

# Mesterséges Intelligencia

Módszertanok, technológiák,  
eszközök

# FONTOS

- Az alábbi anyag munkavázlat, hibákat tartalmazhat. Amennyiben hibát találnak, kérem, a portálon keresztül üzenetben jelezzék, hogy melyik heti előadás, vagy jegyzet melyik részében, milyen hibát véltek felfedezni!
- Az anyagok kizárólag a Széchenyi István Egyetem 2021-2022 tavaszi félévében Mesterséges Intelligencia kurzust felvett hallgatói számára készültek, kizárólag az adott félév kurzusaihoz használható fel!
- Az alábbi hivatkozásokon megnyitott minden fájl automatikusan begyűjti a hallgató különböző egyedi azonosítóit, mely alapján beazonosítható lehet. Ennek megfelelően a hivatkozásokat ne osszák meg egymással (különösen a kurzust nem hallgatókkal), mert abból az egyedi azonosítók visszakereshetők és a személyazonosság meghatározható!
- Az alábbi anyagra vonatkozóan minden jog fenntartva!
- Az anyagok bármely részének vagy egészének nyomtatása, másolása, megosztása, sokszorosítása, terjesztése, értékesítése módosítással vagy módosítás nélkül egyaránt szigorúan tilos!

# A lecke főbb témakörei

- MI fejlesztés főbb jellemzői, kérdései
- MI fejlesztés főbb megközelítései
  - Transfer process
  - Modelin process
  - Data-driven
  - CRISP-DM
- Kapcsolat klasszikus szoftverfejlesztési módszertanokkal
- MI rendszerek infrastruktúra elemei
  - Futtató hardverek
  - Érzékelő/beavatkozó hardverek
  - szoftverek

# Módszertanok, technológiák, eszközök

Korai MI fejlesztések

# Korai MI fejlesztések

- formalizmusok, következtetési mechanizmusok és a tudásalapú rendszerek (TAR) működtetésére szolgáló eszközök fejlesztése
- Jellemzően kis TAR megvalósítások
- különböző megközelítések megvalósíthatóságának vizsgálata
- Ígéretes eredmények ellenére a legtöbb esetben megbukott a (kereskedelmi) nagyobb méretű TAR fejlesztés

# Szoftverfejlesztés analógia

- A késő '60-as években hasonló folyamat volt megfigyelhető a tradicionális szoftverrendszerek fejlesztésénél
  - szoftver krízis
  - a kis akadémiai prototípusokat nem tudták skálázni nagyméretű, karbantartható kereskedelmi rendszerek szintjére
  - a szoftverfejlesztés/szoftvertechnológia (software engineering) területének megalapításához vezetett

# MI fejlesztés

- A tudásfejlesztés (~tudástechnológia, knowledge engineering) hasonló a szoftverfejlesztés/szoftvertechnológia területéhez
  - Cél: TAR fejlesztés egy mérnöki területté formálása
  - magába foglalja az építési és karbantartási folyamatok vizsgálatát, valamint a megfelelő módszertanok, nyelvek és céleszközök fejlesztését a TAR fejlesztéshez

# MI integrálása

- Az „MI” fejlesztése önmagában nem elég
  - Be kell építeni egy olyan szoftver- és hardverkörnyezetbe, amely hasznossá teszi
    - Ez tartalmazhat akár más MI megoldásokat is
- A rendszerintegrálás speciális változata



# MI fejlesztés: főbb módszertani megközelítések

- Átadási folyamat (transfer process)
- Modellezési folyamat (modeling process)
- Adatvezérelt (data-driven)

# Általánosságban az MI fejlesztésről

- Legfontosabb kérdések a fejlesztés közben:
  - Mi a jó kérdés?
  - Mi a legjobb megoldás a problémára?
  - Van adatunk hozzá?
  - Mennyire fog jól működni?
  - Mennyire vihető át a gyártásba?
- Szoftver és tervezési korlátok
- Bizonytalanság a problémában és a megoldásban
- Minden iteráció a probléma jobb megértéshez vezet

