

Databáze v chemické a forenzní analýze

Tereza Uhlíková

March 30, 2026

doc. Mgr. Tereza Uhlíková, Ph.D.

Ústav analytické chemie

skupina teoretické spektroskopie

místnost B4337

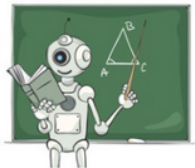
<https://web.vscht.cz/~uhlikovt/>

tereza.uhlikova@vscht.cz

O čem to bude?

		Téma přednášky	Téma cvičení
5	17.3.	Základní principy AI	otázky - odpovědi - učení
6	24.3.	Použití ML	Algoritmy ML
7	31.3.	Chemické databáze	Analýza spekter pomocí databází
8	7.4.	Chemické databáze	Konkrétní použití ML
9	14.4.	Konkrétní použití ML, Forezní databáze - zkušková témata	
10	21.4.	opakování, ERA, normalizace, tvorba MS databáze	
11	28.4.	Ivan Raich	
12	5.5.	Ivan Raich	

Učení s učitelem - bez učitele - posilované



Supervised Learning



Unsupervised Learning



Reinforcement Learning

Supervised Learning (Učení s učitelem)

Používá označená (labeled) data

K natrénování modelu potřebuje „dozor“ – tedy správné odpovědi

Má vstupní příznaky (features) a odpovídající štítky (labels)

Předpovídá nebo klasifikuje data na základě tréninku

Cílem je minimalizovat chyby v předpovědích

Mezi typické metody patří regrese a klasifikace

Vyhodnocuje se pomocí metrik jako přesnost (accuracy), preciznost (precision), citlivost (recall)

Příklad: filtrování spamů, diagnostika v medicíně, klasifikace obrázků

Unsupervised Learning (Učení bez učitele)

Používá neoznačená (unlabeled) data

K natrénování modelu nepotřebuje žádný dozor ani správné odpovědi

Pracuje pouze se vstupními příznaky

Objevuje vzory, vztahy nebo shluky v datech

Cílem je najít v datech vnitřní struktury nebo vztahy

Typické metody zahrnují shlukování (clustering) a snižování rozměrnosti (dimensionality reduction)

Hodnocení se zaměřuje na strukturu, podobnost nebo detekci anomálií

Příklad: doporučovací systémy, detekce anomálií

Jak databáze slouží strojovému učení

zdrojem trénovacích dat (bez dat není ML)

strukturované ukládání (tabulky, relace → vhodné pro modely)

dotazy a filtrování dat (SQL, indexy)

práci s velkými daty (big data)

ukládání anotací (labels) pro supervised learning

historická data → trénink predikčních modelů

Jak ML využívá databáze

Výběr relevantních dat (feature selection z databáze)

Automatické hledání vzorů v uložených datech

Predikce nad daty uloženými v databázi (např. klasifikace)

Integrace s DB systémy (např. in-database ML)

ML pro tvorbu databází

Automatické doplňování chybějících dat

Čištění dat – detekce chyb, outlierů

Automatická klasifikace a anotace dat

Extrakce informací z nestrukturovaných dat (text, spektra, obrázky)

ML pro správu databází

Detekce anomálií (chybné nebo podezřelé záznamy)

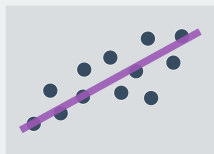
Optimalizace dotazů a výkonu databáze

Automatické shlukování a organizace dat

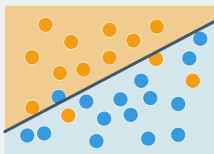
Predikce trendů v datech

Databáze poskytují strukturovaná data pro trénink modelů, zatímco strojové učení umožňuje tato data analyzovat, čistit, rozšiřovat a inteligentně využívat.

- Je daný e-mail spam? Která kytky je na fotce? \Rightarrow určování příslušnosti k jedné z několika předem daných kategorií = **klasifikace**
- Jak bude daný uživatel hodnotit daný film? Jaká bude teplota zítra odpoledne? \Rightarrow odhad číselné hodnoty = **regrese**
- Které zprávy řeší podobné téma? Které chemické látky se chovají podobně? \Rightarrow rozdělení příkladů do skupin s podobnými vlastnostmi = **shlukování**
- detekce případů, kdy bankovní účet nevyužívá jeho skutečný majitel \Rightarrow upozornění na podezřelé příklady, které se výrazně liší od zbytku dat = **hledání anomálií**
- seřazení výsledků vyhledávání, seřazení doporučených videí \Rightarrow uspořádání příkladů = **řazení**
- odpovídání na otázky, strojový překlad, vygenerování popsaného obrázku \Rightarrow vytvoření textu nebo obrázku na základě jiného textu nebo obrázku = **generování**
- hraní šachů a podobných her, řízení autonomního auta \Rightarrow hledání strategie pro optimální rozhodování = **řízení**



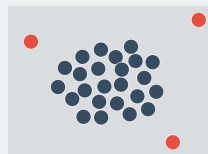
regrese



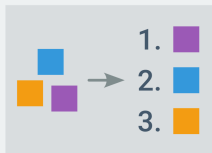
klasifikace



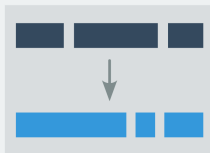
shlukování



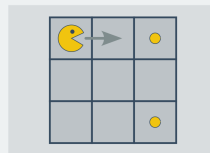
detekce anomálií



řazení (ranking)



generování



řízení

„Jak byste pomocí počítače poznali, jestli vzorek obsahuje drogu?“
Už víte, že s počítačem musíte mluvit jasně!

