ベイジアンネットワーク

白坂貴規

平成 32 年 3 月 20 日

1 画像生成モデル

画像生成モデルとして, **変分オートエンコーダ (variational autoencoder, VAE)** という深層生成モデルがあった. オートエンコーダは, 隠れ層の次元は入出力の次元より小さくし, 入力と出力が同じになるように学習するものであった.

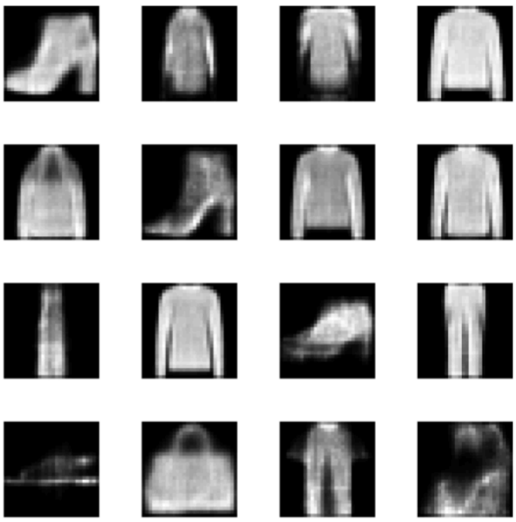

図 1: VAE が生成した画像データ

2 GAN

VAE は生成した画像がぼやけてしまうが, これは学習が足りてないのではなくモデルの枠組み, 学習の仕方に原因がある. VAE では, 生成モデルを正規分布あるいはベルヌーイ分布として明示的にモデル化し, 最尤推定により学習を行っていた. これにより, 誤差関数には 2 乗誤差あるいは交差エントロピー誤差の項が用いられることになるが, これを最小化しようとすると, 全体的に画素を曖昧にさせた方が画像全体として誤差は小さくなるので, VAE ではどうしても生成画像がぼやける傾向が出てしまう. これを改善するために生成モデルの分布をモデルを暗黙的にする GAN が考えられた. データ分布を $p_d(x)$, データ分布に近いモデル分布を $p_g(x)$ とおく. ここでは分布 $p_d(x)$ は分布の形が明示されていない. そのため尤度を測ることもできないので, まずはデータ分布とモデル分布の密度比 $r(x)$ を考える.

$$r(x) = \frac{p_d(x)}{p_g(x)} \tag{1}$$

ここで, データ分布あるいはモデル分布から生成されたラベル付きのデータ集合 $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ を考え, データ分布により生成されたデータのラベルを $y = 1$, モデル分布により生成されたデータのラベルを $y = 0$ とすると, それぞれの分布は次のようになる.

$$p_d(x) = p(x|y = 1) \tag{2}$$

$$p_g(x) = p(x|y = 0) \tag{3}$$

この時, 密度比 $r(x)$ は次のように変形できる.

$$\begin{aligned} r(x) &= \frac{p(x|y=1)}{p(x|y=0)} \\ &= \frac{p(y=1)p(x)}{p(y=1)} \frac{p(y=0)}{p(y=0|x)p(x)} \\ &= \frac{p(y=1|x)}{p(y=0|x)} \frac{1-\pi}{\pi} \end{aligned} \tag{4}$$

ただし,

$$\pi = p(y = 1) \tag{5}$$

である.$p(y = 1|x)$ を推定することができれば, $r(x)$ がもとまるので $p(y = 1|x)$ を近似する分布をパラメータ ϕ を用いて $p_\phi(y = 1|x)$ とする.

$$p(y = 1|x) \simeq q_\phi(y = 1|x) \tag{6}$$

こうすることで, 分布を NN で求めることができるようになる. この $q_\phi(y = 1|x)$ を推定するモデルのことを**識別器**あるいは**鑑別器 (discriminator)** という. 識別器を $D(\phi; x)$ と表す.

$$D(\phi; x) = q_\phi(y = 1|x) \tag{7}$$

これにより, 密度比を考える問題から, 確率的分類器の最適化, すなわち一般的な分類問題に置き換わっ

たことになる．よって誤差関数として，交差エントロピー誤差関数を考えると，

$$(D) = -E_{p(x,y)}[y\ln D(\phi;x)+(1-y)\ln(1-D(\phi;x))] \tag{8}$$

と表せる．符号の煩わしさから $-U(D)$ を考え，次のように式変形を行うと

$$\begin{aligned} & E_{p(x,y)}[y\ln D(\phi;x)+(1-y)\ln(1-D(\phi;x))] \\ = & E_{p(x|y)p(y)}[y\ln D(\phi;x) \\ & +(1-y)\ln(1-D(\phi;x))] \\ = & E_{p(x|y=1)p(y=1)}[\ln D(\phi;x)] \\ & +E_{p(x|y=0)p(y=0)}[\ln(1-D(\phi;x))] \\ = & \pi E_{p_d(x)}[\ln D(\phi;x)] \\ & +(1-\pi)E_{p_g(x)}[\ln(1-D(\phi;x))] \end{aligned} \tag{9}$$

