Part 1

- 2. Print the hyperparameters you found :
  - Params:
    - c = 3, gamma = 0.002 (K = 5)
  - Code :
    - Reference
      - In the guide, it says that since doing a complete grid-search may still be time-consuming, it's recommended to use a coarse grid first. After identifying a “better” region on the grid, a finer grid search on that region can be conducted. Therefore, I conducted 2 grid search in my code.
    - Grid search function

```python
def grid_search(kfold_data, cs, gammas):
    cov = np.zeros((len(cs), len(gammas)))
    max_acc = 0.0
    best = {'c': -1, 'g': -1}
    for i, c in enumerate(cs):
        for j, g in enumerate(gammas):
            # Taking average on the validation accuracy
            total_err = 0.0
            for kf in kfold_data:
                train_idx, val_idx = kf[0], kf[1]
                clf = SVC(C=c, kernel='rbf', gamma=g)
                clf.fit(x_train[train_idx], y_train[train_idx])
                y_pred = clf.predict(x_train[val_idx])
                total_err += sum(np.logical_xor(y_train[val_idx], y_pred))
            avg_acc = 1.0 - (total_err / len(y_train))
            if(avg_acc > max_acc):
                max_acc = avg_acc
                best = {'c': c, 'g': g}
            # save in cov-matrix
            cov[i, j] = avg_acc
    print(f"c : {best['c']}, gamma :{best['g']}, max_acc : {max_acc}")
    return cov, best
```

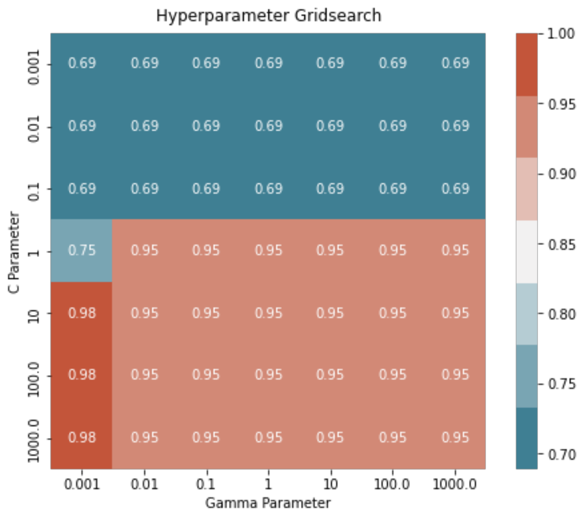
- Do the grid search twice on {c1, gamma1} and {c2, gamma2} respectively.
  - np.round is used to fix the floating point problem.
  - Avg validation accuracy = 0.981

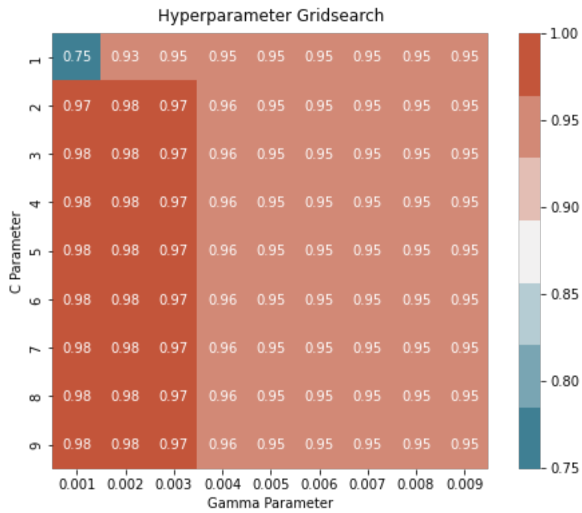
```python
kfold_data = cross_validation(x_train, y_train, k=5)
# Coarse search
c1 = [1e-3, 1e-2, 1e-1, 1, 10, 1e2, 1e3]
gamma1 = [1e-3, 1e-2, 1e-1, 1, 10, 1e2, 1e3]
cov1, best1 = grid_search(kfold_data, c1, gamma1)

# Finer search
c2 = np.arange(1, 10, 1)
gamma2 = np.round(np.arange(0.001, 0.01, 0.001), 3)
cov2, best2 = grid_search(kfold_data, c2, gamma2)
pep8(_ih)
```

```
c : 10, gamma :0.001, max_acc : 0.98
c : 3, gamma :0.002, max_acc : 0.9818181818181818
```

