Past Paper

NC-Exam

Question 1

Let $x^{(1)}, ..., x^{(n)}$ be vectors in $\mathbb{R}^d$. Consider the minimization of the following function

$$C(w) = \frac{1}{2n}\sum_{i=1}^{n}||w - x^{(i)}||_2^2$$

where $\|\cdot\|_2$ is the Euclidean norm, i.e., $||w||_2^2 = \sum_{j=1}^{d} w_j^2$ for $w = (w_1, ..., w_d)^T \in \mathbb{R}^d$

1. What is the global minimiser of C? Give your arguments.
    - The gradient of the objective function is
    $$\nabla C(w) = \frac{1}{n}\sum_{i=1}^{n}(w - x^{(i)}) = w - \frac{1}{n}\sum_{i=1}^{n}x^{(i)}$$
    - According to the first-order optimality condition, we have the minimiser is
    $$w^* = \frac{1}{n}\sum_{i=1}^{n}x^{(i)}$$
2. Suppose we apply stochastic gradient descent to this problem with $w^{(0)} = (0, 0, ..., 0)^T$, step size $\eta_t = \frac{1}{t+1}$ and $x^{(t+1)}$ to compute a stochastic gradient. Compute $w^{(1)}, w^{(2)}, w^{(3)}$. Based on these computation, write a general formula for $w^{(k)}$, $k \leq n$. Give your arguments to explain this general formula.
    - We define $C_i(w) = \frac{1}{2}||w - x^{(i)}||_2^2$. The gradient of C_i is:
    $$\nabla C_i(w) = w - x^{(i)}$$
    - For SGD, we have
    $$\begin{aligned} w^{(1)} &= w^{(0)} - w^{(0)} + x^{(1)} = x^{(1)} \\ w^{(2)} &= w^{(1)} - \frac{1}{2}(w^{(1)} - x^{(2)}) = \frac{1}{2}x^{(1)} + \frac{1}{2}x^{(2)} \\ w^{(3)} &= w^{(2)} - \frac{1}{3}(w^{(2)} - x^{(3)}) = \frac{1}{3}x^{(1)} + \frac{1}{3}x^{(2)} + \frac{1}{3}x^{(3)} \end{aligned}$$
    - Generalizing from here, we can conclude by induction that
    $$w^{(t)} = \frac{1}{t}\sum_{i=1}^{t}x^{(i)}, 0 < t \leq n$$
    - Indeed:
    $$\begin{aligned} w^{(t)} &= w^{(t-1)} - \frac{1}{t}\nabla C_t(w^{(t-1)}) = w^{(t-1)} - \frac{1}{t}(w^{(t-1)} - x^{(t)}) \\ &= (\frac{t-1}{t})w^{(t-1)} + \frac{1}{t}x^{(t)} = \frac{t-1}{t}\frac{1}{t-1}\sum_{i=1}^{t-1}x^{(i)} + \frac{1}{t}x^{(t)} \\ &= \frac{1}{t}\sum_{i=1}^{t}x^{(i)} \end{aligned}$$
    where in the last second indentity we use the induction hypothesis (1). Then $w^{(n)} = \frac{1}{n}\sum_{i=1}^{n}x^{(i)}$
3. Suppose we apply gradient descent to train a neural network. In which step of gradient descent do we use backpropagation?
    - We use backpropagation to compute gradient in the gradient descent algorithm.

Question 2

We have the following questions related to convolutional neural networks(CNNs). Please answer them with justification

1. Given the convolution kernel $W \in R^{3x3}$ and matrix $A \in \mathbb{R}^{5x5}$, we perform the following convolution with padding = 0 and stride = 1:
$$W \circledast A = B$$

where ⊛ represents a 2D convolution, and A and W are respectively given as follows:

$$A = \{a_{ij}\} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 2 & 3 & 4 \\ 4 & 5 & 1 & 2 & 3 \\ 3 & 4 & 5 & 1 & 2 \\ 2 & 3 & 4 & 5 & 1 \end{bmatrix} and\ W = \{w_{ij}\} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Please fill in the missing entries of the matrix $B \in \mathbb{R}^{3x3}$ below:

$$B = \{b_{ij}\} = \begin{bmatrix} \dots & \dots & 0 \\ \dots & \dots & \dots \\ 5 & \dots & \dots \end{bmatrix}$$

