CAP 5610
Assignment #3 Solution

March 28, 2025

Arman Sayan

---

1 Kernel Computation Cost [30 points]

1. [10 points] Consider we have a two-dimensional input space such that the input vector is $x = (x_1, x_2)^T$. Define the feature mapping $\phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)^T$. What is the corresponding kernel function, i.e., $K(x, z)$? Do not leave $\phi(x)$ in your final answer.

Ans:
To find the kernel function $K(x, z)$, we use the definition of the kernel function as follows:

$$K(x, z) = \phi(x)^T \phi(z)$$

where $\phi(x)$ represents feature mapping.
Given that $\phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)^T$, for another input vector $z = (z_1, z_2)^T$, the feature mapping is $\phi(z) = (z_1^2, \sqrt{2}z_1z_2, z_2^2)^T$.
Now, we can compute the dot product of the feature mappings as follows:

$$\begin{aligned} K(x, z) &= \phi(x)^T \phi(z) \\ &= \begin{bmatrix} x_1^2 & \sqrt{2}x_1x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} z_1^2 \\ \sqrt{2}z_1z_2 \\ z_2^2 \end{bmatrix} \\ &= x_1^2 z_1^2 + 2x_1x_2z_1z_2 + x_2^2 z_2^2 \end{aligned}$$

With a deduction from the above calculations, we can observe the final expression as

$$K(x, z) = (x_1z_1 + x_2z_2)^2$$

Since $x_1z_1 + x_2z_2$ is the dot product of the input vectors x and z, namely $x^T z$, the corresponding kernel function is

$$\boxed{\boldsymbol{K(x, z) = (x^T z)^2}}$$

2. [20 points] Suppose we want to compute the value of the kernel function $K(x, z)$ from the previous question on two vectors $x, z \in \mathbb{R}^2$. How many additions and multiplications are needed if you

   i. [10 points] Map the input vector to the feature space and then perform the dot product on the mapped features?

Ans:
Using the feature mappings
$$\phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)^T \text{ and } \phi(z) = (z_1^2, \sqrt{2}z_1z_2, z_2^2)^T,$$
the kernel function $K(x, z)$ can be computed as

$$\begin{aligned} K(x,z) &= \phi(x)^T\phi(z) \\ &= \begin{bmatrix} x_1^2 & \sqrt{2}x_1x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} z_1^2 \\ \sqrt{2}z_1z_2 \\ z_2^2 \end{bmatrix} \\ &= x_1^2z_1^2 + 2x_1x_2z_1z_2 + x_2^2z_2^2 \end{aligned}$$

For the feature mapping of the vector x, we need to compute terms x_1^2, x_2^2, and $\sqrt{2}x_1x_2$. To calculate these terms, we need 3 multiplications. Since we have 2 input vectors x and z, we need to perform 3 multiplications for each vector, which results in 6 multiplications for mapping the vectors to the feature space.
After mapping the vectors to the feature space, we need to compute the dot product of the feature mappings as
$$K(x, z) = x_1^2z_1^2 + 2x_1x_2z_1z_2 + x_2^2z_2^2$$
This requires 3 multiplications for each term in the summation and 2 additions to sum three terms.
In total, we need $\boxed{\textbf{9 multiplications and 2 additions}}$ to compute the value of $K(x, z)$.

ii. [10 points] Compute through the kernel function you derived in question 1?

Ans:
The kernel function derived in question 1 is $K(x, z) = (x^T z)^2$.
To compute the value of the kernel function $K(x, z)$ using this kernel function, we need to calculate the dot product of the input vectors x and z, namely $x^T z$.
The dot product of two vectors x and z is calculated as

$$x^T z = x_1 z_1 + x_2 z_2$$

This requires 2 multiplications for terms $x_1 z_1$ and $x_2 z_2$ and 1 addition to sum those terms.
After computing the dot product, we need to square the result to obtain the value of the kernel function $K(x, z)$.
Squaring the result requires 1 multiplication.
In total, we need $\boxed{\textbf{3 multiplications and 1 addition}}$ to compute the value of $K(x, z)$.

2 Activation Functions and Loss Functions [30 points]

1. [20 points] For this assignment, you are encouraged to consult Dr. GOOGLE and Dr. ChatGPT. But please explain things in your own language while you write the answer. For each of the following activation functions, briefly describe the type of non-linearity (if any) it introduces and discuss their pros and cons.
   (a) Linear Activation Function
   (b) Sigmoid Activation Function
   (c) Tanh Activation Function
   (d) ReLU (Rectified Linear Unit) Activation Function

Ans:

(a) **Linear Activation Function:**

$$f(x) = x$$

Linear activation function introduces a linear relationship between the input and output where the output is directly proportional to the input. It is a simple activation function that does not introduce any non-linearity to the model.

It is simple and computationally efficient, making it useful for regression problems. However, it does not introduce any non-linearity, meaning it cannot capture complex relationships in data, and stacking multiple linear layers does not increase the model's learning capacity.

(b) **Sigmoid Activation Function:**

$$f(x) = \frac{1}{1 + e^{-x}}$$

Sigmoid activation function introduces a non-linearity that squashes the input values between 0 and 1 in a S-shaped curve.

It is useful for binary classification problems where the output is required to be in the range of 0 and 1. Furthermore, it provides a smooth gradient that helps in training the model. However, it suffers from the vanishing gradient problem, which makes training deep neural networks difficult. Also, it is not zero-centered, namely the output is always positive, which can lead to convergence issues.

(c) **Tanh Activation Function:**

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Tanh activation function introduces a non-linearity that squashes the input values in a S-shaped curve, similar to sigmoid activation function, but output values between -1 and 1.

