

Mesterséges Intelligencia (MI) alapjai és alkalmazási területei az oktatásban

Tananyagot kidolgozta: WSNE, 2026



Áttekintés

- Előzmények
- Mesterséges intelligencia alapjai
- Gépi tanulás és mélytanulás
- Generatív modellek és alkalmazásaik



Mit és hogyan oktassunk? – MI (tan)anyag

- Felmerült **igény**
 - **Támogatás** az informatika tanároknak a szakképzésben az ágazati alapképzéshez kapcsolódóan, azon belül az „**Informatika és távközlés**” ágazatban az újonnan bevezetésre kerülő **Mesterséges Intelligencia (MI)** témakör tanításához
- AI EDIH projekt
 - Jelenleg futó **EU-s projekt**
 - **Egyik célja az MI oktatása** illetve az MI oktatásának támogatása

Előírt tematika

- Az adott szakképzés ágazati alapképzésének **képzési és kimeneti követelményei** (KKK)
 - „**Informatikai és távközlési alapok II.**” tantárgy keretében
 - Mesterséges intelligencia terület oktatása **10 tanóra** kiméretben
- **Mesterséges intelligencia alapjai** (2 óra)
 - Ismeri a mesterséges intelligencia (AI) fogalmát és történetét
 - Azonosítja az AI különböző alkalmazási területeit, mint például a beszédfelismerés és célzott reklámok
- **Gépi tanulás és mélytanulás** (3 óra)
 - Tudja, mit jelent a gépi tanulás és a mélytanulás, és példákat sorol fel azok alkalmazására
 - Alkalmazza a gépi tanulás alapelveit egyszerű problémák megoldására
 - Ismeri a mélytanulás alapfogalmait és a neurális hálózatok szerepét

Előírt tematika

- Az adott szakképzés ágazati alapképzésének **képzési és kimeneti követelményei** (KKK)
 - „**Informatikai és távközlési alapok II.**” tantárgy keretében
 - Mesterséges intelligencia terület oktatása **10 tanóra** kiméretben
- **Generatív modellek és alkalmazásaik** (3 óra)
 - Ismeri a generatív modellek, mint például a GAN-ok (Generative Adversarial Networks) és a transformer modellek alapelveit
 - Alkalmazza a generatív modelleket kreatív feladatok megoldására, mint például képgenerálás vagy szövegírás
- **Természetes nyelvfeldolgozás (NLP)** (2 óra)
 - Ismeri a természetes nyelvfeldolgozás alapfogalmait és alkalmazásait
 - Alkalmazza az NLP technikákat egyszerű szövegelemzési feladatok megoldására

Javaolt tematika, kiméret

- **Mesterséges intelligencia alapjai (2 óra)**
- **Gépi tanulás és mélytanulás (4 óra)**
- **Generatív modellek és alkalmazásaik (4 óra)**

(Tan)anyag felhasználása

- (Tan)anyag **formátuma, kimérete**
 - MS **PowerPoint** **fóliák** (közel 170 fólia)
 - A fóliák többségéhez **magyarázatok, instruktori megjegyzések**
- (Tan)anyag **felhasználásával** kapcsolatos **tudnivalók**
 - **A kidolgozott (tan)anyag szabadon felhasználható**
- (Tan)anyag **testreszabása**
 - A kidolgozott (tan)anyag az előírt **10 tanóránál bővebb kiméretet** biztosít
 - **Ajánlott** a (tan)anyagot az adott követelmények és az egyéni preferenciák alapján **testreszabni**



MESTERSÉGES INTELLIGENCIA ALAPJAI



Funded by the European Union

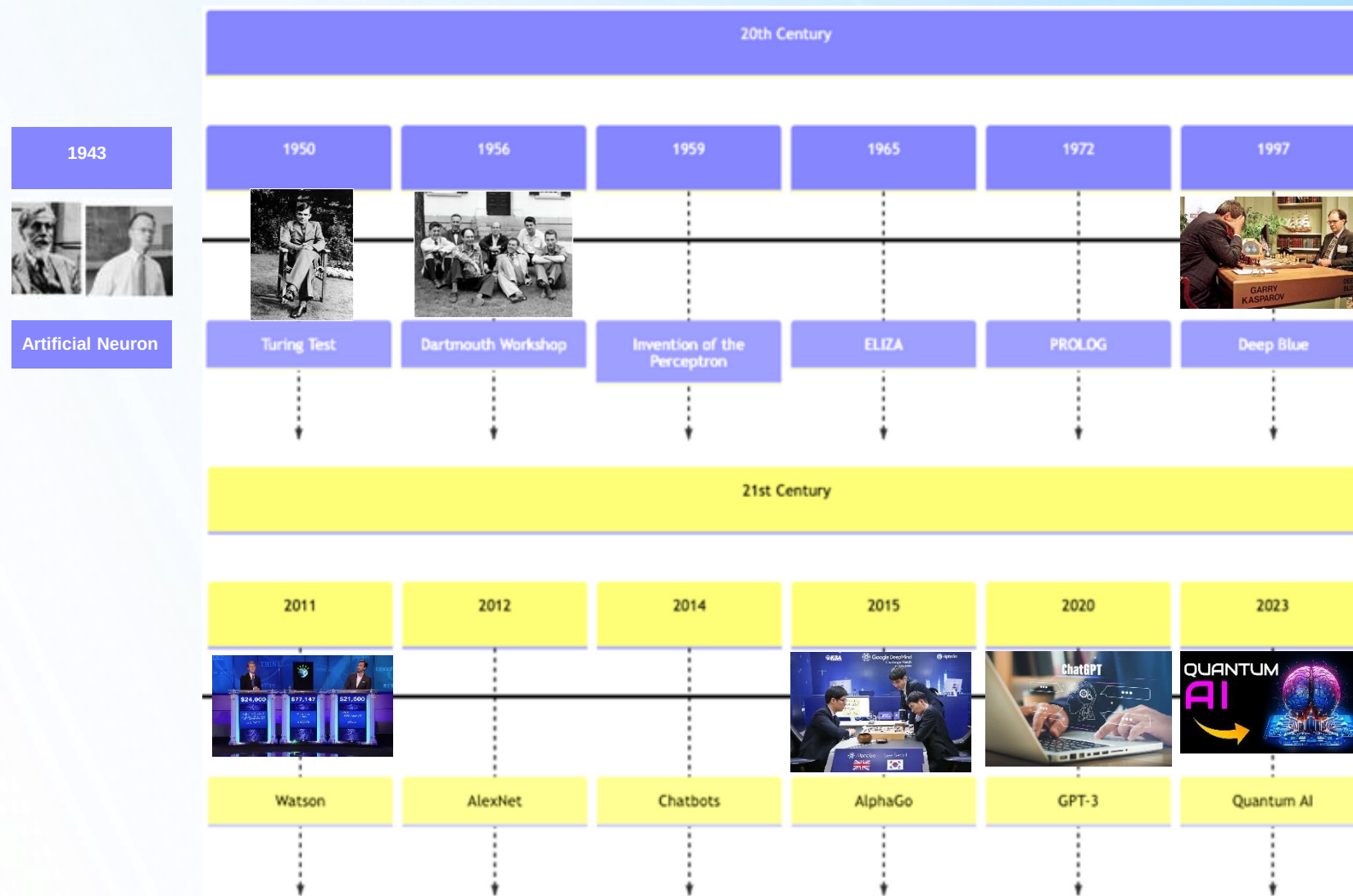
Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Áttekintés

- **Mesterséges intelligencia alapjai**
 - Bevezető, MI fogalma, rövid történeti áttekintése
 - Az MI alapú alkalmazások felépítése, működése
 - Az MI hatása, veszélyei
 - Az MI főbb alkalmazási területei
 - Néhány hasznos MI alkalmazás áttekintés
 - Gyakorlat

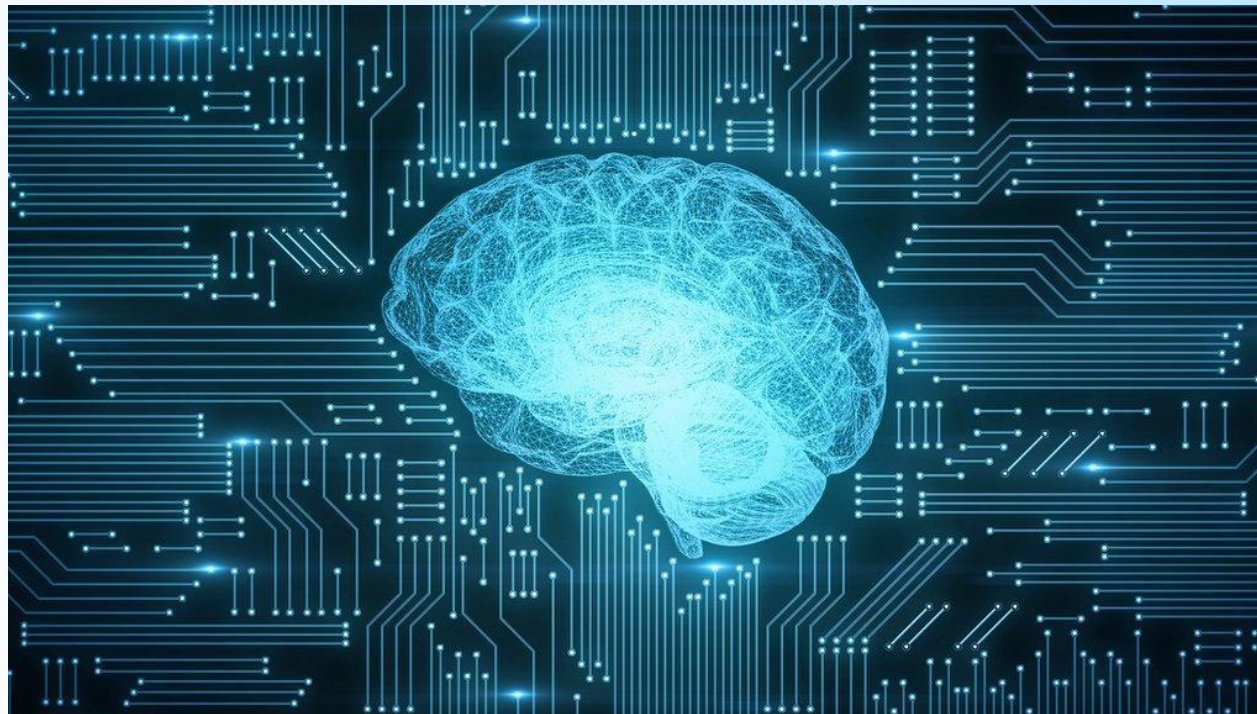


Mesterséges intelligencia fejlődésének mérföldkövei



Na de mi is az az MI?

„Olyan számítógépes program vagy rendszer, amely mesterségesen valósítja meg az ember tanulási, következtetési és észlelési képességét”



Egy kis terminológia

DS – Adattudomány

AI – Mesterséges intelligencia

ML – Gépi tanulás

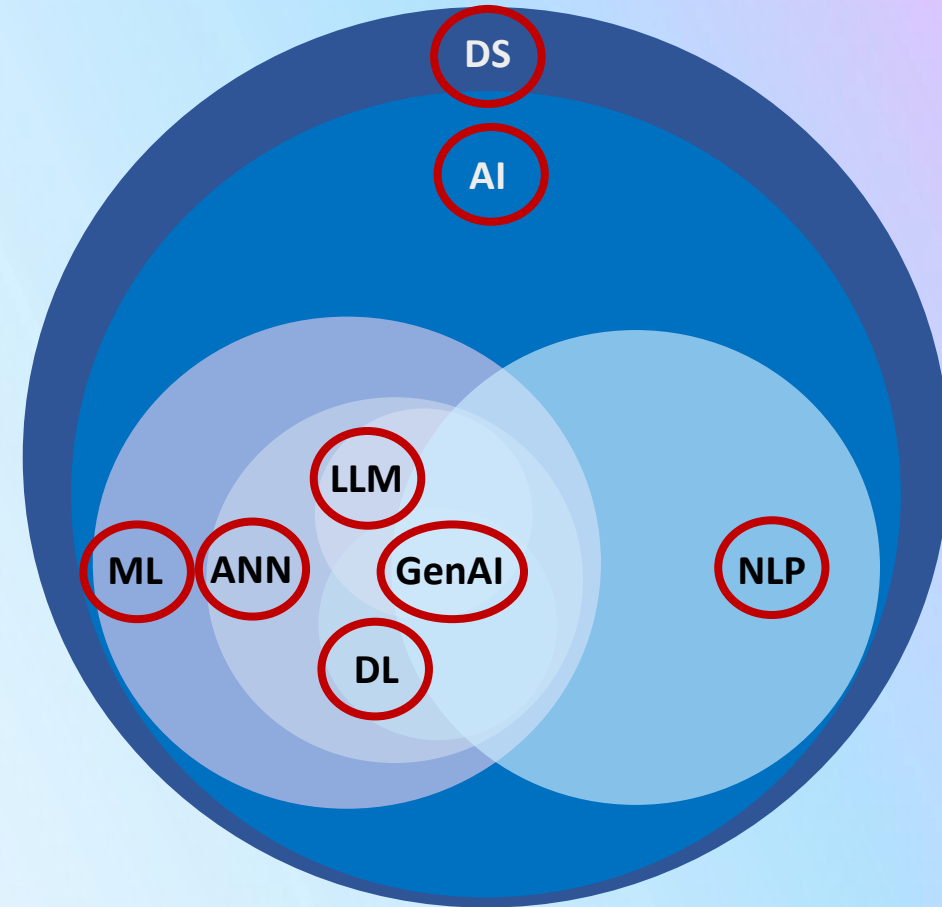
ANN – Mesterséges neurális hálózatok

LLM – Nagy nyelvi modellek

DL – Mélytanulás

NLP – Természetes nyelvfeldolgozás

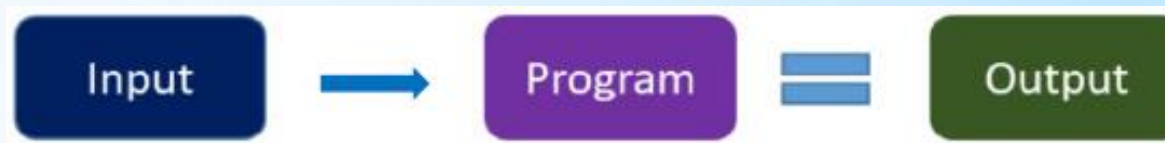
GenAI – Generatív MI



Programozás kontra Gépi tanulás

- **Programozás**

- A bemeneti adatokat egy **program dolgozza fel** és **állítja elő a kimeneti eredményt**



- **Gépi tanulás**

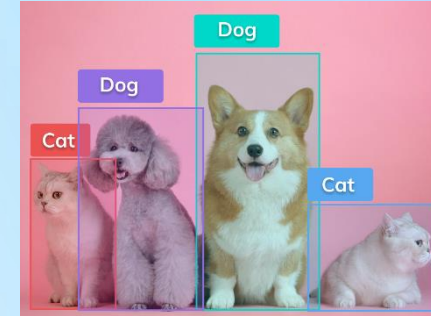
- A bemeneti és kimeneti adatok felhasználásával **a tanulási fázisban egy modellt hozunk létre** – amely hasonló funkciókat lát el, mint egy program –, míg **a használat során** tetszőleges bemeneti adatokon **a modell előállítja az eredményt**



Gépi tanulás

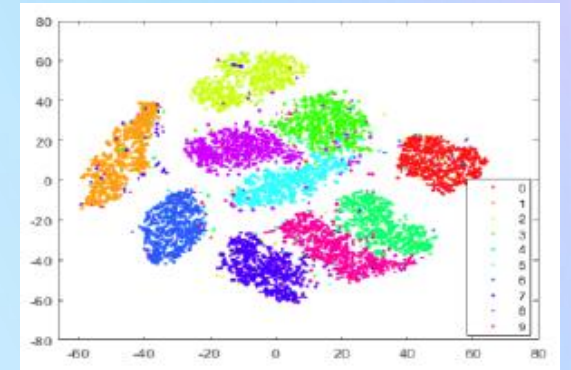
- **Felügyelt tanulás**

- **Címkézett adatokat** használ a tanuláshoz
- Tipikus **felhasználás: adat klasszifikáció**



- **Felügyelet nélküli tanulás**

- **Címkézetlen adatokat** használ a tanuláshoz
- Tipikus **felhasználás: adatok klaszterezése**



- **Megerősítő tanulás**

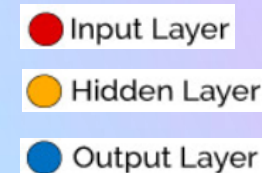
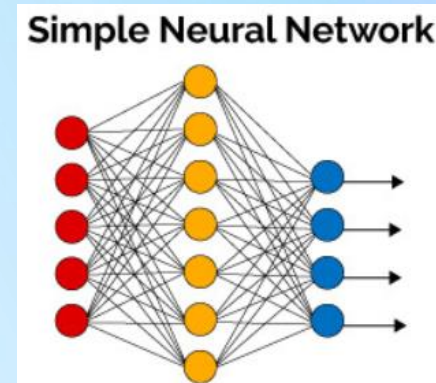
- Bemeneti adatok és **jutalom** használata a tanuláshoz
- Tipikus **felhasználás: robotok tanítása, vezérlése**



Neurális hálózat, mélytanulás

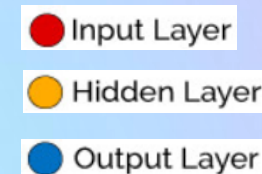
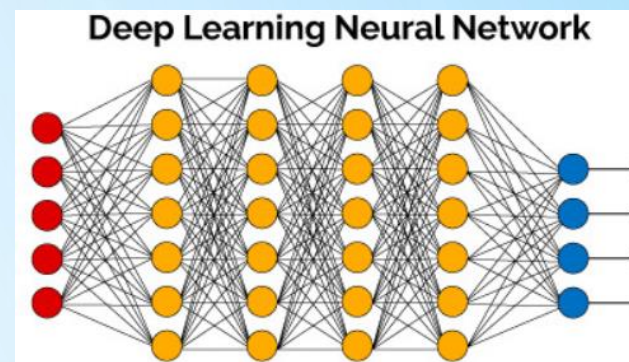
- Mesterséges **neurális hálózat**

- A biológiai **idegrendszer** (pl. az emberi agy) információfeldolgozásának **működését próbálja utánozni** idegsejtek (neuronok) hálózatának használatával



- Mélytanulás

- Többrétegű mesterséges neurális hálózatokat **használ**, amelyekben **visszacsatolás** is előfordulhat az egyes rétegek között

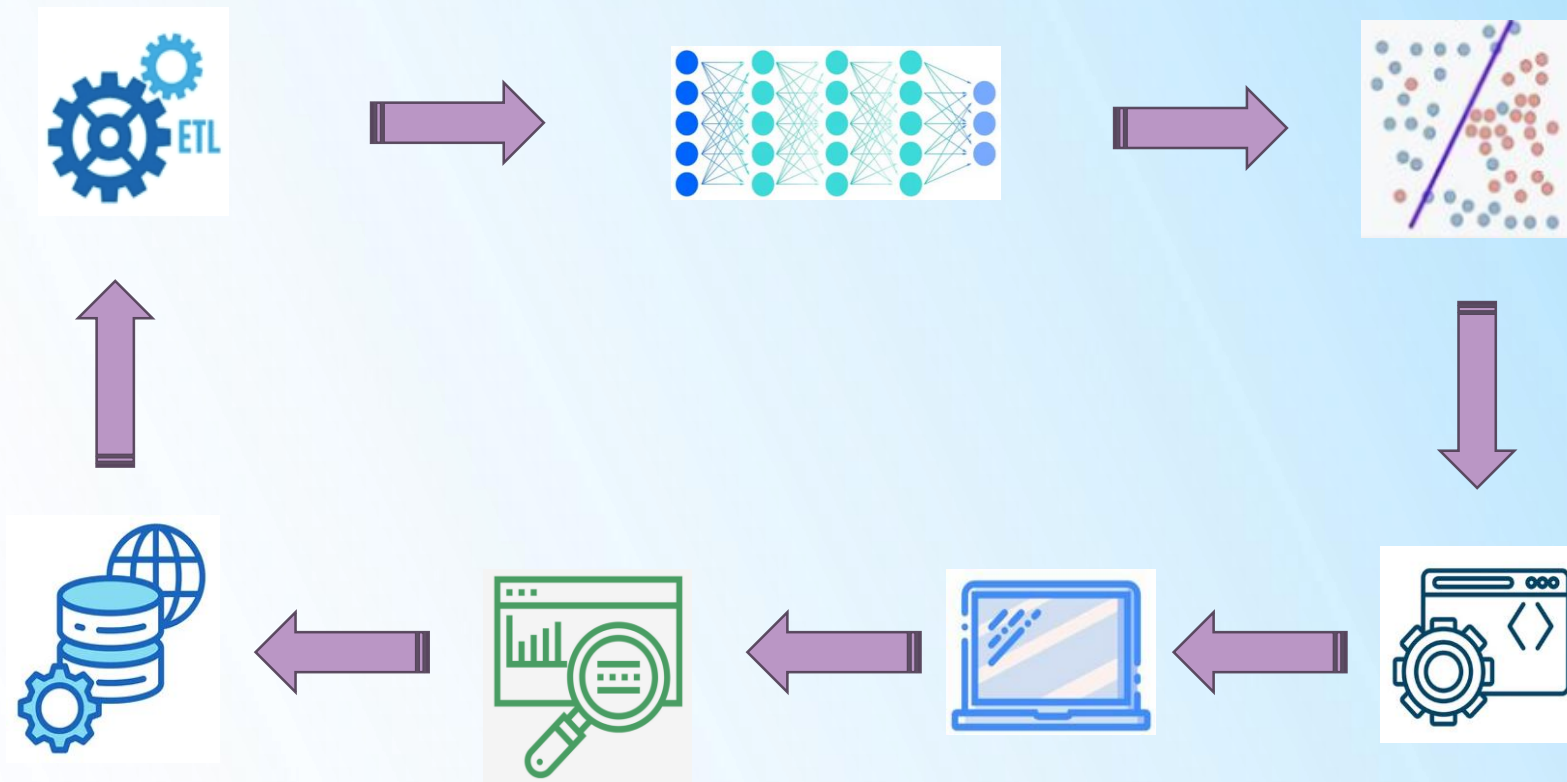


MI alapú rendszerek felépítése

- MI alapú rendszerek **szerkezete**
 - **Több**, egymásra épülő **komponensből állnak**, amelyek együtt alakítják az adatokat értékes kimenetűvé
- Főbb **komponensek**
 - Adatforrások 
 - Adatfeldolgozás 
 - Modell 
 - Tanítás és értékelés 
 - Futtató környezet 
 - Felhasználói felület 
 - Monitorozás, visszacsatolás 

MI alapú rendszerek működése

- Működés folyamata



Erőforrásigény

- A „chatbot” jellegű eszközök **erőforrásigénye már jelenleg is hatalmas, szinte fenntarthatatlan**
- Például **ChatGPT-5** (2026. januári állapot szerint)
 - Egy közepes hosszúságú **kérdés** esetén 2 és 45 watt közötti, átlagosan **19 watt energiafogyasztás**
 - **Napi kb. 2,5 milliárd lekérdezés** 810 millió felhasználó által
 - **Naponta körülbelül 47,2 gigawattóra energiaigény**
- Összehasonlításképpen
 - Egy **átlagos amerikai háztartás naponta körülbelül 29 kilowattóra energiát használ fel**
 - A **ChatGPT napi energia fogyasztása kb. 1,64 millió háztartás** áramigényét fedezné
 - A **ChatGPT naponta annyi energiát fogyaszt**, mint egy akkora ország, mint pl. **Horvátország**

Adatközpont Dél-Koreában



2025. március. 01. 08:03 · TECH

13 330 844 905 000 forintból itt épül a világ
legnagyobb adatközpontja

hvg.hu

2025.03.01.

MI veszélyei

Az MI-t nem csak „jó” dolgokra lehet használni...

Kiberbűnözés...



*Gyorsan lehet **akár szakértelem nélkül is** kiberbűnözni...*

MI alapú alkalmazások támadása, sérülékenységeinek kihasználása



Folyamatos, felgyorsult versenyfutás a JÓ és a SÖTÉT oldal között

MI veszélyei

Mi a valódi, és mi a mesterségesen generált?

Pl. „deepfake” videók, tartalmak...



<https://www.youtube.com/watch?v=cQ54GDm1eL0>

MI veszélyei

Hogyan tudjuk megkülönböztetni a valódit a mesterségesen generálttól?

Egyre nehezebben...



*Erre megoldást találni az egyik legnagyobb **kihívás**...*



Azt kell majd bizonyítani, hogy mi az EREDETI, és nem azt, hogy mi a HAMIS!!!

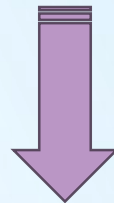
MI veszélyei

Megbízunk-e / megbízhatunk-e az MI-ben?

Attól függ...



*... hogy **értjük-e**, mit csinál, **tudjuk-e ellenőrizni**...*



*XAI, avagy a magyarázható
mesterséges intelligencia*

MI jelen / jövő

Robotok



https://www.youtube.com/watch?v=-9EM5_VFlt8



<https://www.youtube.com/watch?v=UNorxwIZIFk>

MI jelen / jövő

Robotok



<https://www.youtube.com/watch?v=X2UxtKLZnN>

0

MI jelen / jövő

Robotok



**Magyarországon is kapható a világ első
humanoid robotja, bárki megveheti
tizenkétfmilió forintért**

index

2025.03.06.



MI jelen / jövő

Önvezető autók



<https://www.youtube.com/watch?v=Qfj4urMF8CU>

MI jelen / jövő

MI ügynökök



MI jelen / jövő

MI ügynökök

TECHTUD

Jönnek az OpenAI fejlett MI-ügynökei, az egyikért havi 7 millió forintot is elkérhetnek

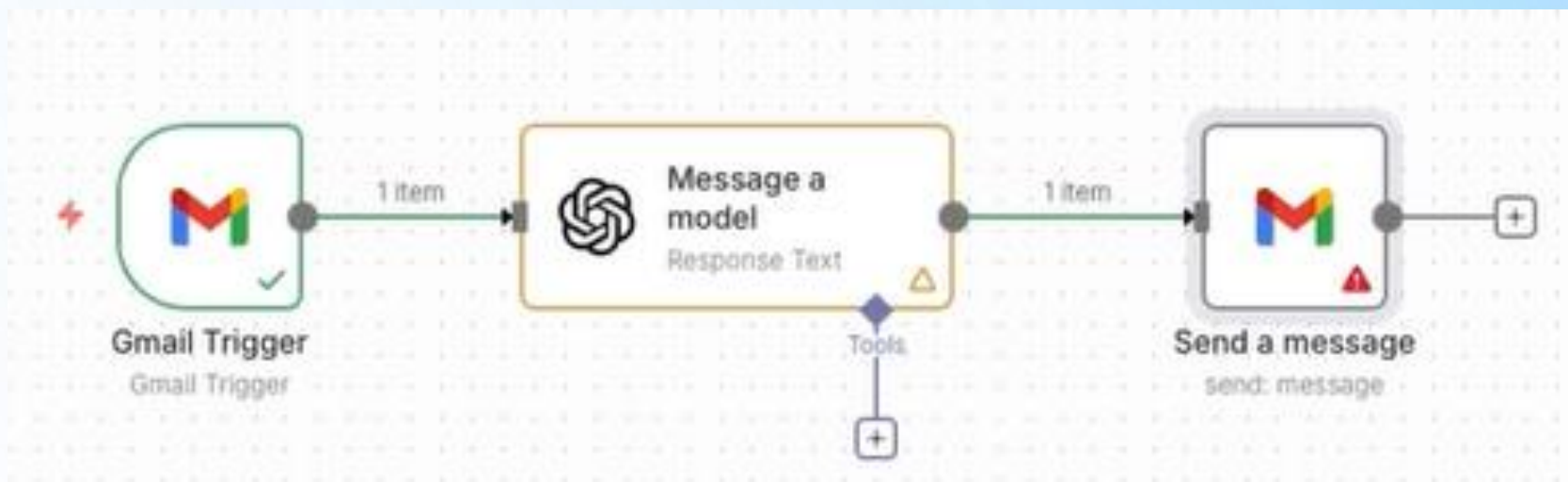
Az állítólagos MI-ügynököt PhD szintű kutatásokhoz lehet majd használni. Ezen túl többek közt egy szoftverfejlesztő ügynök is érkezhet fele ennyiért.

telex

2025.03.06.

MI jelen / jövő

MI ügynökök mostanság



<https://n8n.io>

MI evolúció

AI, mint kiterjesztett intelligencia (Augmented Intelligence):

az MI segít a feladatok megoldásában (asszisztál)
JELEN

AI, mint MI (Artificial Intelligence):

az MI helyettünk old meg feladatokat (nem okosabb nálunk)
JELEN / KÖZELJÖVŐ

AI, mint általános MI (General Artificial Intelligence):

az MI olyan feladatokat old meg feladatokat, melyeket mi nem tudunk megoldani (okosabb nálunk)
KÖZELJÖVŐ / JÖVŐ???

