Generowanie cyfr pisanych odręcznie na bazie sieci neuronowej o architekturze autoencodera

Analiza i implementacja różnych modeli generatywnych

Prowadzący: Adam Świtoński

Politechnika Śląska

Maj 2024

Plan prezentacji

Opis projektu

Cel projektu

Zastosowanie sieci neuronowej o architekturze autoencodera do generowania niskorozdzielczych obrazów cyfr pisanych odręcznie.

- Trenowanie różnych wariantów autoencoder'a
- Generowanie nowych cyfr poprzez podawanie losowych wartości na wejście dekodera
- Badanie wpływu struktury sieci (liczba warstw, liczba neuronów) oraz parametrów uczenia
- Wykorzystanie bazy danych MNIST

Rozważane warianty

Klasyczny autoencoder, wariacyjny autoencoder (VAE), GAN, Diffusion, VQ-VAE, Conditional VAE

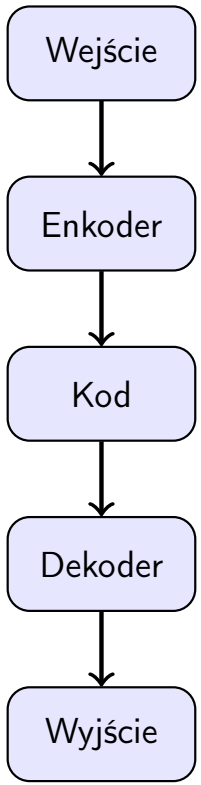
Autoencoder

Autoencoder to rodzaj sieci neuronowej, która uczy się kompresować dane wejściowe do reprezentacji o niższym wymiarze (kod), a następnie rekonstruować oryginalne dane z tej reprezentacji.

Zastosowania:

- Redukcja wymiarowości
- Denoising (odszumianie)
- Generowanie nowych danych

Zalety: prostota, szybki trening, interpretowalność
Wady: ograniczona zdolność generatywna, brak kontroli nad rozkładem latentnym
Bibliografia: [2]

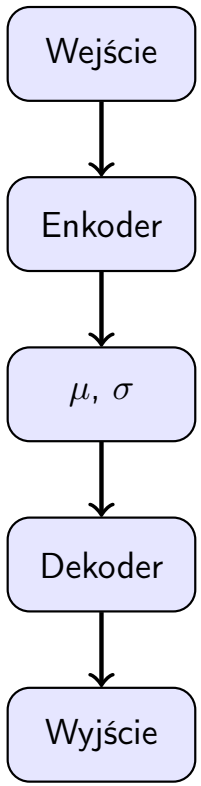


Wariacyjny Autoencoder (VAE)

VAE to probabilistyczne rozszerzenie autoencodera, które modeluje rozkład latentny danych.

Kluczowe cechy:

- Modelowanie rozkładu latentnego (μ, σ)
- Regularyzacja poprzez KL-dywergencję
- Generowanie nowych próbek przez próbkowanie



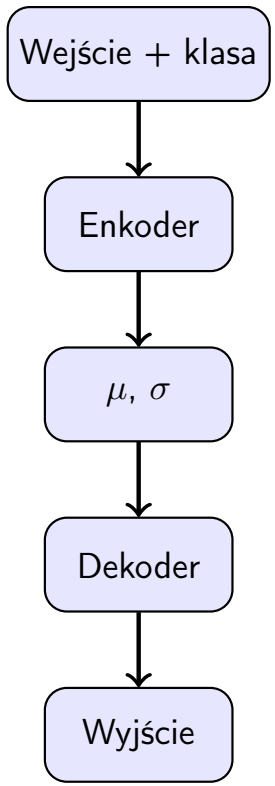
Zalety: generatywność, ciągła przestrzeń latentna, możliwość interpolacji
Wady: rozmyte próbki, trudność w trenowaniu
Bibliografia: [4]

Conditional VAE

Conditional VAE (CVAE) to wariacyjny autoencoder, który dodatkowo warunkuje generowanie na zadanej klasie (np. cyfra).