ここで，データ集合に関して，各ラベルのデータが等しい時

$$\pi = 1-\pi = \frac{1}{2} \tag{10}$$

であるから，最終的な識別器の目的関数 $V(D)$ は

$$V(D) = E_{p_d(x)}[\ln D(\phi;x)] + E_{p_g(x)}[\ln(1-D(\phi;x))] \tag{11}$$

で，識別器の目的はこれを最大化することである．もし，最適な識別器 $D_*(\phi;x)$ が得られたとすると，

$$D_*(\phi;x) = D_*(x) = p(y=1|x) \tag{12}$$

となるが，この時

$$D_*(x) = \frac{r}{r+1} = \frac{p_d(x)}{p_d(x)+p_g(x)} \tag{13}$$

に収束する．これを目的関数 $V(D)$ に代入すると，

$$\begin{aligned} V(D_*) &= E_{p_d}[\ln\frac{p_d}{p_d+p_g}] + E_{p_g}[\ln(1-\frac{p_d}{p_d+p_g})] \\ &= \int p_d \ln\frac{p_d}{p_g}\,\mathrm{d}x + \int p_g \ln\frac{p_g}{p_d+p_g}\,\mathrm{d}x \\ &= \int p_d \ln\frac{2p_d}{p_d+p_g}\,\mathrm{d}x - \int p_d \ln 2\,\mathrm{d}x \\ &\quad + \int p_g \ln\frac{2p_g}{p_d+p_g}\,\mathrm{d}x - \int p_g \ln 2\,\mathrm{d}x \\ &= \int p_d \ln\frac{2p_d}{p_d+p_g}\,\mathrm{d}x \int p_g \ln\frac{2p_g}{p_d+p_g}\,\mathrm{d}x \\ &\quad -2\ln 2 \\ &= 2D_{JS}[p_d||p_g] - 2\ln 2 \end{aligned} \tag{14}$$

つまり，$V(D_*)$ は $p_d(x)$ と $p_g(x)$ の JS ダイバージェンスに対応している．ここで，の JS ダイバージェンスとは，KL ダイバージェンスを D_{KL} とすると

$$D_{JS}[p||q] = \frac{1}{2}D_{KL}[p||\frac{p+q}{2}] + \frac{1}{2}D_{KL}[q||\frac{p+q}{2}] \tag{15}$$

で定義されるものである．

次に生成モデルについて考える．潜在変数 z を仮定すると，周辺化により

$$p_g(x) = \int p(x|z)p(z)\,\mathrm{d}z \tag{16}$$

となるが，今度は $p(z|x)$ を近似する分布として $q_\theta(x|z)$ を導入する．

$$p(x|z) \simeq q_\theta(x|z) \tag{17}$$

こうすることで，この分布を NN で求められる．生成器を $G(\theta;z)$ で表すと，

$$G(\theta;z) = q_\theta(x|z) \tag{18}$$

と表す．最適な $G(\theta;z)$ を得るための目的関数は，D_* を用いて

$$V(D_*,G) = E_{p_d(x)}[\ln D_*(x)] + E_{p(z)}[\ln(1-D_*(G(\theta;z)))] \tag{19}$$

と表せ，生成器は $V(D_*,G)$ を最小化する．

3 まとめ

以上をまとめると，GAN は以下の識別器の学習と生成器の学習を片方を固定して交互に行うことになる．識別器の学習は，生成器 $G(\theta;z)$ を固定した上で，

$$\max_\theta E_{p_d(x)}[\ln D(\phi;x)] + E_{p(z)}[\ln(1-D(\phi;G(\theta;z)))] \tag{20}$$

を計算し，生成器の学習は，識別器 $D(\phi;x)$ を固定した上で

$$\min_\theta E_{p(z)}[\ln(1-D(\theta;G(\theta;z)))] \tag{21}$$

を計算する．

参考文献

[1] Generative Adversarial Nets https://arxiv.org/pdf/1406.2661.pdf

[2] 巣籠悠輔. 詳説ディープラーニング (生成モデル編) https://note.com/yusugomori/n/n945f51cabc03