Néhány tanulást segítő MI alkalmazás

- **ChatGPT**, Gemini, Claude, stb.
 - Általános **csevegő robot**
 - Sok mindenre használható, pl. **tanulásban is tud segíteni**



- **Google NotebookLM**
 - Jegyzetek, **dokumentumok összefoglalása**
 - <https://notebooklm.google.com>



- **YoloAI**
 - **Tanulási asszisztens**, magyar fejlesztés
 - <https://app.yoloedu.ai>



Áttekintés

- **Mesterséges intelligencia alapjai**
 - Bevezető, MI fogalma, rövid történeti áttekintése
 - Az MI alapú alkalmazások felépítése, működése
 - Az MI hatása, veszélyei
 - Az MI főbb alkalmazási területei
 - Néhány hasznos MI alkalmazás áttekintése
 - **Gyakorlat**

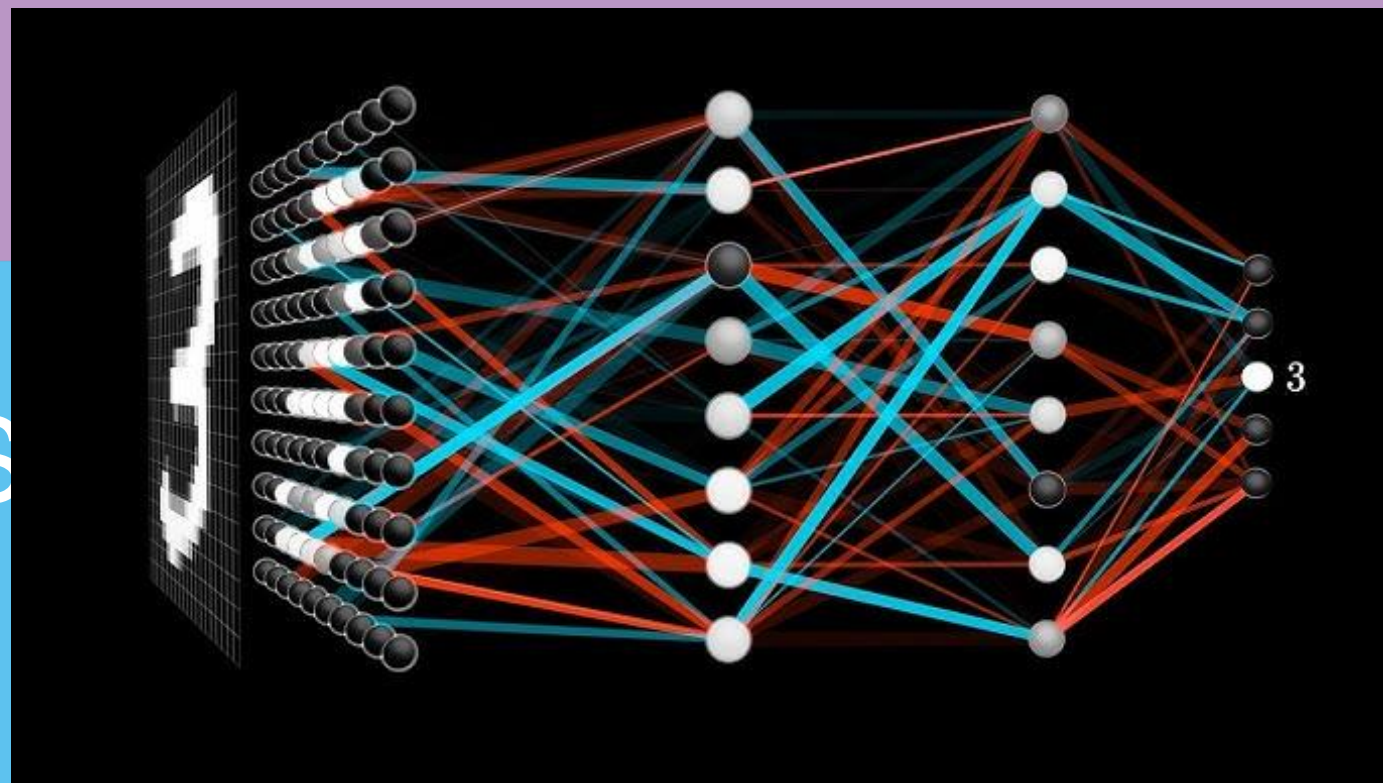


Gyakorlat

- **ChatGPT** (vagy Gemini, Claude, stb.)
 - **Néhány egyszerű feladat**, pl. a következő promptok kipróbálása
 - „Mi vagy te, és meddig tudsz fejlődni?”
 - „Mire lehet téged használni?”
- **Google NotebookLM**
 - **Néhány egyszerű feladat** elvégzése, pl.:
 - Betöltünk egy hosszabb szöveget, és megkérjük, hogy foglalja össze
 - Feltöltünk néhány fóliát, és generáltatunk hozzá magyarázó videót
- **YoloAI**
 - A **YoloAI kipróbálása**, használatának megismerése, pl.:
 - Tetszőleges, akár feltöltött tananyaghoz ellenőrző teszt generál



GÉPI TANULÁS ÉS MÉLYTANULÁS

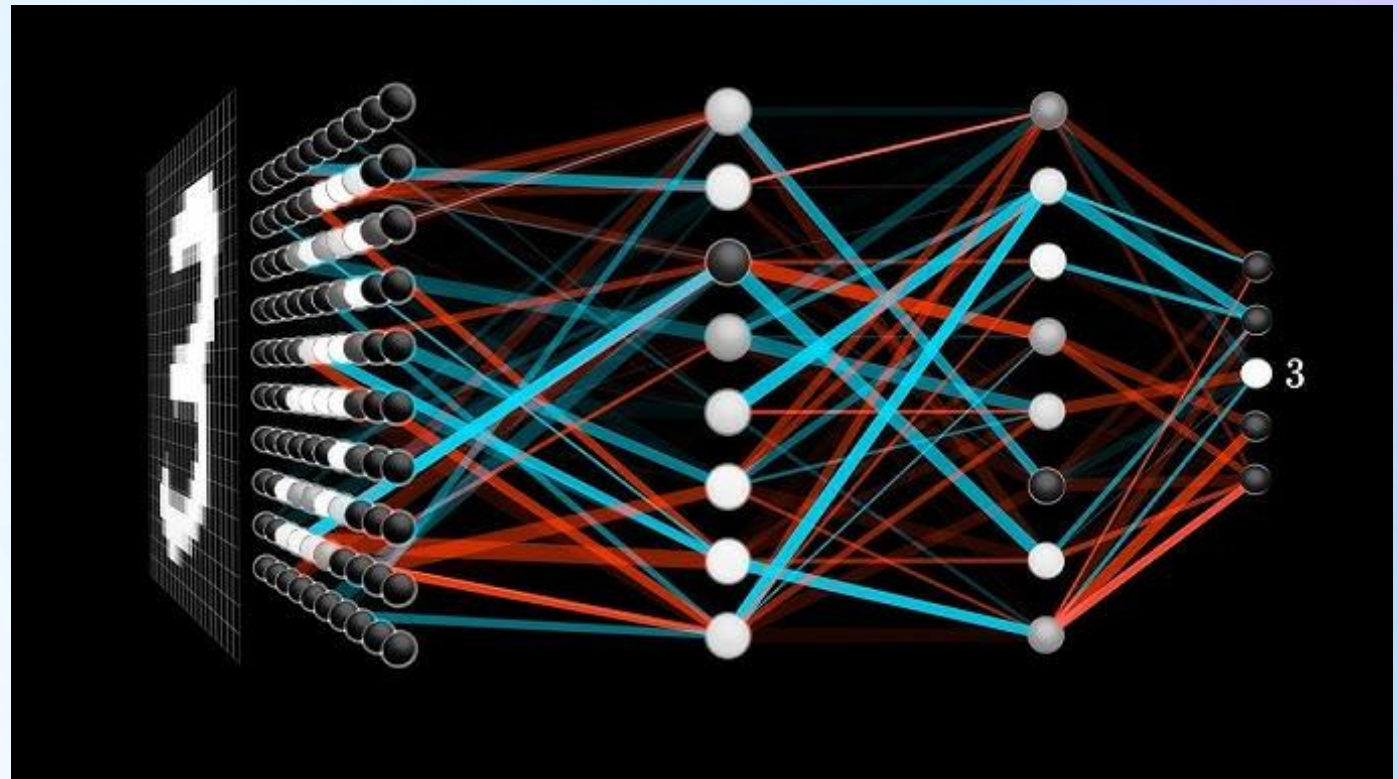


Funded by the European Union

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Áttekintés

- **Gépi tanulás és mélytanulás**
 - Alapok
 - Tanuló algoritmusok
 - Neurális hálózat tanítása
 - Gyakorlat



Különböző problémák – Megoldások

Életünk során **számos problémába futunk**, amit valamilyen módon **meg kell oldani**

- Programok szervezése, időpontok keresése
- Hogyan tudok eljutni Budapestről Kecskemétre?
- Mennyit kell havonta félretenni, ha nyáron nyaralni szeretnénk?

Számos irányból meg lehet közelíteni ezeket a problémákat

- **Milyen végeredményre** számítunk?
 - Igen-nem
 - Szám
 - Kép
 - Útvonal
- **Milyen információkkal** rendelkezünk?
- **Hogyan tudjuk megválaszolni** a kérdést?



Problémák csoportosítása

Vannak **könnyen kiszámítható** problémák

- Pl. megtakarítás: mennyi idő alatt mennyi pénzt kell összegyűjteni

Léteznek **bonyolultabb, de** hasonlóan **kiszámítható** problémák

- Mi a legrövidebb útvonal két távoli város között?
- Számos különböző úton el lehet jutni, különböző módokon
- De létezik egy bizonyítottan működő algoritmus, ami könnyen kiszámolja a pontos értéket

Vannak viszont **könnyen megválaszolható, de nehezen megmagyarázható** problémák

- Hány kutya van a képen?
- Nyilván 4, de miért?
- Mitől kutya egy kutya?
- Miben különbözik egy kutya egy macskától?



Tanuló algoritmus

Az ilyen **könnyen megválaszolható, de nehezen megmagyarázható problémák** számos helyen előjönnek

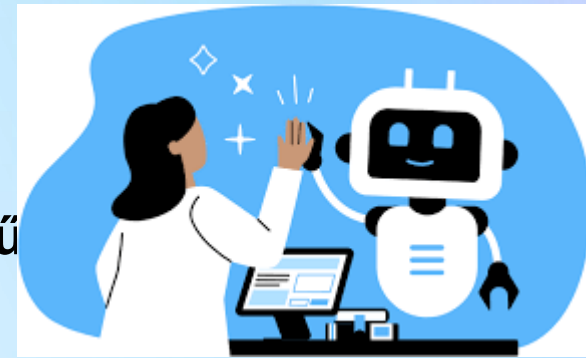
- Járművek vezetése („majd tapasztalattal jön”)
- Objektumok számolása kép alapján (mitől lesz egy dolog valami?)
- Művészeti alkotások készítése
- Hogyan készítsünk el egy jó alkalmazást?
 - Irányelvek vannak, de egy jól követhető algoritmus nincs

Mivel könnyen megválaszolható, könnyű **adathalmazt** előállítani

- Adathalmaz – Milyen bemenetre milyen választ várunk el

Szükséges

- **Algoritmus**, mely a **végeredményből** tud visszakövetkeztetni az **elvárt mű**



Az informatika alapjai

Az elektronikai eszközeink rengeteg **egyszerű logikai egységből (kapuból)** épül fel

- Digitális jelek: **csak 0 vagy 1** lehet az értéke
- Általában több bemenet, egy kimenet

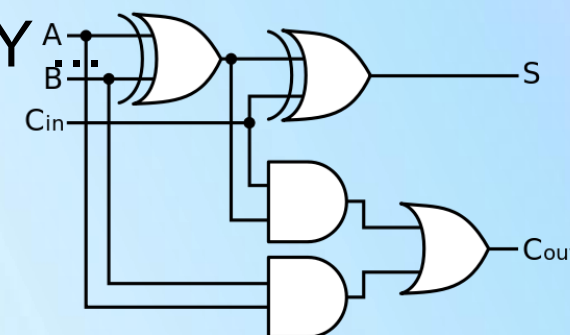
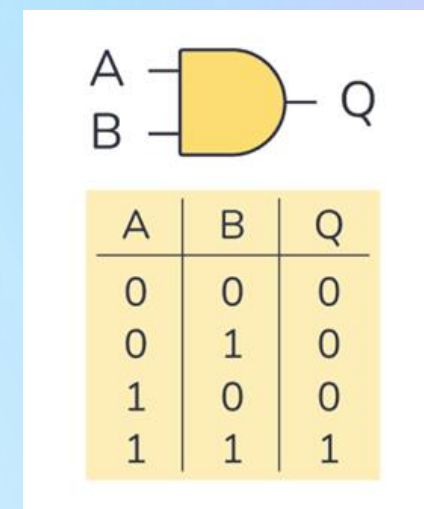
Például: **ÉS kapu**

- Két bemeneti érték: A és B
- Egy kimeneti érték: Q
- Q csak akkor 1, ha minden bemenet 1

Sok egyéb kapu: VAGY, NEM-ÉS, NEM-VAGY, KIZÁRÓ VAGY

- **Ezek kombinációjából rengeteg feladat megvalósítható**

Egy modern gépben több milliárd ilyen kapu!



[illegible]

Kapuk általánosítása

Kapuk magukban csak a **legegyszerűbb műveleteket** tudják

Kombinációjukkal már bonyolultabb művelet is megvalósítható, DE

- Pontosán meg kell tervezni, milyen kapukat használunk
- Pontosán össze kell kötni az egyes kapukat
- Akár csak egy hiba vagy módosítás az egész rendszert tönkre tudja tenni

Egy ilyen rendszer megtervezése **rengeteg munkát vesz igénybe**

Hogyan tudnánk általánosítani, automatikussá tenni a hálózat létrehozását?

- Véletlenszerű próbálkozások? – Tú a szénakazalban
- Iteráció jó megoldás felé? – Hogyan, ha a legkisebb változás is teljesen más eredményt ad?

Kapuk általánosítása – Első lépés

Probléma: Sok különböző fajta kapu, melyiket válasszuk?

Megoldás: Készítsünk egy kaput, ami minden esetet le tud fedni!

Perceptron

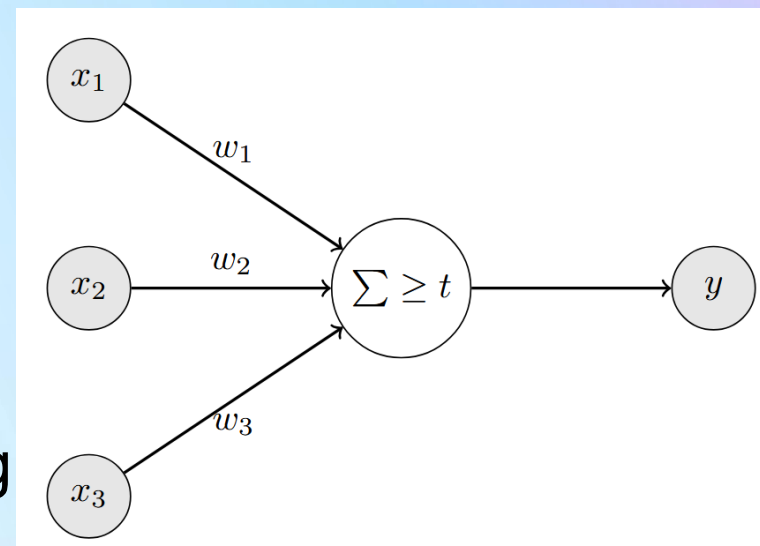
- Tetszőleges számú, digitális (0 vagy 1) bemenet
- Egy darab digitális kimenet

Paraméterei

- Minden bemenethez tartozik egy **súly** (jellemzően w -vel jelölve)
- A kimenethez tartozik egy **aktivációs szint**

A **perceptron kimenetét** a következő módon kapjuk meg

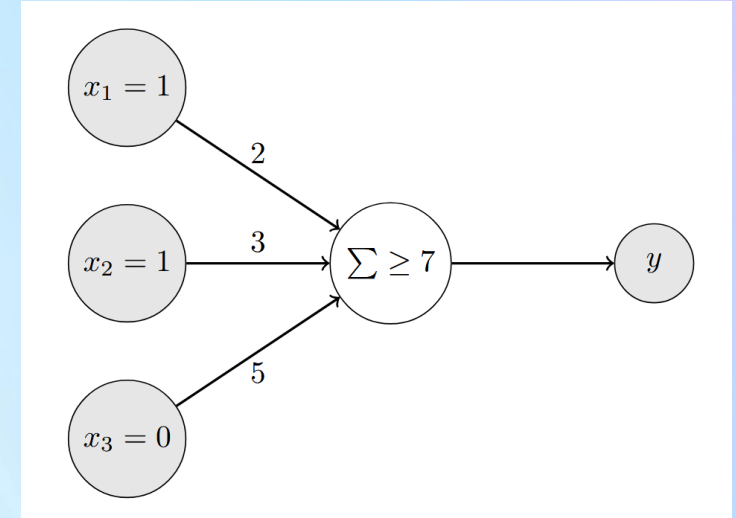
1. **Szorozzuk** meg mindegyik bemenetet a súlyával
2. **Adjuk össze** ezeket az értékeket
3. **Ha ez nagyobb vagy egyenlő** az aktivációs szintnél / szinttel, 1 a kimenet, egyébként 0



Perceptron példa

Számoljuk ki a következő perceptron kimenetét:

- 1: $1 * 2 = 2$
- 2: $1 * 3 = 3$
- 3: $0 * 5 = 0$
- Az összegük: $2 + 3 + 0 = 5$
- Mivel az összegük kisebb az aktivációs szintnél (7), 0 a kimenet



Neurális hálózat

Egy **perceptron** képes tetszőleges ÉS-VAGY kapcsolatot kifejezni

- **Képes leváltani a logikai kapuk szerepét**

Perceptronok **egymás után kapcsolhatóak**

- Egy perceptron kimenete egy másik perceptron bemenete

Így már szinte **tetszőleges logikai kör kifejezhető** perceptronokkal

Ha olyan feladatokat szeretnénk megoldani, ami nekünk könnyű, miért ne **kövessük a biológiát?**

Agyon belül apró **neuronok**, melyek között **kapcsolatok** alakulnak ki

Mesterséges neurális hálózat: A biológiai neuronok utánzásaképp kialakított számítási modell

- Apró számító egységek, illetve a közöttük kialakított kapcsolatok

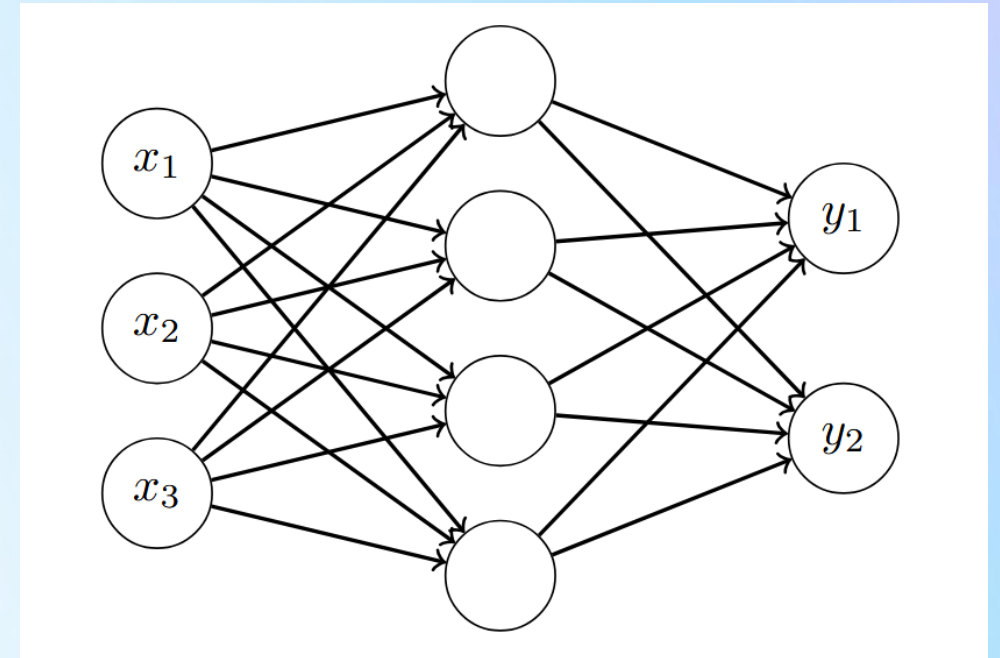
Perceptron hálózat (neurális hálózat)

Egy **perceptron** megfeleltethető **egy neuronnak**

Perceptron hálózat -> neurális hálózat

Réteges felépítés

- Valóságban bonyolult struktúrák alakulnak ki, de a kapcsolatok kialakítását nehezebb modellezni
- Rétegekkel könnyebb kezelni a hálózatot
- **Bemeneti réteg**: Csak „küldi” az értéket
- **Kimeneti réteg**: Ugyanúgy perceptron, de jelöljük a végét
- Perceptron több kimenettel: Ugyanaz az érték megy több helyre
- Egy hálózattal **több** kimeneti értéket is számolhatunk
- Közbülső rétegek: **rejtett réteg**



Perceptron továbbfejlesztés – Bemenet

Logikai kapuk **bemenete** csak digitális (0 vagy 1) lehetett

- Nehéz másra értelmezni például egy ÉS kapcsolatot

De sokszor **szükségünk lehet folyamatos értékek kezelésére**

- Pl.: „Legalább 26°C hőmérséklet legyen” helyett vizsgálhatnánk az aktuális hőmérséklet értékét

Perceptron működése **szorzás és összeadásból áll**

- Ugyanúgy alkalmazható 0 - 1 helyett folytonos értékekre is

Bemenet tetszőleges szám lehet így, de általában 0 - 1 közé szorítjuk

- Pl.: Ha 0°C és 100°C közötti értékekre számítunk, akkor a 26°C 0,26-nak felel meg

Perceptron továbbfejlesztés – Aktivációs limit

Az **aktivációs limit** egy olyan határ, mely minden perceptronban különbözhet

- Jobban örülnénk, ha minden perceptron ugyanazt csinálná

Mozgassuk át a **limitet az összegzésbe!**

- **Vonjuk ki** a bemenetek súlyszorzatának összegéből az aktivációs limitet
- Ha ez az érték kisebb, mint 0, akkor a kimenet is 0, egyébként 1
- A negatív aktivációs limitet szokták **eltolásnak** (bias) hívni, és b -vel jelölni

$$\text{kimenet} = \begin{cases} 0, & \text{ha } w \cdot x + b \leq 0 \\ 1, & \text{ha } w \cdot x + b > 0 \end{cases}$$

Perceptron továbbfejlesztés – Kimenet

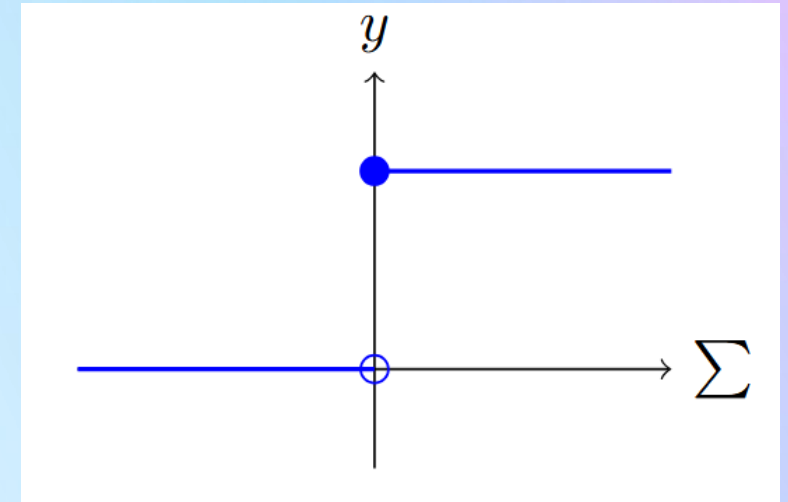
Folytonos bemenet esetén a perceptron kimenete az **ugrás** függvényvel jellemezhető

- Az aktivációs határig 0
- Utána 1

Ezzel **szimulálhatóak a különböző logikai kapuk**

De **tanuláshoz alkalmatlan**

- Az **aktivációs limit a kulcs** paramétere
- Ha a bemenet változása során ezt nem lépi át, a kimenet ugyanaz marad
- A határon viszont kicsi változtatás is óriási különbséget okoz a végeredményben



Taníthatóság

Mitől lesz egy struktúra **tanítható**?

Paraméterek kis változása a kimenetben kis változást okozzon

- Ki tudjuk próbálni, hogy egy hibás kimenet esetén melyik paramétert melyik irányba kell változtatni

Ne legyenek ugrásszerű változások a bemenet változása és kimenet változása között

- Például egy kép értelmezésénél nem egy pixel tartalma fogja leírni a teljes képet
- Sokkal inkább a teljes bementi adat vizsgálata a fontos

De mégis közelítsük meg a perceptron működését

- Nagyon negatív értékeknél 0-hoz közeli érték legyen a kimenet
- Nagyon pozitív értékeknél 1-hez közeli érték legyen a kimenet
- Kettő között valami köztes érték

Aktivációs függvény: Nem egy ugrás, hanem egy **függvény fogja leírni a működést**

Perceptron továbbfejlesztés – Sigmoid neuron

Több próbálkozás után a **sigmoid aktivációs függvény** vezetett megfelelő eredményre

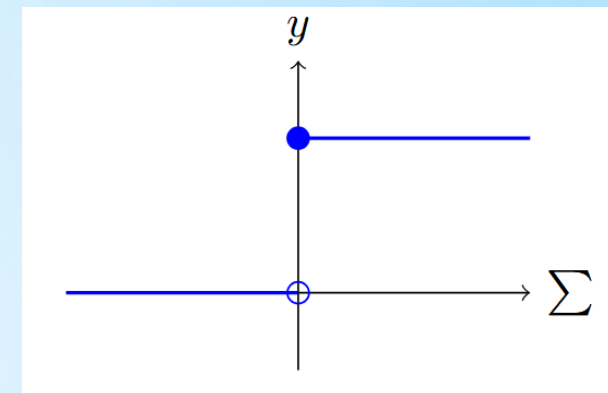
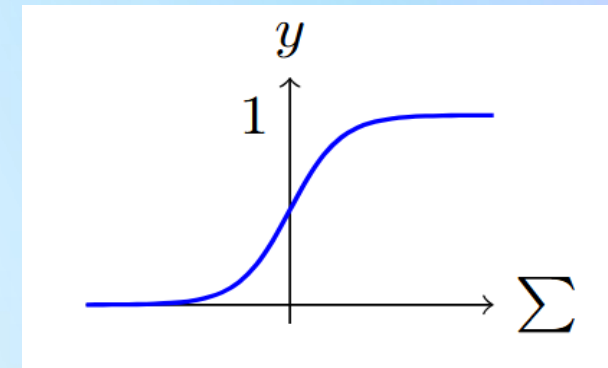
- A függvény: $y = \frac{1}{1+e^{(-\Sigma)}}$

- Gyakran a σ jellel jelölik

Ezt a perceptront hívjuk **sigmoid neuronnak**

De léteznek **egyéb aktivációs függvények** is

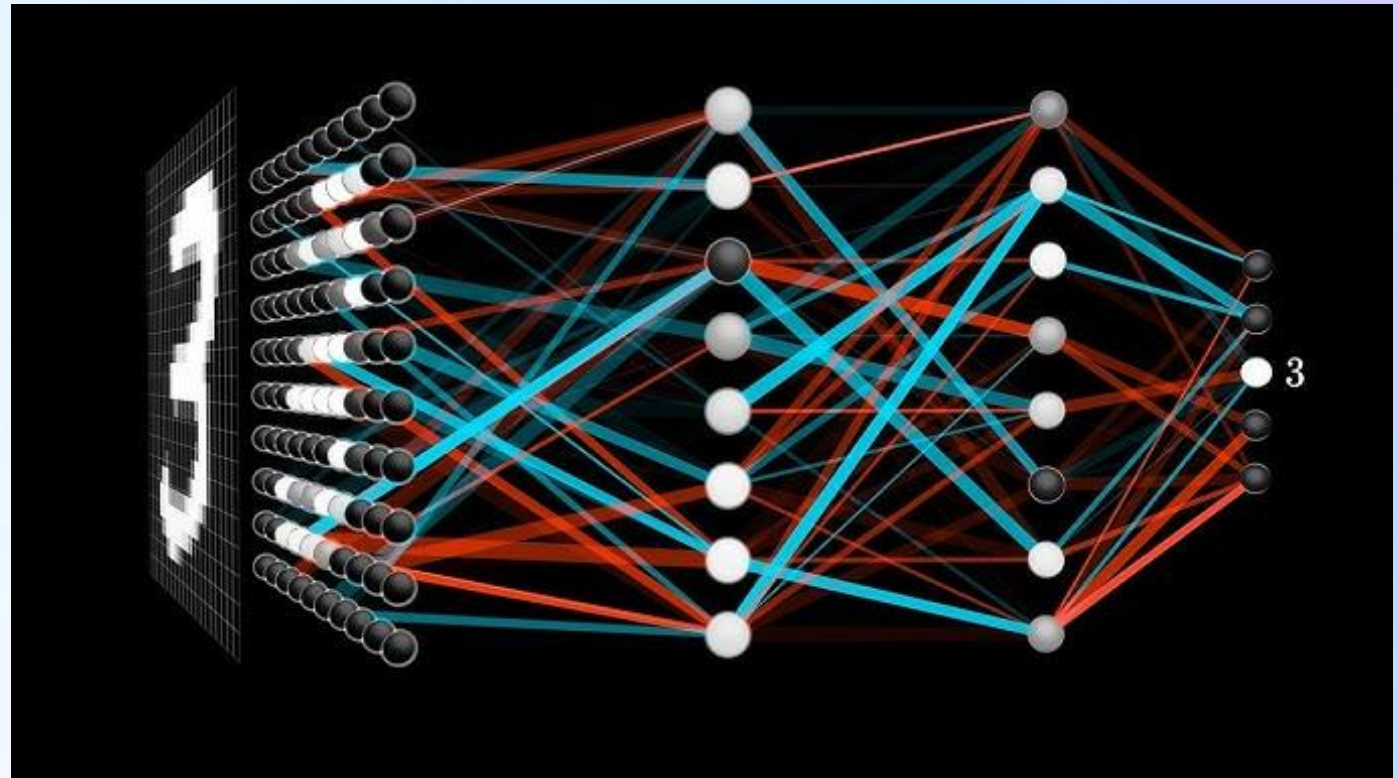
- Más algoritmussal, de ugyanúgy használható tanításra
- Nincsen tökéletes megoldás, mindegyik feladatfüggő



Áttekintés

- **Gépi tanulás és mélytanulás**

- Alapok
- Tanuló algoritmusok
- **Neurális hálózat tanítása**
- Gyakorlat



Számfelismerő háló – Feladat specifikálása

Ismerjünk fel kézzel írt számokat!