Kluczowe cechy:

- Warunkowanie generowania na etykietach
- Możliwość sterowania procesem generacji
- Łączenie etykiet z danymi wejściowymi



Zalety: kontrola nad generowanymi danymi, elastyczność
Wady: większa złożoność, wymaga etykiet
Bibliografia: [6]

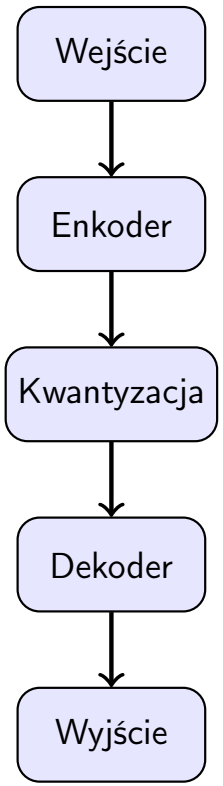
VQ-VAE

VQ-VAE to autoencoder, w którym przestrzeń latentna jest kwantyzowana do skończonego zbioru wektorów (słownik kodów).

Kluczowe cechy:

- Dyskretna przestrzeń latentna
- Kwantyzacja wektorowa
- Słownik kodowy

Zalety: dyskretna reprezentacja, dobre wyniki w generowaniu sekwencji
Wady: trudność w trenowaniu, konieczność doboru rozmiaru słownika
Bibliografia: [5]



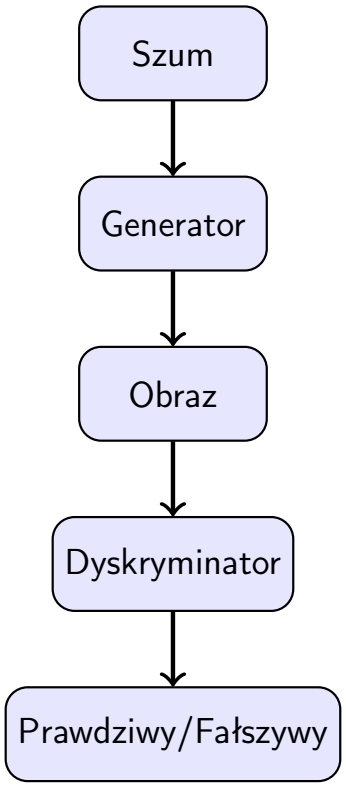
Generative Adversarial Network (GAN)

GAN to model generatywny składający się z dwóch sieci: generatora (tworzy próbki) i dyskryminatora (odróżnia próbki prawdziwe od fałszywych).

Kluczowe cechy:

- Układ rywalizujący (gra dwuosobowa)
- Generator tworzy coraz lepsze próbki
- Dyskryminator staje się coraz trudniejszy do oszukania

Zalety: realistyczne próbki, duża elastyczność
Wady: trudność w trenowaniu, niestabilność, mode collapse
Bibliografia: [1]



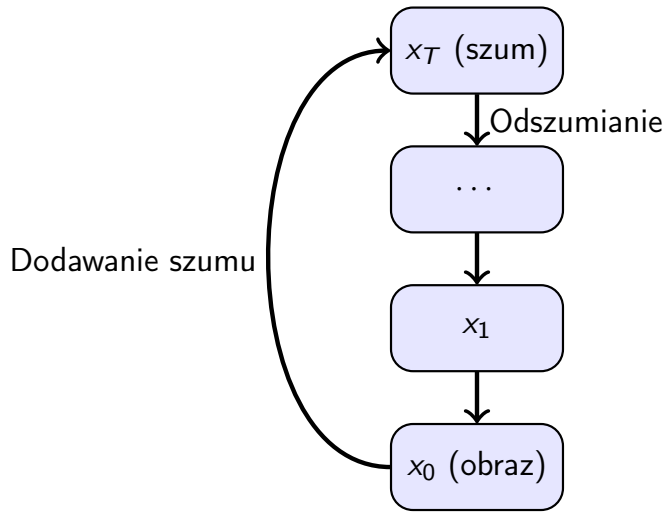
Diffusion Model

Model dyfuzji to nowoczesny model generatywny, który uczy się odszumiania danych przez odwracanie procesu stopniowego dodawania szumu.

