1. In SL, what happens when you get a training error ε = 0?

   Training error is zero, it means the predicted function h(x) is as the same as the target function. But for the test of the model, different data will be used. Then the prediction made by the model will not be optimal. Because the model can only generate a reliable function h(x) by using exactly the same data in the training set. --> overfitting

   The model generated with many parameters may has small bias and large variance.

2. When you a training a SL model, how do you decide when to stop training?

   When we get a good prediction from the model, i.e. when the error between the target function and the predicted function h(x) is significantly small, we can decide to stop training.

   When J(theta) gets close (reach) the minimum, we can decide to stop training.

3. In linear regression, what happens when you add more terms to your model?

   The model will suffer from larger variance but smaller bias.

   The curve fits the data perfectly, will not reflect the wider pattern of the relationship between x and y. (week 2 chapter6 Folie bias vs variance)

4. What values are given to the parameters vector θ initially? and why?

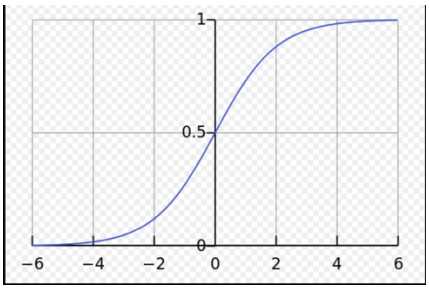
   The parameters vector theta is initially random number. Because they are going to be modified during the training process.

5. What is the difference between regression and logistic regression?

   Linear regression uses the general linear equation $Y = b_0 + \sum (b_i X_i) + \epsilon$, where Y is a continuous dependent variable and independent variables X_i are usually continuous.

   Logistic regression is another generalized linear model procedure using the same basic formula, but instead of the continuous Y, its regressing for the probability of a categorical outcome. In simplest form, this means the output could either be 0 or 1. (classification with discrete Y)

6. What is the shape of the sigmoid function and why is it used in logistic regression?



The sigmoid function is bounded between 0 and 1, and its derivative is easy to calculate.

The main reason to use a sigmoid function is because it is a simple way of introducing non-linearity to the model.

7. Why is useful dimension reduction in machine learning?

   If there are many features, but not all the features are relevant for the model prediction.

   We can reduce the features (or the dimensions) to generate a model.

   PCA (principle component analysis)

   It reduces the time and storage space required

   Removal of multi-collinearity improves the performance of the machine learning model.

   It becomes easier to visualize the data when reduced to very low dimensions such as 2D or 3D

8. Which are the 2 main steps of the k-means algorithm?

   1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
   2. Repeat until convergence: {

      For every i, set

      $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

      For each j, set

      $$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}.$$

      }

9. How does PCA work?

   Principle component analysis

   Reduce dimensions without loss of many information

   The steps for applying PCA are

   1. Get some data.
   2. Subtract the mean.
   3. Calculate the covariance matrix.
   4. Calculate the eigenvectors and eigenvalues of the covariance matrix.
   5. Choosing components and forming a feature vector.
   6. Deriving the new data set.

10. What is the general EM algorithm?

Expectation Maximization

Repeat until convergence {

1. (E-step) For each i, j set

$$w_j^{(i)} := p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma)$$

2. (M-step) Update the parameters

$$\begin{aligned} \phi_j &= \frac{1}{m} \sum_{i=1}^{m} w_j^{(i)}, \\ \mu_j &= \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}, \\ \Sigma_j &= \frac{\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^\top}{\sum_{i=1}^{m} w_j^{(i)}}. \end{aligned}$$

}

11. What are the parameters estimated in the GDA algorithm?

$$\begin{aligned} \phi &= \frac{1}{m} \sum_{i=1}^{m} 1\{y^{(i)} = 1\} \\ \mu_0 &= \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}} \\ \mu_1 &= \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}} \\ \Sigma &= \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^\top \end{aligned}$$

12. In linear regression with Gaussian basis functions, what changes you need to make in your algorithm when going from one dimension to multiple dimensions?

    Multiple dimensions: Mean vector, covariance matric (the relationships between the features)

    One dimension: mean value and variance value.

    Instead of having only one x, in the multiple dimensions we have several different x's. For determine the mean values and variance of the Gaussian distributions: if the values are not in the same range, then they should be normalized to the values between 0 and 1. And the basis function also changes.

$$\phi_j(\mathrm{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_\mathbf{j})^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_\mathbf{j}) \right)$$

13. Why is it useful the distortion function $J(c,\mu) = \sum_{i=1}^{m} ¿ \vee x^{(i)} - \mu_{c^{(i)}} \vee ¿^2$?

    J measures the sum of squared distance between each training example $x^{(i)}$ and the cluster centroid $\mu_{c(i)}$ to which it has been assigned. The distortion function J is a non-convex function, and so coordinate descent on J is not guaranteed to converge to the global minimum. In other words, k-means can be susceptible to local optima. We can only find the local minimum because of the non-convex function.

14. How is the Return defined in reinforcement learning?

    Episodic tasks:

$$R_t = r_{t+1} + r_{t+2} + \cdots + r_T$$

    r_{t+1} is the reward received after the t-th transition.

    Continuing tasks:

Discounted return:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1},$$

where $\gamma, 0 \le \gamma \le 1$, is the **discount rate**.

15. What are the main difference between dynamic programming methods and temporal difference methods?

    - DP solves for the optima policy or value function by recursion.

- DP requires a model of the environment.
- DP samples
  - TD does not require a model of the environment
  - TP is iterative
  - TP does not sample

16. What is the main difference between Sarsa and Q-learning?

Sarsa updates the Q(s,a)-value with random action, which is selected according to the ε-greedy.

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma Q(s',a') - Q(s,a)\right]$$

Q-learning updates the Q(s,a)-value with the action gives the greatest reward value.

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)\right]$$

17. Give the Bellman equation for V^{π} (Give equation)

Bellman equation for V^{π} :

$$V^{\pi}(s) = \sum_a \pi(s,a) \sum_{s'} P^a_{ss'}\left[R^a_{ss'} + \gamma V^{\pi}(s')\right]$$

18. What is $V^{¿}$? (Give equation)

Optima state-value function

$$V^*(s) = \max_{\pi} V^{\pi}(s) \quad \text{for all } s \in S$$

$$V^{\pi}(s) = \sum_a \pi(s,a) \sum_{s'} P^a_{ss'}\left[R^a_{ss'} + \gamma V^{\pi}(s')\right]$$

19. What is $\pi^{¿}$? (Give equation)

Optima policy

$$\pi^*(s) = \arg\max_{a \in A(s)} Q^*(s,a)$$

20. How do you apply reinforcement learning methods when you have an infinite number of states?

    Apply supervised learning method: linear regression

    Represent state as feature vectors:

    for each $s \in S$:

    $$\vec{\phi}_s = \left(\phi_s(1), \phi_s(2), \ldots, \phi_s(n)\right)^T$$

    $$V_t(s) = \vec{\theta}_t^T \vec{\phi}_s = \sum_{i=1}^{n} \theta_t(i)\phi_s(i)$$