Módszertanok, technológiák, eszközök

Átadási folyamat

# Transfer Process

- Korai '80-as évek
- *„This transfer and transformation of problem-solving expertise from a knowledge source to a program is the heart of the expert-system development process.”*
- Az emberi tudás átvitele egy implementált tudásbázisba
  - alap feltevés, hogy a TAR fejlesztéséhez szükséges tudás már létezik és csak be kell gyűjteni
  - leggyakrabban szakértők interjúján keresztül történik, melynek során feladatokat oldanak meg
  - jellemzően a tudást szabályok formájában tárolták, amit egy interpreter dolgozott fel

# Transfer Process

- Különböző tudásbázisok alapos vizsgálata kimutatta, hogy a szabályoknál (production rules) alkalmazott egyetlen reprezentációs formalizmus sem támogatta megfelelően a különböző tudástípusok reprezentálását
  - a tudásbázisok karbantartása emiatt nehéz és időigényes lett
- A transzfer megközelítés csak kisméretű, jellemzően prototípus rendszerek fejlesztésére volt alkalmas
- Felismerték, hogy maga az alapfeltevés hibás
  - nem csak a meglévő (explicit) tudás szükséges
  - fontos a tacit (rejtett/hallgatólagos) tudás is

# A tudás/ismeret típusai

- Explicit
- Tacit
- Beágyazott
  - eszközökben, folyamatokban, kultúrában, rutinban

# Módszertanok, technológiák, eszközök

## Waterman – tudástechnológia

*Fő forrás: Sziray József Dr. - Varga Ágnes: Szakértői rendszerek.  
Győr : Széchenyi István Egyetem, 2006. 151 p. [elektronikus jegyzet  
(pdf) ]*

# Waterman: tudástechnológia

- Szakértői/tudás alapú rendszerek fejlesztése (1986)
  - Lényegében hasonló folyamatok a modern rendszereknél is
- 6 fázis
  - Probléma felmérése
  - Adatok és tudás megszerzése
  - Prototípus rendszer kifejlesztése
  - Teljes rendszer kifejlesztése
  - Rendszer kiértékelése, átdolgozása
  - Rendszer integrálása, karbantartása



# Waterman: tudástechnológia

- A fázisok határait esetenként nehéz meghúzni
- Természetes része az állandó iteratív átalakítás
- Kritikus a megfelelő szereplők, adatok és eszközök megválasztása
- Klasszikus (régi-módi) MI-k esetén még ma is használható
  - Főként szakértői rendszerek
- A főbb folyamatok mai gyakorlatban is jelen vannak
  - De megjelent az agilis szemlélet
    - Hatékony (humán)erőforrás kihasználás
  - Modern megoldások specifikus igényei miatt helyenként eltérhet

# Tudástechnológia: a probléma felmérése

- A probléma jellege?
  - Milyen típusú a probléma?
    - Osztályozás, kiválasztás, csoportosítás, diagnózis stb.

# Tudástechnológia: a probléma felmérése

- A probléma jellege?
  - Milyen típusú a probléma?
    - Osztályozás, kiválasztás, csoportosítás, diagnózis stb.
    - Hatással van az alkalmazandó erőforrások (eszközök és emberek) körére

# Tudástechnológia: a probléma felmérése

- A probléma jellege?
- Projekt résztvevők?
  - Két kritikus résztvevő
    - Tudásmérnök (knowledge engineer)
    - Területi szakértő (domain expert)

# Tudástechnológia: a probléma felmérése

- A probléma jellege?
- Projekt résztvevők?
  - Két kritikus résztvevő
    - Tudásmérnök (knowledge engineer)
      - Aki képes:
        - » megtervezni,
        - » megépíteni,
        - » tesztelni
      - egy intelligens rendszer
    - Területi szakértő (domain expert)

# Tudástechnológia: a probléma felmérése

- A probléma jellege?
- Projekt résztvevők?
  - Két kritikus résztvevő
    - Tudásmérnök (knowledge engineer)
    - Területi szakértő (domain expert)
      - Az adott szakterületen
        - » Széleskörű tudással rendelkezik
        - » Kellő tapasztalattal rendelkezik
        - » Képes a feladatok megoldására az adott területen

# Tudástechnológia: a probléma felmérése

- A probléma jellege?
- Projekt résztvevők?
- Projekt tárgya?
  - Célkitűzés
    - Versenyelőny, költségek csökkentése,  
döntések/termék/szolgáltatás minőségének  
javítása stb.