It is zero-centered, which helps in faster convergence during training. Also, it provides stronger gradients for hidden layers compared to sigmoid activation function. However, just like sigmoid activation function, it also suffers from the vanishing gradient problem, making it difficult to train deep neural networks. Furthermore, it is not suitable for sparse data. Lastly, it is computationally expensive due to the exponential operations.

(d) **ReLU (Rectified Linear Unit) Activation Function:**

$$f(x) = \max(0, x)$$

ReLU activation function introduces a non-linearity that outputs the input as it is if it is positive, and zero otherwise, making computations simple.

It is computationally efficient and helps in faster convergence during training compared to sigmoid and tanh. Also, it does not suffer from the vanishing gradient problem, making it suitable for training deep neural networks. Lastly, it remains one of the most widely used activation functions due to its strong performance in deep networks. However, it suffers from the dying ReLU problem where neurons can die during training and stop learning when inputs become negative and ReLU outputs 0. Also, it is not zero-centered, which can lead to convergence issues. Lastly, it is not suitable for small input values as it ignores negative information.

2. [10 points] For each of the following loss functions, briefly describe their mathematical formula and discuss for which type of learning task (classification, regression, etc.) they are most appropriate.
   (a) Mean Squared Error Loss
   (b) Binary Cross Entropy Loss
   (c) Hinge Loss
   (d) Softmax Cross Entropy Loss

Ans:

(a) **Mean Squared Error Loss:**

Mean Squared Error (MSE) loss is a regression loss function that measures the average squared difference between the predicted values $\hat{y}_i$ and actual values y_i.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where N is the number of samples in the dataset, y_i is the actual target value, and $\hat{y}_i$ is the predicted value.

The squaring ensures that large errors are penalized more heavily than small errors.

It is most appropriate for regression tasks where the model is required to predict continuous values.

(b) **Binary Cross Entropy Loss:**

Binary Cross Entropy loss is a classification loss function that measures the difference between the predicted probability $\hat{y}_i$ and actual class labels for binary classification tasks.

$$\text{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

where N is the number of samples in the dataset, y_i is the actual class label such as 0 or 1, and $\hat{y}_i$ is the predicted probability between 0 and 1.

The logarithm makes wrong confident predictions highly penalized.

It is most appropriate for binary classification tasks where the model is required to predict probabilities for two classes.

(c) **Hinge Loss:**

Hinge loss is a classification loss function that measures the margin of the predicted class label $\hat{y}_i$ from the decision boundary for binary classification tasks.

$$\text{Hinge Loss} = \frac{1}{N} \sum_{i=1}^{N} \max(0, 1 - y_i \hat{y}_i)$$

where N is the number of samples in the dataset, y_i is the actual class label such as -1 or 1, and $\hat{y}_i$ is the predicted class label.

If the true label y_i and predicted label $\hat{y}_i$ have the same sign and a large margin greater than 1, the loss is 0. If that is not the case, it penalizes the prediction proportionally, encouraging a large decision margin between different classes.

It is most appropriate for binary classification tasks where the model is required to predict class labels, like in SVMs.

(d) **Softmax Cross Entropy Loss:**

Softmax Cross Entropy loss is a classification loss function that measures the difference between the predicted and actual class labels for multi-class classification tasks.

$$\text{Softmax Cross Entropy} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log\left(\frac{e^{\hat{y}_{i,c}}}{\sum_{k=1}^{C} e^{\hat{y}_{i,k}}}\right)$$

where N is the number of samples in the dataset, C is the number of classes, $y_{i,c}$ is 1 if the true class is c and 0 otherwise, and $\hat{y}_{i,c}$ is the raw predicted probability for class c before applying softmax.

The softmax function converts raw model outputs into probabilities, and encourages high confidence in the correct class while penalizing incorrect ones. The cross-entropy term compares the predicted probability of the correct class to true label.

It is most appropriate for multi-class classification tasks where each sample belongs to one of many classes and the model is required to predict probabilities for multiple classes.

3 Linear SVM Implementation [40 points]

Solution for Q3:

Ans:

Please check the source codes and outputs included in the appendix named as

svm.py and **SVM.ipynb**

for the solution.

