Modele generatywne na chmurach punktów 3D

Jakub Zadrożny

Maj 2019

1 Podstawowe VAE

W pierwszej części projektu zaimplementowany został podstawowy autoenkoder wariacyjny. Dalej zakładamy, że dysponujemy zbiorem danych treningowych

$$\mathcal{X} = \{x_i \in \mathbb{R}^d\}_{i \in I}$$

dla pewnego d – wymiaru danych.

Ponadto zakładamy, że dane są obserwacjami zmiennej losowej o rozkładzie następującej postaci

$$f(z, x; \theta) = f(z) f(x|z; \theta) \tag{1}$$

Dodatkowo niech

$$\begin{aligned} z &\sim \mathcal{N}(0, I_k) \\ x|z &\sim \mathcal{N}(\mu_x(z;\theta), \mu_\sigma(z;\theta) I_d) \end{aligned} \tag{2}$$

gdzie μ_x, μ_σ są skomplikowanymi obliczeniami wykonywanymi przez sieć neuronową sparametryzowaną przez θ.

1.1 ELBO

Naszym celem jest odtworzenie parametrów rozkładu generującego θ oraz rozkładu $f(z|x;\theta)$, który nazywamy *reprezentacją* danych generowanych przez proces opsiany w (1) oraz (2).

Niestety z powodu zastosowania skomplikowanych, nieliniowych transformacji dokładne odtworzenie rozkładu $f(z|x;\theta)$ jest niemożliwe. W tym celu wprowadzamy pewne przybliżenie tego rozkładu – nazwijmy je $g(z|x;\phi)$.

Niech $g(z|x;\theta)$ będzie gęstością rozkładu normalnego ze średnią $\rho_x(x;\phi)$ i wariancją $\rho_\sigma(x;\phi)$, gdzie ρ_x, ρ_σ są reprezentowane przez sieci neuronowe parametryzowane przez ϕ. Wtedy

$$\begin{aligned} D_{KL}(g(z|x;\phi)||f(z|x;\theta)) &= \mathbb{E}_{z\sim g(z|x;\phi)}\left[-\log\frac{f(z|x;\theta)}{g(z|x;\phi)}\right] = \\ &= \mathbb{E}_{z\sim g(z|x;\phi)}\left[-\log\frac{f(z|x;\theta)f(x;\theta)}{g(z|x;\phi)f(x;\theta)}\right] = \\ &= \mathbb{E}_{z\sim g(z|x;\phi)}\left[-\log\frac{f(z|x;\theta)f(x;\theta)}{g(z|x;\phi)}\right] + \mathbb{E}_{z\sim g(z|x;\phi)}\left[\log f(x;\theta)\right] = \\ &= \mathbb{E}_{z\sim g(z|x;\phi)}\left[-\log\frac{f(z,x;\theta)}{g(z|x;\phi)}\right] + \log f(x;\theta) \end{aligned} \tag{3}$$

Zatem

$$\log f(x;\theta) = D_{KL}(g(z|x;\phi)||f(z|x;\theta)) + \mathbb{E}_{z\sim g(z|x;\phi)}\left[\log\frac{f(z,x;\theta)}{g(z|x;\phi)}\right] \tag{4}$$

Ponieważ $D_{KL}(\cdot\,||\cdot\,) \geqslant 0$, więc

$$\begin{aligned} \log f(x;\theta) &\geqslant \mathbb{E}_{z\sim g(z|x;\phi)}\left[\log\frac{f(z,x;\theta)}{g(z|x;\phi)}\right] = \\ &= \mathbb{E}_{z\sim g(z|x;\phi)}\left[\log\frac{f(x|z;\theta)f(z)}{g(z|x;\phi)}\right] = \\ &= \mathbb{E}_{z\sim g(z|x;\phi)}\left[\log f(x|z;\theta)\right] - \mathbb{E}_{z\sim g(z|x;\phi)}\left[-\log\frac{f(z)}{g(z|x;\phi)}\right] = \\ &= \mathbb{E}_{z\sim g(z|x;\phi)}\left[\log f(x|z;\theta)\right] - D_{KL}(g(z|x;\phi)||f(z)) \end{aligned} \tag{5}$$

Zatem dla dowolnego rozkładu aproksymującego $g(z|x;\phi)$ otrzymujemy dolne ograniczenie na prawdopodobieństwo wygenerowania zaobserwowanych danych. Dlatego część wzoru po prawej stronie od ostatniej równości nazywamy ELBO (*evidence lower bound*). Ponadto pierwszy składnik odpowiada jakości rekonstrukcji obserwacji ze zmiennej ukrytej z, więc nazywany jest kosztem rekonstrukcji, natomiast drugi to odległość KL rozkładu aproksymującego $f(z|x;\theta)$ od naszego założenia na jego temat.

