TTIC 31230 Fundamentals of Deep Learning

Problems for VAEs

Problem 1. Mutual Information as Channel Capacity

The mutual information between two random variables x and y is defined by

$$I(x,y) = E_{x,y} \ \ln \frac{P(x,y)}{P(x)P(y)} = KL(P(x,y), P(x)P(y))$$

Mutual information has an interpretation as a channel capacity.

Suppose that we draw a random bit $y \in \{0,1\}$ with $P(0) = P(1) = 1/2$ and send it across a noisy channel to a receiver who gets $y' = y \oplus \epsilon$ where ϵ is an independent "noise variable" with $\epsilon \in \{0,1\}$, where $\oplus$ is exclusive or (y gets flipped when $\epsilon = 1$), and where the "noise" ϵ has a probability P of being 1.

(a) Solve for the channel capacity $I(y,y')$ as a function of P in units of bits. When measured in bits, this channel capacity has units of bits received per message sent.

Solution:

$$\begin{aligned} I(y,y') &= H(y) - H(y|y') \\ H(y) &= 1 \text{ bit} \end{aligned}$$

$$\begin{aligned} H(y|y') &= P(y=y')(-\log_2 P(y=y')) + P(y \neq y')(-log_2 P(y \neq y')) \\ &= P(\epsilon = 0)(-\log_2 P(\epsilon = 0)) + P(\epsilon = 1) - log_2 P(\epsilon = 1) \\ &= (1-P)\log_2 1/(1-P) + P \log_2 1/P \\ &= H(P) \end{aligned}$$

(b) Explain why your answer to part (a) makes sense in terms of what the receiver knows for $P = 1/2$ and when $P = 1$.

Solution: For $P = 1/2$ we have $H(P) = 1$ bit and $I(y,y') = H(y) - H(P) = 0$ and the receiver knows nothing about y. For $P = 1$ we have $H(P) = 0$ and $I(y',y) = 1$ bit. Note that in this case y' is $1 - y$ so y' carries full information about y.

Problem 2. A Variational Upper Bound on Mutual Information

(a) Consider an arbitrary distribution $P(z, y)$. Show the variational equation

$$I(y, z) = \inf_Q \; E_{y \sim P(y)} KL(P(z|y), Q(z))$$

where Q ranges over distributions on z. Hint: It suffices to show

$$I(y, z) \leq \; E_y KL(P(z|y), Q(z))$$

and that there exists a Q achieving equality.

Solution:

$$\begin{aligned}
& I(y, z) \\
= \; & E_{y \sim \text{pop}} \; KL(P(z|y), P(z)) \\
= \; & E_{y, z \sim P(z|y)} \left(\ln \frac{P(z|y)}{Q(z)} + \ln \frac{Q(z)}{P(z)} \right) \\
= \; & E_{y \sim P(y)} \; KL(P(z|y), Q(z)) + \left(E_{y \sim \text{pop},\; z \sim P(z|y)} \; \ln \frac{Q(z)}{P(z)} \right) \\
= \; & E_y \; KL(P(z|y), Q(z)) + E_{z \sim P(z)} \; \ln \frac{Q(z)}{P(z)} \\
= \; & E_y \; KL(P(z|y), Q(z)) - KL(P(z), Q(z)) \\
\leq \; & E_{y \sim P(y)} \; KL(P(z|y), Q(z))
\end{aligned}$$

Equality is achieved when $Q(z) = P(z)$.

(b) Consider a rate-distortion autoencoder.

$$\Phi^* = \text{argmin} \; I_\Phi(y, z) + \lambda E_{y \sim \text{pop},\; z \sim P_\Phi(z|y)} \; \text{Dist}(y, y_\Phi(z)).$$

Here $I_\Phi(y, z)$ is defined by the distribution where we draw y from pop and z from $P_\Phi(z|y)$. We will write $P_{\text{pop}}(z)$ for the marginal on z under this distribution.

$$P_{\text{pop}}(z) = E_{y \sim \text{Pop}} \; P_\Phi(z|y)$$

Based on the result from part (b) rewrite the above definition of rate-distortion autoencoder to be a minimization over three independent models $P_\Phi(z)$ and $P_\Phi(y|z)$ and $P_\Phi(z|y)$ (although these models share parameters we will assume that Φ is sufficiently rich that the models are independently optimizable).

