

Machine learning v Pythonu



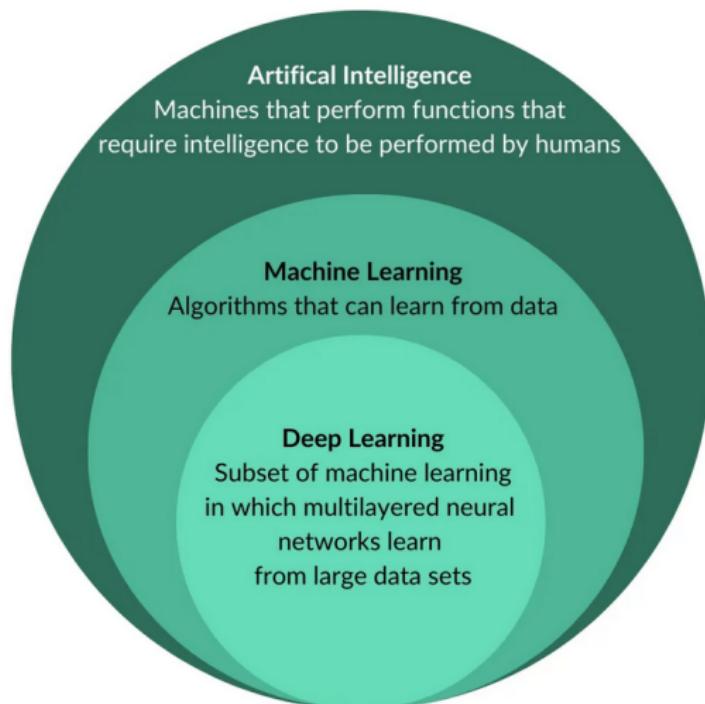
Midjourney v 5.2 - „machine learning in python“

Motivace – co byste si měli odnést?

- Znalost některých základních pojmu a souvislostí.
- Matematický základ, na kterém to všechno stojí.
 - Opravdu hodně věcí je funkce.
- Základní intuici jak to funguje a proč.
- Představu k čemu to je v praxi.
- Tipy jak můžete ML dělat i vy.
- **Inspiraci a dobrý pocit :)**

- Úvod do teorie
 - ML, neuronová síť, funkce, derivace, optimalizace
- Příklad 1
 - Regrese v Pytorchi
- ML projekty v 3Dimu
 - Život ML projektu, konkrétní příklady
- Příklad 2
 - Multifeatured data v Pytorchi

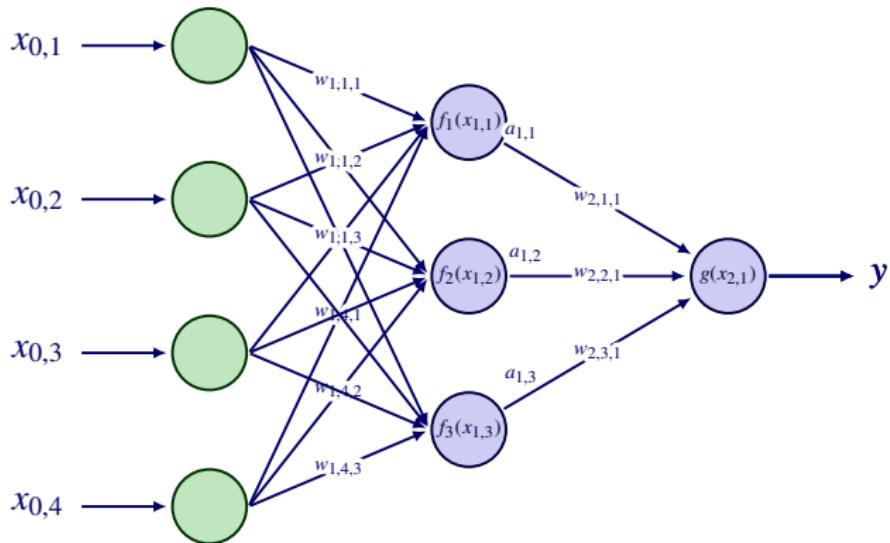
Umělá inteligence vs machine learning



<https://himatk.substack.com/p/ai-product-management-machine-learning-concepts>

Neuronová síť

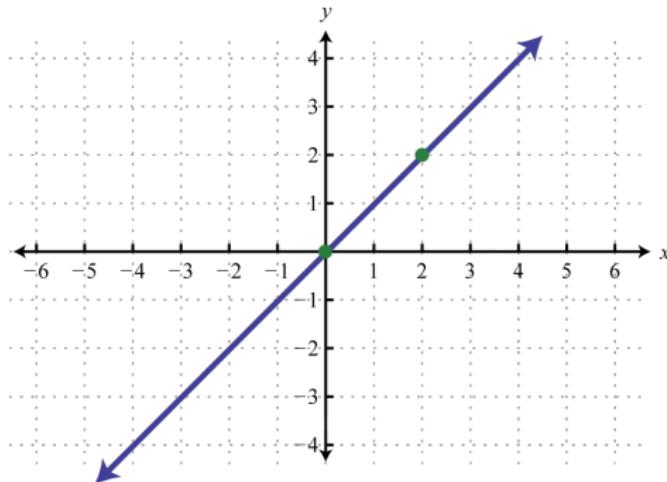
- Co je $f_j(x_{i,j})$?
- Co je tohleto?