# Tudástechnológia: a probléma felmérése

- A probléma jellege?
- Projekt résztvevők?
- Projekt tárgya?
- Szükséges erőforrások?
  - Minden, ami a rendszer megépítéséhez szükséges
    - Hardver infrastruktúra
    - Szoftver infrastruktúra
    - Szaktudás és adatforrások
      - Szakértők, szakkönyvek, kézikönyvek, adatbázisok, minták stb.
  - Anyagi fedezet



# Tudástechnológia: az adatok és tudás megszerzése

- Problémakör jobb megismerése
  - Adatok begyűjtése, elemzése
  - Új ismeretek szerzése
- Rendszer tervének pontosítása az új ismeretek alapján
- Az adatok gyakran heterogén forrásból származnak
  - Minőség, felépítés eltérhet
  - ➔ Szükség lehet az adatok kondicionálására
- Az alkalmazott eszközöktől függően eltérő jellegű adatokra van szükség

# Tudástechnológia: az adatok és tudás megszerzése

- Az adatok begyűjtése során is keletkezhet új ismeret
- Ismeretszerzés
  - Iteratív folyamat
  - A területhez kapcsolódó szakirodalom megismerése
    - Alaposabb megértés érdekében szakértői interjúk
      - Nehéz folyamat
      - Tipikus esetek bemutatása
        - » Megoldással, részletes magyarázattal a következtetés menetére vonatkozóan
- Meghatározható absztrakt szintű megoldási stratégia
- A prototípusnál használandó eszközök kiválaszthatók

# Tudástechnológia: az adatok és tudás megszerzése

- Tipikus problémák a beszerzett adatokkal
  - Inkompatibilis adatok
    - A különböző forrásból származó adatok kódolása eltérhet
    - Az adatforrás és a feldolgozáshoz használt eszköz által elvárt kódolás eltérhet
    - Az adatok transzformációja nagy körültekintést igényel

# Tudástechnológia: az adatok és tudás megszerzése

- Tipikus problémák a beszerzett adatokkal
  - Inkompatibilis adatok
  - Inkonzisztens adatok
    - Nem összefüggő adatok
    - Az adatforrások eltérő tartalmú adatokból állnak
      - Például közösségimédia, keresőmotor használat és TV nézési adatok kombinálása
    - Az adatforrások eltérő pontosságúak

# Tudástechnológia: az adatok és tudás megszerzése

- Tipikus problémák a beszerzett adatokkal
  - Inkompatibilis adatok
  - Inkonzisztens adatok
  - Hiányos adatok
    - Üres adatmezők
      - Például opcionális adatok
      - Sérült adatok
      - Mulasztás miatt hiányzó adatok
    - Kezelhető
      - Elvetjük a hiányos rekordokat
      - Pótoljuk a hiányzó adatmezőket

# Tudástechnológia: prototípus rendszer kifejlesztése

- Ez már önmagában is egy intelligens rendszer létrehozása
- A teljes rendszer kicsiben megépített változata
- Kimondottan a probléma megértésének, a koncepció és a megoldás tesztelése a cél
  - Jól közelítettük meg a problémát?
  - Értjük a probléma „viselkedését”?
  - Megfelelő eszközöket használunk?
  - Megfelelő/elegendő adatot használunk?
- Szükség lehet a szakértő bevonására

# Tudástechnológia: prototípus rendszer kifejlesztése

- A különböző megoldások eltérő tesztelési stratégiát kívánnak meg
- A tesztesetek általában már megoldott esetek adatait tartalmazzák
  - A rendszert azonos bemeneti paraméterekkel látjuk el és megfigyeljük a viselkedését
- Előfordulhat, hogy teljesen új alapokra épített prototípust kell fejleszteni az eredmények alapján, ez természetes folyamat

