Generalization of iterative sampling in autoencoders

No Author Given

No Institute Given

Supplementary materials

Proofs

Proposition 1. *A perfect lossy autoencoder of a random variable X with density p is a diffeomorphism r of $\mathbb{R}^n$ minimizing the following expected loss:*

$$\mathcal{L}_\lambda(r) = \mathbb{E}_{X\sim p}\left[||x - r(x)||_2^2 + \lambda \log|\mathscr{J} r|\right] \tag{1}$$

for some $\lambda > 0$, where $\mathscr{J} r$ being the Jacobian matrix $(\mathscr{J} r)_{ij} = \frac{\partial r_i}{\partial x_j}$.

Proof. As a perfect lossy autoencoder minimizes the loss $\mathcal{L}_2$ over the set of lossy models, by introducing a Langrangian multiplier $\lambda > 0$ it is equivalent to minimize over the whole universe of $\mathbb{R}^n$-diffeomorphisms with a penalization $\lambda.MI(X, r(X))$ of the loss. Since r is a function, then the joint distributiuon density $p(x, r(x))$ is the same as the density $p(x)$ and subsequently the mutual information can be rewritten as $\mathbb{E}_{X\sim p}\left[-\log p(r(x))\right]$.

Since r is a diffeomorphism, by change of variable: $p(r(x)) = p(x)|\mathscr{J}^{-1} r|$. Therefore, the loss to be minimized writes:

$$\mathcal{L}_\lambda(r) = \mathbb{E}_{X\sim p}\left[||x - r(x)||_2^2 - \lambda \log p(x)|\mathscr{J}^{-1} r|\right] \tag{2}$$

Since $p(x)$ is constant with respect to r, the minimizers of 3 are the same as the minimizers of:

$$\mathcal{L}_\lambda(r) = \mathbb{E}_{X\sim p}\left[||x - r(x)||_2^2 + \lambda \log|\mathscr{J} r|\right] \tag{3}$$

We now aim at describing the analytical solution of equation 1. We formulate the problem in an Euler-Langrange setting by defining the following multivariate function:

$$\mathcal{H}(x, r, r') = p(x).\left[||x - r||_2^2 + \lambda \log|r'|\right] \tag{4}$$

Where $x, r \in \mathbb{R}^n$, $r' \in \mathbb{R}^n \times \mathbb{R}^n$ and $|r'|$ denote the determinant of r'.

Using this formulation, finding a minimizer of 1 is equivalent to finding a minimizer r of $\int_{\mathbb{R}^n} \mathcal{H}(x, r(x), \mathscr{J} r)dx$.

Following [**?**], Volume 1, Chapter IV, eq. 18 and 25, it should satisfy in particular:

$$\frac{\partial \mathcal{H}}{\partial r_i} = \sum_j \frac{\partial}{\partial x_j}\frac{\partial \mathcal{H}}{\partial r'_{ij}} \quad \forall i = 1, ...n \tag{5}$$

We have the following derivatives:

$$\begin{aligned}
\frac{\partial \mathcal{H}}{\partial r_i} &= 2.p(x).(r_i(x) - x_i) \\
\frac{\partial \mathcal{H}}{\partial r'_{ij}} &= \lambda.p(x).(r'^{-1})_{ji} \\
\frac{\partial}{\partial x_j}\frac{\partial \mathcal{H}}{\partial r'_{ij}} &= \lambda.\frac{\partial p}{\partial x_j}.(r'^{-1})_{ji} \\
&\quad - \lambda.p(x).\Big[r'^{-1}.(\frac{\partial}{\partial x_j}r')r'^{-1})\Big]_{ji}
\end{aligned} \tag{6}$$

Assuming that $\forall x \in \mathbb{R}^n$ and $p(x) \neq 0$, replacing the derivatives in 5 and dividing it by $2.p(x)$ we get the following identity:

$$\begin{aligned}
r_i(x) =& x_i + \frac{\lambda}{2}\Big[\sum_j \frac{\partial \log p}{\partial x_j}.(\mathscr{J}^{-1}r)_{ji} \\
&- \sum_j \Big[\mathscr{J}^{-1}r.(\frac{\partial}{\partial x_j}\mathscr{J}r)\mathscr{J}^{-1}r)\Big]_{ji}\Big]
\end{aligned} \tag{7}$$

