

Deep Learning 2 (na 26.11.23)

1. GAN.

Przeanalizować sieć GAN i napisać elementarne kody pokazujące jak działają oddzielnie dyskryminator i generator oraz jak działa cały model GAN.

2. GAN.

Za pomocą sieci GAN wygenerować dane dopasowane do zadanego rozkładu. Dane prawdziwe generować jako funkcję lekko zaburzoną szumem, np. wielomian n -tego stopnia, sinusoidę $(-\pi, \pi)$, segment okręgu, kwadratu, itp., etykieta=1. Dane generowane generować z losowych wektorów o wymiarze P , etykieta=0. Zwizualizować co epokę jak punkty generowane coraz lepiej pasują do rozkładu punktów prawdziwych.

3. GAN.

Korzystając z sieci GAN napisać własny generator przykładów. Trening generatora proszę wizualizować co epokę na wszystkich klasach. Generator dobrze szkolony będzie generować coraz lepsze obrazy przedstawicieli klas, aż w końcu trudno będzie odróżnić zbiór rzeczywisty, od wygenerowanego.

4. GAN.

Pokazać w kodzie (działający) przykład dowolnej sieci ze zbioru: CycleGAN, StyleGAN, cGAN.

5. Przykłady adversarialne.

Proszę wytrenować sieć CNN do dobrego rozpoznawania przedstawicieli klas z danych. Następnie proszę poeksperymentować z losowym zaburzeniem obrazu z danej klasy tak, aby skutecznie zmylić klasyfikator CNN.

6. Autoencoder.

Zbudować koder i dekodek (autoencoder) własnych przykładów, nie obrazów. Celem treningu takiego układu jest dobra kompresja modelu, która daje wgląd w tzw. przestrzeń ukrytą cech (latent space). Zwizualizować grupy punktów należące do różnych klas w przestrzeni ukrytej.

7. Autoencoder.

Zbudować Autoencoder obrazów składający się z kodera ($N \times (\text{Conv2D} + \text{MaxPool})$) i dekodera ($N \times \text{Conv2DTranspose}$), gdzie w każdej z tych części dokonuje się stopniowa kompresja/dekompresja obrazu. W obu częściach sieci zbadać przydatność różnych funkcji aktywacji (ostatnią funkcją aktywacji w dekodzie powinien być zasadniczo "sigmoid"). Warstwa Conv2DTranspose działa jak odwrotna konwolucja (nie w sensie matematycznym, tylko funkcjonalnym, tj. z "filtru" robi "obraz"). Dobroć zmiennych warstwy ukrytej przetestować na zbiorze walidacyjnym.

8. VAE.

Generować obiekty za pomocą VAE (Variational Autoencoder).

9. RBM.

Zaimplementować samodzielnie uczenie typu CD (Contrastive Divergence) na podstawie wykładu, pisząc procedury przejścia pomiędzy warstwą "wizualną" i ukrytą w sieci RBM. Zamodelować macierz W , wektory h i v oraz proces losowania. Sprawdzić na prostym przykładzie.

10. RBM.

Użyć mechanizmu CD do wyuczenia z danych reprezentacji ukrytej i wykorzystać tę reprezentację do przewidzenia wartości "wizualnej". Tego typu sieć RBM nadaje się do tworzenia rekomendacji tj. znajdowania powiązań typu użytkownik-produkt z wiedzy o częściowych wyborach różnych produktów przez różnych użytkowników. Użyć zbiorów dotyczących preferencji filmowych lub książkowych, itp.