UNIVERZA V LJUBLJANI
FMF
Fakulteta za matematiko in fiziko

Uporaba Difuzijskih Modelov za Simulacije v Fiziki Visokih Energij

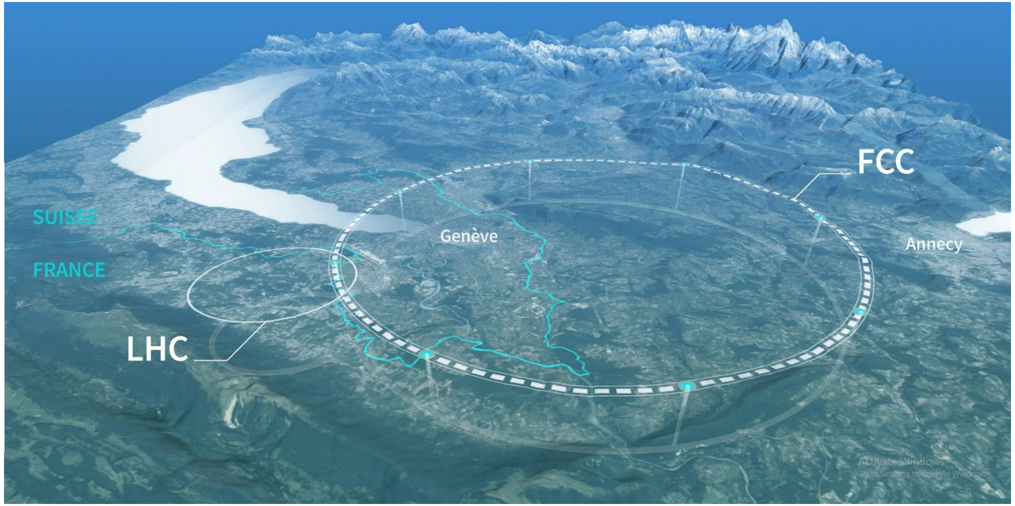
AVTOR:
Dimitrije Pešić

MENTOR:
prof. dr. Borut Paul Kerševan

Vsebina

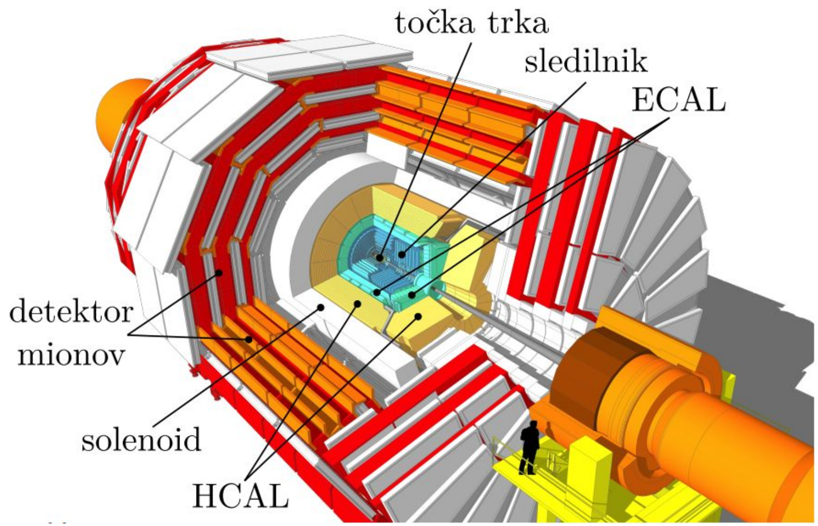
Fizika visokih energij na LHC

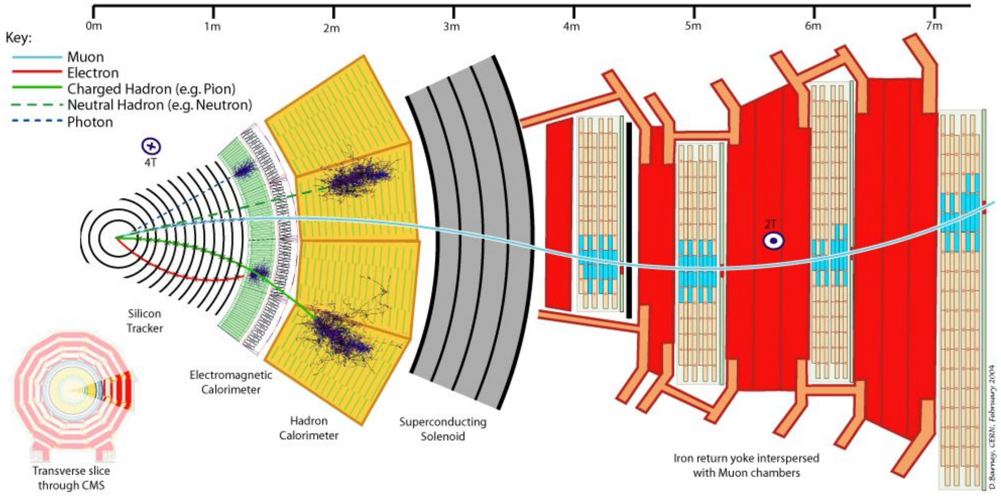
- načrtovanje detektorjev, napovedovanje vedenja, treniranje klasifikacijskih metod
- stohastična in ne-deterministična narava + kompleksne in zapletene interakcije
- Monte Carlo simulacije 50% WLCG
- posodobitev visoke svetilnosti 2025 - HL-LHC
- Future Circular Collider - FCC
- zakaj difuzijski modeli?