Note that you need to write down B on your answer sheet

- The key point to answer this question is to how to do convolutions with stride 1 and zero padding. In this case, it will result in a reduced matrix size of 3x3. Hence:

$$B = \begin{bmatrix} -15 & -5 & 0 \\ 15 & -15 & -5 \\ 5 & 15 & -15 \end{bmatrix}$$

2. Next, following the question (a) above, during convolution each element of A will be multiplied by some element of the convolution kernel $W \in R^{3x3}$ a number of times. Fill in the following matrix with the number of times each element is multiplied by an element of the weight matrix W:

$$\begin{bmatrix} 1 & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 9 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

Note that you need to write down W on your answer sheet

- This question has not been taught over lectures. However, as long as the student understands how convolutions work, it is straightforward to answer this question. For each entry of A, simply counts the number of times the kernel has occurred on that pixel. As such, the matrix is given as follows, which is a centrosymmetric matrix.

$$\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$$

3. We then calculate the following loss function f, which involves B computed in the question (a) above and some matrix $C \in \mathbb{R}^{3x3}$:

$$f = \sum_{i=1}^{3} (b_{ii} - c_{ii})^2$$

where c_{ij} denotes the entries of matrix C which are defined as follows:

$$C = \{c_{ij}\} = \begin{bmatrix} -10 & 0 & 0 \\ 0 & -10 & 0 \\ 0 & 0 & -10 \end{bmatrix}$$

As is the case with neural networks, the weights of a CNN will be updated using backpropagation which relies on the chain rule to calculate the derivates of the loss function with respect to the weights. Use the chain rule to calculate and evaluate the derivatives $\frac{\partial f}{\partial w_{11}}$ and $\frac{\partial f}{\partial a_{11}}$, where w_{11} and a_{11} are the first entry of W and A respectively. Note that W and A are given in the question (a).

- This question is about the chain rules, which we have covered in the first few lectures. To answer this question correctly, the student should understand how convolutions work as well as how to derive a derivative when it comes to convolutions. We have designed the question carefully by using same coefficients so the derivations should have a pattern and hence is easier
- 这个问题是关于链式法则的，我们在前几节课中已经讲过了。为了正确回答这个问题，学生应该理解卷积的工作原理，以及如何在卷积中导出导数。我们精心设计了这个问题，使用相同的系数，因此导出应该有一个模式，因此更容易

1. 对 $\frac{\partial f}{\partial w_{11}}$ 的计算：

$$\frac{\partial f}{\partial w_{11}} = \sum_{i=1}^{3} \frac{\partial f}{\partial B_{ii}} \cdot \frac{\partial B_{ii}}{\partial w_{11}} = 2 \times \sum_{i=1}^{3} (B_{ii} + 10) \times 1 = -30$$

2. 对 $\frac{\partial f}{\partial a_{11}}$ 的计算：

$$\frac{\partial f}{\partial a_{11}} = \frac{\partial f}{\partial B_{11}} \cdot \frac{\partial B_{11}}{\partial a_{11}} = 2 \times (B_{11} + 10) \times (-1) = 10$$

