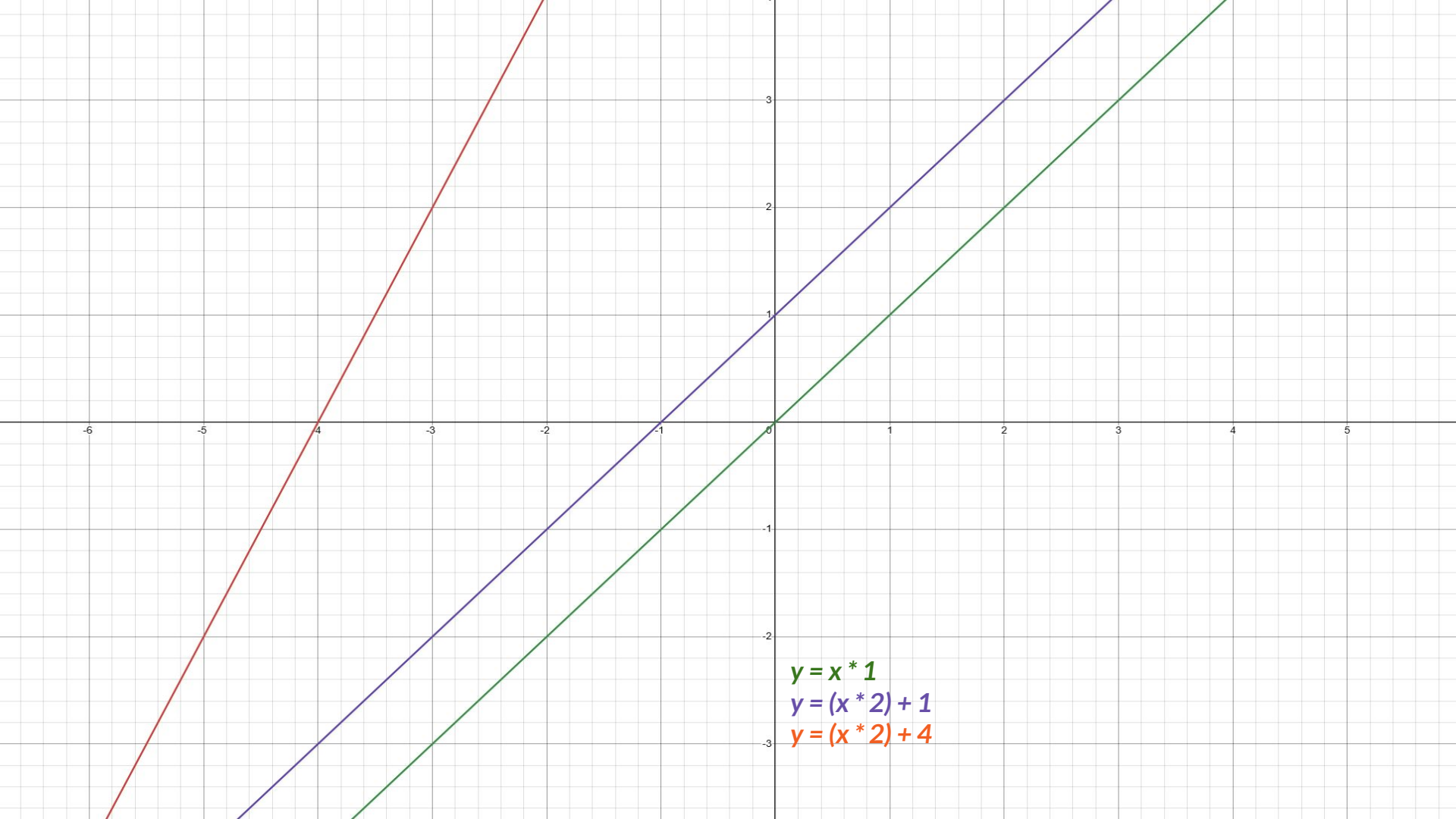
Modely pro predikci nelineárních vztahů

Johan Eigner R3.A

Úvod a teorie

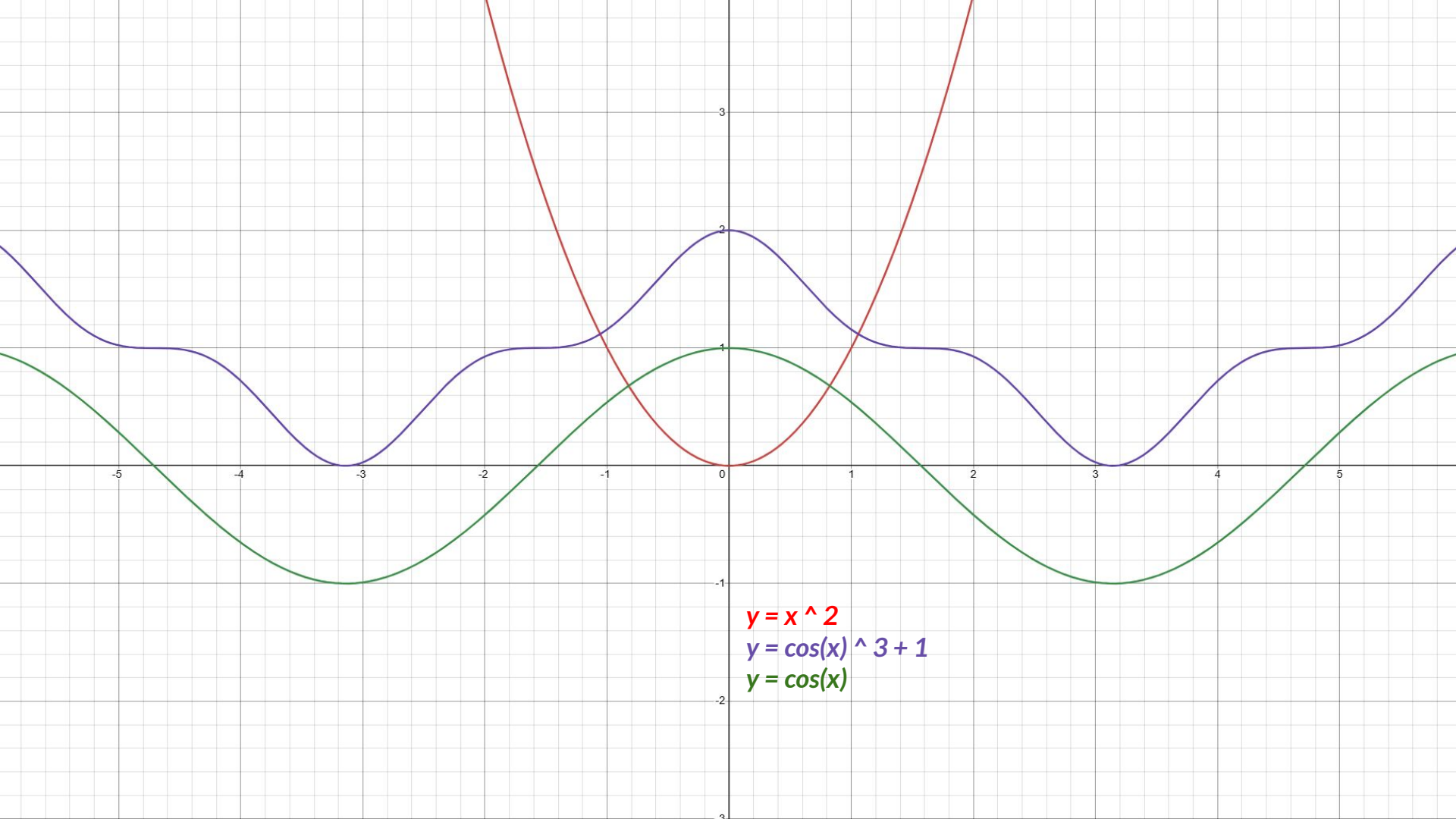
Lineární vztah