A Appendix

svm.py

March 28, 2025

[ ]:
```python
"""Support Vector Machine (SVM) model."""

# This source code is modified by Arman Sayan.

# Last Edit: March 28, 2024

import numpy as np


class SVM:
    def __init__(self, n_class: int, lr: float, epochs: int, reg_const: float):
        """Initialize a new classifier.

        Parameters:
            n_class: the number of classes
            lr: the learning rate
            epochs: the number of epochs to train for
            reg_const: the regularization constant
        """
        self.w = None  # Weight matrix of shape (D, C), initialized during␣
 ↪training
        self.alpha = lr
        self.epochs = epochs
        self.reg_const = reg_const
        self.n_class = n_class

    def calc_gradient(self, X_train: np.ndarray, y_train: np.ndarray) -> np.
 ↪ndarray:
        """Calculate gradient of the svm hinge loss.

        Inputs have dimension D, there are C classes, and we operate on
        mini-batches of N examples.

        Parameters:
            X_train: a numpy array of shape (N, D) containing a mini-batch
                of data
            y_train: a numpy array of shape (N,) containing training labels;
```

```
                y[i] = c means that X[i] has label c, where 0 <= c < C

        Returns:
            the gradient with respect to weights w; an array of the same shape
                as w
        """
        N, D = X_train.shape

        # Compute class scores for all samples
        scores = X_train @ self.w  # Shape (N, C)

        # Extract the scores of the correct classes
        correct_class_scores = scores[np.arange(N), y_train].reshape(-1, 1)

        # Compute the margins for all classes
        margins = np.maximum(0, scores - correct_class_scores + 1)

        # Zero-out the margins for the correct classes
        margins[np.arange(N), y_train] = 0  # Ignore correct class

        # Binary indicator: 1 where margin > 0
        indicator = (margins > 0).astype(float)  # Indicator for incorrect␣
↪classes

        # For each example, subtract total count of violations from the correct␣
↪class column
        indicator[np.arange(N), y_train] = -np.sum(indicator, axis=1)  # Adjust␣
↪correct class

        # Compute gradient and add L2 regularization
        grad = (X_train.T @ indicator) / N + self.reg_const * self.w  # Add␣
↪regularization
        return grad

    def train(self, X_train: np.ndarray, y_train: np.ndarray):
        """Train the classifier.

        Hint: operate on mini-batches of data for SGD.

        Parameters:
            X_train: a numpy array of shape (N, D) containing training data;
                N examples with D dimensions
            y_train: a numpy array of shape (N,) containing training labels
        """
        N, D = X_train.shape

        # Initialize weights randomly if not already initialized
```

```python
        if self.w is None:
            self.w = np.random.randn(D, self.n_class) * 0.01  # Initialize small random weights

        # Perform gradient descent for a number of epochs
        for epoch in range(self.epochs):
            # Compute the gradient of the current loss
            gradient = self.calc_gradient(X_train, y_train)

            # Update weights using the gradient
            self.w -= self.alpha * gradient  # Update weights

            # Every 100 epochs, compute and print the average hinge loss
            if epoch % 100 == 0:
                scores = X_train @ self.w  # (N, C)
                correct_class_scores = scores[np.arange(N), y_train].reshape(-1, 1)  # (N, 1)
                margins = np.maximum(0, scores - correct_class_scores + 1)  # (N, C)
                margins[np.arange(N), y_train] = 0  # Zero out correct class
                loss = np.mean(np.sum(margins, axis=1))  # Hinge loss averaged over batch
                print(f"Epoch {epoch}: Loss = {loss:.4f}")


    def predict(self, X_test: np.ndarray) -> np.ndarray:
        """Use the trained weights to predict labels for test data points.

        Parameters:
            X_test: a numpy array of shape (N, D) containing testing data;
                N examples with D dimensions

        Returns:
            predicted labels for the data in X_test; a 1-dimensional array of
                length N, where each element is an integer giving the predicted
                class.
        """
        # Compute class scores and return the index of the highest score (best class)
        return np.argmax(X_test @ self.w, axis=1)
```

SVM

March 28, 2025

CAP 5610 Assignment #3: Kernels, Activation Function, SVM

This source code is modified by Arman Sayan.

Last Edit: March 28, 2024

[1]:
```
import random
import numpy as np
from data_process import get_CIFAR10_data, get_MUSHROOM_data
from scipy.spatial import distance
from models.svm import SVM
#from kaggle_submission import output_submission_csv
%matplotlib inline

# For auto-reloading external modules
# See http://stackoverflow.com/questions/1907993/
 ↪autoreload-of-modules-in-ipython
%load_ext autoreload
%autoreload 2
```

1 Loading Mushroom

In the following cells we determine the splitting of the mushroom dataset. TRAINING + VALIDATION = 0.8, TESTING = 0.2

[2]:
```
# TRAINING = 0.6 indicates 60% of the data is used as the training dataset.
VALIDATION = 0.2
```

[3]:
```
data = get_MUSHROOM_data(VALIDATION)
X_train_MR, y_train_MR = data['X_train'], data['y_train']
X_val_MR, y_val_MR = data['X_val'], data['y_val']
X_test_MR, y_test_MR = data['X_test'], data['y_test']
n_class_MR = len(np.unique(y_test_MR))

# IMPORTANT!!!!
# When above code is unccommented and update the labels
# in the dataset according to the document, training,
# validation and test accuracies stays flat,
```

```
# for all the hyperparemeter settings.
# Hence, the code is commented out.

# Change the labels from {0, 1} to {-1, 1}
# according to the documentation
#y_train_MR = 2 * y_train_MR - 1  # Converts 0 → -1, 1 → 1
#y_val_MR = 2 * y_val_MR - 1
#y_test_MR = 2 * y_test_MR - 1

print("Number of train samples: ", X_train_MR.shape[0])
print("Number of val samples: ", X_val_MR.shape[0])
print("Number of test samples: ", X_test_MR.shape[0])
```

```
Number of train samples:  4874
Number of val samples:  1625
Number of test samples:  1625
```

1.0.1 Get Accuracy

This function computes how well your model performs using accuracy as a metric.

[4]:
```
def get_acc(pred, y_test):
    return np.sum(y_test == pred) / len(y_test) * 100
```

2 Support Vector Machines (with SGD)

Next, you will implement a "soft margin" SVM. In this formulation you will maximize the margin between positive and negative training examples and penalize margin violations using a hinge loss.

We will optimize the SVM loss using SGD. This means you must compute the loss function with respect to model weights. You will use this gradient to update the model weights.

SVM optimized with SGD has 3 hyperparameters that you can experiment with: - **Learning rate** - similar to as defined above in Perceptron, this parameter scales by how much the weights are changed according to the calculated gradient update. - **Epochs** - similar to as defined above in Perceptron. - **Regularization constant** - Hyperparameter to determine the strength of regularization. In this case it is a coefficient on the term which maximizes the margin. You could try different values. The default value is set to 0.05.