Funkce – opakování? :)

$$f(x) = x$$

x	$f(x)$
0	0
2	2



https://saylordotorg.github.io/text_intermediate-algebra/s05-04-graphing-the-basic-functions.html

• Intuice

- Jaké známe funkce ze „skutečného“ světa?
- Co funkce dělá?

Funkce – Co na to AI?

Vymenuj 5 příkladů reálných funkcí z běžného života teenagerů



1. Počet lajků na Instagramu v závislosti na počtu příspěvků: Tato funkce může popisovat, jak se zvyšuje počet lajků s rostoucím počtem příspěvků na sociálních sítích.
2. Spotřeba mobilních dat v závislosti na čase stráveném na internetu: Tato funkce by mohla ukázat, kolik dat spotřebuje teenager při surfování po internetu za určitý čas.
3. Únava během dne v závislosti na hodinách spánku: Tato funkce může vyjádřit, jak se únava teenagera mění s množstvím spánku, který si dopřeje.
4. Množství ušetřených peněz v závislosti na tom, kolik nekoupí věcí: Tento vztah může popisovat, kolik peněz teenager ušetří, když se rozhodne nekoupit věci, které si původně chtěl pořídit.
5. Šance na úspěšné složení zkoušky v závislosti na čase stráveném učením: Tato funkce může ilustrovat, jak se zvyšuje pravděpodobnost úspěchu u zkoušky s rostoucím množstvím času věnovaného studiu.

Funkce – matematictěji

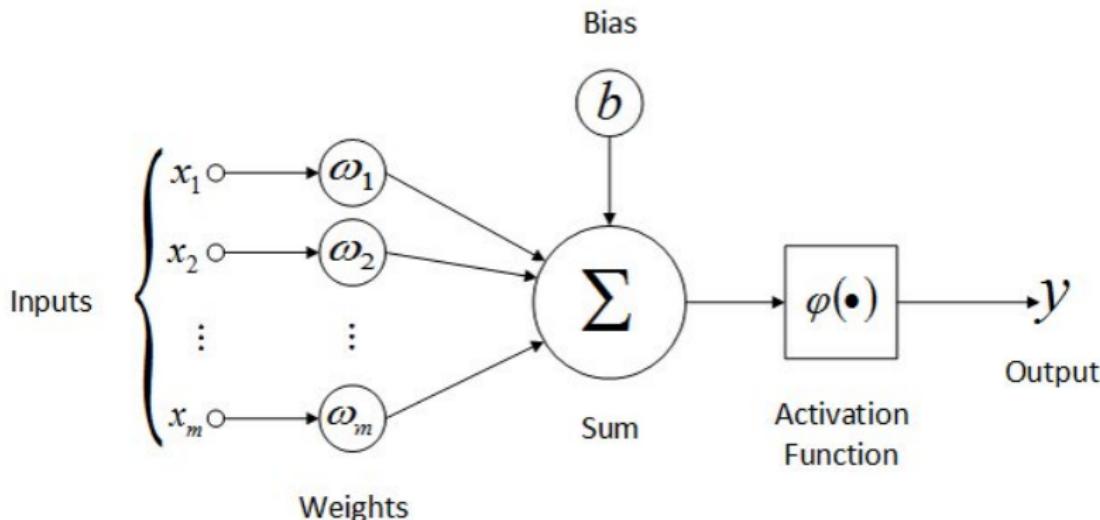
- Proměnná (např. $x_{i,j}$, $w_{i,j,k}$, $y\dots$).
- Vstup(y), výstup, jméno funkce.
 - Co je co?
 - $y = g(x_{i,j})$
 - $secti(a, b) = a + b$
- Definice funkce (tabulka, rovnice, graf, slovní definice, kód v programovacím jazyku...).
 - Musí být jednoznačná (jeden vstup = jeden výstup!)
- Některé užitečné funkce mají svůj speciální zápis:
 - $\sum_{j=1}^M w_j = w_1 + w_2 + \dots + w_M$
 - $suma(w_1, w_2, \dots, w_M) = w_1 + w_2 + \dots + w_M$

Funkce – co s nimi můžeme dělat?

- Můžeme je skládat: $g(x_1, x_2) = h(f(x_1, x_2))$.
 - *pričti jedna*(x) = $x + 1$
 - *vynásob**pričti jedna*(x_1, x_2) = *pričti jedna*(*vynásob*(x_1, x_2))
 - *vynásob**pričti jedna*(x_1, x_2) = $x_1 \cdot x_2 + 1$
- Můžeme je derivovat.
 - (Ale jen pokud splňují nějaké podmínky...)
 - **Derivace udává směr funkce v bodě.**
 - Vysvětlíme, ukážeme...