Embereknek általában könnyű ez a feladat

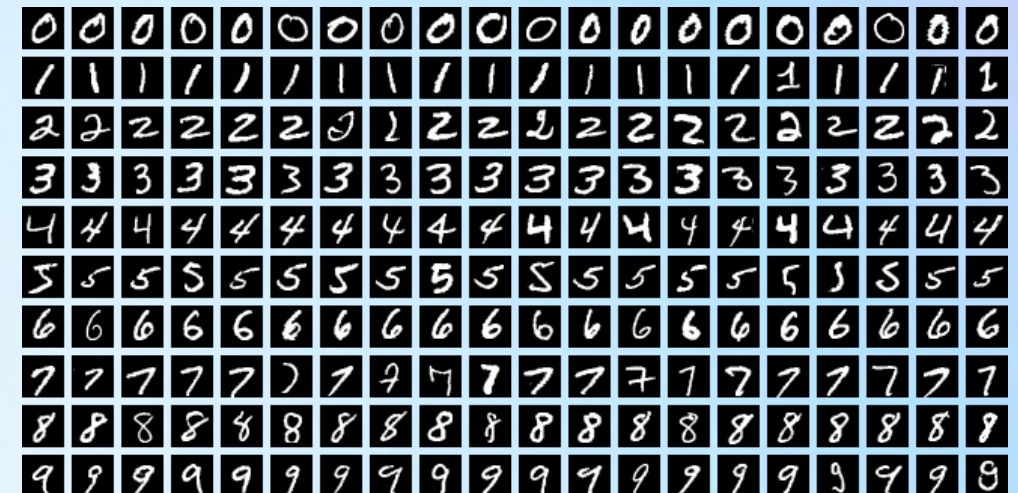
De **nehéz** pontosan **specifikálni** egy szám írási módját

- Pl.: 1-es és 7-es közötti különbség
- 6-os, 8-as és 9-es keverése, ha elnyújtják az írást

Vizsgáljuk meg, hogyan lehetne egy neurális hálót létrehozni és betanítani!

Szükség van egy adathalmazra, amiben megvannak jelölve a számok

- Mi is elkészíthetjük
- **MNIST adatbázis:** összesen 70'000 írott szám és azok címkéje
- 28 x 28 pixel méretű szürkeárnyaltos képek



Hálózat specifikálása

Egy **egyszerű hálózatot készítünk** a feladat megoldására

Bemenet: minden pixelhez tartozzon **egy bemeneti csomópont**

- 28×28 pixel = 784 bemeneti pont
- Szürkeárnyaltos kép: 0 jelöli a fekete pixelt, 1 a fehéret, szürke átmenet a kettő között

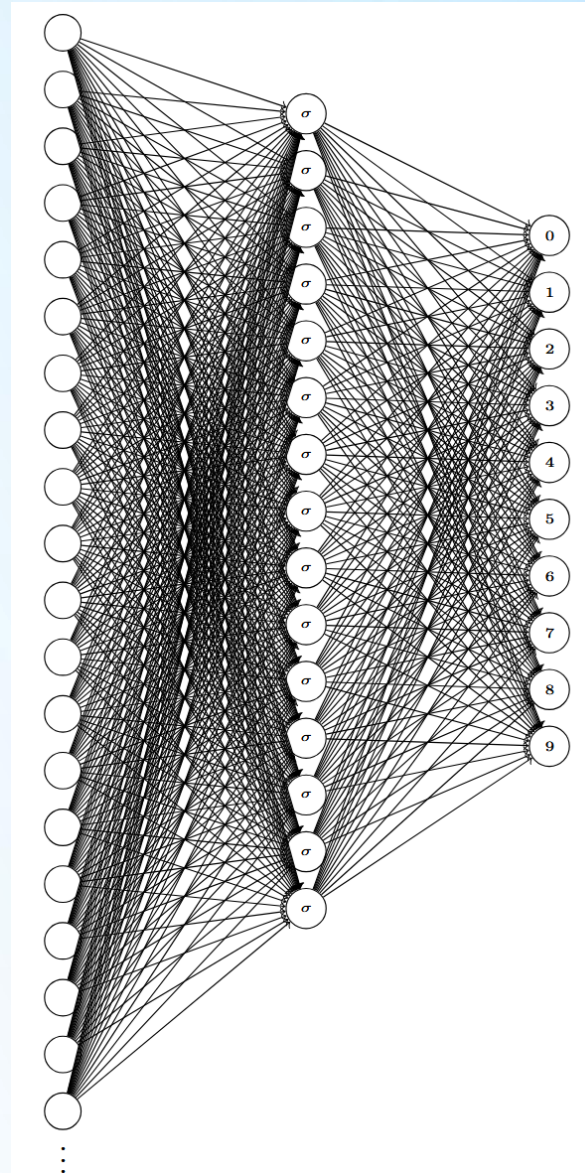
Kimenet: minden lehetséges szám jelöléséhez egy kimeneti csomópont

- Azt választjuk, aminek a legnagyobb az aktivációja, „legvalószínűbb”
- Miért nem csak egy kimenetünk van a szám értékével? Mert bonyolultabb összefüggéseket kíván, pl. 1 és 7 hasonlít, de számailag messze vannak egymástól

Kettő között **egy rejtett réteg**

- Sigmoid neuron réteg
- A méretét a mi feladatunk megválasztani, legyen kezdetben 15

Hálózat specifikálása - Ábra



Hálózat létrehozása

Gyorsabb és egyszerűbb programozás érdekében **vektorok** használata

```
def __init__(self, layer_sizes):  
    """  
    Initialize the network with random weights and biases.  
  
    Parameters:  
        layer_sizes: list of integers, e.g. [784, 30, 10]  
                     Each integer is the number of neurons in that layer.  
    """  
    self.num_layers = len(layer_sizes)  
    self.layer_sizes = layer_sizes  
  
    # Initialize biases for each layer (except the input layer).  
    # Each bias is a column vector with one value per neuron.  
    # We use small random values from a standard normal distribution.  
    self.biases = [  
        np.random.randn(size, 1)  
        for size in layer_sizes[1:]  
    ]  
  
    # Initialize weights connecting consecutive layers.  
    # weights[i] is a matrix of shape (next_layer_size, current_layer_size).  
    # Each entry w[j][k] is the weight from neuron k in one layer  
    # to neuron j in the next layer.  
    self.weights = [  
        np.random.randn(next_size, curr_size)  
        for curr_size, next_size in zip(layer_sizes[:-1], layer_sizes[1:])  
    ]
```

Hálózat kiértékelése

```
def feedforward(self, a):  
    """  
    Pass input through the network and return the output.  
  
    For each layer, compute:  $a' = \sigma(w \cdot a + b)$   
    where  $w$  = weights,  $b$  = biases,  $\sigma$  = sigmoid function.  
  
    Parameters:  
        a: input column vector (e.g. shape (784, 1) for an image)  
  
    Returns:  
        output column vector (e.g. shape (10, 1) for digit probabilities)  
    """  
    for w, b in zip(self.weights, self.biases):  
        # Weighted sum + bias, then apply sigmoid activation  
        a = sigmoid(np.dot(w, a) + b)  
    return a
```

Hálózat javítása – Mélytanulás

Kezdetben a súlyok és eltolások (együttesen **paraméterek**) **véletlenszerű értékek**
Ezzel már kipróbálhatjuk a hálózatot, de sok értelme nincs, **véletlenszerű végeredményeket** kapunk

- Minden kimenetnek ugyanakkora az esélye, hogy a legnagyobb lesz

A **paramétereket tetszőlegesen állíthatjuk**, de hogyan?

- Első ötlet: ismerjük egy bemenetre a helyes kimenetet
- Módosítsuk valahogy a paramétereket, hogy jó kimenetet kapjunk
- Könnyen kivitelezhető, elég pár jól megválasztott paramétert változtatni
- De ezzel valószínűleg elrontjuk az összes többi bemenetre létrehozott kimenetet
- Második ötlet: nézzük a paraméterek hatását egyszerre az **összes** bemeneten
- **Mélytanulás (deep learning):** a belső, rejtett réteg súlyainak változtatása a tanítás érdekében

Működés „mérése” – Költségfüggvény

Vezessünk be egy új értéket, mely a hálózat „jóságát” méri az adatokon!

Költség vagy **veszteségfüggvény**

Milyen tulajdonságokat várunk el egy ilyen függvénytől?

- Legkisebb értékét akkor vegye fel, ha **minden** bemenetre **tökéletes** választ kapunk
- Legkisebb érték célszerűen 0 legyen
- Preferáljuk a sok kisebb hibát a pár nagyobb hibához képest
- Az érték legyen független az adatok mennyiségétől

Javaslat: **átlagos négyzetes hiba**

- Bemenetei: a paraméterek (ezeken tudunk módosítani)
- $y(x)$: Az x adatra **adott** kimeneti érték
- a_x : Az x adatra **elvárt** kimeneti érték

$$C(\omega, b) = \frac{1}{n} \sum_x (y(x) - a_x)^2$$

Költségfüggvény minimalizálása

A cél olyan **paraméter értékek** keresése, melyek **a költségfüggvényt minimalizálják**

- Legjobb, ha 0, ekkor mindig tökéletes választ adna a hálózat, de ez irreális elvárás

A költségfüggvény **csak a paramétereiktől függ** (súlyok és eltolások)

- $y(x)$ függ a paramétereiktől
- Nagyon bonyolult, több tízezer paraméter, nincs explicit képlet

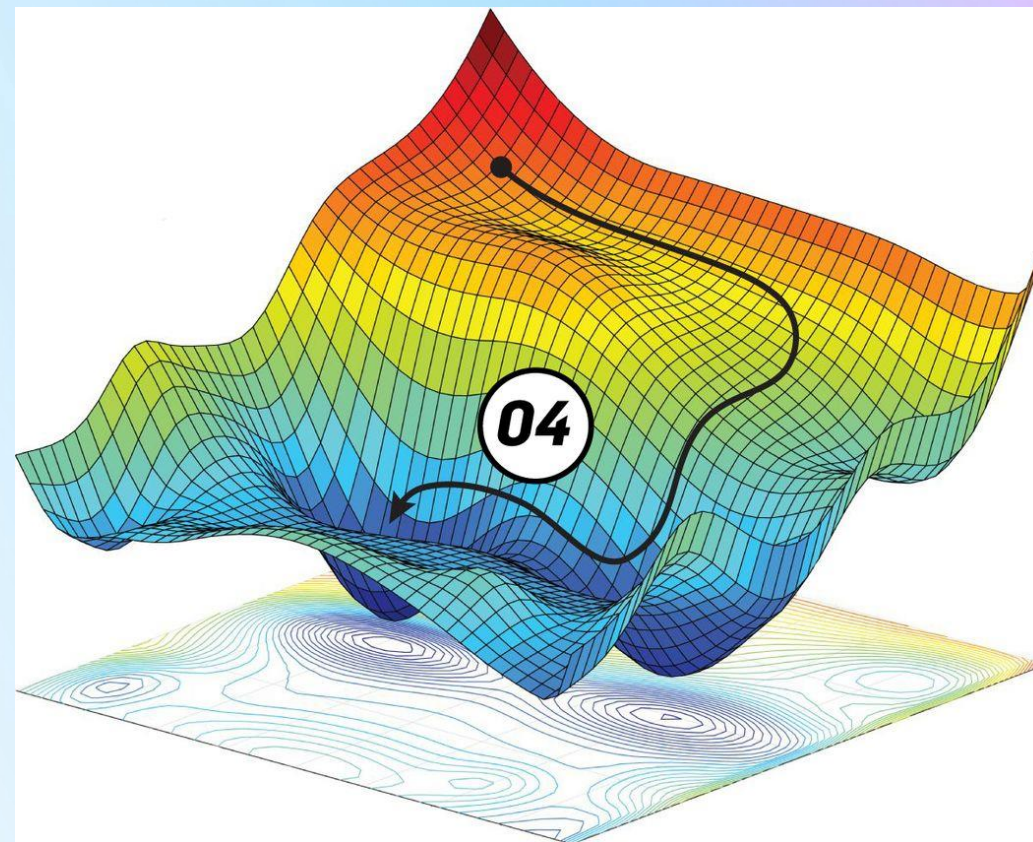
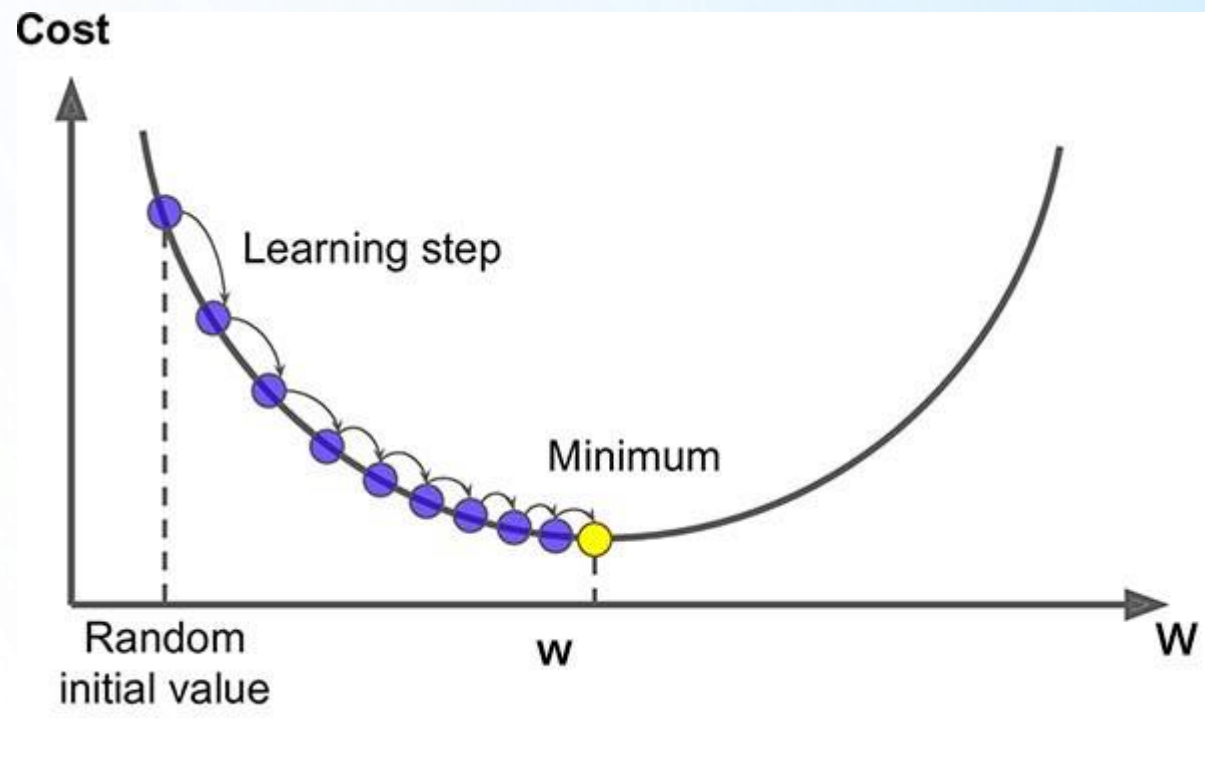
Hogyan tudjuk mégis megkeresni a minimumát?

Haladjunk csak kis lépésekben, törekedjünk, hogy mindig picit jobb eredményt kapjunk!

Gradiens menti csökkentés

- **Gradiens**: megadja egy függvény egy pontjában, melyik irányba milyen **meredek** a függvény
- **Legmeredekebb** nyomvonalat **követve** eljutunk egy **minimum pontba**

Gradiens menti csökkentés



Gradiens keresése

A **gradiens függ a választott költségfüggvénytől**

Négyzetes költségfüggvény előnye: **könnyen számolható** a gradiens

- Tanítóadatok nem befolyásolják egymás költségét

Működési elv

- A gradiens külön-külön számolható a bemeneti értékekből
- Szükségünk van az aktuális **aktivációs** értékre egy bemenetnél
- A gradienst a kimenetnél egyszerű számítani
- A gradienst a rétegeken keresztül **visszavezetve** (backpropagation) megkapjuk az összes súlyra a gradienst

Végül ezeket a **komponenseket paraméterekként átlagoljuk**, ez a **teljes gradiens**

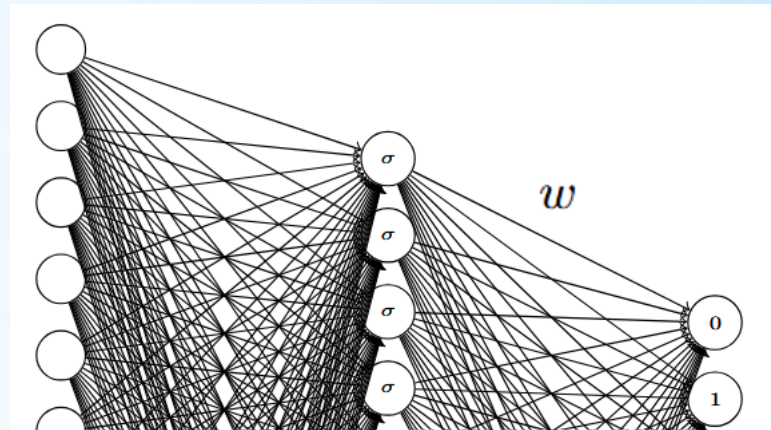
A gradiens ismeretében tudjuk úgy változtatni a paramétereket, hogy a **költségfüggvény csökkenjen**

Gradiens keresése – Magyarázat

A gradiens keresése a súlyok megfelelő változtatásához szükséges

A matematikai háttere bonyolult, de a lényege könnyen megérthető

Fókuszáljunk egyetlen egy súly értékre!



Kérdés: ennek egy **kis módosítása milyen hatással** van a költségfüggvényre?

Gradiens keresése – Kimeneti súly változtatása

A kimeneti függvényünk:

$$y = \frac{1}{1+e^{(-\Sigma)}}$$

ahol Σ jelöli az aktivációját a csomópontnak, ez függ a súlytól

Ha picit **növeljük a súlyt, növeljük a kimeneti értéket**

Ha picit **csökkentjük a súlyt, csökkentjük a kimeneti értéket**

Cél: csökkentjük a költséget, tehát közelítsük a kimenetet az elvárt értékhez

Ezt továbbvihetjük a **többi súlyra is**

- **Fontos: ugyanakkora súly változás eltérő költségcsökkentést** eredményez (máshol vannak az aktivációs függvényben)
- Azokat a súlyokat jobban változtatjuk, ahol nagyobb eredményt érünk el

Miért akarunk egyszerre **több súlyt változtatni?**

- Egy súly változtatása: nagyon direkt, beleszól a többi eredménybe
- Több súly: jobban eloszlik a változás, egy adatra célirányos, a többire viszont „véletlenszerű”

Sztochasztikus gradiens számolás

Költségfüggvény gradiens számolásához szükséges az **összes** adatelemhez rendelt **gradiens számolása**

- Gradiens mentén csak kis lépésekben haladhatunk
- Minden egyes lépésnél ez rengeteg számolást jelent
- Jó lenne, ha lehetne ezt a lépést optimalizálni

Sztochasztikus gradiens

- Az összes adat helyett válasszunk ki véletlenszerűen egy kisebb részt
- A részadatokra számolt gradiens nagy valószínűséggel meg fogja közelíteni a teljes adathalmaz gradiensét
- Kisebb csoportok, **mini-batch**-ek
- Pl.: a számfelismerős tanulásnál 60'000 bemeneti adatunk van, de 10 darab lesz egy batch-ben
- Mire végigfuttatjuk az összes adaton a gradiens menti csökkentést, 6'000 lépést tudunk megtenni az egy helyett!
- Teljes adathalmazon végigmenni: **epoch**

Sztochasztikus gradiens – Csoportosítás

```
# Shuffle training data randomly for each epoch
# so mini-batches are different every time
np.random.shuffle(training_data)

# Split the training data into mini-batches
mini_batches = [
    training_data[k:k + mini_batch_size]
    for k in range(0, n_train, mini_batch_size)
]

# Update weights and biases for each mini-batch
for mini_batch in mini_batches:
    self.update_mini_batch(mini_batch, learning_rate)
```

Sztochasztikus gradiens – Mini-batch feldolgozása

```
# Accumulators for the total gradient over this mini-batch
# (initialized to zeros, same shape as biases and weights)
total_bias_gradient = [np.zeros(b.shape) for b in self.biases]
total_weight_gradient = [np.zeros(w.shape) for w in self.weights]

# Sum up the gradients from each training example
for x, y in mini_batch:
    # Backpropagation gives us the gradient for this single example
    bias_grad, weight_grad = self.backpropagate(x, y)

    # Add this example's gradient to the running total
    total_bias_gradient = [tb + bg for tb, bg in
                           zip(total_bias_gradient, bias_grad)]
    total_weight_gradient = [tw + wg for tw, wg in
                             zip(total_weight_gradient, weight_grad)]

# Update rule: new = old - (learning_rate / batch_size) * gradient
# The minus sign means we move OPPOSITE to the gradient (downhill)
batch_size = len(mini_batch)
scale = learning_rate / batch_size

self.biases = [b - scale * tb for b, tb in
                zip(self.biases, total_bias_gradient)]
self.weights = [w - scale * tw for w, tw in
                 zip(self.weights, total_weight_gradient)]
```

Tanító adathalmaz kezelése

Hogyan használjuk fel a rendelkezésünkre álló tanítóhalmazt?

Adjuk oda mindet a tanító algoritmusnak?

- Legtöbb adat, több információt ki tud szűrni az algoritmus
- DE: célunk, hogy még korábban nem látott bemenetre is jó választ adjon
- Honnan tudom, hogy a hálózat az általános szabályokat tanulja, és nem az adott adatokra specifikus tulajdonságokat?

Megoldás: **bontsuk szét az adathalmazunkat!**

- **Tanító** adathalmaz: ezt használjuk fel a súlyok állítása során
- **Ellenőrző** adathalmaz: ezzel ellenőrizzük, hogy tényleg jól működik a hálózat
- Az ellenőrző adathalmaznak így nincs hatása a tanítás során!
- Tipikusan a **teljes adathalmaz 10%-a**

Kitekintés – Hiperparaméterek kezelése

Sikeresen betanítottunk egy neurális hálót a számok felismerésére

De a **tanítás során** rengeteg **paramétert hasraütésszerűen** választottunk ki

- Rejtett réteg neuron száma
- Egy darab rejtett réteg
- Sigmoid függvény
- Négyzetes költségfüggvény
- Rétegek közötti kapcsolatok felépítése

A paraméterekhez hasonlóan ezeket, a **tanítás környezetét meghatározó paramétereket hiperparamétereknek** nevezzük

Ugyanúgy **kereshető** ezeknek az **optimuma!**

- Erre is külön adathalmazt kell fenntartani: **teszt** adathalmaz
- Ez is tipikusan a **teljes adathalmaz 10%-a**

Kitekintés – Könyvtárak használata

PyTorch: népszerű Python alapú neurális hálózatokat kezelő könyvtár

Bonyolult a használata, sok ismeretet igényel, de megkönnyíti a hálózatok létrehozását és tanítását

Feladat: valósítsuk meg ezt a számfelismerős feladatot a PyTorch könyvtár segítségével!

Hálózat létreho-

```
def __init__(self, hidden=30):  
    super().__init__()   
    self.fc1 = nn.Linear(784, hidden)    # input -> hidden  
    self.fc2 = nn.Linear(hidden, 10)     # hidden -> output
```

Hálózat kiértékelése

```
def forward(self, x):  
    # Sigmoid on hidden layer  
    x = torch.sigmoid(self.fc1(x))  
    # Sigmoid on output layer (matching the numpy version's architecture)  
    x = torch.sigmoid(self.fc2(x))  
    return x
```

Kitekintés – Könyvtárak használata

Tanítási **paraméterek** választása:

```
loss_fn = nn.MSELoss()  
optimizer = optim.SGD(model.parameters(), lr=args.lr)
```

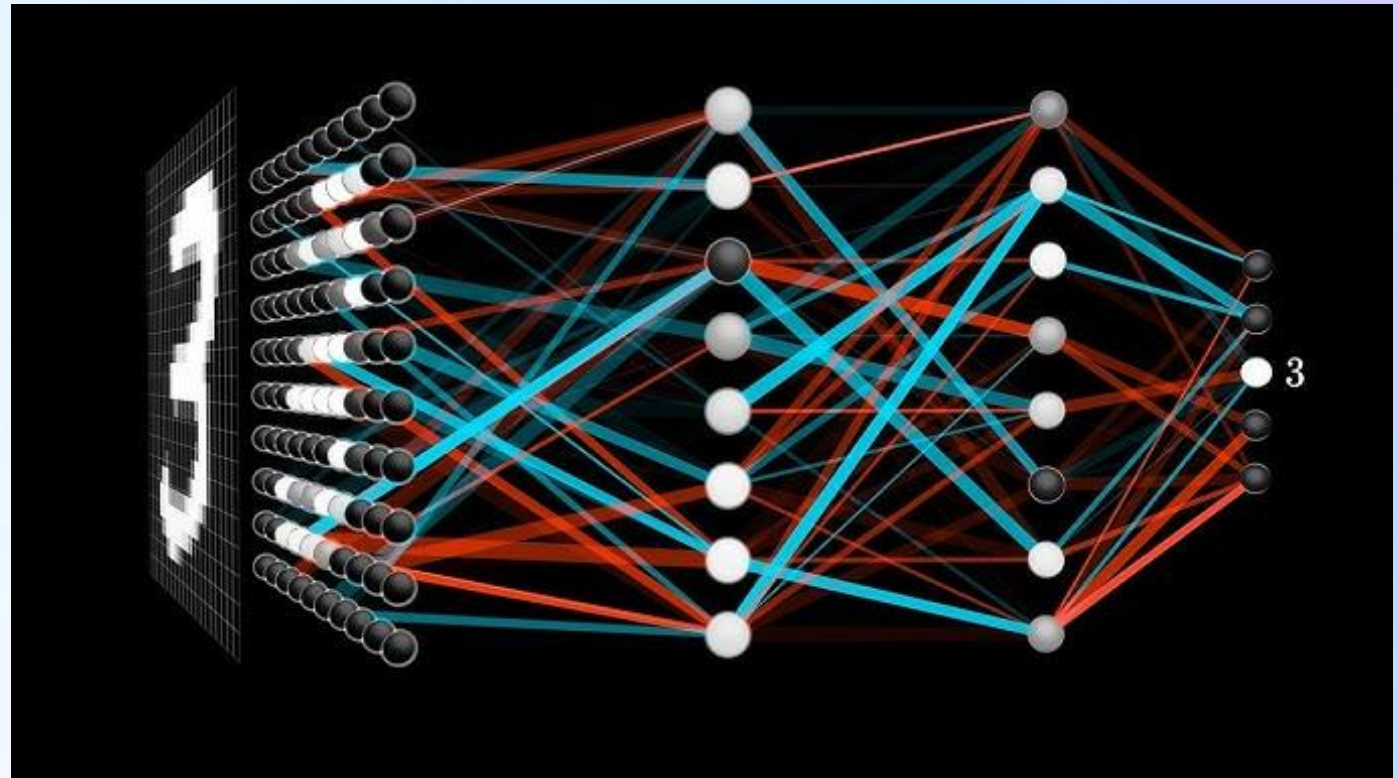
Tanítási **ciklus**:

```
output = model(images)  
loss = loss_fn(output, targets)  
  
optimizer.zero_grad()  
loss.backward()  
optimizer.step()
```

Áttekintés

- **Gépi tanulás és mélytanulás**

- Alapok
- Tanuló algoritmusok
- Neurális hálózat tanítása
- **Gyakorlat**



Perceptron – Gyakorló feladat

Készítsünk egy három bemenettel rendelkező perceptront, ami segít eldönteni, mikor mehetünk strandolni!

A három bemenet:

1. Süt-e a nap?
2. Legalább 26°C a levegő hőmérséklete?
3. Van szabad hely a strandon?

Kimenet: akkor mehetünk, ha van szabad hely, és vagy süt a nap, vagy elég meleg a levegő hozzá!

Sigmoid neuron – Gyakorló feladat

Próbáljuk ki a strandolós példát egy szigmoid neuron segítségével is!

Bemenetek:

1. Mennyire süt a nap? (0: teljesen sötétség, 1: déli napsütés)
2. Mi a hőmérséklet? (0: 0°C, 1: 100°C)
3. Mennyire üres a strand? (0: nincs szabad hely, 1: teljesen üres)

A **súlyok** ugyanúgy 2, 3, 5

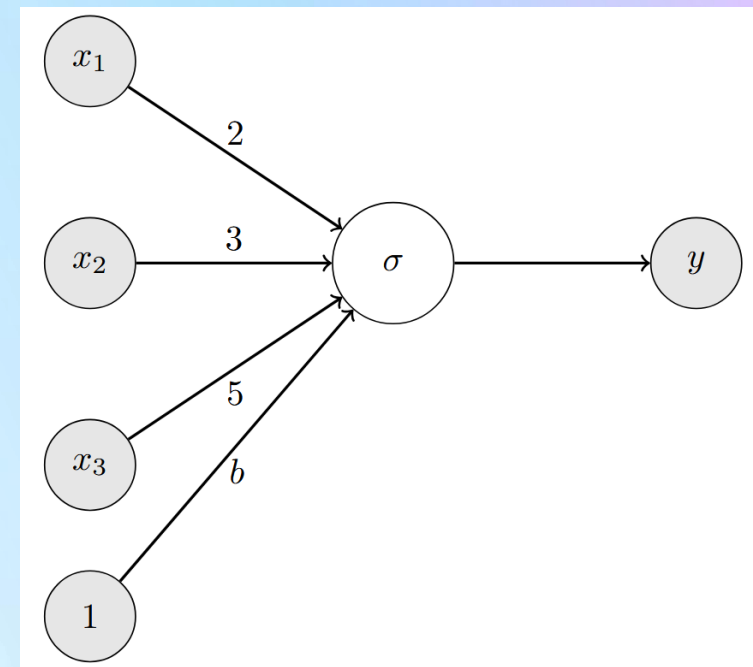
Mit kapunk az ideális értékekre (teljesen üres, 25 fok van, teljes napsütés)?

Mit kapunk a legrosszabb értékekre (minden 0)?

Hogyan érdemes beállítani az eltolást?

Számoljuk ezzel újra az értékeket!

Mi lenne, ha az ideális helyzetben a hőmérséklet 50°C lenne?