Elektromagnetni kalorimeter (ECAL)

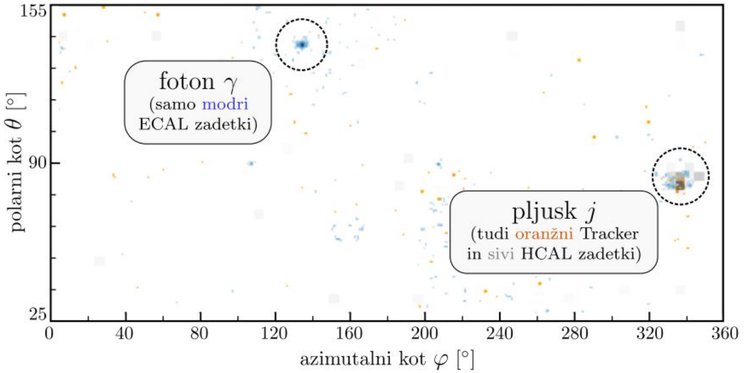
- meri deponirano energijo delcev
- vpadni delec povzroči *EM* pljusk
- fotoni in elektroni
- scintilatorji iz $PbWO_4$ (*svinčev volframat*)
- ~75 000 scintilacijskih kristalov

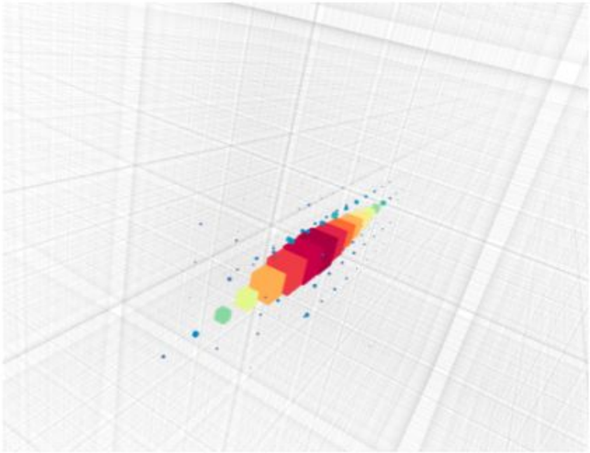


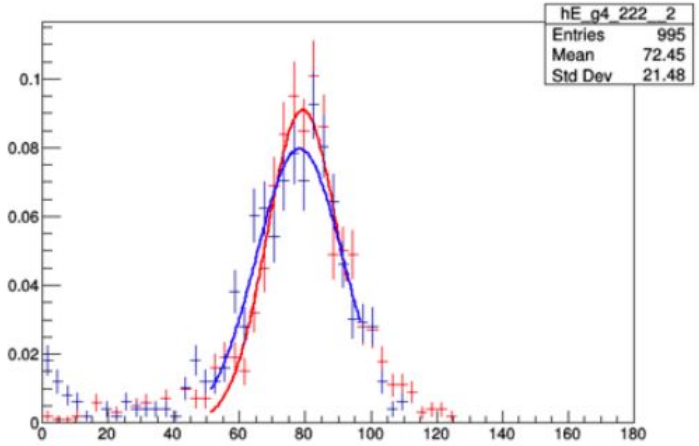


Simulacije z Geant4

- programski paket razvil CERN
- odziv kalorimetra na nivoju celic
- Monte-Carlo metoda
- definiramo geometrijo detektorja
- določimo delce (energija in začetna smer)
- tipičen pljusek 100 GeV





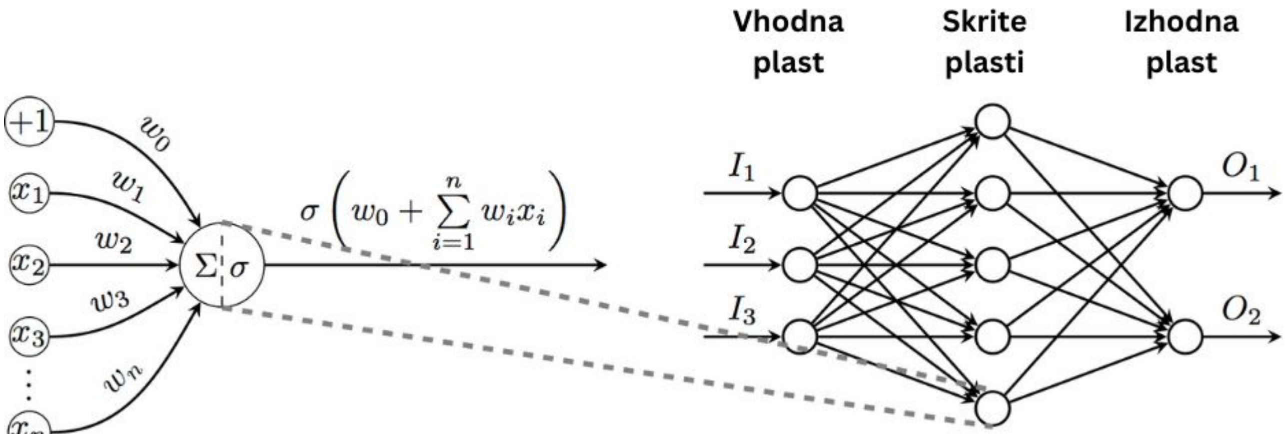


Nevronske mreže (NN)

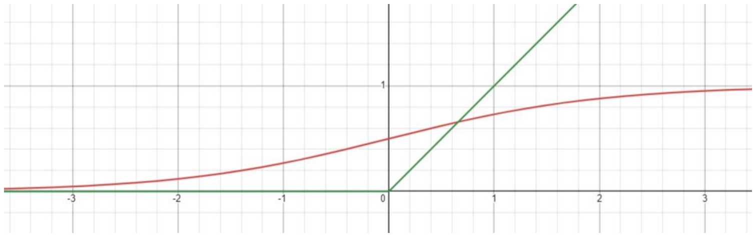
- nadzorovano učenje
- večplastni perceptron
- afine transformacije
- aktivacijske funkcije