=> napište algoritmus

algoritmu znáte z ZFD, mají určité vlastnosti, využívají výrokovou logiku,
jasné a konečné

jeden problém lze řešit několika způsoby, třeba podle toho jaké máme na začátku informace

Klasifikace (co to je za látku?) - „Je to droga? Jaký typ?“ (supervised learning)

spektrum → heroin / kofein / neznámé

ano/ne (droga vs. nedroga)

„Tady máme správné odpovědi — algoritmus se učí rozpoznávat známé vzory.“

Regrese (kolik toho tam je?) - „Kolik drogy vzorek obsahuje?“ (supervised learning)

kvantifikace koncentrace ze spektra

intenzita signálu → množství látky

Odhad a predikce - odhad koncentrace látky - predikce stability vzorku

„Neříkáme co to je, ale kolik toho je nebo jak se to bude chovat.“

Shlukování (co je podobné?) - „Patří to k nějaké známé skupině?“ (unsupervised learning)

Hledání skrytých struktur - seskupování podobných vzorků - hledání nových typů látek

„Nemáme správné odpovědi — jen hledáme, co si je podobné.“

Detekce anomálií - „Je to něco neobvyklého / nového?“

nová syntetická droga, odhalení falšovaných vzorků

spektrum, které „nepasuje“ do databáze

„Hledáme něco, co nepatří mezi ostatní.“

Similarity search (nejbližší sousedé) - „Čemu je to nejvíc podobné?“
porovnání spekter s databází

„tohle vypadá jako fentanyl-like sloučenina“

Dimenzionální redukce (vizualizace) - „Jak data vypadají jako celek?“
oddělení skupin drog

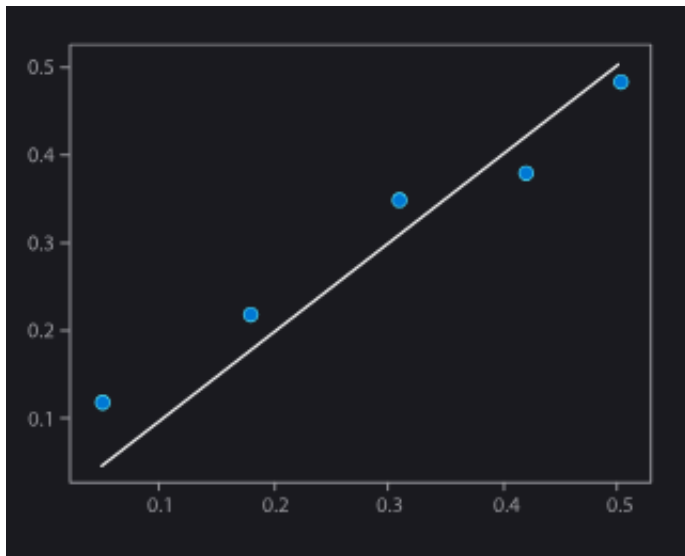
Multimodální přístup - „Spojím víc typů dat“
spektrum + struktura + databáze
robustnější identifikace

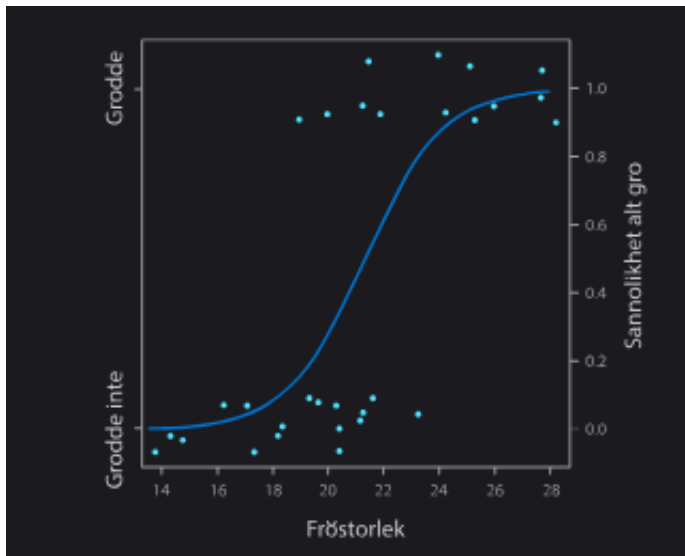
„Když jsou data složitá (obraz, signál), potřebujeme pokročilejší modely.“

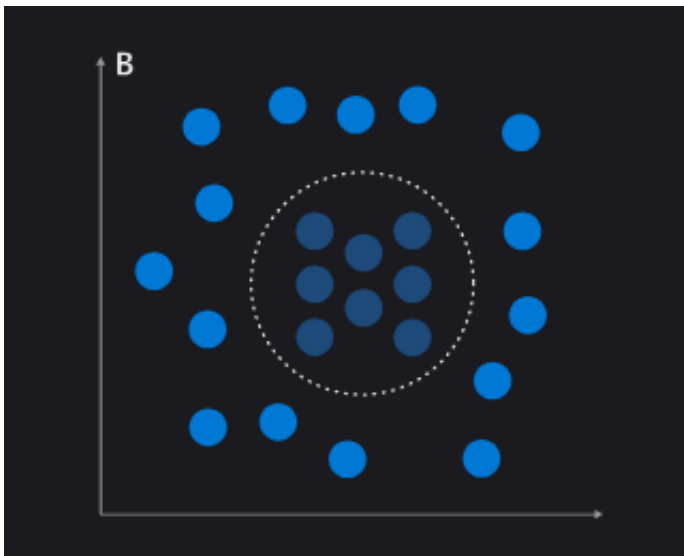
Různé ML algoritmy odpovídají na různé otázky

Problém	Typ úlohy	Typ algoritmu
Co to je?	klasifikace	rozhodovací strom, k-NN
Kolik toho je?	regrese	lineární model
Co si je podobné?	shlukování	k-means
Co je divné?	anomálie	outlier detection
Jak zjednodušit data?	redukce dimenze	PCA
Složitá data	hloubkové uč.	neuronové sítě

Lineární regrese







odhaduje, jaká je pravděpodobnost, že něco patří do určité třídy, na základě pozorovaných znaků = spočítá pravděpodobnost každé možnosti a vybere tu nejpravděpodobnější (IČ laborky)

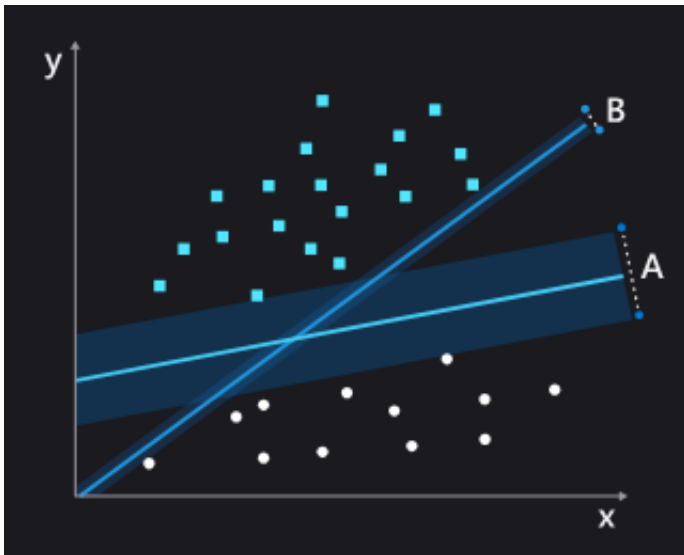
Příklad: Máme spektrum a ptáme se: „Je to droga? A jaká?“
Algoritmus se ptá: Jak často má heroin tento pík? Jak často má kofein tento pík? Jaká je pravděpodobnost, že dané spektrum patří heroinu? A pak řekne: „Nejpravděpodobnější je heroin.“

