Differential Entropy, VAEs

Submit a PDF of your answers to Canvas

1. Recall the entropy of a discrete random variable is defined as

$$H(X) = E_X\left[\log \frac{1}{p(X)}\right].$$

For continuous random variables, this quantities is referred to as the *differential* entropy, often denoted as $h(X)$, and computed as

$$h(X) = \int_{-\infty}^{\infty} p(x) \log \frac{1}{p(x)}\, dx.$$

The definition extends to continuous random vectors:

$$h(\boldsymbol{x}) = \int_{\mathbb{R}^n} p(\boldsymbol{x}) \log \frac{1}{p(\boldsymbol{x})}\, d\boldsymbol{x}.$$

a) Let $\boldsymbol{x}$ be a random vector, $\boldsymbol{\mu}$ be a constant vector, and define $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\mu}$. Find an expression for the differential entropy of $\boldsymbol{y}$ in terms of $h(\boldsymbol{x})$.

b) Let $\boldsymbol{x} \sim \mathscr{N}(0, \boldsymbol{\Sigma})$. Compute the differential entropy of $\boldsymbol{x}$. Use log base e. *Hint:* It is helpful to use the fact that $\mathrm{Tr}(\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}) = \mathrm{Tr}(\Sigma^{-1} \boldsymbol{x}\boldsymbol{x}^T)$, which is often called the trace trick.

c) What is the differential entropy of $\boldsymbol{x} \sim \mathscr{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$?

2. Implement a variational autoencoder on the MNIST dataset to synthesize handwritten digits. You may find a number of tutorials helpful (for example, `https://keras.io/examples/generative/vae/` or `https://www.tensorflow.org/tutorials/generative/cvae`). You may use a simple feed-forward architecture, or a convolutional network structure. Make sure to cite any resources you use. Display a number of randomly generated MNIST images and write a short description of the implementation.