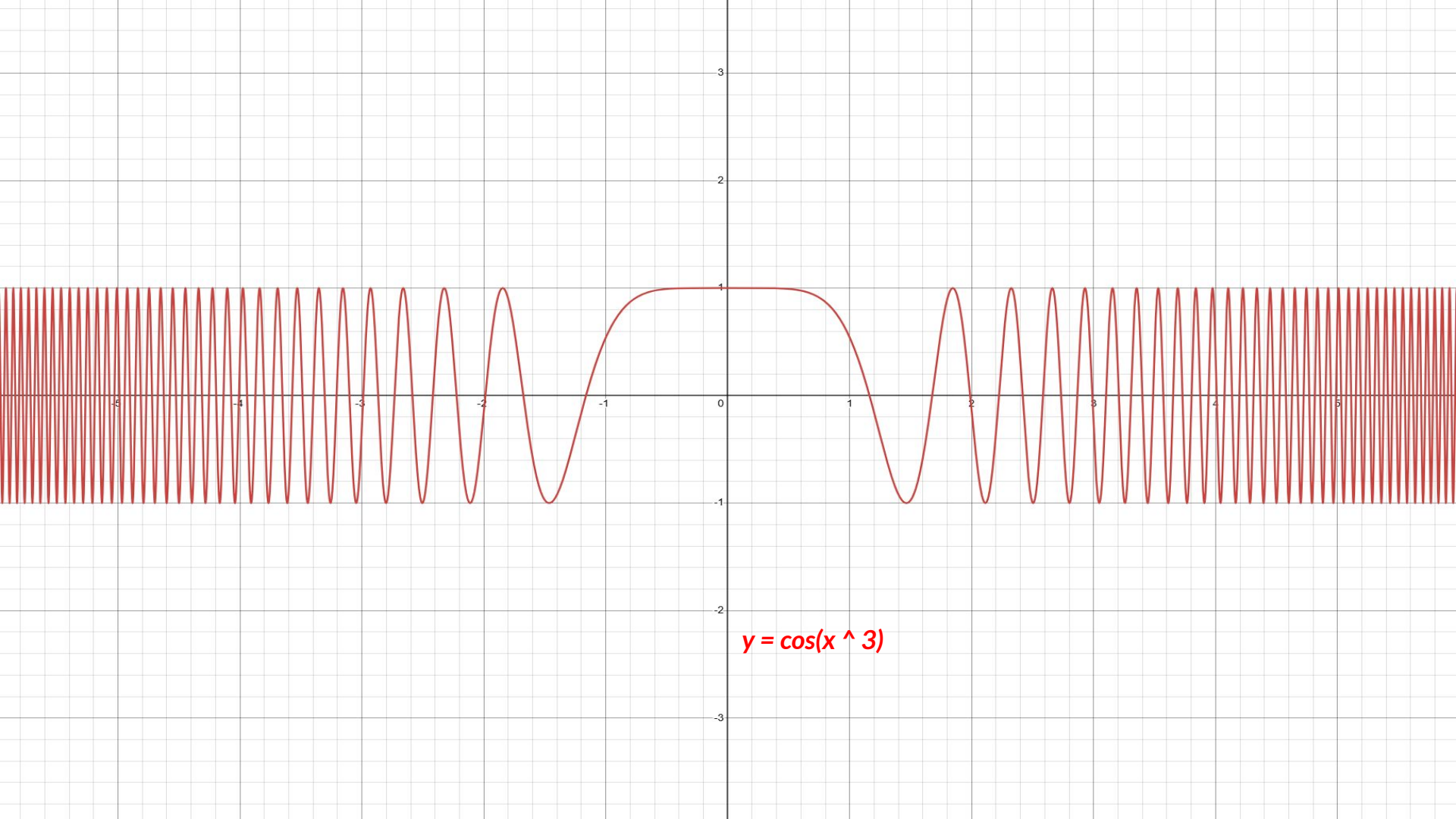
- x se mění v závislosti na y
- grafem bude vždy přímka
- například ***y = x * 2*** nebo ***y = (x + 3) / 7***
- každý lineární vztah lze vyjádřit jako ***y = a * x + b***
- pokud máme sadu x, y jsme schopni vztah pomocí rovnice odvodit
- *a* určuje sklon přímky, ***b*** určuje posun