- 3. Plot the grid search results of your SVM.
  - Plot :
    - Coarse grid search / Finer grid search





  - Code :
    - Heatmap function

```python
import seaborn as sns
import matplotlib.pyplot as plt


def heatmap_sns(c, gamma, cov):

    plt.figure(figsize=(8, 6))
    ax = sns.heatmap(
        cov,
        vmin=np.min(cov), vmax=1.0, annot=True,
        xticklabels=gamma,
        yticklabels=c,
        cmap=sns.diverging_palette(220, 20, n=7),
        square=True
    )
    ax.set_title('Hyperparameter Gridsearch', fontdict={'fontsize': 12}, pad=9)
    ax.set_xlabel("Gamma Parameter")
    ax.set_ylabel("C Parameter")
```

    - Plot 2 maps

```python
# Coarse grid search
heatmap_sns(c1, gamma1, cov1)
```

```python
# Fine grid search
heatmap_sns(c2, gamma2, cov2)
```

- 4. Train your model by {C: 3, gamma: 0.002} on the whole training data and evaluate the performance on the test set.
  - Result
    - Test acc = 0.916, Val acc = 0.981 (see Q2)

```python
# Train the model by {C: 3, gamma: 0.002} on the whole training data
best_model = SVC(C=best2['c'], kernel='rbf', gamma=best2['g'])
best_model.fit(x_train, y_train)

# Testing accuracy
y_pred = best_model.predict(x_test)
print("Accuracy score: ", accuracy_score(y_pred, y_test))
pep8(_ih)
```

```
Accuracy score:  0.9166666666666666
```

Part 2

Let's take a look at some properties of kernal first.
Given valid kernals $k_j(x,x')$ for $j \in \mathbb{N}$, the following kernals will also be valid.

① $K(x,x') = \alpha k_1(x,x')$ for $\alpha \geq 0$

Pf: For gram matrix $K = \alpha K_1$

$\forall u \in \mathbb{R}^n,\ u^T K u = \alpha \cdot u^T K_1 u \geq 0$

② $K(x,x') = K_1(x,x') + \alpha$ for $\alpha \geq 0$

Pf: Let ϕ_1 denote the feature map of K_1.

Then, using the feature map $\phi: x \mapsto [\phi_1(x), \sqrt{\alpha}]^T$,

we have: $\langle \phi(x), \phi(x') \rangle = \langle \phi_1(x), \phi_1(x') \rangle + \alpha = k_1(x,x') + \alpha = k(x,x')$

③ $k(x,x') = \sum_{j=1}^{m} \alpha_j k_j(x,x')$, $\alpha_j \geq 0$

Pf: For gram matrix $K = \sum_{j=1}^{m} \alpha_j K_j$

$\forall u \in \mathbb{R}^n,\ u^T K u = \sum_{j=1}^{m} \alpha_j\, u^T K_j u \geq 0$

due to the positivity of α_j and the validity of the kernals K_j

④ $k(x,x') = k_1(x,x')\, k_2(x,x')$

Pf: By construction, the gram matrix is given by

$K = K_1 \odot K_2$, where $\odot$ denotes the Hadamard (entrywise) product.

Given that K_1 and K_2 are symmetric positive semi-definite matrices,

their eigendecompositions $K_1 = \sum_{i=1}^{n} \lambda_i u_i u_i^T$ and $K_2 = \sum_{j=1}^{n} \mu_j v_j v_j^T$ have

positive eigenvalues $\lambda_i \geq 0$ and $\mu_j \geq 0$. This leads to:

$$K = \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \mu_j (u_i u_i^T) \odot (v_j v_j^T)$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \mu_j (u_i \odot v_j)(u_i \odot v_j)^T$$

$$= \sum_{k=1}^{n^2} \gamma_k \omega_k \omega_k^T, \text{ where } \gamma_k = \lambda_{\lfloor \frac{k}{n} \rfloor} \mu_{k \bmod n} \geq 0$$

$$\text{and } \omega_k = u_{\lfloor \frac{k}{n} \rfloor} \odot v_{k \bmod n}$$

Thus, $\forall u \in \mathbb{R}^n,\ u^T K u = \sum_{k=1}^{n^2} \gamma_k u^T \omega_k \omega_k^T u = \sum_{k=1}^{n^2} \gamma_k (\omega_k^T u)^2 \geq 0$