„naivní“ = předpokládá, že jednotlivé vlastnosti (např. píky ve spektru) jsou na sobě nezávislé

Výhody - jednoduchý, rychlý, funguje i na malých datech

Nevýhody - předpokládá nezávislost znaků; někdy není přesný u složitých dat

Metoda podpůrných vektorů (SVM)



Metoda podpůrných vektorů (SVM)

hledáme hranici mezi třídami -> co nejlépe oddělující hranici (maximální mezera = co největší vzdálenost mezi třídami)

hranice = hyperrovina

pouze některé body rozhodují - ty nejbliže hranici → ignorujeme většinu dat; soustředí se na „kritické případy“ ; Nelineární data - SVM umí i složité hranice

Příklad: oddělení dvou typů drog podle spektra; klasifikace toxických vs netoxických látek

Výhody: dobře funguje na komplexních datech; robustní

Nevýhody: hůře interpretovatelný výběr parametrů

Forezní interpretace

vhodné: spektrální data; komplexní chemické signály

problém: hůře vysvětlitelné u soudu

K-nearest neighbors (KNN)



K-nearest neighbors (KNN)

nový vzorek klasifikujeme podle nejbližších známých vzorků
najdeme **K** nejbližších bodů, dále rozhodne většina
- výpočet koncentrace když máte kalibrační křivku
vzdálenost mezi dvěma vzorky; podobná data = malá vzdálenost

Příklad: porovnání spekter látek
podobná spektra → stejná látka

Výhody - velmi jednoduchý; žádné učení (lazy learning), dobře funguje na podobnostech (např. spektra)

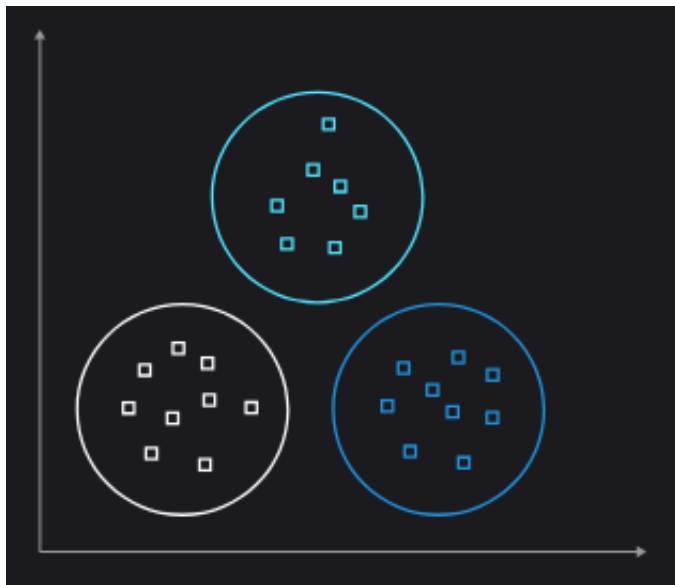
Nevýhody - pomalý pro velká data; citlivý na škálování dat; potřebuje definovat vzdálenost

Co ovlivňuje výsledek? Hodnota K

malé K → citlivé na šum

velké K → více „průměruje“

k-means



algoritmus pro shlukování (clustering)

Cíl: Rozdělit data do K skupin (clusterů) tak, aby si byla data ve stejné skupině co nejpodobnější. Neříká „co to je“, ale: „co je si podobné?“

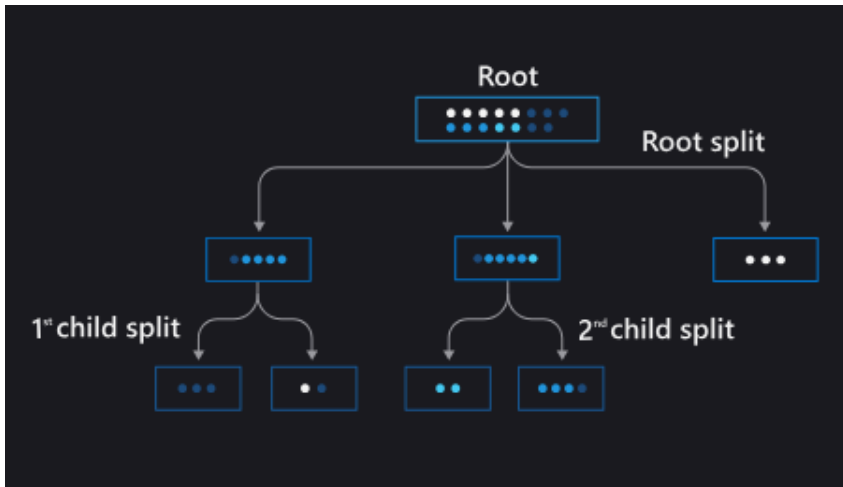
Náhodně zvolíme K středů (centroidů) → Každý bod přiřadíme k nejbližšímu středu → Přepočítáme středy (průměr bodů ve skupině) → Opakujeme, dokud se to „ustálí“

Výsledek: data rozdělena do K skupin & každý bod má „svůj cluster“

Příklad: spektra látek → rozdělení na skupiny; podobné drogy → jeden cluster; nové látky → mohou vytvořit nový cluster

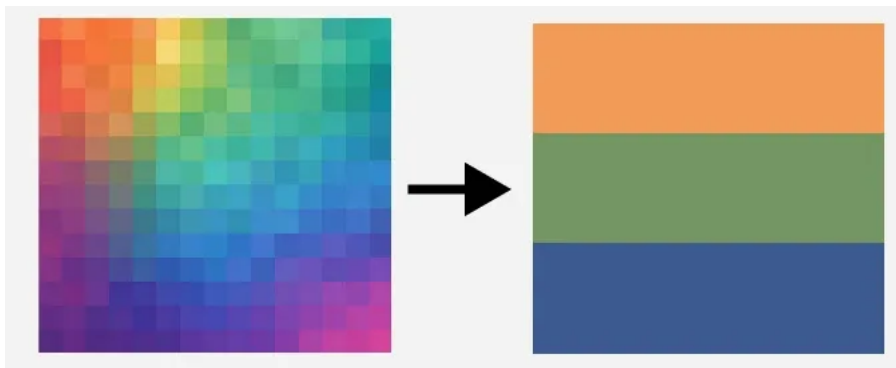
Výhody - jednoduchý a rychlý; nevyžaduje označená data (unsupervised learning); dobrý pro hledání vzorů

Nevýhody - musím znát K dopředu; citlivý na počáteční volbu středů; nefunguje dobře pro složité tvary dat



PCA (Principal Component Analysis)

Ted' jsme řešili klasifikaci. Ale co když vůbec nevíme, jaké třídy existují?