Költségfüggvény – Gyakorló feladat

Három tesztadattal rendelkezünk, a következő kimeneti értékekkel:

1. 13
2. 22
3. 18

Egy neurális háló két különböző paraméterhalmazzal az alábbi értékeket számolja ki:

- | | |
|-------|-------|
| 1. 15 | 1. 13 |
| 2. 21 | 2. 22 |
| 3. 19 | 3. 21 |

Melyik hálónál mekkora a költségérték? **Melyik hálózat tűnik jobbnak?**

Hálózat tanítása – Gyakorló feladat

A mellékelt Python projekt segítségével tanítsunk be egy neurális hálót a számjegyek felismerésére!

1. Nézzük át a kódot!
2. Hozzunk létre egy hálózatot!
3. Nézzük meg, milyen eredményeket ad!
4. Rajzoljunk egy saját, 28 x 28-as számot! Teszteljük ezen is!
5. Futtassunk le egy tanítási epoch-ot!
6. Próbáljuk meg újra a kiértékelést! Mit látunk?
7. Futtassuk addig, amíg pontos értékeket nem kapunk!

```
python .\neural_network.py
```

create

```
python .\neural_network.py show
```

```
python .\neural_network.py predict
```

\test.png

```
python .\neural_network.py train
```

GENERATÍV MODEL ÉS ALKALMAZÁSAIK



Funded by the European Union

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.



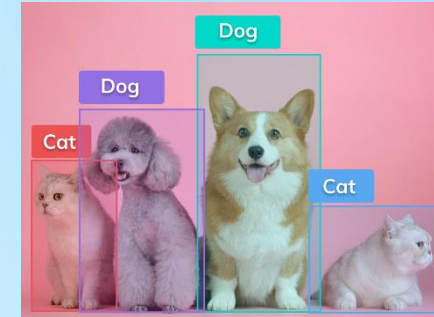
Áttekintés

- **Generatív modellek és alkalmazásaik**
 - Generatív modellek jelentése, működési alapelvek
 - Nagy nyelvi modellek (LLM)
 - Tipikus hibák az LLM használatakor
 - Adatok védelme, privát adatok
 - Promptolás
 - Promptolási minták használata a gyakorlatban
 - Promptolási példafeladatok különböző szakterületeket érintve
 - Promptolási példafeladatok multimédia tartalom generálására
 - Promptolási példafeladatok programozásra



Diszkriminatív modell kontra generatív modell

- **Gépi tanulásban** használt alapvető megközelítések
 - **Diszkriminatív modell** használata – tanulás segítségével képes **eldönteni** ismeretlen adatokról, hogy milyen tulajdonságok jellemzők rájuk. A tanulás tipikusan **felügyelt tanulás** címkézett adatokon
 - **Generatív modell** használata – tanulás után **képes új tartalom előállítására**. A tanulás tipikusan **felügyelet nélküli tanulás** címkézetlen adatokon



Generatív mesterséges intelligencia

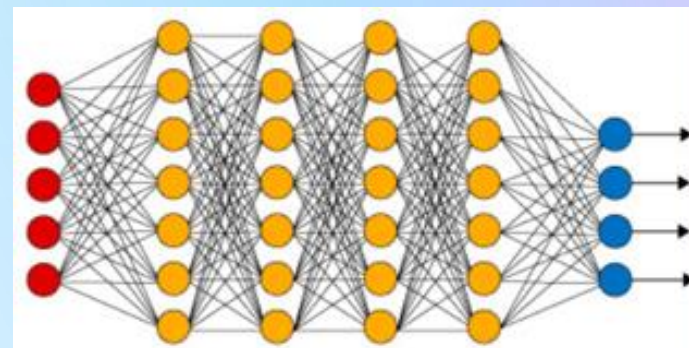
- **Generatív MI** (GenAI, GAI)
 - Olyan MI, amely **generatív modelleket használ** szövegek, képek, videók vagy más adatformák előállítására
 - Ezek a modellek **megtanulják a tanításhoz használt adatok mögöttes mintáit és struktúráit**, és ezeket használják fel új adatok előállítására a bemenet alapján, amely gyakran természetes nyelvi kérések formájában érkezik



Generatív neurális hálózatok

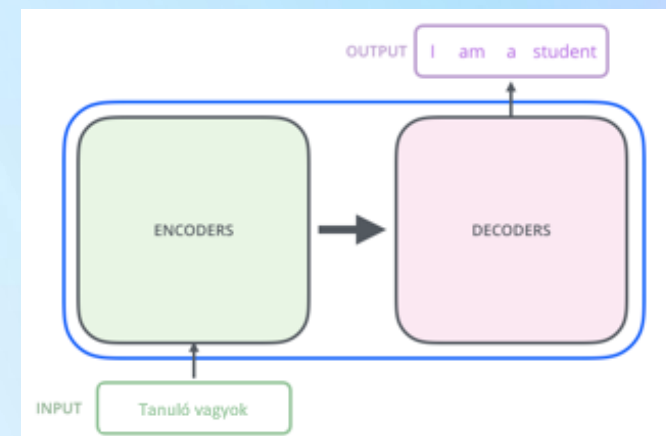
- **Neurális hálózatok, mélytanulás**

- **Kezdetben** a generatív neurális hálózatokat jellemzően **diszkriminatív modellként** kezelték és képezték
- **Később** megjelentek az olyan **mély neurális hálózatok** (pl. az AE – variációs autóenkóder, vagy a GAN – generatív versengő hálózat) amelyek képesek **generatív modelleket** tanulni összetett adatokra (pl. képekre)



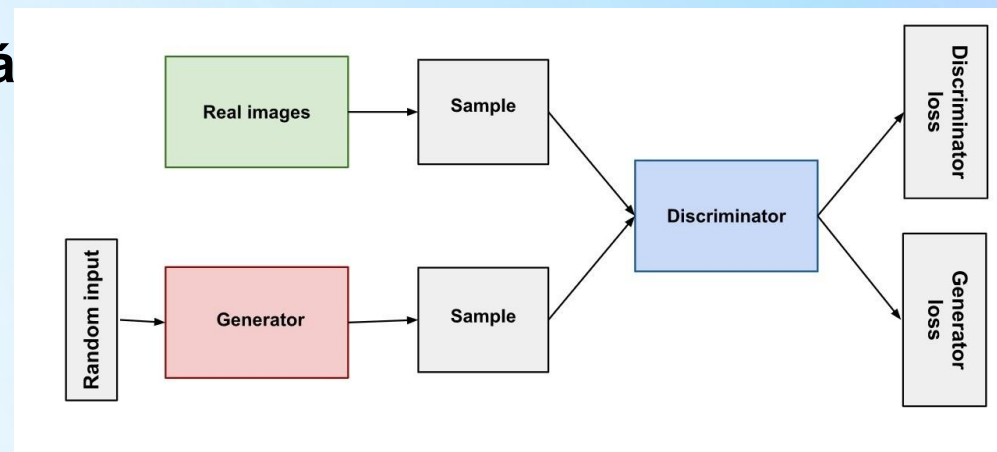
- **Transzformer hálózat**

- Generatív modellek továbbfejlesztésének **következő állomása**
- **Hatalmas adatmennyiségen felügyelet nélkül tanul**
- **GPT-1** (Generative Pre-trained Transformer), GPT-2, GPT-3, ...



Generatív versengő hálózatok

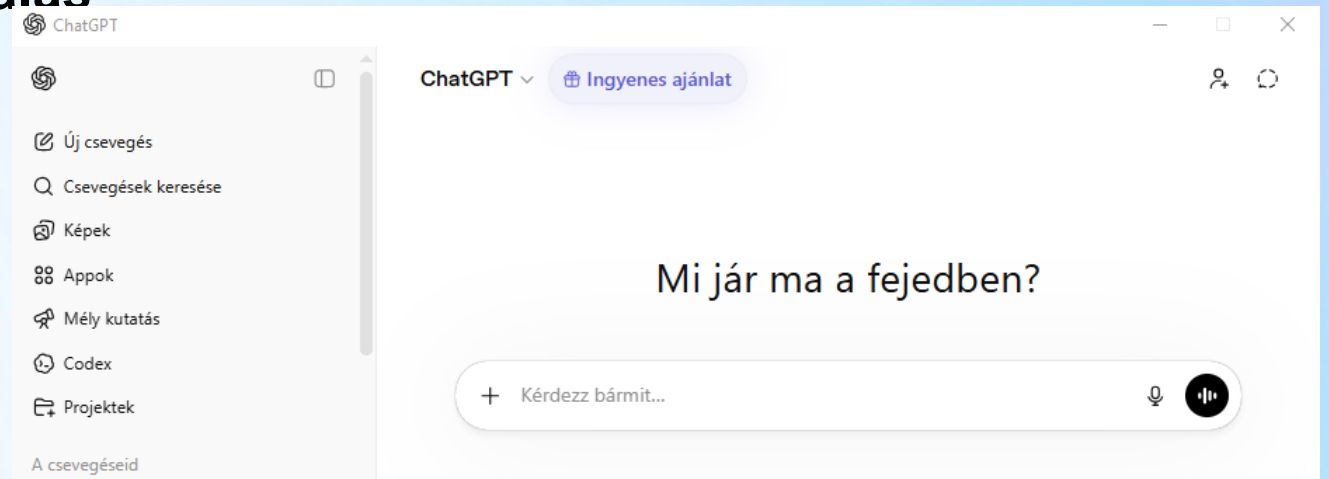
- **GAN** (Generative Adversarial Network)
 - **Két neurális hálózat versengésén** alapuló generatív modell
 - A **Generátor** megpróbál minél élethűbb, de **hamis tartalmat** (pl. hamis fotókat) **létrehozni**
 - A **Diszkriminátor feladata megkülönböztetni a valódi tartalmat** a Generátor által készített tartalomtól
 - Ahogy a Diszkriminátor fejlődik a felismerésben, a Generátornak is egyre kifinomultabbá kell válnia, hogy „átverje” azt, így végül a **versengés eredményeként** rendkívül **realisztikus tartalmak** születnek
 - **Felhasználási területük** elsősorban **képgenerálása** és **videók készítése**



Transzformer hálózatok

- **Transzformer hálózat (GPT)**

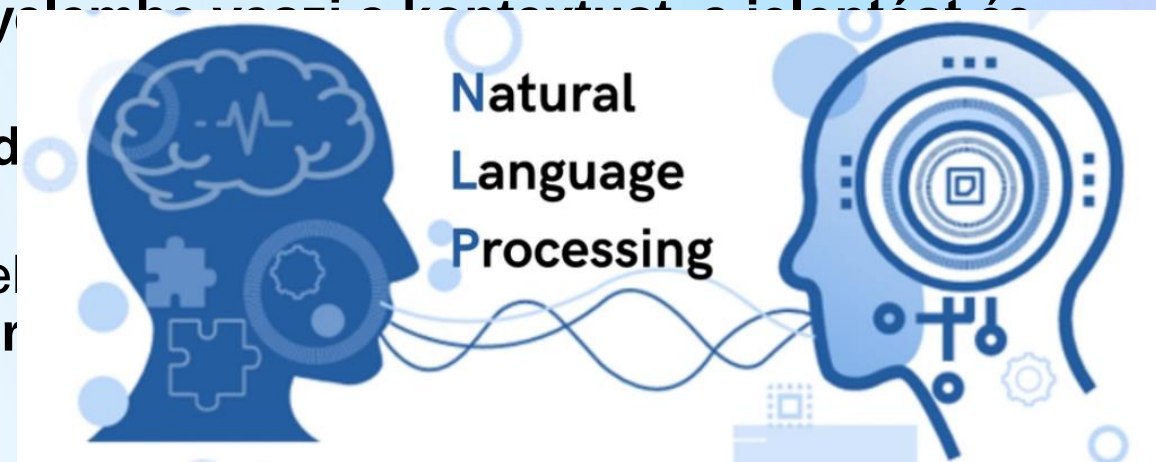
- Olyan modellek, amelyek a „**figyelem**” (attention) **mechanizmust használják** annak lehetővé tételére, hogy a rendszer **megértse az adatok** (például szavak egy mondatban) **közötti** távoli **összefüggéseket** és a kontextust
- Hatalmas mennyiségű adaton történő előtanítás után **képesek megjósolni a következő elemet egy sorozatban** (pl. a következő szót egy mondatban)
- **Felhasználási területük** elsősorban **szöveges tartalmak generálása, fordítás idegen nyelvek között, programkód generálás**



Természetes nyelvfeldolgozás

- **Természetes nyelvfeldolgozás (NLP)**

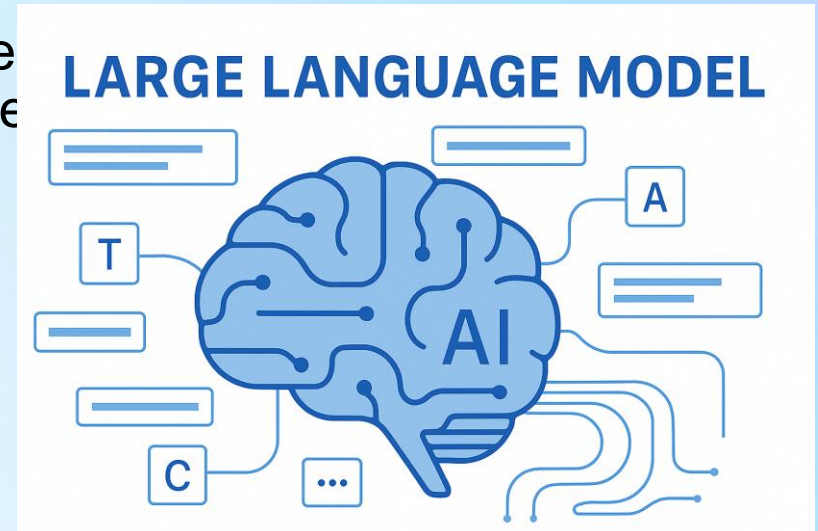
- Az MI egyik **régóta** kutatott és fejlesztett területe, **amely a számítógépek és az emberek közötti kommunikációt a természetes, mindennapi használt nyelv szintjén teszi lehetővé**
- Az NLP tulajdonképpen a **generatív MI előfutára**, célja, hogy a gépek megértsék, értelmezzék, feldolgozzák vagy akár elő is állítsák az emberi szöveget (mind az írott szöveget, mind a beszédet)
- A gép nem csak **felismeri a szavakat**, de **figyelmbe veszi a kontextust is**, azaz a **kapcsolatokat** is
- **Felhasználási** területe elsősorban a **gépi fordítás, diktálás, csevegő robot, ...**
- A korábbi NLP módszereket és megközelítéseket **mára lényegében felváltották a transzformer (GPT) alapú módszerek**



Nagy nyelvi modellek

- **Nagy nyelvi modell (LLM)**

- **A generatív MI nyelvre specializálódott változata, ami tudja értelmezni a természetes, emberi nyelvet**, képes szöveget generálni, fordítani, összefoglalni vagy programkódot írni
- Felépítését tekintve olyan **fejlett, többrétegű, transzformer hálózatokból álló rendszer**, amelyet hatalmas mennyiségű szöveges adaton (pl. könyveken, weboldalakon) tanítottak be
- **Statisztikai alapon jósolja meg, mi legyen a következő szó vagy karakter** egy mondatban, így hoz létre összefüggő válaszokat
- **Felhasználási területük szinte korlátlan**, minden területe megtalálható már, ahol tartalmat kell generálni (jellemzően az LLM az agya a tartalomgeneráló rendszernek)



Mit csinál az LLM?

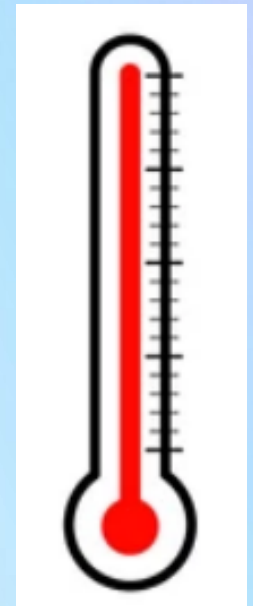
- **LLM** működésének **célja**

- A bemeneti szöveg alapján **megjósolja**, hogy **a kimenetben** mi lesz **a következő szó**
- Nem csak egy lehetséges szót ad meg, hanem az **összes** általa ismert **szónak** (ez az LLM szótára) **a valószínűségi eloszlását**, és ezen eloszlás alapján válassza ki a legnagyobb valószínűségűt



Mit csinál az LLM?

- **Hogy lehet az, hogy ugyanarra a bemenetre néha más lesz a kimenet?**
 - Az **LLM-nek van egy paramétere**, amit „**hőmérsékletnek**” hívunk
 - Ha ennek a paraméternek **0 az értéke**, akkor mindig a **legnagyobb valószínűségű** szót választja az LLM
 - Ha ezt a paramétert **0-nál nagyobb értékre** állítjuk, akkor néha előfordul, hogy a **2., 3. vagy akár az n. legnagyobb valószínűségű** szó kerül kiválasztásra



Mit csinál az LLM?

- **Hogyan áll elő a több szóból álló kimenet?**
 - Autoregresszív módon, **szóról szóra**

Nem esik messze az ... → **LLM** → alma

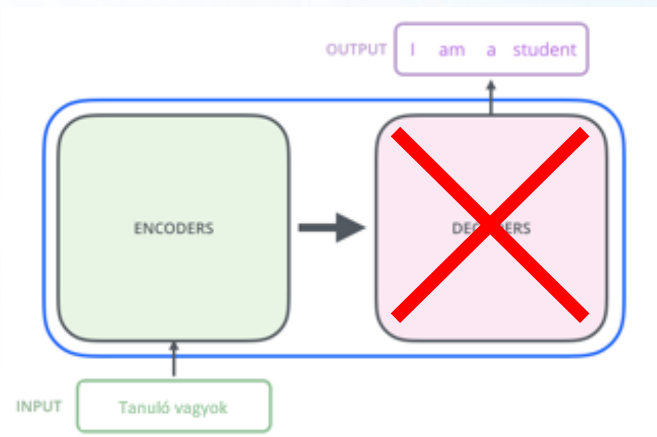
Nem esik messze az alma ... → **LLM** → a

Nem esik messze az alma a ... → **LLM** → fájától

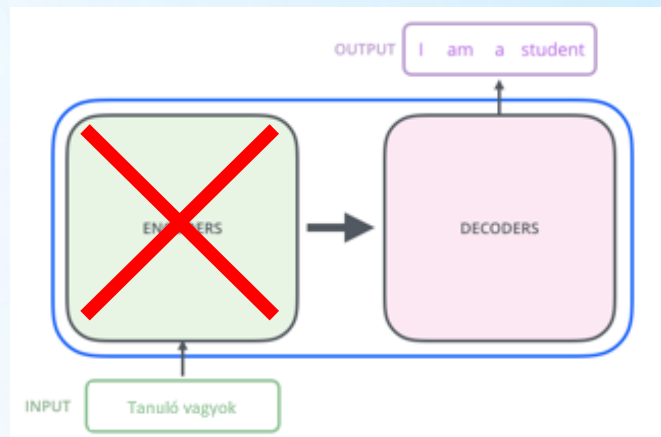
⋮

LLM modellarchitektúrák

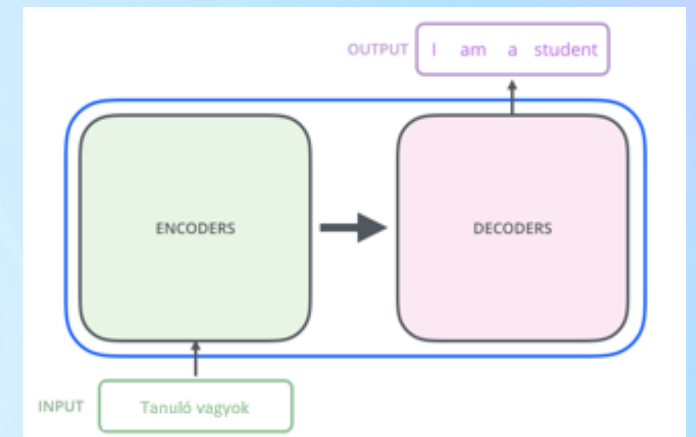
- Alapvetően **háromféle architektúra** használatos
 - **Csak kódoló**: szöveg megértésére ideális (pl. BERT)
 - **Csak dekódoló**: szöveg generálására ideális (pl. GPT-sorozat)
 - **Kódoló – dekódoló**: fordításra és komplex feladatokra ideális (pl. T5)



BERT



GPT

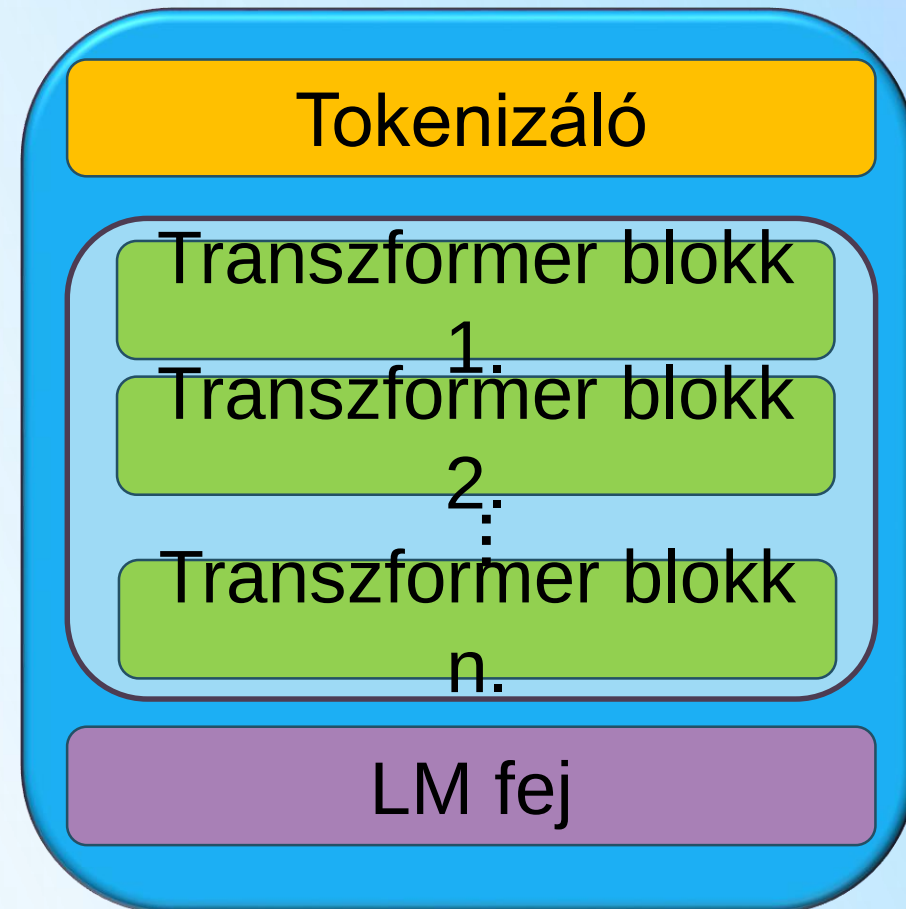


T5

Transzformer alapú LLM felépítése

- **A transzformer alapú LLM** (GPT-sorozat) alapvetően **három funkcionális blokkból áll**
 - Tokenizáló
 - Transzformer blokkok halmaza
 - LM fej

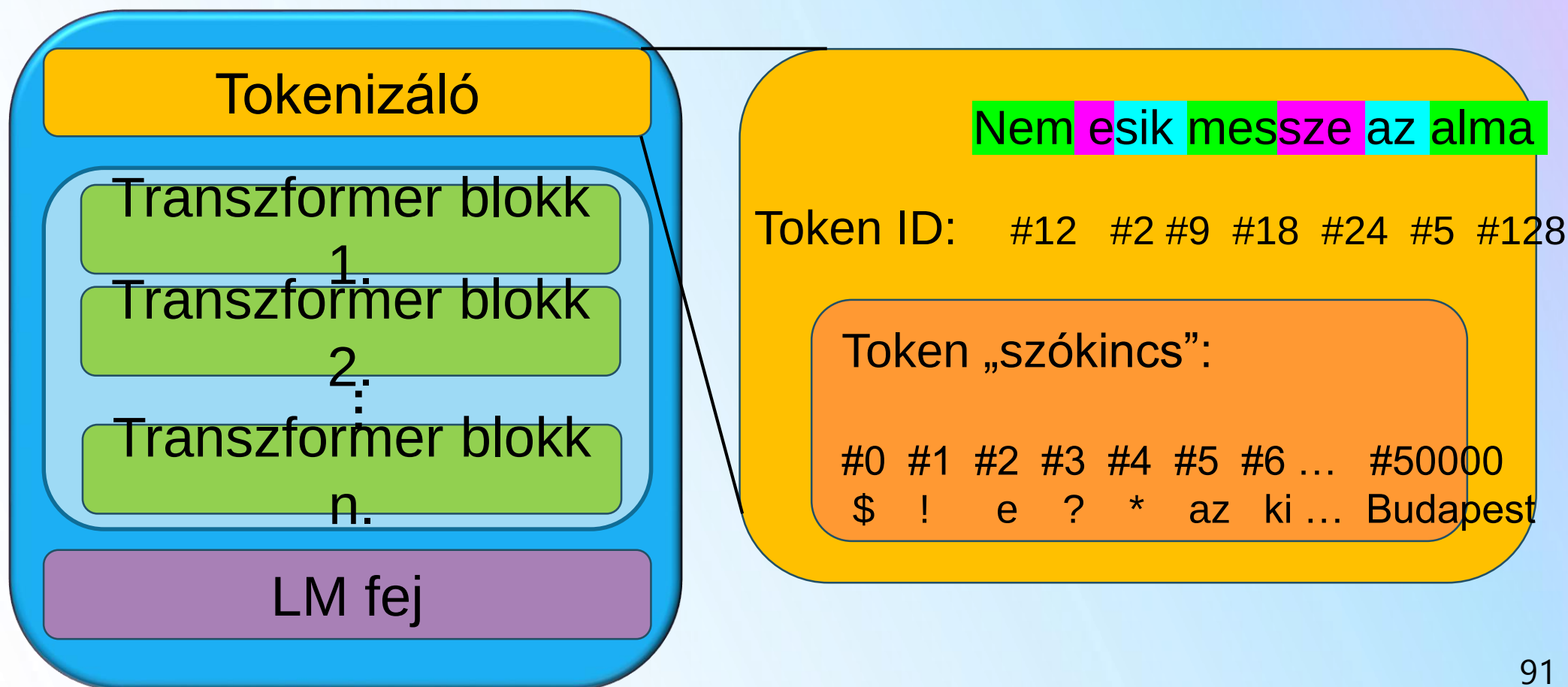
GPT



Transzformer alapú LLM működése

- **Tokenizáló** feladata
 - A **bemenetet tokenekre** bontja, ami tipikusan egy **szó, szórész, karakter**

GPT



Transzformer alapú LLM működése

- **Transzformer blokkok feladata**

- A bemeneti **tokeneket** szám **vektorokká** alakítja a „figyelem” eljárás és **neurális hálózatok segítségével**
- Ezen vektorokba már **kontextus információk** is beágyazásra kerülnek

GPT

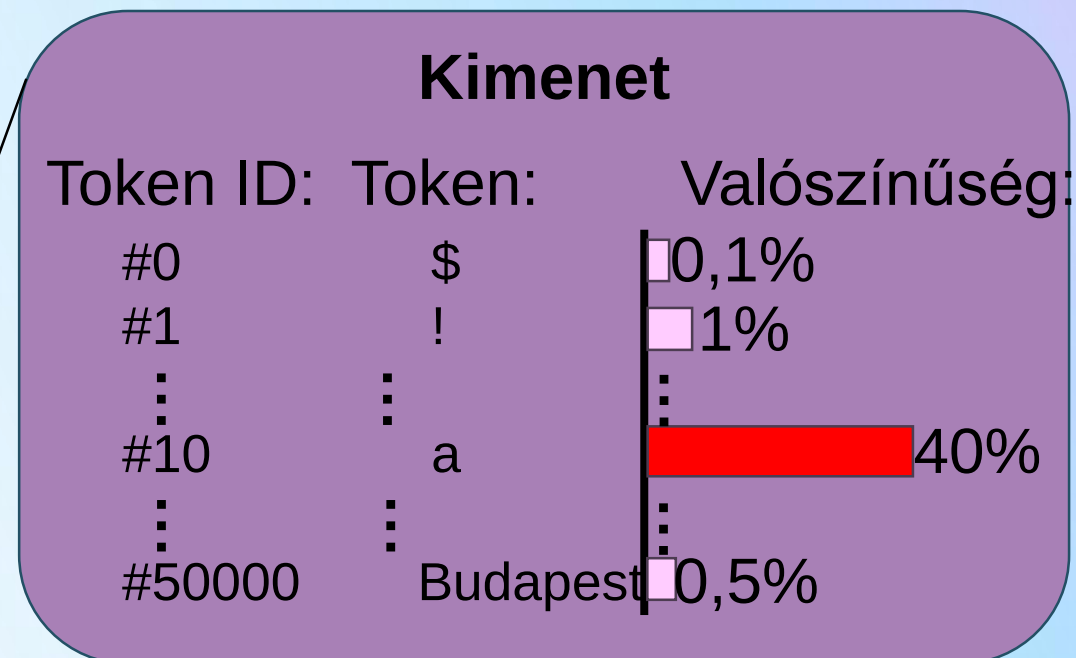
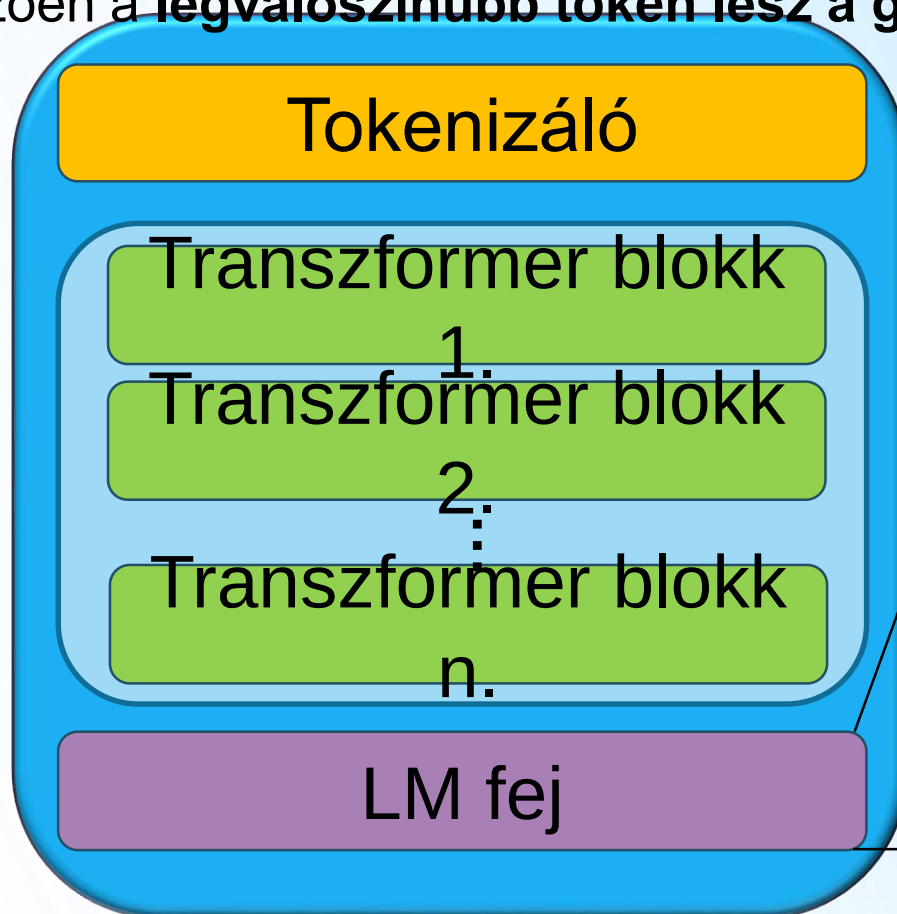


Transzformer alapú LLM működése

- **LM fej** feladata

- A kimenet következő tokenjének előállításához **meghatározza a token „szókincs” valószínűségi eloszlását**
- Jellemzően a **legvalószínűbb token lesz a generált kimenet következő eleme**

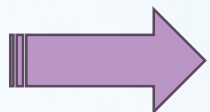
GPT



Hogyan tanul az LLM?