# Tudástechnológia: teljes rendszer kifejlesztése

- A prototípus rendszer fejlesztése során szerzett tapasztalatok alapján
- Főbb lépései
  - Tervezés
  - Ütemezés
  - Költségvetés elkészítése
  - Teljesítmény kritériumok definiálása
- Újabb adatokkal és ismeretekkel kell bővíteni a rendszert
- Megfelelő (felhasználói) interfész(ek) kialakítása



# Tudástechnológia: a rendszer kiértékelése és átdolgozása

- A klasszikus rendszerekkel szemben nem jól definiálhatók a problémák és megoldásuk
- A kiértékelés a szoftver céloknak való megfelelés ellenőrzése
  - Megrendelői/felhasználói elégedettség
- Formális kiértékelés jellemzően minta adatok alapján
  - Az elfogadási kritériumok a prototípus fejlesztésének végén kerülnek meghatározásra
  - Ha az eredmények nem elfogadhatók, a rendszer teljes revíziójára lehet szükség

# Tudástechnológia: a rendszer integrálása és karbantartása

- A fejlesztési ciklus végső fázisa
- Szoftver-környezetbe való illesztés
  - Megfelelő interfészek
- Karbantartási program
- A felhasználó számára biztosítani kell a rendszer üzemeltethetőségét és karbantarthatóságát
  - Ide értjük a rendszerbe épített tudás „fejlesztését” is

# Módszertanok, technológiák, eszközök

Modellezési folyamat

# Modeling Process

- A TAR építés modellezési tevékenységnek is tekinthető
- Olyan számítógépes modell létrehozása, melynek célja olyan problémamegoldási képesség megvalósítása, ami felér a terület szakértőjével
- Nem az cél, hogy lemásolja/szimulálja a szakértő kognitív folyamatait, hanem az, hogy olyan modellt hozzunk létre, ami hasonló eredményeket hoz
- A szakértő tudása lehet rejtett/nem tudatos, amit nem tud megfogalmazni (tacit), így nem érhető el közvetlenül, de a tudás beszerzése során építeni kell rá

# Modeling Process

- A modellezési szemlélet következményei
  - egy modell csak a valóság egy megközelítése
    - végtelen folyamat, melynek célja az elvárt viselkedés lehető legjobb megközelítése
  - A modellezés egy ciklikus folyamat
    - új megfigyelések révén finomítani, módosítani kell a meglévő modellt
  - a modellezés függ a tudásmérnök személyes értelmezésétől
    - ez a folyamat gyakran hibás és a modell kiértékelése a valósághoz viszonyítva elengedhetetlen a megfelelő modell létrehozásához (visszacsatolás)
    - a modellezési folyamat minden fázisban módosíthatónak kell lennie a modellnek
- A modellezési szemlélet jellemzői
  - A rendszer/folyamat mély ismerete szükséges
  - Nem lehet végtelen komplexitást kezelni
    - Esetenként egyszerűsítés szükséges
  - Költséges és időigényes

# További említésre méltó modellek és módszerek

- Problem Solving Methods (Clancey)
  - Cover-and-Differentiate
  - Propose-and-Revise
  - SALT (PSM: Propose-and-Revise)
- Role-Limiting Methods (RLM)
  - Configurable Role-Limiting Methods
- Knowledge Acquisition and Documentation Structuring (KADS)
  - CommonKADS
- Model-based and Incremental Knowledge Engineering (MIKE)
- PROTÉGÉ-II
- ...