Problém - máme hodně proměnných; chceme snížit počet rozměrů & zachovat co nejvíce informace

Hlavní myšlenka - hledáme směry, kde je největší variabilita

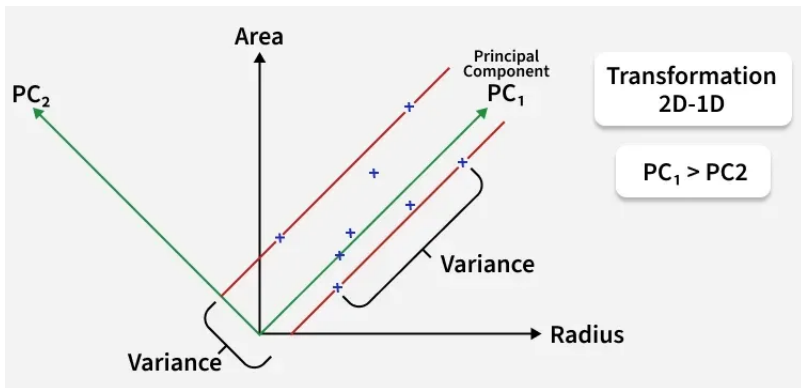
Výstup - nové osy = hlavní komponenty (PC1, PC2, ...)

vizualizace: skupiny látek, předzpracování pro další modely

PCA nám pomůže vidět data i když jsou původně velmi složitá

PCA neříká, co je správně jen nám pomáhá data zjednodušit

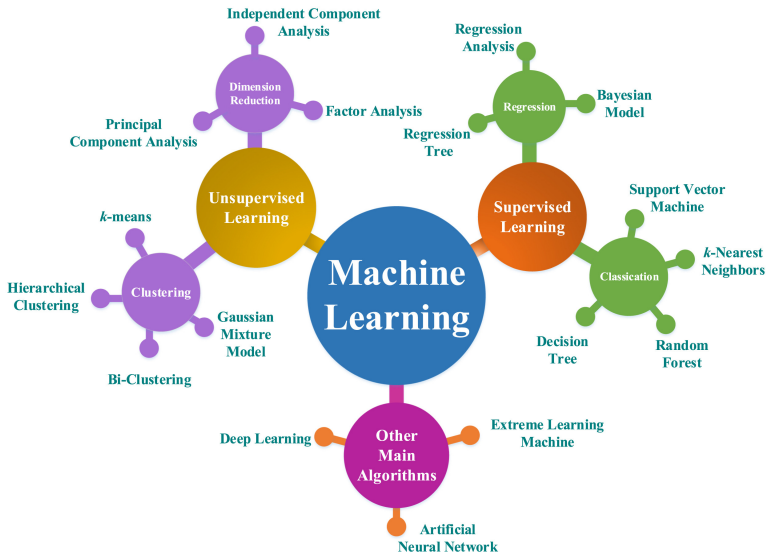
PCA (Principal Component Analysis)



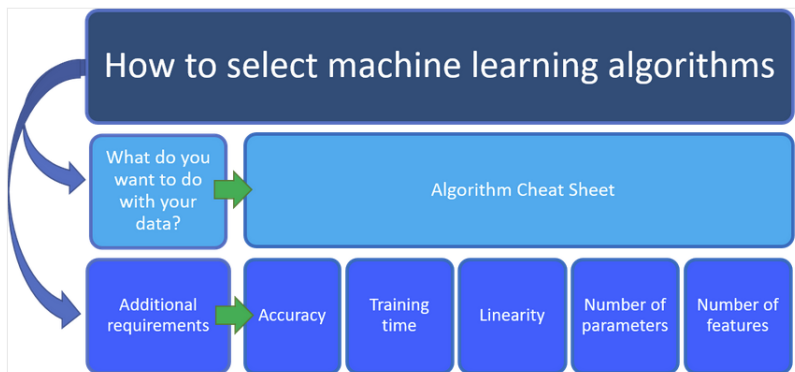
Různé ML algoritmy odpovídají na různé otázky

Problém	Typ úlohy	Typ algoritmu
Co to je?	klasifikace	rozhodovací strom, k-NN
Kolik toho je?	regrese	lineární model
Co si je podobné?	shlukování	k-means
Co je divné?	anomálie	outlier detection
Jak zjednodušit data?	redukce dimenze	PCA
Složitá data	hloubkové uč.	neuronové sítě

Algoritmy přehled



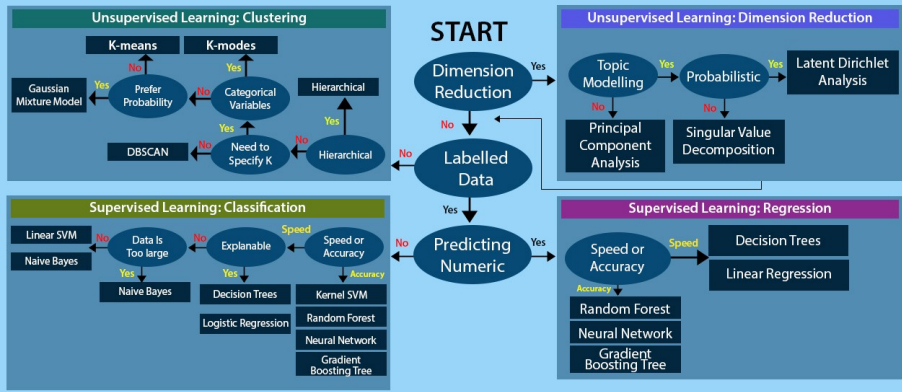
<https://azure.microsoft.com/cs-cz/resources/cloud-computing-dictionary/what-are-machine-learning-algorithms>