Question 3

1. An auto-encoder(AE) consists of an encoding unit f_φ, a latent representation z, and a decoding uniy g_θ. The goal of an auto-encode is to learn to produce output $\hat{x} = g_\theta(z) = g_\theta(f_\varphi(x))$ for a given input x, such that $\hat{x} = x$ and where $z = f_\varphi(x)$
    1. During the model training process, the auto-encoder model can naively learn the identity function making it a useless model. Describe briefly why we consider such a model as a useless model and how we make the auto-encoder learn a useful latent representation z instead of an identity function
        - The model is considered useless as it is learning $z = f_\varphi(x)$ and $\hat{x} = g_\theta(Z)$. This does not enable the model to learn any latent representation which can be useful for dimensionality reduction, compression, clustering, or semi-supervised learning.
            - 该模型被视为无用，因为它正在学习 $z = f_\varphi(x)$ 和 $\hat{x} = g_\theta(Z)$。这不允许模型学习任何可用于降维、压缩、聚类或半监督学习的潜在表示。
        - The solution to this problem is the introduction of a layer between the encoder f_φ and the decoder g_θ as a 'bottleneck' layer, such that it forces the latent representation z by having its dimensionality much smaller than the dimensionality of the input x.
            - 解决此问题的方法是在编码器 f_φ 和解码器 g_θ 之间引入一个层作为"瓶颈"层，这样它就强制潜在表示 z 的维数远小于输入 x 的维数。
    2. Given the trained auto-encoder consisting of f_φ, g_θ and z, consider that it has the ability for good self-reconstruction. Is this auto-encoder suitable for generating synthetic(or new) data? Justify your answer with brief reasoning.
        - The AE is not trained for generation since it is trained primarily for self-reconstruction. In the latent representation z space, there are gaps as the model tries to keep similar data mapped to 'clusters' while keeping gap between dissimilar data to achieve good self-reconstruction ability. This gap, however, does not allow it to learn to generate new data.
            - AE 未接受生成训练，因为它主要接受自我重建训练。在潜在表示 z 空间中，存在差距，因为模型试图将相似的数据映射到"聚类"，同时保持不同数据之间的差距以实现良好的自我重建能力。然而，这种差距不允许它学习生成新数据。
2. A variational auto-encoder(VAE) relies on the following loss function which consists of two terms:

    $$L_{VAE} = L_{rec} + L_{reg}$$

    where L_{rec} represents the reconstruction loss and L_{reg} represents the regularisation loss. Briefly describe about what happens if we exclude the L_{reg} term from the VAE loss function such that $L_{VAE} = L_{rec}$ to train the VAE
    - If we minimize only the reconstruction loss, the encoder in VAE can learn to predict means that are arbitrarily far.
        - 如果我们仅最小化重建损失，VAE 中的编码器就可以学会预测任意远的均值。
    - They can take values outside the area covered by the 'prior'. In such case, the 'ideal' standard deviations for reconstruction loss are zero for every x.
        - 它们可以取"先验"覆盖区域之外的值。在这种情况下，对于每个 x，重建损失的"理想"标准差为零。
3. A generative adversarial network (GAN) consists of two units: a generator G_θ to generate fake data samples and a discriminator D_φ to recognise whether a sample is real or fake. We need to design a good loss function to train the GAN discriminator unit for learning its parameters φ. We have designed the below loss function for discriminator learning:

    $$\min_{\varphi} E_{x \sim p_{data}(x)} \left[-\log(1 - D_\varphi(x)) \right] + E_{z \sim p(z)} [-\log(D_\varphi(G_\theta(z)))]$$

    where E denotes the expectation operator, x $p_{data}(x)$ denotes input sample x drawn from real data distribution $p_{data}(x)$, and $z \sim p(z)$ denotes GAN generated data z drawn from fake data distribution p(z).
    Briefly describe what this loss function is doing and how it can be changed to help in GAN discriminator training.
    - The current loss function assigns a high loss value $-\log(1 - D_\varphi(x))$ when the sample x is real and a low loss value $-\log(D_\varphi(G_\theta(z)))$ when the sample z is fake. This is opposite of what it should be doing.
        - 当前损失函数在样本 x 为真时分配较高的损失值 $-\log(1 - D_\varphi(x))$，而在样本 z 为假时分配较低的损失值 $-\log(D_\varphi(G_\theta(z)))$。这与它应该做的事情正好相反。
    - To correct it, we need to change this loss function to the following form to make it work for discriminator learning:

        $$\min_{\varphi} E_{x \sim p_{data}(x)} \left[-\log(D_\varphi(x)) \right] + E_{z \sim p(z)} [-\log(1 - D_\varphi(G_\theta(z)))]$$

Mock-Exam

Question 1

Consider a set of two input-output pairs $(1, 1)$ and $(-1, -1)$. Let the loss function be the least square and we wish to find a linear model $x \mapsto x^T w$, where x^T is the transpose of $x \in \mathbb{R}^2$ and $w = (w_1, w_2) \in \mathbb{R}^2$. Then the objective function becomes

$$C(w) = \frac{1}{2}\left(\frac{1}{2}(1,2)w - 1\right)^2 + \frac{1}{2}\left(\frac{1}{2}(-1,2)w + 1\right)^2.$$

Let us build our prediction model by minimizing the above objective function.