⑤ $k(x,x') = q(k_1(x,x'))$, $q(\cdot)$ is a polynomial with nonnegative coefficients.

Pf: Suppose $q(x) = a_n x^n + a_{n-1} x^{n-1} + \dots a_1 x^1 + a_0$, we now obtain

$k(x,x') = a_n (k_1(x,x'))^n + a_{n-1}(k_1(x,x'))^{n-1} + \dots a_1 k_1(x,x') + a_0$

By repeatedly using rule $\overset{\alpha \cdot k}{①}$, $\overset{\Sigma \alpha k}{③}$, $\overset{k \cdot k}{④}$, we can verify $k(x,x')$ a valid kernal.

⑥ $k(x,x') = \exp(k_1(x,x'))$

Pf: According to Taylor expansion, $k(x,x') = \sum_{n=0}^{\infty} \frac{1}{n!} (k_1(x,x'))^n$

By rule ⑤, we can prove $k(x,x')$ a valid kernal.

⑦ $k(x,x') = f(x)\, k_1(x,x')\, f(x')$

Pf: Since $k_1(x,x')$ is a valid kernal, it can be written as

$k_1(x,x') = \phi_1(x)^T \phi_1(x')$

We can obtain $k(x,x') = f(x) k_1(x,x') f(x')$

$= [f(x)\phi_1(x)]^T [f(x')\phi_1(x')]$

1.

(a) $k(x,x') = (k_1(x,x')^2) + (k_1(x,x')+1)^2$

(i) $(k_1(x,x'))^2 = k_1(x,x')k_1(x,x')$, is a valid kernal by rule $\overset{k\cdot k}{④}$

(ii) $(k_1(x,x')+1)$ is a valid kernal by rule $\overset{k+\alpha}{②}$

$\Rightarrow (k_1(x,x')+1)^2$ is a valid kernal by rule $\overset{k\cdot k}{④}$

(iii) $(k_1(x,x'))^2 + (k_1(x,x')+1)^2$ is a valid kernal by rule $\overset{\Sigma\alpha k}{③}$ $\blacksquare$

(b) $k(x,x') = (k_1(x,x'))^2 + \exp(\|x\|^2) * \exp(\|x'\|^2)$

(i) $(k_1(x,x'))^2$ is a valid kernal by rule $\overset{k\cdot k}{④}$

(ii) $\exp(\|x\|^2) + \exp(\|x'\|^2)$ is a valid kernal

Pf: Using a explicit feature map $\phi: x \mapsto f(x) = e^{\|x\|^2}$, we have

$e^{\|x\|^2} * e^{\|x'\|^2} = f(x)f(x') = \phi(x)^T\phi(x')$, which is a valid kernal.

(iii) $(k_1(x,x'))^2 + [\exp(\|x\|^2)\exp(\|x'\|^2)]$ is a valid kernal by rule $\overset{\Sigma\alpha k}{③}$ $\blacksquare$

2.

① If K is a positive semi-definite matrix, we have

$K = V\Lambda V^T$ where V is an orthonormal matrix with eigenvector v_t, and Λ is a diagonal matrix contains all eigenvalues $\lambda_t \geq 0$

Consider the feature map $\phi: x_n \longmapsto (\sqrt{\lambda_t}\, v_{tn})_{n=1}^{N} \in R^N$

we find that:

$\phi(x_n)^T\phi(x_m) = \sum_{t=1}^{d} \lambda_t v_{tn} v_{tm} = (V\Lambda V^T)_{nm} = K_{nm} = k(x_n, x_m)$

② By definition, we define the gram matrix $K = \Phi^T\Phi$, which is an N by N symmetric matrix with elements $K_{nm} = \phi(x_n)^T\phi(x_m) = k(x_n, x_m)$

Let $q(x) = x^T K x = x^T\Phi^T\Phi x = (\Phi x)^T(\Phi x) = \|\Phi x\|^2 \geq 0$ for $\forall x \in \mathbb{R}^N$.

