextus ablak hosszával)**
**az attention számítás/memóriaigénye $O(N^2)$ szerint nő!**

Google/Meta/OpenAI/stb.:



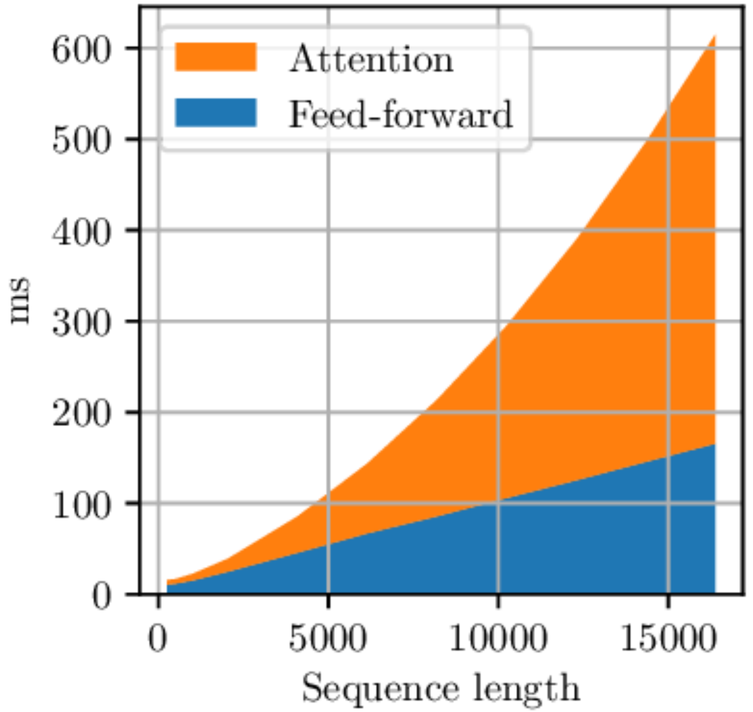
Mindenki más:



A gyakorlatban sok-sok optimalizációs trükk:
KV cache, súlyok durva kvantálása (akár <8 bit!!), Mixture-of-Experts, lineáris (softmax nélküli) attention…

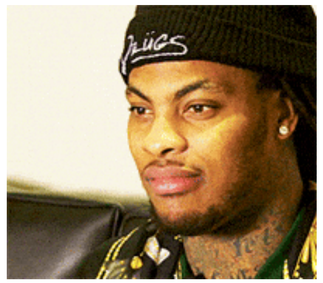
# Transformer

## A feketeleves…

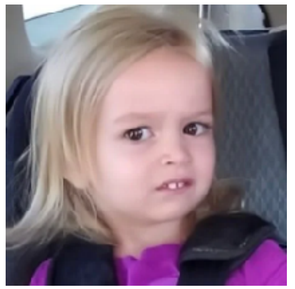



**A tokenek számával (kontextus ablak hosszával)**
**az attention számítás/memóriaigénye $O(N^2)$ szerint nő!**

Google/Meta/OpenAI/stb.:



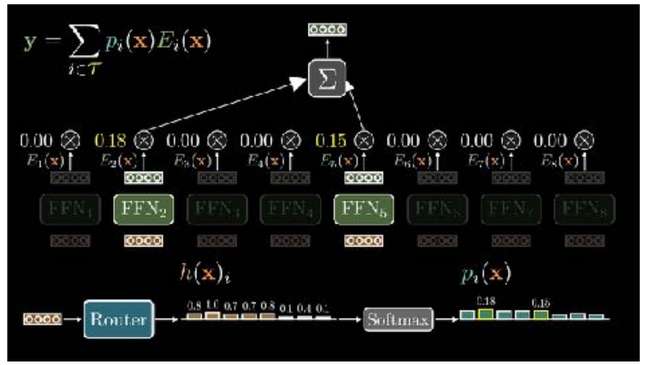
Mindenki más:

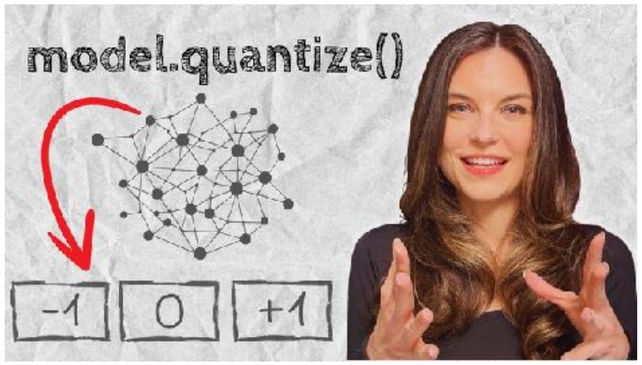


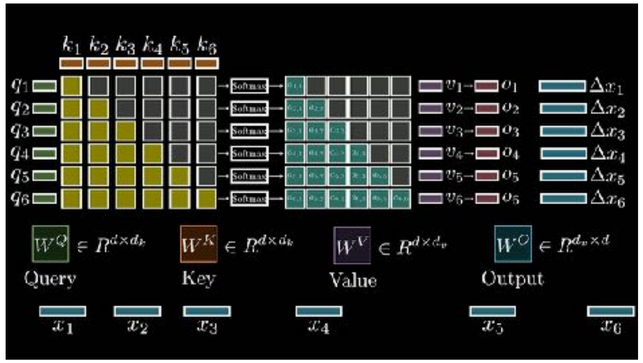
A gyakorlatban sok-sok optimalizációs trükk:
KV cache, súlyok durva kvantálása (akár <8 bit!!), Mixture-of-Experts, lineáris (softmax nélküli) attention…