You will implement the SVM using SGD in the **models/svm.py**

The following code: - Creates an instance of the SVM classifier class - The train function of the SVM class is trained on the training data - We use the predict function to find the training accuracy as well as the testing accuracy

2.1 Train SVM on Mushroom

[5]:
```
lr_list = [0.001, 0.005, 0.01, 0.05, 0.1]
n_epochs_list = [100, 500, 1000, 2000, 5000]
reg_const_list = [0.001, 0.01, 0.05, 0.1, 0.5]

#svm_MR = SVM(n_class_MR, lr, n_epochs, reg_const)
#svm_MR.train(X_train_MR, y_train_MR)
```

[6]:
```
#pred_svm = svm_MR.predict(X_train_MR)
#print('The training accuracy is given by: %f' % (get_acc(pred_svm,␣
↪y_train_MR)))
```

2.1.1 Validate SVM on Mushroom

[7]:
```
#pred_svm = svm_MR.predict(X_val_MR)
#print('The validation accuracy is given by: %f' % (get_acc(pred_svm,␣
↪y_val_MR)))
```

2.2 Test SVM on Mushroom

[8]:
```
#pred_svm = svm_MR.predict(X_test_MR)
#print('The testing accuracy is given by: %f' % (get_acc(pred_svm, y_test_MR)))
```

2.3 Reporting the Optimal Hyperparameter Setting

[9]:
```
def TuneSVMHyperparameters(X_train_MR, y_train_MR, X_val_MR, y_val_MR,␣
↪X_test_MR, y_test_MR, n_class_MR, param_grid):
    best_train_acc = 0
    best_val_acc = 0
    best_test_acc = 0
    best_model = None
    best_params = {}
    results = []

    for lr in param_grid['lr']:
        for epochs in param_grid['epochs']:
            for reg in param_grid['reg_const']:
                print(f"Training SVM with lr={lr}, epochs={epochs},␣
↪reg_const={reg}...")

                # Initialize and train model
                svm_MR = SVM(n_class_MR, lr, epochs, reg)
                svm_MR.train(X_train_MR, y_train_MR)

                # Evaluate on training set
                pred_train = svm_MR.predict(X_train_MR)
```

```
            train_acc = get_acc(pred_train, y_train_MR)

            # Evaluate on validation set
            pred_val = svm_MR.predict(X_val_MR)
            val_acc = get_acc(pred_val, y_val_MR)

            # Evaluate on test set
            pred_test = svm_MR.predict(X_test_MR)
            test_acc = get_acc(pred_test, y_test_MR)

            print("Training Accuracy:   %.4f" % train_acc)
            print("Validation Accuracy: %.4f" % val_acc)
            print("Testing Accuracy:    %.4f" % test_acc)

            # Update best if validation accuracy is highest
            if val_acc > best_val_acc:
                best_val_acc = val_acc
                best_train_acc = train_acc
                best_test_acc = test_acc
                best_model = svm_MR
                best_params = {
                    'lr': lr,
                    'epochs': epochs,
                    'reg_const': reg
                }

    print("\nBest Hyperparameters Found:")
    print(best_params)
    print("Best Training Accuracy:   %.4f" % best_train_acc)
    print("Best Validation Accuracy: %.4f" % best_val_acc)
    print("Best Testing Accuracy:    %.4f" % best_test_acc)

    return best_model, best_params
```