Solution:

$$\Phi^* = \underset{\Phi}{\text{argmin}}\, E_{y\sim \text{pop}, z\sim P_\Phi(z|y)}\ \ln \frac{P_\Phi(z|y)}{P_\Phi(z)} + \lambda\, \text{Dist}(y, y_\Phi(z)).$$

Problem 3. Modeling Rounding with Continuous Noise.

Consider a rate-distortion autoencoder with y and z continuous.

$$\Phi^* = \underset{\Phi,\Phi}{\text{argmin}}\ E_{y\sim \text{Pop}} KL(p_\Phi(z|y), p_\Phi(z)) + \lambda E_{y\sim \text{Pop},\ z\sim P(z|y)}\ \text{Dist}(y, y_\Phi(z)).$$

Define $p_\Phi(z|y)$ by $z = z_\Phi(y) + \epsilon$ with $z_\Phi[y] \in \mathbb{R}^d$ and ϵ drawn uniformly from $[0,1]^d$. In other words, we add noise drawn uniformly from $[0,1]$ to each component of $z_\Phi(y)$.

Define $p_\Phi(z)$ to be log-uniform in each dimension. More specifically $p_\Phi(z)$ is defined by drawing $s[i]$ uniformly from the interval $[0, s_{\max}]$ and then setting $z[i] = e^s$ so that $\ln z[i]$ is uniformly distributed over the interval $[0, s_{\max}]$. This gives

$$\begin{aligned} dz &= e^s ds &= z ds \\ dp &= \frac{1}{s_{\max}} ds \\ p_\Phi(z[i]) &= \frac{dp}{dz} &= \frac{1}{s_{\max} z[i]} \end{aligned}$$

Assume that we have that $z_\Phi(y) \in [1, e^{s_{\max}} - 1]^d$ so that with probability 1 over the draw of ϵ we have $\ln(z_\Phi(y) + \epsilon) \in [0, s_{\max}]$.

(a) For $z \in [z_\Phi(y), z_\Phi(y) + 1]$ what is $p_\Phi(z|y)$?

Solution: 1

(b) Solve for $KL(p_\Phi(z|y), p_\Phi(z))$ in terms of $z_\Phi(y)$ under the above specifications and simplify your answer for the case of $z_\Phi(y)[i] >> 1$.

Solution:

$$
\begin{aligned}
& KL(p_\Phi(z|y), p_\Phi(z)) \\
= \;& E_{z\sim P_\Phi(z|y)} \; \ln \frac{p_\Phi(z_\Phi(y))}{p_\Phi(z)} \\
= \;& E_{z\sim P_\Phi(z|y)} \; \sum_i \ln \frac{1}{1/(s_{\max} z[i])} \\
= \;& \sum_i E_{z[i]} \; \ln(s_{\max} z[i]) \\
= \;& \left(\sum_i \int_{z_\Phi(y)[i]}^{z_\Phi(y)[i]+1} \ln z \; dz \right) + d \ln s_{\max} \\
= \;& \left(\sum_i [z \ln z - z]_{z_\Phi(y)[i]}^{z_\Phi(y)[i]+1} \right) + d \ln s_{\max} \\
= \;& \left(\sum_i [z \ln z]_{z_\Phi(y)[i]}^{z_\Phi(y)[i]+1} \right) + d \ln s_{\max} - d \\
= \;& \left(\sum_i \ln(z_\Phi(y)[i]+1) + z_\Phi(y)[i](\ln(z_\Phi(y)[i]+1) - \ln z_\Phi(y)[i]) \right) + d \ln s_{\max} - d \\
= \;& \left(\sum_i \ln(z_\Phi(y)[i]+1) + z_\Phi(y)[i] \ln \left(1 + \frac{1}{z_\Phi(y)[i]} \right) \right) + d \ln s_{\max} - d \\
\approx \;& \left(\sum_i \ln z_\Phi(y)[i] \right) + d \ln s_{\max} - d \quad \text{for } z_\Phi(y)[i] >> 1
\end{aligned}
$$