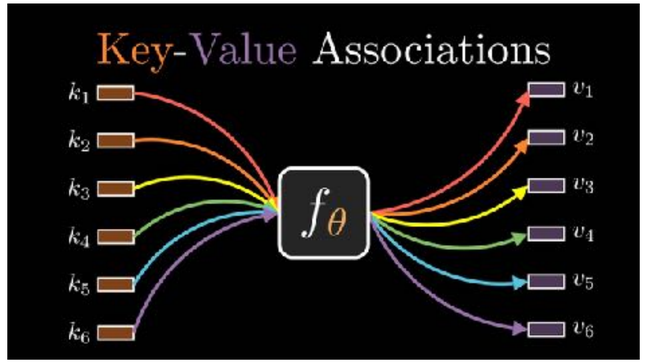
# Transformer

## Optimalizációs trükkök (Érdeklődőknek)

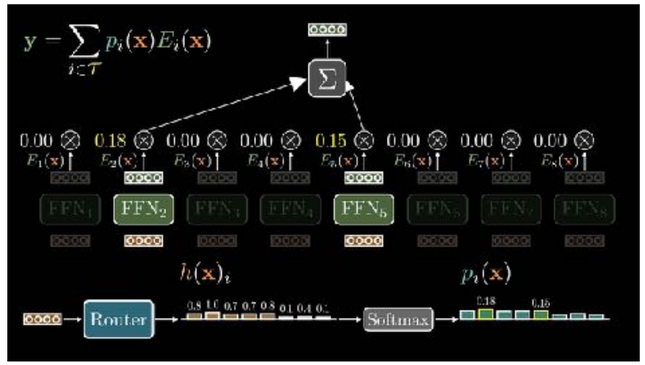



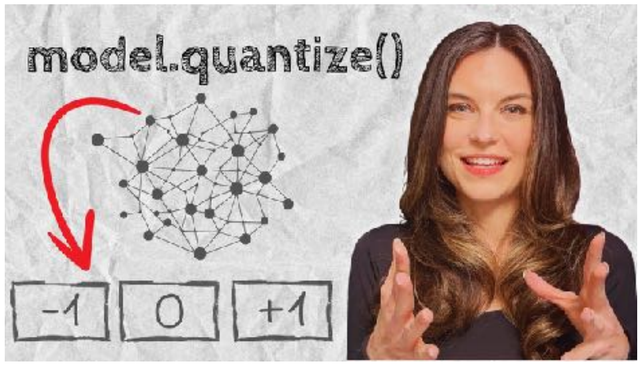



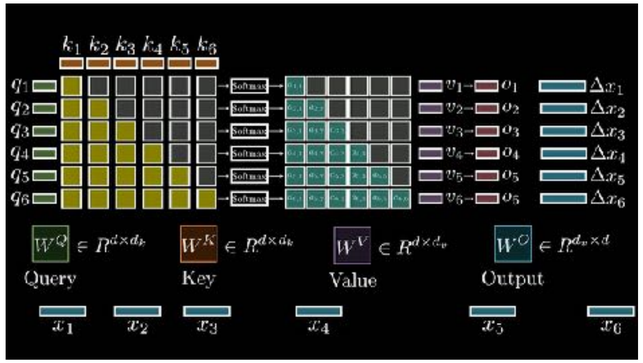

https://ngrok.com/blog/quantization

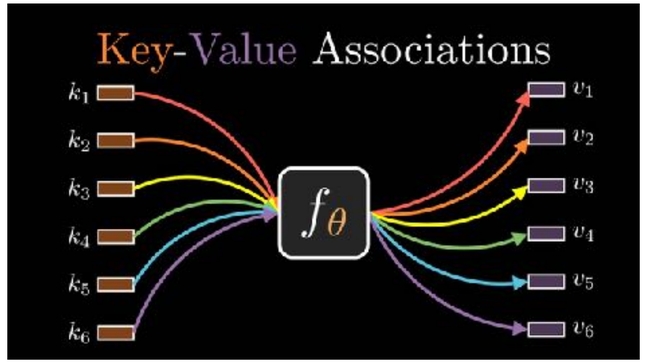
# Transformer

## Optimalizációs trükkök (Érdeklődőknek)

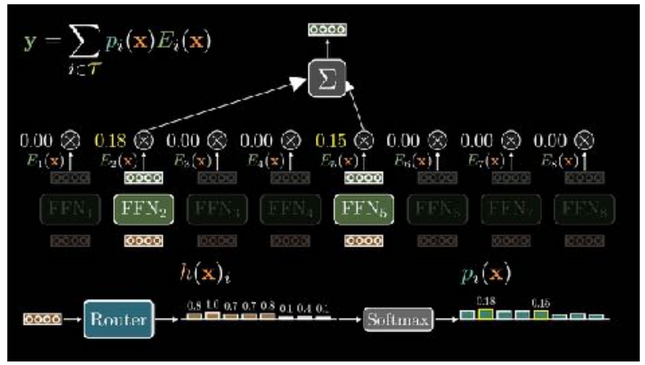



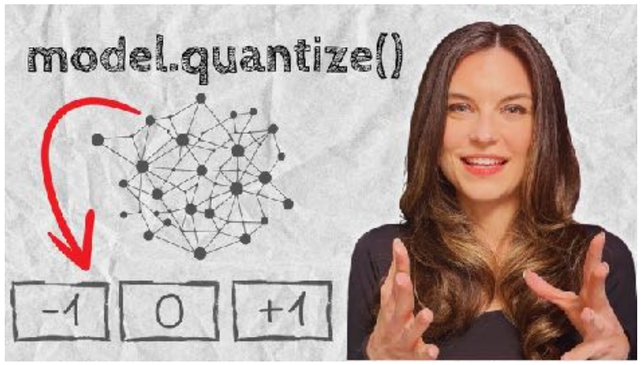



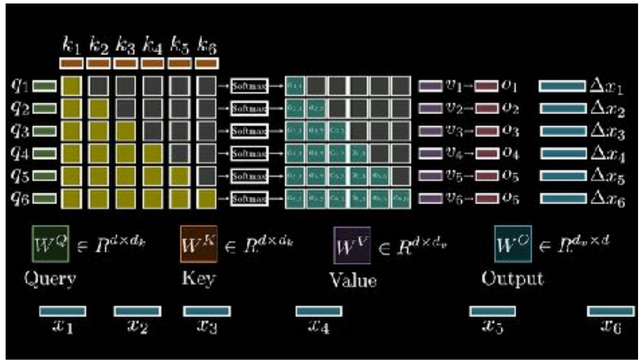

https://ngrok.com/blog/quantization

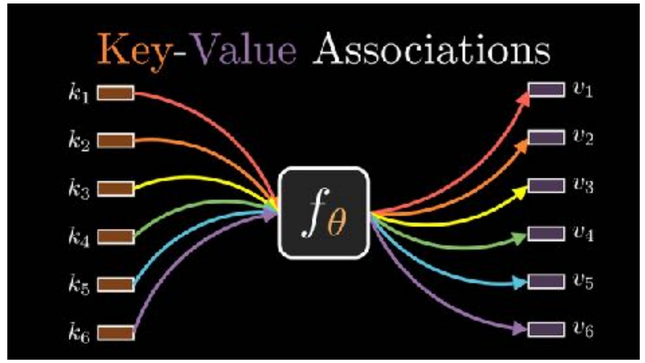
# Transformer

## Optimalizációs trükkök (Érdeklődőknek)

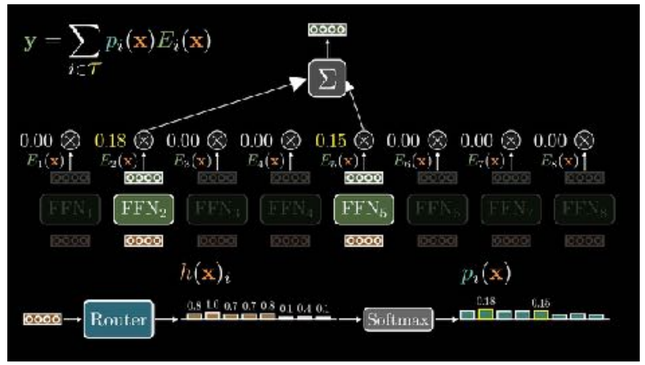



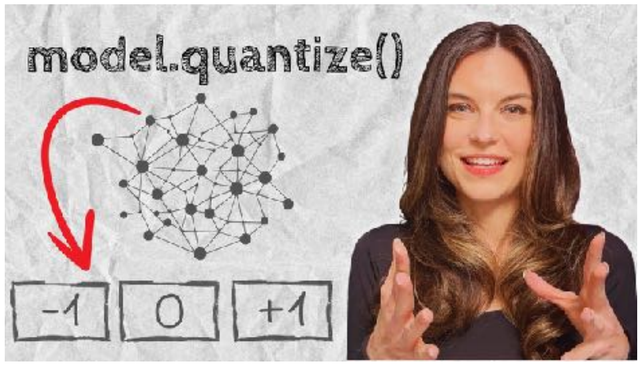



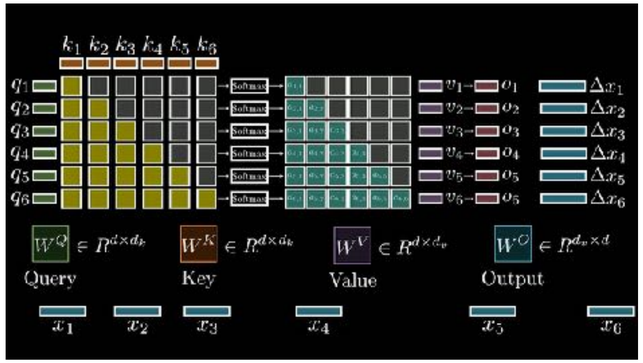

https://ngrok.com/blog/quantization

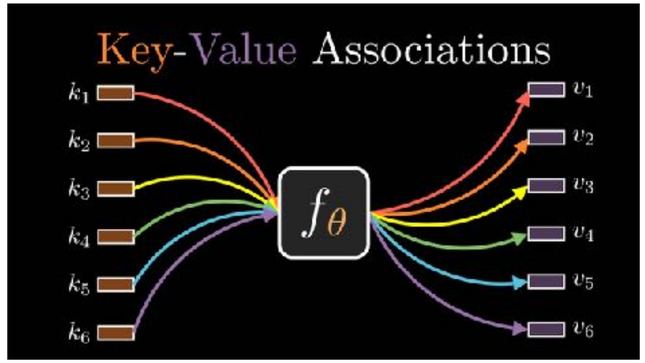
# Transformer

## Optimalizációs trükkök (Érdeklődőknek)

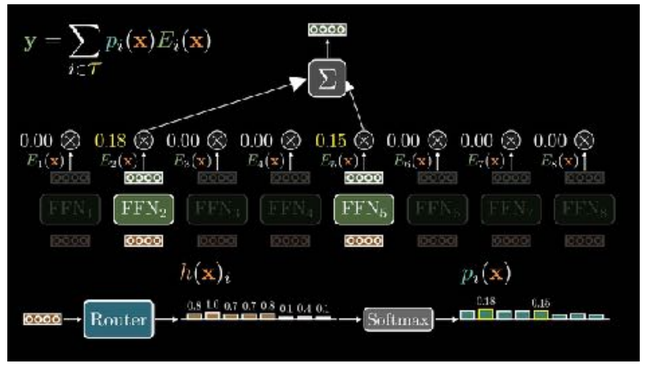



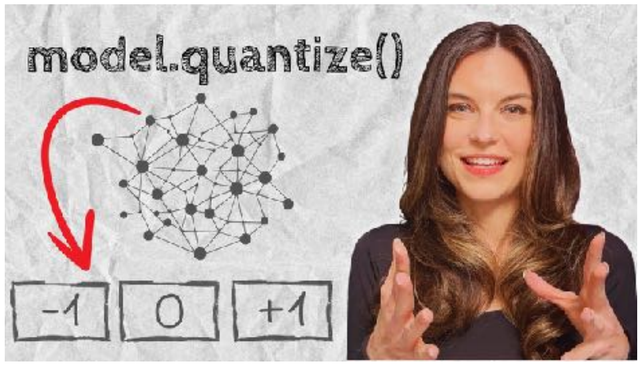



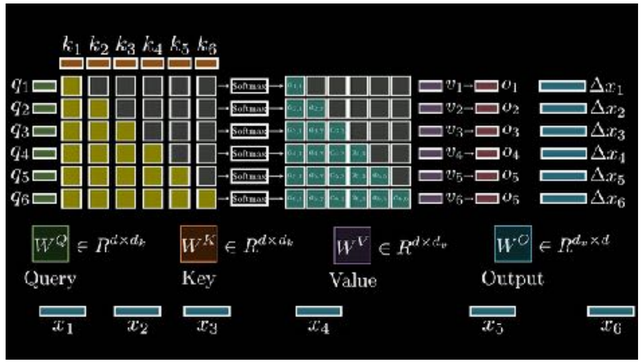

https://ngrok.com/blog/quantization

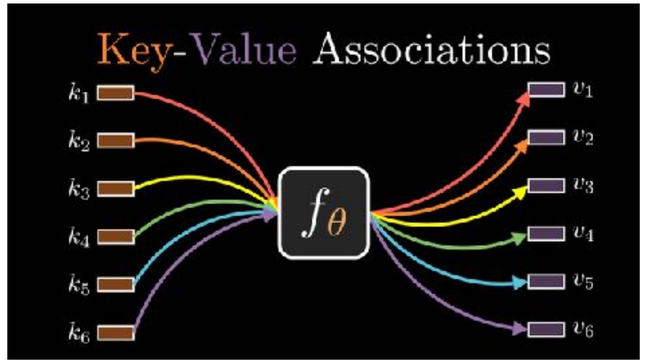
# Transformer

## Optimalizációs trükkök (Érdeklődőknek)

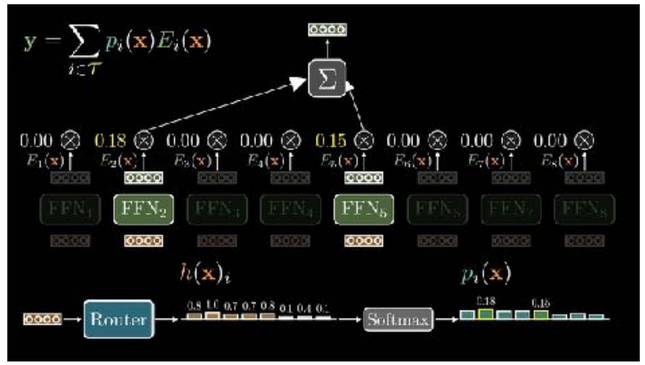



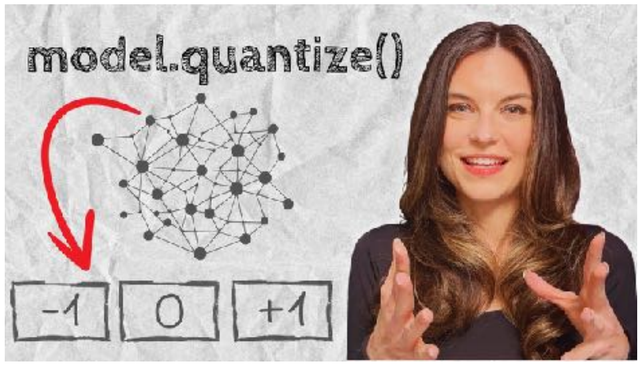



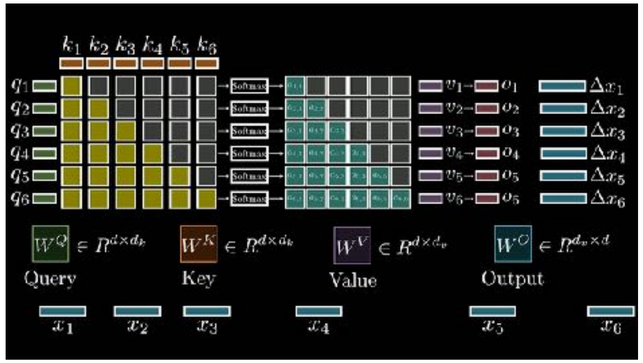

https://ngrok.com/blog/quantization

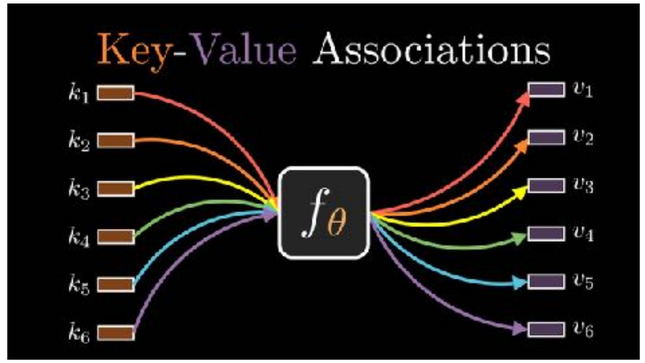
# Transformer

## Optimalizációs trükkök (Érdeklődőknek)


https://ngrok.com/blog/quantization

# Transformer

## Előnyök és hátrányok

**A Jó**

- Globális kontextus
- Input mérete (tokenek száma) tetszőleges lehet
- Dinamikus súlyozás, “in-context learning”
- Multimodális (“moslékot is eszik” © Szemenyei M. )
- Csak “enyhén” nemlineáris
- Nagyon jól párhuzamosítható
- Korlátlanul skálázható a rétegek száma/mérete

**A Rossz**

- ***Brutális*** számításigény ($O(N^2)$)
- A lokalitást is meg kell tanulnia
- A tanítás *nagyon* sok adatot igényelhet

# Transformer

## Alkalmazások — Szövegfeldolgozás

- A szöveget először **tokenizáljuk**…
  - Pl. Byte-Pair Encoding (BPE)

Peter Piper picked a peck of pickled peppers

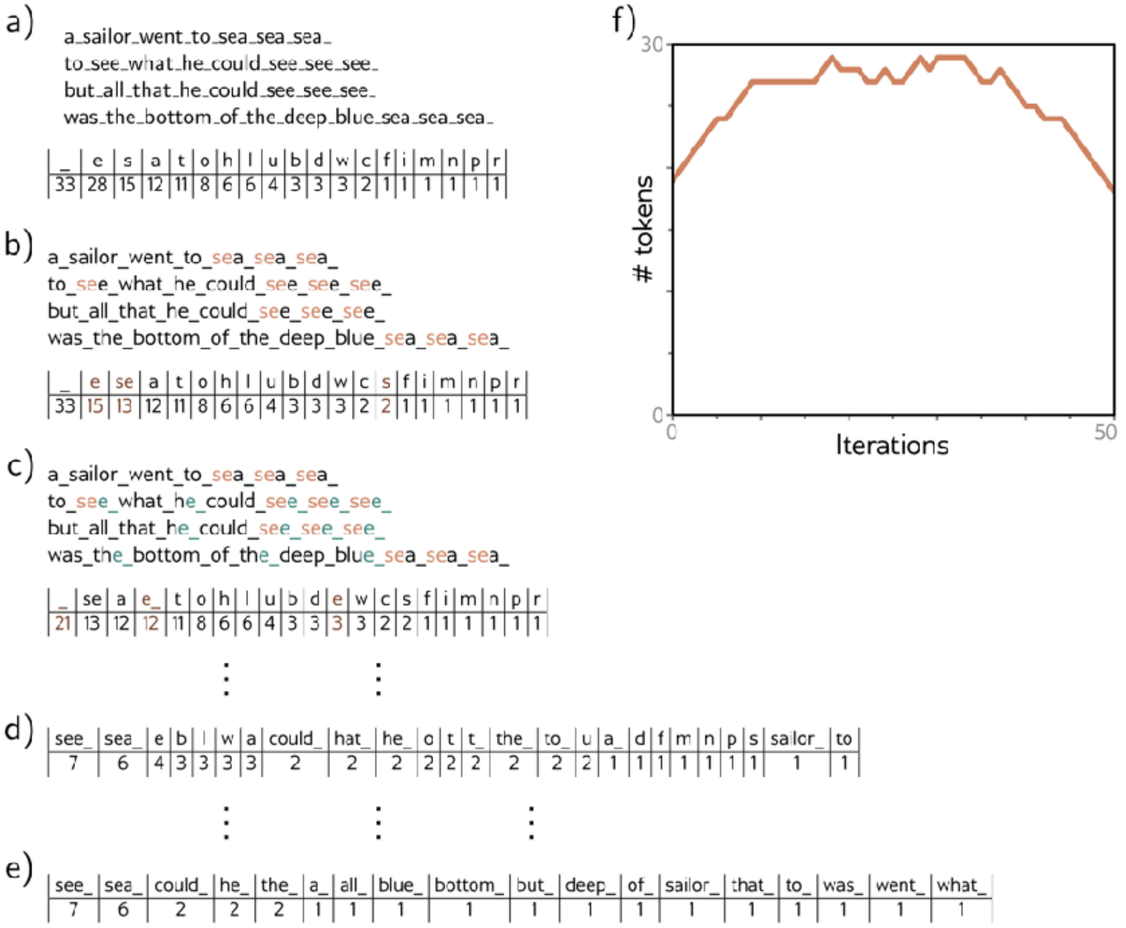
Peter Piper picked a peck of pickled peppers

Peter Piper picked a peck of pickled peppers

Peter Piper picked a peck of pickled peppers

Peter Piper picked a peck of pickled peppers

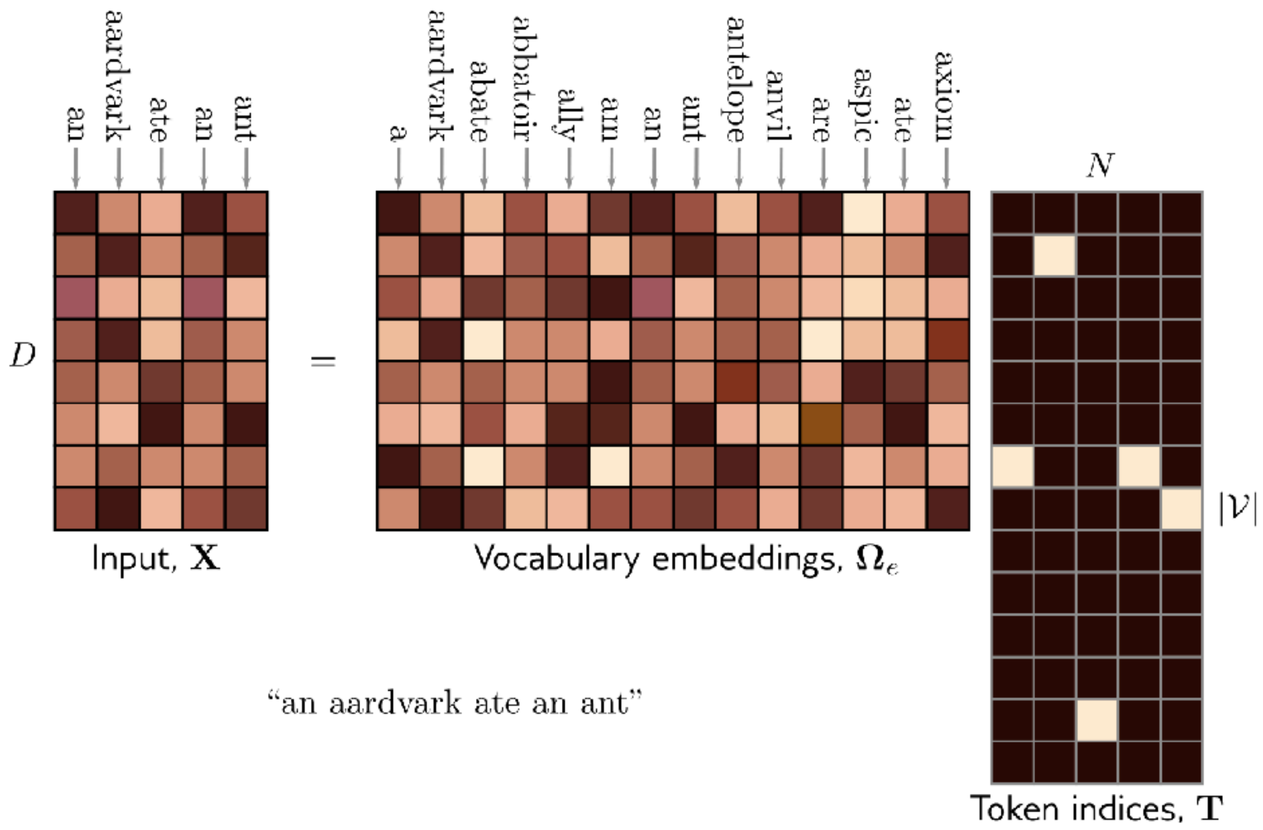
Peter Piper picked a peck of pickled peppers



# Transformer

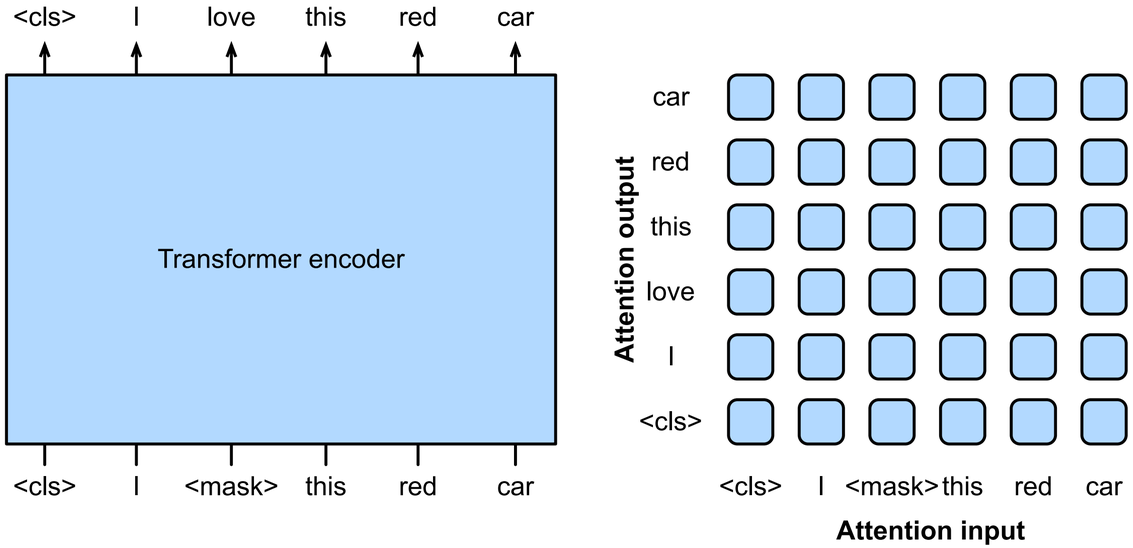
## Alkalmazások — Szövegfeldolgozás

- A tokeneket vektorokra képezzük le egy enkóderrel (beágyazás, **embedding**)
- Transformer alapú enkóderek:
  - BERT
  - T5
  - CLIP

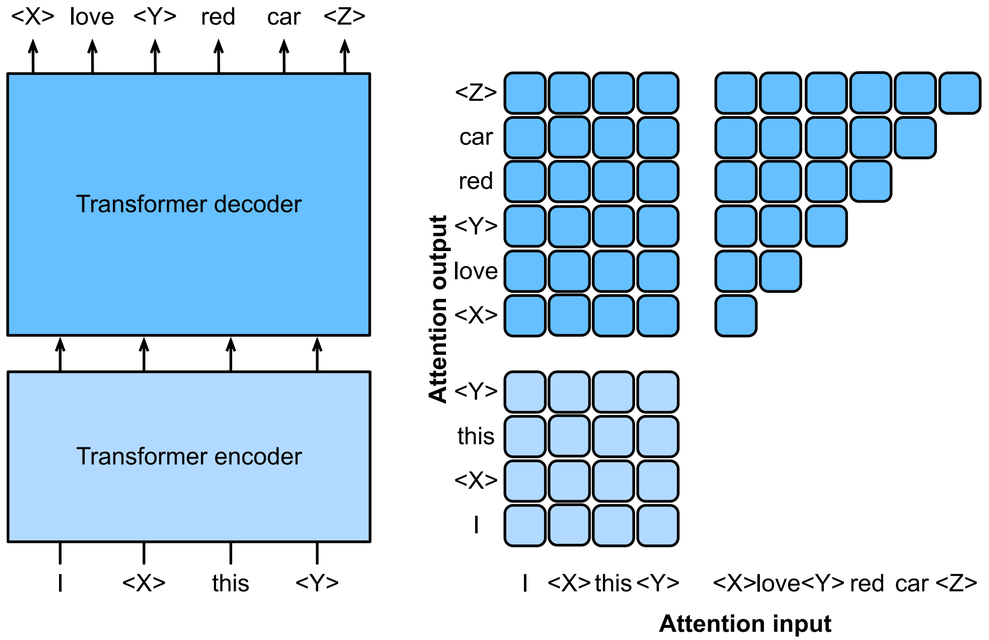


# Transformer

## Alkalmazások — Szövegfeldolgozás




Bidirectional Encoder Representations from Transformers (**BERT**): *Enkóder* architektúra