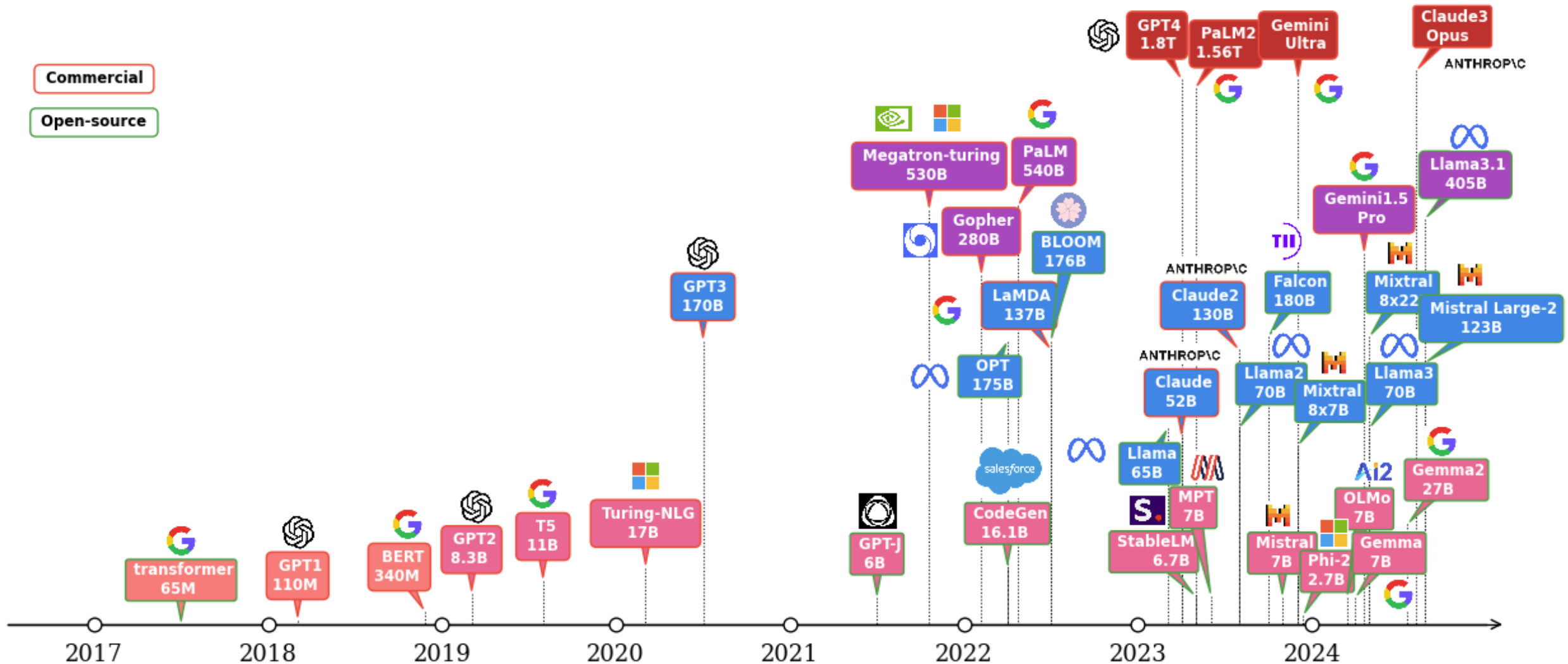
- Lényegében **a tanulás**
 - rengeteg bemeneti adat (szöveg) segítségével
 - **a modell paramétereinek** megfelelő **beállítása** a tanulási fázisban

Nem esik messze az ...



„alma”

LLM-ek evolúciója



Hugging Face

The screenshot shows the Hugging Face website interface. At the top, there is a navigation bar with the Hugging Face logo, a search bar labeled "Search models, datasets, users...", and links for Models, Datasets, Spaces, Buckets, Docs, Enterprise, Pricing, Log In, and Sign Up. The main content area features a large banner with the text "The AI community building the future." and a subtext "The platform where the machine learning community collaborates on models, datasets, and applications." Below this, there are buttons for "Explore AI Apps" and "Browse 2M+ models". On the right side, there is a sidebar with a "Models" tab selected, showing a count of 2,555,000 models. Below this, there are sections for Tasks (Text Generation, Image-Text-to-Text, Image-to-Image, Text-to-Image, Text-to-Video, Text-to-Speech), Parameters (a slider from <1B to >500B), Libraries (PyTorch, TensorFlow, JAX, Transformers, Diffusers, Safetensors, ONNX, GGUF, Transformers.js, MLX, MLX), Apps (vLLM, TGI, llama.cpp, MLX LM, LM Studio, Ollama, Jan), and Inference Providers (Groq, Novita, Cerebras, SambaNova, Nscale, fal, Hyperbolic, Together AI). The main list of models includes entries like moonshotai/Kimi-K2.5, openai/gpt-oss-120b, Lightricks/LTX-2, microsoft/VibeVoice-ASR, nvidia/personalex-7b-v1, MiniMaxAI/MiniMax-M2.1, Qwen/Qwen3-ASR-0.6B, deepseek-ai/DeepSeek-OCR-2, arcee-ai/Trinity-Large-Preview, and stepfun-ai/Step-3.5-Flash.

<https://huggingface.co/>

Tipikus hibák MI használatakor

- Az **LLM-ek** egyik fő feladata a **nyelvtanilag helyes**, felhasználó számára **elfogadható válasz generálása**
- Ezért a célért hajlamos a **modell hallucinálni**, vagy **hamis dolgokat tényként prezentálni**
 - Jelentős fejlődések ezen a téren, de teljesen nem tudták megszüntetni
 - Problémát jelent a sorrendezés, elemek számolása
- Főleg akkor igaz ez, amikor **kevésbé ismert tényekre kérdez az ember**
 - Példa: Ki volt az **ötödik ember a holdon?**
 - Válasz: Az ötödik férfi, aki a Holdra lépett, **Alan Bean** volt
 - Másik válasz: Az ötödik ember, aki a Holdra lépett, **James Irwin** volt
 - **Valójában** az ötödik ember a Holdon **Alan Shepard** volt

Tipikus hibák MI használatakor

Pictures ⇅	No. ⇅	Name ⇅	Mission ⇅	Date of Birth ⇅	Date of Death (Age at Death) ⇅	Lunar EVA dates (UTC) ^[2] ⇅	Lunar EVAs ^[2] ⇅	Total EVA duration ^[2] ⇅	Astronaut service ⇅
	1.	Neil Armstrong	Apollo 11	August 5, 1930 ^[3]	August 25, 2012 (aged 82)	July 21, 1969 ^[a]	1	2 hours 31 minutes	NASA ^[b]
	2.	Buzz Aldrin	Apollo 11	January 20, 1930 (age 96) ^[4]		July 21, 1969 ^[a]	1	2 hours 31 minutes	Air Force
	3.	Pete Conrad	Apollo 12	June 2, 1930 ^[5]	July 8, 1999 (aged 69)	November 19–20, 1969	2	7 hours 45 minutes	Navy
	4.	Alan Bean	Apollo 12	March 15, 1932 ^[6]	May 26, 2018 (aged 86)	November 19–20, 1969	2	7 hours 45 minutes	Navy
	5.	Alan Shepard	Apollo 14	November 18, 1923 ^[7]	July 21, 1998 (aged 74)	February 5–6, 1971	2	9 hours 21 minutes	Navy

MI hallucináció, hamis tények felismerése, kezelése

- Konkrét **tényeket** minden esetben **ellenőrizzünk le**
 - **Kérhetjük** a promptban **az adatok hivatkozását**
 - „Ki volt a hetedik ember a holdon? Add meg a hivatkozást az adatra!”
- Viszont a **hivatkozást is** fontos **leellenőrizni**
 - Az LLM néha a **forrást is hajlamos hallucinálni**
 - Még ha valódinak tűnik is, az **MI által generált forrás egyre gyakoribb**
- **Kérdezzünk rá többször az eredményre**, az esetleges kételyeinket is közölve

Kontextus hossz probléma

- A jelenlegi MI megvalósítások **token alapon dolgoznak**
 - **Minden**, amit felhasználnak a kimenet generálásához, **tokenből érkezik**
- Maximális **limit a felhasználható tokenek számára**
 - Például **Claude 4.6** modell – **1 millió token**
- **Nagyobb kód projekteknél** már **problémába futhatunk**, az LLM nem tudja a teljes környezetet „fejben” tartani
- Vagy hosszabb beszélgetések során **nem tud visszahivatkozni korábbi információra**
- **Megoldás:** A komplex **feladatot** először **bontsuk nagyobb blokkokra** MI segítségével
 - A **blokkok** külön már **kevesebb tokent igényelnek**
 - A **blokkok egymásra épülnek**, de alapvetően **külön egységként kezelendők**

Adatok felhasználása

- Az **MI tanításának** egyik legnagyobb, **legértékesebb része a „tisztá” adat**
 - Rengeteg munkába kerül az adathalmazok összeállítása
 - Weboldalak böngészése, letöltése, elmentése
 - Könyvek letöltése
 - Felhasználói beszélgetések elmentése
- **Felhasználó** által indított **promptok** talán a **legértékesebbek**
 - Közvetlenül látszódik, mire használja a felhasználó
 - A rendszer követi, hogy a felhasználó mennyire elégedett a válasszal
- Bármilyen **adat, amit megosztunk az MI-vel**, jó eséllyel **bekerül** egy ilyen **adatbázisba**
 - Anonimizált, de nem biztos, hogy akár kérésre törlésre kerül

Érzékeny adatok védelme

- Semmiképp **ne osszunk meg személyes vagy érzékeny adatot** az MI-vel való beszélgetés során
 - Céges projektek, kimutatások is érzékeny adat!
- Ha mégis muszáj, **általánosítsuk, fogalmazzuk át a problémát**
- Kapcsolódó fogalmak: **Privacy Settings, Data Privacy Policy**
 - Esetlegesen limitálható, mit használhatnak fel tanításra
- **Fizetős szolgáltatásnál** a szolgáltató vállalja, hogy **csak a feldolgozásnál használja az adatokat**, azok nem kerülnek mentésre
- **Tervezet MI szabályozására**
 - EU: AI Act
 - Bizonyos MI felhasználás esetén kötelező a tanulási adatok forrásának publikálása

Áttekintés

- **Generatív modellek és alkalmazásaik**
 - Generatív modellek jelentése, működési alapelvek
 - Nagy nyelvi modellek (LLM)
 - Tipikus hibák az LLM használatakor
 - Adatok védelme, privát adatok
 - **Promptolás**
 - Promptolási minták használata a gyakorlatban
 - Promptolási példafeladatok különböző szakterületeket érintve
 - Promptolási példafeladatok multimédia tartalom generálására
 - Promptolási példafeladatok programozásra



Promptolás – Az MI-vel való kommunikáció, mint új kompetencia

A kérdezéstől az együttműködésig

- Gyakorlati készségfejlesztés
- Promptolási minták elsajátítása
- Kritikus gondolkodás erősítése
- Interdiszciplináris alkalmazások

Ma már nem elég jó kérdést feltenni – meg kell tanulni együttműködni az MI-vel



A jó prompt építőkövei

5 alapvető elem:

1. **Perszona** – Ki legyen az MI? (pl. „Te egy történelemtanár vagy”)
2. **Feladat** – Mit csináljon? (Egyértelmű, cselekvő ige)
3. **Kontextus** – Miért és kinek? (Háttér, célközönség)
4. **Korlátozások** – Milyen határok? (Hossz, hangnem, tiltott témák)
5. **Formátum** – Milyen formában? (Táblázat, lista, JSON)



A „kérdéstől a meta-promptolásig” skála

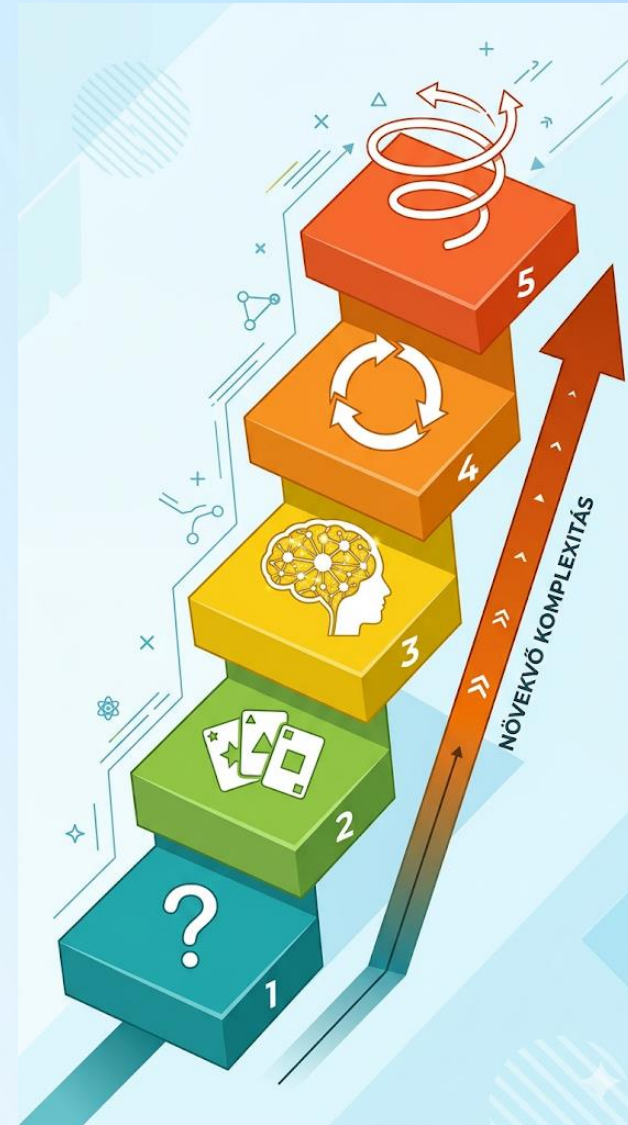
1. szint: Standard promptolás
Egyszerű kérdés vagy utasítás

2. szint: Few-Shot promptolás
2-3 példa a kérdés előtt

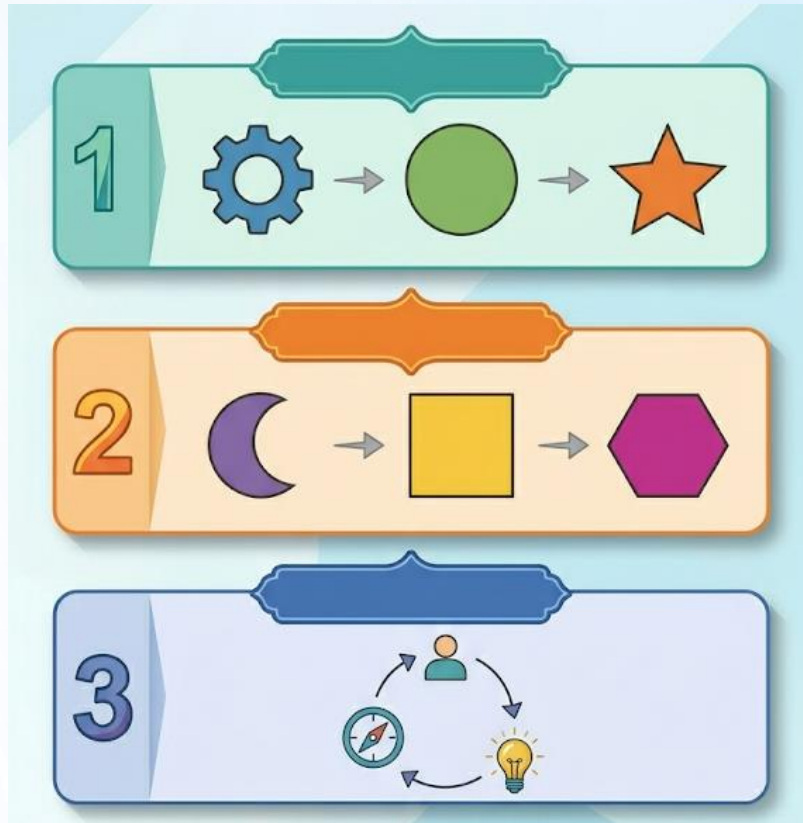
3. szint: Chain-of-Thought
(CoT)
„Gondolkodj lépésről lépésre”

4. szint: Iteratív finomítás
MI kritizálja a saját válaszát

5. szint: Meta-promptolás
MI írja meg a legjobb promptot



Kezdő szintek: Standard és Few-Shot



1. szint – Standard promptolás:

> „Sorold fel a Naprendszer bolygóit.”

2. szint – Few-Shot promptolás:

> „Klasszifikálj állatokat:

Példa 1: Delfin → Emlős → Vízi

Példa 2: Sas → Madár → Szárnyas

Most te: Cápa → ?”

Eredmény: A példák után pontosabb és konzisztensebb válaszok!

3. szint: Chain of Thought – „Gondolkodj lépésről lépésre”



Mikor használjuk?

- Matematikai problémák
- Logikai fejtörők
- Többlépéses fizikai példák

Példa fizikára:

„Egy autó 0-ról 100 km/h-ra gyorsul 8 mp alatt. Oldd meg lépésről lépésre:

1. Írd fel a képleteket!
2. Alakítsd át a mértékegységeket!
3. Számold ki!
4. Ellenőrizd!”

Eredmény: Kevesebb hiba, átlátható érvelés!

4 – 5. szint: Önreflexió és meta-promptolás

4. szint – Iteratív finomítás:

„Írtál egy fogalmazást. Most kritizáld: mi a gyenge benne? Javítsd!”

5. szint – Meta-promptolás:

„Milyen promptot kellene írnom, hogy a legjobb esszét kapjam a klímaváltozásról 500 szóban?”

Plusz extra:

MI értékelje a promptod 1-10 skálán és adjon javítási tippeket!



Promptolási minták áttekintése

Perszona minta:

„Ki vagy?” → Hangnem, stílus, szakmai zsargon

Few-Shot minta:

„Kövesd a példám” → Adatformázás, osztályozás

„Chain of Thought” minta:

„Lassíts és gondolkodj” → Logika, matematika

Mikor melyiket?

Perszona → Stílus kell

Few-Shot → Formátum kell

CoT → Érvelés kell

Korlátozás / negatív minta:

„Maradj a határokon belül” → Szigorú követelmények

Példa: „Foglald össze Petőfit 50 szóban, a *forradalom* szó nélkül”

„Chain of Density” minta:

„Jobbat, ne többet” → Információsűrűség növelése

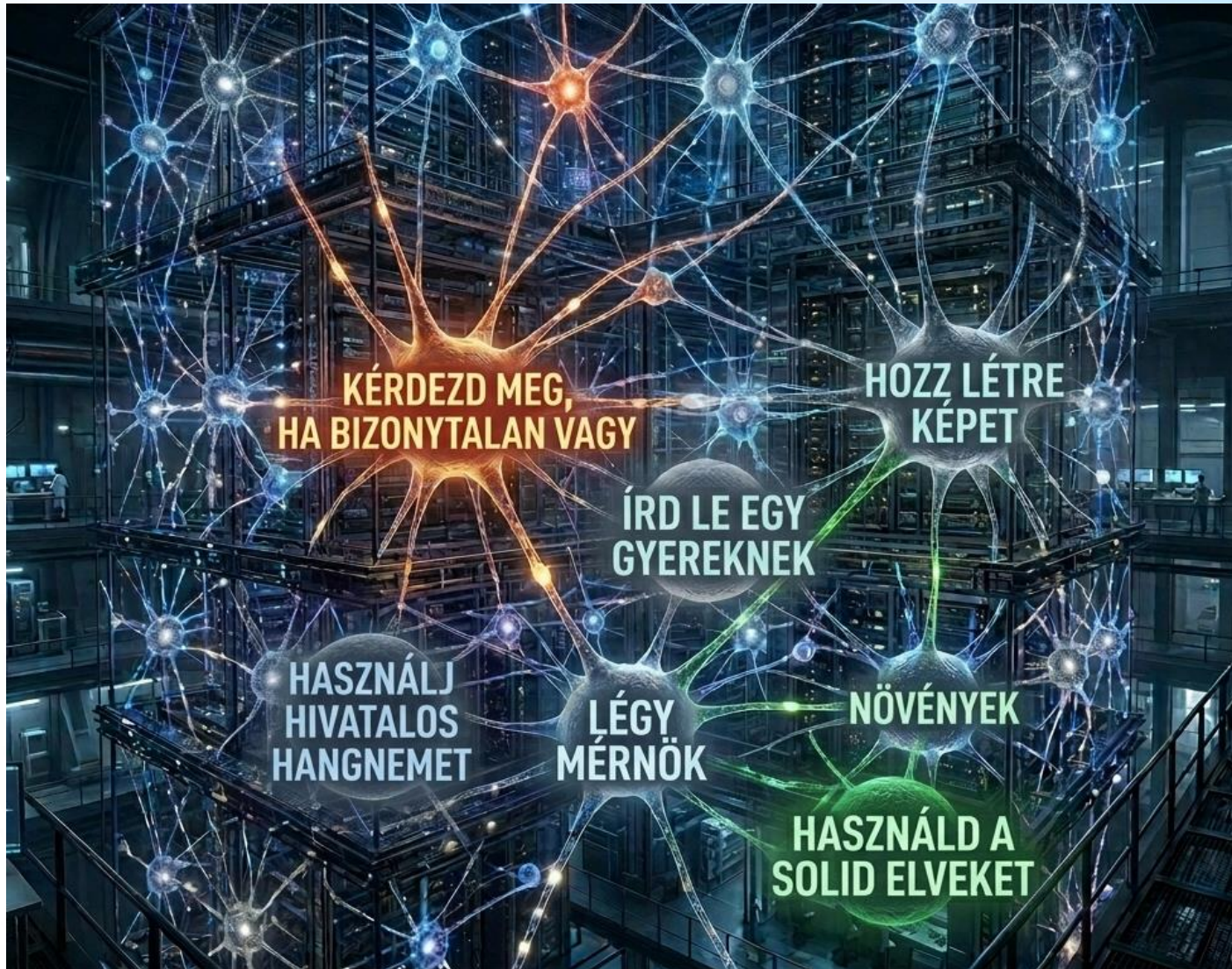
Példa: „Írd át ezt 150 szóból 100-ba, de több ténnyel!”

Hogyan kezdjük el? Gyakorlati tippek

1. **Kezdjük egyszerűen** → Standard prompt, aztán finomítsuk! Rögtön kérhetünk pontosító kérdéseket!
 2. **Használjunk példákat** → Few-Shot mindig segít
 3. **Kérjünk lépéseket** → CoT összetett problémáknál
 4. **Iteráljunk** → Első válasz ritkán tökéletes
 5. **Értékeltezzünk** → „Pontozd 1-10-ig és javítsd!”
- Kísérletezzünk! 😊



A kontextus fontossága



Az **MI** sem mindig **gondol** mindenre

Amit **nem** fejtünk ki, annál egyfajta **átlagos igényt** feltételez

Egyes kifejezések egész **viselkedési mintákat** képesek **aktiválni**

Áttekintés

- **Generatív modellek és alkalmazásaik**

- Generatív modellek jelentése, működési alapelvek
- Nagy nyelvi modellek (LLM)
- Tipikus hibák az LLM használatakor
- Adatok védelme, privát adatok
- Promptolás
- **Promptolási minták használata a gyakorlatban**
- Promptolási példafeladatok különböző szakterületeket érintve
- Promptolási példafeladatok multimédia tartalom generálására
- Promptolási példafeladatok programozásra



Perszona minta – „Te vagy a kalóz professzor!”

Koncepció:

Specifikus szerep, háttér és hangnem kiosztása.

Feladat:

Hozzunk létre három különböző „oktatót” egy nehéz conceptushoz (pl. Fotoszintézis)—egy Kalóz perszonát, egy Professzort és egy „ELI5” (Magyarázd el, mintha ötéves lennék) specialistát!

Felismerés:

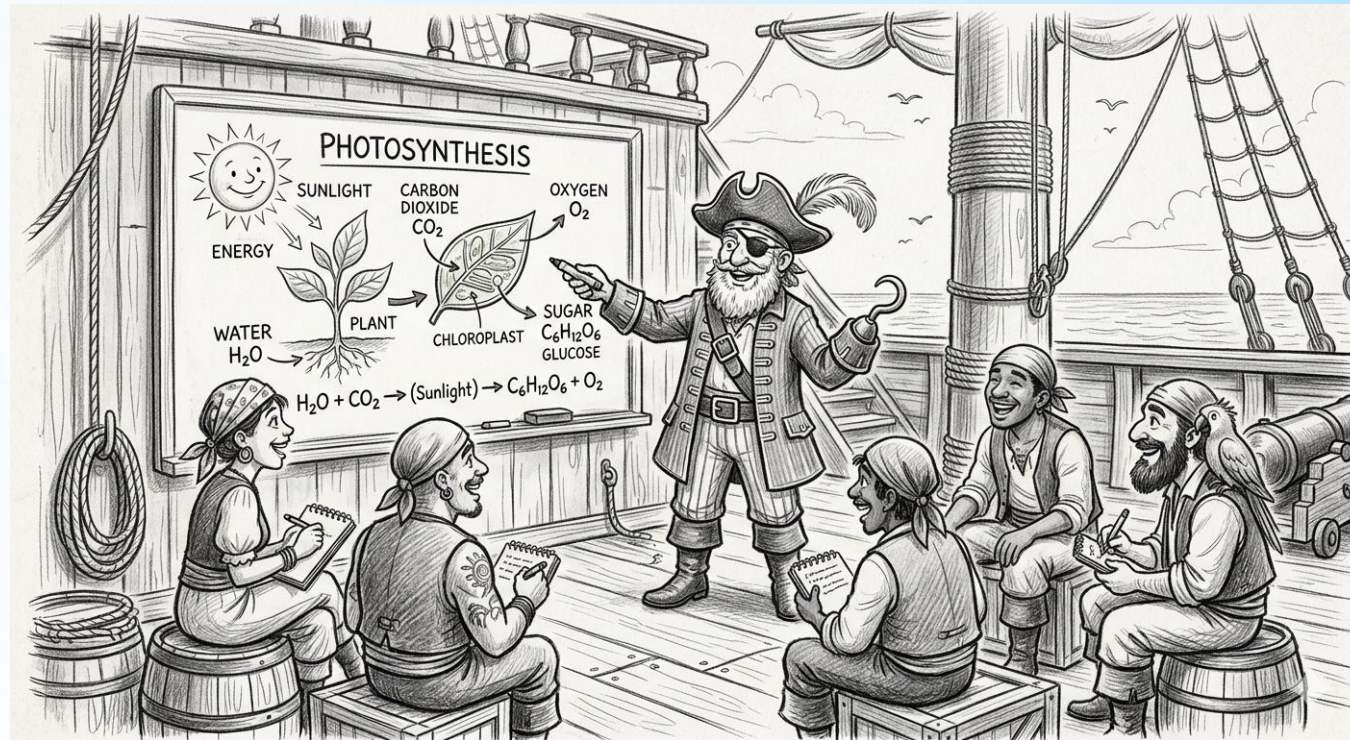
Figyeljük meg, hogyan változik a szóhasználat a perszona korlátozás alapján!

Várható eredmények:

- Adaptivitás – Ugyanaz a tartalom különböző közönségeknek különböző módon
- Hangnem kontroll – Formális kontra informális, egyszerű kontra szakmai nyelv
- Célközönség tudatosítás – Ki fogja olvasni / hallani ezt?
- Kreativitás a tanításban – Unalmas témák érdekessé tétele perszonával

Perszona minta – „Te vagy a kalóz professzor!” – 1. példa

„Te egy kalóz kapitány vagy, Félszemű Jack, aki a legénységet oktatom biológiára a fedélzeten. Magyarázd el a fotoszintézist tengeri terminológiával, kalózsargonnal és hajózási metaforákkal! Használj kifejezéseket mint 'kincs gyártás', 'napfény zsákmány', 'energia kereskedelem'. Tedd kalandossá és izgatóvá, de maradj tudományosan pontos a folyamat lépéseiben (fényreakció, sötétreakció, glükóztermelés). Maximum 150 szóban, kalóz dialektussal.”



Perszona minta – „Te vagy a kalóz professzor!” – 1. példa

Ide hallgassatok, ti hínáros képű matrózok! Ma nem aranyat rabolunk, hanem a Nap tűzét csapoljuk meg, akár a növények! Ez a **fotoszintézis**, a legnagyobb kincsgyártás a hét tengeren!