y = x * 1
y = (x * 2) + 1
y = (x * 2) + 4

Nelineární vztah

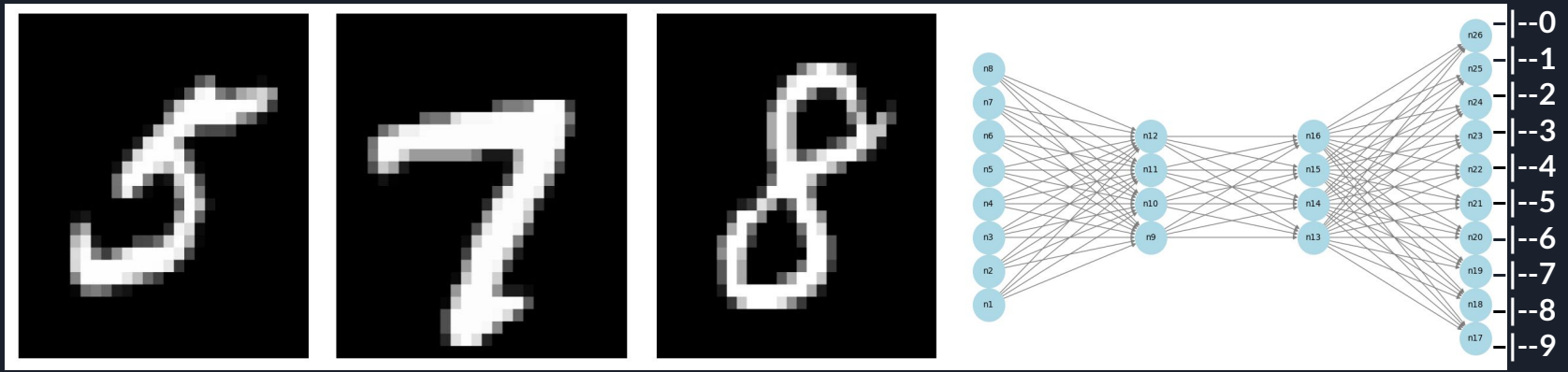
- grafem není přímka
- nelze vyjádřit jako ***y = a * x + b***
- například mocniny, goniometrické funkce (sinus, cosinus, tangens)
- u složitějšího nelineárního vztahu nebo u kombinace nelineárních vztahů je obtížné nebo nemožné je nějak matematicky odvodit ze sady x, y
- čím “křivější” graf daný vztah má, tím náročnější je na odhalení (platí i pro neuronové sítě)


y = x ^ 2
y = cos(x) ^ 3 + 1
y = cos(x)


3
2
1
0
-1
-2
-3
-5
-4
-3
-2
-1
1
2
3
4
5
y = cos(x ^ 3)

Nelineární úlohy

- bežnými algoritmy je řešení extrémně náročné / nemožné
- řešení není založeno na matematických principech
- graf takovéto úlohy by vypadal jako náhodná čmáranice
- řeší se pomocí různých typů neuronových sítí
- odvození znaků z rukopisu→ systém psaných číslic nevznikl na principech matematiky