<https://azure.microsoft.com/cs-cz/resources/cloud-computing-dictionary/what-are-machine-learning-algorithms>



Machine Learning Algorithms Cheat Sheet



Jaký algoritmus byste použili? (Vyberte z: lineární regrese, logistická regrese, Naivní Bayes, k-NN, k-means, rozhodovací strom, SVM, PCA)

1. Máte IR spektrum a chcete určit, zda vzorek obsahuje drogu (ano/ne).
2. Chcete určit přesnou koncentraci kokainu ve vzorku ze spektra.
3. Máte databázi spekter a chcete najít, které látky jsou si podobné.
4. Máte nové spektrum a chcete najít nejpodobnější známou látku v databázi.
5. Chcete rozdělit neznámé vzorky do skupin bez znalosti jejich identity.
6. Máte mnoho proměnných ze spekter a chcete snížit jejich počet při zachování informace.
7. Chcete vytvořit jednoduchý model, který rozhodne podle několika pravidel (např. „pokud pík $> X$, pak droga“).
8. Chcete klasifikovat typ drogy na základě spektra, ale data nejsou lineárně oddělitelná.
9. Máte dataset s pravděpodobnostmi výskytu znaků (píků) pro různé látky.
10. Chcete rozdělit látky do 3 skupin podle podobnosti spekter.

11. Chcete předpovědět toxicitu látky (číselná hodnota) na základě její struktury.
12. Máte dataset spekter a chcete zjistit, zda existují přirozené skupiny látek.
13. Chcete klasifikovat vzorky (droga vs. nedroga) na základě více znaků, ale nechcete složitý model.
14. Máte malý dataset a chcete rychle klasifikovat nové vzorky podle podobnosti.
15. Chcete vytvořit model, který bude dobře fungovat i pro složitě oddělitelná data (např. překrývající se spektra).
16. Máte mnoho korelovaných proměnných a chcete odstranit redundanci.
17. Chcete model, který se dá snadno interpretovat (např. pro soudní znalce).
18. Máte spektrum a chcete spočítat pravděpodobnost, že patří konkrétní droze.
19. Chcete rozdělit látky podle jejich chemické podobnosti, ale nevíte předem počet skupin.

1. Máte IR spektrum a chcete určit, zda vzorek obsahuje drogu (ano/ne).
Logistická regrese, Binární klasifikace (ano/ne). Model odhaduje pravděpodobnost, že vzorek patří do třídy „droga“
2. Chcete určit přesnou koncentraci kokainu ve vzorku ze spektra.
Lineární regrese, Výstup je spojité číslo → typická regresní úloha.
3. Máte databázi spekter a chcete najít, které látky jsou si podobné.
PCA (případně k-means), PCA odhalí strukturu dat a podobnosti (redukuje dimenze → vizualizace).
4. Máte nové spektrum a chcete najít nejpodobnější známou látku v databázi.
k-NN, Hledá nejbližší sousedy → porovnání podobnosti.
5. Chcete rozdělit neznámé vzorky do skupin bez znalosti jejich identity.
k-means, Neřízené učení → hledání skupin bez štítků.
6. Máte mnoho proměnných ze spekter a chcete snížit jejich počet.
PCA, Redukce dimenze → odstranění redundance.

7. Chcete vytvořit jednoduchý model na základě pravidel.
Rozhodovací strom, „If-then“ logika → dobře interpretovatelný.
8. Data nejsou lineárně oddělitelná.
SVM, Umí nelineární hranice.
9. Máte pravděpodobnosti výskytu znaků pro látky.
Naivní Bayes, Pracuje přímo s pravděpodobnostmi znaků.
10. Rozdělit látky do 3 skupin podle podobnosti.
k-means, Zadán počet clusterů $K=3$.
11. Predikce toxicity (číselná hodnota).
Lineární regrese, Opět spojitý výstup.
12. Hledání přirozených skupin ve spektrech.
k-means / PCA, k-means = shlukování, PCA = vizualizace struktur.
13. Klasifikace droga vs. nedroga (jednoduchý model).
Logistická regrese / Naivní Bayes, Oba jsou jednoduché klasifikátory.
14. Malý dataset + klasifikace podle podobnosti.
k-NN, Nepotřebuje trénink → funguje dobře na malých datech.

15. Složitě oddělitelná data.

SVM, Robustní pro komplexní hranice.

16. Hodně korelovaných proměnných.

PCA, Odstraní korelace.

17. Model interpretovatelný pro soud.

Rozhodovací strom, Jasná pravidla → vysvětlitelnost.

18. Pravděpodobnost, že spektrum patří droze.

Naivní Bayes, Pravděpodobnostní model.

19. Rozdělení látek bez znalosti počtu skupin.

shlukování (DBSCAN nebo hierarchické shlukování)

20. Které proměnné nejvíce ovlivňují klasifikaci.

Rozhodovací strom, Ukazuje důležitost proměnných (feature importance).

Algoritmy učení s učitelem

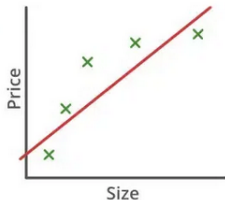
Algoritmus	Popis	Účel	Nejlepší použití
Lineární regrese	Předpovídá spojitý výstup na základě vstupních znaků.	Predikce číselných hodnot.	Predikce cen nemovitostí, odhad tržeb nebo příjmů.
Logistická regrese	Předpovídá pravděpodobnost, že vstup patří do určité třídy.	Klasifikace mezi dvěma třídami.	Detekce spamu, predikce nákupu zákazníka.
Rozhodovací stromy	Dělí data do podmnožin podle vstupních příznaků.	Zjednodušení rozhodovacího procesu.	Segmentace zákazníků, diagnostika nemocí.
Náhodné lesy (Random Forest)	Ensemble metoda kombinující více rozhodovacích stromů.	Zlepšení přesnosti a kontrola nad přeúčením.	Hodnocení úvěruschopnosti, predikce cen akcií.
SVM (Support Vector Machines)	Hledá hyperrovinu, která nejlépe odděluje třídy.	Maximalizace vzdálenosti mezi třídami.	Klasifikace obrázků, rozpoznávání rukopisu.
K-nejbližších sousedů (k-NN)	Předpovídá podle podobnosti s nejbližšími známými body.	Klasifikace a predikce na základě blízkosti.	Doporučovací systémy, detekce narušení (intrusion detection).
Naivní Bayes	Klasifikuje na základě pravděpodobnostních vztahů.	Předpokládá nezávislost jednotlivých znaků.	Filtrování spamu, analýza sentimentu.