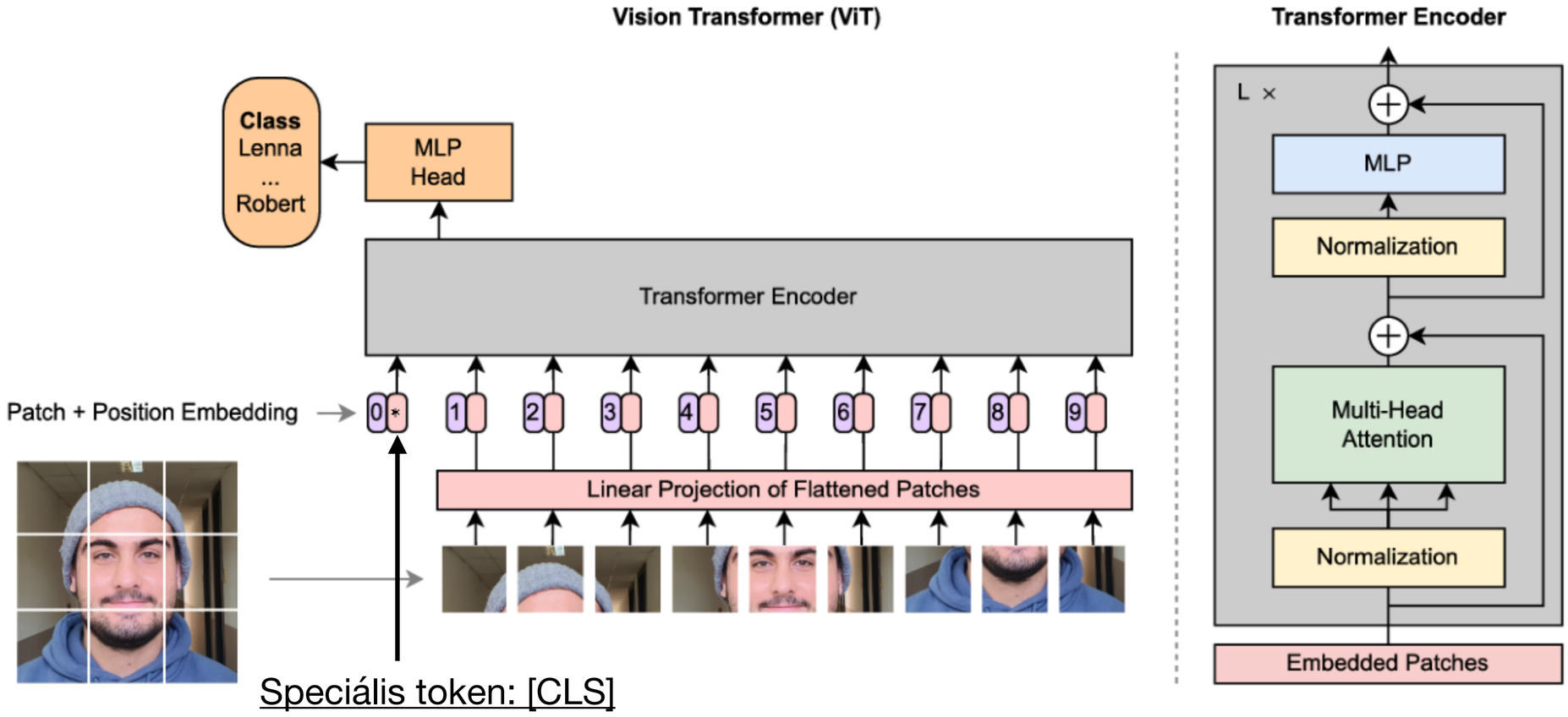

Text-to-Text Transfer Transformer (**T5**): *Enkóder-dekóder* architektúra

# Transformer

## Alkalmazások — Vision Transformer (ViT)

AN IMAGE IS WORTH 16X16 WORDS:
TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

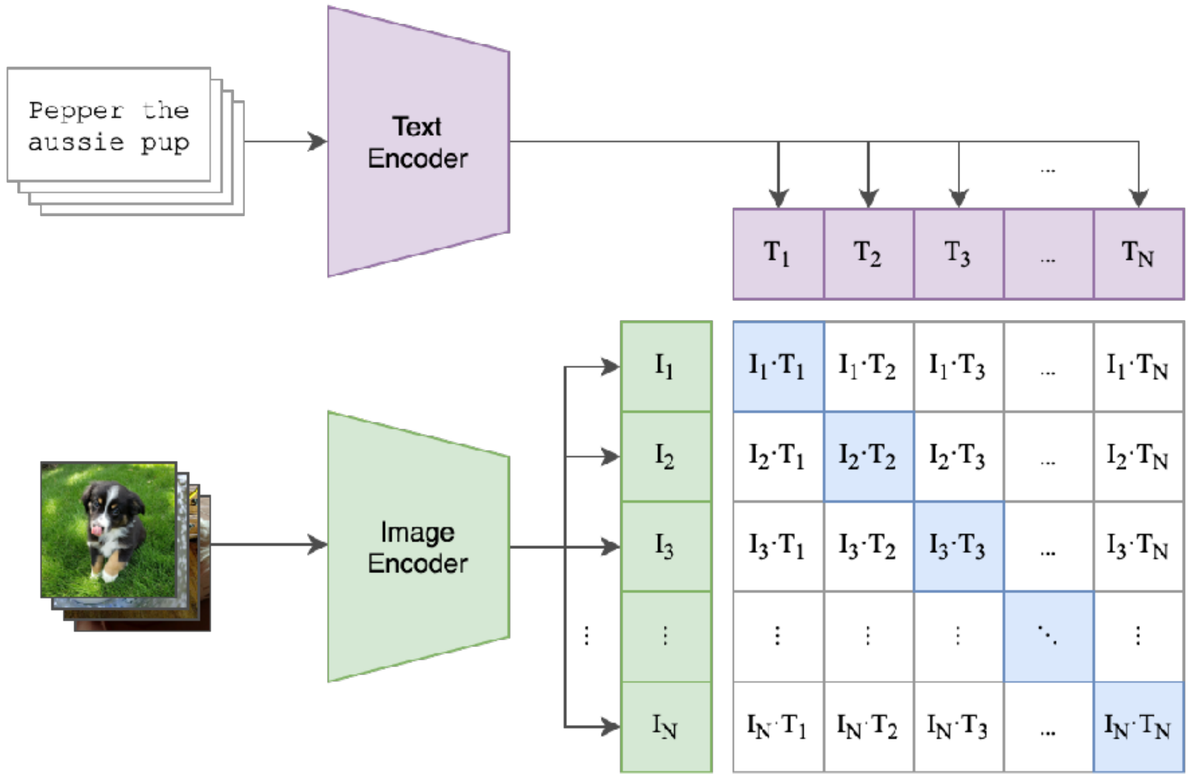
**Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†**
*equal technical contribution, †equal advising
Google Research, Brain Team



# Transformer

## Alkalmazások — ViT — CLIP

- **Contrastive Language-Image Pretraining (CLIP)**
- Szöveg es kép transformer enkódereket *egyszerre* tanítunk kép-szöveg párokon
- Kontrasztív tanítás: összetartozó párok beágyazása hasonlítson, nem összetartozóké különbözzön (skalárszorzat szerint)

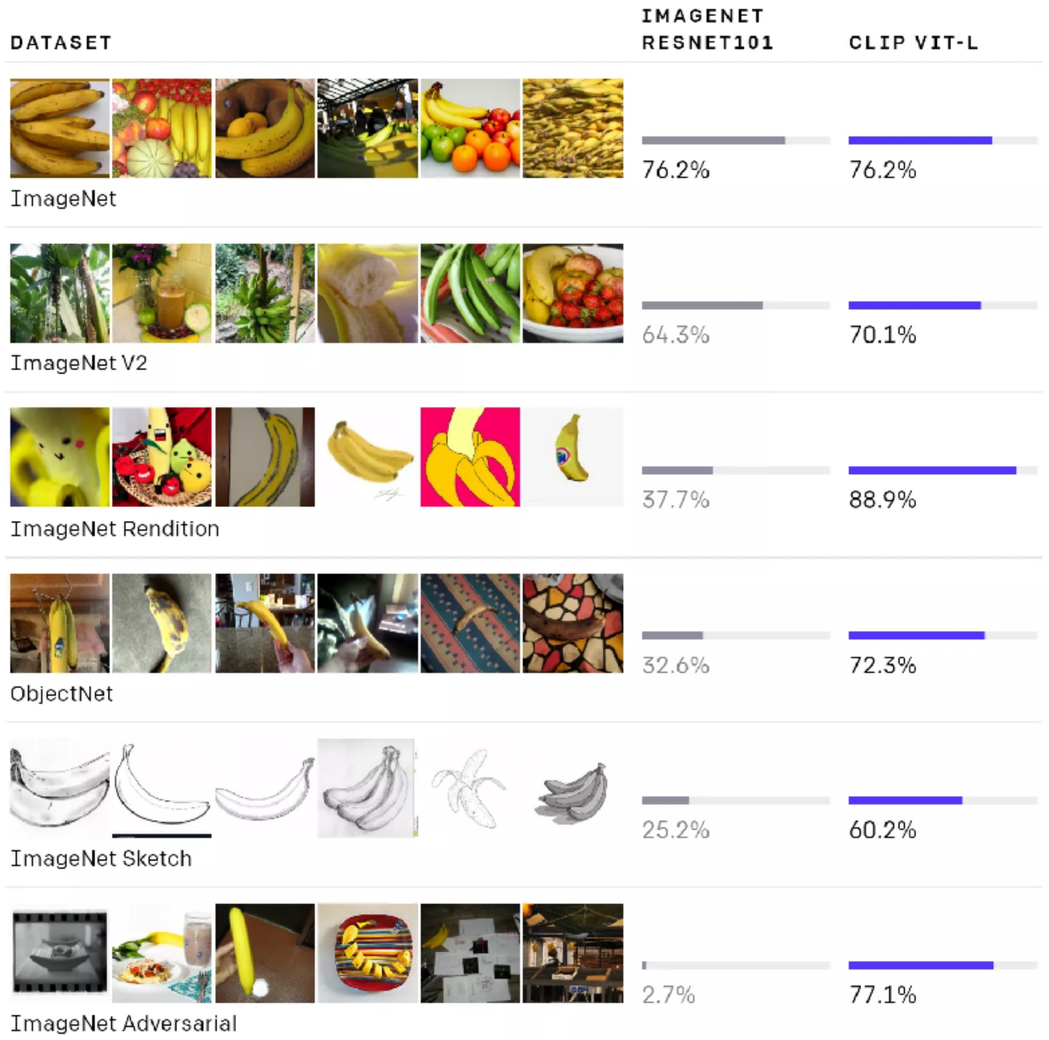



https://openai.com/index/clip/

# Transformer

## Alkalmazások — ViT — CLIP

- A CLIP pl. képek osztályozásában és keresésében is nagyon erős!
  - Szimplán kiválasztjuk a legjobban illeszkedő címkét / képet.
- Architektúra: GPT2 + ViT — 300 millió paraméter!
- Tanító adat: 400 millió(!) kép-szöveg pár
- Tanítás: többszáz V100 GPU-n, több héten át — becsült költsége > 1 millió USD(!!!)



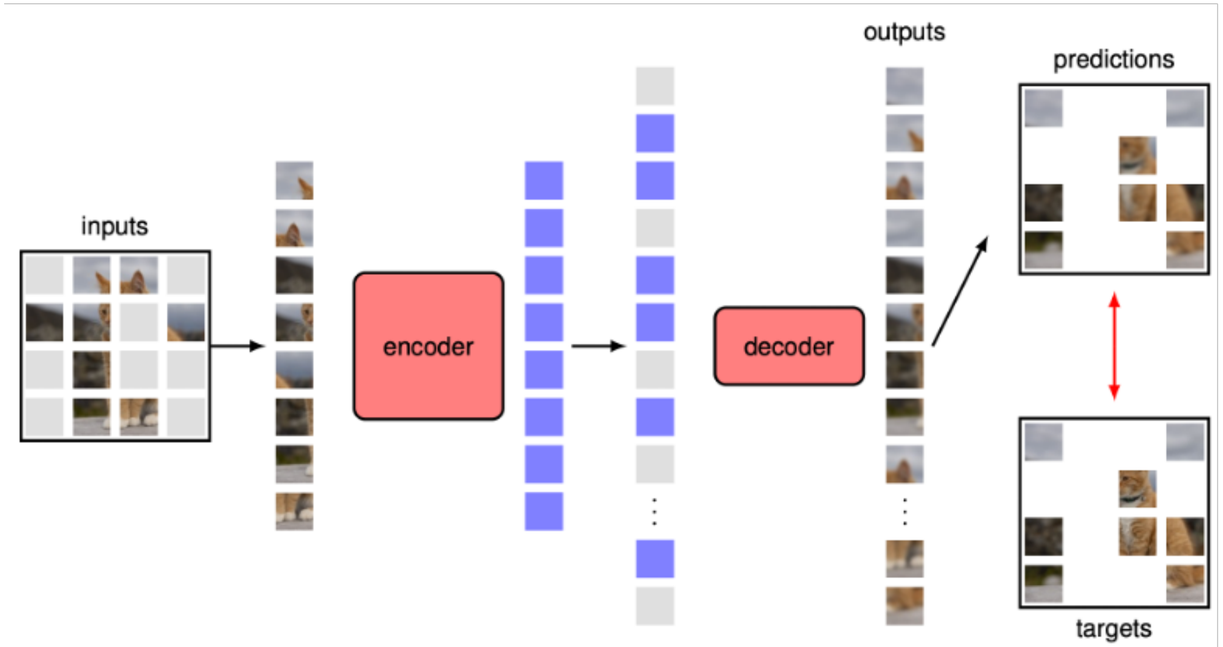
# Transformer

## Alkalmazások — ViT — MAE

- Szöveges enkódereken sokat javított az *önfelügyelt* (maszkolt rekonstrukciós) tanítás
- Masked Autoencoder (**MAE**): maszkoljuk ki a kép egy részét (akár 80+%-át!!!), tanuljuk meg rekonstruálni
- Tipikusan ViT architektúrájú

**Masked Autoencoders Are Scalable Vision Learners**

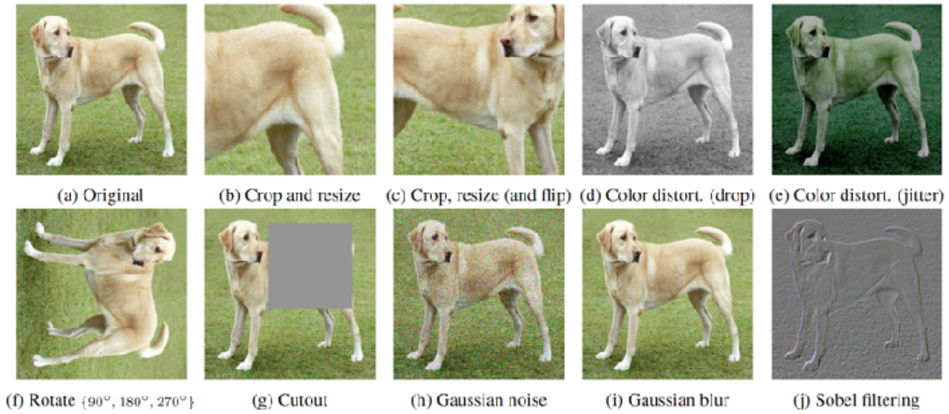
Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick




# Transformer

## Alkalmazások — ViT — SimCLR

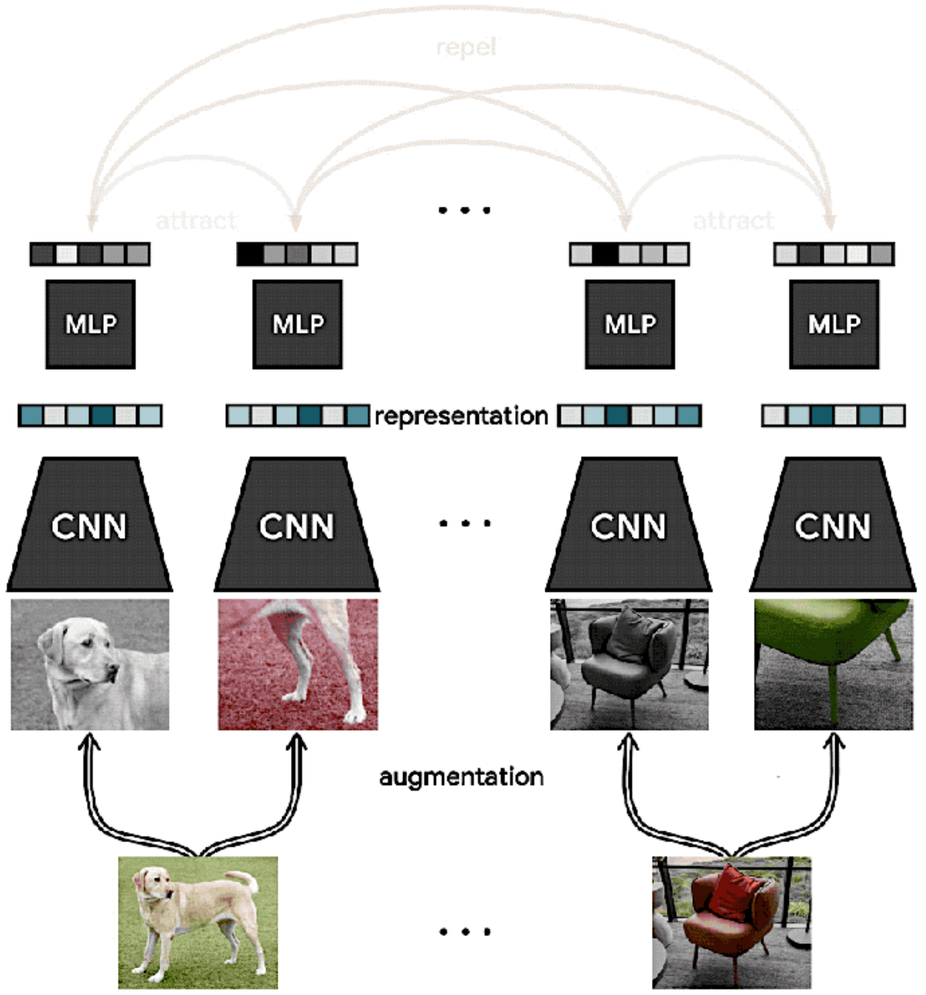
- **SimCLR**: képi enkóder (CNN vagy ViT) önfelügyelt tanítása, különféle augmentációkkal:
  - Random crop, szín torzítás, blur, stb.