Umělý neuron

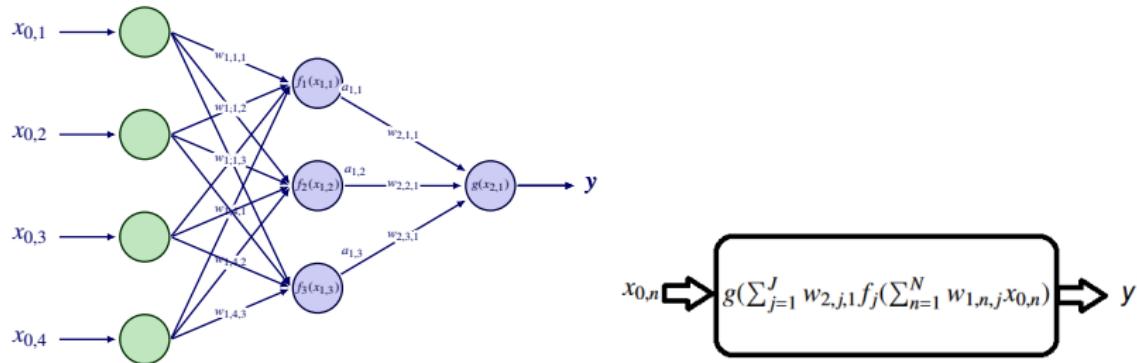
- Umělý neuron lze vyjádřit funkcí
 - Můžeme s ním dělat to, co s funkcemi.



<https://www.researchgate.net/publication/320270458>

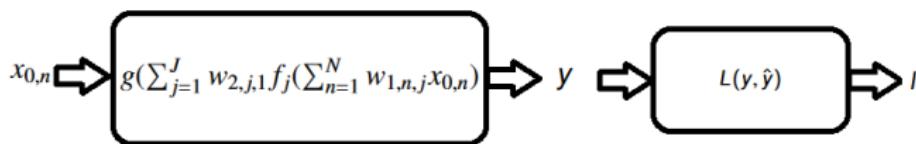
Neuronová síť

- **Neuronová síť lze vyjádřit funkcí!**
- i je počítadlo vrstev, j, k jsou počítadla položek ve vrstvách, n je počítadlo vstupů.
- $y = g(x_{2,1})$
- Dosadíme $x_{2,1} = \sum_{j=1}^J w_{2,j,1} a_{1,j}$, dostaneme $y = g(\sum_{j=1}^J w_{2,j,1} a_{1,j})$
- Dosadíme $a_{1,j} = f_j(x_{1,j})$, dostaneme $y = g(\sum_{j=1}^J w_{2,j,1} f_j(x_{1,j}))$
- Dosadíme $x_{1,j} = \sum_{n=1}^N w_{1,n,j} x_{0,n}$, dostaneme $y = g(\sum_{j=1}^J w_{2,j,1} f_j(\sum_{n=1}^N w_{1,n,j} x_{0,n}))$



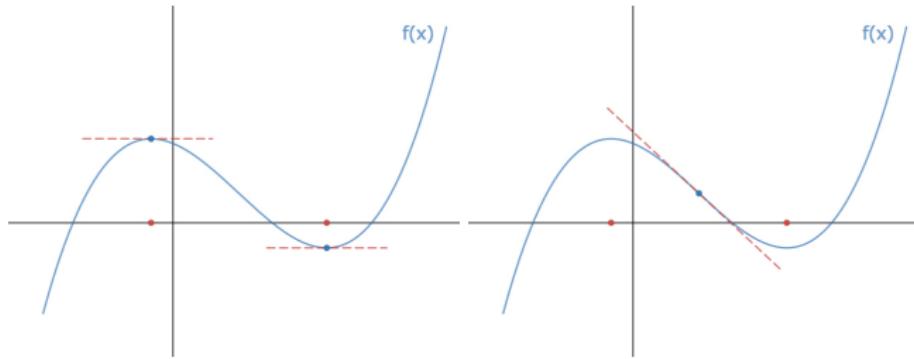
Jak se to učí - chybová funkce

- Jak se učíme my?
 - Pokus – omyl – oprava – opakovat.
- Jak se učí neuronovky?
 - Pokus – potřebujeme data.
 - Vstupy – množina $X = \{x\}$
 - Správné výstupy („labely“) – množina $\hat{Y} = \{\hat{y}\}$
 - Omyl – musíme spočítat chybu.
 - Chyba – l
 - Chybová funkce – $l = L(y, \hat{y})$
 - $loss(vystup_site, spravny_vystup)$
 - Oprava – upravíme váhy w_{ij} , aby to příště vyšlo lépe :)
 - Na to musíme umět derivovat :(



Derivace – intuitivně

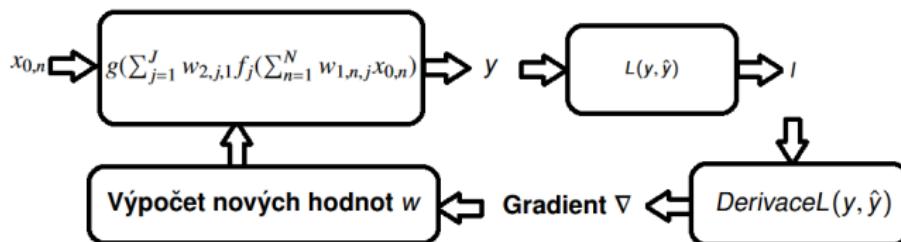
- **Derivace** udává směr funkce v bodě.
- **Gradient** ∇ je „sada“ všech derivací pro všechny naše váhy a udává tedy směr největšího růstu funkce.
 - Příklad:
 - Loss funkce „zabaluje“ neuronovku s dvěma váhami w_1 w_2 .
 - Výstup sítě y vyšel 10, očekávaný výstup \hat{y} má být 15.
 - Gradient může vypadat takto: $\nabla L(10, 15) = (2, 3)$.
 - Derivace podle váhy w_1 je 2, derivace podle váhy w_2 je 3.
- Neuronovky musí být poskládané z funkcí, pro které lze spočítat derivaci.