The local minima of the loss $\mathcal{L}_\lambda(r)$ are described by their first order expension following:

Proposition 2. *The first order term in the expansion with respect to λ of a minimizer of the loss defined in equation 1 is $\frac{1}{2}\frac{\partial \log p}{\partial x_i}$, and a perfect lossy autoencoder satisfies:*

$$r(x) = x + \tfrac{\lambda}{2}\tfrac{\partial \log p}{\partial x_i} + o(\lambda)$$

as $\lambda \to 0$.

Proof. Let us denote by $g(x)$ the first-order term of the expansion of r with respect to λ:

$$r(x) = x + \lambda.g(x) + o(\lambda) \tag{8}$$

Inducing the following expansions:

$$\begin{aligned}
\mathscr{J}r &= I + \lambda.\mathscr{J}g + o(\lambda) \\
\mathscr{J}^{-1}r &= I - \lambda.\mathscr{J}g + o(\lambda)
\end{aligned} \tag{9}$$

Substituting in equation 7 the expansions 8 and 9, we get:

$$\begin{aligned}
g_i(x) + o(1) =& \frac{1}{2}\sum_j \Big[\frac{\partial \log p}{\partial x_j}\Big[I - \lambda.\mathscr{J}g + o(\lambda)\Big]_{ji}. \\
&- \Big[(I - \lambda.\mathscr{J}g + o(\lambda))(\lambda\frac{\partial}{\partial x_j}\mathscr{J}g) \\
&(I - \lambda.\mathscr{J}g + o(\lambda))\Big]_{ji}\Big]
\end{aligned} \tag{10}$$

The only 0-order λ term comes from the first member of the summation, and we finally get:

$$g_i(x) = \frac{1}{2} \sum_j \frac{\partial \log p}{\partial x_j} I_{ji} = \frac{1}{2} \frac{\partial \log p}{\partial x_i} \tag{11}$$

Heuristic

Algorithm 1 Langevin with few restarts: heuristics to avoid oversampling high-density regions

Input:

- AEs (trained auto-encoders)
- C (Trained classifier)
- $R = 10$ (Random walk steps)
- $N_s = 10,000$ (Number of random walks)
- $\lambda = 0.1$ (Regularization coefficient)

Output:

- X = [ ] (Synthetic samples)
- Y = [ ](Synthetic labels)

for $n = 0$ to N_s **do**
  $x_{n_{AE}} \sim \mathcal{N}(0, I)$
  for $i = 0$ to R **do**
    if $i > 1$ **then**
      Draw $\epsilon_n \sim \mathcal{N}(0, I)$
      $x_{i+1_{AE}} = AE(x_{i_{AE}}) + (\lambda * \epsilon_n)$
      $X = X \cup [x_{i_{AE}}]$
      $Y = Y \cup [C.predict(x_{i_{AE}})]$
    end if
  end for
end for