[10]:
```
param_grid = {
    'lr': lr_list,
    'epochs': n_epochs_list,
    'reg_const': reg_const_list
}

best_model, best_params = TuneSVMHyperparameters(
    X_train_MR, y_train_MR,
    X_val_MR, y_val_MR,
    X_test_MR, y_test_MR,
    n_class_MR,
    param_grid
)
```

```
Training SVM with lr=0.001, epochs=100, reg_const=0.001…
Epoch 0: Loss = 0.9670
Training Accuracy:   80.7755
Validation Accuracy: 78.5231
Testing Accuracy:    80.9231
Training SVM with lr=0.001, epochs=100, reg_const=0.01…
Epoch 0: Loss = 1.0199
Training Accuracy:   79.3188
Validation Accuracy: 77.0462
Testing Accuracy:    80.1231
Training SVM with lr=0.001, epochs=100, reg_const=0.05…
Epoch 0: Loss = 1.0291
Training Accuracy:   79.2162
Validation Accuracy: 77.2308
Testing Accuracy:    79.8154
Training SVM with lr=0.001, epochs=100, reg_const=0.1…
Epoch 0: Loss = 1.0174
Training Accuracy:   79.7497
Validation Accuracy: 77.2923
Testing Accuracy:    80.3692
Training SVM with lr=0.001, epochs=100, reg_const=0.5…
Epoch 0: Loss = 0.9822
Training Accuracy:   80.0164
Validation Accuracy: 78.0923
Testing Accuracy:    80.8615
Training SVM with lr=0.001, epochs=500, reg_const=0.001…
Epoch 0: Loss = 0.9988
Epoch 100: Loss = 0.5157
Epoch 200: Loss = 0.4401
Epoch 300: Loss = 0.4049
Epoch 400: Loss = 0.3839
Training Accuracy:   87.0332
Validation Accuracy: 85.4154
Testing Accuracy:    86.0308
Training SVM with lr=0.001, epochs=500, reg_const=0.01…
Epoch 0: Loss = 0.9741
Epoch 100: Loss = 0.5032
Epoch 200: Loss = 0.4381
Epoch 300: Loss = 0.4052
Epoch 400: Loss = 0.3842
Training Accuracy:   86.8281
Validation Accuracy: 85.2308
Testing Accuracy:    86.1538
Training SVM with lr=0.001, epochs=500, reg_const=0.05…
Epoch 0: Loss = 1.0289
Epoch 100: Loss = 0.5099
Epoch 200: Loss = 0.4394
Epoch 300: Loss = 0.4063
```

```
Epoch 400: Loss = 0.3855
Training Accuracy:   86.9717
Validation Accuracy: 85.2923
Testing Accuracy:    86.1538
Training SVM with lr=0.001, epochs=500, reg_const=0.1…
Epoch 0: Loss = 1.0197
Epoch 100: Loss = 0.5082
Epoch 200: Loss = 0.4394
Epoch 300: Loss = 0.4063
Epoch 400: Loss = 0.3856
Training Accuracy:   86.9101
Validation Accuracy: 85.2308
Testing Accuracy:    85.9692
Training SVM with lr=0.001, epochs=500, reg_const=0.5…
Epoch 0: Loss = 0.9783
Epoch 100: Loss = 0.5177
Epoch 200: Loss = 0.4478
Epoch 300: Loss = 0.4155
Epoch 400: Loss = 0.3961
Training Accuracy:   86.5819
Validation Accuracy: 84.8615
Testing Accuracy:    85.4154
Training SVM with lr=0.001, epochs=1000, reg_const=0.001…
Epoch 0: Loss = 1.0111
Epoch 100: Loss = 0.5298
Epoch 200: Loss = 0.4474
Epoch 300: Loss = 0.4090
Epoch 400: Loss = 0.3867
Epoch 500: Loss = 0.3707
Epoch 600: Loss = 0.3580
Epoch 700: Loss = 0.3474
Epoch 800: Loss = 0.3383
Epoch 900: Loss = 0.3305
Training Accuracy:   88.4284
Validation Accuracy: 87.0154
Testing Accuracy:    87.0154
Training SVM with lr=0.001, epochs=1000, reg_const=0.01…
Epoch 0: Loss = 0.9361
Epoch 100: Loss = 0.5046
Epoch 200: Loss = 0.4403
Epoch 300: Loss = 0.4087
Epoch 400: Loss = 0.3881
Epoch 500: Loss = 0.3724
Epoch 600: Loss = 0.3599
Epoch 700: Loss = 0.3492
Epoch 800: Loss = 0.3400
Epoch 900: Loss = 0.3320
Training Accuracy:   88.3668
```

```
Validation Accuracy: 86.9538
Testing Accuracy:    86.9538
Training SVM with lr=0.001, epochs=1000, reg_const=0.05…
Epoch 0: Loss = 0.9946
Epoch 100: Loss = 0.5157
Epoch 200: Loss = 0.4442
Epoch 300: Loss = 0.4100
Epoch 400: Loss = 0.3888
Epoch 500: Loss = 0.3732
Epoch 600: Loss = 0.3609
Epoch 700: Loss = 0.3505
Epoch 800: Loss = 0.3415
Epoch 900: Loss = 0.3338
Training Accuracy:   88.4694
Validation Accuracy: 86.8923
Testing Accuracy:    87.0154
Training SVM with lr=0.001, epochs=1000, reg_const=0.1…
Epoch 0: Loss = 1.0266
Epoch 100: Loss = 0.5228
Epoch 200: Loss = 0.4470
Epoch 300: Loss = 0.4112
Epoch 400: Loss = 0.3897
Epoch 500: Loss = 0.3742
Epoch 600: Loss = 0.3620
Epoch 700: Loss = 0.3520
Epoch 800: Loss = 0.3434
Epoch 900: Loss = 0.3360
Training Accuracy:   88.0796
Validation Accuracy: 86.8923
Testing Accuracy:    86.8923
Training SVM with lr=0.001, epochs=1000, reg_const=0.5…