Problem 4. Rounding RDA

We consider the following modification of RDAa

$$
\begin{aligned}
\text{RDA}: \; \Phi^* &= \operatorname*{argmin}_\Phi \; E_{y\sim \text{Pop},\, z\sim P_\Phi(z|y)} \; -\ln \frac{P_\Phi(z)}{P_\Phi(z|y)} + \lambda \text{Dist}(y, y_\Phi(z)) \\
\text{Rounding RDA}: \; \Phi^*, \Psi^* &= \operatorname*{argmin}_\Phi \; E_{y\sim \text{Pop},\, z:=\text{round}(z_\Psi(y))} \; -\ln P_\Phi(z) + \lambda \text{Dist}(y, y_\Phi(z))
\end{aligned}
$$

Here $\text{round}(z) \in \mathcal{Z}$ where $\mathcal{Z}$ is a discrete set of vectors defined independent of the choice of y. For example, rounding might map each real number in z to the nearest integer as was done in Balle et al. 2017. Or rounding might

map the vector z to the nearest center vector resulting from K-means vector quantization as in VQ-VAE. Other roundings are possible. The Rounding RDA corresponds to practical image compression where $-\log_2 P_\Phi(\text{round}(z_\Psi(y)))$ is (approximately) the number of bits in the compressed file.

(a) What is $\nabla_\Psi \ln P_\Phi(\text{round}(z_\Psi(y))$? **Solution**: zero

(b) What is $\nabla_\Psi \text{Dist}(y, y_\Phi(\text{round}(z_\Psi(y))))$? **Solution**: zero

To optimize Ψ Balle et al. used two tricks. They replaced $P_\Phi(\text{round}(z_\Psi(y)))$ with $p_\Phi(z_\Psi(y))$ where $p_\Phi(z)$ is a continuous density, and they replace the rounding operation with additive noise. Although rounding will be used for image compression, gradient descent is then done on

$$\Phi^*, \Psi^* = \underset{\Phi,\Psi}{\text{argmin}}\, E_{y,\epsilon}\ -\ln p_\Phi(z_\Psi(y)) + \lambda \text{Dist}(y_\Phi(z_\Psi(y) + \epsilon))$$

To model rounding to the nearest integer we take each dimension of ϵ to be drawn uniformly over the interval $(-1/2, 1/2)$.

(c) The density $p_\Phi(\tilde{z})$ defines a discrete distribution on the discrete values $\tilde{z} \in Z$ defined by

$$P_\Phi(\tilde{z}) = P_{z\sim p_\Phi}(\text{round}(z) = \tilde{z})$$

Consider the case where $\mathcal{Z}$ is the discrete set of vectors with integer coordinates. Assume that the density $p_\Phi(z)$ is locally approximated by its first order Taylor expansion

$$p_\Phi(z + \Delta z) = p_\Phi(z) + (\nabla_z p_\Phi(z))^\top \Delta z$$

Assuming the first order Taylor expansion is exact, give a closed-form expression for the discrete distribution $P_\Phi(\tilde{z})$ in terms of the continuous density $p_\Phi(z)$. Hint: write $P_\Phi(\tilde{z})$ as an expectation over ϵ drawn from the uniform distribution on $[-1/2, 1/2]^d$ where d is the dimension of z.

Solution: For an vector $\tilde{z}$ with integer coordinates we have

$$\begin{aligned}
P_\Phi(\tilde{z}) &= P_{z\sim p_\Phi}(\text{round}(z) = \tilde{z}) \\
&= \int_{\epsilon\in[-1/2,1/2]^d} p_\Phi(\tilde{z} + \epsilon)\ d\epsilon \\
&= E_{\epsilon\sim\text{uniform}[-1/2,1/2]^d}\ p_\Phi(\tilde{z} + \epsilon) \\
&= E_{\epsilon\sim\text{uniform}[-1/2,1/2]^d}\ p_\Phi(\tilde{z}) + (\nabla_{\tilde{z}} p_\Phi(\tilde{z}))^\top \epsilon \\
&= p_\Phi(\tilde{z}) + E_{\epsilon\sim\text{uniform}[-1/2,1/2]^d}\ (\nabla_{\tilde{z}} p_\Phi(\tilde{z}))^\top \epsilon \\
&= p_\Phi(\tilde{z}) + (\nabla_{\tilde{z}} p_\Phi(\tilde{z}))^\top E_{\epsilon\sim\text{uniform}[-1/2,1/2]^d}\ \epsilon \\
&= p_\Phi(\tilde{z})
\end{aligned}$$

