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:

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

Problem 3. 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.

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

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

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

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.

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

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 $C[K, I].\text{grad}$ from the new loss term only and $L[P, I].\text{grad}$ 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 $L[P, I].\text{grad}$

Solution:

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

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 4. 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

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

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 (**??**) can be rewritten as

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

Are there any problems with doing SGD on the optimization defined by (**??**) 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 $s \sim \text{Pop}$ 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 6. 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.

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

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

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

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

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

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

Solution:

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

(b) Write the SGD update equation for gradient descent on (**??**) 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

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