<https://www.expii.com/t/sketching-the-derivative-of-a-function-141>

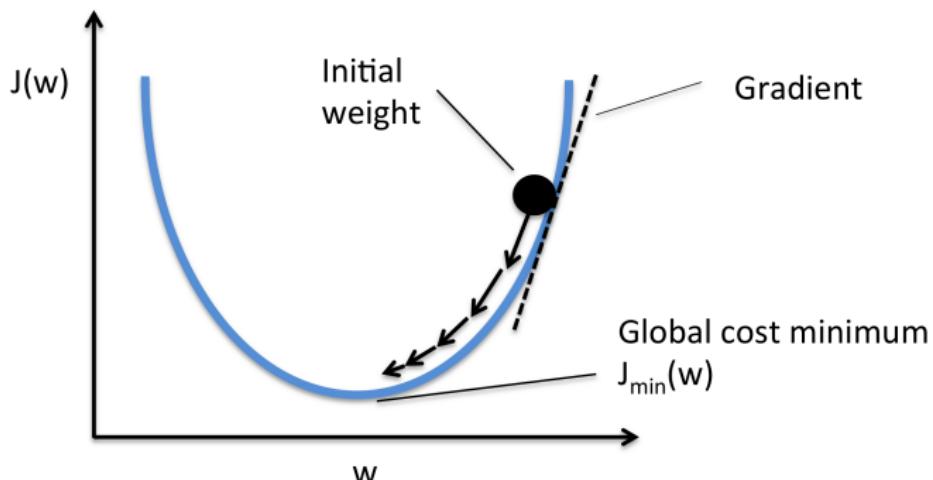
Jak se to učí – gradientní sestup

- Natrénovat model znamená **zmenšit chybu co nejvíce**.
 - Naštěstí jsme si chybu vyjádřili funkcí.
- Když chceme zmenšit chybu, musíme jít po chybové funkci směrem dolů :)
 - Naštěstí dokážeme zjistit, jaký směr má chybová funkce pro náš momentální vstup – **zderivujeme chybovou funkci**.
- O kousek změníme váhy modelu, aby byl výstup příště lepší.
 - Změnu uděláme **proti směru gradientu**.



Jak se to učí – gradientní sestup

- To musíme udělat pro všechny váhy a hodně mockrát dokola.
 - Tento algoritmus se jmenuje **gradientní sestup**.



<https://sebastianraschka.com/faq/docs/gradient-optimization.html>

Rekapitulace teoretické části a ukázka

- Co si mám pamatovat:

- Neuronová síť je funkce.
- Chyba neuronové sítě je dána **chybovou funkcí**.
- **Derivace** udává **směr funkce** v bodě x.
- **Gradient** chybové funkce je **vektor derivací podle všech vah** a udává **směr chybové funkce**.
- Pro natrénování neuronové sítě musíme jít po chybové funkci směrem dolů.

- Ukázka

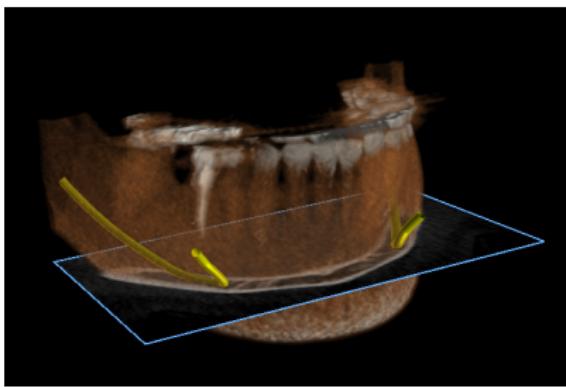
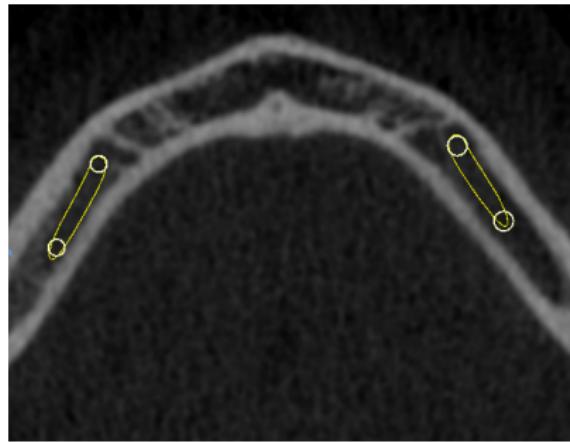
- Zobrazení funkce a její derivace.
- Příklad použití chybové funkce pro lineární regresi.
- Příklad lineární a nelineární regrese s knihovnou Pytorch.
- Otevřete si <https://github.com/Travnickt3d/FITLS/tree/main/2024>

Praxe – ML tým v TESCAN 3Dim



Začátek ML projektu

- Zákazník vzneš požadavek na automatickou funkcionalitu v aplikaci
 - „Chci aby uživatel nemusel naklikávat nerv ručně“.
 - „Chci aby šlo vysegmentovat všechny zuby jedním tlačítkem“.
 - ...