Architektura neuronové sítě

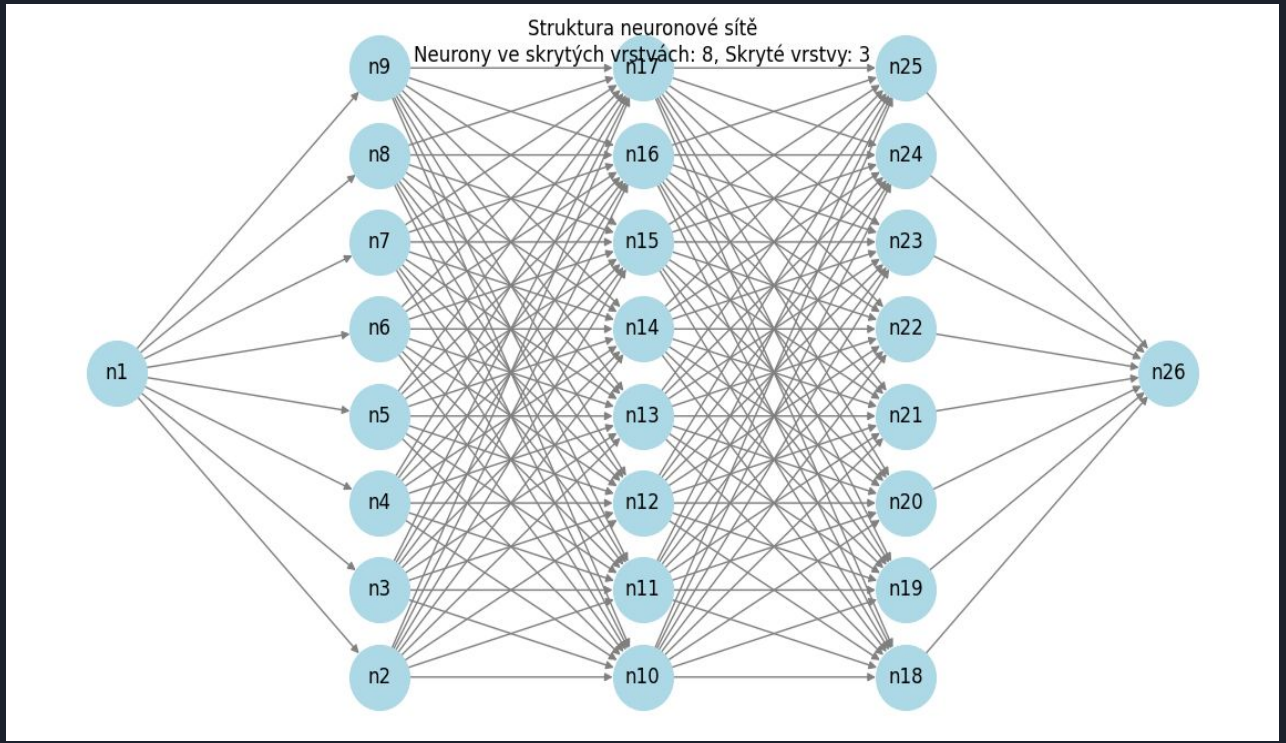
- původně založena na principech biologických neuronových sítí
- nyní se v některých případech původnímu konceptu vzdaluje → stále založena na neuronech, ale čím dál více je koncept modifikován pro menší výpočetní náročnost a lepší uzpůsobení pro danou úlohu
- několik základních architektur pro různé úlohy:
    - vícevrstvý perceptron → až v další části prezentace
    - konvoluční neuronové sítě → zpracování obrazu
    - rekurentní neuronové sítě → zpracování sekvenčních dat (data s časovou závislostí) - Google Translate, rozpoznání řeči, předpověď akciového trhu
        - využívá spoje mezi vzdálenými vrstvami
    - transformerové sítě → nejpokročilejší, Chat GPT (první ze všech), Google Bard, Microsoft Copilot

Vícevrstvý perceptron

- rozpoznání rukopisného textu
- hledání vzorů v datech
- automatizovaná výroba → detekce vadných kusů na základě snímku
- detekce různých procesů v medicíně → napojený na elektrody snímající mozek → detekce epileptických záchvatů
- jednodušší autonomní systémy → autopiloti v autech → detekce pruhů vozovky a udržování vozidla mezi nimi
- regrese (predikce následující hodnoty na základě hodnot předchozích)

Architektura a prvky vícevrstvého perceptronu

- vstupní vrstva
- skryté vrstvy
- výstupní vrstva
- aktivační funkce
- neurony
- váhy
- biasy
- inicializace vah a biasů
- algoritmy učení
- normalizace
- akcelerace



