



Homework 9  
Due 5pm, Wednesday, May 20, 2024

**Problem 1: Anomaly detection via AE.** In this problem, you will use an autoencoder to perform anomaly detection between the MNIST and the Kuzushiji(崩し字)-MNIST (KMNIST) [1] datasets. KMNIST contains handwritten Japanese characters. Download the starter code `anomaly_detection.py` and implement the following steps. In step 1, load the MNIST and KMNIST datasets, and split the MNIST training dataset into “training” and “validation” sets. (Together with the “test” set you will have three datasets in total.) In step 2, define the AE model. In step 3, instantiate the model and select the Adam optimizer. In step 4, train the AE with the training data  $X_1, \dots, X_N$  with loss

$$\ell(\theta, \varphi) = \sum_{i=1}^N \|X_i - D_\varphi(E_\theta(X_i))\|^2,$$

where  $E_\theta$  is the encoder and  $D_\varphi$  is the decoder. Do not use the validation set in this stage. In step 5, define the score function

$$s(X) = \|X - D_\varphi(E_\theta(X))\|^2$$

and calculate the mean and standard deviation of

$$\{s(Y_i)\}_{i=1}^M$$

where  $Y_1, \dots, Y_M$  are the validation data. Define a threshold to be mean + 3 standard deviations, and define inputs with score function value exceeding this threshold to be anomalies. In step 6, check how many of the MNIST images within the test set are classified as anomalies and report the type I error rate. In step 7, check how many of the KMNIST images are classified as non-anomalies and report the type II error rate.



Figure 1: KMNIST images

## References

- [1] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha, Deep learning for classical Japanese literature, *NeurIPS ML for Creativity Workshop*, 2018.

**Problem 2: 1D flow to Gaussian.** Consider the flow

$$f_\theta(x) = \sum_{i=1}^n e^{w_i} (\Phi_{\mu_i, \exp(\tau_i)}(x) - 0.5),$$

where  $\theta = (w_1, \dots, w_n, \mu_1, \dots, \mu_n, \tau_1, \dots, \tau_n)$  and

$$\Phi_{\mu, \sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x \exp\left(-\frac{1}{2} \left(\frac{s-\mu}{\sigma}\right)^2\right) ds.$$

Note that  $f_\theta: \mathbb{R} \rightarrow \mathbb{R}$ . Download the starter code `normalizingFlow1d.py` and fit the flow model with  $n = 5$  and  $p_Z \sim \mathcal{N}(0, 1)$ .

*Remark.* Since  $p_Z$  is an unbounded distribution, we do not require  $w_1, \dots, w_n$  to be normalized.

**Problem 3: Affine coupling layer with permutations.** Consider the affine coupling layer defined as follows. Let  $\Omega \subseteq \{1, \dots, n\}$  and  $0 < |\Omega| < n$ . Define  $\Omega^c = \{1, \dots, n\} \setminus \Omega$ . For  $x \in \mathbb{R}^n$ , define

$$x_\Omega \in \mathbb{R}^{|\Omega|}, \quad x_{\Omega^c} \in \mathbb{R}^{n-|\Omega|}$$

to be the sub-vectors of  $x$  with the indices within  $\Omega$  and  $\Omega^c$  selected. Define  $z_\Omega$  and  $z_{\Omega^c}$  analogously for  $z \in \mathbb{R}^n$ . The affine coupling layer is

$$\begin{aligned} z_\Omega &= x_\Omega \\ z_{\Omega^c} &= e^{s_\theta(x_\Omega)} \odot x_{\Omega^c} + t_\theta(x_\Omega), \end{aligned}$$

where  $s_\theta: \mathbb{R}^{|\Omega|} \rightarrow \mathbb{R}^{n-|\Omega|}$  and  $t_\theta: \mathbb{R}^{|\Omega|} \rightarrow \mathbb{R}^{n-|\Omega|}$ . Show that

$$\log \left| \frac{\partial z}{\partial x} \right| = \mathbf{1}_{n-|\Omega|}^\top s_\theta(x_\Omega).$$

*Clarification.* We are not assuming  $|\Omega| = n/2$ .

