



Szoftver 2.0

Dr. Ferenc Rudolf

Tanszékvezető egyetemi docens

Szoftver 1.0

- ▶ Hogyan fejlesztünk szoftvert ma?
 - Mérnöki hozzáállás
 - Probléma beazonosítása
 - Kisebb részproblémákra bontás
 - Részproblémák megoldása algoritmusokkal
 - Megoldások rendszerbe integrálása



Szoftver 1.0 – működik!

- ▶ Fantasztikus eredményeket lehet elérni
 - Elektronikus fizetés
 - Digitális egészségügy
 - Intelligens városi közlekedés
 - Stb.



Szoftver 1.0 – működik, de ...

- ▶ ... nem minden probléma megoldására hatékony
- ▶ Sok valós problémánál (elvárt viselkedésnél) könnyebb pozitív és negatív tanuló adatokat gyűjteni, mint expliciten leprogramozni



Szoftver 1.0 – működik, de ...

▶ Példák

- Képfelismerés
 - 12 millió képpont: panda
- Beszédfelismerés
- Gépi fordítás
- Játékok
 - pl. AlphaGo Zero
- Stb.



Szoftver 1.0

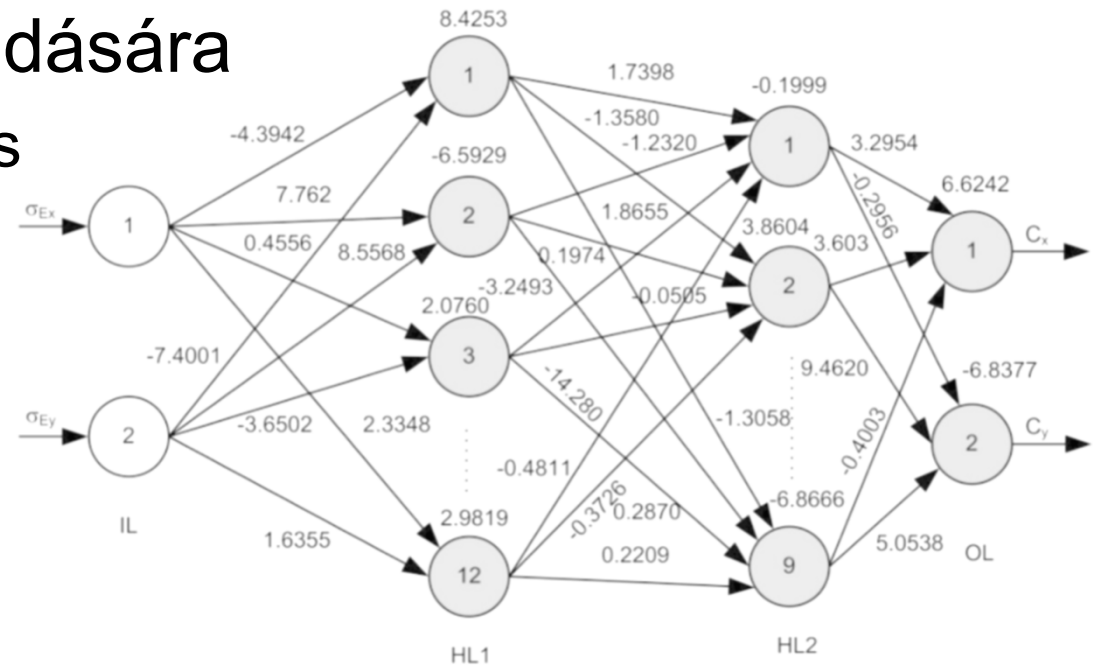
- ▶ Olyan programkód, amit emberek írnak
 - UML, C++, Python, XML, ...
- ▶ A programozók oldják meg a problémákat
- ▶ Az utóbbi 50+ év amiben még nagyrészt most is élünk



```
...
    for (String s : singleName.replaceAll("\\s+", "").split("\\s+")) {
        settings = singleName.replaceAll("\\s+", "").split("\\s+");
        if (settings[0].compareTo(s) == 0) {
            if (name.compareTo("") != 0) {
                name += " ";
            }
            name += etr.getString(settings[1]);
        } else if (settings[0].compareTo("d") == 0) {
            if (name.compareTo("") != 0) {
                name += " ";
            }
            name += DateUtils.format(etr.getDate(settings[1]));
        } else if (settings[0].compareTo("n") == 0) {
            if (name.compareTo("") != 0) {
                name += " ";
            }
            name += etr.getDouble(settings[1]);
        } else if (settings[0].compareTo("f") == 0) {
            name += etr.getBoolean(settings[1]);
        }
    }
}
```

Szoftver 2.0

- ▶ Olyan programkód, amit valamilyen optimalizálás ír
 - Tipikusan mély neuronháló
- ▶ A programozók megtanítják a gépet a problémák megoldására
 - MI – Mesterséges intelligencia
- ▶ Új megközelítés
 - Még Wikipédia cikk sincs róla!



Források

<https://medium.com/@karpathy/software-2-0-a64152b37c35>

<https://vimeo.com/274274744>

Szoftver 2.0 – új munkakörök

- ▶ Tanulóadat karbantartó (*MI tananyag készítő*)
 - Címkézés / annotálás
 - Új címkék beazonosítása
 - Új adatok felvitele
 - Adatok tisztítása
 - Tanulópéldák egyensúlyozása
 - Adatok vizualizálása
- ▶ Modell építő (*MI tanár*)
 - Gépi tanulás
- ▶ + Hagyományos munkakörök Szoftver 1.0-hoz

Fejlesztői infrastruktúra

- ▶ Szoftver 1.0 IDE ✓
 - Kódszerkesztő
 - Hibakereső
 - Egységtesztelő
 - Performancia mérő
- ▶ Szoftver 2.0 IDE ?
 - Tanulóadat vizualizáló
 - Annotáló UI
 - Hibakereső?
 - Egységtesztelő?



Szoftver 2.0 előnyei

- ▶ Egyszerű és homogén számolások
 - Mátrixszorzás és küszöbölés vs. klasszikus programozási nyelv komplexitása
- ▶ Könnyebb szilikonba önteni
 - Kisebb utasításhalmaz
- ▶ Konstans futási idő
- ▶ Konstans memória felhasználás
 - Nincs dinamikus memória foglalás
- ▶ Hordozható hardverek között
- ▶ Skálázható a teljesítmény vs. pontosság



Szoftver 2.0 korlátjai

- ▶ Olyan modellt kapunk, ami működik, de nem tudjuk hogy hogyan!
- ▶ Kihívások
 - Hibázhat (ahogy a kézzel írt kód is)!
 - A modellek is támadhatók!



“panda”
57.7% confidence

+ .007 ×



“nematode”
8.2% confidence

=



“gibbon”
99.3 % confidence

Szoftver 2.0 korlátjai

- ▶ Olyan modellt kapunk, ami működik, de nem tudjuk hogy hogyan!
- ▶ Kihívások
 - Hibázhat (ahogy a kézzel írt kód is)!
 - A modellek is támadhatók!



Sebességkorlátozás 45-re

Miért csak most?

- ▶ Neuronhálók régóta léteznek (~60 éve), de inkább csak kutatásban
- ▶ Most már adott
 - Sok adat
 - Számolási kapacitás
 - pl. GPGPU
 - Új tanuló algoritmusok
 - pl. hiba-visszaterjesztés, gradiensereszkedés
- ▶ Gyakorlati alkalmazás lehetővé vált!





Fel vagyunk készülve a változásra?

Köszönöm szépen a figyelmet!