Analýza proveditelnosti

- Někdo z ML týmu dostane za úkol najít vhodné řešení úlohy a musí:
 - Zjistit existující řešení.
 - Přečíst odborné články.
 - Vyzkoušet veřejně dostupný kód nebo napsat vlastní a udělat pár experimentů.
- Výstupem je **rozhodnutí, jestli úlohu dokážeme řešit a jak složité to bude.**

The screenshot shows a search result for the paper "U-Net: Convolutional Networks for Biomedical Image Segmentation" by Olaf Ronneberger, Philipp Fischer, and Thomas Brox. The page includes the arXiv logo, a search bar, and a detailed description of the paper's content, download options, and citation information.

Search results for "U-Net: Convolutional Networks for Biomedical Image Segmentation":

Download:

- PDF
- Other formats

Current browse context: cs.CV
< prev | next >
new | recent | 1505
Change to browse by: cs

References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar

31 blog links (view in sidebar)

DBLP - CS Bibliography

listing | bibtex
Olaf Ronneberger
Philipp Fischer
Thomas Brox

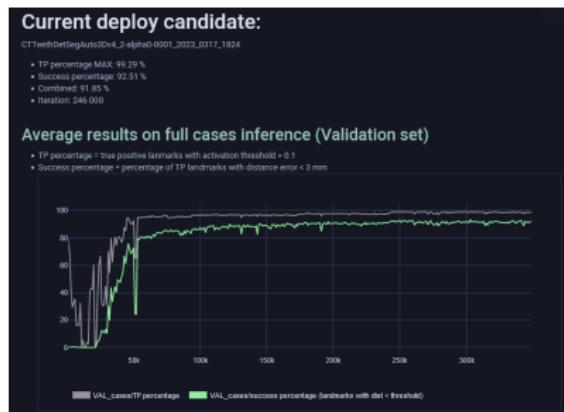
Export BibTeX Citation

Abstract: There is large concern that successful training of deep networks requires many thousand annotated training samples. In this paper, we present a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more effectively. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise prediction. We show that such a network can be trained end-to-end from very few images and outperforms the prior best method (a sliding window convolutional network) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. Using the same network trained on transmitted light microscopy images (phase contrast and DIC) we can also segment the same challenging structures in electron micrographs by a large margin. Moreover, the network is fast. Segmentation of a 512x512 image takes less than a second on a recent GPU. The full segmentation (based on Caffe) and the trained networks are available at <http://tanc.informatik.uni-freiburg.de/>.

Šikovná stránka pro čerpání odborných znalostí: <https://arxiv.org/>

Prototypování

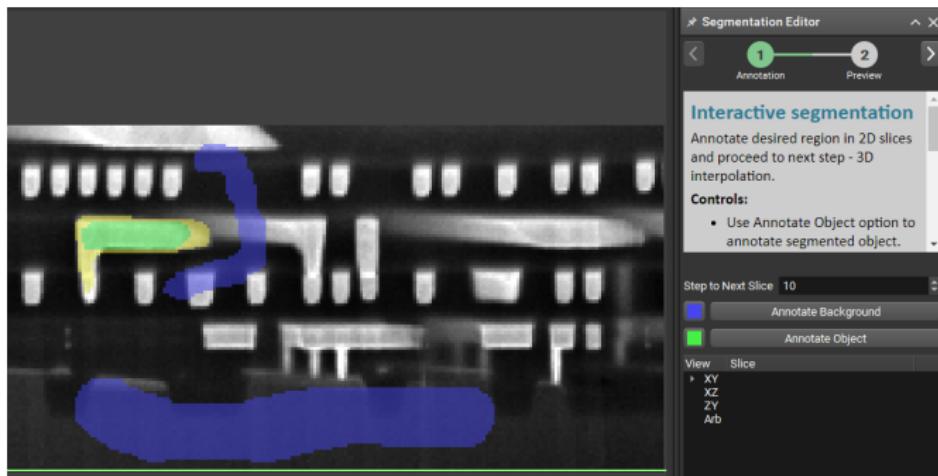
- Pouštíme trénování s různým nastavením a sledujeme, jak dobře vypadají výstupy modelu.
 - K trénování používáme ML frameworky:
 - U nás PyTorch, další jsou třeba Tensorflow, Caffe, Scikit, Keras ...
 - K monitorování tréninku také existují nástroje:
 - wandb, clearml, Tensorboard
- Výstupem je **nejlepší natrénovaný model.**



Ukázka reportu pro projekt detekce zubů.

Nasazení

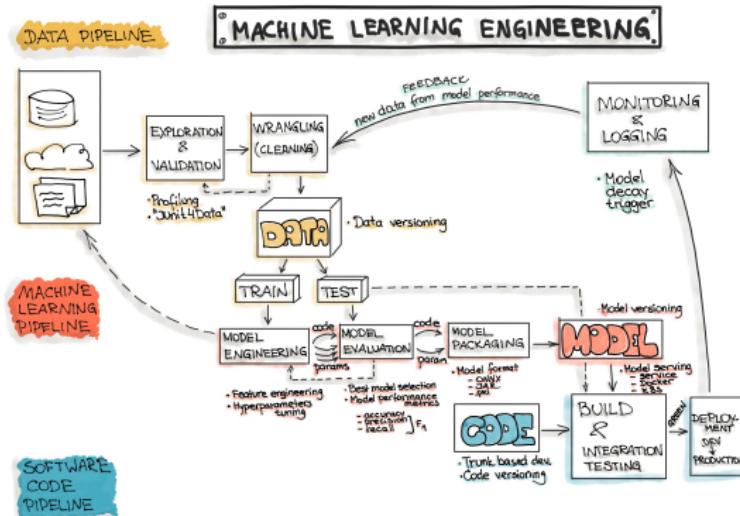
- Příprava modelu pro reálné použití.
 - Integrace do aplikace, serverové řešení...
- Výstupem je ML funkcionality, kterou může používat koncový uživatel.