Epoch 0: Loss = 1.0116
Epoch 100: Loss = 0.5148
Epoch 200: Loss = 0.4452
Epoch 300: Loss = 0.4130
Epoch 400: Loss = 0.3935
Epoch 500: Loss = 0.3799
Epoch 600: Loss = 0.3693
Epoch 700: Loss = 0.3607
Epoch 800: Loss = 0.3536
Epoch 900: Loss = 0.3476
Training Accuracy:   87.8129
Validation Accuracy: 86.7077
Testing Accuracy:    86.5846
Training SVM with lr=0.001, epochs=2000, reg_const=0.001…
Epoch 0: Loss = 1.0039
Epoch 100: Loss = 0.5102
Epoch 200: Loss = 0.4395
```

```
Epoch 300: Loss = 0.4057
Epoch 400: Loss = 0.3842
Epoch 500: Loss = 0.3684
Epoch 600: Loss = 0.3558
Epoch 700: Loss = 0.3451
Epoch 800: Loss = 0.3361
Epoch 900: Loss = 0.3283
Epoch 1000: Loss = 0.3215
Epoch 1100: Loss = 0.3154
Epoch 1200: Loss = 0.3099
Epoch 1300: Loss = 0.3048
Epoch 1400: Loss = 0.3003
Epoch 1500: Loss = 0.2963
Epoch 1600: Loss = 0.2926
Epoch 1700: Loss = 0.2891
Epoch 1800: Loss = 0.2858
Epoch 1900: Loss = 0.2828
Training Accuracy:   90.2954
Validation Accuracy: 89.0462
Testing Accuracy:    88.9231
Training SVM with lr=0.001, epochs=2000, reg_const=0.01…
Epoch 0: Loss = 0.9371
Epoch 100: Loss = 0.4998
Epoch 200: Loss = 0.4373
Epoch 300: Loss = 0.4060
Epoch 400: Loss = 0.3858
Epoch 500: Loss = 0.3706
Epoch 600: Loss = 0.3583
Epoch 700: Loss = 0.3479
Epoch 800: Loss = 0.3390
Epoch 900: Loss = 0.3312
Epoch 1000: Loss = 0.3244
Epoch 1100: Loss = 0.3184
Epoch 1200: Loss = 0.3128
Epoch 1300: Loss = 0.3077
Epoch 1400: Loss = 0.3032
Epoch 1500: Loss = 0.2991
Epoch 1600: Loss = 0.2954
Epoch 1700: Loss = 0.2919
Epoch 1800: Loss = 0.2886
Epoch 1900: Loss = 0.2856
Training Accuracy:   90.1723
Validation Accuracy: 88.9231
Testing Accuracy:    88.8615
Training SVM with lr=0.001, epochs=2000, reg_const=0.05…
Epoch 0: Loss = 0.9964
Epoch 100: Loss = 0.5190
Epoch 200: Loss = 0.4439
```

```
Epoch 300: Loss = 0.4083
Epoch 400: Loss = 0.3868
Epoch 500: Loss = 0.3708
Epoch 600: Loss = 0.3582
Epoch 700: Loss = 0.3478
Epoch 800: Loss = 0.3390
Epoch 900: Loss = 0.3313
Epoch 1000: Loss = 0.3247
Epoch 1100: Loss = 0.3187
Epoch 1200: Loss = 0.3134
Epoch 1300: Loss = 0.3086
Epoch 1400: Loss = 0.3042
Epoch 1500: Loss = 0.3002
Epoch 1600: Loss = 0.2967
Epoch 1700: Loss = 0.2934
Epoch 1800: Loss = 0.2903
Epoch 1900: Loss = 0.2874
Training Accuracy:   90.1108
Validation Accuracy: 88.8000
Testing Accuracy:    88.8615
Training SVM with lr=0.001, epochs=2000, reg_const=0.1…
Epoch 0: Loss = 1.0251
Epoch 100: Loss = 0.5267
Epoch 200: Loss = 0.4521
Epoch 300: Loss = 0.4164
Epoch 400: Loss = 0.3943
Epoch 500: Loss = 0.3779
Epoch 600: Loss = 0.3650
Epoch 700: Loss = 0.3543
Epoch 800: Loss = 0.3450
Epoch 900: Loss = 0.3371
Epoch 1000: Loss = 0.3303
Epoch 1100: Loss = 0.3243
Epoch 1200: Loss = 0.3188
Epoch 1300: Loss = 0.3139
Epoch 1400: Loss = 0.3095
Epoch 1500: Loss = 0.3054
Epoch 1600: Loss = 0.3018
Epoch 1700: Loss = 0.2985
Epoch 1800: Loss = 0.2955
Epoch 1900: Loss = 0.2927
Training Accuracy:   89.9467
Validation Accuracy: 88.5538
Testing Accuracy:    88.6769
Training SVM with lr=0.001, epochs=2000, reg_const=0.5…
Epoch 0: Loss = 0.9746
Epoch 100: Loss = 0.5128
Epoch 200: Loss = 0.4454
```

```
Epoch 300: Loss = 0.4138
Epoch 400: Loss = 0.3945
Epoch 500: Loss = 0.3809
Epoch 600: Loss = 0.3702
Epoch 700: Loss = 0.3616
Epoch 800: Loss = 0.3545
Epoch 900: Loss = 0.3485
Epoch 1000: Loss = 0.3434
Epoch 1100: Loss = 0.3390
Epoch 1200: Loss = 0.3351
Epoch 1300: Loss = 0.3316
Epoch 1400: Loss = 0.3286
Epoch 1500: Loss = 0.3260
Epoch 1600: Loss = 0.3237
Epoch 1700: Loss = 0.3216
Epoch 1800: Loss = 0.3198
Epoch 1900: Loss = 0.3180
Training Accuracy:   88.6541
Validation Accuracy: 87.4462
Testing Accuracy:    87.6308
Training SVM with lr=0.001, epochs=5000, reg_const=0.001…
Epoch 0: Loss = 0.8939
Epoch 100: Loss = 0.4974
Epoch 200: Loss = 0.4358
Epoch 300: Loss = 0.4056
Epoch 400: Loss = 0.3855
Epoch 500: Loss = 0.3703
Epoch 600: Loss = 0.3580
Epoch 700: Loss = 0.3476
Epoch 800: Loss = 0.3386
Epoch 900: Loss = 0.3308
Epoch 1000: Loss = 0.3240
Epoch 1100: Loss = 0.3178
Epoch 1200: Loss = 0.3122
Epoch 1300: Loss = 0.3071
Epoch 1400: Loss = 0.3025