Problem 5. VQ-VAEs

In a VQ-VAE the rounding operation is parameterized by a tensor $C[K, I]$ giving K center vectors of the form $C[k, I]$. We now consider rounding-RDAs defined by the following objective.

$$\Phi^*, \Psi^*, C^* = \underset{\Phi,\Psi,C}{\text{argmin}} \; E_{y\sim \text{Pop},\; \hat{L}:=\text{round}_C(L_\Psi(y))} \; -\ln P_\Phi(\hat{L}) + \lambda \text{Dist}(y, y_\Phi(\hat{L}))$$

In the VQ-VAE we are controlling the rate with the parameter K giving the number of clusters. In the optimization problem the prior term $P_\Phi(\hat{L})$ is being held as uniform over all $\hat{L}$ and can be ignored. Assuming L_2 distortion we are then left with

$$\Phi^*, \Psi^*, C^* = \underset{\Psi,\Psi,C}{\text{argmin}} \; E_y \frac{1}{2}||y - y_\Phi(\text{round}_C(L_\Psi(y)))||^2$$

This has well defined gradients for Φ and C but, because of rounding, not for Ψ. We are now trying to minimize the expected loss of the following forward calculation where $L[P, I]$ is a sequence of vectors.

$$\begin{aligned}
y &\sim \text{Pop} \\
L &= L_\Psi(y) \\
k[p] &= \underset{k}{\text{argmin}} \; ||C[k, I] - L[p, I]|| \\
\hat{L}[p, I] &= C[k[p], I] \\
\hat{y} &= y_\Phi(\hat{L}) \\
\text{Loss} &= \frac{1}{2}||y - \hat{y}||^2
\end{aligned}$$

The straight through gradient for a rounding operation is given by

$$L.\text{grad} \;\texttt{+=}\; \hat{L}.\text{grad}$$

(a) 10 points. Give a for loop for computing $C[K, I]$.grad from $\hat{L}$.grad as defined by backpropagation on the above computation.

Solution:

$$\text{for } p \quad C[k[p], I].\text{grad} \;\texttt{+=}\; \hat{L}[p, I].\text{grad}$$

(b) 15 points. The published formulation of VQ-VAE uses the following gradient updates.

$$\begin{aligned}
L.\text{grad} \;&\texttt{+=}\; \hat{L}.\text{grad} \\
L.\text{grad} \;&\texttt{+=}\; \beta(L - \hat{L}) \\
\text{for } p \;\; C[k[p], I].\text{grad} \;&\texttt{+=}\; \tilde{\eta}(C[k[p], I] - L[p, I])
\end{aligned}$$

Actually, this has been modified from the published form to add a learning rate adjustment parameter $\tilde{\eta}$.

Give an additional loss term so that the published version is equivalent to taking the gradient of $C[K, I]$.grad from the new loss term only and $L[P, I]$.grad from both the straight-through gradient and the gradient of the new loss term.

Solution: The additional loss is

$$\frac{1}{2}\beta||L[P, I] - \hat{L}[P, I]||^2 = \sum_p \frac{1}{2}\beta||L[p, I] - C[k[p], I]||^2$$

(c) 15 points. Give a complete set of backpropagation updates defined by backpropagation on both loss terms and using straight-through backpropagation to $L[P, I]$.grad

Solution:

$$\begin{array}{rrcl} & L.\text{grad} & \texttt{+=} & \hat{L}.\text{grad} \\ \text{for } p & C[k[p], I].\text{grad} & \texttt{+=} & \hat{L}[p, I].\text{grad} \\ & L.\text{grad} & \texttt{+=} & \beta(L - \hat{L}) \\ \text{for } p & C[k(t), I].\text{grad} & \texttt{+=} & \beta(C[k(t), I] - L[p, I]) \end{array}$$

Here any hyper-parameter for the learning rate for $C[K, I]$ must be handled elsewhere (in the optimizer).