1. Simplify the objective function to the form of

   $$C(w) = c_1 w_1^2 + c_2 w_2^2 + c_3 w_1 + c_4 w_2 + c_5 w_1 w_2 + c_6,$$

   where $c_k \in \mathbb{R}$ are coefficients, and w_i is the i-th coordinate of $w \in \mathbb{R}^2$. After that, compute the gradient of $C(w)$ in terms of c_k.

   - It is clear that

     $$4C(w) = w_1^2 + 4w_1w_2 + 4w_2^2 - 2w_1 - 4w_2 + 1 + w_1^2 + 4w_1w_2 + 4w_2^2 + 2w_1 - 4w_2 + 1 = 2w_1^2 + 8w_2^2 - 4w_1 + 8w_2 + 2$$

     Therefore

     $$C(w) = \frac{1}{2}w_1^2 + 2w_2^2 - w_1 + \frac{1}{2} \; and \; \begin{pmatrix} w_1 - 1 \\ 4w_2 \end{pmatrix}$$

2. Consider gradient descent with the initial point $w^{(0)} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$ and learning rate $\eta = 0.5$. Write down the process of calculating $w^{(1)}$ and $w^{(2)}$. After that, calculate $C(w^{(1)})$ and $C(w^{(2)})$.

   - For the gradient descent, we know

     $$w^{(1)} = w^{(0)} - 0.5\nabla C(w^{(0)}) = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

     so $C(w^{(1)}) = 5/2$

     $$w^{(2)} = w^{(1)} - 0.5\nabla C(w^{(1)}) = \begin{pmatrix} 1.5 \\ 1 \end{pmatrix}$$

     so $C(w^{(2)}) = 17/8$

3. Consider gradient descent with momentum. Assume $w^{(0)} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$, learning rate $\eta = 0.5$ and momentum rate $\alpha = 0.5$. Write down the process of calculating $w^{(1)}$ and $w^{(2)}$. After that, calculate $C(w^{(1)})$ and $C(w^{(2)})$.

   - For the gradient descent with momentum, we know

     $$v^{(1)} = \alpha v^{(0)} - 0.5\nabla C(w^{(0)}) = -\frac{1}{2}\begin{pmatrix} 2 \\ 4 \end{pmatrix}$$
     $$w^{(1)} = w^{(0)} + v^{(1)} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

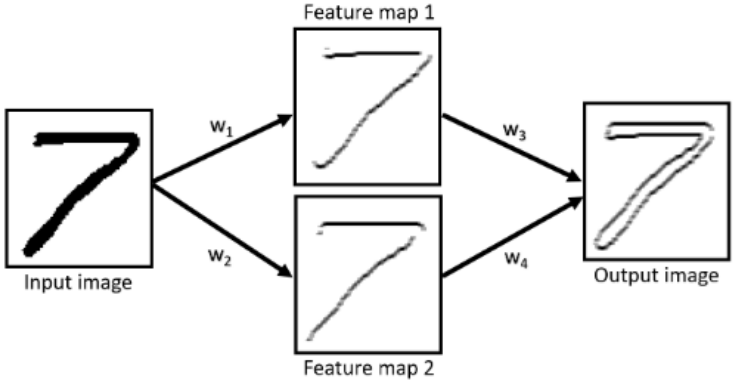
     so $C(w^{(1)}) = 5/2$

     $$v^{(2)} = \alpha v^{(1)} - 0.5\nabla C(w^{(1)}) = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$
     $$w^{(1)} = w^{(0)} + v^{(1)} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

     so $C(w^{(2)}) = 0$

Question 2

Answer the following questions related to convolutional neural networks (CNNs):



1. A CNN is shown in the figure below. We use the nonlinear ReLU activation function in the first layer and the linear activation function in the output layer. Note that for better visualisation, in the images we use white regions to denote 0 and darker regions to denote larger values.

    1. Design appropriate convolution kernels of size 3 × 3 for the first layer such that the feature maps 1 and 2 are these displayed in the figure. Please justify your answer.

$$W_1 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

        - One of kernels should detect dark/light horizontal boundaries, while the other should detect light/dark horizontal boundaries. It does not matter which one is W_1 or W_2. One kernel should have a positive gradient in the up-down direction while another kernel should have a negative gradient in the up-down direction.
            - 其中一个内核应检测暗/亮水平边界，而另一个内核应检测亮/暗水平边界。哪一个是 W_1 或 W_2 并不重要。一个内核应在上下方向上具有正梯度，而另一个内核应在上下方向上具有负梯度。

    2. Design appropriate convolution kernels of size 3 × 3 for the output layer such that the output is that displayed in the figure. Please justify your answer.

$$W_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad W_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

        Two kernels that add the feature maps from the previous layer.