# Módszertanok, technológiák, eszközök

Adatvezérelt folyamat

# Data-driven megközelítés

- Jellemzően gépi tanulás módszerek
  - Olyan rendszer építése, ami minta esetek alapján próbál megfelelő megoldást találni
- Nagy mennyiségű adat szükséges
  - Igazán Big Data
  - ...dolgoznak rajta, hogy ez csökkenthető legyen, de nem minden esetben megvalósítható
- Számos módon megvalósítható
  - A leggyakoribb a neurális hálók
- Előnye, hogy nem kell szakértő által megfogalmazott tudást formalizálni



# Gépi tanulás életciklus

- **Projekt célkitűzésének definiálása**
  - Üzleti cél meghatározása
  - Területi szakértelem gyűjtése
  - Modell kritériumok rangsorolása
  - Megvalósíthatóság, kockázatok és siker kritériumok
  - Döntés a folytatásról
- **Adatok felkutatása és begyűjtése**
  - Megfelelő adat felkutatása
  - Feltáró adatelemzés végrehajtása
  - Hiányosságok feltárása és pótlása
  - Feature engineering
- **Adat modellezése**
  - Változók kiválasztása
  - Különböző modell-jelöltek kialakítása
  - Modellek validációja és a megfelelő kiválasztása
- **Feldolgozás és közlése**
  - Modell értelmezése
  - A modellben feltártak átadása
- **Implementálás, dokumentálás, karbantartás**
  - A modellt alkalmazó rendszer implementálása
  - A modellezési folyamat dokumentálása
  - Modell megfigyelési és karbantartási rendszer kidolgozása

Módszertanok, technológiák,  
eszközök

CRISP-DM

# CRISP-DM

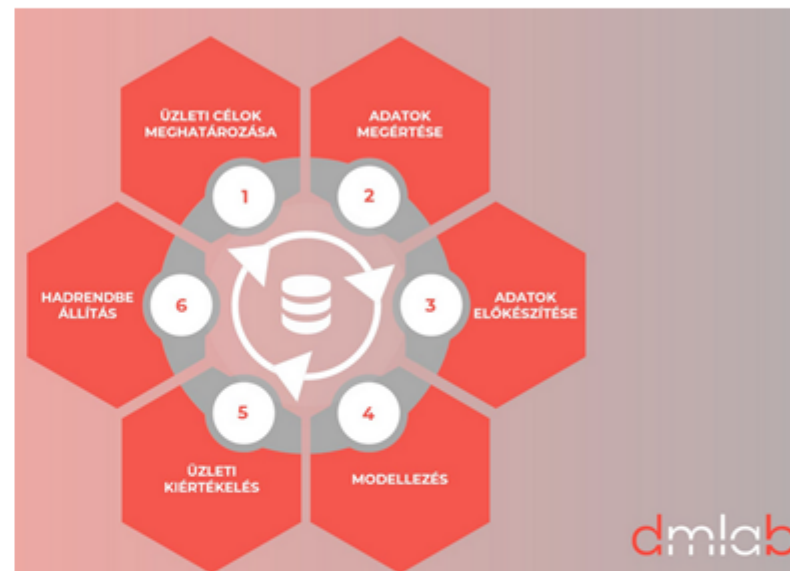
- Cross-industry standard process for data mining
- <https://dmlab.hu/blog/crisp-dm-modszertan-mi-az-es-hogyan-hasznald/>
  - „A CRISP-DM módszertan az egyik leggyakrabban használt módszertan, ami iránytűként szolgál abban, hogyan lehet egy üzletileg sikeres adatelemzési projektet véghezvinni”

# CRISP-DM

- „Egy adatelemzési projekt sikerességéhez számos komponensre van szükség.
- Tudnod kell:
  - Mi a valós üzleti cél, aminek elérésében az adatelemzés segítséget nyújthat?
  - Hogyan lehet a fenti üzleti célt adatelemzési céllá fordítani?
  - Milyen adatok állnak rendelkezésre az elemzés létrehozásához?
  - Hogyan kell ezeket az adatokat felhasználni a leghatékonyabb és üzletileg is sikeres megoldás kialakításához?
  - Milyen modellezési technikákat érdemes a cél elérése érdekében használni?
  - Hogyan lehet egy adatelemzési megoldást üzletileg kiértékelni?
  - Hogyan lehet a megoldást felhasználni, integrálni az eddigi üzleti folyamatokba?”