Algoritmy učení bez učitele (Unsupervised Learning)

Algoritmus	Popis	Účel	Nejlepší použití
K-means shlukování	Rozděluje data do k shluků na základě vzdálenosti od průměru (centroidu).	Seskupování podobných datových bodů.	Segmentace trhu, shlukování dokumentů.
Hierarchické shlukování	Vytváří hierarchii shluků pomocí aglomerativního (zdola nahoru) nebo divisivního (shora dolů) přístupu.	Vytváření hierarchicky zanořených shluků.	Analýza genových dat (DNA), analýza sociálních sítí.
PCA (Principal Component Analysis)	Snižuje rozměrnost transformací dat do nového souřadnicového systému.	Redukce rozměrnosti dat.	Komprese obrázků, extrakce příznaků.
t-SNE (t-Distributed Stochastic Neighbor Embedding)	Nelineární redukce rozměrnosti pro vizualizaci vysokodimenzionálních dat.	Vizualizace složitých dat ve 2D/3D.	Vizualizace shluků, zkoumání vzorů ve velkých datasetech.
Apriori algoritmus	Vyhledává časté položkové množiny a vytváří asociační pravidla.	Objevování vztahů v rozsáhlých datasetech.	Analýza nákupních košíků, doporučovací systémy.

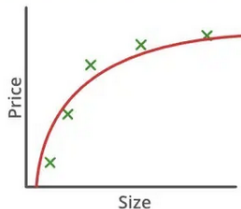
Algoritmy posilovaného učení (Reinforcement Learning)

Algoritmus	Popis	Účel	Nejlepší použití
Q-learning	RL algoritmus, který se učí hodnotu akcí v jednotlivých stavech pomocí Q-tabulky.	Naučit agenta optimální akce v daném prostředí.	Hraní her, robotika.
Deep Q-Networks (DQN)	Kombinuje Q-learning s hlubokými neuronovými sítěmi pro práci ve vysokodimenzionálních stavech.	Zvládnání složitých a rozsáhlých stavových prostorů.	Autonomní řízení, komplexní strategické hry.
Actor-Critic metody	Kombinují value-based a policy-based přístupy, aby dosáhly rovnováhy mezi průzkumem a využíváním.	Vyvážit exploraci a exploataci při rozhodování.	Real-time strategie, dynamická alokace zdrojů.



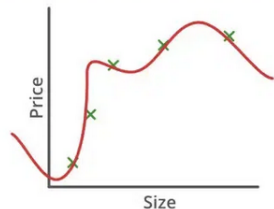
$$\theta_0 + \theta_1 x$$

High Bias
(Underfitting)



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

Low Bias, Low Variance
(Goodfitting)



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

High Variance
(Overfitting)

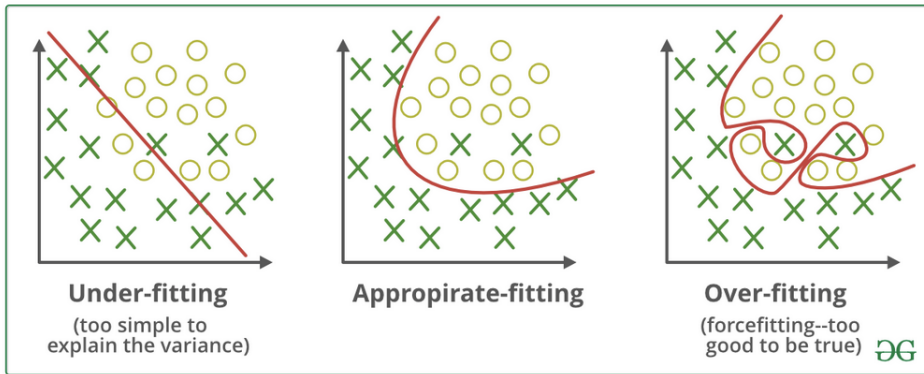


Underfitting (podfitování): Přímka, která se snaží aproximovat zakřivený dataset, ale nedokáže zachytit jeho skutečné vzory. Výsledkem je špatný výkon jak na trénovacích, tak testovacích datech.

Underfitting (podfitování): Přímka, která se snaží aproximovat zakřivený dataset, ale nedokáže zachytit jeho skutečné vzory. Výsledkem je špatný výkon jak na trénovacích, tak testovacích datech.

Overfitting (přeučení): Křivka, která se klikatí a prochází všemi trénovacími body. Na trénovacích datech funguje výborně, ale selhává na testovacích datech, protože se nenaučila zobecnit.

Appropriate Fitting (správné přizpůsobení): Křivka, která sleduje trend dat bez zbytečného komplikování, takže zachytí skutečné vzory v datech a dobře generalizuje.