(d) 10 points. We now have three versions of training — end-to-end with straight through as in part (a), the published version as in part (b), and the backpropagation on the both loss terms with straight-through as defined in part (c). For which of these three training algorithms is it true that at a stationary point $C[k, I]$ is mean of the vectors assigned to class k?

Solution: Of the three, this is only true for the published version.

Problem 6. This problem is on VAE language modeling. Consider a VAE where the signal s is a word string $w_1, \ldots, w_T$ (as in problem 2). In the VAE we can have a continuous latent variable z. The VAE optimization problem is then

$$\Phi^*, \Theta^*, \Psi^* = \underset{\Phi,\Theta,\Psi}{\text{argmin}} \; E_{s\sim \text{Pop},\; z\sim p_\Psi(z|s)} \; \ln \frac{p_\Psi(z|s)}{p_\Phi(z)} \; - \; \ln P_\Theta(s|z) \qquad (1)$$

Here the first "rate term" is defined on densities and the final "distortion term" is defined for a discrete sentence s. To explicitly handle the reparameterization trick will take the encoder density to be a Gaussian. For a Gaussian encoder we

compute a mean vector $\hat{z}_\Psi(s)$ and a variance $\sigma^2_\Psi(s)[i]$ for each component $z[i]$ of z. The Gaussian density for the encoder is then.

$$p_\Psi(z[i]|s) \propto \exp(-(z[i] - \hat{z}_\Psi(s)[i])^2/(2\sigma^2_\Psi(s)[i])$$

(a) For a noise value $\epsilon \in \mathbb{R}$ drawn from $\mathcal{N}(0,1)$, and for given values $\hat{z} \in \mathbb{R}$ and $\sigma^2 \in \mathbb{R}$, define a deterministic function $z(\hat{z}, \sigma^2, \epsilon)$ such that over the draw of the noise ϵ we have that $z(\hat{z}, \sigma^2, \epsilon)$ has the density

$$p(z) \propto \exp(-(z - \hat{z})^2/(2\sigma^2)).$$

Solution: $z(\hat{z}, \sigma^2, \epsilon) = \hat{z} + \sigma\epsilon$

(b) Applying your solution to part (a) to the individual components of z equation (6) can be rewritten as

$$\Phi^*, \Theta^*, \Psi^* = \underset{\Phi,\Theta,\Psi}{\text{argmin}} \; E_{s\sim \text{Pop},\, \epsilon\sim\mathcal{N}(0,I)} \; \ln \frac{p_\Psi(z|s)}{p_\Phi(z)} \; - \; \ln P_\Theta(s|z) \qquad (2)$$

Are there any problems with doing SGD on the optimization defined by (7) due to the use of continuous z and discrete s? Explain your answer.

Solution: There are no problems here. Since $P(s|z)$ is a computable and z is continuous we can compute z.grad which can then passed back to the encoder Ψ through the computation of $z(\hat{z}_\Psi(y), \Sigma_\Psi(y), \epsilon)$. We get a clear advantage of VAEs over GANs for s discrete.

(c) It can be shown that if we hold the encoder Ψ fixed then the optimal value of the prior density $p_\Phi(z)$ is just the marginal on z of the distribution defined by sampling $s \sim$ Pop and $z \sim p_\Psi(z|s)$. We can write this marginal as $p_{\text{Pop},\Psi}(z)$. Now consider the rate term when $p_\Phi(z) = p_{\text{Pop},\Psi}(z)$.

$$\text{rate} = E_{s\sim\text{Pop},\; z\sim P_\Psi(z|s)} \; \ln \frac{p_\Psi(z|s)}{p_{\text{Pop},\Psi}(z)}$$

Write this rate term as a differential mutual information.

Solution:

$$\text{rate} = I_{\text{Pop},\Psi}(s, z)$$

This has a channel capacity interpretation. It is the information capacity (information rate) of the communication channel that takes input y to output z. This is typically a nice finite number of bits (or nats) even for continuous densities. Adding noise to $\hat{z}_\Psi(y)$ intuitively limits its precision and limits the information that z carries about s.