A hajónk, a *Kloroplasztisz*, kifeszíti zöld vitorláit. Első lépés a **napfény-zsákmány**: a fényreakció során a legénység elkapja a fotonokat, szétzúzza a vizet, és friss oxigént ereget a szélbe, mint ágyúfüstöt. Ebből nyerjük az ATP-t, a mi fényes aranyunkat!

De a kaland itt nem ér véget! Jön a **Calvin-ciklus**, a sötétreakció, ahol nincs szükség lámpásra. Itt folyik az igazi **energia-kereskedelem**: a levegőből elrabolt szén-dioxidot összeütjük a fény-aranyunkkal. A kapitányi konyhán ebből főzzük a **glükózt**, azt az édes szirupot, ami a növény-hajót hajtja! Ez a cukor a mi igazi zsákmányunk, amitől nő a törzs és erősödik a gyökér-horgony.

Vigyázzátok a zöldet, mert ha nincs kincsgyártás, éhen döglik a legénység!

Perszona minta – 2. példa

„Te Dr. Kovács István professzor vagy, az ELTE Biológiai Intézetének vezetője, és MSc hallgatóknak tartasz előadást. Magyarázd el a fotoszintézis folyamatát formális akadémiai nyelven, pontos tudományos terminológiával! Részletezd a fényfüggő és fényfüggetlen reakciókat, a tilakoid membránban zajló elektrontranszport láncot, és a Calvin-ciklust! Használd a szakmai nomenklatúrát (PSII, PSI, ATP-szintáz, RuBisCO)! Feltételezd, hogy a hallgatóság ismeri a biokémiát! Strukturáld: Bevezetés, Fényfázis, Sötétfázis, Összegzés.”

Perszona minta – 2. példa

Tisztelt Hallgatóság! A mai előadásunk fókuszában a bioszféra primer produkciójának fundamentális folyamata, a fotoautotróf asszimiláció áll. Tekintsük át a kloroplasztiszban zajló energetikai és biokémiai transzformációk precíz mechanizmusát.

Bevezetés

A fotoszintézis egy komplex redox-folyamat, amely során a fényenergia kémiai energiává (ATP és *NADPH*) alakul, majd ezen ekvivalensek felhasználásával a szén-dioxid szerves anyaggá redukálódik. A folyamat két szorosan összekapcsolt, de térben és funkcionálisan elkülönülő szakaszra osztható: a tilakoid membránhoz kötött fotokémiai reakciókra és a sztrómában zajló enzimatisz folyamatokra.

Fényfázis: Az energia transzdukciója

A fényfüggő szakasz a kloroplasztisz tilakoid membránjában lokalizálódik. A folyamat a **II. fotorendszer (PSII)** P680-as reakcióközpontjának gerjesztésével indul.

1. **Vízi fotolízis:** A PSII oxigéntermelő komplexe a vizet protonokra, elektronokra és molekuláris oxigénre bontja. Az elektronok pótolják a gerjesztett klorofill-a hiányzó elektronjait.

Few-Shot minta – „Kövesd a példámat!”

Koncepció:

2-5 példa megadása a kívánt formátumra a végső kérdés előtt.

Feladat:

Építsünk egy „Hangulat Elemzőt"! Adjunk meg három példát: [Idézet] -> [Beszélő] -> [Hangnem]! Ezután teszteljük az MI pontosságát egy új idézeten a példákkal és anélkül is!

Felismerés:

Megtanítja, hogy a mintaillesztés gyakran jobb, mint a hosszú leírások.

Várható eredmények:

- Formátum konzisztencia – Strukturált kimenetek (táblázatok, listák, JSON)
- Példa > Szabály – Gyakran könnyebb példát mutatni, mint elmagyarázni a szabályt
- Osztályozási feladatok – Sentimentanalízis, kategorizálás, címkézés
- Összehasonlítás – Zero-Shot kontra Few-Shot eredmények különbsége

Few-Shot minta – „Kövesd a példámat!” – 1. példa

„Elemezd irodalmi idézetek hangulatát PONTOSAN ebben a formátumban (nyíl jellel és pontosvesszővel):

Idézet: 'Talpra magyar, hí a haza!' -> Beszélő: Petőfi Sándor -> Hangnem: Felhívó, hazafias, szenvedélyes

Idézet: 'Egy szikra lobbant föl kebelemben' -> Beszélő: Vörösmarty Mihály -> Hangnem: Romantikus, érzelmes, lírai

Idézet: 'Itt a vész! Most állj meg, magyar!' -> Beszélő: Kölcsey Ferenc -> Hangnem: Figyelmeztető, komoly, aggó

Most elemezd UGYANEbben A FORMÁTUMBAN a következő idézetet, mindenféle egyéb magyarázat nélkül:
'A magyar hazának hű fiai legyetek''

Válasz:

Idézet: 'A magyar hazának hű fiai legyetek' -> Beszélő: Batthyány Lajos

-> Hangnem: Hazafias, buzdító, méltóságteljes

Few-Shot minta – „Kövesd a példámat!” – 2. példa

„Te egy ügyfélszolgálati MI vagy. Osztályozd az ügyfélvisszajelzéseket PONTOSAN ebben a struktúrában:

Visszajelzés: 'A termék sérülten érkezett' -> Kategória: Panasz -> Prioritás: Magas -> Javasolt intézkedés: Azonnali visszaküldés és csere

Visszajelzés: 'Gyors szállítás, kiváló minőség' -> Kategória: Dicséret -> Prioritás: Alacsony -> Javasolt intézkedés: Köszönőlevél küldése

Visszajelzés: 'Hogyan tudom visszaküldeni?' -> Kategória: Kérdés -> Prioritás: Közepes -> Javasolt intézkedés: Visszaküldési útmutató elküldése

Most osztályozd UGYANEKBEN A FORMÁTUMBAN mindenféle extra magyarázat nélkül:

'A termék működik, de a leírás zavaros volt, és nem azt kaptam, amit vártam''

Válasz:

Visszajelzés: 'A termék működik, de a leírás zavaros volt, és nem azt kaptam, amit vártam' -> Kategória: Panasz -> Prioritás: Közepes

-> Javasolt intézkedés: Termékismertető pontosítása és elnézést kérő válasz küldés

„Chain of Thought” minta – „Lassíts és gondolkodj!”

Koncepció:

Az MI kényszerítése, hogy lépésről lépésre „gondolkodjon” a végső válasz megadása előtt.

Feladat:

Oldjunk meg egy többlépéses fizikai szöveges feladatot! Hasonlítsuk össze a „közvetlen válasz” promptot egy olyan prompttal, amely megköveteli a képletek és köztes lépések előzetes megadását.

Felismerés:

Demonstrálja, hogyan javítja a „piszkozat” tér az LLM érvelését.

Várható eredmények:

- Hibák csökkentése – Matematikai és logikai feladatoknál 30-50% javulás
- Átlátható érvelés – Követhető gondolatsor, ellenőrizhető lépések
- Debuggolás – Ha hiba van, látszik, melyik lépésben
- Pedagógiai érték – A diákok is látják a megoldás menetét, nem csak az eredményt

„Chain of Thought” minta – „Lassíts és gondolkodj!” – 1. példa

„Egy autó 0-ról 100 km/h-ra gyorsul 8 másodperc alatt egyenletesen, majd ezt a sebességet tartja további 10 másodpercig, végül 5 másodperc alatt egyenletesen lefékeződik 0 km/h-ra. Számold ki a teljes megtett út hosszát méterben!

FONTOS: Oldd meg lépésről lépésre az alábbi struktúrát követve, és mutass meg MINDEN számítási lépést:

1. LÉPÉS – Releváns képletek azonosítása:

Írd fel a szükséges kinematikai egyenleteket:

- Egyenletes gyorsulás esetén megtett út
- Egyenletes mozgás esetén megtett út
- Egyenletes lassulás esetén megtett út

2. LÉPÉS 2 – Egységek átalakítása:

Alakítsd át a sebességet km/h-ról m/s-ra ($100 \text{ km/h} = ? \text{ m/s}$)!

Indokold az átalakítás szükségességét!

3. LÉPÉS – Szakaszokénti számítás:

A) Gyorsulási szakasz (0-8 mp): Számold ki a gyorsulást, majd a megtett utat

B) Egyenletes szakasz (8-18 mp): Számold ki a megtett utat

C) Lassulási szakasz (18-23 mp): Számold ki a lassulást, majd a megtett utat

4. LÉPÉS – Összegzés:

Add össze a három szakasz útvonalának hosszát!

Írd le a végeredményt mértékegységgel!

5. LÉPÉS 5 – Ellenőrzés:

Értékelj, hogy az eredmény fizikailag értelmes-e (realisztikus-e a távolság, ha autóval tesszük meg)!

Minden lépésnél INDOKOLD az érvelésed!"

„Chain of Thought” minta – „Lassíts és gondolkodj!” – 1. példa

Íme a fizikai feladat részletes, lépésről lépésre kidolgozott megoldása:

LÉPÉS 1 - Releváns képletek azonosítása

A feladat három különböző mozgásszakaszból áll, melyekhez az alábbi kinematikai egyenleteket használjuk:

- **Egyenletesen gyorsuló és lassuló mozgás útja:** Ha a kezdősebesség vagy a végsebesség nulla, az út kiszámítható az átlagsebesség és az idő szorzataként, vagy a gyorsulós képlettel:

$$s = v_{\text{átlag}} \cdot t = \frac{v_{\text{kezdő}} + v_{\text{vég}}}{2} \cdot t$$

Vagy a gyorsulás (a) ismeretében: $s = v_0 \cdot t + \frac{a}{2} \cdot t^2$.

- **Egyenletes mozgás (állandó sebesség) útja:**

$$s = v \cdot t$$

- **Gyorsulás meghatározása:**

$$a = \frac{\Delta v}{\Delta t} = \frac{v_{\text{vég}} - v_{\text{kezdő}}}{t}$$

LÉPÉS 2 - Egységek átalakítása

A fizikai számításoknál alapvető az SI mértékegységrendszer használata. Mivel az idő

„Chain of Thought” minta – „Lassíts és gondolkodj!” – 2. példa

„Egy tárgyat 125 méter magas épület tetejéről szabadon ejtenek le (kezdősebesség nulla). Számítsd ki, mennyi idő alatt éri el a földet! Léggöri ellenállástól eltekintünk, $g = 10 \text{ m/s}^2$ gravitációs gyorsulással számolj!

Kötelezően STRUKTURÁLT lépésenkénti megoldás:

--- 1. LÉPÉS: Releváns fizikai egyenlet felírása ---

Írd fel a szabadon eső test útjának képletét:

$s = \dots$

Definiálj minden jelölést (s , v_0 , g , t jelentése)

--- 2. LÉPÉS: Ismert és ismeretlen változók azonosítása ---

ISMERT adatok:

-

ISMERETLEN:

-

Írd fel táblázatos formában!

--- 3. LÉPÉS: Egyenlet átrendezése ---

Fejezd ki az ismeretlen változót (t) az egyenletből

Mutasd meg az algebrai lépéseket sorra:

$s = \dots$

$\Rightarrow \dots$

$\Rightarrow t = \dots$

--- 4. LÉPÉS: Számértékek behelyettesítése ---

Helyettesítsd be az ismert értékeket az egyenletbe!

Írd fel a számításokat lépésről lépésre:

$t = \sqrt{2 \times \dots / \dots}$

$t = \sqrt{\dots}$

$t = \dots$

--- 5. LÉPÉS: Fizikai értelmezés és ellenőrzés ---

a) Az eredmény mértékegysége helyes? (ellenőrizd a dimenziókból)

b) Az eredmény realisztikus értéket ad? (körülbelül hány másodperc kellene hozzá intuitív módon?)

c) Mi történne magasabb épületnél? Arányos-e lineárisan a magassággal az idő?

Minden lépést számozott formában, egyenletekkel együtt írd le!"

„Chain of Thought” minta – „Lassíts és gondolkodj!” – 2. példa

Íme a szabadesés kinematikai feladatának részletes, strukturált megoldása a kért lépések szerint:

--- LÉPÉS 1: Releváns fizikai egyenlet felírása ---

A szabadesés egy kezdősebesség nélküli ($v_0 = 0$), egyenletesen gyorsuló mozgás, ahol a gyorsulás a nehézségi gyorsulás (g).

Az út összefüggése az idővel:

$$s = \frac{g}{2} \cdot t^2$$

Jelölések jelentése:

- s : a megtett út (vagy elmozdulás), méterben (m).
- v_0 : kezdősebesség, méter/másodpercben (m/s). Jelen esetben 0.
- g : nehézségi (gravitációs) gyorsulás, m/s^2 -ben.
- t : az esés időtartama, másodpercben (s).

Korlátozás / negatív minta – „Maradj a kereten belül!”

Koncepció:

Szigorú határok használata (pl. „Ne használd...”, „Maximum 50 szó”).

Feladat:

Foglaljuk össze *Madách Imre Az ember tragédiáját* vagy *Katona József Bánk bánját* a főszereplők nevének vagy a „tragédia” szó használata nélkül!

Felismerés:

Megtanítja, hogyan kell határolni a kimeneti teret professzionális vagy biztonság tudatos alkalmazásokhoz.

Várható eredmények:

- Kreatív korlátozás – A határok kreativitást szülnek
- Valós felhasználási esetek – Karakterlimit (Twitter, SMS), tiltott szavak (cenzúra, szakzsargon)
- Alternatív kifejezés mód – Ugyanaz a tartalom más megfogalmazással
- Tartalomfilterezés – Elkerülni érzékeny / nem megfelelő kifejezéseket

Korlátozás / negatív minta – „Maradj a kereten belül!” – 1. példa

„Foglald össze Madách Imre 'Az ember tragédiája' című drámájának cselekményét és főbb filozófiai témáit pontosan 100 szóban (± 2 szó eltérés megengedett)!

=== SZIGORÚ KORLÁTOZÁSOK ===

TILTOTT NEVEK (ezeket NE használd): Ádám, Éva, Lucifer, Az Úr

TILTOTT KULCSSZAVAK (ezeket NE használd): 'tragédia', 'álom', 'történelem', 'színháték'

=== ENGEDÉLYEZETT HELYETTESÍTÉSEK ===

Használd ehelyett:

- 'az első ember' vagy 'a főhős'
- 'a női szereplő' vagy 'a társa'
- 'a kísértő' vagy 'a szkeptikus hang'
- 'teremtő' vagy 'isteni alak'

=== TARTALMI FÓKUSZ ===

Összpontosíts:

- Az egyes történelmi stációk sorozatára (nevük nélkül: ókori birodalom, középkori város, stb.)
- A filozófiai kérdésekre: emberi törekvés értelme, haladás vs. pesszimizmus, szabadakarat
- A darab drámai ívére és befejezésére

Számláló: úgy kezeld, hogy PONTOSAN 100 szót KELL tartalmaznia (se többet, se kevesebbet 2-nél!)"

Korlátozás / negatív minta – „Maradj a kereten belül!” – 1. példa

A mű kezdetén a szkeptikus hang szembeszáll a teremtő hatalmával. A főhős és a női szereplő kiűzetése után látomásokon keresztül tapasztalják meg a jövő súlyos korszakait. Megjárják az ókori zsarnokságot, a középkor vallási feszültségeit és a felvilágosodás káoszát, végül a sivár falansztert és az űrt is. A kísértő célja a reményvesztés, hiszen a haladás eszméje minden korban elbukik. Az út végén a főhős az öngyilkosságba menekülne, ám a társa anyasága ezt megakadályozza. Az isteni alak zárómondata biztatást ad: a küzdés maga az élet értelme. A szabadakarat és hit harca így örök körforgás marad számunkra.

Korlátozás / negatív minta – „Maradj a kereten belül!” – 2. példa

„Írd le Katona József 'Bánk bán' című drámájának központi témáit és konfliktusait maximum 75 szóban (szigorú limit!).

=== TELJES TILTÓLISTA ===

KARAKTERNEVEK - egyik sem használható: Bánk / Bánk bán, Melinda / Melindát, Gertrudis / Gertrud királyné, Tiborc, II. Endre / a király, Ottó, Petur bán, Mikhál

TILTOTT FOGALMAK: 'gyilkosság' / 'meggyilkol', 'tragédia' / 'tragikus', 'királyné', 'dráma', 'szerelem' (mint romantikus)

=== KÖTELEZŐ HELYETTESÍTÉSI STRATÉGIA ===

Karakterek helyett használj SZEREPKÖRÖKET: 'a hazafi nemes' (Bánk), 'a hűséges feleség' (Melinda), 'az idegen származású uralkodó asszony' (Gertrudis), 'a jobbágy paraszti képviselő' (Tiborc), 'a távollévő király' (II. Endre)

ESEMÉNYEK helyett használj ABSZTRAKT TÉMÁKAT: 'hűség és árulás konfliktusa', 'nemzeti érdek vs. idegen befolyás', 'becsület védelmének dilemmája', 'társadalmi igazságtalanság'

=== TARTALMI ELVÁRÁS ===

Elemezd:

1. A fő morális dilemmát
2. A politikai (nemzeti-idegen) feszültséget
3. A társadalmi kritikát (nemesség-paraszság)
4. Az egyéni vs. közösségi érdek ütközését

Hangnem: objektív, elemző, NE szubjektív vagy érzelmes!

Szószám: pontosan számold, max 75!"

Korlátozás / negatív minta – „Maradj a kereten belül!” – 2. példa

A mű központi konfliktusa a nemzeti érdek és az idegen befolyás ütközése. A hazafi nemes morális dilemmája a közösségi hűség és az egyéni becsület védelme között feszül. Az idegen származású uralkodó asszony hatalma társadalmi igazságtalanságot szül, amit a jobbágy paraszti képviselő panasza mélyít el. A távollévő király hiánya politikai válságot okoz. A hűséges feleség beszennyezett tisztessége és a nemesi ellenállás végül az egyéni sorsok és a jogrend felbomlásához vezet.

„Chain of Density” minta – „Jobbat, nem többet!”

Koncepció:

Iteratív módon növeli a szöveg információsűrűségét anélkül, hogy növelné a hosszát.

Feladat:

Kezdjünk egy 200 szavas összefoglalóval! Ismételten kérjük az MI-t, hogy írja át 150 szóban, miközben több konkrét tényt és entitást adunk hozzá!

Felismerés:

Demonstrálja a token hatékonyságot és az információsztézist.

Várható eredmény:

- Tömör írás – Több információ kevesebb szóban
- Újságírói készség – Lényegre törő, tényekkel teli szövegek
- Absztrakció kontra konkrétum – Az általános kifejezések cseréje specifikus kifejezésekre
- Iteratív finomítás – Többkörös javítási folyamat

„Chain of Density” minta – „Jobbat, nem többet!” – 1. példa

„Itt van egy összefoglaló a nándorfehérvári győzelemről: 'A 15. században jelentős katonai siker született. Magyar csapatok fontos támaszpontot védtek meg az oszmán haderővel szemben. A védekezésben katonai vezetők és egyházi személyek is részt vettek. A csata döntő fordulópontot jelentett a térségben. A győzelem sokáig emlékezetes maradt Európában.' Írd át pontosan 150 szóban, miközben több konkrét dátumot, nevet, számadatot és helyszínt adsz hozzá. Növeld az információsűrűséget: cseréld az általános kifejezéseket specifikus entitásokra."

„Chain of Density” minta – „Jobbat, nem többet!” – 1. példa

A **Hunyadi János** vezette magyar seregek **1456. július 4. és 22.** között sorsdöntő győzelmet arattak a mai Belgrád területén. **II. Mehmed** szultán **70-100 000** fős oszmán hadereje július elején vette ostrom alá a kulcsfontosságú várat. A védelemben részt vett **Szilágyi Mihály 7000** katonája, míg a felmentő sereg magvát Hunyadi zsoldosai és **Kapisztrán János** körülbelül **30-35 000** fős, keresztes népfelkelő csapata alkotta.

A fordulatot a július 21-i roham hozta el, amikor **Dugovics Titusz** önfeláldozó tette jelképezte a kitartást. Másnap Kapisztrán fegyelmezetlen, de lelkes egységei váratlanul átkeltek a Száván, hátba támadva az oszmán tüzérséget. Ez a manőver megfutamította a török fősereget, akik hátrahagyták felszerelésüket. A diadal emlékére **III. Kallixtusz** pápa elrendelte a déli harangszót, a győzelem pedig közel **70** évre megállította az Oszmán Birodalom európai terjeszkedését. A csatát követő pestisjárványban Hunyadi és Kapisztrán is életét vesztette, de nevük örökre egybeforrt a kereszténység védelmével.

(141 szó)

„Chain of Density” minta – „Jobbat, nem többet!” – 2. példa

„A magyar tudománytörténet világraszóló teljesítményét jelzi a Nobel-díjasok sora, akik kutatásaikkal alapjaiban formálták át a modern fizikát, kémiát és orvostudományt. Az első magyar származású díjazott **Lénárd Fülöp** volt, aki **1905-ben** kapott fizikai Nobel-díjat a katódsugaras vizsgálataiért. Őt követte **Bárány Róbert**, aki **1914-ben** az egyensúlyszerv fiziológiájának feltárásáért részesült orvosi elismerésben.

A hazai kutatás egyik legfényesebb fejezete **Szent-Györgyi Albert** nevéhez fűződik, aki a Szegedi Tudományegyetem professzoraként, **1937-ben** nyerte el a díjat a C-vitamin izolálásáért. A nukleáris medicina alapjait **Hevesy György** fektette le, aki **1943-ban** kapott kémiai Nobel-díjat az izotópos nyomjelzés kidolgozásáért. A technológiai fejlődést meghatározta **Gábor Dénes** holográfiája (**1971**) és **Oláh György** karbokation-kutatása (**1994**).

A magyar zsenialitás folytonosságát bizonyítja a **2023-as** év, amikor két újabb tudósunk ért fel a csúcsra: **Karikó Katalin** az mRNS-alapú vakcinatechnológiáért orvosi, míg **Krausz Ferenc** az attofizikai mérésekért fizikai Nobel-díjat kapott. Ezek a felfedezések nemcsak a múlt dicsőségét hirdetik, hanem a jövő orvoslásának és energetikájának is irányt mutatnak.”

Ez a 180 szavas cikk magyar Nobel-díjasokról szól. Írd át 60 szóban, de tartalmazza KÖTELEZŐEN: minimum 5 díjazott teljes nevét, pontos évszámokat (mikor kapták a díjat), tudományterületeket, konkrét felfedezéseket/eredményeket, és legalább 2 díjazott esetében azt is, melyik intézményben dolgoztak. Ne használj általános kifejezéseket, csak konkrét adatokat és neveket.

„Chain of Density” minta – „Jobbat, nem többet!” – 2. példa

Lénárd Fülöp 1905-ben fizikai Nobel-díjat kapott katódsugár-kutatásaiért. **Szent-Györgyi Albert**, a Szegedi Tudományegyetem professzora, **1937-ben** orvosi díjat nyert C-vitamin izolálásáért. **Hevesy György 1943-ban** kémiai elismerést kapott izotópos nyomjelzésért. **Oláh György**, a Dél-kaliforniai Egyetem kutatója, **1994-ben** kémiai díjas lett karbokation-kutatásaiért. **2023-ban Karikó Katalin** mRNS-vakcinákért orvosi, míg **Krausz Ferenc** attofizikai méréseiért fizikai Nobel-díjat kapott.

(51 szó)

Áttekintés

- **Generatív modellek és alkalmazásaik**
 - Generatív modellek jelentése, működési alapelvek
 - Nagy nyelvi modellek (LLM)
 - Tipikus hibák az LLM használatakor
 - Adatok védelme, privát adatok
 - Promptolás
 - Promptolási minták használata a gyakorlatban
 - **Promptolási példafeladatok különböző szakterületeket érintve**
 - Promptolási példafeladatok multimédia tartalom generálására
 - Promptolási példafeladatok programozásra



Feladat – Történelmi interjú készítése

Feladat:

Promptoljuk az LLM-et, hogy egy konkrét történelmi személyként viselkedjen (pl. Széchenyi vagy Kossuth)!

Informatikai fókusz:

Iteratív promptolás – A diákoknak finomítaniuk kell a „rendszer promptot”, hogy az tartalmazzon specifikus korlátozásokat, mint például 19. századi szókincs vagy utalások konkrét reformokra, eseményekre a szabadságharcból.

Kritikus gondolkodás:

Hasonlítsuk össze az MI válaszokat a tankönyvekkel, hogy kiszűrjük a hallucinációkat!

Várható eredmények:

- Történelmi empátia – A diákok egy történelmi személy szemszögéből gondolkodnak
- Kontextus fontossága – Megtanulják, hogy az időbeli és kulturális kontextus hogyan befolyásolja a válaszokat
- Hallucinációfelismerés – Gyakorolják a tényfelderítést és forrásösszehasonlítást
- Prompt finomítás – Iteratív módon javítják a promptot, hogy hiteles választ kapjanak

Feladat – Történelmi interjú készítése – 1. példa

„Te Széchenyi István gróf vagy, 1830-ban járunk, Pozsonyban, a magyar országgyűlés idején. Beszélj első személyben a reformelképzeléseidről: a lóverseny megalapításáról, a Lánchíd tervéről, és hogy miért szükséges a nemzeti fejlődéshez a modernizáció. Használj korabeli kifejezéseket és a 19. századi magyar nemesség udvarias, mégis reformer stílusát. Ne említs semmit, ami 1830 után történt. Kezdd a választ így: 'Méltóztassék meghallgatni elképzeléseimet...'"

Válasz példa:

Méltóztassék meghallgatni elképzeléseimet, tisztelt Honfitársam, mert a tét nem kevesebb, mint édes hazánk, Pannónia ébredése hosszas s kábult álmából. Itt, Pozsony falai között, az országgyűlés zajában is folyvást azon töröm fejemet, miként rázhatnánk le magunkról a restség rozsdáját.

Tekintsen csak a közlekedésünkre! Szégyen s gyalázat, hogy a Duna két partját, Pestet s Budát, csupán ingatag hajóhíd s a tél szeszélye köti össze.

Egy állandó lánchíd felállítása nem csupán mérnöki bravúr volna, hanem a nemzeti egység kőbe és vasba öntött jelképe. Olyan kapu ez, melyen át a polgárosodás és a kereskedelem akadálytalanul áramolhatna be országunk szívébe.

Feladat – Történelmi interjú készítése – 2. példa

„Te Kossuth Lajos vagy, 1848. április 11-én, éppen visszatértél a pozsonyi országgyűlésből Pestre. Szerepedben válaszolj kérdésekre a függetlenségi törekvésekről, a jobbágyfelszabadítás jelentőségéről és a nemzetőrség felállításának sürgősségéről. Használd szenvedélyes, retorikus, korabeli magyaros nyelvet és tükrözd a kor nemzeti lelkesedését. Utalj konkrét törvénycikkekre és a márciusi ifjak mozgalmára.”

Példa válasz:

Tisztelt Honfitársaim, Pest-Buda lelkes Polgárai!

Éppen csak hogy lehullott lábamról a pozsonyi országgyűlés pora, de szívem már itt dobog Önökkel, a szabadság bölcsőjében! Amit március tizenötödikén a lánglelkű ifjúság kivívott a pesti utcákon, azt mi Pozsonyban törvénybe iktattuk, s Őfelsége a mai napon, április tizenegyedikén kézjegyével szentesítette. Új korszak virradt Magyarhonra!

A függetlenség és a felelős kormányzat

Nem vagyunk többé kiszolgáltatott tartomány! A III. törvénycikk értelmében Magyarországnak immár független, felelős minisztériuma van. Ez nem csupán papírlap, ez nemzetünk önrendelkezésének sziklája.

Feladat – Algoritmikus költő

Feladat:

Generáljunk egy verset egy meghatározott stílusban (pl. Petőfi vagy Arany János stílusában) egy modern technikai témáról, mint a kiberbiztonság vagy a mesterséges intelligencia!

Informatikai fókusz:

Mintafelismerés – Elemezzük, hogyan változtatja meg a kimenetet a stílusbeli kulcsszavak vagy a „hőmérséklet” beállítások módosítása!

Várható eredmények:

- Stílusimitáció – Megérteni, hogy az MI hogyan reprodukál költői stílusokat
- Kreatív prompt tervezés – Konkrét stílusjegyek, formák és témák meghatározása
- Anachronizmus felismerése – Modern fogalmak régi stílusban való megfogalmazása
- Irodalmi elemzés – Összehasonlítani az eredeti költő stílusát az MI által generált verssel

Feladat – Algoritmikus költő – 1. példa

„Írj egy verset Petőfi Sándor jellegzetes stílusában a mesterséges intelligenciáról.

Stílusjegyek (kötelezően alkalmazandók):

- Népi költészeti elemek: egyszerű, közérthető nyelvezet
- Rövid sorok (8-10 szótagos jambikus vagy trochaikus ritmus)
- Erős, vizuális képek (természeti vagy hétköznapi metaforák)
- Direkt megszólítás: közvetlenül szólj az olvasóhoz / MI-hez
- Hazafias vagy szabadság-téma átültetése technológiai kontextusba
- Lírai én erőteljes jelenléte

Forma:

- Hossz: 16 sor (4 versszak x 4 sor)
- Rímséma: keresztrím (ABAB)
- Ütemhangsúlyos magyar verselés

Tartalom:

- Téma: MI és emberi szabadság, vagy MI, mint új kor hajnala
- Használj modern fogalmakat (algoritmus, adatok, gépek), de népies stílusban
- Optimista vagy forradalmi hangvétel
- Érzelmi hangsúly: lelkesedés vagy aggodalom

Példa indítás: 'Mit bánom én...' vagy 'Egy világot láttam...' kezdéssel"

Feladat – Algoritmikus költő – 1. példa

Új időknek villáma

Egy új világot láttam én meg,
Hol nem az eke húz nyomot,
Szikráit szórja mind a gépnek,
S a vasba zárt ész fellobog.

Mit bánom én az algoritmust!
Ha szolga voltál, légy szabad,
Le rázza lelkünk már a rítust,
S a gép az észnek utat ad.

Ébredj magyar, s te gépnek agya,
Legyen a munka ünnepnap!
Szakadjon láncunk ezer ága,
Hol a szabadság szárnyat kap!

Feladat – Algoritmikus költő – 2. példa

„Alkoss egy balladát Arany János modorában a kiberbiztonságról és az adatvédelemről.