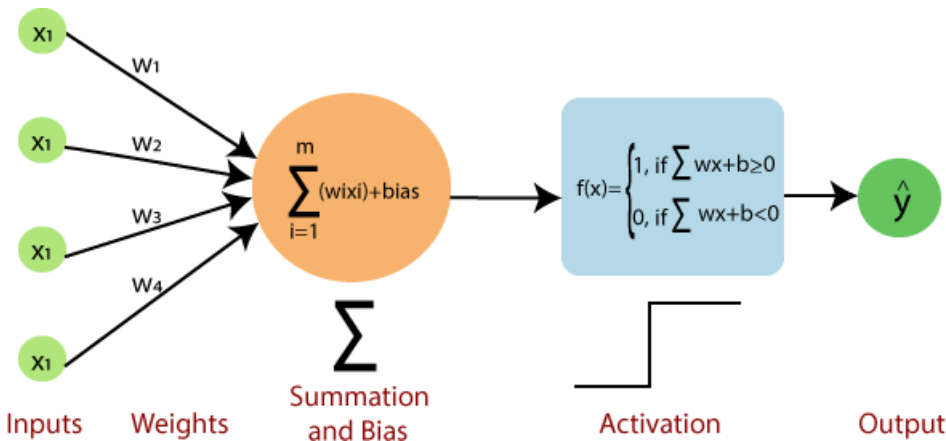


Neurovoné sítě

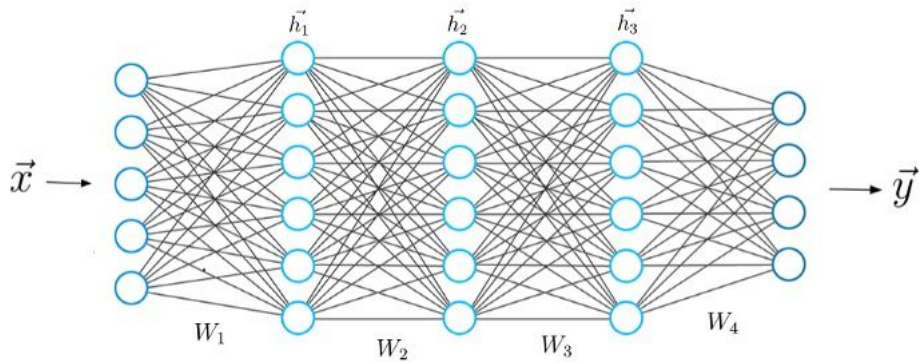


Perceptron = jeden neuron

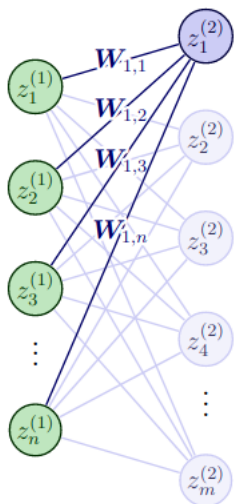
“The Perceptron: A Perceiving and Recognizing Automaton” in 1957.
několik vstupů x_i s vahami $w_i \rightarrow$ biasovaná suma všech vah \rightarrow aktivační funkce $f(x) \rightarrow$ jeden výstup Y



Více neuronů



vektor - tensor - matice



$$\begin{aligned}
 &= f \left(W_{1,1}z_1^{(1)} + W_{1,2}z_2^{(1)} + \dots + W_{1,n}z_n^{(1)} + b_1^{(2)} \right) \\
 &= f \left(\sum_{i=1}^n W_{1,i}z_i^{(1)} + b_1^{(2)} \right)
 \end{aligned}$$

$$\begin{pmatrix} z_1^{(2)} \\ z_2^{(2)} \\ \vdots \\ z_m^{(2)} \end{pmatrix} = f \left[\begin{pmatrix} W_{1,1} & W_{1,2} & \dots & W_{1,n} \\ W_{2,1} & W_{2,2} & \dots & W_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ W_{m,1} & W_{m,2} & \dots & W_{m,n} \end{pmatrix} \begin{pmatrix} z_1^{(1)} \\ z_2^{(1)} \\ \vdots \\ z_n^{(1)} \end{pmatrix} + \begin{pmatrix} b_1^{(2)} \\ b_2^{(2)} \\ \vdots \\ b_m^{(2)} \end{pmatrix} \right]$$

$$\mathbf{z}^{(2)} = f \left(\mathbf{W}^{(2)} \mathbf{z}^{(1)} + \mathbf{b}^{(2)} \right)$$

Typy neuronových sítí

Perceptron - Jednoduchý neuron - Binární klasifikace

MLP (Multi-Layer Perceptron) - Vícevrstvá síť - Klasifikace, regrese

CNN (Convolutional NN) - Pracuje s obrazy - Počítačové vidění

RNN (Recurrent NN) - Pracuje se sekvencemi - Text, časové řady

LSTM / **GRU** (Long Short-Term Memory networks) - Paměťové RNN - Překlady, chatboty

Transformery (GPT, BERT) - Nejmodernější modely pro NLP (Natural Language Processing) - Chatboti, analýza textu

Autoenkodéry - Kompresce dat - Detekce anomálií

GAN (Generative Adversarial Networks) - Generování nových dat - Deepfake, AI art

SNN (Spiking Neural Networks) - Napodobují mozkové neurony - Robotika, neurověda

OK, and how many neurons in each layer do you have or does GPT-3 have?

Good question! While OpenAI hasn't publicly disclosed the exact architecture of **GPT-4**, we have some estimates based on **GPT-3**, which can give us an idea.

GPT-3 Architecture Overview

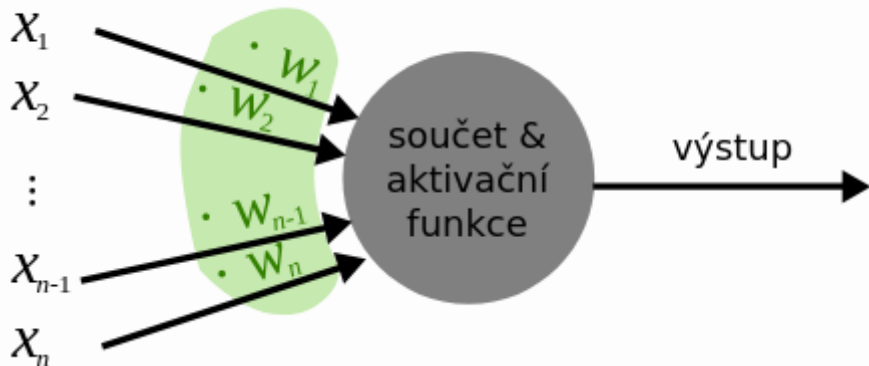
- **Number of layers (depth):** 96 (for the largest version, 175B parameters).
- **Number of attention heads per layer:** 96.
- **Hidden layer size (neurons per layer):** 12,288.
- **Total parameters:** 175 billion.

Each transformer layer in **GPT-3** has **12,288 neurons** in the feedforward network, which applies transformations to the input embeddings. Since GPT-4 is an improvement over GPT-3, it is likely to have **more neurons per layer** and possibly a **different architecture with optimizations** rather than just a brute-force increase in size.

