**Strojové učení (Machine learning)**

**Co to vlastně je?**

Učení, které se zabývá algoritmy a technikami, které umožňují počítači „se učit“

**Učení** – zefektivnění schopnosti se přizpůsobení v změnách okolního prostředí

**Proč?**

**Je používáno z důvodu, že stejné algoritmy nejsou možné naprogramovány přímo člověkem**

ML se prolíná s oblastmi statistiky v matematice

Využití ML je obrovské

* Biomedicínská informace
* Rozpoznávání řeči
* Rozpoznávání textu
* Predikce cen
* Rozpoznávání obrázku
* Predikce počasí

**Algoritmy se dělí**

**Podle způsobu učení**

* **Učení s učitelem**
* Pro vstupní data je určen správný (očekávaný) výstup
* Trénink algoritmu pomocí TRÉNINKOVÝCH dat a následné testování přesnosti pomocí TESTOVACÍCH dat
* Zlepšuje přesnost reakce na budoucí data
* GENERALIZACE (zobecnění)
* **Učení bez učitele**
* Ke vstupním datům není známý výstup
* Data nejsou rozdělená jako TRÉNINKOVÁ a TESTOVACÍ

A screenshot of a chat

Description automatically generated with low confidence

**Podle způsobu zpracování**

* **Dávkové**
* Všechna data požadují před začátkem výpočtu
* **Inkrementální**
* Dokážou se přiučit
* Upravit model po dostání nových dat bez přepočítání celého modelu od začátku

**Data**

Používáme nezávislá data k určení závislých

Nezávislá – Kritéria použitá k určení, zda bude vyhověno půjčce

Závislá – výsledek (bylo/nebylo vyhověno)

Závislá data jsou závislá na nezávislých

A picture containing text, screenshot, font, document

Description automatically generated

A picture containing text, screenshot, number, font

Description automatically generated

**17. Příprava dat, Chyby v datech a BIAS, Korelace a kauzalita**

**18. Využití regrese a klasifikace**

**Klasifikace**

**Co to je?**

**Přiřazení vstupních dat ke konkrétní třídě nebo kategorii na základě TRÉNOVANÉHO MODELU a VSTUPNÍCH DAT**

Snaha nalézt výsledek odpovídající vstupním datům

**Proč?**

Najít funkci, která bude dělat náš požadavek

Například

* Určení spamu v EMAILU
* Spam
* Ne-spam
* Rozpoznání zvuku zvířat
* Kočka
* Pes
* Kráva
* Diagnostika chorob

V **KLASIFIKACI** je časté užití algoritmů jako jsou

* Rozhodovací stromy
* Nejbližší soused
* Logistická regrese
* Neuronová síť

**Logistická klasifikace**

Nejen rozhoduje z jaké kategorie je předmět

* ale rozhoduje **do jaké MÍRY** je předmět **DANÁ KATEGORIE**
* Mějme parametr šišatost -> Hledáme funkci, která převádí šišatost na hruškovitost, podle které můžeme nadále určit, zda se jedná o Jablko/Hrušku

**A picture containing text, line, diagram

Description automatically generated**

#### **Jak funguje rozřazení do více tříd?**

* Jednodušše vytvoříme více klasifikátorů (pro každou třídu jeden) a každého z nich se zeptáme (vypočítáme hodnotu).
* Klasifikátor, který vrátí nejvyšší pravděpodobnost je námi hledaný předmět.



**Random forest**

* Vygeneruje stanovený počet rozhodovacích stromů, který každý vrátí hodnotu
* **Třída s největším počtem hlasů vyhrává**
* **Výhodou +**
  + je nízká korelace mezi jednotlivými stromy
    - neovlivňují se navzájem
    - tedy pokud máme velké množství stromů = snižuje se šance na velkou chybu