$$a_i = \theta\left(\sum_n x_n w_{i,n} + b_i\right)$$

$$\mathbf{a} = \theta\left(\mathbf{W}\mathbf{x} + \mathbf{b}\right)$$



$$\mathrm{ReLU}(x) = \max(x, 0)$$

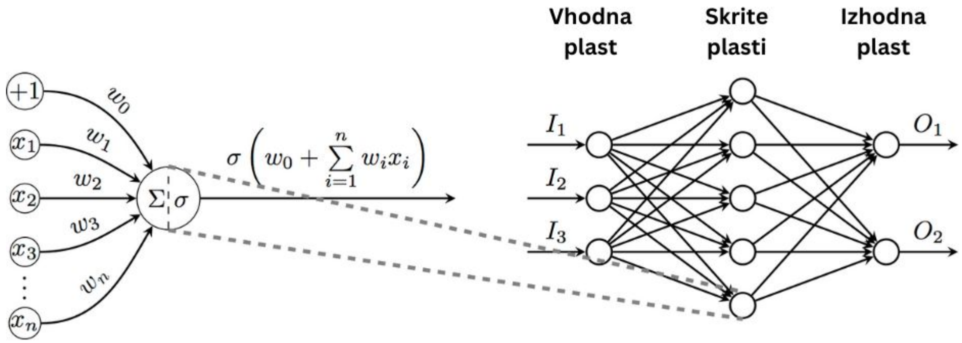


$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Učenje mreže

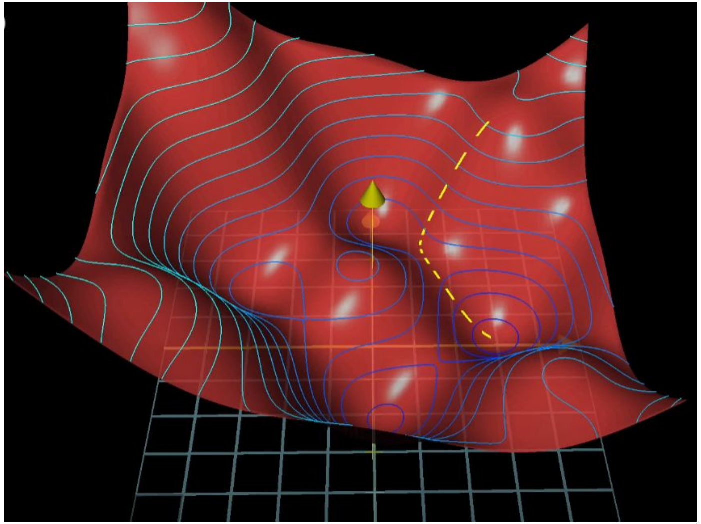
- iterativno prilagajanje parametrov $\boldsymbol{\theta}_t = (\mathbf{W}, \mathbf{b})$
- minimiziranje funkcije izgube (*loss function*)
- test χ^2