2. We apply N convolutional kernels with stride=1 and padding = 0 to a 11 by 11 colour image, which results in a 5 by 5 feature map, with 10 channels. We then apply M convolutional kernels of size (H, W, D) to this feature map with stride = 2 and padding = 1. This results in a 4 by 4 output, with 5 channels. Please identify the values for (N, M, H, W, D). Please justify your answer.

> 第一步：第一层卷积
>
> - **输入图像尺寸**：11 x 11的彩色图像。
> - **输出特征图**：5 x 5，共10个通道。
> - **步长（Stride）**：1
> - **填充（Padding）**：0
>
> 由于没有特别提到卷积核的尺寸，我们可以通过卷积输出尺寸的公式来推断卷积核的尺寸：
>
> $$\text{输出尺寸} = \left(\frac{\text{输入尺寸} - \text{核尺寸} + 2 \times \text{填充}}{\text{步长}} \right) + 1$$
>
> 插入已知的数值，求解卷积核尺寸：
>
> $$5 = \left(\frac{11 - \text{核尺寸} + 2 \times 0}{1} \right) + 1$$
>
> $$\text{核尺寸} = 7$$
>
> 这意味着每个卷积核的尺寸为 7x7，卷积核的数量 (N) 为10（对应输出特征图的10个通道）。
>
> 第二步：第二层卷积
>
> - **输入特征图**：5 x 5，10个通道。
> - **输出**：4 x 4，5个通道。
> - **步长（Stride）**：2
> - **填充（Padding）**：1
>
> 使用上述同样的公式来确定第二层卷积的参数：
>
> $$4 = \left(\frac{5 - \text{核尺寸} + 2 \times 1}{2} \right) + 1$$
>
> $$4 = \left(\frac{5 - \text{核尺寸} + 2}{2} \right) + 1$$
>
> $$\text{核尺寸} = 1$$
>
> 因此，第二层的卷积核尺寸 (H2 = 1, W2 = 1)，核深度 (D2) 需要与输入特征图的通道数相同，即10。第二层卷积核的数量 (M) 为5，对应输出特征图的5个通道。
>
> 结论 根据这个推理过程，我们可以确认：
>
> - (N = 10) （第一层的卷积核数量）
> - (M = 5) （第二层的卷积核数量）

- (H2 = 1, W2 = 1) （第二层卷积核的高度和宽度）
- (D2 = 10) （第二层卷积核的深度）

3. In CNNs, apart from ReLU there exist other nonlinear activation functions such as Sigmoid and Tanh. For the collection of neurons in a single layer, what would be the Jacobian matrices of the Sigmoid and Tanh functions, respectively. The Jacobian matrix is defined as:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Where $f_i(i = 1, .., m)$ denote the nonlinear activate function (Sigmoid or Tanh) and $x_j(j = 1, .., n)$ the neurons (variables) fed to that nonlinearity. Note: you can simplely use Sigmoid' and Tanh' to denote the derivatives of Sigmoid and Tanh, respectively.

- Both Jacobian matrices should be a diagonal matrix. For Sigmoid, the diagonal entries are Sigmoid'(xi), where xi denote input neurons. For tanh, the diagonal entries are tanh'(xi).