- **Kontrasztív loss**: azonos képek különböző augmentációira produkáljon hasonló látenst, különböző képekre különbözőt

A Simple Framework for Contrastive Learning of Visual Representations

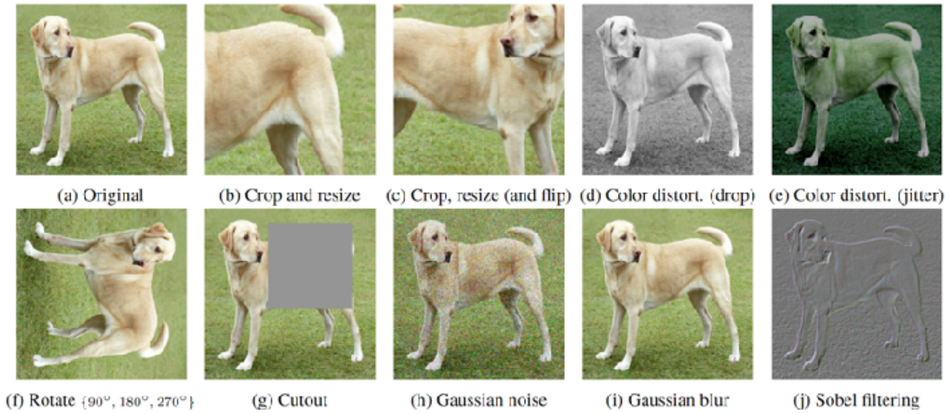
Ting Chen [1] Simon Kornblith [1] Mohammad Norouzi [1] Geoffrey Hinton [1]



# Transformer

## Alkalmazások — ViT — SimCLR

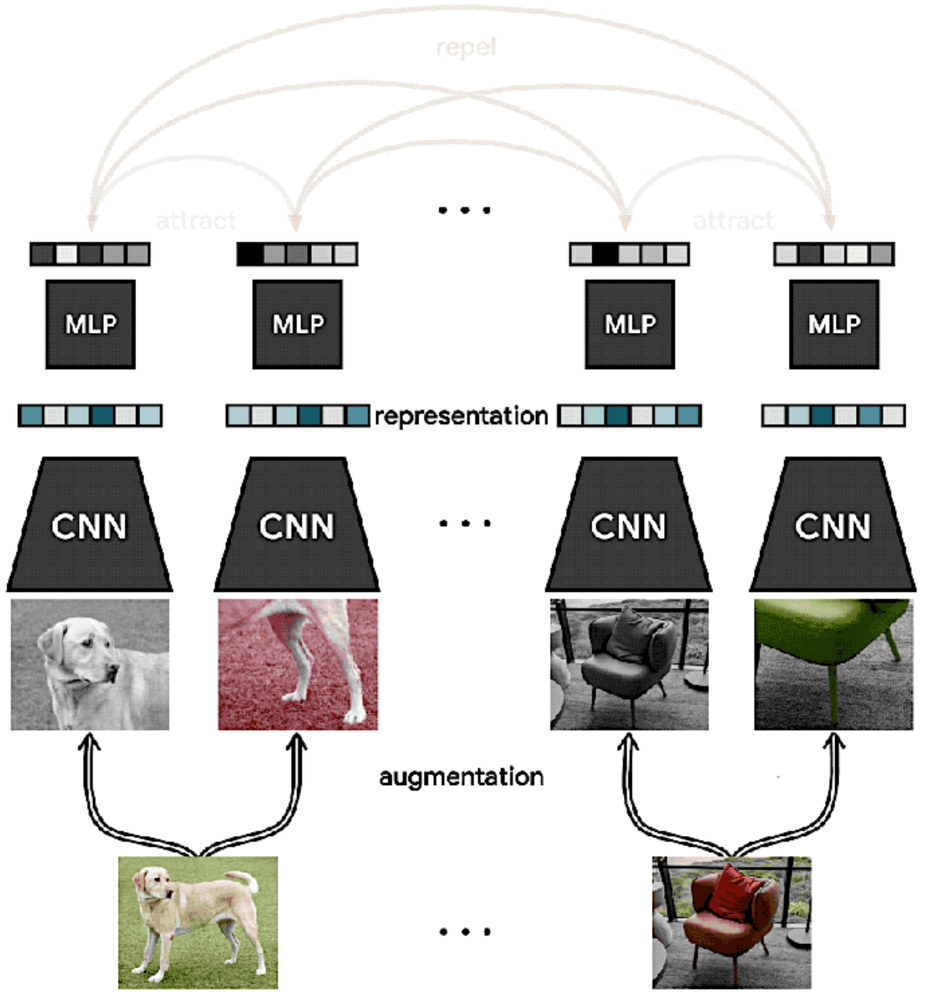
- **SimCLR**: képi enkóder (CNN vagy ViT) önfelügyelt tanítása, különféle augmentációkkal:
  - Random crop, szín torzítás, blur, stb.




- **Kontrasztív loss**: azonos képek különböző augmentációira produkáljon hasonló látenst, különböző képekre különbözőt

A Simple Framework for Contrastive Learning of Visual Representations

Ting Chen [1] Simon Kornblith [1] Mohammad Norouzi [1] Geoffrey Hinton [1]



# Transformer

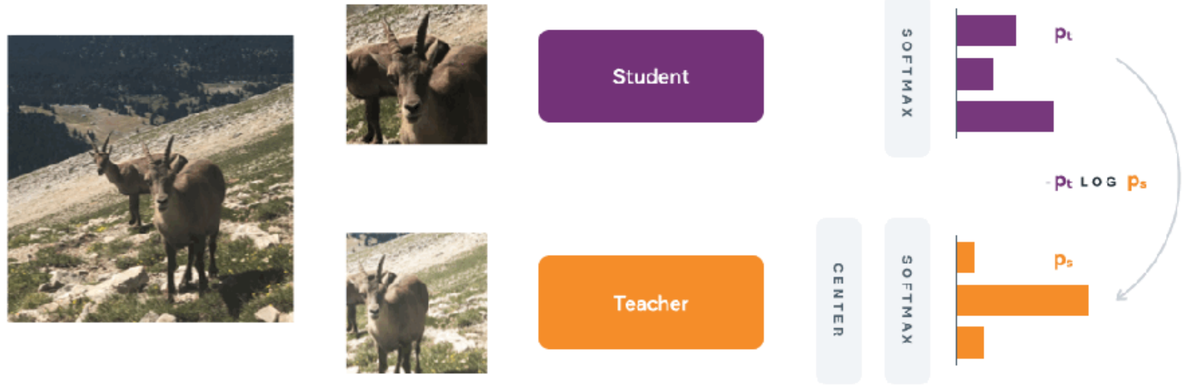
## Alkalmazások — ViT — CLIP

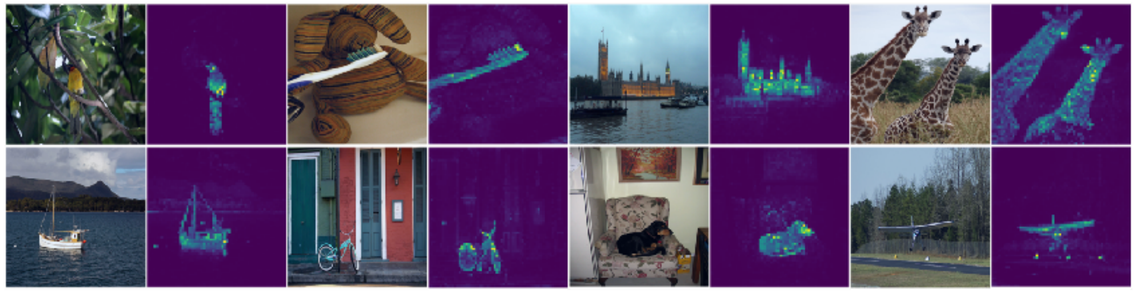
- Self-Distillation with No Labels (**DINO**): “trükkös” önfelügyelt tanítási módszer
- Egy kép random cropjait kódoljuk be ugyanazon (ViT) háló két példányával:
  - Diák: ezt tanítjuk
  - Tanár: súlyai a diák súlyainak mozgóátlaga, nem tanítjuk
- (Ön)-“Disztilláció”: a diák próbálja reprodukálni a tanár kimenetét

**Emerging Properties in Self-Supervised Vision Transformers**

Mathilde Caron[1,2] Hugo Touvron[1,3] Ishan Misra[1] Hervé Jegou[1]
Julien Mairal[2] Piotr Bojanowski[1] Armand Joulin[1]

[1] Facebook AI Research [2] Inria* [3] Sorbonne University



Attention maszkok a [CLS] tokenre

# Transformer

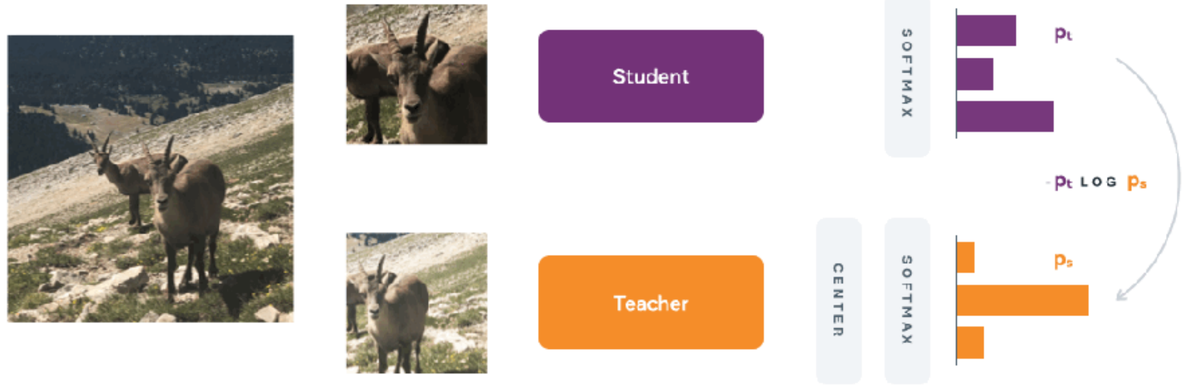
## Alkalmazások — ViT — CLIP

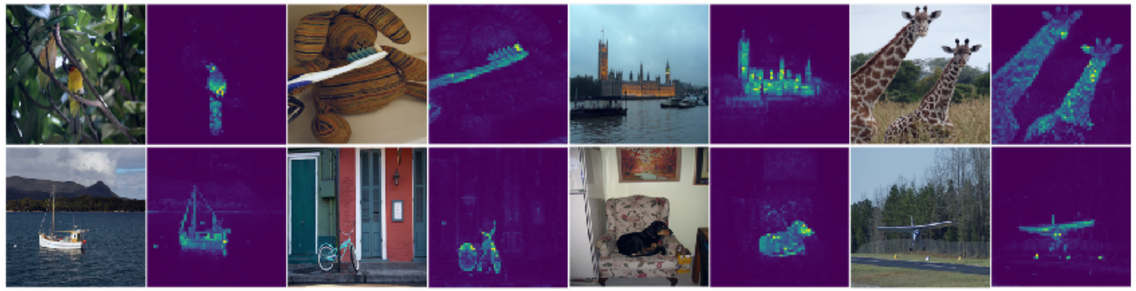
- Self-Distillation with No Labels (**DINO**): “trükkös” önfelügyelt tanítási módszer
- Egy kép random cropjait kódoljuk be ugyanazon (ViT) háló két példányával:
  - Diák: ezt tanítjuk
  - Tanár: súlyai a diák súlyainak mozgóátlaga, nem tanítjuk
- (Ön)-“Disztilláció”: a diák próbálja reprodukálni a tanár kimenetét

**Emerging Properties in Self-Supervised Vision Transformers**

Mathilde Caron[1,2] Hugo Touvron[1,3] Ishan Misra[1] Hervé Jegou[1]
Julien Mairal[2] Piotr Bojanowski[1] Armand Joulin[1]