Epoch 1500: Loss = 0.2983
Epoch 1600: Loss = 0.2945
Epoch 1700: Loss = 0.2909
Epoch 1800: Loss = 0.2876
Epoch 1900: Loss = 0.2845
Epoch 2000: Loss = 0.2817
Epoch 2100: Loss = 0.2790
Epoch 2200: Loss = 0.2765
Epoch 2300: Loss = 0.2741
Epoch 2400: Loss = 0.2719
Epoch 2500: Loss = 0.2697
Epoch 2600: Loss = 0.2677
```

```
Epoch 2700: Loss = 0.2657
Epoch 2800: Loss = 0.2639
Epoch 2900: Loss = 0.2621
Epoch 3000: Loss = 0.2605
Epoch 3100: Loss = 0.2588
Epoch 3200: Loss = 0.2573
Epoch 3300: Loss = 0.2558
Epoch 3400: Loss = 0.2543
Epoch 3500: Loss = 0.2529
Epoch 3600: Loss = 0.2515
Epoch 3700: Loss = 0.2501
Epoch 3800: Loss = 0.2488
Epoch 3900: Loss = 0.2474
Epoch 4000: Loss = 0.2462
Epoch 4100: Loss = 0.2449
Epoch 4200: Loss = 0.2437
Epoch 4300: Loss = 0.2425
Epoch 4400: Loss = 0.2413
Epoch 4500: Loss = 0.2402
Epoch 4600: Loss = 0.2391
Epoch 4700: Loss = 0.2380
Epoch 4800: Loss = 0.2369
Epoch 4900: Loss = 0.2358
Training Accuracy:   91.8342
Validation Accuracy: 90.8923
Testing Accuracy:    91.3846
Training SVM with lr=0.001, epochs=5000, reg_const=0.01…
Epoch 0: Loss = 1.0045
Epoch 100: Loss = 0.5160
Epoch 200: Loss = 0.4436
Epoch 300: Loss = 0.4095
Epoch 400: Loss = 0.3885
Epoch 500: Loss = 0.3727
Epoch 600: Loss = 0.3601
Epoch 700: Loss = 0.3495
Epoch 800: Loss = 0.3403
Epoch 900: Loss = 0.3323
Epoch 1000: Loss = 0.3253
Epoch 1100: Loss = 0.3191
Epoch 1200: Loss = 0.3134
Epoch 1300: Loss = 0.3082
Epoch 1400: Loss = 0.3035
Epoch 1500: Loss = 0.2993
Epoch 1600: Loss = 0.2955
Epoch 1700: Loss = 0.2919
Epoch 1800: Loss = 0.2885
Epoch 1900: Loss = 0.2854
Epoch 2000: Loss = 0.2825
```

```
Epoch 2100: Loss = 0.2798
Epoch 2200: Loss = 0.2772
Epoch 2300: Loss = 0.2748
Epoch 2400: Loss = 0.2726
Epoch 2500: Loss = 0.2705
Epoch 2600: Loss = 0.2684
Epoch 2700: Loss = 0.2664
Epoch 2800: Loss = 0.2646
Epoch 2900: Loss = 0.2628
Epoch 3000: Loss = 0.2611
Epoch 3100: Loss = 0.2595
Epoch 3200: Loss = 0.2579
Epoch 3300: Loss = 0.2564
Epoch 3400: Loss = 0.2550
Epoch 3500: Loss = 0.2536
Epoch 3600: Loss = 0.2522
Epoch 3700: Loss = 0.2508
Epoch 3800: Loss = 0.2495
Epoch 3900: Loss = 0.2482
Epoch 4000: Loss = 0.2469
Epoch 4100: Loss = 0.2457
Epoch 4200: Loss = 0.2445
Epoch 4300: Loss = 0.2433
Epoch 4400: Loss = 0.2422
Epoch 4500: Loss = 0.2410
Epoch 4600: Loss = 0.2399
Epoch 4700: Loss = 0.2388
Epoch 4800: Loss = 0.2378
Epoch 4900: Loss = 0.2367
Training Accuracy:   91.8137
Validation Accuracy: 90.8923
Testing Accuracy:    91.3231
Training SVM with lr=0.001, epochs=5000, reg_const=0.05…
Epoch 0: Loss = 0.9793
Epoch 100: Loss = 0.5149
Epoch 200: Loss = 0.4422
Epoch 300: Loss = 0.4072
Epoch 400: Loss = 0.3856
Epoch 500: Loss = 0.3697
Epoch 600: Loss = 0.3570
Epoch 700: Loss = 0.3465
Epoch 800: Loss = 0.3376
Epoch 900: Loss = 0.3299
Epoch 1000: Loss = 0.3232
Epoch 1100: Loss = 0.3173
Epoch 1200: Loss = 0.3120
Epoch 1300: Loss = 0.3072
Epoch 1400: Loss = 0.3028
```

```
Epoch 1500: Loss = 0.2989
Epoch 1600: Loss = 0.2953
Epoch 1700: Loss = 0.2920
Epoch 1800: Loss = 0.2889
Epoch 1900: Loss = 0.2861
Epoch 2000: Loss = 0.2834
Epoch 2100: Loss = 0.2809
Epoch 2200: Loss = 0.2786
Epoch 2300: Loss = 0.2764
Epoch 2400: Loss = 0.2743
Epoch 2500: Loss = 0.2724
Epoch 2600: Loss = 0.2705
Epoch 2700: Loss = 0.2688
Epoch 2800: Loss = 0.2671
Epoch 2900: Loss = 0.2654
Epoch 3000: Loss = 0.2638
Epoch 3100: Loss = 0.2623
Epoch 3200: Loss = 0.2609
Epoch 3300: Loss = 0.2595
Epoch 3400: Loss = 0.2582
Epoch 3500: Loss = 0.2569
Epoch 3600: Loss = 0.2556
Epoch 3700: Loss = 0.2544
Epoch 3800: Loss = 0.2532
Epoch 3900: Loss = 0.2521
Epoch 4000: Loss = 0.2509
Epoch 4100: Loss = 0.2499
Epoch 4200: Loss = 0.2488
Epoch 4300: Loss = 0.2477
Epoch 4400: Loss = 0.2467
Epoch 4500: Loss = 0.2457
Epoch 4600: Loss = 0.2447
Epoch 4700: Loss = 0.2438
Epoch 4800: Loss = 0.2428
Epoch 4900: Loss = 0.2419
Training Accuracy:   91.4444
Validation Accuracy: 90.4000
Testing Accuracy:    91.0769
Training SVM with lr=0.001, epochs=5000, reg_const=0.1…
Epoch 0: Loss = 1.0580
Epoch 100: Loss = 0.5153
Epoch 200: Loss = 0.4467
Epoch 300: Loss = 0.4130