Problem 7. This problem is on VAEs when both z and s are discrete. Is the discreteness of z an issue in this case? Explain your answer.

Solution: Yes, the discreteness of z is an issue. This is true independent of the nature of s. A differential change in parameters will not change a discrete z and $z.\text{grad} = 0$. So the standard back-propagation into the encoder fails. VQ-VAE back-propagates into the encoder using a K-means loss term together with straight-through gradients. Discreteness of s is not a problem.

Problem 8. Training Vector Quantization

Vector quantization (VQ) can be interpreted as introducing symbols. It uses an embedding matrix $E[K, I]$ giving an embedding vector $E[k, I]$ for each of K discrete "symbols". To make the notation more compact we will write $e(k)$ for the embedding vector $E[k, I]$ of the symbol k. We define the quantization operation to map a vector to the symbol whose embedding is nearest to that vector.

$$\text{nearest}_E(x) = \underset{k}{\text{argmin}} \ ||x - e(k)||$$

We consider a VQ-VAE where the latent variable is a single symbol (from a possibly large collection of K symbols). In this case the VQ-VAE optimizes the following objective.

$$\Phi^* = \underset{\Phi}{\text{argmin}} \ E_{y\sim\text{Pop}} \ ||y - y_\Phi(e(\text{nearest}_E(x_\Psi(y))))||^2 \tag{3}$$

$$\Psi^* = \underset{\Psi}{\text{argmin}} \ E_{y\sim\text{Pop}} \left\{ \begin{array}{ll} & ||y - y_\Phi(e(\text{nearest}_E(x_\Psi(y))))||^2 \\ + & \beta||x_\Psi(y) - e(\text{nearest}_E(x_\Psi(y)))||^2 \end{array} \right. \tag{4}$$

$$E^* = \underset{E}{\text{argmin}} \ E_{y\sim\text{Pop}} \ ||x_\Psi(y) - e(\text{nearest}_E(x_\Psi(y)))||^2 \tag{5}$$

I have written this as a separate objective function for each component of the model. The objective for a component defines a gradient for that component. Multiple simultaneous objectives define a multi-player game. We hope to reach a Nash equilibrium where this is defined as a parameter setting where all the objectives have zero gradients — each "player" is doing a locally best (or at least stationary) response. Multiple objectives can be implemented by putting stop gradients (detachments) in each objective to prevent the optimization of one component from affecting the other components.

The objective (4) defines the gradient for Ψ. In VQ-VAE we compute a gradient for (4) using the "straight-through" gradient for back-propagation through vector quantization. The VQ straight-through gradient can be written as

$$\nabla_x f(e(\text{nearest}_E(x))) \approx \nabla_e f(e)|_{e=e(\text{nearest}_E(x))}$$

(a) Give an `+=` equation for incorporating $e(\text{nearest}_E\ x).\text{grad}$ into $x.\text{grad}$.

Solution:

$$x.\text{grad} \texttt{ += } e(\text{nearest}_E(x)).\text{grad}.$$

(b) Write the SGD update equation for gradient descent on (5) using learning rate η.

Solution:

$$e(\text{nearest}_E(x_\Psi(y))) \texttt{ += } 2\eta(x_\Psi(y) - e(\text{nearest}_E(x_\Psi(y))))$$

(c) Assuming $\eta < 1/2$, rewrite your solution to (b) in the form of a rolling average update on $e(k)$ showing that $e(k)$ is a rolling average of the vectors of the form $x_\Psi(y)$ satisfying $\text{nearest}_E(x_\Psi(y)) = k$.

