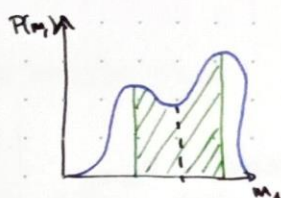


← Bayesovský odhad parametrů

Společně se marginalizované rozdělení hustoty



$$m_1: m_1 + n_{10} = n_{10}$$

toho pro každý parametr; sestaví se korelační grafy (corner plots)



Pravděpodobnost se počítá Monte Carlo metodou, protože normalizace integrálem je příliš složitá. Vygenerují se náhodné sady parametrů, které splňují předpoklady modelu $\pi(\theta)$, vypočítá se očekávaný signál a určí se pravděpodobnost, že měřený signál odpovídá očekávanému. Normalizace se pak určuje z počtu vygenerovaných sad parametrů.

Předpoklady modelu jsou zcela jistě fyzikální omezení (např. hmotnost musí být nezáporná) nebo empirické / observační omezení a rozdělení (např. maximální hmotnost a rozložení ~~hmotnosti černých děr~~ hmotnosti černých děr)

□ Strojové učení

$$f_{\Theta}: X \mapsto Y$$

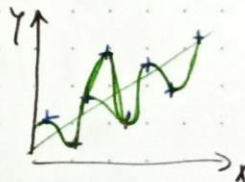
↑
parametry ML modelu

↑
aproximace parametrů signálu (fyz. modelu)

Učení s učitelem: Hledání parametrů Θ , které se kterými model minimalizuje odchylku od pozorování.
 $\forall X: \Theta: \min(Y - f_{\Theta})$

Platí věta o všeobecné aproximovatelnosti:
 Vždy lze najít f_{Θ} takové, které s libovolnou přesností aproximuje data.

sem spadá fitování



Řešíme předvídatelnost vs. přehnat
 ↓
 underfitting vs. overfitting

Hlavní úlohy ve strojovém učení:

- Klasifikace - do které množiny vstupní data spadají (zde: jsou data gravitační vlna nebo ne)
- Regrese - z dat předvidá nějaké pořadové veličiny (inverzní úloha)

Učení bez učitele: nemáme Y , model se pravidla hledá sám
(takto fungují LLM)

PERCEPTRON



Nejjednodušší model, základní stavební kámen neuronových sítí

$$y = w \cdot x + b$$

váha bias

V principu vstup (i výstup) nemusí být číslo, ale i vektor at už prostorový, tak časová řada

Na začátku modelu v náhodných (typicky náhodných) stavech, procesem učení se tato čísla mění

UČÍCÍ SMYČKA

1. Začínáme s trénovacími daty $(x, f(x))$ ← správné hodnoty hledané funkce pro x („labels“)
2. Nastavíme náhodné hodnoty w a b
3. Vypočítáme $y = w \cdot x + b$ (vložíme data do perceptronu)
4. Spočítáme rozdíl $(f(x) - y)$... „loss function“ $\mathcal{L}(f(x) - y)$
... zde mnoho možností: absolutní hodnota, kvadrát, logaritmus, gaussian, ...
5. Zpětná propagace - klíčový krok! - úprava hodnot w a b podle $\mathcal{L}$

cílem učení je postupně (iterativně) snižovat hodnotu $\mathcal{L}$

$$\Theta_{n+1} = \Theta_n - \eta_n \nabla_{\Theta} \mathcal{L}(f(x) - y)$$

„epocha“

gradient $\mathcal{L}$ v jednotlivých váhách a bisech

míra učení - jak velké skoky v parametrech se mají provést
může se postupně snižovat, aby učení nepřestřelalo minimum
(stejně jako Newtonově-Raphsonově metodě hledání kořenu funkce)

Není cílem dosáhnout $\mathcal{L} = 0$, to by byl overfitting, což je nežádoucí (fituje šum). Zvolení hranice konvergence je jednou z hlavních úloh strojového učení; pro klasifikační problémy chceme být výrazně blíže nule než u regresečních problémů

Střední kvadratická odchylka je typickou šablonou pro loss function

$$\mathcal{L} \propto \frac{1}{MN} \sum_i (f(x_i) - y_i)^2$$

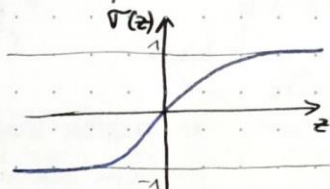
počet trénovacích párů
počet parametrů v Θ