Principal Component Analysis

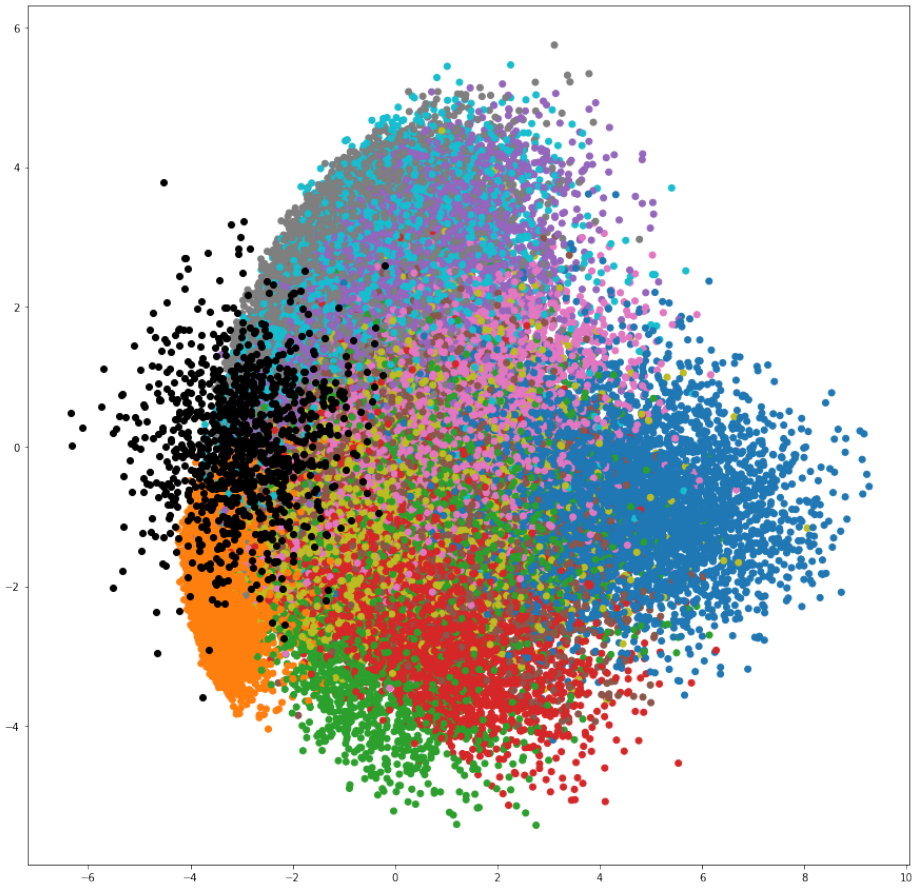

Fig. 1. Visualization of the training data of the MNIST dataset embedded in the vector subspace spanned by the first two principal components. After projecting 1000 random points ($x_n \sim \mathcal{N}(0, I)$) into the subspace, we observe a relatively clear separation between them (black dots) and the training set (colored dots).