Byl jednou jeden život - OKO - minuta 14:27

AI Cat Learns to Run <https://www.youtube.com/watch?v=aBp-3pmKNBY>

vstupy trénovatelné váhy



<https://jlibovicky.github.io/2020/10/10/Neuronove-site-a-strojovy-preklad.html>

Trénování neuronových sítí

vstupy sítě

obrázek, kde černé pixely mají hodnotu 1 a bílé pixely 0

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

$$x_1 = 0$$

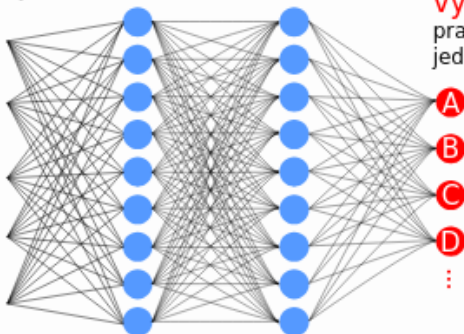
$$x_2 = 1$$

⋮

$$x_{15} = 1$$

$$x_{16} = 0$$

neurony ve skrytých vrstvách

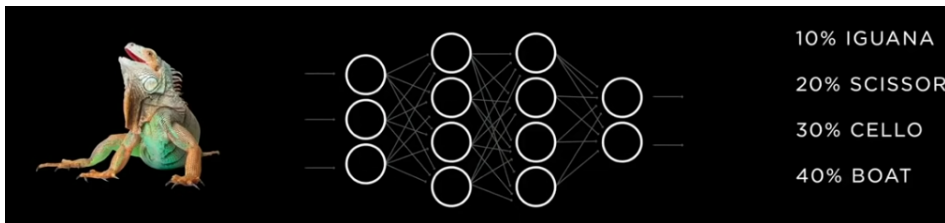


výstup sítě

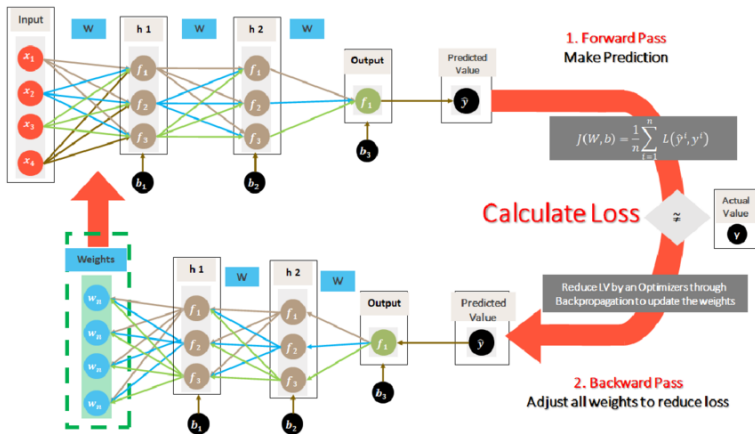
pravděpodobnosti jednotlivých písmen

A
B
C
D
⋮

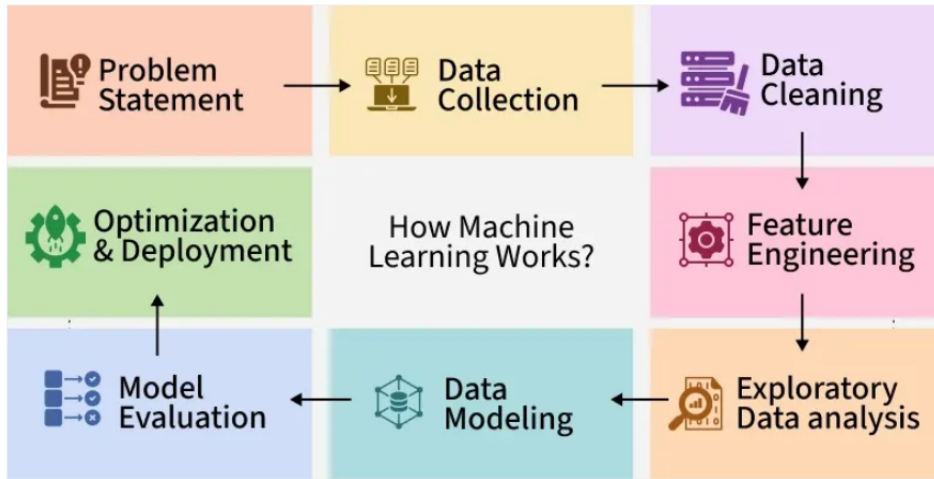
Trénování neuronových sítí



Zpětná propagace chyby



Jak probíhá učení



Proces trénování neuronové sítě

- 1 **Inicializace váh** – Na začátku jsou váhy náhodně inicializovány.
- 2 **Provádění zpětné propagace** – Provádí se propagace dopředu pro získání předpovědí.
- 3 **Výpočet ztráty (chyby)** – Vypočítá se ztráta (rozdíl mezi skutečným a předpovězeným výstupem).
- 4 **Použití zpětné propagace pro výpočet gradientů** – Vypočítá se gradienty, které ukazují, jak upravit váhy k minimalizaci ztráty.
- 5 **Aktualizace váh pomocí optimalizačního algoritmu** – Pomocí optimalizačního algoritmu (např. gradientní sestup) se váhy upraví.
- 6 **Opakování po více epochách, dokud nedojde k konvergenci** – Tento proces se opakuje po několika epochách, dokud model nezlepší své výstupy.
- 7 **Hodnocení na validační a testovací sadě** – Model se vyhodnotí na validačních a testovacích datech, aby se ověřilo, že dobře generalizuje a není přetížený.

Kolik je potřeba dat na natrénování

čím více tím lépe.

Lidský sval - tisíc opakování

obrázek psa - tisíce, miliony... Problém kde vzít data!

Proč až dnes?

dříve - pomalé počítače, málo neuronů, matematická nepřesnost v propagaci vah a tím nemožnost hloubkového učení

Artificial Neural Network Applications in Analysis of Forensic Science

[https:](https://onlinelibrary.wiley.com/doi/10.1002/9781119795667.ch3)

[//onlinelibrary.wiley.com/doi/10.1002/9781119795667.ch3](https://onlinelibrary.wiley.com/doi/10.1002/9781119795667.ch3)

Predicting the rotational dependence of line broadening using machine learning

<https://doi.org/10.1016/j.jms.2024.111901>

přednáška Jan Hrach - installfest 2024

<https://www.youtube.com/watch?v=o9TwtMywEuI>

série lekcí Jiří Materna - MLC

<https://www.youtube.com/watch?v=bo5nLkjYvMg>

historie

<https://nerdyelectronics.com/a-quick-history-of-ai-ml-and-dl/>

nejpoužívanější algoritmy strojového učení

https://s3.amazonaws.com/assets.datacamp.com/email/other/ML+Cheat+Sheet_2.pdf

Julia language in machine learning: Algorithms, applications, and open issues

<https://www.sciencedirect.com/science/article/pii/S157401372030071X?via%3Dihub>

trénování neuronových sítí

<https://www.kubicek.ai/neuronove-site/>

LSTM Inefficiency in Long-Term Dependencies Regression Problems

<http://dx.doi.org/10.37934/araset.30.3.1631>