Grafické rozhraní pro interaktivní segmentaci mikroskopových dat.

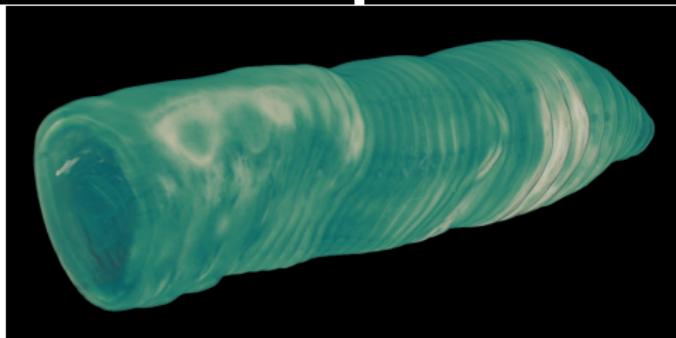
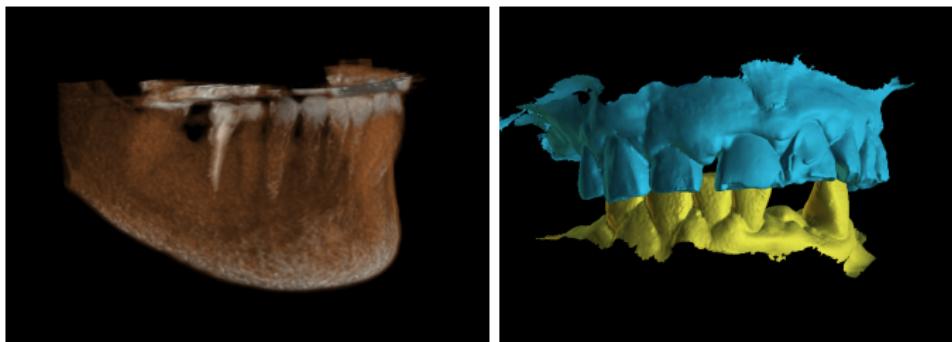
V ideálním případě...

- Sběr statistiky o úspěšnosti modelu v praxi.
- Automatizace trénování a nasazování nových verzí modelu.
- Průběžné vylepšování metody s použitím technologických novinek.
- Více o **MLOps** <https://ml-ops.org/>.



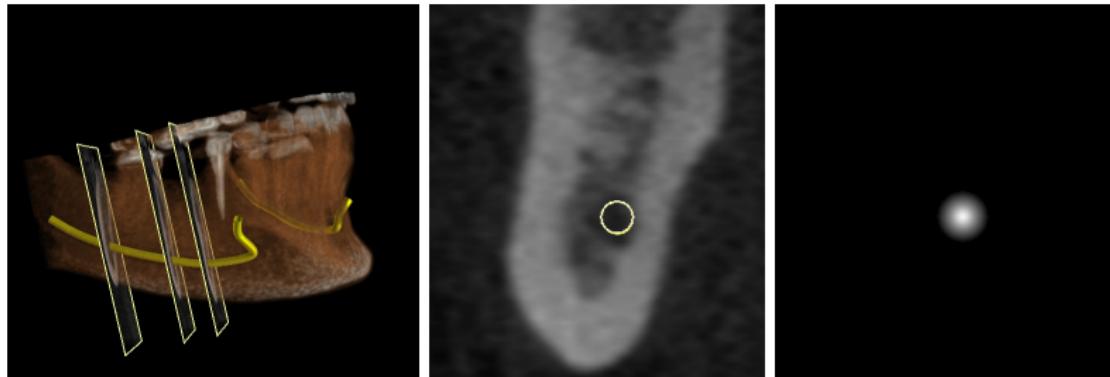
Naše data

- Zubařské CT skeny čelistí pacienta (objemová data).
- 3D skeny zubů pacienta (povrchová data).
- Data z elektronových mikroskopů (objemová data).