[1] Facebook AI Research [2] Inria* [3] Sorbonne University

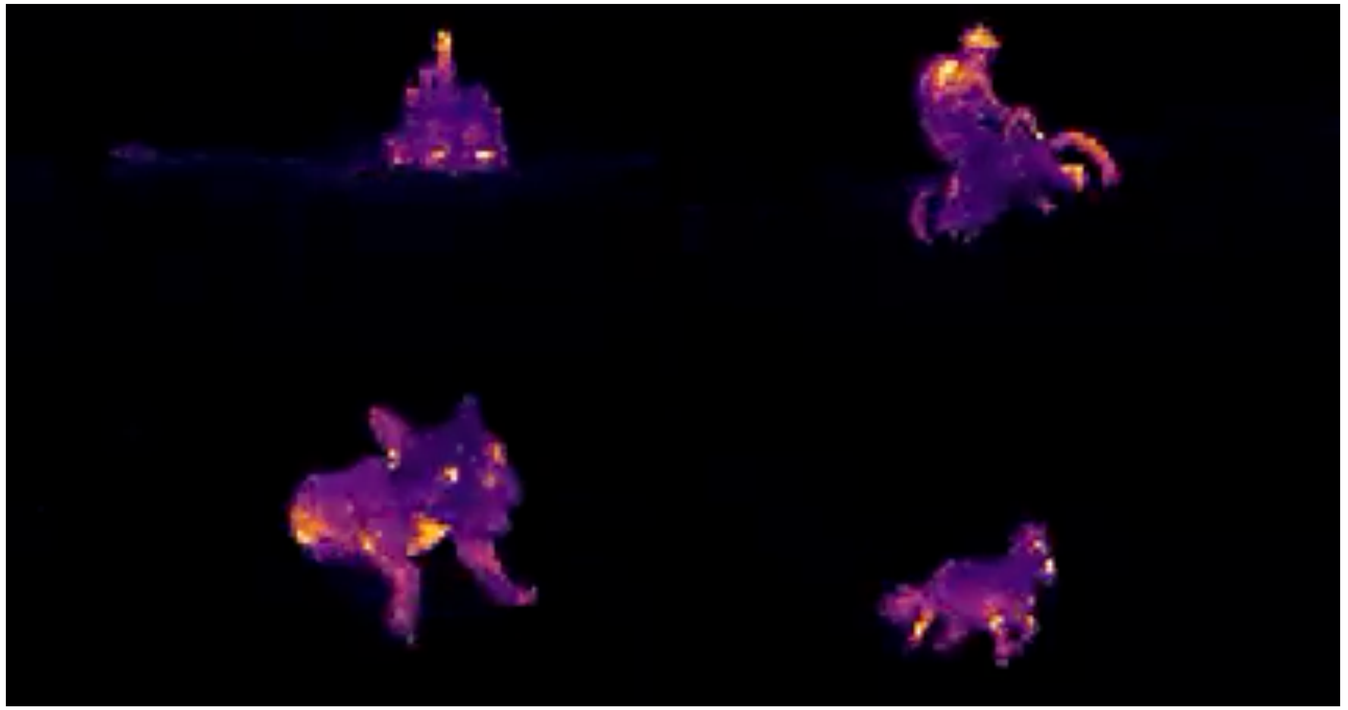


Attention maszkok a [CLS] tokenre

# Transformer

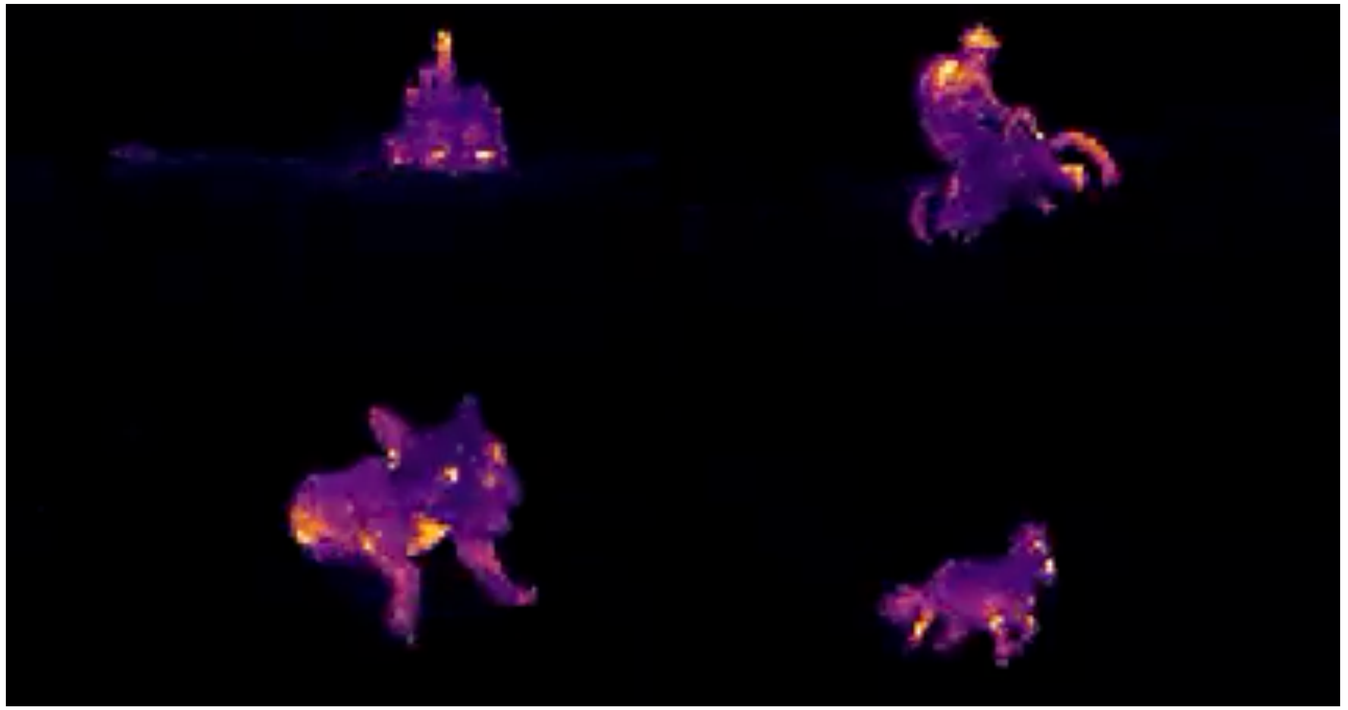
## Alkalmazások — ViT — CLIP



A DINO enkóderek (V1/V2/V3) “state-of-the-art” képfeldolgozást tesznek lehetővé!

# Transformer

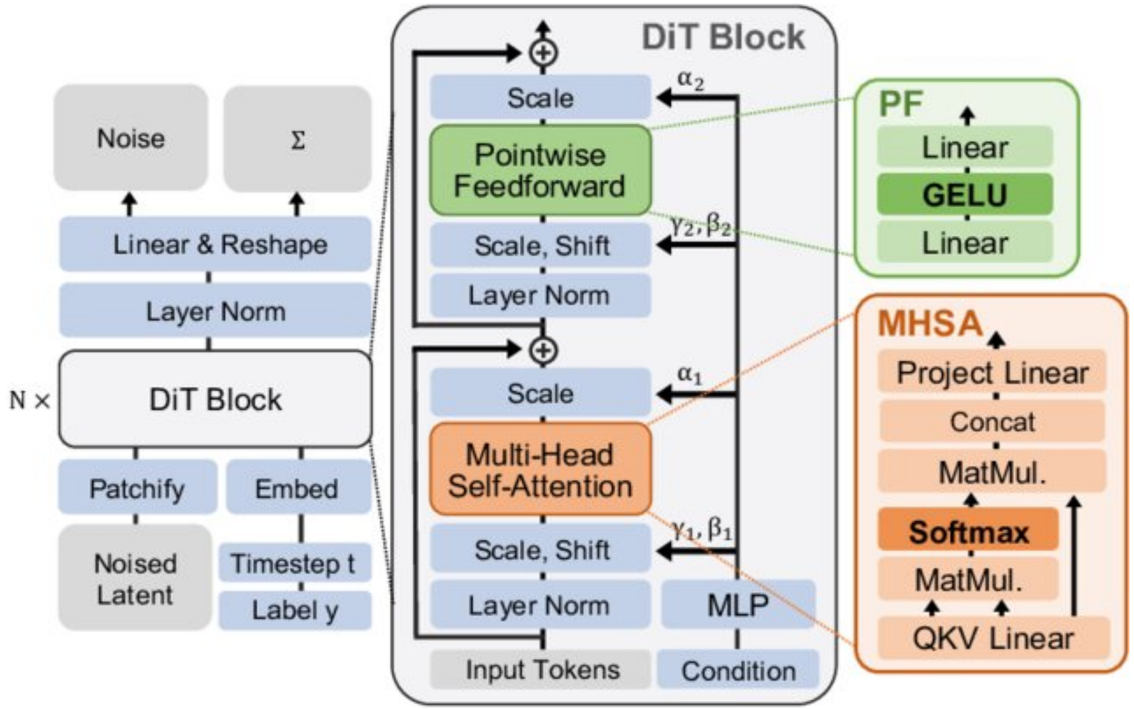
## Alkalmazások — ViT — CLIP



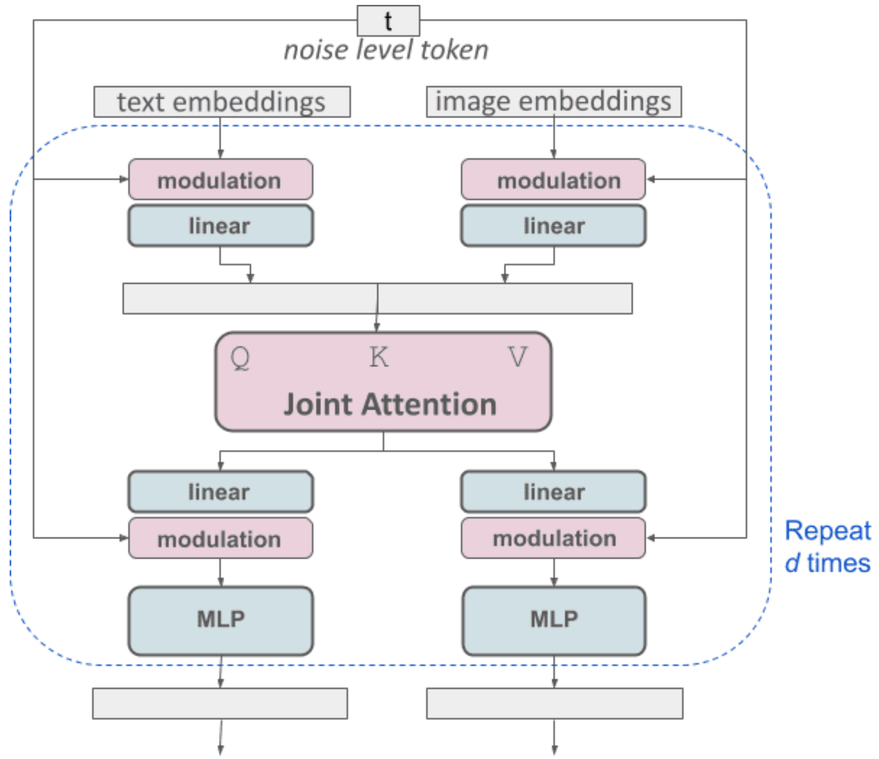
A DINO enkóderek (V1/V2/V3) “state-of-the-art” képfeldolgozást tesznek lehetővé!