1.2 Zadanie optymalizacyjne

Chcemy znaleźć układ parametrów $< \theta, \phi >$, który daje najlepszą gwarancję na prawdopodobieństwo wygenerowania zaobserwowanych danych (ELBO). W tym celu posłużymy się lekko zmodyfikowanym algorytmem SGD. Naszym zadaniem jest znalezienie

$$\begin{aligned}\max_{\theta,\phi} \hat{\mathcal{L}}(\mathcal{X}, \theta, \phi) &= \sum_{i \in I} \mathcal{L}(x_i, \theta, \phi) = \\ &= \sum_{i \in I} \left(\mathbb{E}_{z \sim g(z|x_i;\phi)} \left[\log f(x_i|z;\theta) \right] - D_{KL}(g(z|x_i;\phi)||f(z)) \right)\end{aligned} \tag{6}$$

Ponieważ bardziej naturalnym zadaniem jest minimalizowanie funkcji kosztu, to rozwiążemy równoważne zadanie znalezienia

$$\min_{\theta,\phi} -\hat{\mathcal{L}}(\mathcal{X}, \theta, \phi) \tag{7}$$

Żeby posłużyć się algorytmem SGD musimy umieć wyliczać i różniczkować oba składniki funckji (L).

1.2.1 Koszt KL

Odległość KL dwóch rozkładów normalnych o następujących parametrach

$$\begin{aligned}\mathcal{N}_0 &\sim \mathcal{N}(\mu_0, \Sigma_0) \\ \mathcal{N}_1 &\sim \mathcal{N}(\mu_1, \Sigma_1)\end{aligned}$$

dla pewncyh $\mu_0,\ \mu_1 \in \mathbb{R}^k,\ \Sigma_0, \Sigma_1 \in \mathbb{R}^{k \times k}$, wynosi

$$D_{KL}(\mathcal{N}_0||\mathcal{N}_1) = \frac{1}{2} \left(\operatorname{tr}(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T \Sigma_1^{-1}(\mu_1 - \mu_0) - k + \log \frac{\det \Sigma_1}{\det \Sigma_0} \right)$$

Ponieważ zakładamy, że $f(z)$ jest rozkładem $z \sim \mathcal{N}(0, I_k)$, więc

$$D_{KL}(g(z|x;\phi)||f(z)) = \frac{1}{2} \sum_{i=1}^{k} \left(\rho_x(x;\phi)_i^2 + \rho_\sigma(x;\phi)_i^2 - \log(\rho_\sigma(x;\phi)_i^2) - 1 \right) \tag{8}$$

Wzór (8) można wyliczać i różniczkować analitycznie.

1.2.2 Koszt rekonstrukcji

Drugiego składnika funckji $\mathcal{L}$, czyli kosztu rekonstrukcji, nie da się wyznaczyć analitycznie. Aby objeść ten problem, możemy metodą Monte Carlo oszacować wartość oczekiwaną przez średnią

$$\mathbb{E}_{z\sim g(z|x;\phi)}\left[\log f(x_i|z;\theta)\right] \sim \frac{1}{m}\sum_{i=1}^{m} -\log f(x|z_i;\theta)$$

gdzie $z_i \sim g(z|x;\phi)$. Taką wartość potrafimy już wyliczyć, ale nie potrafimy propagować gradientu do parametrów ϕ przez zaobserwowane wartości z_i.

Wprowadzimy repearametryzację zmiennych z_i – możemy zauważyć, że zmienna $z_i = \rho_x(x;\phi) + \epsilon_i\rho_\sigma(x;\phi)$ gdzie $\epsilon_i \sim \mathcal{N}(0, I_k)$ ma rozkład $g(z|x;\phi)$ a ponadto możemy propagować gradient do parametrów ϕ. Otrzymaliśmy zatem następujące przybliżenie na $\mathcal{L}$

$$\mathbb{E}_{z\sim g(z|x;\phi)}\left[\log f(x_i|z;\theta)\right] \sim \frac{1}{m}\sum_{i=1}^{m} -\log f(x|z_i = \rho_x(x;\phi) + \epsilon_i\rho_\sigma(x;\phi);\theta)$$

gdzie $\epsilon_i \sim \mathcal{N}(0, I_k)$

Ponieważ $x|z \sim \mathcal{N}(\mu_x(z;\theta), \mu_\sigma(z;\theta)I_d)$, więc

$$-\log f(x|z;\theta) = \sum_{i=1}^{d}\left(\frac{1}{2}\log(2\pi) + \log(\mu_\sigma(z;\theta)_i) + \frac{(x-\mu_x(z;\theta)_i)^2}{2\mu_\sigma(z;\theta)_i^2}\right)$$