Solution: For $\text{nearest}_E(x_\Psi(y)) = k$ we have

$$e(k) = (1 - 2\eta)e(k) + 2\eta x_\Psi(y)$$

Problem 3. 25 pts This problem is on VAE language modeling (in contrast to GAN language modeling). Consider a VAE where the signal s is a word string $w_1, \ldots, w_T$ (as in problem 2). In the VAE we can have a continuous latent variable z. The VAE optimization problem is then

$$\Phi^*, \Theta^*, \Psi^* = \underset{\Phi,\Theta,\Psi}{\text{argmin}} \; E_{s\sim \text{Pop},\, z\sim p_\Psi(z|s)} \; \ln \frac{p_\Psi(z|s)}{p_\Phi(z)} \; - \; \ln P_\Theta(s|z) \qquad (6)$$

Here the first "rate term" is defined on densities and the final "distortion term" is defined for a discrete sentence s. To explicitly handle the reparameterization trick will take the encoder density to be a Gaussian. For a Gaussian encoder we compute a mean vector $\hat{z}_\Psi(s)$ and a variance $\sigma^2_\Psi(s)[i]$ for each component $z[i]$ of z. The Gaussian density for the encoder is then.

$$p_\Psi(z[i]|s) \propto \exp(-(z[i] - \hat{z}_\Psi(s)[i])^2/(2\sigma^2_\Psi(s)[i])$$

(a) For a noise value $\epsilon \in \mathbb{R}$ drawn from $\mathcal{N}(0,1)$, and for given values $\hat{z} \in \mathbb{R}$ and $\sigma^2 \in \mathbb{R}$, define a deterministic function $z(\hat{z}, \sigma^2, \epsilon)$ such that over the draw of the noise ϵ we have that $z(\hat{z}, \sigma^2, \epsilon)$ has the density

$$p(z) \propto \exp(-(z - \hat{z})^2/(2\sigma^2)).$$

Solution: $z(\hat{z}, \sigma^2, \epsilon) = \hat{z} + \sigma\epsilon$

(b) Applying your solution to part (a) to the individual components of z equation (6) can be rewritten as

$$\Phi^*, \Theta^*, \Psi^* = \underset{\Phi,\Theta,\Psi}{\operatorname{argmin}} \; E_{s\sim \mathrm{Pop},\, \epsilon\sim\mathcal{N}(0,I)} \; \ln \frac{p_\Psi(z|s)}{p_\Phi(z)} \; - \; \ln P_\Theta(s|z) \tag{7}$$

Are there any problems with doing SGD on the optimization defined by (7) due to the use of continuous z and discrete s? Explain your answer.

Solution: There are no problems here. Since $P(s|z)$ is a computable and z is continuous we can compute z.grad which can then passed back to the encoder Ψ through the computation of $z(\hat{z}_\Psi(y), \Sigma_\Psi(y), \epsilon)$. We get a clear advantage of VAEs over GANs for s discrete.

(c) It can be shown that if we hold the encoder Ψ fixed then the optimal value of the prior density $p_\Phi(z)$ is just the marginal on z of the distribution defined by sampling $s \sim \mathrm{Pop}$ and $z \sim p_\Psi(z|s)$. We can write this marginal as $p_{\mathrm{Pop},\Psi}(z)$. Now consider the rate term when $p_\Phi(z) = p_{\mathrm{Pop},\Psi}(z)$.

$$\mathrm{rate} = E_{s\sim \mathrm{Pop},\; z\sim P_\Psi(z|s)} \; \ln \frac{p_\Psi(z|s)}{p_{\mathrm{Pop},\Psi}(z)}$$

Write this rate term as a differential mutual information.

Solution:

$$\mathrm{rate} = I_{\mathrm{Pop},\Psi}(s, z)$$

This has a channel capacity interpretation. It is the information capacity (information rate) of the communication channel that takes input y to output z. This is typically a nice finite number of bits (or nats) even for continuous densities. Adding noise to $\hat{z}_\Psi(y)$ intuitively limits its precision and limits the information that z carries about s.

Problem 4. 25 pts This problem is on VAEs when both z and s are discrete. Is the discreteness of z an issue in this case? Explain your answer.

Solution: Yes, the discreteness of z is an issue. This is true independent of the nature of s. A differential change in parameters will not change a discrete z and z.grad $= 0$. So the standard back-propagation into the encoder fails. VQ-VAE back-propagates into the encoder using a K-means loss term together with straight-through gradients. Discreteness of s is not a problem.