*Hint.* Find a permutation  $\sigma$  such that

$$\frac{\partial z}{\partial x} = P_{\sigma^{-1}} \begin{bmatrix} I & 0 \\ * & \text{diag}(e^{s_\theta(x_\Omega)}) \end{bmatrix} P_\sigma.$$

**Problem 4:  $D_{\text{KL}}$  of continuous random variables.** The KL-divergence between continuous random variables  $X \sim f$  and  $Y \sim g$ , where  $f$  and  $g$  are probability density functions in  $\mathbb{R}^d$ , is

$$D_{\text{KL}}(X \| Y) = \int_{\mathbb{R}^d} f(x) \log \left( \frac{f(x)}{g(x)} \right) dx.$$

(a) Show that

$$D_{\text{KL}}(X \| Y) \geq 0.$$

(b) Show that if  $X = (X_1, \dots, X_d)$  is a continuous random variable such that  $X_1, \dots, X_d$  are independent and  $Y = (Y_1, \dots, Y_d)$  is a continuous random variable such that  $Y_1, \dots, Y_d$  are independent, then

$$D_{\text{KL}}(X \| Y) = D_{\text{KL}}(X_1 \| Y_1) + \dots + D_{\text{KL}}(X_d \| Y_d).$$

**Problem 5:**  $D_{\text{KL}}$  of Gaussian random variables. Let  $\mathcal{N}(\mu, \Sigma)$  denote the Gaussian distribution with mean  $\mu$  and covariance  $\Sigma$ . So if  $X \sim \mathcal{N}(\mu, \Sigma)$ , then

$$\mathbb{E}[X] = \mu, \quad \mathbb{E}[(X - \mu)(X - \mu)^\top] = \Sigma.$$

Show that

$$D_{\text{KL}}(\mathcal{N}(\mu_0, \Sigma_0) \parallel \mathcal{N}(\mu_1, \Sigma_1)) = \frac{1}{2} \left( \text{tr}(\Sigma_1^{-1} \Sigma_0) + (\mu_1 - \mu_0)^\top \Sigma_1^{-1} (\mu_1 - \mu_0) - d + \log \left( \frac{\det \Sigma_1}{\det \Sigma_0} \right) \right),$$

where  $d$  is the underlying dimension of the random variables  $\mathcal{N}(\mu_0, \Sigma_0)$  and  $\mathcal{N}(\mu_1, \Sigma_1)$ . Assume  $\Sigma_0$  and  $\Sigma_1$  are positive definite.

**Problem 6:** *When maximizing a lower bound is tight.* Consider the optimization problem

$$\underset{\theta \in \Theta}{\text{maximize}} \quad f(\theta).$$

Informally assume  $f$  is an intractable function, i.e., evaluating  $f(\theta)$  is difficult. However, assume there exists a decomposition

$$f(\theta) = g(\theta, \phi) + h(\theta, \phi) \quad \forall \phi \in \Phi,$$

where  $g$  is tractable, i.e., evaluating  $g(\theta, \phi)$  is easy,  $h(\theta, \phi) \geq 0$  for all  $\theta \in \Theta$  and  $\phi \in \Phi$ , and for any  $\theta \in \Theta$  there exists a  $\phi \in \Phi$  such that  $h(\theta, \phi) = 0$ , i.e., for any  $\theta \in \Theta$ ,

$$\min_{\phi \in \Phi} h(\theta, \phi) = 0$$

and the minimum is attained. Now we consider the following problem with the tractable objective function

$$\underset{\theta \in \Theta, \phi \in \Phi}{\text{maximize}} \quad g(\theta, \phi).$$

Show that the two optimization problems are equivalent in the sense that

$$\text{argmax } f = \{\theta \mid (\theta, \phi) \in \text{argmax } g\}.$$

*Hint.* Use the fact that

$$\sup_{\theta, \phi} g(\theta, \phi) = \sup_{\theta} \left( \sup_{\phi} g(\theta, \phi) \right).$$

*Remark.* Training variational autoencoders involves maximizing the variational lower bound (VLB/ELBO). If the encoder network is infinitely expressive (if the encoder network can represent any function), maximizing the VLB is equivalent to maximizing the log-likelihood. This problem abstracts the explanation of why that is the case.