jednak metryka ta niezbyt dobrze nadaje się do chmur punktów, ponieważ np. chcielibyśmy uznawać permutację punktów oryginalnej chmury za dobrą rekonstrukcję. Dlatego zamiast wyliczać *stricte* $\log f(x|z;\theta)$ skorzystamy ze zmodyfikowanego *Chamfer distance* danego wzorem

$$CD(\mathcal{X}_1, \mathcal{X}_2) = \sum_{x\in\mathcal{X}_1}\min_{y\in\mathcal{X}_2}||x-y||_2^2 + \sum_{x\in\mathcal{X}_2}\min_{y\in\mathcal{X}_1}||x-y||_2^2 \tag{9}$$

gdzie $\mathcal{X}_1$, $\mathcal{X}_2$ są zbiorami punktów wielowymiarowych. Ściślej mówiąc, możemy potraktować $\mu_x(z;\theta) \in \mathbb{R}^{3\cdot m}$ jako chmurę m punktów trójwymiarowych, oznaczmy ją $\hat{y}$. Ponadto dla $y \in \hat{y}$ niech $\sigma(y)$ oznacza 3-elementowy wektor wariancji $\mu_\sigma(z;\theta)$ utworzony ze składowych odpowiadających y. Za koszt rekonstrukcji przyjmiemy

$$\begin{aligned}\mathcal{L}_{rec}(\hat{x}|z,\theta,\phi) = &\sum_{x\in\hat{x}}\min_{y\in\hat{y}}\left(-\log p_{y,\sigma(y)}(x)\right) + \\ &+\sum_{y\in\hat{y}}\min_{x\in\hat{x}}\left(-\log p_{x,\sigma(y)}(y)\right)\end{aligned} \tag{10}$$

gdzie $p_{v,s}(x)$ jest gęstością rozkładu normalnego o średniej v i macierzy kowariancji sI w punkcie x.

Po usunięcu stałych wyrazów można to zapisać jako

$$\begin{aligned}\mathcal{L}_{rec}(\hat{x}|z,\theta,\phi) = &\sum_{x\in\hat{x}} \min_{y\in\hat{y}} \sum_{i=1}^{3} \left(\log(\sigma(y)_i) + \frac{(x_i - y_i)^2}{2\sigma(y)_i^2} \right) + \\ &+ \sum_{y\in\hat{y}} \min_{x\in\hat{x}} \sum_{i=1}^{3} \left(\log(\sigma(y)_i) + \frac{(x_i - y_i)^2}{2\sigma(y)_i^2} \right)\end{aligned} \tag{11}$$

W obecnej wersji modelu dla uproszczenia przyjęto, że $\mu_\sigma(z;\theta) = \alpha$ dla wszystkich z i niezależnie od parametrów θ (tzn. przyjęto stałą wariancję dla danych wyjściowych). Wtedy wzór (11) upraszcza się do

$$\begin{aligned}\mathcal{L}_{rec}(\hat{x}|z,\theta,\phi) &= \frac{1}{2\alpha^2} \left(\sum_{x\in\hat{x}} \min_{y\in\hat{y}} ||x-y||_2^2 + \sum_{y\in\hat{y}} \min_{x\in\hat{x}} ||x-y||_2^2 \right) = \\ &= \frac{1}{2\alpha^2} CD(\hat{x},\hat{y})\end{aligned} \tag{12}$$

1.3 Implementacja praktyczna

W tej sekcji przedstawiona została praktyczna implementacja teoretycznego zadania optymalizacyjnego opsianego w sekcji **??**. Ściślej, zaimplementowany został model uczący się reprezentacji danych ze zbioru *Modelnet40* (**??**).

1.3.1 *Modelnet40*

1.3.2 Przestrzeń *reprezentacji danych*

W rozważanej implementacji przyjęto 128-wymiarową przestrzeń *reprezentacji danych*.

1.3.3 Architektura modelu

W opisie rozwiązywanego zagadnienia optymalizacyjnego (sekcja **??**) użyte zostały abstrakcyjne obliczenia ρ oraz μ realizowane przez sieci neuronowe. Do praktycznej implementacji należy nadać tym obliczeniom pewną architekturę. Naturalnym pierwszym rozwiązaniem jest realizowanie tych obliczeń przez zwykłą sieć MLP (z dodatkiem *batch normalization* [**??**]). Obliczenia

$\rho_x,\ \rho_\sigma$ są realizowane przez jedną sieć MLP, natomiast μ_x przez drugą (przypomnijmy, że dla uproszczenia przyjęto, że μ_σ jest stałe). Do testów przyjęto, że obie sieci $\rho_1,\ \mu_1$ są 4-warstwowe, w której każda ukryta warstwa składa się z 1024 neuronów.