Question 3

1. Consider the standard Variational Auto-Encoder (VAE), with encoder $f_\phi(x)$ parameterized by ϕ that predicts mean $\mu_\phi(x)$ and standard deviation $\sigma_\phi(x)$ of a multidimensional Gaussian that is the conditional (posterior) $p_\phi(z|x) = N(\mu_\phi(x), \sigma^2_\phi(x))$, and a decoder $g_\theta(z)$ parameterized by θ.
Assume that half-way through VAE training we process input x1 and get encoder outputs $\mu_\phi(x_1) = (0.1, 0.2)$, and $\sigma_\phi(x_1) = (0.05, 0.2)$. Using the re-parameterization trick, we sample code ˜z to give as input to the decoder. What value of ˜z is most likely to give the best reconstruction of x1? Explain why.
   - The mean μ_ϕ predicted by the encoder.
     - 编码器预测的均值μ_ϕ。
   - Explanation 1: According to the posterior $p_\phi(z|x)$ by encoder, the mean is the most probably code for input x, therefore will give the reconstruction that best matches x.
     - 解释 1：根据编码器的后验$p_\phi(z|x)$，均值是输入 x 最可能的代码，因此将给出与 x 最匹配的重构。
   - Explanation 2: The mean is the z that will be sampled the most during training when performing a forward pass on x, and therefore the value that will be most commonly reconstructed as x.
     - 解释 2：均值是在对 x 进行前向传递时训练期间采样最多的 z，因此是最常重构为 x 的值。
2. Consider a Generative Adversarial Network (GAN) that consists of Generator G that takes as input noise vector z, and of a Discriminator D that given input x it outputs D(x). We assume that value D(x) = 1 means that D predicts with certainty that input x is a real data point, and D(x) = 0 means D predicts with certainty that x is a fake, generated sample.
   1. Assume that at the beginning of training, parameters of G and D are initialized randomly. Then, D is trained for few SGD iterations, while G remains fixed (untrained). After the few updates to D's parameters, is the value D(G(z)) likely to be closer to 0 or 1? Explain why.
      - The value is likely to be closer to 0. This is because G with random weights will produce terrible images. D will easily learn to separate bad fakes from real examples, even with a few SGD iterations, therefore predicting D(G(z)) ≈ 0.
        - 该值可能更接近 0。这是因为具有随机权重的 G 会产生糟糕的图像。即使只进行几次 SGD 迭代，D 也可以轻松学会将劣质假货与真实示例区分开来，因此预测 D(G(z)) ≈ 0。
   2. After the whole training process of the GAN has finished, assume that G has been optimized ideally. What would be the most likely value for D(G(z))? Explain why.
      - The ideal value would be D(G(z)) = 0.5. Explanation 1: Ideally, G has learned to generated data that are are perfectly realistic. Then D cannot distinguish between real and fake data and its accuracy is chance, 50%. Explanations 2: It has been theoretically proven that the optimal discriminator predicts the ratio pdata(x)/(pdata(x) + pmodel (x)). If the two distributions are the same for optimal G, the ratio is 0.5.
        - 理想值应该是 D(G(z)) = 0.5。解释 1：理想情况下，G 已经学会生成完全真实的数据。那么 D 就无法区分真实数据和虚假数据，其准确率是偶然的 50%。解释 2：理论上已经证明，最佳判别器预测比率 pdata(x)/(pdata(x) + pmodel (x))。如果最佳 G 的两个分布相同，则比率为 0.5。
3. Assume you are a Machine Learning Engineer. You are given a large database of photos of objects (all from same data distribution). You wish to create an object classifier based on neural-networks. The object class is labelled only on a few of the images. Assume the number of unlabelled data is high (no possible overfit). You are instructed to train an unsupervised model on the unlabelled data, and afterwards use its trained parameters to initialize a classifier, which you can then refine with supervised learning on the few labelled images. You can choose between a basic Auto-Encoder (AE), a Variational Auto-Encoder (VAE) and a Generative Adversar- ial Network (GAN) (basic versions taught). What model would you choose? Explain why the other two are suboptimal.
   - The AE is the most optimal.
     - AE 是最优的。
   - The classifier benefits by pre-trained parameters that tend to cluster the data. Clustering is a result of the reconstruction loss, which is optimized by AE.

        - 分类器受益于倾向于对数据进行聚类的预训练参数。聚类是重构损失的结果，由 AE 进行优化。
    - The VAE additionally minimizes a Regularizer, which opposes the reconstruction loss, therefore is less ideal.
        - VAE 还最小化了正则化器，它与重构损失相反，因此不太理想。
    - The basic GAN cannot be used, because it does not have an encoder to learn mapping x → z where z a potentially useful (e.g. clustered) representation of the data.
        - 基本 GAN 无法使用，因为它没有编码器来学习映射 x → z，其中 z 是数据的潜在有用（例如聚类）表示。