# Transformer

## Alkalmazások — Diffusion Transformer



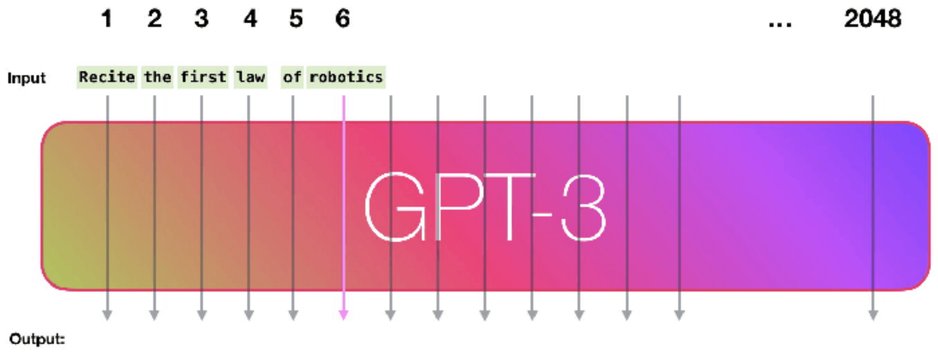

**Diffusion Transformer (DiT)**



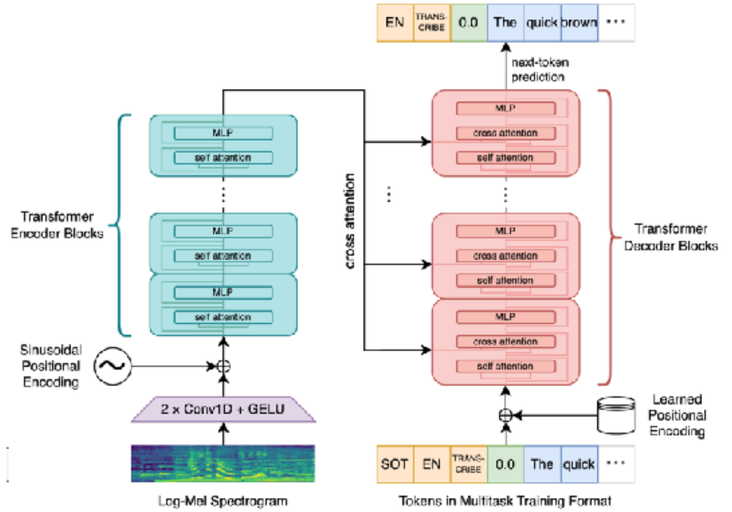

**Multi-Modal DiT (MMDiT)**

# Transformer

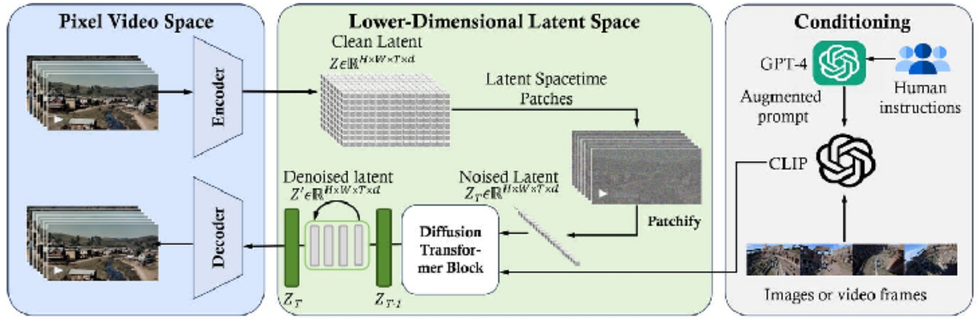
## Alkalmazások



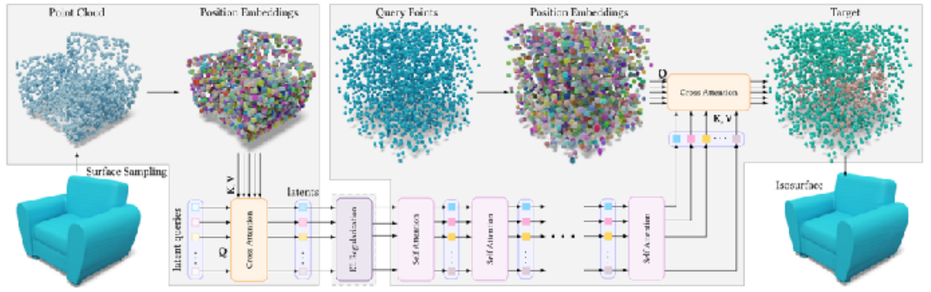

Szöveggenerálás




Hangfelismerés



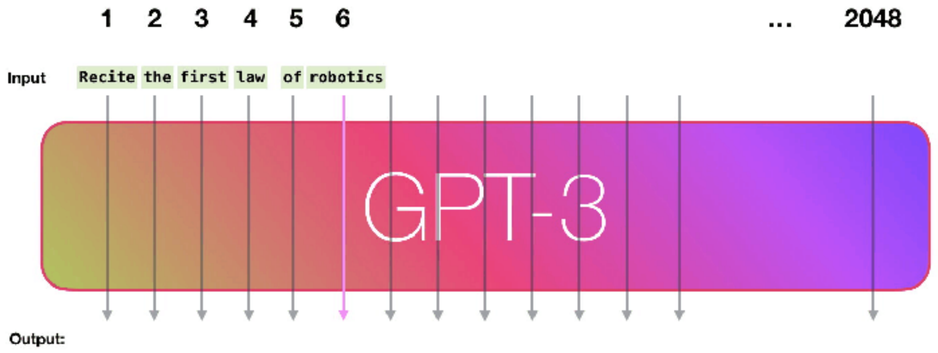

Videógenerálás



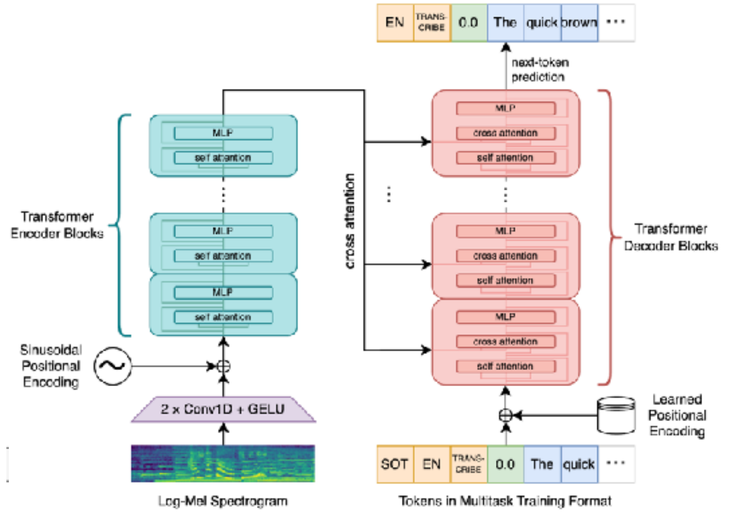

3D generálás

# Transformer

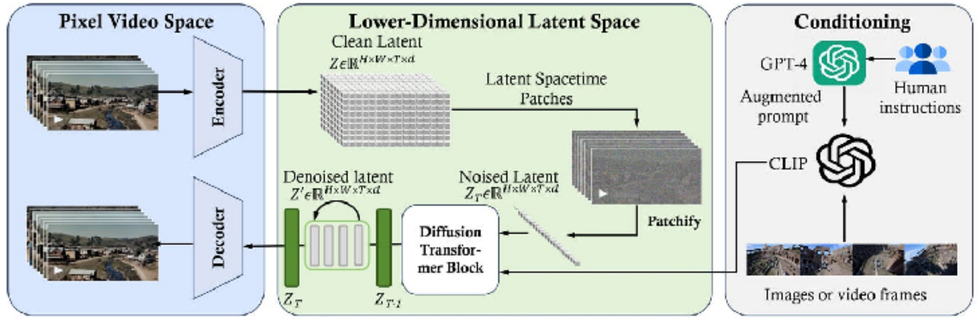
## Alkalmazások



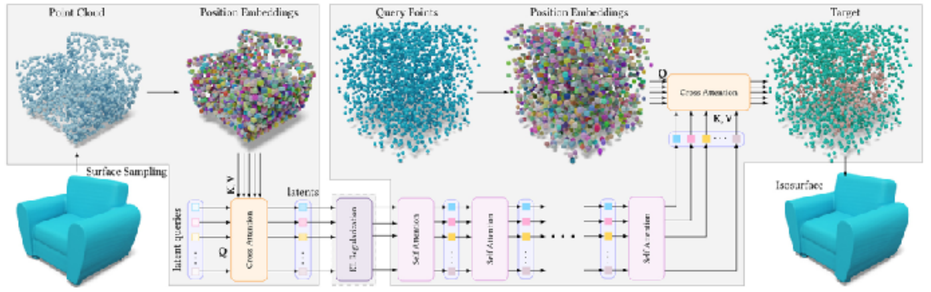

Szöveggenerálás




Hangfelismerés



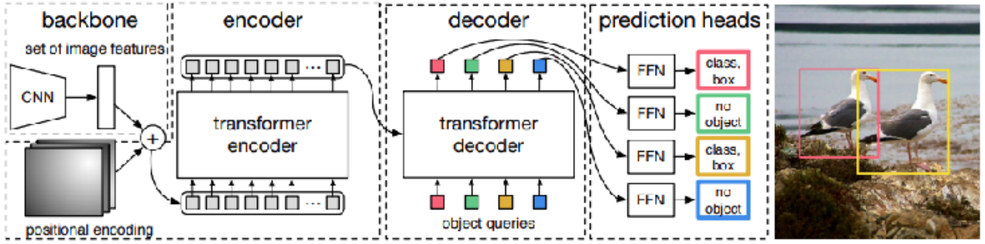

Videógenerálás




3D generálás

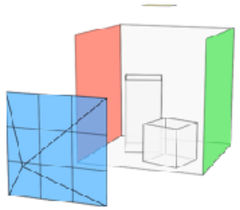
# Transformer

## Alkalmazások

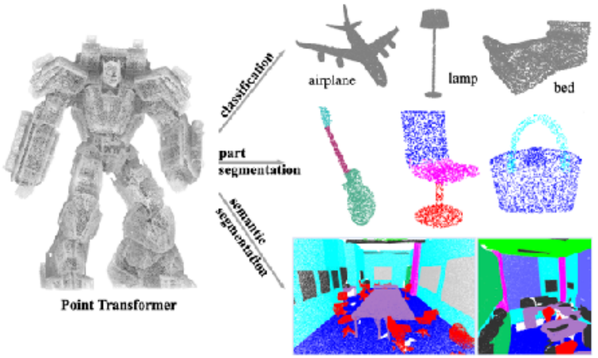




Objektum detektálás

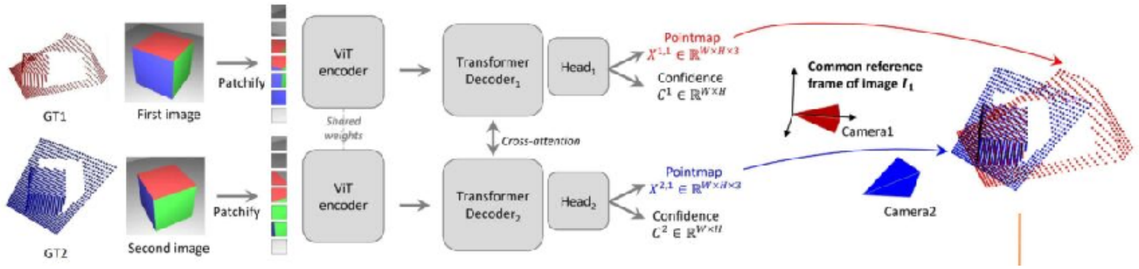
Szegmentálás (Segment Anything)



Neurális renderelés



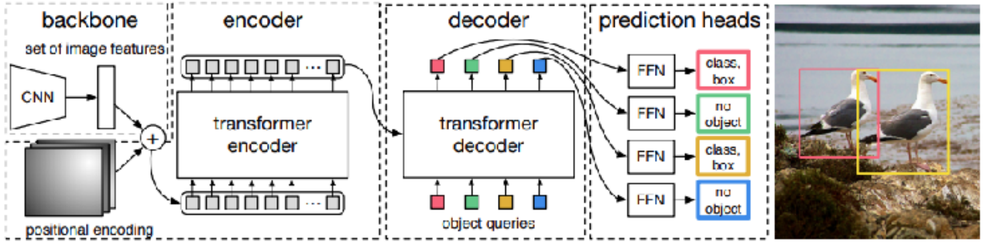

3D geometria feldolgozása




3D látás

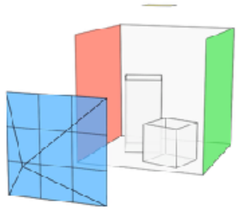
# Transformer

## Alkalmazások

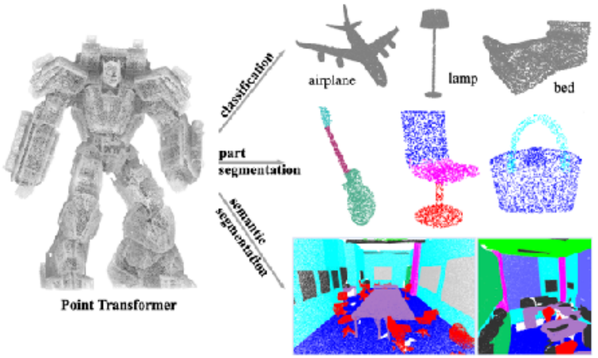




Objektum detektálás

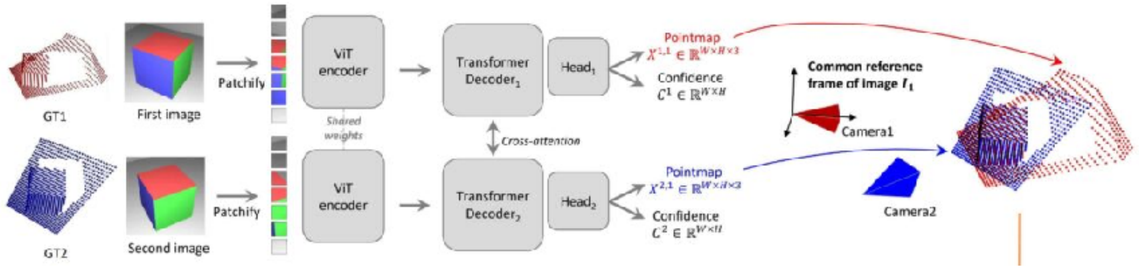
Szegmentálás (Segment Anything)



Neurális renderelés



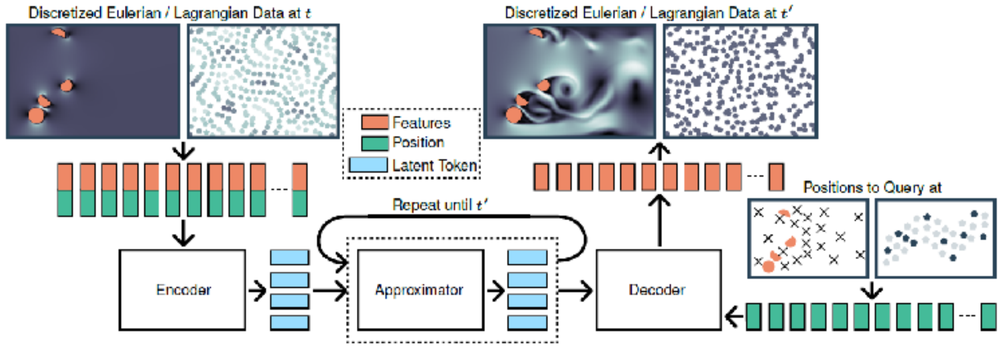

3D geometria feldolgozása



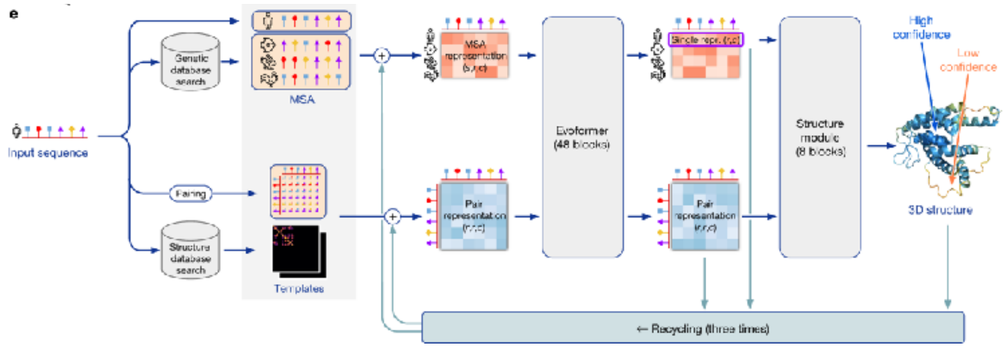

3D látás

# Transformer

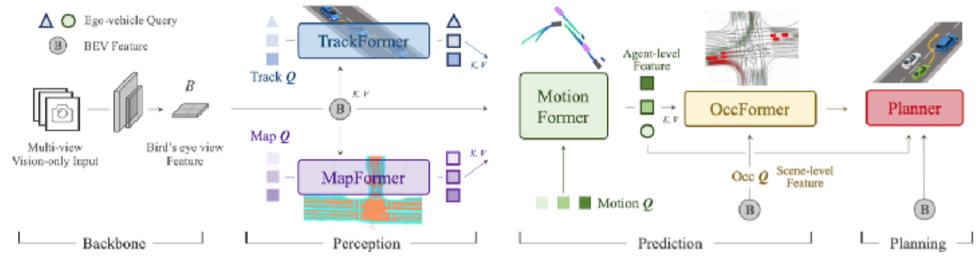
## Alkalmazások

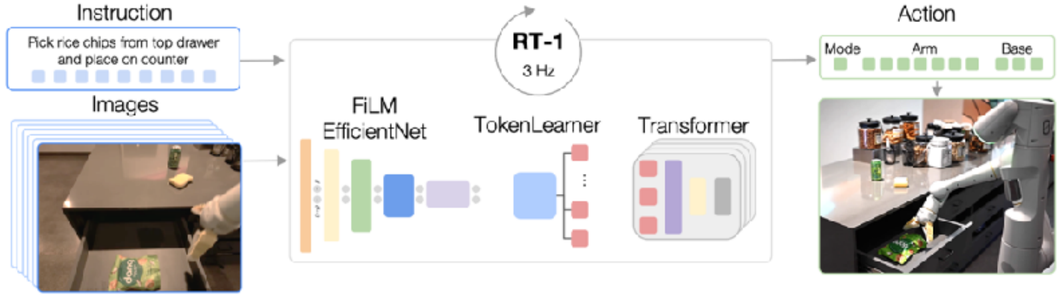

Fizikai szimuláció




AlphaFold — Kémiai Nobel-díj!