Thus, K is positive semi-definite.

Therefore, K is positive semi-definite $\Leftrightarrow$ $k(x_n, x_m)$ is a valid kernal. $\blacksquare$

3.

From setting $\frac{\delta}{\delta w} J(w)$ to zero, we have

$$w = \frac{-1}{\lambda} \sum_{n=1}^{N} \{ w^T \phi(x_n) - t_n \} \phi(x_n) = \sum_{n=1}^{N} a_n \phi(x_n) = \Phi^T a, \quad \text{— ①}$$

where $\Phi = \begin{bmatrix} \vdots \\ \phi(x_n) \\ \vdots \end{bmatrix}^T$, $a = (a_1, \dots a_N)^T$ with $a_n = \frac{-1}{\lambda}\{ w^T \phi(x_n) - t_n \}$

$$a_n = \frac{-1}{\lambda} \{ w^T \phi(x_n) - t_n \}$$

$$= \frac{-1}{\lambda} \{ w_1 \phi_1(x_n) + w_2 \phi_2(x_n) + \dots w_M \phi_M(x_n) - t_n \}$$

$$= \frac{-w_1}{\lambda} \phi_1(x_n) - \frac{w_2}{\lambda} \phi_2(x_n) - \dots \frac{w_M}{\lambda} \phi_M(x_n) + \frac{t_n}{\lambda}$$

$$= (c_n - \frac{w_1}{\lambda}) \phi_1(x_n) + (c_n - \frac{w_2}{\lambda}) \phi_2(x_n) + \dots (c_n - \frac{w_M}{\lambda}) \phi_M(x_n) \quad \text{— ②} \quad \blacksquare$$

where $c_n = \frac{t_n}{\lambda} \cdot \frac{1}{\phi_1(x_n) + \phi_2(x_n) + \dots \phi_M(x_n)}$

Now, we know that a_n is a linear combination of $\phi(x_n)$

Therefore, for dual reprensentation $J(a)$, we have :

$$J(a) = \frac{1}{2} a^T K K a - a^T K t + \frac{1}{2} t^T t + \frac{\lambda}{2} a^T K a$$

$$= \frac{1}{2} a^T \Phi \Phi^T \Phi \Phi^T a - a^T \Phi \Phi^T t + \frac{1}{2} t^T t + \frac{\lambda}{2} a^T \Phi \Phi^T a, \quad K = \Phi \Phi^T$$

$$= \frac{1}{2} w^T \Phi \Phi^T w - w \Phi^T t + \frac{1}{2} t^T t + \frac{\lambda}{2} w^T w \quad \text{by}$$

$$= J(w)$$

4.

① $\|x - x'\|^2 = x^T x + x'^T x' - 2 x^T x'$

② $k(x, x') = \exp(-\|x - x'\|^2 / 2\sigma^2)$

$$= \exp\left(\frac{-1}{2\sigma^2} x^T x\right) \exp(2 x^T x') \exp\left(\frac{-1}{2\sigma^2} x'^T x'\right)$$

$$= \exp\left(\frac{-x^T x}{2\sigma^2}\right) \sum_{i=0}^{\infty} \frac{(2 x^T x')^i}{i!} \exp\left(\frac{-x'^T x'}{2\sigma^2}\right)$$

$$= \sum_{i=0}^{\infty} \underline{\exp\left(\frac{-x^2}{2\sigma^2}\right) \sqrt{\frac{2^i}{i!}} (x^i)^T} \; \underline{\sqrt{\frac{2^i}{i!}} (x'^i) \exp\left(\frac{-x'^2}{2\sigma^2}\right)}, \quad x \in \mathbb{R}^1$$

$$= \phi(x)^T \phi(x'), \quad \text{where } \phi(x) = \exp\left(\frac{-x^2}{2\sigma^2}\right) \begin{bmatrix} 1 \\ \sqrt{\frac{2}{1!}} x \\ \sqrt{\frac{2^2}{2!}} x^2 \\ \vdots \end{bmatrix} \quad \blacksquare$$