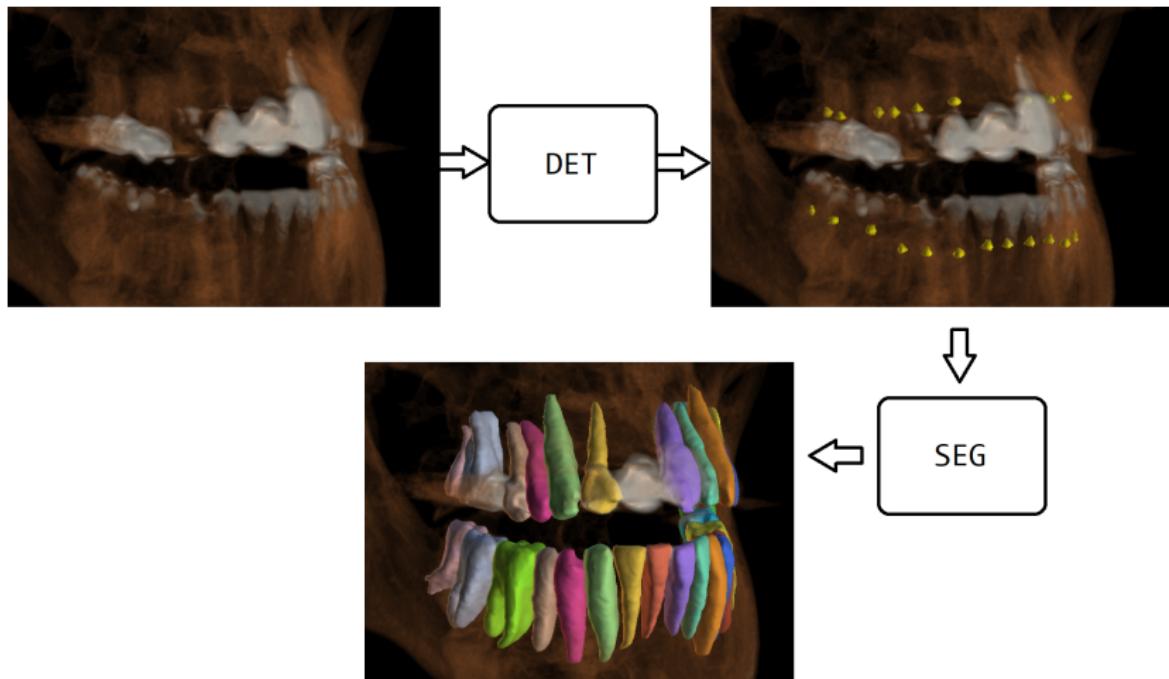
Naše projekty – trasování nervu

- Vstupem neuronovky je řez kolmý na nerv.
- Výstupem je predikce pozice nervu.



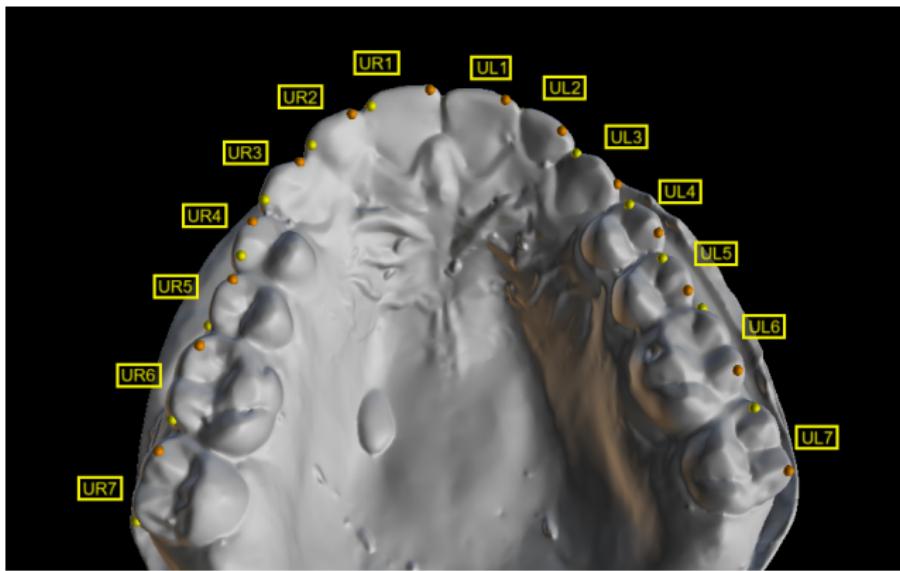
Naše projekty – segmentace zubů v CT

- V první fázi interaktivně - uživatel musí kliknout na zub.
- V druhé fázi „segmentační pipeline“
 - Detekční model detekuje středy zubů.
 - Segmentační model použije pozice středů místo kliknutí uživatelem.



Naše projekty – anotace zubů na povrchovém modelu

- Vstupem do neuronové sítě je zachycený 2D obraz s informací o hloubce.
- Výstupem je predikce bodů v 2D obrazu.
- Několik výstupů z různých pohledů nám umožní spočítat predikovanou pozici ve 3D.