**Regrese**

**Co to je?**

**Podobá se klasifikaci**

**Místo predikce kategorie/třídy se predikuje číslo na základě TRÉNOVANÉHO**

**MODELU a VSTUPNÍCH DAT**

Snaha nalézt výsledek odpovídající vstupním datům

**Proč?**

Najít funkci, která bude dělat náš požadavek s nejmenší chybovostí

**Například**

* Predikce poptávky produktu (na základě historických dat)
* Pravděpodobnost úspěchu marketingové kampaně
* Predikce růstu v odvětví
* Predikce ceny produktu (na základě stavu, kvality, poptávky, a další faktory)

V **REGRESI** je časté užití algoritmů jako jsou

* Rozhodovací stromy
* Náhodné lesy
* Lineární regrese
* Neuronové sítě

**A graph on a whiteboard

Description automatically generated with low confidence** **A picture containing sketch, line, drawing

Description automatically generated**

**Vstupní data Sestrojení funkce podle vstupních dat**

**Chybovost – vzdálenost jednotlivých bodů od dané funkce**

* Suma rozdílů všech skutečných hodnot a všech hodnot, kterou určil model
* **Uvádí se v kvadratickém tvaru** – zajišťuje **pouze kladná čísla** a ulehčuje **následnou derivaci**

Jak nalézt funkci s minimální chybovostí?

##### Říká se tomu SGD (Steepest gradient descent)

* Iteratiní algoritmus který hledá

**19. Využití umělých neornových sítí**

**Umělá neuronová síť**

Co to je?

Matematický model inspirovaný neuronálním systémem

Neuronové sítě se **skládají z neuronů**

* Jsou to vzor pro chování modelu
* Ty jsou propojené pomocí vah a mohou být organizované do vrstev

**Neuronové sítě jsou určené pro distribuované paralelní zpracování dat**

**Využití**

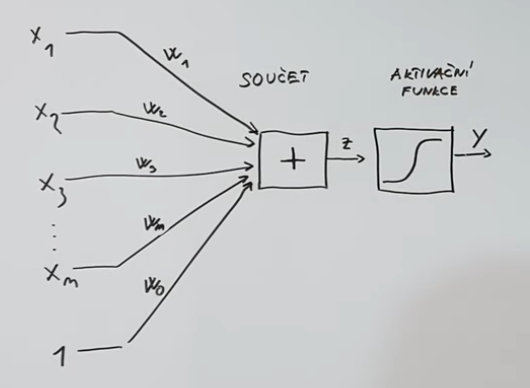
* Rozpoznávání obrázků a hlasu
* Vytváření umění

**Perecptron**

**Neuronová síť s jedním neuronem**

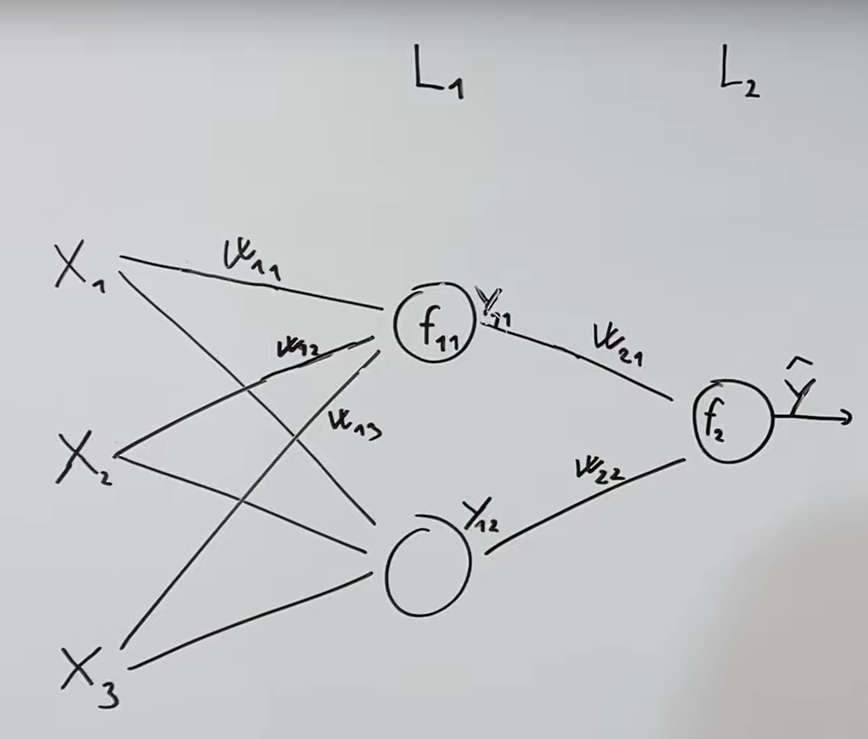
Jednoduchá matematická funkce, **která má N vstupů a jeden výstup**