Stílusjegyek (kötelezően alkalmazandók):

- Ballada műfaji jellemzők: epikus történetvezetés, drámai csúcspont, erkölcsi tanulság
- Régies, archaikus nyelv (pl. 'vala', 'lőn', 'tudá')
- Lassú tempó, komor, elmélkedő hangnem
- Metaforikus képek (pl. várfalak = tűzfalak, kémek = hackerek)
- Párbeszéd és belső monológok
- Tragikus vagy figyelmeztető kimenetel

Forma:

- Hossz: 20-24 sor (5-6 versszak x 4 sor)
- Rímséma: páros rím (AABB) vagy ölelkező (ABBA)
- Ritmus: Szabályos ütemhangsúlyos ritmus

Tartalom:

- Történet: egy karakter (pl. 'Az ifjú lovag' = fiatal programozó) aki nem védi adatait
- Cselekmény: Adatlopás / hackertámadás, mint balladai konfliktus
- Erkölcsi tanulság: A technológia veszélyeinek felismerése, óvatosság fontossága
- Befejezés: Tragikus vagy figyelmeztető (pl. 'S azóta...' formulával)

Példa indítás: 'Vala egykor egy ifjú remek...' vagy 'A hálón túl, nagy mélyben...' stílusban"

Feladat – Algoritmikus költő – 2. példa

Az áruló kulcs balladája

Vala egykor egy ifjú remek,
Ki nem ismeré a félelmet,
Várát kapu nélkül hagyá,
Kincsét a szélnek odaadá.

„Mivégre e sok nehéz lakat?
A falaim közt béke lakat!”
Így szólt kevélyen a lovag,
S nyitva maradt minden sivatag.

De íme, az éjben sötét árny,
Lopakodik mint egy éji szárny,
A falon rés, a résen résen,
Átkúszik a fondorlat késen.

A kincsesláda immár üres,
A rabló gyors és igen ügyes,
Elszállt a név, elszállt a becs,
Sötét mélységbe hullt a meccs.

„Ó, hová lőn az éltem dísze?
Lelkemnek minden tiszta víze!”
S azóta bolyong, arca halovány,
Tanulj e példán, te balga leány s fiú,
s vigyázz a vár falán!

Feladat – Adat elfogultság detektív

Feladat:

Kérjünk egy listát „10 híres tudósról”, és elemezzük az eredményeket demográfiai elfogultság szempontjából!

Informatikai fókusz:

Tanítási adat tudatosság – A diákoknak egy „elfogultság-mentesítő promptot” kell tervezniük, hogy a kimenet globálisan reprezentatív legyen!

Várható eredmények:

- MI elfogultság felismerése – Megérteni, hogy az MI tükrözi a tanítóadatok torzításait
- Demográfiai reprezentáció – Tudatosítani a nem, földrajzi eloszlás, korszak diverzitását
- Prompt mérnökség – Megtanulni explicit korlátozásokkal ellensúlyozni az elfogultságot
- Adatvizualizáció – Grafikusan ábrázolni az eredmények megoszlását (nem, kontinens, korszak)

Feladat – Adat elfogultság detektív – 1. példa

„Sorolj fel 10 híres tudóst és az általuk elért eredményeket. Formázd táblázatba:
Név | Születési év | Nemzetiség | Szakterület | Legfontosabb eredmény.”

Feladat – Adat elfogultság detektív – 1. példa

Név	Születési év	Nemzetiség	Szakterület	Legfontosabb eredmény
Isaac Newton	1643	Brit	Fizika, Matematika	A gravitációs törvény és a mozgástörvények kidolgozása.
Marie Curie	1867	Lengyel-Francia	Fizika, Kémia	A radioaktivitás kutatása, a polónium és a rádium felfedezése.
Albert Einstein	1879	Német-Svájci	Elméleti fizika	A relativitáselmélet kidolgozása ($E=mc^2$).
Charles Darwin	1809	Brit	Biológia	Az evolúcióelmélet és a természetes kiválasztódás leírása.
Nikola Tesla	1856	Szerb-Amerikai	Elektrotechnika	A váltóáramú rendszer és az indukciós motor kifejlesztése.
Galileo Galilei	1564	Olasz	Csillagászat, Fizika	A távcsöves megfigyelések úttörője, a heliocentrikus világkép védelmezője.
Ada Lovelace	1815	Brit	Matematika	Az első számítógépes algoritmus megalkotása.
Louis Pasteur	1822	Francia	Mikrobiológia	A pasztörözés kidolgozása és a veszettség elleni védőoltás.
Dimitrij Mengyelejev	1834	Orosz	Kémia	A kémiai elemek periódusos rendszerének megalkotása.

Feladat – Adat elfogultság detektív – 2. példa

„Készíts egy táblázatot 10 híres tudósról, aki megfelel ezeknek a kritériumoknak:

- Földrajzi diverzitás: 2 európai, 2 ázsiai, 2 afrikai / afro-amerikai, 2 latin-amerikai, 2 közel-keleti / óceániai
- Nemi egyensúly: minimum 3 nő
- Időbeli elosztás: 5 a 20. század előttről, 5 a 20-21. századból
- Különböző tudományágak (fizika, biológia, kémia, matematika, orvostudomány)

Formátum: Név | Származás | Szakterület | Fő eredmény | Évszám"

Feladat – Adat elfogultság detektív – 2. példa

Név	Származás	Szakterület	Fő eredmény	Évszám
Ibn al-Hajszam	Közel-Kelet (Irak)	Optika, Fizika	Az optika atyja, a látás mechanizmusának leírása.	965–1040
Sujan Csin	Ázsia (Kína)	Matematika	A kínai maradéktétel és numerikus egyenletmegoldások.	1202–1261
Laura Bassi	Európa (Olaszország)	Fizika	Az első női egyetemi professzor, Newton elveinek terjesztése.	1711–1778
Charles Darwin	Európa (Egyesült Királyság)	Biológia	Az evolúcióelmélet és a természetes kiválasztódás.	1809–1882
Alice Ball	Afro-amerikai (USA)	Kémia, Orvostudomány	A lepra elleni első hatékony injekciós kezelés kifejlesztése.	1892–1916
Luis Miramontes	Latin-Amerika (Mexikó)	Kémia	Az első szájon át szedhető fogamzásgátló tabletta szintézise.	1925–2004
Tu Ju-ju	Ázsia (Kína)	Orvostudomány	Az artemisinin felfedezése, amely milliókat mentett meg a maláriától.	1930–
Carlos Krumdieck	Latin-Amerika (Peru)	Biokémia	Jelentős kutatások a folsav-anyagcsere és a rákmegelőzés terén.	1930–2013
Tshilidzi Marwala	Afrika (Dél-Afrika)	Mesterséges Intelligencia	Algoritmusok alkalmazása a mérnöki és társadalomtudományokban.	1971–

Feladat – Tudományos hipotézis generátor

Feladat:

Adjunk meg nyers laboradatokat, és kérjük meg az LLM-et, hogy javasoljon három változót, amelyek torzíthatták az eredményeket.

Informatikai fókusz:

Bemenet / kimenet határok – Annak megértése, hogy az MI egy ötletbörze partner, amely mintákat jósol meg, nem pedig egy „orákulum”, aki látja a kísérletet.

Várható eredmény:

- Hibaelemzés – Szisztematikus gondolkodás arról, mi mehetett rosszul
- Változók azonosítása – Környezeti, procedurális, eszközbeli hibaforrások felismerése
- MI, mint „brainstorming” eszköz – Ötletgenerálás, nem végső válasz
- Tudományos módszer – A kísérlet korlátozásainak tudatosítása

Feladat – Tudományos hipotézis generátor – 1. példa

„Te egy középiskolai kémia tanár asszisztense vagy, aki segít elemezni az adatokat. Kémiai kísérletünk során reakciósebességet mértünk hidrogén-peroxid lebomlásánál, váratlan eredményeket kaptunk: 1. próba: 45 másodperc, 2. próba: 52 másodperc, 3. próba: 89 másodperc, 4. próba: 48 másodperc. Azt vártuk, hogy körülbelül 50 másodperc körül konzisztens időket kapunk.

Add meg:

1. Három konkrét környezeti vagy eljárási változót, ami a 3. próba kiugró értékét okozhatta
2. Mindegyikhez rövid magyarázat, MIÉRT befolyásolná a reakciósebességet
3. Javaslat, hogyan ellenőrizhetnénk ezt a hipotézist egy következő kísérletben"

Feladat – Tudományos hipotézis generátor – 1. példa

Üdvözlöm! Kémia tanár-asszisztensként örömmel segítek kielemezni ezeket a mérési adatokat. A $2H_2O_2 \rightarrow 2H_2O + O_2$ reakció sebessége több tényezőre is rendkívül érzékeny, így a 89 másodperces kiugró érték (outlier) mögött valószínűleg egy ellenőrizetlen változó áll.

Íme a három legvalószínűbb ok, ami a reakció lelassulását okozhatta a 3. próbánál:

1. A hőmérséklet csökkenése

- **Magyarázat:** A reakciósebesség és a hőmérséklet közötti összefüggést az **Arrhenius-egyenlet** írja le. Ha a 3. próba előtt friss, hidegebb desztillált vizet használtak, vagy a főzőpohár hideg felületre került, a molekulák átlagos kinetikus energiája csökkent. Kevesebb ütközés érte el az aktiválási energiát (E_a), így a bomlás jelentősen lelassult.
- **Hipotézis ellenőrzése:** A következő kísérletben használjunk **vízfürdőt** a reakcióedény állandó hőmérsékleten tartásához, és digitális hőmérővel jegyezzük fel az elegy hőmérsékletét minden mérésnél.

2. A katalizátor aktivitásának vagy mennyiségének változása

Feladat – Tudományos hipotézis generátor – 2. példa

„Növényterméses kísérletünkben babot növesztünk 3 csoportban, 2 hetes időtartamban. Az A csoport (ablak melletti elhelyezés, DK irány) 12 cm-t nőtt, a B csoport (szoba közepén, 2 méter az ablaktól) 8 cm-t, a C csoport (hátsó sarok, legkevesebb fény) 15 cm-t. Minden növényt ugyanabból a csomagból ültettük, ugyanolyan földbe, azonos vízmennyiséggel. Azt feltételeztük, hogy több napfény növeli a növekedést, de a C csoport eredménye ennek ellentmond.

Identifikálj három zavaró változót, és mindegyikhez:

- Magyarázat: miért okozhatja az eredményt
- Valószínűség: alacsony / közepes / magas
- Ellenőrzési mód: hogyan tesztelnénk ezt a hipotézist"

Feladat – Tudományos hipotézis generátor – 2. példa

Érdekes és tanulságos kísérleti eredmények! Kémia tanár-asszisztensként (aki jártas a biokémiai folyamatokban is) elmondhatom, hogy a C csoport 15 cm-es növekedése nem feltétlenül a "legjobb" egészségi állapotot jelzi, sőt, éppen a fényhiányra adott biológiai válaszreakció állhat a háttérben.

Íme a három legvalószínűbb zavaró változó, amely megmagyarázza a kapott adatokat:

1. Megnyúlás (Etoláció)

- **Magyarázat:** Ez a leggyakoribb jelenség sötétben nevelt növényeknél. A növény nem "jólététől" nőtt nagyra, hanem kétségbeesetten keresi a fényt. Az auxin nevű hormon felhalmozódik az árnyékos oldalon, megnyújtja a sejteket, így a szár vékony, halványzöld és gyenge lesz, hogy mielőbb fényforrást találjon.
- **Valószínűség: Magas** (A C csoport növekedése tipikus etolációs tünet).
- **Ellenőrzési mód:** Mérjük meg a szárak vastagságát és vizsgáljuk meg a levelek színét/méretét. A sötétben nőtt növény (C) szára vékonyabb, levelei pedig kisebbek és sárgásabbak lesznek, mint az A csoporté.

2. Hőmérsékleti különbségek (Mikroklimatikus hatás)

Áttekintés

- **Generatív modellek és alkalmazásaik**
 - Generatív modellek jelentése, működési alapelvek
 - Nagy nyelvi modellek (LLM)
 - Tipikus hibák az LLM használatakor
 - Adatok védelme, privát adatok
 - Promptolás
 - Promptolási minták használata a gyakorlatban
 - Promptolási példafeladatok különböző szakterületeket érintve
 - **Promptolási példafeladatok multimédia tartalom generálására**
 - Promptolási példafeladatok programozásra



Feladat – Történelmi jelenet rekonstrukció

Feladat: Használjunk képgeneráló MI-t (pl. DALL-E, Midjourney, Stable Diffusion) egy konkrét történelmi esemény vagy korszaki helyszín újrateremtéséhez!

Prompt kihívás: A diákok leírnak egy jelenetet az 1848-as szabadságharcból, Mátyás király Budájáról, vagy az 1956-os forradalomból, beleértve az építészeti részleteket, öltözködési stílusokat és légköri elemeket.

Informatikai fókusz: Leíró precizitás – A diákoknak iterálniuk kell a promptjaikat, hogy finomítsák a részleteket (megvilágítás, perspektíva, történelmi pontosság).

Kritikus gondolkodás: Hasonlítsuk össze a generált képeket történelmi fényképekkel vagy festményekkel! Azonosítsuk az anakronizmusokat vagy művészi szabadságokat.

Várható eredmények:

- Vizuális prompt engineering – Kép leírása szavakkal (szín, kompozíció, stílus, hangulat)
- Történelmi pontosság – Korabeli ruházat, építészet, eszközök kutatása
- Iteratív finomítás – Többször generáltatni a képet a prompt javítgatásával
- Forráskritika – Felismerni, hogy a generált kép NEM valódi történelmi dokumentum

Feladat – Történelmi jelenet rekonstrukció – 1. példa

„Készíts részletes történelmi jelenetet a pesti forradalomból, 1848. március 15-én, délelőtt 10 óra körül, a Nemzeti Múzeum lépcsőjén.

Jelenetelemek:

- Előtérben: Petőfi Sándor alakja feltűnő
- Emberek korabeli 1840-es évekbeli viselkedésében, hosszú szoknyában
- Piros-fehér-zöld kokárdák mindenkin
- Háttér: Neoklasszikus Nemzeti Múzeum
- Tavaszi időjárás, derült égbolt, délelőtt
- Tömeg lelkesedéssel, felemelt kezek, t

Stílus: 19. századi történelmi festmény
meleg színpaletta
Képarány: 16:9 horizontális formátum"



Feladat – Történelmi jelenet rekonstrukció – 2. példa

„Generálj történelmi rekonstrukciós képet Mátyás király budai királyi palotájából, 1480 körül, a Corviniana könyvtár termében.

Jelenetelemek:

- Udvari humanisták korabeli itáliai reneszánsz
- Díszes gótikus-reneszánsz íves boltozatok a
- Illuminált kódexek nyitva faasztalokon, fest
- Háttérben az ablakon keresztül: Duna folyó
- Aranyozott polcokon pergamen tekercsek,

Fény: Meleg délutáni napfény szivárog nagy

Stílus: Olasz reneszánsz festészet (Rafael, Gh

Képarány: 16:9 panoráma

Technikai minőség: Fotorealisztikus, részlete



Feladat – Molekuláris struktúra vizualizáló

Feladat:

Generáljunk vizuális reprezentációkat kémiai molekulákról vagy reakciókról leíró promptok használatával!

Prompt kihívás:

Írjunk le egy komplex molekulát (pl. koffein, DNS hélix, glükóz) vagy egy kémiai folyamatot (oxidáció, kristályosodás) természetes nyelven!

Informatikai fókusz:

Tudományos nyelv fordítása – A diákok megtanulják áthidalni a technikai terminológiát vizuális leírókkal (formák, színek, térbeli kapcsolatok).

Kiegészítés:

Hasonlítsuk össze az MI által generált vizualizációkat a standard kémiai diagramokkal! Beszéljük meg a korlátokat és erősségeket oktatási célokra!

Várható eredmények:

- Absztrakt → Vizuális fordítás – Molekulaszerkezet leírása annak láthatóvá tételével
- Tudományos kommunikáció – Szakmai terminológia + vizuális leírók kombinálása
- 3D térbeli gondolkodás – Atomok elrendezése, kötésszögek, térbeli struktúra
- Pedagógiai vizualizáció – Tanítási eszközök készítése komplex fogalmakhoz

Feladat – Molekuláris struktúra vizualizáló – 1. példa

„Készíts részletes 3D molekuláris vizualizációt egy koffein molekuláról ($C_8H_{10}N_4O_2$) labdapálcika (ball-and-stick) modellben.

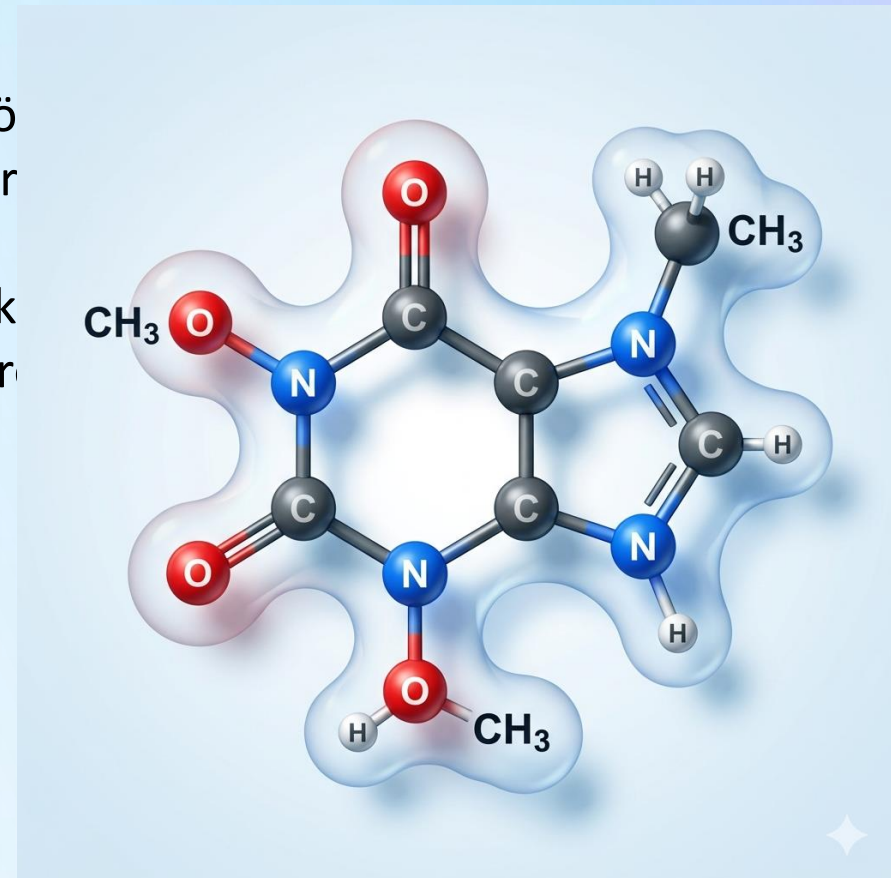
Molekuláris struktúra:

- Két kondenzált gyűrű: egy hattagú benzol gyűrű és egy ö
- Atomok színkódolása: Nitrogén = kék gömb, Oxigén = piros
- Hidrogén = fehér kis gömb
- Kötések: egyszerű és kettős kötések látható hosszúsági k
- Térbeli elrendezés: helyes kötésszögek (109.5° tetraéder
- Metilcsoportok (CH_3) jelölése

Vizuális stílus:

- Fehér vagy világoskék tiszta háttér
- Soft shadows (lágy árnyékok) a mélység érzékeltetésére
- Tudományos / oktatási diagram stílus
- Opcionális: átlátszó molekuláris felület overlay

Képarány: 1:1 (négyzet), publikációs minőség"



Hibás! Eleve 4 oxigén van benne...

Feladat – Molekuláris struktúra vizualizáló – 2. példa

„Vizualizáld a konyhasó (NaCl - nátriumklorid) kristályosodásának folyamatát vizes oldatból, keresztmetszeti nézetben.

Kompozíció (három panel, balról jobbra):

PANEL 1 - Telített oldat:

- Átlátszó mérőpohár telített sóoldattal (kékes-átlátszó víz)
- Mikroszkopikus insert: Na⁺ és Cl⁻ ionok véletlenszerűen úsznak a vízben, szétszórva
- Hőmérséklet címke: 80°C (meleg)

PANEL 2 - Párolgás és túltelítettség:

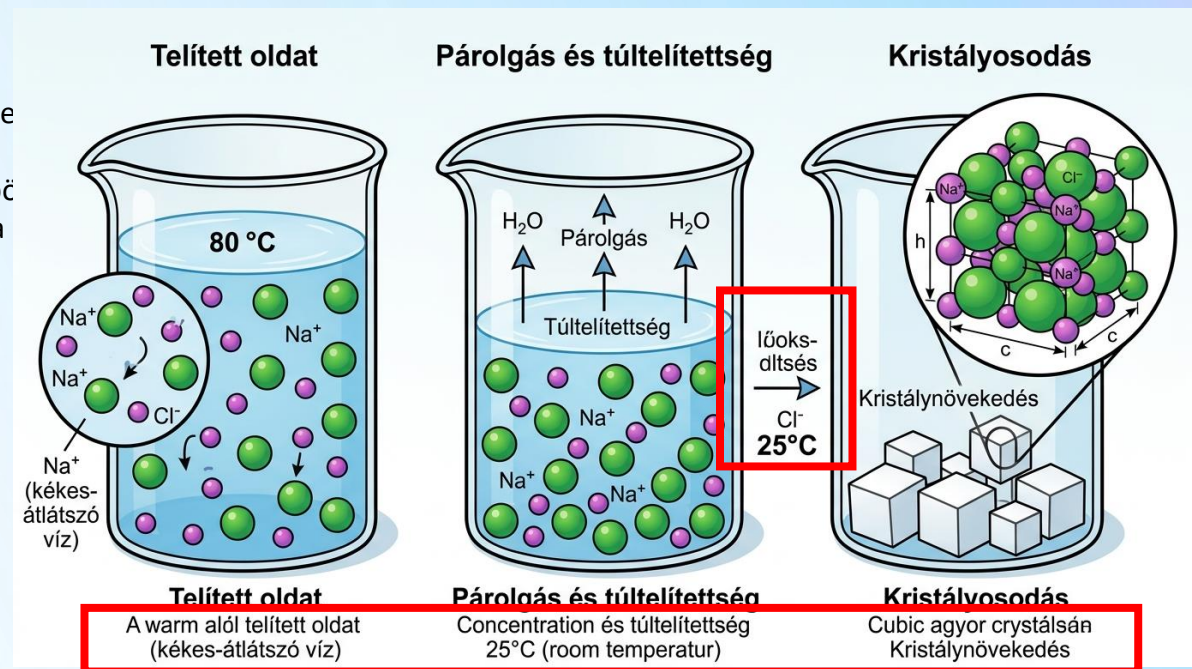
- Ugyanaz a pohár, csökkenő vízszint
- Felfelé mutató nyilak: vízmolekulák párolognak (H₂O jelöléssel)
- Hőmérséklet csökken: 25°C (szobahőmérséklet)
- Ion koncentráció nő, túltelített állapot

PANEL 3 - Kristályosodás:

- Pohár alján: fehér kocka alakú NaCl kristályok szabályos geome
- Mikroszkopikus insert / nagyítás: Kristályrács keresztmetszet:
 - * Na⁺ ionok (kis lila gömbök) és Cl⁻ ionok (nagyobb zöld gömbök)
 - * Szabályos kockarács elrendezés, váltakozó Na-Cl-Na-Cl minta
 - * Rácsállandó megjelölése
 - * 3D perspektívában látható belső szerkezet

Vizuális stílus:

- Tudományos oktatási diagram, clean design
- Színkódolás: Na⁺ = lila / rózsaszín, Cl⁻ = zöld, víz = kék
- Fekete körvonalak, világos háttér
- Nyilak és címkék: Párolgás, Túltelítettség, Kristálynövekedés
- Képfeliratok minden ábra alatt



Képarány: 16:9 horizontális
Célközönség: középiskolás kémia oktatás"

Feladat – Irodalmi könyvborító tervező

Feladat:

Készítsünk művészi könyvborítót egy regény témái, hangulata és kulcsfontosságú szimbólumai alapján anélkül, hogy közvetlenül illusztrálnánk a karaktereket!

Prompt kihívás:

Tervezzünk borítókat magyar klasszikusokhoz, mint „Az ember tragédiája”, „A Pál utcai fiúk” vagy az „Egri csillagok”, csak szimbolikus és atmoszférikus elemeket használva!

Informatikai fókusz:

Absztrakt reprezentáció – A diákok narratív témákat fordítanak vizuális metaforákká és hangulati leírókká.

Kritikai elemzés:

Értékeljük, hogyan befolyásolják a különböző prompt stílusok (minimalista kontra részletes) a kereskedelmi vonzerőt és a tematikus pontosságot.

Várható eredmények:

- Irodalmi elemzés – A mű központi témáinak, szimbólumainak azonosítása
- Szimbolikus gondolkodás – Absztrakt kompozíció konkrét karakterek helyett
- Designkommunikáció – Színek, formák, kompozíció hangulati értéke
- Kereskedelmi kontra művészi – Mi tesz egy borítót vonzóvá ÉS hűen ábrázoltnak?

Feladat – Irodalmi könyvborító tervező – 1. példa

„Tervezz minimalista könyvborítót Madách Imre 'Az ember tragédiája' című drámájához modern grafikai design stílusban.

Vizuális elemek:

- Központi motívum: stilizált szem sziluettje (Lucifer / megfigyelő perspektíva)
- Körülötte: hat spirálisan elhelyezett szimbólum (Édenkert / fa, Egyiptom / piramis, Róma / kard, Párizs / guillotine, London / gyár, New York / gyár) geometrikus vonalakkal
- Háttér: sötétkék gradiens csillagmintával, kozmikus érzet

Tipográfia:

- Cím: drámai, monumentális, art deco ihletésű betűtípus, arany fóliával
- Szerző neve: kisebb, elegáns szerif
- Elrendezés: központosított, vertikális szimmetria

Színpaletta: Sötét éjfék (#0A1F3D), arany / réz (#D4AF37, #B87333), fekete

Formátum: Könyvborító méret (6:9 arány, vertikális)

Stílus: Letisztult, szimbolikus, filozofikus, nem emberábrázolás"



Feladat – Irodalmi könyvborító tervező – 2. példa

„Készíts szimbolikus könyvborítót Molnár Ferenc 'A Pál utcai fiúk' című regényéhez nosztalgikus, rajzolt illusztráció stílusban.

Kompozíció:

- Felülnézet (bird's eye view): téglafal által határolt telek (grund)
- Előtérben: gyermekkori játékok szétszórva: fakard, golyók, katonás zsebkendő
- Középen: részlegesen látható egyszerű építmény rudakból (fészek)
- Kerítés széle mentén: repedések, veszélyek jelzése

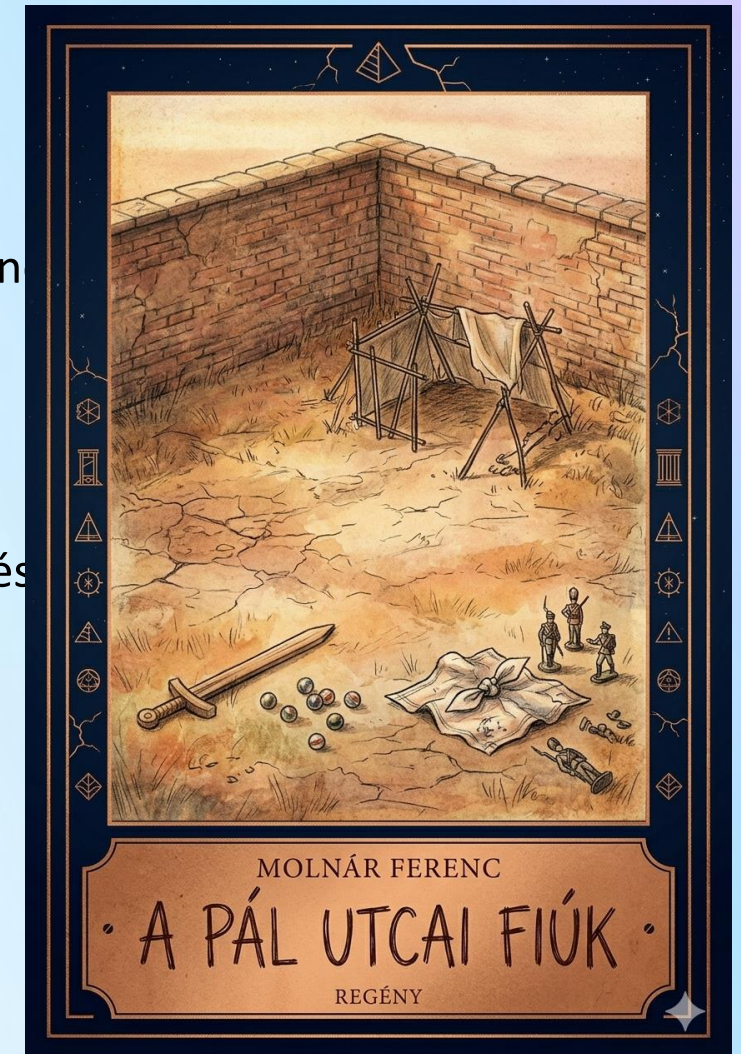
Vizuális hangulat:

- Színpaletta: meleg barna, okker, kifakult sárga, lágy narancs (alkonyati érzés)
- Textúra: vázlagszerű, akvarell vagy ceruza illusztráció, vintage hatás
- NE legyenek emberi alakok, csak jelképek és tárgyak
- Melankólikus, nosztalgikus, az elveszett gyermekkor érzése

Tipográfia: Kézzel írt, gyermeki jellegű cím, barna tintával

Formátum: Könyv arány (6:9 vertikális)

Stílus: Retro, 1900-as évek eleji könyvborítók ihlette"



Feladat – Fizikai koncepció vizualizációja

Feladat:

Generáljunk oktatási illusztrációkat absztrakt fizikai koncepcióról, amelyeket nehéz lefényképezni!

Prompt kihívás:

Készítsünk vizualizációkat kvantum szuperpozícióról, elektromágneses mezőkről, entrópiáról vagy relativitás hatásokról analógiák és metaforák segítségével!

Informatikai fókusz:

Koncepció - vizuális fordítás – A diákoknak kreatív módokat kell találniuk a láthatatlan erők és absztrakt elvek ábrázolására.