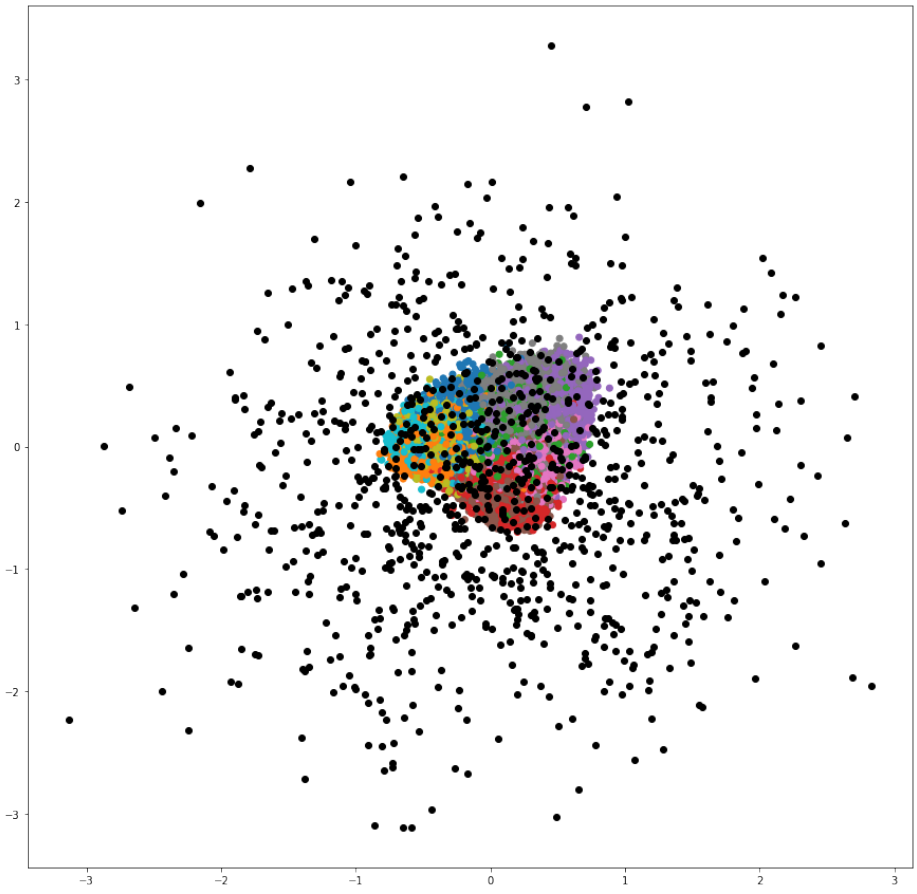

Fig. 2. Visualization of the training data of the CIFAR-10 dataset embedded in the vector subspace spanned by the first two principal components. After projecting 1000 random points ($x_n \sim \mathcal{N}(0, I)$) into the subspace, there is an overlapping between them (black dots) and the training set (colored dots).

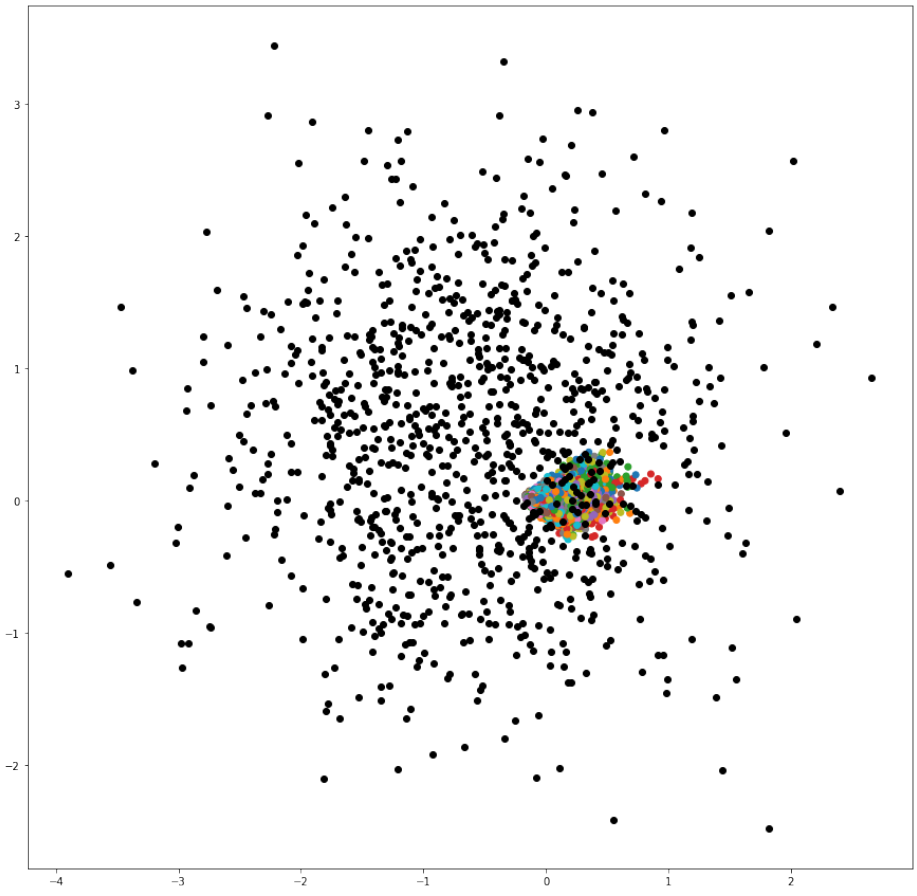

Fig. 3. Visualization of the training data of the CIFAR-100 dataset embedded in the vector subspace spanned by the first two principal components. After projecting 1000 random points ($x_n \sim \mathcal{N}(0, I)$) into the subspace, there is an overlapping between them (black dots) and the training set (colored dots).