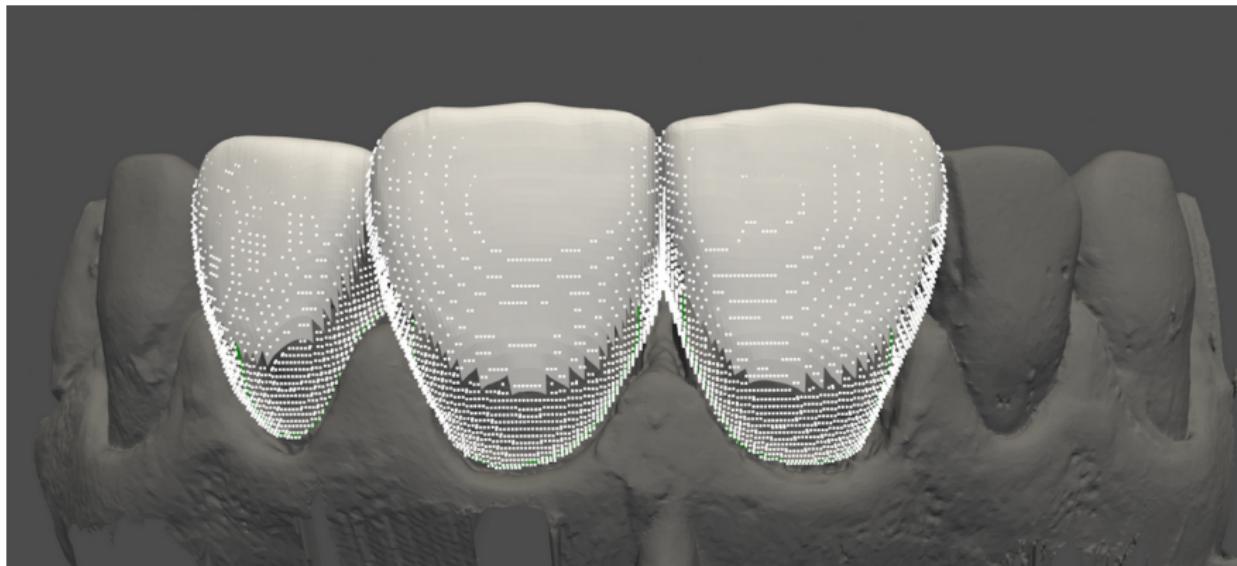
Naše projekty – „perfect smile“

- Automatické zarovnání křivých zubů (pro výrobu rovnátek).
- Vstupem je povrchový model křivých zubů
- Výstupem jsou nové pozice zubů
- Zatím ve fázi vývoje, řešení problematických výstupů...



Naše projekty – Generování korunek

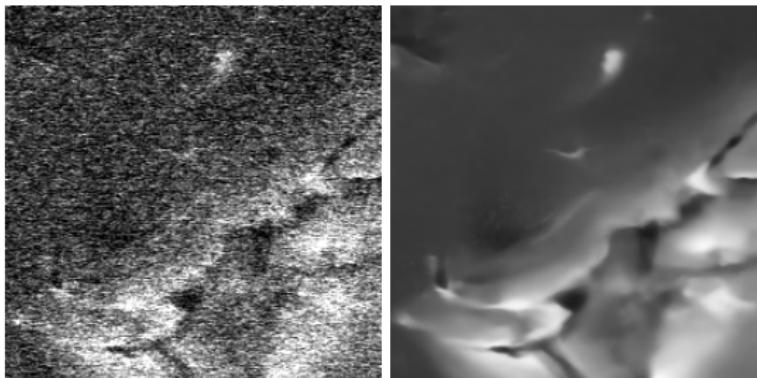
- Složitější architektura tvořená několika moduly.
- Vstupem je snímek čelisti bez zubů.
- Konečným výsledkem je 3D model korunky, který se vyrobí z mračna bodů navržených neuronovkou.



Ukázka vygenerovaných korunek pro tři chybějící zuby.

Naše projekty – Zlepšení kvality obrazu

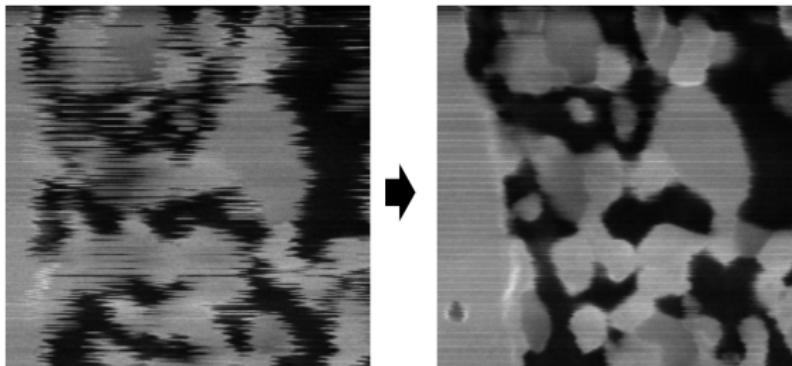
- Elektronový mikroskop snímá „pěkně“ a pomalu nebo „škaredě“ a rychle.
- Sanžíme se neuronovkou uměle zlepšit kvalitu rychle snímaného obrazu.
 - Ideálně v reálném čase...
- Vstupem je „škaredý“ snímek .
- Výstupem je „pěkný“ snímek.



Ukázka „škaredého“ a „vypěkněného“ snímku z elektronového mikroskopu.

Naše projekty – Zarovnání řezů

- Vzorek v mikroskopu se při snímání řezů může posouvat.
- Je potřeba jednotlivé 2D řezy správně zarovnat, aby 3D objem nebyl „rozházený“.
- Podobné úlohy se řeší detekcí stejných bodů v sousedních snímcích, podle kterých se spočítá zarovnání.
- Ukázka ZDE.



Ukázka rozházených a zarovnaných řezů v mikroskopových datech.

Co můžete dělat vy

- Internety jsou plné tutoriálů a jiných vyučových materiálu.
 - Příklady v Kerasu, Úvod do PyTorch, Python...
 - Youtube channel by Alfredo Canziani
- Podívejte se, co nového lidi vymýšlejí.
 - Two Minute Papers, Odborné články na arXiv, Přehled cool produktů a nástrojů postavených na ML
- Zapojte se do soutěže **Středoškolská odborná činnost** v oboru informatika.
 - Předchozí téma byla např.: Rozpoznávání řeči pro mobilní appku, Knihovna pro neuronové sítě v Kotlinu, Zpravodajský systém s využitím umělé inteligence...
- Přihlašte se na obecný bakalářský program FIT VUT a pokračujte navazující magisterskou specializací Strojové učení :)