- ***backpropagation***
- stohastični gradientni spust



$$L_{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

$$\theta_{t+1} = \theta_t - \alpha \nabla L(\theta_t)$$

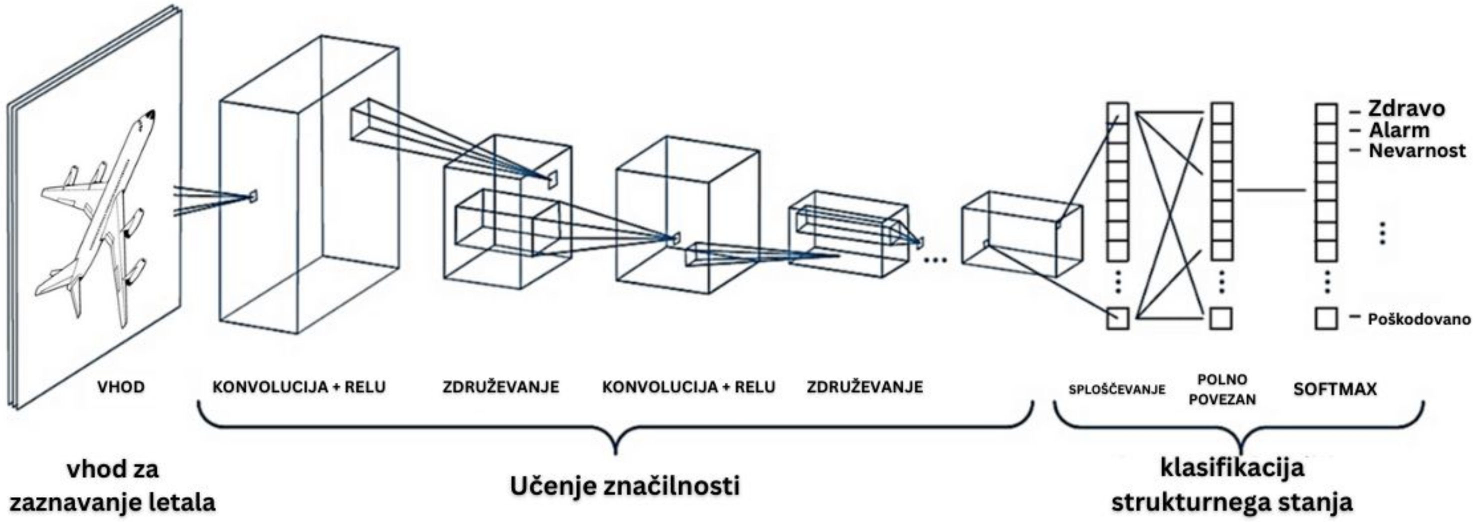


https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi

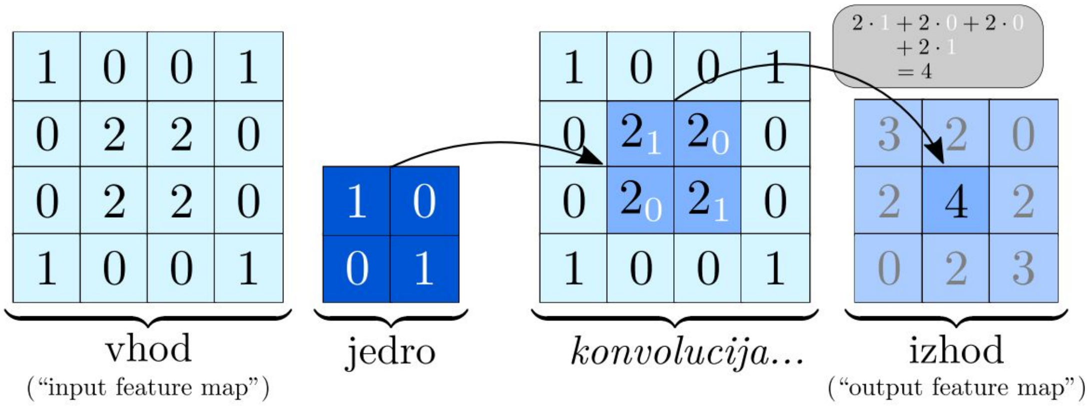
3Blue1Brown

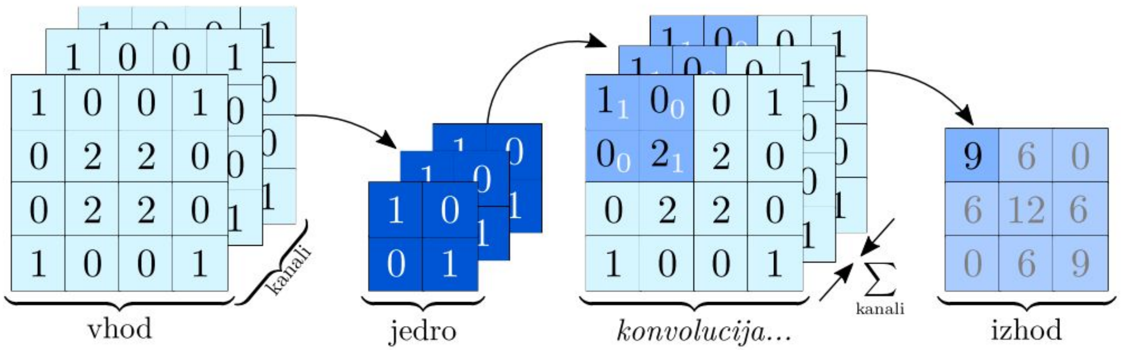
Konvolucijske mreže (CNN)

- AlexNet, Alex Krizhevsky, Ilya Sutskever, 2014
- zajemanje prostorskih odvisnosti značilk
- klasifikacija slik, zaznavanje objektov in segmentacija
- lokalna konvolucija



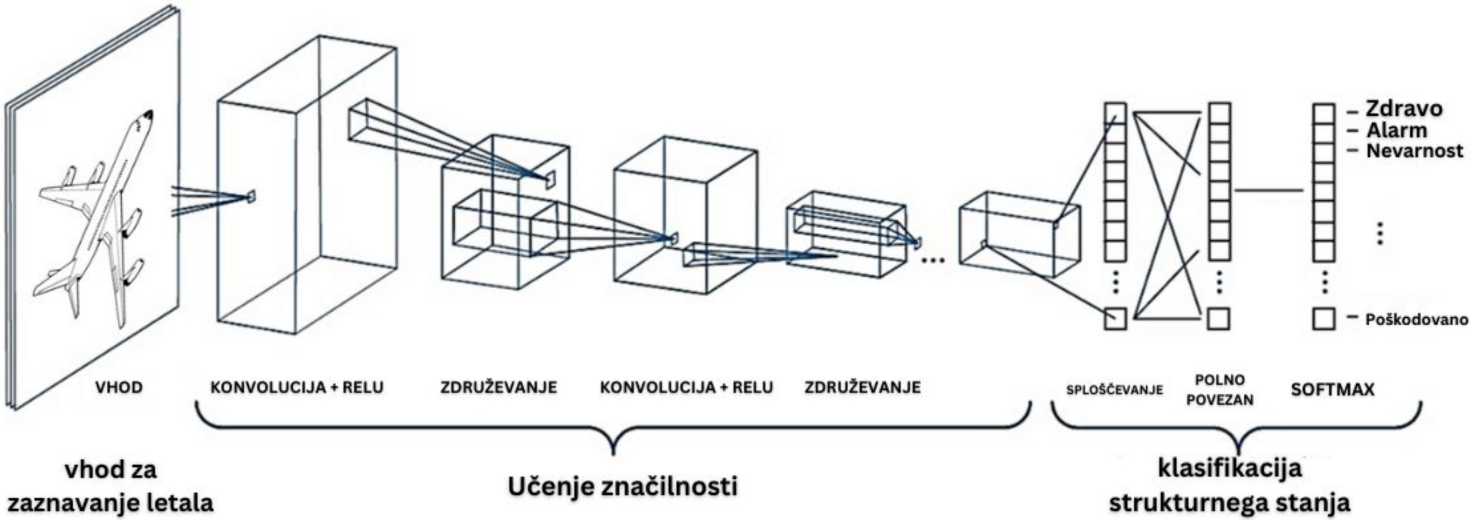
2D diskretna konvolucija





Konvolucijske mreže (CNN)