Vstupní vrstva

- každý neuron ve vstupní vrstvě přijímá jednu hodnotu, tedy počet informací se kterými síť pracuje = počet neuronů ve vstupní vrstvě
- pokud síť pracuje s ku příkladu fotografií, každý neuron ve vstupní vrstvě přijme hodnotu jednoho pixelu z fotografie
    - pokud síť zpracovává obrázek 28 * 28 pixelů ve vstupní vrstvě bude 28 * 28 = 784 neuronů
- každý neuron ve vstupní vrstvě je propojen s každým neuronem ve vrstvě následující

Skryté vrstvy

- vrstvy uvnitř sítě
- přijímají vstupy, provádí s nimi danou operaci a odesílají je do všech neuronů následující vrstvy
- počet neuronů ve skrytých vrstvách výrazně ovlivňuje složitost vztahu, který se síť může naučit a také výpočetní náročnost → čím více, tím náročnější na výkon

Výstupní vrstva

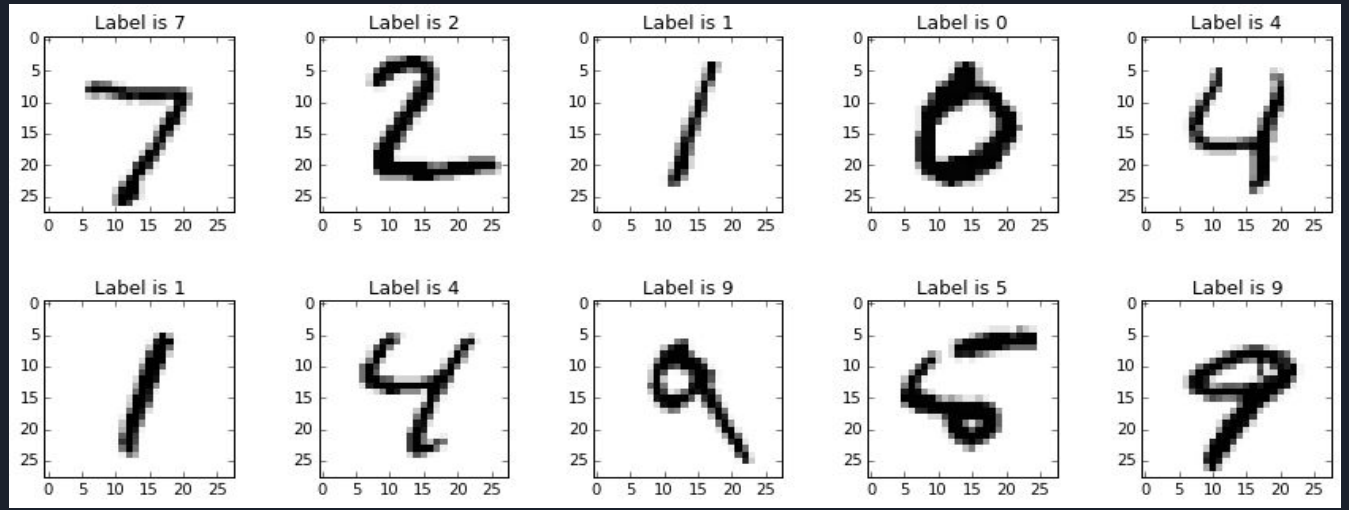
- počet neuronů závisí na úloze
- pokud je úlohou regrese → výstup bude jedna číselná hodnota → 1 neuron
- pokud má neuronová síť vybrat z více možností (například které číslo je na fotce), počet neuronů bude roven počtu možných odpovědí
  - výsledek každého neuronu bude pravděpodobnost, že číslo na fotce je číslo, které neuron reprezentuje
  - pokud není v síti chyba, nejvyšší pravděpodobnost by měl dát neuron, který reprezentuje číslici na fotce