AKTIVAČNÍ FUNKCE

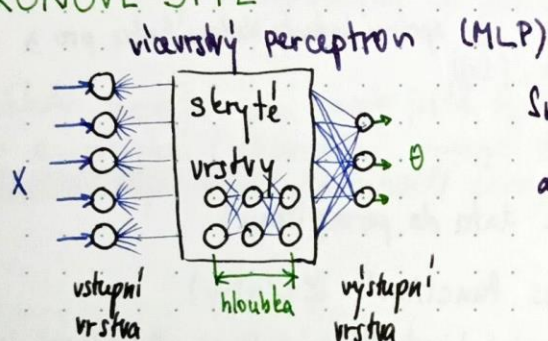
Ukazuje se, že lineární závislost perceptronu nefunguje (nekonverguje, nebo je výsledek špatný; problém s konvergencí obzvláště u větších neuronových sítí)

→ do perceptronu se vkládá nelineární funkce $y = \sigma(w \cdot x + b)$

např. $\sigma(z) = \begin{cases} \tanh(z) \\ \frac{1}{1+e^{-z}} \\ \begin{cases} z > 0 \\ 0, & z \leq 0 \end{cases} \end{cases}$



NEURONOVÉ SÍTĚ



Snažíme se skrýté vrstvy navrhnout tak, aby skrýté vrstvy byly co nejmenší (šetří čas a energii), ale zároveň dávají co nejlepší výsledky

MLP je základní architektura neuronových sítí, je téměř univerzální.

Pro specializované úlohy (např. zpracování obrazu) se osvědčily jiné architektury.

- Multi-Layer Perceptron
- Convolutional Neural Network
- Recurrent Neural Network
- Attention - Based Neural Network Transformer $\Rightarrow$ LLM

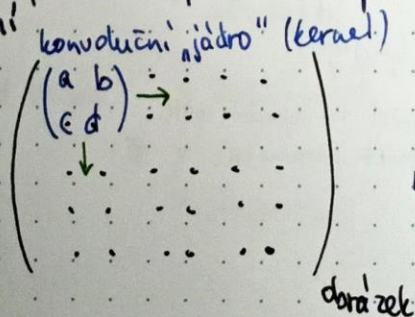
KONVOLUČNÍ NEURONOVÉ SÍTĚ

Vhodná pro statická 2D data (obrázky)

Zachováva symetrie, prostorové frekvence, ...

Obrázek je tenzor x, y , barva, rozdělíme na tři matice podle barvy (RGB)

Máme menší matici, kterou posouváme po každé z obrazových matic a počítáme maticové násobení



parametry konvolučního jádra se trénují
velikost kernelu ovlivňuje, jaké prostorové frekvence dokáže kernel identifikovat

REKURENTNÍ NEURONOVÉ SÍTĚ

Vhodné pro časové řady; spolu se vstupními daty přijímají i výstup předchozí operace

POZORNOSTNÍ TRANSFORMUJÍCÍ SÍTĚ

Aktivní oblast zájmu pro GW, zvládají velmi dlouhé časové řady

Výchozí článek: Attention is all you need (2017)

SOUČASNÉ METODY PRO DETEKCI GRAV. VLN

AMORTIZOVANÁ INFERENCE

Ze signálu potřebujeme odvodit parametry fyz. modelu (složitá inverzní úloha, založená na Bayesovské statistice)

Knihovna (sítí) DINGO

[Potřebujeme rychle detekce grav. vln a získání parametrů, abychom mohli na zdroj namířit teleskopy (multi-messenger astronomy)]

Pipeline:

1. Rychle odhadnout parametry (polohu) pro daný typ signálu
↑ řádově v sekundách až minutách
2. Později bayesovsky přenést parametry
↑ minuty až hodiny

I v druhém kroku potřebujeme zajistit dostatečnou rychlost, protože detekce bude (s novými interferometry) bude čím dál tím více

INTERFERENCEČNÍ MODEL

Výpočet přímé úlohy (vlny z parametrů) je také nákladný výpočet, pomocí strojového učení se interpolují řešení v parametrickém prostoru z přesně napočítaných řešení z několika malých sad parametrů