Dalej rozważono również bardziej skomplikowaną sieć ρ opartą na architekturze *Pointnet* [**??**]. Dokładniej, sieć ρ_2 jest początkowym fragmentem sieci *Pointnet* wyszukującym poszczególne cechy (*features*) z doklejoną warstwą wyjściową (warstwa wyjściowa jest w pełni połączona z wyjściem sieci *Pointnet*). Sieć μ_2 pozostaje prostym MLP, dokładnie takim, jak μ_1.

W tabeli (**??**) znajduje się porównanie dwóch zaproponowanych powyżej architektur. Możemy zauważyć, że oba podejścia osiągają podobne wyniki na danych treningowych, jednak prosta sieć MLP znacznie bardziej *overfituje* niż sieć oparta na *Pointnecie*. Z racji tego, że drugi zestaw sieci znacznie lepiej się generalizuje, do dalszej pracy i testów przyjmiemy sieci oprate na *Pointnecie*.

1.3.4 Trenowanie modelu

Do optymalizacji funckji kosztu danej wzorem (**??**) użyto metody ADAM [**??**]. Początkowe learning rate wynosiło 0.0002 i zmniejszało się o połowę, co każde 1000 epok. Model trenowany był przez 5000 epok (duża liczba epok wynika z niewielkiej ilości próbek). Na rys **??** przedstawiona została zmiana całkowitej funkcji kosztu, kosztu rekonstrukcji oraz kosztu KL w czasie.

1.3.5 Wyniki eksperymentalne

Na wytrenowanym modelu przeprowadzono kilka eksperymentów mających na celu sprawdzić zarówno zdolności modelu do dokładnej rekonstrukcji, jak i jego możliwości generatywne.

Na rys **??** znajdują się oryginalne chmury punktów ze zbioru *Modelnet40* (po lewej) wraz z chmurami zrekonstruowanymi przez dekoder na podstawie zmiennych pośrednich wyliczonych przez enkoder (po prawej).

Jedną z miar jakości rekonstrukcji autoenkoderów jest pokrycie (*coverage*). Pokrycie definiujemy jako procent próbek ze zbioru danych, dla którego najbliższa (w tym przypadku w sensie *Chamfer Distance*) inna próbka spośród całego zbioru dancyh oraz wszystkich rekonstrukcji pochodzi ze zbioru rekonstrukcji. Dla wytrenowanego modelu pokrycie na rozważanym podzbiorze *Modelnetu40* wynosi ...%.

Jedną z największych zalet VAE są jego zdolności generatywne. Klasycznym sposobem demonstracji zdolności generatywnych modelu jest skonstruowanie takiej interpolacji pomiędzy dwoma różnymi obiektami, że każdy z jej kroków pośrednich jest *podobny* (wizualnie lub z użyciem metryki) do próbek z oryginalnego zbioru danych. Rys **??** przedstawiają interpolacje wykonane przez rozważany powyżej model. Można zaobserwować, że kolejne kroki interpolacji coraz bardziej upodobniają obiekt źródłowy do docelowego, jednocześnie zachowując typowe cechy obiektów z oryginalnego zbioru danych.

Własnością wyróżniającą VAE na tle innych enkoderów jest możliwość odgórnego zadania rozkładu zmiennych pośrednich, który model będzie musiał osiągnąć. Dla rozważanego modelu zadano rozkład standardowy wielowymiarowy normalny (o średniej w 0 i identycznościowej macierzy kowariancji). Dzięki temu, możemy tworzyć nowe, nieistniejące w zbiorze danych próbki, przez wylosowanie zmiennej pośredniej z wybranego powyżej rozkładu i przekazanie jej do dekodera. Rys. **??** przedstawia chmury otrzymane w ten właśnie sposób. Możemy zauważyć, że powstałe próbki dobrze pasują do reszty zbioru danych i ponadto prezentują dużą różnorodność (pochodzą z różnych podklas), co świadczy o dużych możliwościach generatywnych modelu.

Rys **??** przedstawiają wykresy median rozkładów zwróconych przez enkoder po zredukowaniu do dwóch wymiarów metodami PCA (górny) i t-SNE (dolny). Możemy zauważyć, że na obu wykresach mediany rozmieszczone są dość jednostajnie i gęsto na kole o środku w punkcie $(0, 0)$. Oznacza to, że reprezentacja próbek ze zbioru jest *ściśnięta* do zera i gęsta, co umożliwia przeprowadzenie interpolacji oraz tworzenie syntetycznych próbek na podstawie wylosowanych zmiennych pośrednich (jak wyżej).