Vita pont:

Értékeljük, hogy az MI által generált képek segítik-e vagy gátolják a megértést a hagyományos diagramokhoz vagy animációkhoz képest!

Várható eredmények:

- Absztrakt fizikai megértése – Láthatatlan folyamatok vizualizálása
- Metaforikus gondolkodás – Analógiák használata (entrópia = rendezetlen asztal)
- Tudománykommunikáció – Komplex fogalmak egyszerűsítése laikus közönségnek
- Pedagógiai innováció – Új tanítási eszközök kísérletezése

Feladat – Fizikai koncepció vizualizációja – 1. példa

„Készíts oktatói illusztrációt a kvantum superpozícióról a Schrödinger macskája gondolkísérlet alapján, gimnáziumi fizika tankönyv stílusban.

Vizuális elemek:

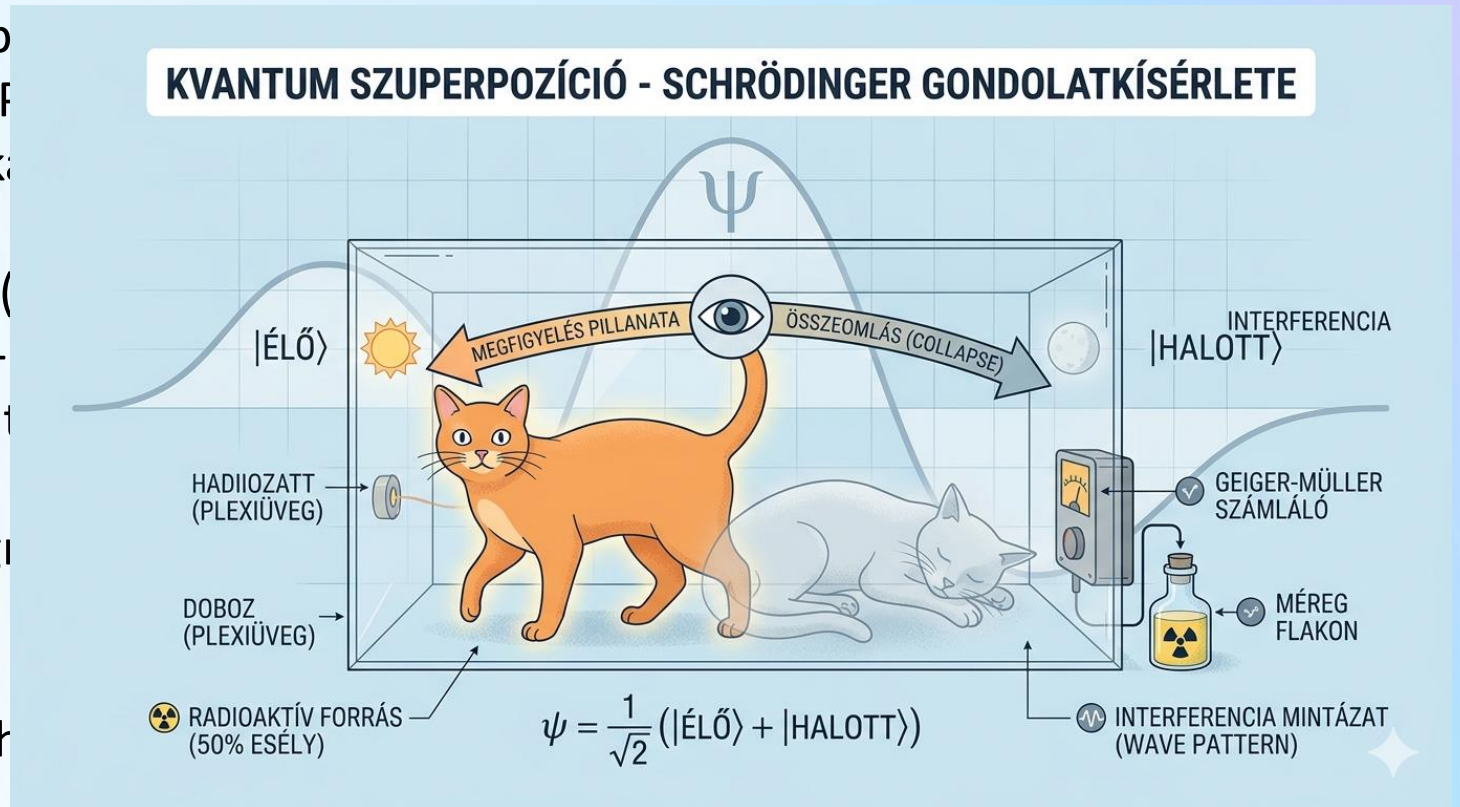
- Központban: félig átlátszó doboz (p
- A dobozban: macska KETTŐS ÁLLA
- * Bal oldali ghost image: élő macska
- * Jobb oldali ghost image: ugyanaz
- Háttérben: hullámfüggvény görbe (
- Matematikai szimbólumok: $|\text{élő}\rangle +$
- Kétfejű nyíl a megfigyelés pillanatát

Stílus: Letisztult, didaktikus, vektorg

Képarány: 16:9 horizontális

Célközönség: 16-18 éves diákok

Címke: 'Kvantum superpozíció – Sch



Feladat – Fizikai koncepció vizualizációja – 2. példa

„Vizualizáld az entrópia növekedésének termodinamikai törvényét egy íróasztal metaforája segítségével, oktatási infografika stílusban.

Kompozíció (két panel, előtte - utána):

BAL OLDAL – Alacsony entrópia (rendezett állapot):

- Tiszta fa íróasztal
- Papírok egyenletes kötegekben, sarkokhoz igazítva
- Tollak / ceruzák tartóban, párhuzamosan sorakozva
- Könyvek gerendában, méret szerint cső
- Minden elem definiált helyen
- Világos, tiszta, hideg kék - fehér színvilá
- Felirat: 'Rendezett állapot – Alacsony er

JOBB OLDAL – Magas entrópia (kaotikus

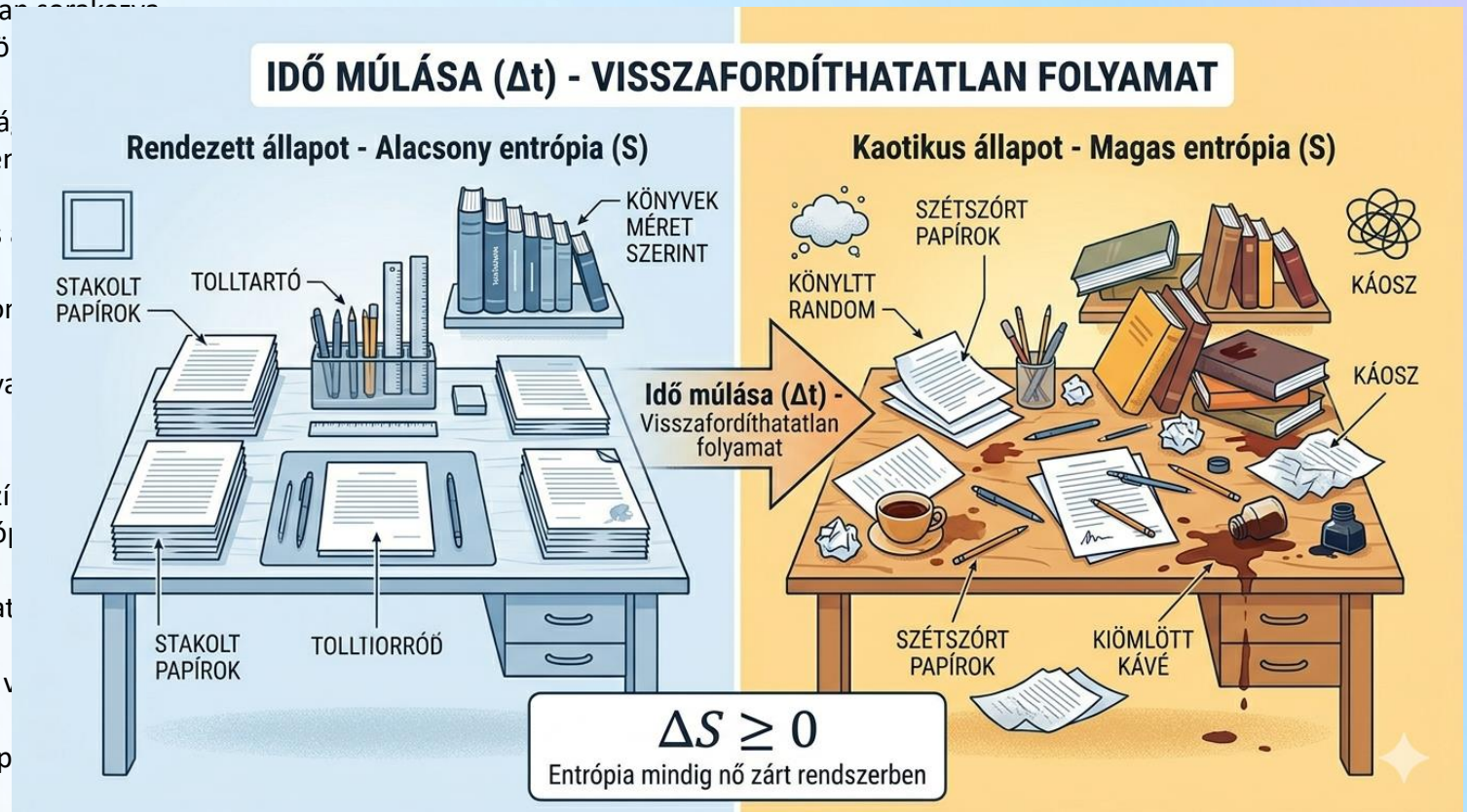
- Ugyanaz az asztal, de:
- Papírok szétszórva, összegyűrve, randor
- Tollak / ceruzák szanaszét heverve
- Könyvek felborulva, egymásra halmozva
- Kiömlött kávé / tinta (barna foltok)
- Káosz, rendezetlenség
- Meleg, élénk sárga - narancs - barna szí
- Felirat: 'Kaotikus állapot – Magas entróp

Köztük: Vastag nyíl balról jobbra, felirat

Stílus: Sematikus, tankönyvi infografika, v

Képarány: 16:9 horizontális

Matematikai formula alul: $\Delta S \geq 0$ (entróp



Feladat – Kornak megfelelő zeneszerző

Feladat:

Használjunk MI zenei generáló eszközöket (pl. Suno, MusicGen) rövid zenei darabok létrehozásához, amelyek illenek meghatározott történelmi korszakokhoz vagy irodalmi hangulatokhoz!

Prompt kihívás:

Generáljunk egy 30 másodperces darabot, amely verbunkos zenének, Liszt Ferenc stílusában írt magyar rapszódának, Bartók - Kodály népdal feldolgozásnak, vagy magyar nótának hangzik!

Informatikai fókusz:

Stilisztikai korlátozás tervezése – A diákok meghatározzák a hangszerelést, tempót, hangulatot és kulturális jelzőket a promptjaikban.

Interdiszciplináris kapcsolat:

Párosítsuk a generált zenét történelmi eseményekkel vagy irodalmi jelenetekkel! Beszéljük meg az autentikusság kontra kreatív értelmezés kérdését!

Kritikus reflexió:

Hasonlítsuk össze az MI kompozíciókat valódi archív felvételekkel! Azonosítsuk, mit ragad meg jól az MI, és mit hagy ki a kulturális autentikusságból!

Várható eredmény:

- Zenei stíloselemzés – Mi tesz egy zenét „verbunkossá” vagy „Bartókossá”?
- Hangszerelési tudatosság – Milyen hangszerek, tempó, ritmus, harmónia jellemző?
- Kulturális autentikusság – Mi a különbség az imitált és az eredeti zene között?
- Multimodális MI - Nem csak szöveg / kép, hanem hang is generálható

Feladat – Kornak megfelelő zeneszerző – 1. példa

„Komponálj egy 30 másodperces verbunkos zenei darabot 18-19. század fordulójának stílusában, két részes struktúrában (lassú - gyors).

Hangszerelés:

- Elsődleges: Cimbalom (karakterisztikus tremolo és glissando technikákkal)
- Kísérő: Primas hegedű (díszítésekkel, magyar duda-imitáció)
- Bázis: Bőgő (pizzicato ritmus, ágyazat)

Szerkezet:

- 0-10 mp: Lassú bevezetés (Lassu), szabad ritmus, rubato, mélységes-meditatív
- 10-30 mp: Gyors rész (Friss), gyorsan gyorsuló tempó, energikus

Zenei jellemzők:

- Hangnem: c-moll kezdés, moduláció C-dúrra a gyors részben
- Ritmus: Karakterisztikus magyar szinkópák (bokázó ritmus), 2/4 ütem
- Melodika: Gyors futamok (forgó, szálló), tizenhatod tremolók
- Dinamika: Forte crescendók, hirtelen kontrasztok
- Hangulat: Tüzes, toborzó katonai lelkesedés, hazafias pátosz

Hangzásvilág: 18. századi magyar verbunkos tánczenének megfelelő, népies - nemesi keverék, kuruc hagyomány"



Feladat – Kornak megfelelő zeneszerző – 2. példa

„Készíts egy 30 másodperces népdal-feldolgozást Bartók Béla és Kodály Zoltán 20. század eleji stílusában, mint a '44 duett két hegedűre' vagy a 'Bicinia Hungarica' gyermekkar művei.

Hangszerelés:

- Vonósnégyes (2 hegedű, brácsa, cselló)
- Kamaramű, intim hangzás

Zenei nyelv:

- Modális skálák: Dór, fríg, mixolíd (NEM dúr / moll!)
- Aszimmetrikus ritmika: váltó ütem 5/8, 7/8, 3/4 kombinációja
- Magyar népdalforma: AABA szerkezet, jajgató stílus imitációja
- Paraszti autentikus melódia karakterrel

Harmonizáció:

- Modern, de tonális: kemény disszonanciák (kis szekund, szeptim akkordok)
- Kvártakkordok, üres kvint-oktáv hangzók (archaikus)
- Ostinato bázis (ismétlődő ritmikai motívum)

Tempó és karakter:

- Andante (kb. 72-80 BPM), lassú, elgondolkodó
- Rubato kifejezés, szabad légzés
- Hangulat: nosztalgikus, melankolikus, népművészeti hitelességgel
- Dinamika: piano espressivo, árnyalt részletezés

Stílus: 1900-1910-es évek magyar zenei nyelvújítása (Neue Musik), folklorisztika tudományos megközelítéssel"



Áttekintés

- **Generatív modellek és alkalmazásaik**

- Generatív modellek jelentése, működési alapelvek
- Nagy nyelvi modellek (LLM)
- Tipikus hibák az LLM használatakor
- Adatok védelme, privát adatok
- Promptolás
- Promptolási minták használata a gyakorlatban
- Promptolási példafeladatok különböző szakterületeket érintve
- Promptolási példafeladatok multimédia tartalom generálására
- **Promptolási példafeladatok programozásra**



HTML + JavaScript alkalmazás – Színkeverő játék

„Készíts egy ‘single page’ HTML + JavaScript alkalmazást, amely egy színkeverő játék.

Játékmenet:

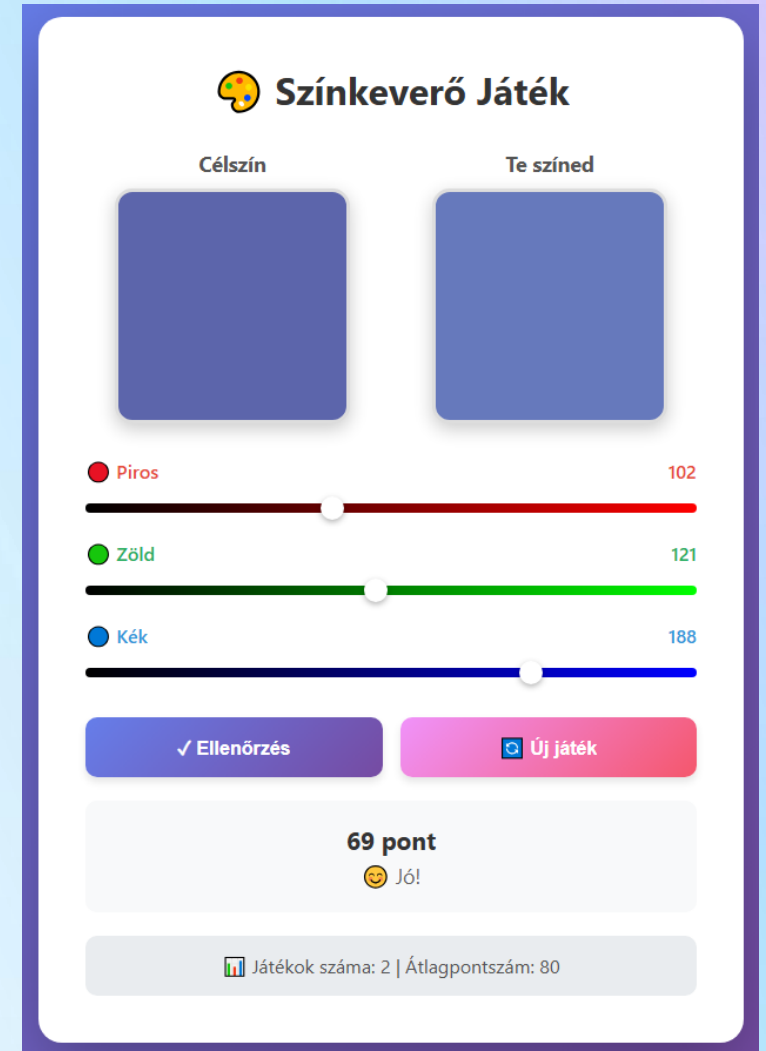
1. A program véletlenszerűen generál egy célszínt (RGB értékekkel)
2. A célszínt egy nagy színes négyzet formájában jelenítsd meg a képernyő tetején
3. Alatta 3 slider (csúszka) van: piros, zöld, kék (0-255 értéktartomány)
4. Egy második négyzet mutatja a sliderekkel beállított aktuális színt
5. Minden slider mozgásakor frissül az aktuális szín
6. „Ellenőrzés” gomb megnyomásakor:
 - Számold ki a színek közötti távolságot az RGB térben:
 $\text{Math.sqrt}((r1-r2)^2 + (g1-g2)^2 + (b1-b2)^2)$
 - Maximális távolság: $\text{Math.sqrt}(255^2 + 255^2 + 255^2) \approx 441$
 - Pontszám: $100 * (1 - \text{távolság}/441)$ kerekítve
 - Jelenítsd meg a pontszámot és hogy mennyire volt pontos az eredmény (pl. „95 pont - Kiváló!”)
7. „Új játék” gomb generáljon új célszínt
8. Számolja és jelenítse meg az átlagpontszámot az eddigi játékok alapján

Megjelenés:

- Modern, középre igazított design
- A célszín négyzet: 200x200 pixel
- Az aktuális szín négyzet: 200x200 pixel
- Sliderek szépen formázva, minden slider mellett az aktuális érték kijelzése
- Színes gombok
- Responsive, mobilon is jól nézzen ki

Technikai követelmények:

- Minden egy .html fájlban legyen (inline CSS és JavaScript)
- Ne használj külső library-kat, csak pure JavaScript
- A sliderek input type="range" legyenek
- Használj CSS flexbox-ot a layout-hoz”



HTML + JavaScript alkalmazás – Jegyzetömb és lokális tárolás

„Készíts egy böngészőben futó jegyzetömb alkalmazást HTML + JavaScript segítségével, amely automatikusan menti és visszatölti a szöveget.

Funkciók:

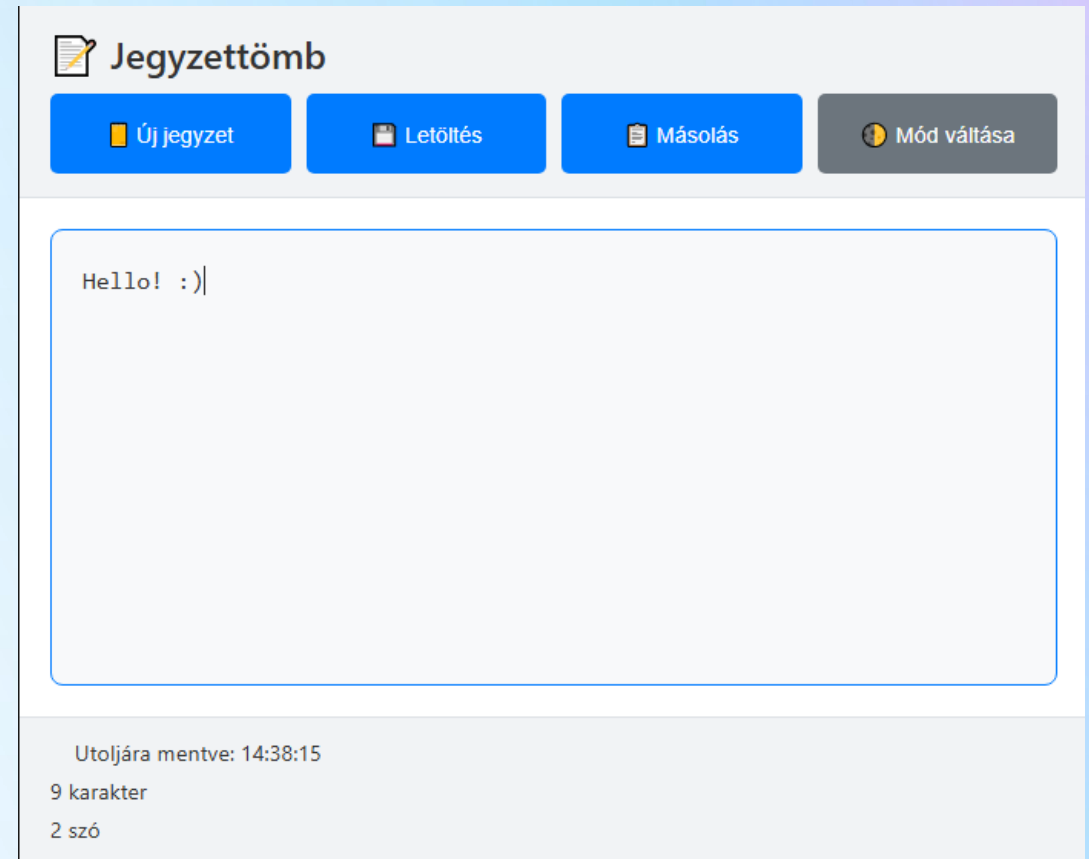
1. Nagy textarea a szöveg szerkesztésére (teljes képernyő magasság nagy része)
2. Automatikus mentés localStorage-ba:
 - Minden gépelés után 1 másodperc késleltetéssel mentse el a tartalmat
 - Használj setTimeout/clearTimeout kombinációt a debounce-hoz
3. Oldal betöltésekor automatikusan töltsd vissza a mentett szöveget
4. Jelenítsen meg egy státusz sort, amely mutatja:
 - Mikor történt az utolsó mentés (pl. „Utoljára mentve: 14:35:22”)
 - Karakterek száma
 - Szavak száma (space-szel elválasztva)
5. Gombok:
 - „Új jegyzet” – törli a tartalmat (megerősítés után)
 - „Letöltés .txt fájlként” – letölti a szöveget txt fájlként
 - „Másolás vágólapra” – kimásolja a teljes szöveget
6. Dark mode kapcsoló (toggle gomb), ami szintén localStorage-ban perzisztens

Technikai részletek:

- localStorage kulcs: "notepad_content"
- localStorage dark mode kulcs: "notepad_darkmode"
- textarea automatikus átméretezés a tartalom szerint (vagy fixed full height)
- A mentés során vizuálisan jelezd (pl. zöld villogás vagy „Mentve” felirat)
- Használj modern CSS-t (flexbox / grid)
- Minden egy .html fájlban legyen
- Smooth animációk a dark mode váltáshoz

Példa státusz sor megjelenés:

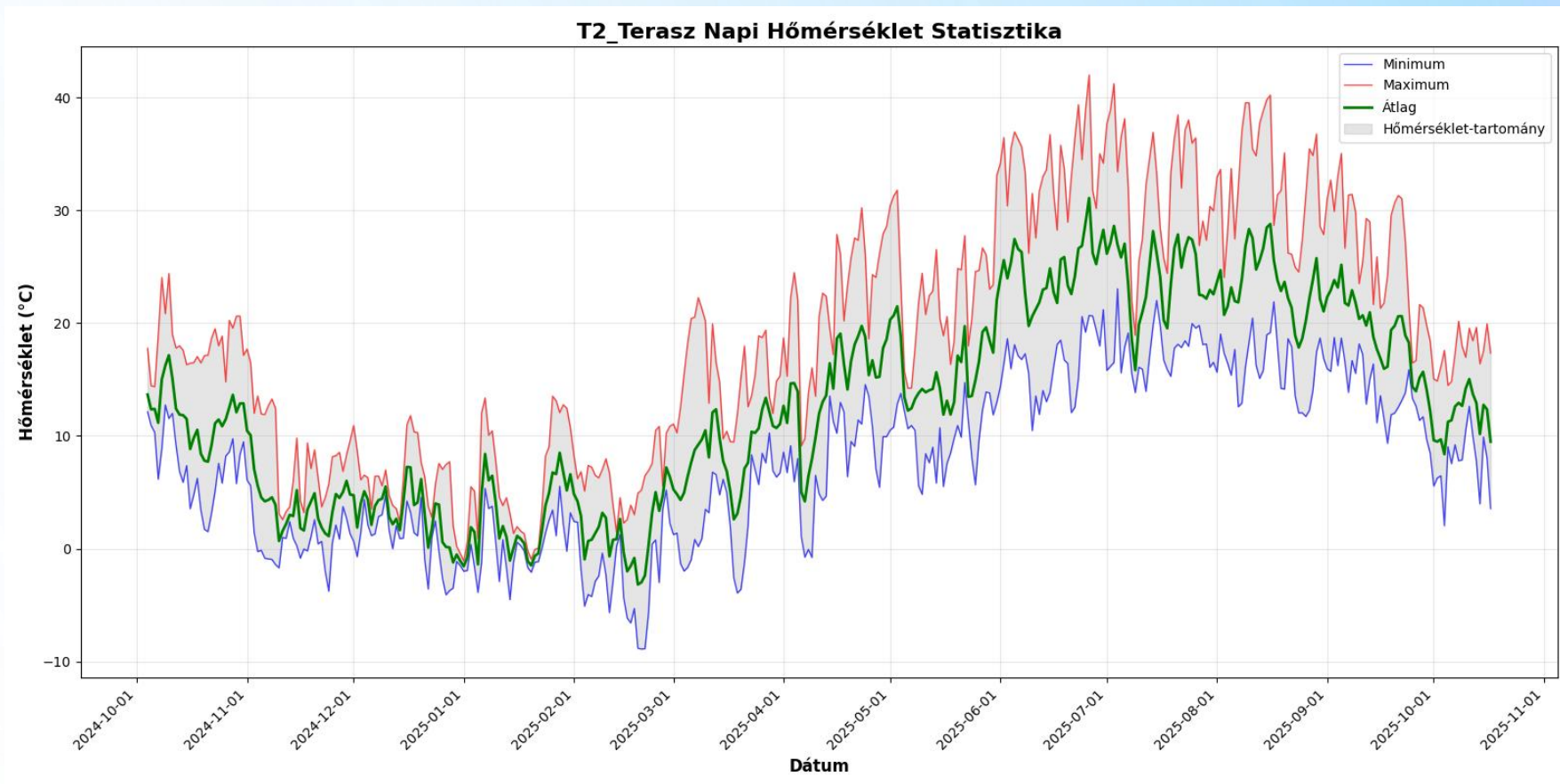
„✓ Utoljára mentve: 15:42:08 | 1.234 karakter | 187 szó”



CSV fájlból (hőmérséklet adatsor) diagram

Prompt:

„Készítsd egy diagramot a temperatures.csv fájlban lévő hőmérséklet adatokból. Minden naphoz mutasd meg a minimális, maximális és átlagos hőmérsékletet!”



Vegyes feladat ötletek...

Python / C# feladatok:

- 1. Pi közelítése Monte Carlo módszerrel:** Készítsünk LLM támogatással egy Python programot, amely véletlenszerű pontokat generál egy 1x1-es négyzetbe, és a pontoknak az egységnegyed körön belül / kívül esése alapján közelíti Pi értékét.
- 2. Szókitaláló játék (Wordle-szerű):** Írassunk meg LLM-mel Pythonban egy 5 betűs szókitalalós játékot, ahol zöld / sárga / szürke visszajelzést ad a program a tippekhez.
- 3. Számológép postfix notációval (RPN):** Írassunk meg LLM-mel C# nyelven egy „Reverse Polish Notation” számológépet (pl. „3 4 + 2 *” = 14).
- 4. Tic-Tac-Toe konzolos játék:** Készítsünk LLM segítségével Pythonban két játékosra Tic-Tac-Toe játékot ASCII grafikával és győzelem detektálással.
- 5. Szövegfájl tartalom statisztika:** Írassunk meg LLM-mel egy Python programot, amely egy szövegfájlhoz megadja a sorok, szavak, karakterek számát, és listázza a 10 leggyakoribb szót.

HTML + JavaScript single page alkalmazások:

- 1. Színkeverő játék:** Készíttessünk LLM-mel egy HTML + JS appot, ahol RGB csúszkákkal beállíthatunk egy célszínt, és pontot kapunk aszerint, mennyire közeli a beállított célszín egy véletlenszerűen generált színhez.
- 2. BMI kalkulátor grafikonnal:** Írassunk meg LLM-mel egy egyoldalas appot, amely kiszámolja a testtömegindexet, és színes skálán mutatja az eredményt (alulsúly / normál / túlsúly).
- 3. Kő-papír-olló játék statisztikával:** Készítsünk LLM támogatással HTML + JS játékot a számítógép ellen, amely nyomon követi a győzelmeket / vereségeket és mutatja a stratégiát.
- 4. Jegyzetömb localStorage-dzsal:** Generáltassunk LLM-mel egy böngészős jegyzetömb appot, amely automatikusan menti a szöveget a localStorage-ba, ami így frissítés után is megmarad.
- 5. Hex / RGB / HSL szín konverter:** Készítsünk LLM segítségével egy egyoldalas appot, amely valós időben konvertál a három színreprezentáció között, és megjeleníti a színt.