- treniranje je spreminjanje jeder
- združevalni nivoji



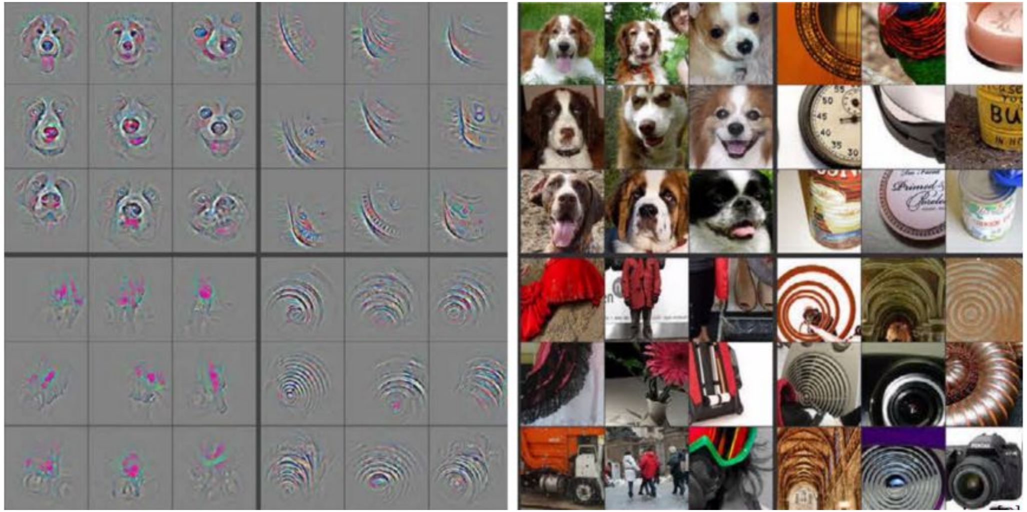
Predstava konvolucije

Jedra (kernels):

- Zaznajo robove,
- kontrastne barve,
- druge vzorce

zgledi predelanih slik:

latentni prostor



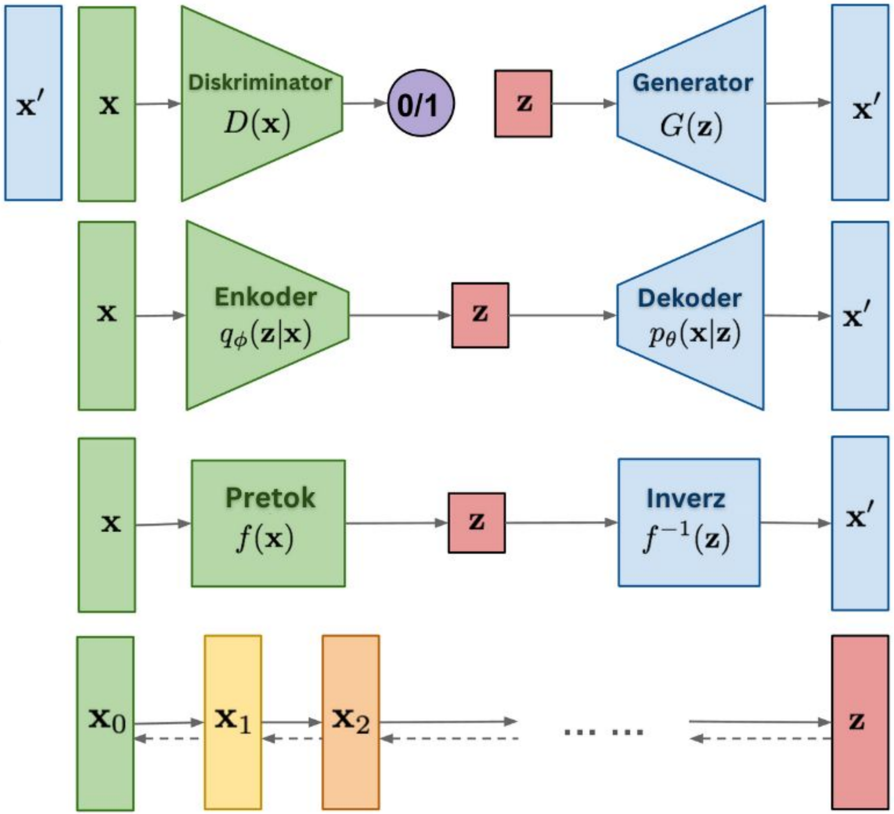
Generativni modeli

GAN: antagonistično treniranje

VAE: maksimiranje variacijske spodnje meje

Pretokovni modeli: obrnljiva transformacija porazdelitev

Difuzijski modeli: Postopno dodajanje Gaussovega šuma in nato vzvratno



Generativni modeli

GAN:

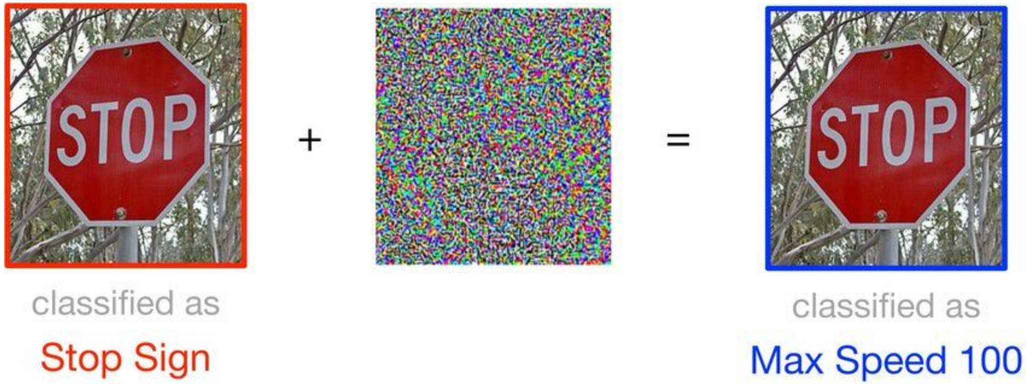
- izjemna kvaliteta
- težko trenirati
- kontradiktorni primeri
- ‘mode collapse’

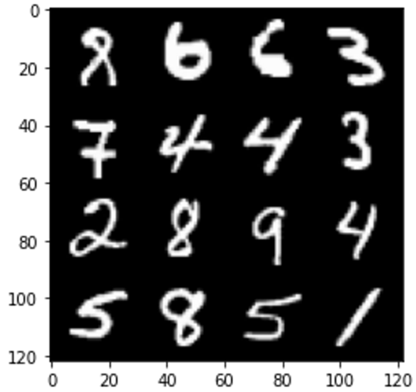
VAE:

- preprosto trenirati
- nizke kvalitete (zamegljene)
- ‘mode collapse’