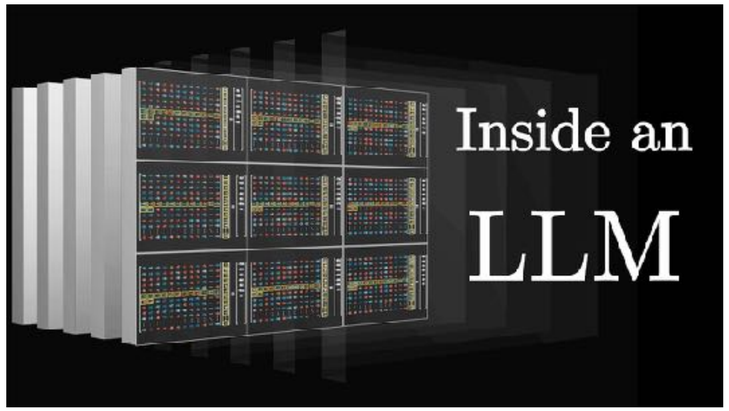

Önvezető járművek

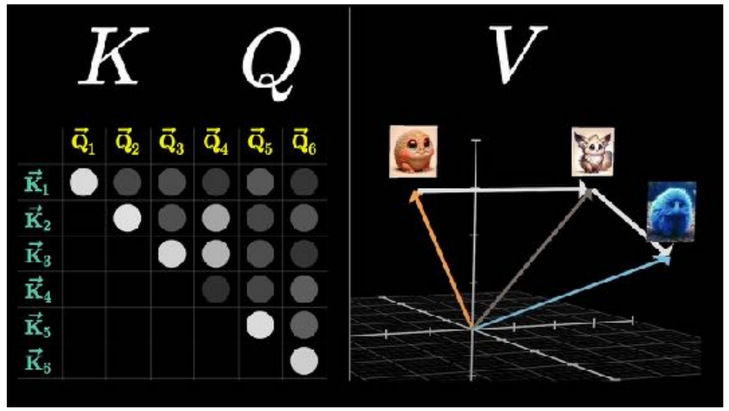


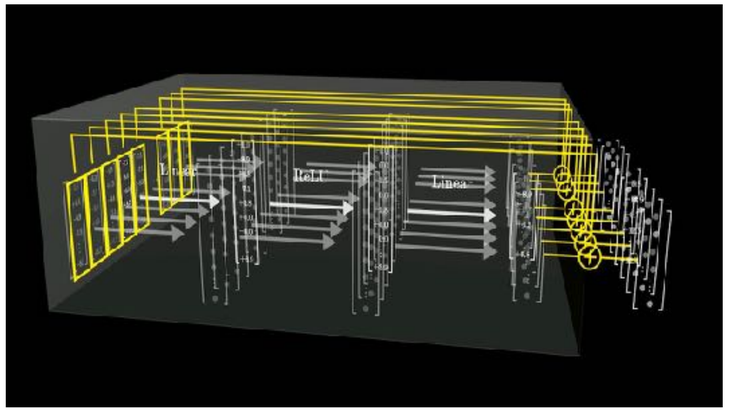

Robotika

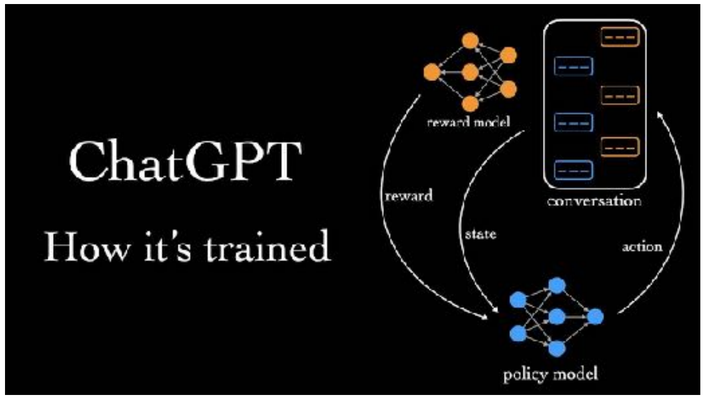
# Transformer

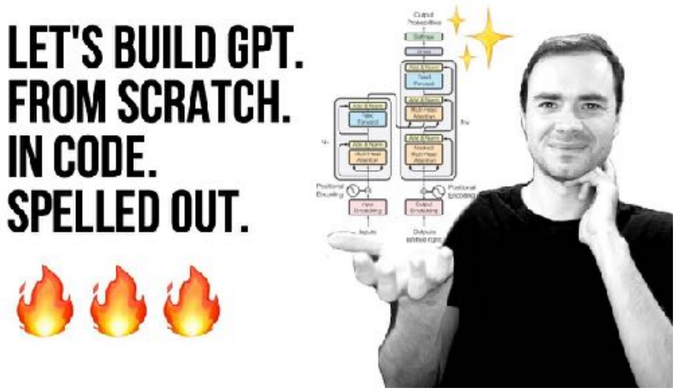
## Videó ajánló



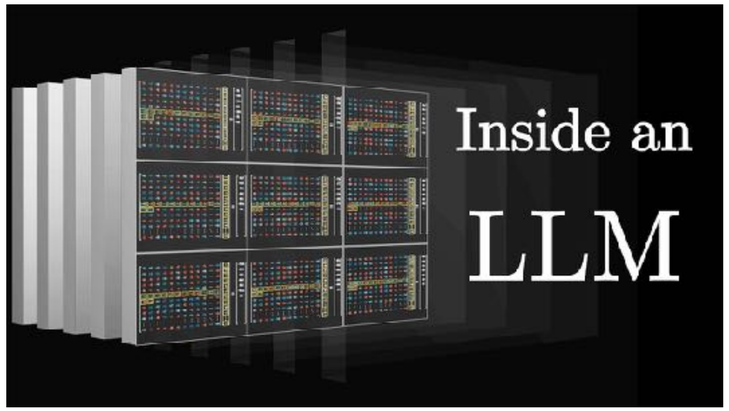



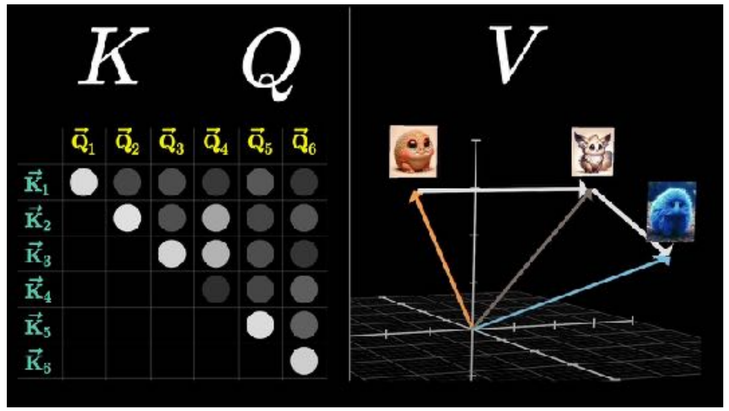


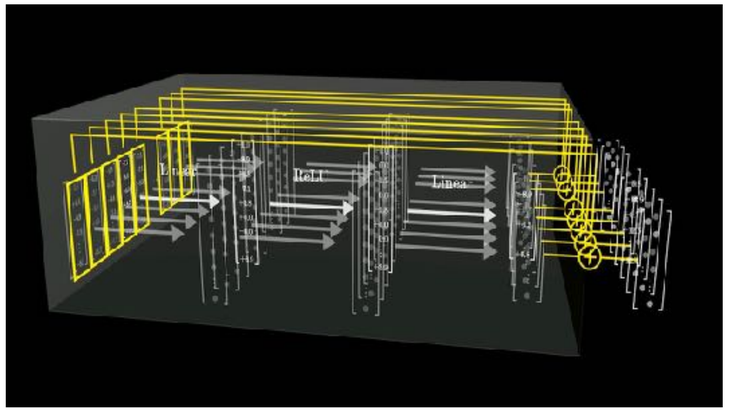



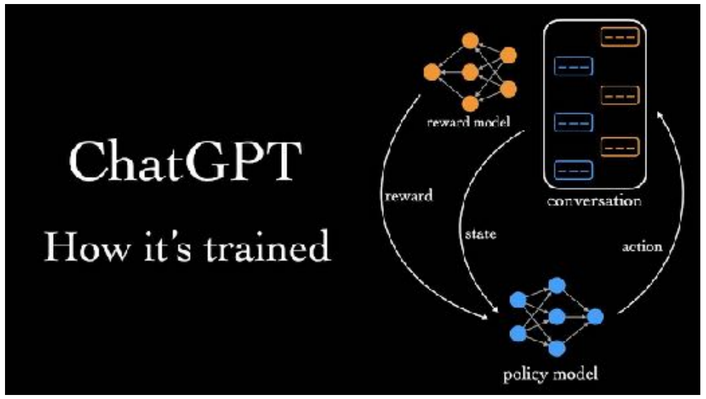
# Transformer

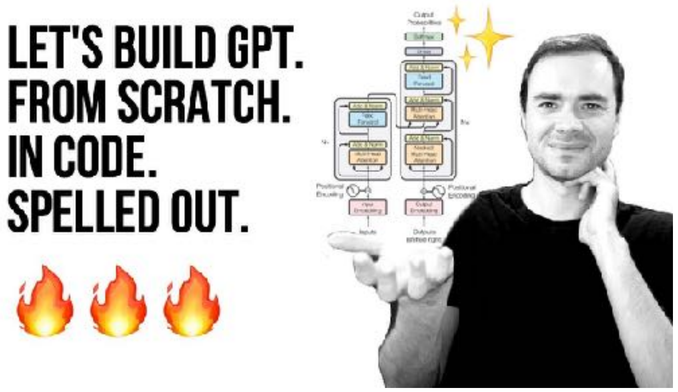
## Videó ajánló



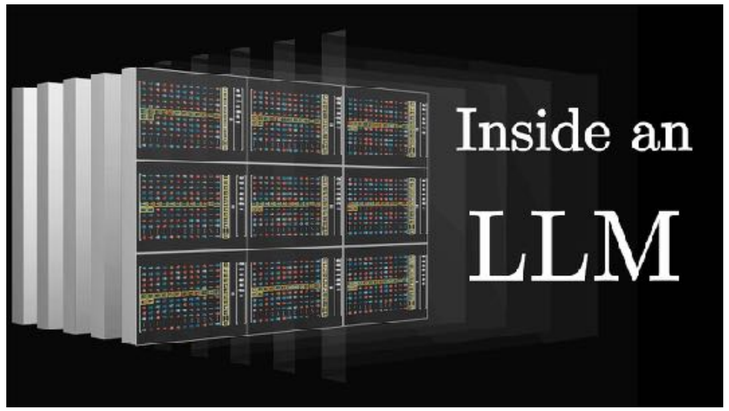



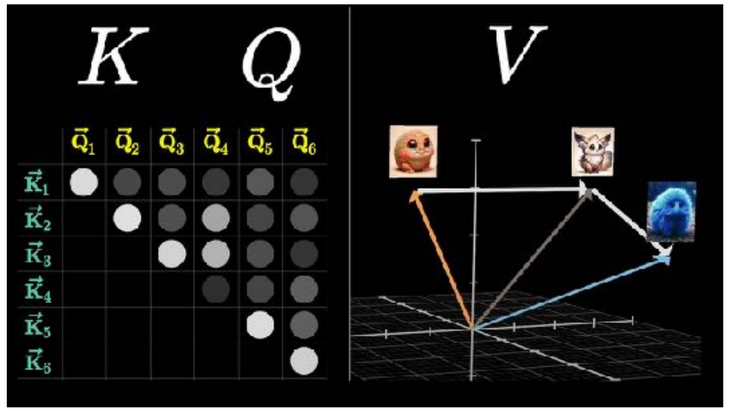


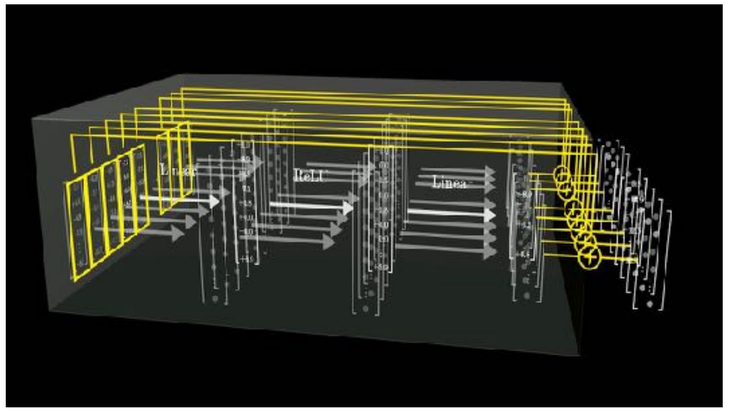



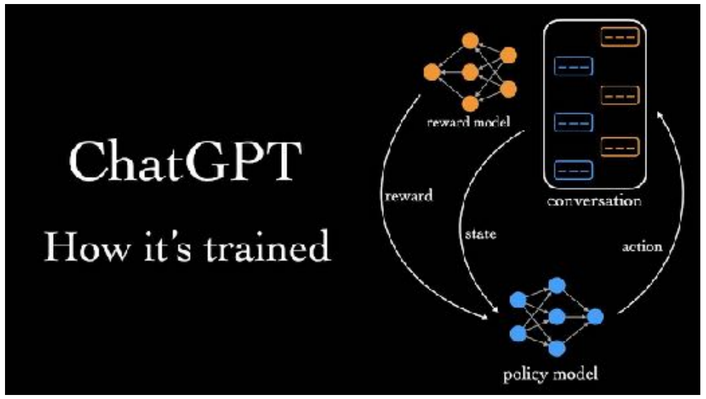
# Transformer

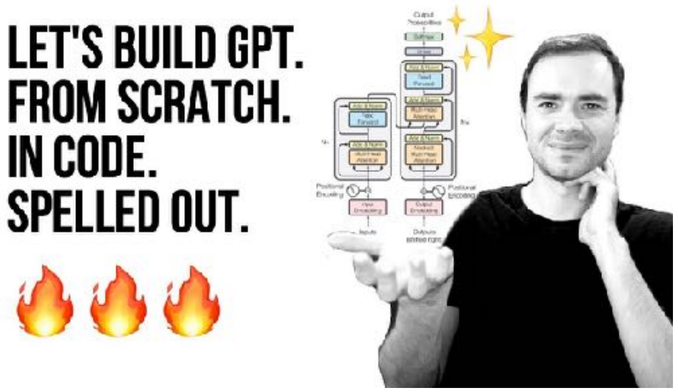
## Videó ajánló



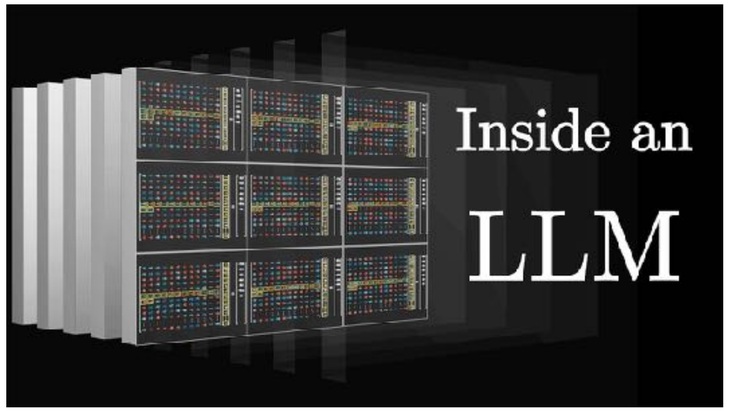



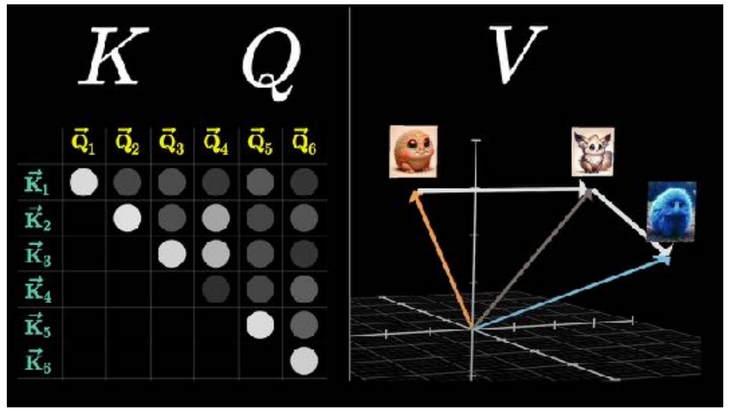


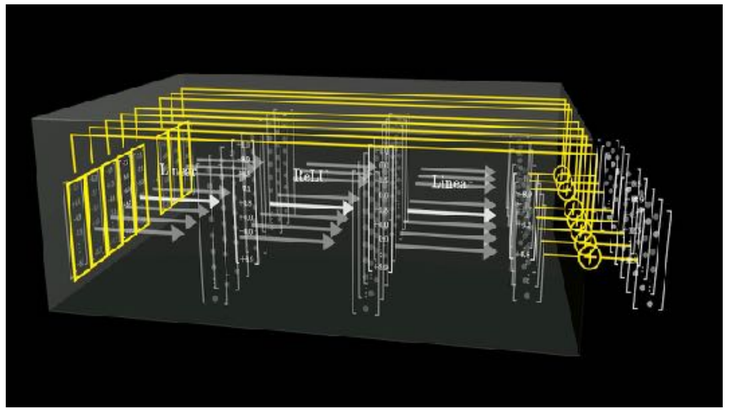



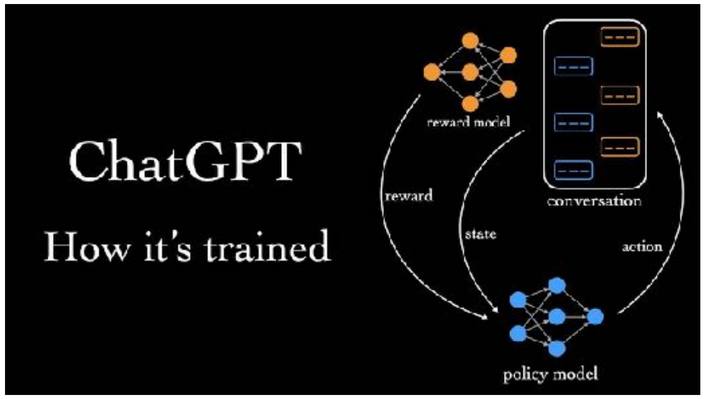
# Transformer

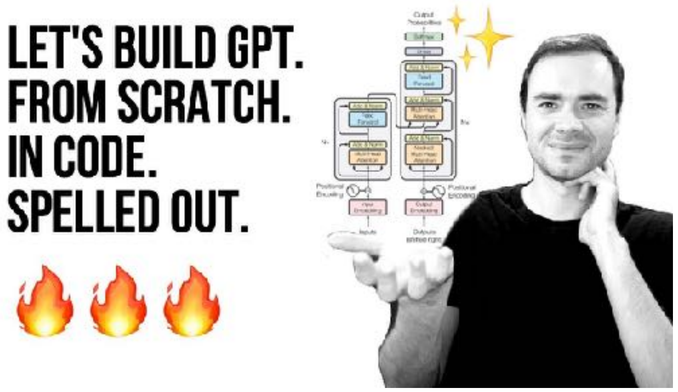
## Videó ajánló



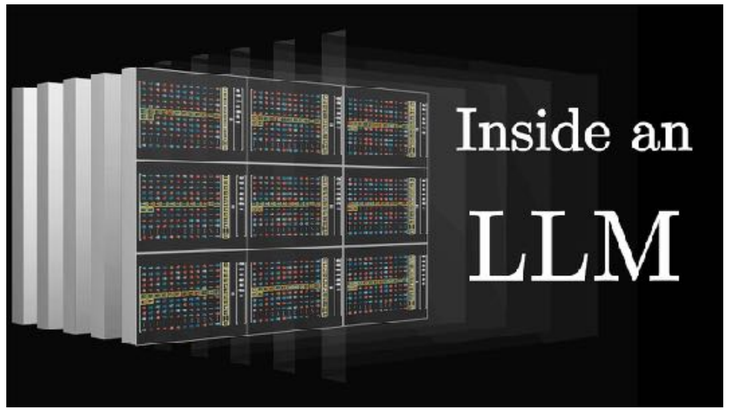



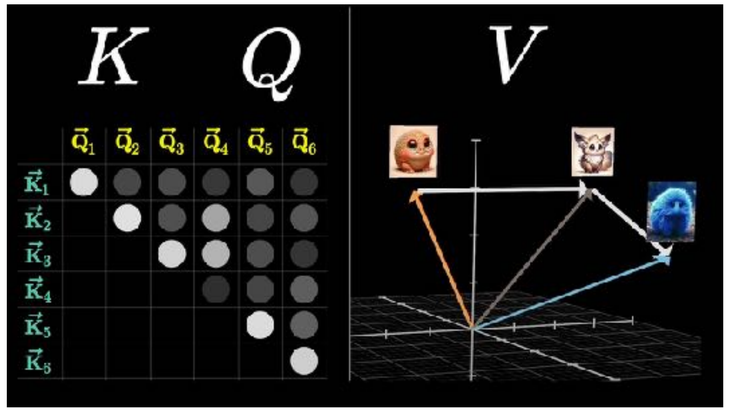


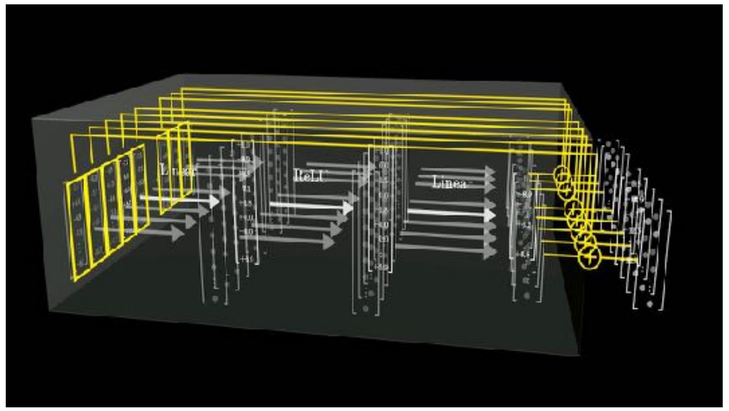



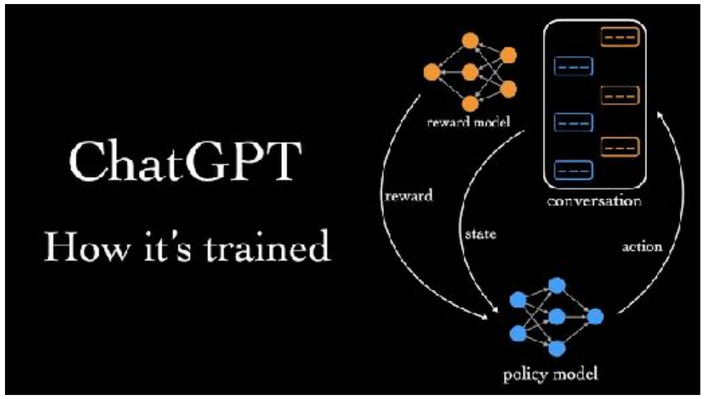
# Transformer

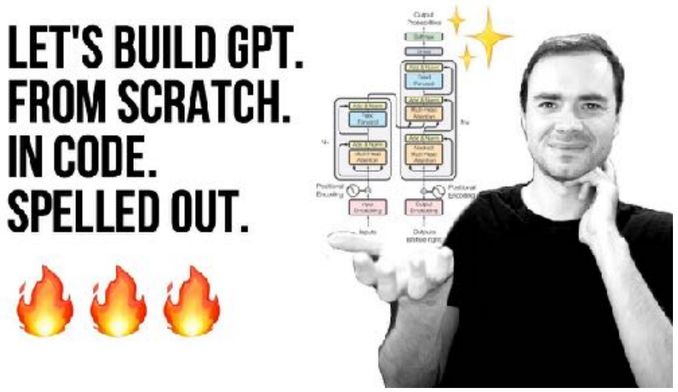