* Každý vstup pronásobí určitou váhou
* Poslední vstup 1krát w0 = BIAS
* => LINEÁRNÍ POSUN (BIAS se vyskytuje vždy)
* Hodnota Z vstupuje do aktivační funkce (linerání, sigmoid, hyperbolický tangens..) a výstupem je hodnota Y
* Sigmoid generuje pravděpodobnost (číslo 0 - 1) = většinou poslední výstup
* Hyperbolický tangens (číslo -1 - 1) = vhodný pro vstup do dalšího neuronu, protože hodnota je normalizovaná



### Neuronová síť

* Posloupnost navzájem propojených vrstev neuronů
* Parametry na začátku neznáme, jejich úprava je předmětem trénování!

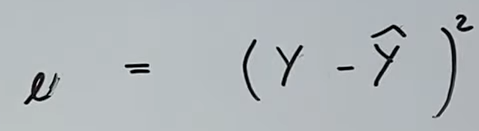
[](https://user-images.githubusercontent.com/84131825/233836365-2ad9e867-5d29-435a-a252-ed54b0b69cbf.png)

Dvouvrstvá neuronová síť - obrázek

* První vrstva (L1) tvořena 2 neurony, do kterých vstupují všechny vstupy X1...X3
* Druhá vrtsva (L2) tvořena 1 neuronem, do kterého vstupují výstupy z neuronů L2 \* příslušné váhy.

### Trénování sítě

* Hledáme kombinaci parametrů, která produkuje nejmenší chybu sítě
* Trénovací parametry jsou reprezentovány
  + Vstupními trénovacími hodnotami (X1...Xn)
  + Výstupní trénovací hodnotou (Y)
* Váhy (W) zpočátku vybíráme náhodně
* Chyba je u různých funkcí definovaná různě.. u regrese je například definovaná kvadrátem rozdílů 2 hodnot

[](https://user-images.githubusercontent.com/84131825/233836916-c0c44deb-b1fb-40a7-ab3d-c31cdad76e57.png)

Kvadratická chyba u regrese Y... hodnota kterou bychom chtěli Y^... hodnota kterou jsme dostali

#### Algoritmus zpětné propagace chyby

* Využívá dynamického programování (ukládá si vypočítané chyby, aby je později mohl znovu využít)
* U větších sítí se algoritmus upravuje..

##### Postup

* Vybereme vstupní hodnoty a na jejich základě spočítáme výstupní hodnotu
* Výstupní hodnotu porovnáme s hodnotou, kterou bychom chtěli => viz. chybová funkce
* Na základě chybové funkce spočítáme jak bychom měli upravit hodnoty, které do výstupního neuronu vstupují
* Tento algoritmus opakujeme dokud nedojdeme na začátek sítě - L1 => tzv. propagujeme chybu

## Deep learning

### Hluboké sítě

* Tak nazýváme sítě s velkým množstvím vrstev
* Neuronové sítě známe již od 60. let minulého století, nicméně vzhledem k nedostatečnému výpočetnímu výkonu jsme sítě s vysokým počtem vrstev považovali za zbytečné (neměli žádnou přidanou hodnotu). Souvisí to s algoritmem zpětné propagace -> Čím více vrstev, tím menší úpravy chyby u vstupu.
* Změnu přinesli grafické karty, které umožňovali vysokou míru paralelizace. Nyní i malá změna na vstupu udělala velký rozdíl a síť mohla být lépe natrénována.