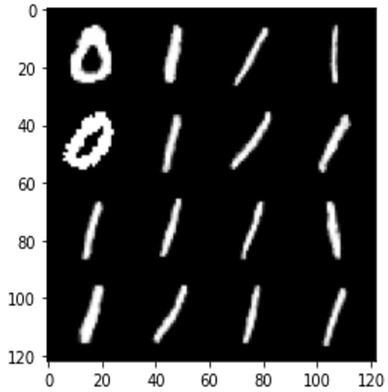
FLOW:

- dobra kvaliteta
- hitri pri vzorčenju
- slabo modelirajo multimodalne distribucije





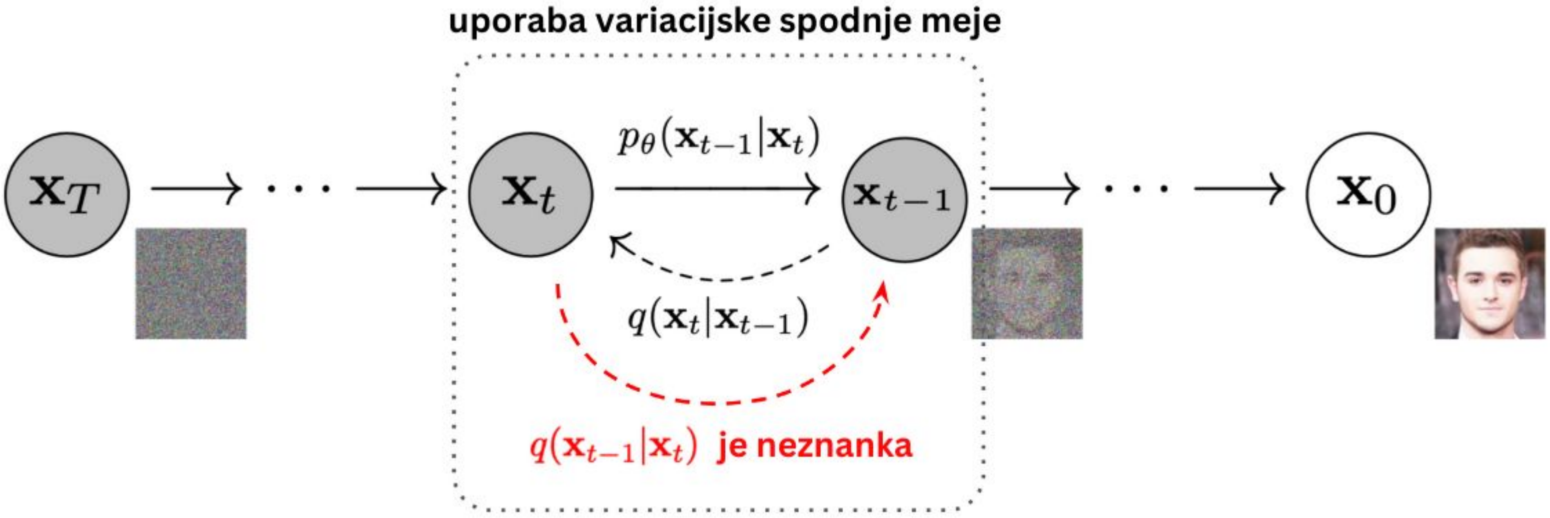
Train Data Point



GAN Generated Data Point

Difuzijski modeli

- postopno dodajanje in odstranjevanje šuma
- veriga Markova
- notacija



Difuzijski proces naprej

- iterativno apliciranje šuma
- razporejevalnik variance
- reparametrizacija

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$$

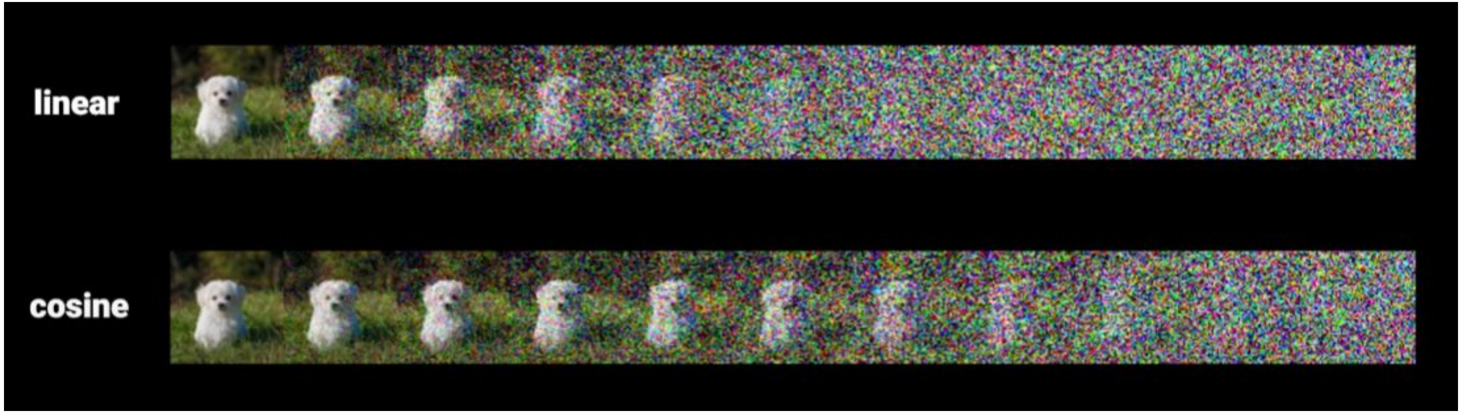
$$\alpha_t := 1 - \beta_t \qquad \bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$$

Reparameterization Trick:

$$\mathcal{N}(\mu, \sigma^2) = \mu + \sigma \cdot \epsilon$$

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$$

$$x_t(x_0, \epsilon) = x_t = \sqrt{\bar{\alpha}_t}x_0 + \epsilon\sqrt{1-\bar{\alpha}_t}, \qquad \epsilon \sim \mathcal{N}(0, \mathbf{I})$$