A variational autoencoder (VAE) has three parts, an encoder enc, a decoder dec and a prior pri. The VAE is defined by the following objective.

$$\mathrm{enc}^*, \mathrm{dec}^*, \mathrm{pri}^* = \mathrm{argmin}_{\mathrm{enc},\mathrm{dec},\mathrm{pri}} \; \mathrm{ELBO}(z, y)$$

$$\text{ELBO}(y,z) = E_{y\sim \text{pop}, z\sim P_{\text{enc}}(z|y)}\left[-\ln \frac{P_{\text{pri}}(z)P_{\text{dec}}(y|z)}{P_{\text{enc}}(z|y)}\right]$$

Recall that the cross entropy $H(P,Q)$ for a sampling distribution P and a model distribution Q is defined by

$$H(P,Q) = E_{w\sim P}\left[-\ln Q(w)\right]$$

A conditional entropy $H(w|u)$ is defined for a joint distribution P(w,u) by

$$H(w|u) = E_{(w,u)\sim P}\left[-\ln P(w|u)\right]$$

Part 1: Write ELBO(y,z) as a cross entropy (on a joint distribution of y and z) minus a conditional entropy.

Solution:

$$\text{ELBO}(y,z) = H(\text{pop}(y)P_{\text{enc}}(z|y),\ P_{\text{pri}}(z)P_{\text{dec}}(y|z)) - H(P_{\text{enc}}(z|y))$$

Here $H(P_{\text{enc}}(z|y))$ refers to the joint distribution on y and z defined by $\text{Pop}(y)P_{\text{enc}}(z|y)$.

Part 2: Suppose we fix the encoder arbitrarily and optimize only the prior and the decoder. What is the joint distribution on y and z defined by the optimal models $P_{\text{pri}^*}(z)$ and $P_{\text{dec}^*}(y|z)$? Justify your answer.

Solution: The optimal models for the prior and the decoder define the same joint distribution as that defined by the population and then encoder. This is because the optimal solution to a cross entropy is the sampling distribution.

Part 3. Suppose that we leave the encoder fixed arbitrarily and train just the prior and the decoder, as in part 2, and then sample y (perhaps an image) by sampling z from the prior and y from the decoder. Assuming universality for the prior and decoder, but with the encoder untrained, will this result in sampling y from the population distribution? Explain your answer.

Solution:

Yes, this will sample from the population because the joint distribution defined by the prior and the decoder will match the joint distribution defined by the population and encoder.

Problem

This is a problem on vector quantization. Vector quantization converts vectors to tokens. Converting vectors to tokens allows images and sounds to be modeled with transformers.

Consider $k \in \{1, 2, \ldots, K\}$. I will call k a token.

For each token $k \in \{1, 2, \ldots, K\}$ let $e(k) \in R^d$ be a vector embedding of the token k.

This is the same formal set-up as having vector embeddings for language tokens (words).

Let e be the embedding matrix (the matrix $e(k)[i]$ giving the ith component of the vector $e(k)$).

Let $\hat{k}$ be a tokenization function mapping any $x \in R^d$ to a token $\hat{k}(x)$.

We want to optimize the embedding matrix e and the tokenization function $\hat{k}$ so that $e(\hat{k}(x))$ is near x.

More formally, let ρ be a probability density on R^d. We want the following.

$$e^*, \hat{k}^* = \text{argmin}_{e,\hat{k}} \; E_{x\sim\rho} \; ||e(k(x)) - x||^2$$

(a) Write a gradient descent update equation for optimizing the embedding matrix e for a fixed embedding function $\hat{k}$.

(b) Write a equation defining an optimal embedding function $\hat{k}$ for a fixed embedding matrix e.

Comments:

Loyd's Algorithm for k-means vector quantization replaces (a) with a closed form solution for $\hat{k}$ given e and alternates the optimization of $\hat{k}$ and e.

In vector quantization for deep learning models we also have an embedding model emb and ab decoding model dec and want to solve the optimization problem.

$$e^*, \hat{k}^*, \text{emb}^*, \text{dec}^* = \text{argmin}_{e,\hat{k},\text{emb},\text{dec}} \; E_{x\sim\rho} \; ||\text{dec}(e(k(\text{emb}(x)))) - x||^2$$

Actually it is a little more complicated that this equation because the embedding and the decoding operate on the whole image or sound wave and produce a matrix or array of vectors which are then tokenized individually.