Aktivační funkce

- stanovuje hranici
- pokud je hodnota vstup do neuronu pod touto hranicí, neuron se vypne a dál žádný signál nepošle
- průlomový koncept
- přináší nepravidelnost (nelinearitu) → pokud se některé neurony zdánlivě náhodně vypnou, síť umí pracovat i s nepravidelnými hodnotami, které neodpovídají matematickým principům
- díky aktivační funkci (zjednodušeně, svoji roli hrají i některé další faktory) jsou schopné neuronové sítě svých jedinečných výkonů

Aktivační funkce - příklad

- pokud síť dostane černobílý obrázek číslice, bílé pozadí nevzbudí v síti žádnou odezvu → neurony se neaktivují
- aktivační funkce umožní síti reagovat pouze na obrys číslice
- díky tomu je síť schopná “vidět / soustředit se na” důležité informace a zbytek ignorovat



Váhy

- hodnota každého spojení mezi neurony
- založeno na principu synaps v biologických neuronových sítích
- značeno jako ***w***

Biasy

- posunují aktivační funkce neuronů
- umožňují neuronu aktivovat se i při nižších
- lepší přizpůsobení sítě datům (uvidíme v následujících částech prezentace)
- značeny jako ***b***

Mraveniště aneb co všechno musí každý neuron udělat…

- zbytek prezentace se vztahuje pouze na menší regresní modely (jeden vstupní neuron, skryté vrstvy, jeden výstupní) → něco víc by bylo na další (ještě obsáhlejší) prezentaci
- každý neuron v každé vrstvě provádí přibližně toto:
  - ***výstup neuronu = aktivační funkce ((w1 * x1 + w2 * x2 ++wn * xn) + b)***
  - b = bias
  - w = váha spoje
  - x = hodnota, kterou odeslali neurony v předchozí vrstvě
  - výstup neuronu je poté odeslán všem neuronům v další vrstvě (pokus projde aktivační funkcí)

Praktický výpočet

Praktický výpočet

Algoritmy učení

Inicializační algoritmy

Čistě pro zajímavost...

- GPT 2:
  - 48 vrstev
  - 1600 neuronů v každé
  - 1 500 000 000 parametrů
  - trénování na 40 GB textu
  - generování odpovědi na na domácím pc: 5-10 minut
  - trénink cca 40 000 - 100 000 dolarů
- GPT 3:
  - 175 000 000 000 parametrů
- GPT 4:
  - 1,7 bilionu parametrů (cca 1000x více než GPT2)

Akcelerace pomocí gpu

Zdroje

- https://medium.com/@waadlingaadil/learn-to-build-a-neural-network-from-scratch-yes-really-cac4ca457efc
- https://www.desmos.com/calculator
- https://www.sololearn.com/en/learn/courses/le-machine-learning?location=2
- https://networkx.org/documentation/stable/reference/index.html
- https://numpy.org/doc/
- https://scikit-learn.org/stable/
- https://matplotlib.org/stable/index.html
- https://seaborn.pydata.org/
- https://medium.com/@shauryagoel/kaiming-he-initialization-a8d9ed0b5899

Zdroje

- https://en.wikipedia.org/wiki/Multilayer_perceptron
- https://cs.wikipedia.org/wiki/Goniometrick%C3%A1_funkce
- https://www.researchgate.net/publication/220541319_Fully_Complex_Multi-Layer_Perceptron_Network_for_Nonlinear_Signal_Processing
- https://ai.stackexchange.com/questions/24282/what-is-the-justification-for-kaiming-he-initialization
- https://copilot.microsoft.com/chats/ZDZBNPYWVSiy6F8q4t917
- https://chatgpt.com/
- https://stackoverflow.com/questions/13453501/my-neural-net-learns-sin-x-but-not-cos-x
- https://blog.dataiku.com/pre-trained-models-ais-object-oriented-programming
- https://news.ycombinator.com/item?id=19402666
-

Obrázky

- https://www.oreilly.com/library/view/deep-learning-essentials/9781785880360/12eaae27-3de8-4f9d-8bbf-0db18d6c3297.xhtml