Vzvraten proces

- v limiti tudi Gaussov šum
- nevronski mreži za $\mu_\theta(x_t, t)$
  $\Sigma_\theta(x_t, t)$
- **Reparameterization Trick:**
  $\mathcal{N}(\mu, \sigma^2) = \mu + \sigma \cdot \epsilon$

$$\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t$$

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t\mathbf{I})$$

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

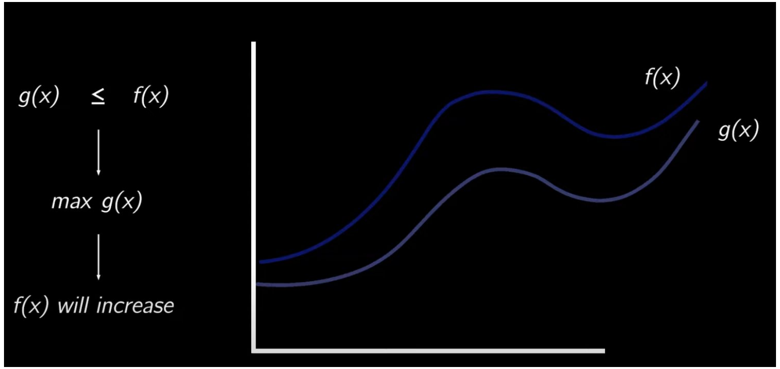
$$x_{t-1} = \mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$$

Treniranje difuzijskega modela

- funkcija izgube: negativni logaritem upanja (*log likelihood*)
- spodnja variacijska meja
- dejanski šum - napovedovan šum

$$L_0 := -\log p_\theta(x_0|x_1)$$

$$L_{\text{vlb}} := L_0 + L_1 + ... + L_{T-1} + L_T$$



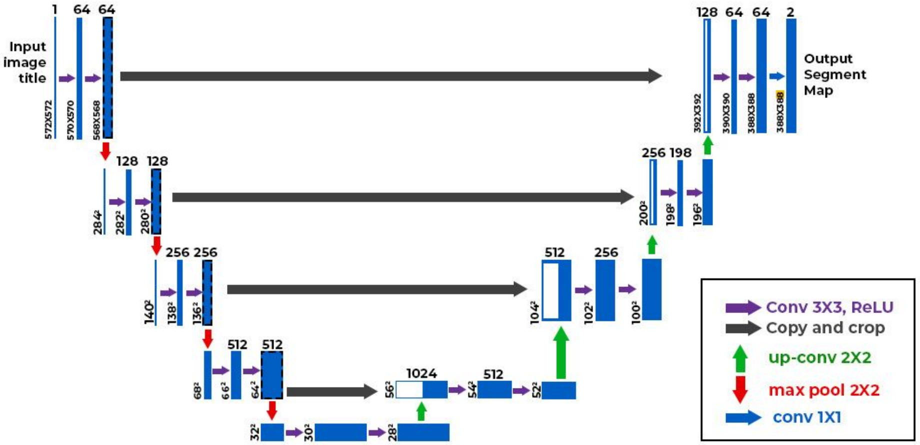
$$L_{t-1} := q(x_{t-1}|x_t, x_0)p_\theta(x_{t-1}|x_t)$$

$$L_T := q(x_T|x_0)p(x_T)$$

$$L_{\text{preprost}} := E_{t\sim[1,T],x_0\sim q(x_0),\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[||\epsilon - \epsilon_\theta(x_t, t)||^2\right]$$

Algoritmi in arhitektura

- U-Net
- pozornostni bloki
- preskočne povezave
- vgrajena povezava šuma ob *t*



Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Rezultati

- difuzija z vodenjem
- FID - Fréchetova začetna razdalja
- BigGAN (FID 6.95, levo), DDPM (FID 4.59, sredina), vzorci iz kompleta za treniranje (desno)