# CRISP-DM

- „A módszertan 6 fő fázisa:
  - Üzleti célok meghatározása (Business Understanding)
  - Adatok megértése (Data Understanding)
  - Adatok előkészítése (Data Preparation)
  - Modellezés (Modeling)
  - Üzleti kiértékelés (Evaluation)
  - Hadrendbe állítás (Deployment)”



# Módszertanok, technológiák, eszközök

MI tesztelés

# MI tesztelési életciklus

- Követelmény elemzés
- Teszt tervezés
- Tesztek kidolgozása, adatok előkészítése
- Teszt végrehajtása
- Tesztelési ciklus zárása

# Módszertanok, technológiák, eszközök

MI fejlesztéshez használt  
népszerű szoftverek



# Caffe

- eredetileg a Berkeley egyetemen fejlesztett
- mélytanulási keretrendszer (deep learning framework)
- népszerű
  - sebesség
  - a GPU alapú számítások támogatása
  - gyakori modellek támogatása
- Caffe2
  - továbbfejlesztett változata
  - a Facebook adta ki
  - 2018 során a PyTorchba olvasztották

# PyTorch

- Python programozási nyelvre épül
- tudományos számítási csomag
- Alapvetően két fő célközönségnek szánják
  - a GPU alapú számításoknál
    - jellemzően General Purpose GPU (GPGPU)
    - NumPy csomag alternatívájának
  - Mélytanulási kutatásoknál
    - rugalmas
    - gyors

# TensorFlow

- Google kutatói által fejlesztett
- bonyolult numerikus számítások kezelésére alkalmas programkönyvtár
- egyaránt alkalmas kutatásra és termékfejlesztésre
- közkedvelt a gépi tanulás kutatók és fejlesztők között

# Cognitive Toolkit

- Microsoft
- nyíltforrású mélytanulási eszköztár
- kereskedelmi színvonalú fejlesztésekhez

# Keras

- Python alapú
- Nyíltforrású
- neurális hálózat programkönyvtár
- képes együttműködni más népszerű könyvtárakkal
  - például a TensorFlow
  - Microsoft Cognitive Toolkit.

# DeepLearning4j

- Java nyelvhez
  - De támogat más JVM alapú nyelveket
    - Pl. Kotlin, Scala
- mélytanulási programkönyvtár

# Jupyter Notebook

- Nyíltforrású
- web alapú alkalmazás
- lehetővé teszi olyan dokumentumok létrehozását, melyek "élő" (futtatható) forráskódot, formulákat és egyéb elemeket is tartalmaz

# NLP Architect

- Intel
- Python programozási nyelvhez
- nyíltforrású programkönyvtár
- korszerű mélytanulási módszerek érhetőek el
  - a természetes nyelvfeldolgozás (Natural Language Processing, NLP) fókuszú



# Weka

- Ingyenes
- Nyíltforrású
- gépi tanulási szoftver
- számos adatbányászati feladat megoldására szolgáló algoritmust tartalmazó gyűjtemény
- Lehetőséget nyújt többek között az adatok előkészítésére, osztályozási, regressziós, valamint klaszterezési feladatokra is.

# OpenCV

- Nyíltforrású
- akadémiai és kereskedelmi célra egyaránt ingyenesen használható
- gépi látás és gépi tanulási programkönyvtár
- Több programozási nyelvet támogat
  - például, C++, Java, Python stb
- számos platformot támogat
  - Windows, Linux, Mac OS, iOS, Android

# R

- Nyíltforrású
- általános célú adatelemzésre szolgáló szoftverkörnyezet
  - főként statisztikai számításokra
  - lineáris és nemlineáris modellezésre
  - idősorok elemzésére
  - Osztályozásra
  - klaszterezésre
  - Grafikára
- több operációs rendszeren is fut
- használják a gépi tanulás területén
  - támogat olyan mélytanulási rendszereket, mint a TensorFlow, vagy a Keras.