Alternative way to show that $k(x, x')$ is valid :

$$K(x, x') = \exp\left(\frac{-x^2}{2\sigma^2}\right) \exp(2 x^T x') \exp\left(\frac{-x'^2}{2\sigma^2}\right)$$

(i) $x^T x'$ is a linear kernal $\Rightarrow 2x^T x'$ is a valid kernal by rule ① αk

(ii) $\exp(2 x^T x')$ is a valid kernal by rule ⑥ e^k

(iii) $f(x) \exp(2 x^T x') f(x')$, $f(x) = \exp\left(\frac{-x^2}{2\sigma^2}\right)$ is a valid kernal by rule ⑦

5.

$$L(x, \lambda) = (x-2)^2 + \lambda[(x+3)(x-1)-2]$$

$$= (1+\lambda)x^2 + (2\lambda-4)x + (4-5\lambda)$$

Set $\nabla L_x(x,\lambda) = 2x(1+\lambda) + (2\lambda-4) = 0$

$$\Rightarrow x = \frac{2-\lambda}{(1+\lambda)}$$

$$L(x(\lambda), \lambda) = (\lambda-2)^2/(1+\lambda) + (2\lambda-4)(2-\lambda)/(1+\lambda) + (4-5\lambda)$$

$$= -(\lambda-2)^2/(1+\lambda) + (4-5\lambda)$$

$$= \begin{cases} \dfrac{-(\lambda-2)^2}{(1+\lambda)} + (4-5\lambda), & \lambda > -1 \\ -\infty, & \lambda \le -1 \end{cases}$$

Thus, the dual problem is :

maximize $-(\lambda-2)^2/(1+\lambda) + (4-5\lambda)$

subject to $\lambda \ge 0$ $\blacksquare$

Check :

① The optimal value p^* to the original problem is

constraint: $x^2+2x-5 \le 0 \Rightarrow -1-\sqrt{6} \le x \le -1+\sqrt{6}$

Thus, $p^* = (-1+\sqrt{6}-2)^2 = (-3+\sqrt{6})^2 = 15-6\sqrt{6}$

② The optimal value p^* to the dual problem is :

(i) $L = -(\lambda-2)^2/(\lambda+1) + (4-5\lambda)$

$$\frac{\delta L}{\delta \lambda} = 0 = \frac{-2(\lambda-2)(\lambda+1) + (\lambda-2)^2}{(\lambda+1)^2} - 5$$

$$\Rightarrow (\lambda-2)(-\lambda-4) - 5\lambda^2 - 10\lambda - 5 = 0, \ \lambda \ne -1$$

$$\Rightarrow -6\lambda^2 - 12\lambda + 3 = 0, \ \lambda \ne -1$$

$$\Rightarrow 2\lambda^2 + 4\lambda - 1 = 0, \ \lambda \ne -1$$

$$\Rightarrow \lambda = \frac{-4 \pm 2\sqrt{6}}{4} = -1 + \frac{\sqrt{6}}{2} \text{ since } \lambda \ge 0$$

(ii)

For $\lambda = -1 + \frac{\sqrt{6}}{2}$, $L = p^*$

$$L = -\left(-3+\frac{\sqrt{6}}{2}\right)^2 / \frac{\sqrt{6}}{2} + \left(9 - \frac{5}{2}\sqrt{6}\right)$$

$\frac{2}{\sqrt{6}}$ $= \left(\frac{-21}{2} + 3\sqrt{6}\right) / \frac{\sqrt{6}}{2} + 9 - \frac{5}{2}\sqrt{6}$

$$= \frac{-\overset{7}{\cancel{21}}\sqrt{6}}{2\cancel{\sqrt{6}}} + 6 + 9 - \frac{5}{2}\sqrt{6}$$

$$= 15 - 6\sqrt{6}$$