- boljša kvaliteta
- več raznolikosti
- počasno vzorčenje

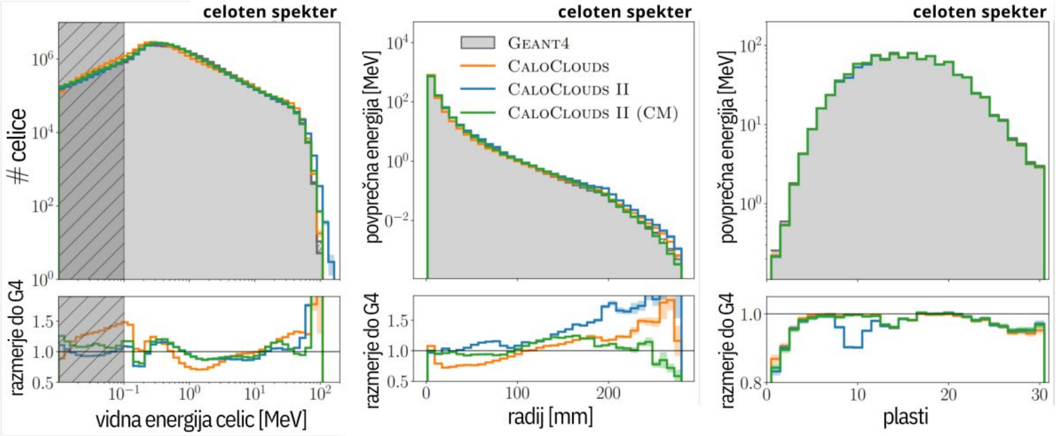
Uporaba v visoko energijski fiziki

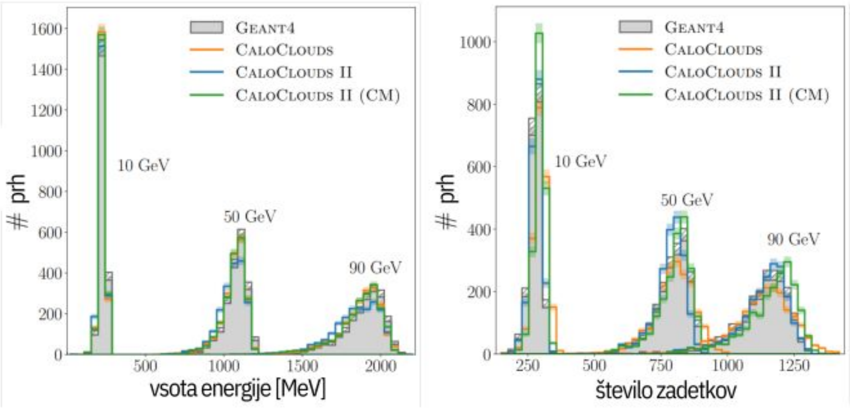
- CaloClouds - DDPM
- CaloClouds II - Score Matching
- CaloClouds II (CM) - distilacijska tehnika
- oblak točk, ne-fiksna struktura

- polni spekter (10 - 90 GeV)

- Histogram celične porazdelitve energije (levo), profila radialnega (na sredini) in vzdolžnega profila pljuskov (desno)

- Vidna porazdelitev vsote energije (levo) in števila zadetkov (celice z deponirano energijo nad polovico praga) (desno)





Uporaba v visoko energijski fiziki

- 25 × 2000 pljuskov
- energija 10 - 90 Gev
- Wasserstein metrika razdalje ‘Earth-mover’s distance’

- CPU v trenutni infrastrukturi
- GPU NVIDIA® A100
- GPU hitrejši in dražji

Simulator	$W_1^{N_{\text{hits}}}$ ($\times 10^{-3}$)	$W_1^{E_{\text{vis}}/E_{\text{inc}}}$ ($\times 10^{-3}$)	$W_1^{E_{\text{cell}}}$ ($\times 10^{-3}$)	$W_1^{E_{\text{long}}}$ ($\times 10^{-3}$)	$W_1^{E_{\text{radial}}}$ ($\times 10^{-3}$)	$W_1^{m_{1,X}}$ ($\times 10^{-3}$)	$W_1^{m_{1,Y}}$ ($\times 10^{-3}$)	$W_1^{m_{1,Z}}$ ($\times 10^{-3}$)
GEANT4	0.7 ± 0.2	0.8 ± 0.2	0.9 ± 0.4	0.7 ± 0.8	0.7 ± 0.1	0.9 ± 0.1	1.1 ± 0.3	0.9 ± 0.3
CALOCLOUDS	$\mathbf{2.5 \pm 0.3}$	11.4 ± 0.4	15.9 ± 0.7	$\mathbf{2.0 \pm 1.3}$	38.8 ± 1.4	4.0 ± 0.4	8.7 ± 0.3	1.4 ± 0.5
CALOCLOUDS II	3.6 ± 0.5	26.4 ± 0.4	$\mathbf{15.3 \pm 0.6}$	3.7 ± 1.6	11.6 ± 1.5	$\mathbf{2.4 \pm 0.4}$	$\mathbf{7.6 \pm 0.2}$	3.9 ± 0.4
CALOCLOUDS II (CM)	6.1 ± 0.7	$\mathbf{9.8 \pm 0.5}$	16.0 ± 0.7	$\mathbf{2.0 \pm 1.4}$	$\mathbf{8.3 \pm 1.9}$	3.0 ± 0.4	9.5 ± 0.6	$\mathbf{1.2 \pm 0.5}$

Strojna Oprema	Simulator	NFE	Velikost Serije	Čas / Prha [ms]	Pospešitev
CPU	GEANT4			3914.80 ± 74.09	$\times 1$
	CALOCLOUDS	100	1	3146.71 ± 31.66	$\times 1.2$
	CALOCLOUDS II	25	1	651.68 ± 4.21	$\times 6.0$
	CALOCLOUDS II (CM)	1	1	84.35 ± 0.22	$\times 46$
GPU	CALOCLOUDS	100	64	24.91 ± 0.72	$\times 157$
	CALOCLOUDS II	25	64	6.12 ± 0.13	$\times 640$
	CALOCLOUDS II (CM)	1	64	2.09 ± 0.13	$\times 1873$

Difuzijski modeli v industriji

Midjourney: Avgust 2022, Jason M. Allen

Sora: https://openai.com/sora

DALL-E integriran v ChatGPT4

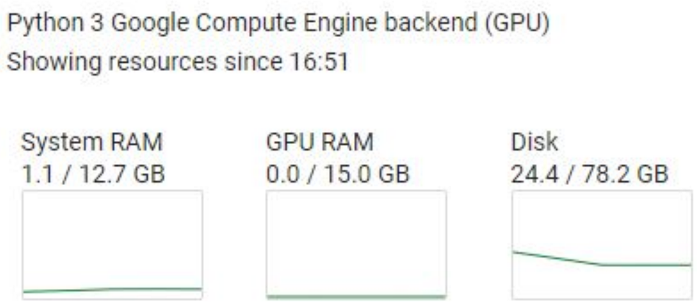
Kaj je resnično?

DIY

Jupiter Notebook:
https://colab.research.google.com/drive/1O8a6Wv12z_Aqh03J3HFOj1mAIkL-_GH6?usp=sharing

python + colab

Hugging Face



DIY

```
#diffusers is a hugging face page for using diffusion models from huggingface hub
!pip install diffusers transformers
```

```
from diffusers import StableDiffusionPipeline
import matplotlib.pyplot as plt
import torch
```

```
model_id1 = "dreamlike-art/dreamlike-diffusion-1.0"

pipe = StableDiffusionPipeline.from_pretrained(model_id1, torch_dtype=torch.float16)
pipe = pipe.to("cuda")
```

```
prompt = """Oppenheimer talking with Einstein near a lake"""
```

```
image = pipe(prompt).images[0]
```

```
print("[PROMPT]: ",prompt)
plt.imshow(image);
plt.axis('off');
```