# Módszertanok, technológiák, eszközök

MI szolgáltatások

# MI szolgáltatások

- A komoly MI megoldások gyakran igényelnek jelentős hardver infrastruktúrát
  - Ez számos kihívást jelent
    - Az eszközök költségesek
    - Megfelelő helyet kell találni az üzemeltetéshez
    - A karbantartás komoly erőforrásokat köt le
  - Megoldás lehet a felhő alapú technológia
- A tech-óriások számos felhő alapú MI szolgáltatást biztosítanak
- <https://colab.research.google.com/>

# Google

- A Google számos szolgáltatást nyújt
  - például a gépi tanulás, a nyelvfeldolgozás, vagy a gépi látás területén
  - <https://cloud.google.com/products/ai/>

# Google

- Különös figyelmet érdemel a Google Colab rendszere
  - Ingyenes
  - GPU használat lehetséges
  - saját MI projektek valósíthatóak meg
    - például Python nyelven olyan eszközök segítségével, mint a Jupyter Notebook, a TensorFlow, vagy a Keras.

# Microsoft

- Azure szolgáltatásába integrált
  - főként gépi látás, beszédfelismerés és gépi fordítás területekre fókuszál.
  - <https://azure.microsoft.com/hu-hu/overview/ai-platform/>



# IBM

- Az IBM Watson megoldására épül
- természetes nyelvfeldolgozási módszereket alkalmazó
  - felhő-alapú szolgáltatás
  - a gépi fordításon túl különböző nyelvi elemzéseket is lehetővé tesz.
- <https://www.ibm.com/watson/services/natural-language-understanding/>

# Módszertanok, technológiák, eszközök

MI megoldást futtató hardver

# MI megoldást futtató hardver

- Lényegében a feldolgozó egység
- Alapvetően két fő típus
  - On-device MI
    - CPU/FPGA/ASIC/GPU stb
  - Cloud MI
- A valóságban ez sokkal bonyolultabb témaköröket érint
  - On-premises
  - Edge devices
  - Fog computing
  - stb

# MI megoldást futtató hardver

- CPU
  - Bonyolult számítások gyorsan
  - Kevés párhuzamos szál
- GPU
  - Egyszerűbb számítások nem túl gyorsan
  - Rengeteg párhuzamos szál
  - nVidia jelentős előnyben a többi gyártóval szemben (CUDA/cuDNN)
  - AMD nagy hátrányban (OpenCL)
- FPGA
  - Field Programable Gate Array
  - Robosztus rendszer hozható létre egy chipen
  - Különböző funkcióval rendelkező blokkok hozhatók létre

# MI megoldást futtató hardver

- ASIC
  - Application Specific Integrated Chips
  - Hasonló az FPGA-hoz, de nem módosítható
    - ...sőt, sokkal lassabban módosítható
  - Masszív párhuzamosítás
  - Intel® Nervana™ Neural Network Processor
- SoC
  - System-on-Chip
- Quantum számítógép
  - (talán nem annyira) távoli jövő
  - Hatalmas jelentősége lenne a gépi tanulásban
    - ...és így a MI-ben

# Módszertanok, technológiák, eszközök

Beavatkozók és érzékelők

# Beavatkozó hardverek

- Elektromos beavatkozók
  - Villanymotorok
    - egyenáramú servomotorok
    - váltóáramú motorok
    - léptetőmotorok
  - Solenoidok
- Hidraulikus beavatkozók
  - Valamilyen hidraulikus folyadékot használnak
- Pneumatikus beavatkozók
  - Nagynyomású levegőt használnak

# Beavatkozó hardverek

- Léptető motor
- Servomotor
- Motorvezérlők



# Érzékelő hardverek

- Érzékelők főbb típusai
  - Látás/képalkotás
  - Hőmérséklet
  - Sugárzás
  - Távolság
  - Nyomás
  - Pozíció
  - Részecske
  - Mozgás
  - Fém
  - Szint
  - Szivárgás
  - Páratartalom
  - Gáz és vegyi anyag
  - Erő
  - Áramlás
  - Anyagok felületi és felszín alatti hibák (Flaw)
  - Láng
  - Elektromosság