Kluczowe cechy:

- Proces forward (dodawanie szumu)
- Proces reverse (przewidywanie i usuwanie szumu)
- Iteracyjne próbkowanie

Zalety: wysoka jakość generowanych próbek, stabilność treningu
Wady: długi czas generowania, złożoność obliczeniowa
Bibliografia: [3]



Porównanie modeli generatywnych

Model	**Zalety**	**Wady**
Autoencoder (2006)	Prostota implementacji, szybki trening	Słaba generatywność, rozmyte obrazy
VAE (2013)	Solidne podstawy teoretyczne, ciągła przestrzeń latentna	Rozmyte obrazy, trudność balansowania rekonstrukcji i KL divergencji
GAN (2014)	Ostre, realistyczne próbki	Niestabilność treningu, mode collapse
Conditional VAE (2015)	Kontrola nad procesem generacji, warunkowanie na klasach	Większa złożoność implementacji, wymaga etykiet
VQ-VAE (2017)	Ostrzejsze obrazy, dobra kompresja	Trudniejszy do trenowania, problemy z kwantyzacją
Diffusion (2020)	Najlepsza jakość obrazów, stabilny trening	Powolne próbkowanie, wysoka złożoność obliczeniowa

Wyzwania implementacyjne

- **Dobór architektury:** Liczba warstw, liczba neuronów, funkcje aktywacji
- **Dobór wymiarowości przestrzeni latentnej:** Zbyt mała - utrata informacji, zbyt duża - brak generalizacji
- **Balansowanie funkcji straty:** Np. w VAE balans między rekonstrukcją a regularyzacją KL
- **Stabilność treningu:** Szczególnie w przypadku GAN-ów
- **Efektywność obliczeniowa:** Modele dyfuzji wymagają wielu kroków podczas generowania
- **Ocena jakości wygenerowanych próbek:** Metody ilościowe vs jakościowe

Przykłady zastosowań

Generowanie danych syntetycznych:

- Augmentacja danych w uczeniu maszynowym
- Syntetyczne dane dla trenowania innych modeli
- Generowanie przykładów do zastosowań edukacyjnych

Zastosowania praktyczne:

- Transfer stylu pisma
- Uzupełnianie brakujących fragmentów
- Korekta i poprawa pisma odręcznego
- Konwersja cyfr między różnymi stylami

Bibliografia I

[1] Ian Goodfellow i in. "Generative adversarial nets". W: *Advances in neural information processing systems*. 2014, s. 2672–2680.

[2] Geoffrey E Hinton i Ruslan R Salakhutdinov. "Reducing the dimensionality of data with neural networks". W: *Science* 313.5786 (2006), s. 504–507.

[3] Jonathan Ho, Ajay Jain i Pieter Abbeel. "Denoising diffusion probabilistic models". W: *Advances in neural information processing systems*. T. 33. 2020, s. 6840–6851.

[4] Diederik P Kingma i Max Welling. "Auto-encoding variational bayes". W: *arXiv preprint arXiv:1312.6114* (2013).

[5] Aaron van den Oord, Oriol Vinyals i Koray Kavukcuoglu. "Neural discrete representation learning". W: *Advances in neural information processing systems*. 2017, s. 6306–6315.

Bibliografia II

[6] Kihyuk Sohn, Xinchen Yan i Honglak Lee. “Learning structured output representation using deep conditional generative models”. W: *Advances in neural information processing systems*. 2015, s. 3483–3491.