## Videó ajánló



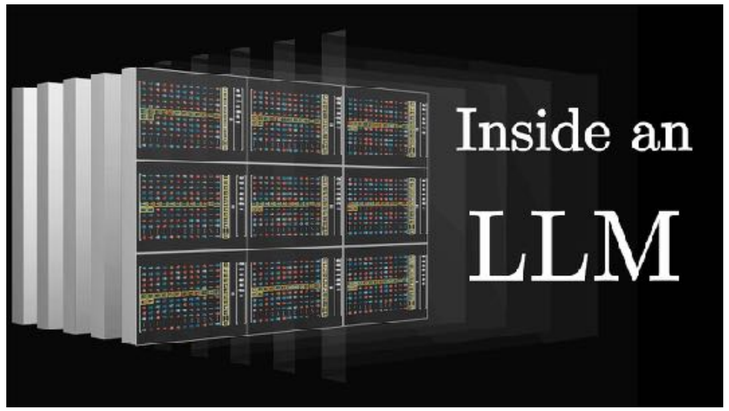



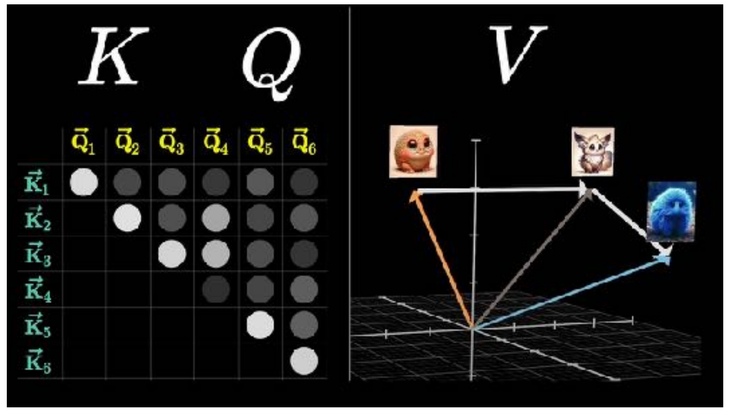


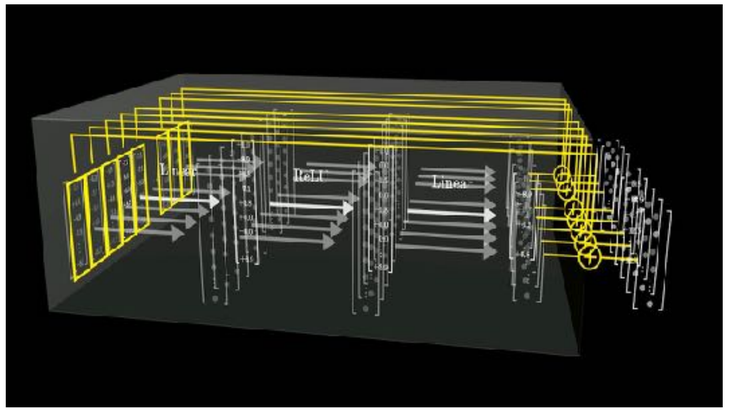



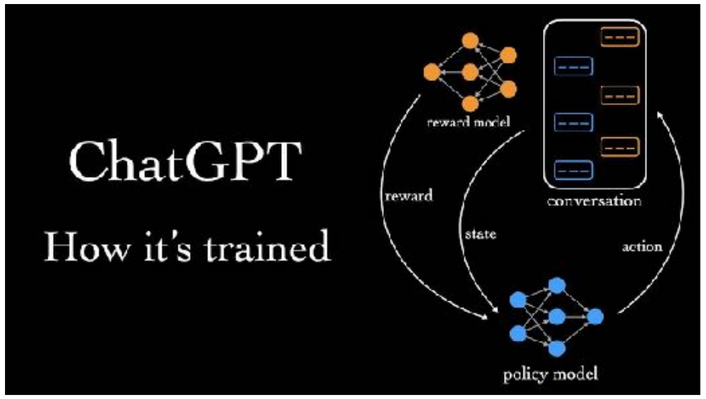
# Transformer

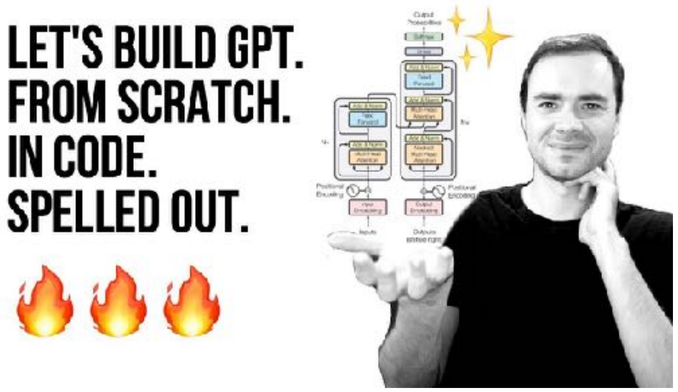
## Videó ajánló



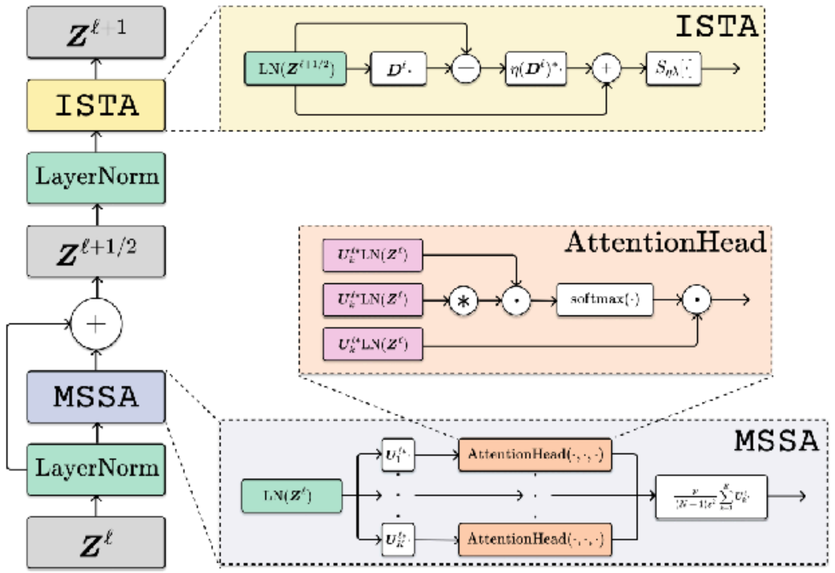



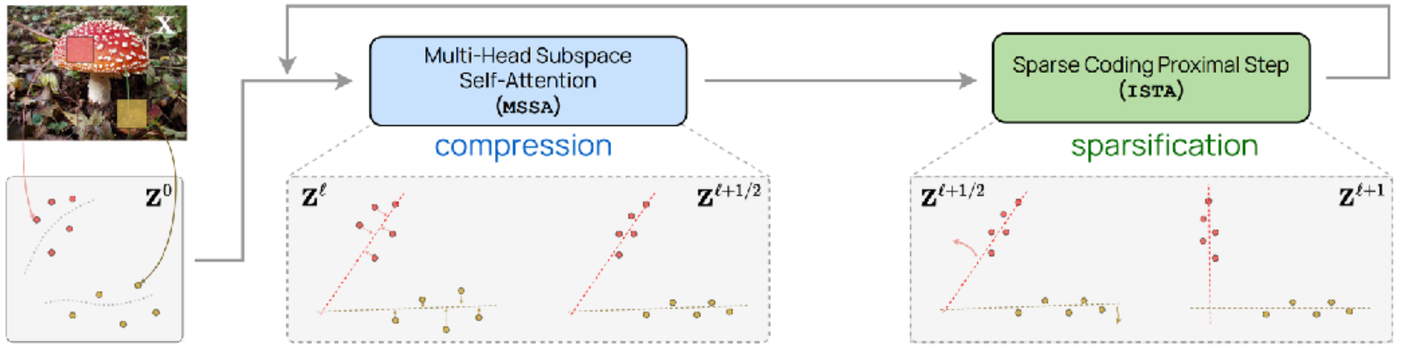


# Transformer

## Érdeklődőknek: “White-Box” Transformer




**Principles and Practice of Deep Representation Learning**

*Or A Mathematical Theory of Memory*

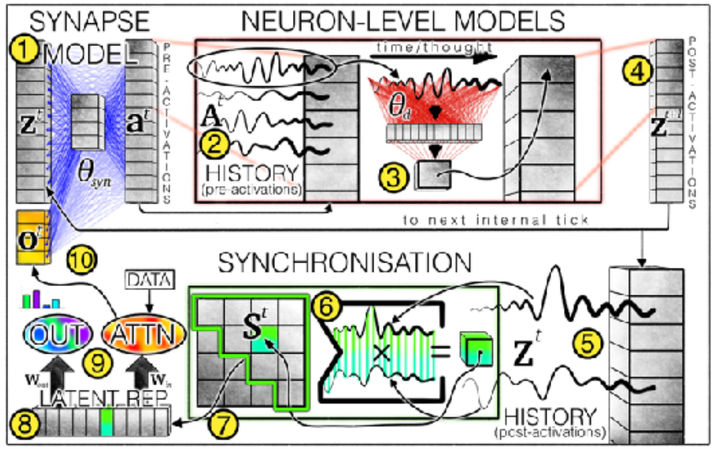
Sam Buchanan · Druv Pai · Peng Wang · Yi Ma

https://ma-lab-berkeley.github.io/deep-representation-learning-book/

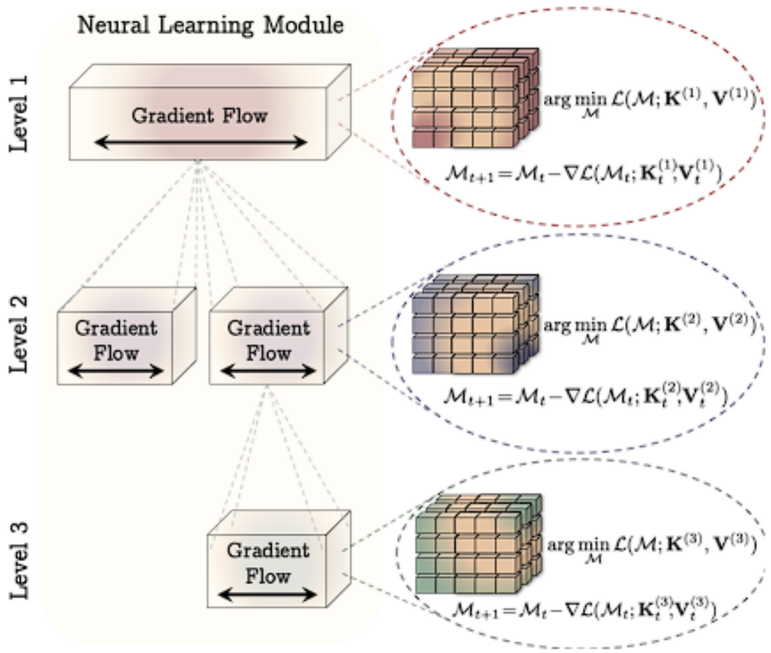
Kapcsolat a klasszikus jelfeldolgozással / kódoláselmélettel

# Transformer

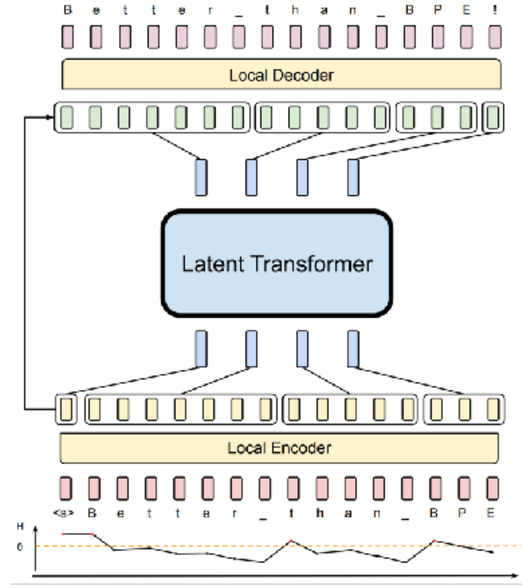
## Érdeklődőknek: “Post-Transformer” Architektúrák



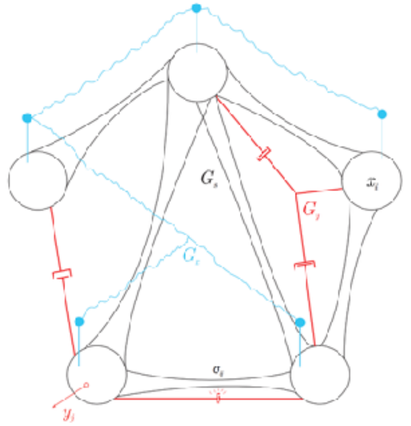

Continuous Thought Machines




Nested Learning




Byte Latent Transformer

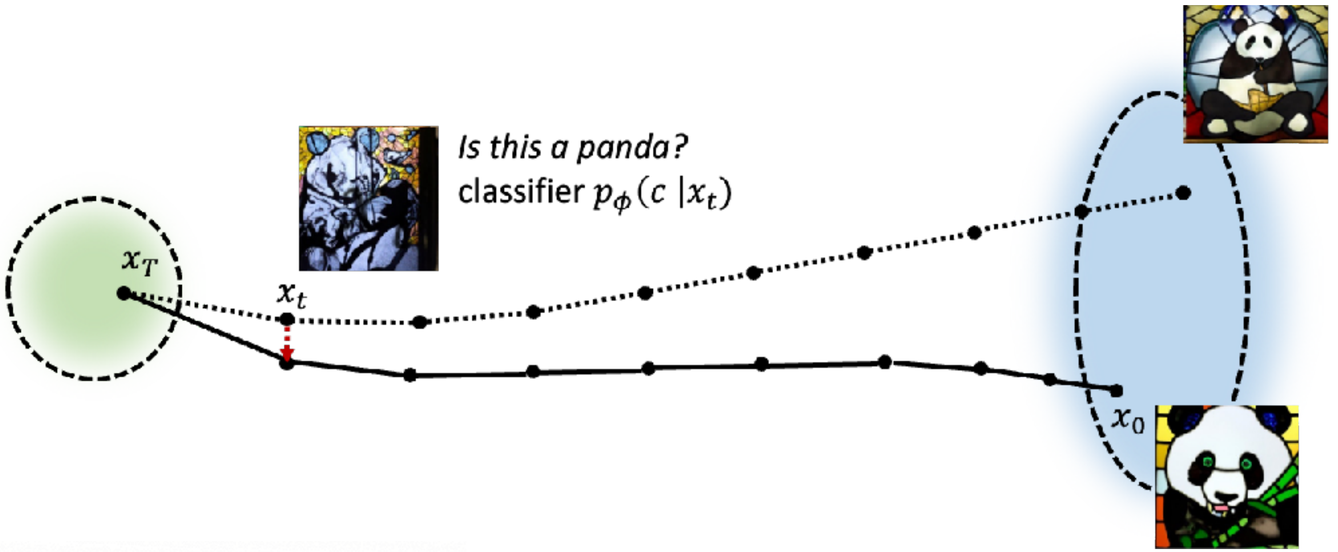


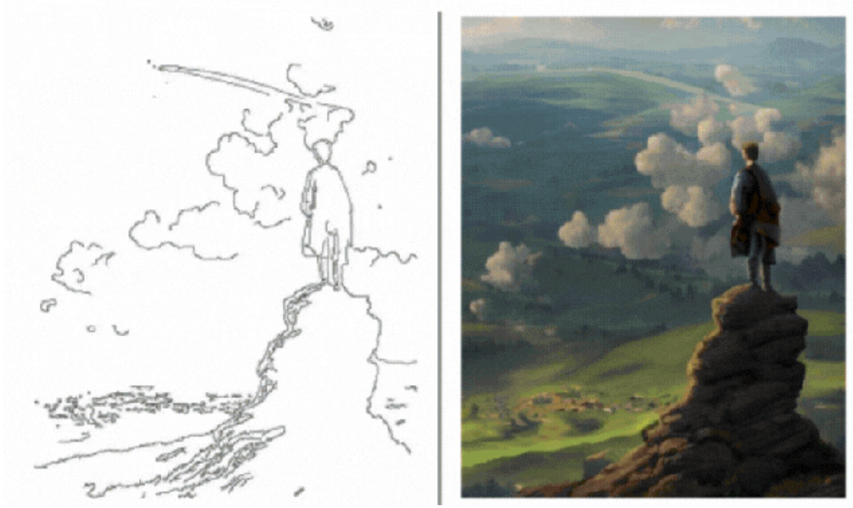
Baby Dragon Hatchling

# Következő előadás:

## Vezérelt Képgenerálás

- Vezérelt generálás
- Guidance
- Finomhangolás



# Következő előadás:

## Vezérelt Képgenerálás

- Vezérelt generálás
- Guidance
- Finomhangolás