Epoch 400: Loss = 0.3915
Epoch 500: Loss = 0.3756
Epoch 600: Loss = 0.3631
Epoch 700: Loss = 0.3525
Epoch 800: Loss = 0.3435
```

```
Epoch 900: Loss = 0.3359
Epoch 1000: Loss = 0.3292
Epoch 1100: Loss = 0.3233
Epoch 1200: Loss = 0.3180
Epoch 1300: Loss = 0.3132
Epoch 1400: Loss = 0.3088
Epoch 1500: Loss = 0.3049
Epoch 1600: Loss = 0.3014
Epoch 1700: Loss = 0.2981
Epoch 1800: Loss = 0.2951
Epoch 1900: Loss = 0.2923
Epoch 2000: Loss = 0.2897
Epoch 2100: Loss = 0.2873
Epoch 2200: Loss = 0.2849
Epoch 2300: Loss = 0.2828
Epoch 2400: Loss = 0.2808
Epoch 2500: Loss = 0.2789
Epoch 2600: Loss = 0.2771
Epoch 2700: Loss = 0.2753
Epoch 2800: Loss = 0.2737
Epoch 2900: Loss = 0.2722
Epoch 3000: Loss = 0.2707
Epoch 3100: Loss = 0.2694
Epoch 3200: Loss = 0.2680
Epoch 3300: Loss = 0.2667
Epoch 3400: Loss = 0.2655
Epoch 3500: Loss = 0.2643
Epoch 3600: Loss = 0.2631
Epoch 3700: Loss = 0.2620
Epoch 3800: Loss = 0.2609
Epoch 3900: Loss = 0.2598
Epoch 4000: Loss = 0.2588
Epoch 4100: Loss = 0.2578
Epoch 4200: Loss = 0.2568
Epoch 4300: Loss = 0.2559
Epoch 4400: Loss = 0.2549
Epoch 4500: Loss = 0.2540
Epoch 4600: Loss = 0.2532
Epoch 4700: Loss = 0.2523
Epoch 4800: Loss = 0.2515
Epoch 4900: Loss = 0.2507
Training Accuracy:   91.0751
Validation Accuracy: 90.0923
Testing Accuracy:    90.4000
Training SVM with lr=0.001, epochs=5000, reg_const=0.5…
Epoch 0: Loss = 0.9423
Epoch 100: Loss = 0.5106
Epoch 200: Loss = 0.4445
```

```
Epoch 300: Loss = 0.4132
Epoch 400: Loss = 0.3941
Epoch 500: Loss = 0.3807
Epoch 600: Loss = 0.3702
Epoch 700: Loss = 0.3617
Epoch 800: Loss = 0.3547
Epoch 900: Loss = 0.3488
Epoch 1000: Loss = 0.3437
Epoch 1100: Loss = 0.3393
Epoch 1200: Loss = 0.3355
Epoch 1300: Loss = 0.3321
Epoch 1400: Loss = 0.3292
Epoch 1500: Loss = 0.3266
Epoch 1600: Loss = 0.3243
Epoch 1700: Loss = 0.3222
Epoch 1800: Loss = 0.3203
Epoch 1900: Loss = 0.3185
Epoch 2000: Loss = 0.3169
Epoch 2100: Loss = 0.3155
Epoch 2200: Loss = 0.3142
Epoch 2300: Loss = 0.3129
Epoch 2400: Loss = 0.3118
Epoch 2500: Loss = 0.3108
Epoch 2600: Loss = 0.3098
Epoch 2700: Loss = 0.3089
Epoch 2800: Loss = 0.3082
Epoch 2900: Loss = 0.3074
Epoch 3000: Loss = 0.3068
Epoch 3100: Loss = 0.3062
Epoch 3200: Loss = 0.3056
Epoch 3300: Loss = 0.3050
Epoch 3400: Loss = 0.3045
Epoch 3500: Loss = 0.3041
Epoch 3600: Loss = 0.3037
Epoch 3700: Loss = 0.3033
Epoch 3800: Loss = 0.3029
Epoch 3900: Loss = 0.3025
Epoch 4000: Loss = 0.3022
Epoch 4100: Loss = 0.3019
Epoch 4200: Loss = 0.3017
Epoch 4300: Loss = 0.3014
Epoch 4400: Loss = 0.3012
Epoch 4500: Loss = 0.3009
Epoch 4600: Loss = 0.3007
Epoch 4700: Loss = 0.3005
Epoch 4800: Loss = 0.3003
Epoch 4900: Loss = 0.3002
Training Accuracy:   89.4953
```

```
Validation Accuracy: 88.1231
Testing Accuracy:    88.3077
Training SVM with lr=0.005, epochs=100, reg_const=0.001…
Epoch 0: Loss = 0.8627
Training Accuracy:   86.9307
Validation Accuracy: 85.2923
Testing Accuracy:    85.9077
Training SVM with lr=0.005, epochs=100, reg_const=0.01…
Epoch 0: Loss = 0.8918
Training Accuracy:   86.9717
Validation Accuracy: 85.3538
Testing Accuracy:    86.0308
Training SVM with lr=0.005, epochs=100, reg_const=0.05…
Epoch 0: Loss = 0.9149
Training Accuracy:   86.9922
Validation Accuracy: 85.2923
Testing Accuracy:    86.0923
Training SVM with lr=0.005, epochs=100, reg_const=0.1…
Epoch 0: Loss = 0.9181
Training Accuracy:   86.6639
Validation Accuracy: 84.8615
Testing Accuracy:    85.7231
Training SVM with lr=0.005, epochs=100, reg_const=0.5…
Epoch 0: Loss = 0.9208
Training Accuracy:   86.5408
Validation Accuracy: 84.6769
Testing Accuracy:    85.6615
Training SVM with lr=0.005, epochs=500, reg_const=0.001…
Epoch 0: Loss = 0.9437
Epoch 100: Loss = 0.3694
Epoch 200: Loss = 0.3231
Epoch 300: Loss = 0.2982
Epoch 400: Loss = 0.2819
Training Accuracy:   90.6442
Validation Accuracy: 89.2308
Testing Accuracy:    89.6615
Training SVM with lr=0.005, epochs=500, reg_const=0.01…
Epoch 0: Loss = 0.9637
Epoch 100: Loss = 0.3723
Epoch 200: Loss = 0.3239
Epoch 300: Loss = 0.2980
Epoch 400: Loss = 0.2815
Training Accuracy:   90.5827
Validation Accuracy: 89.1692
Testing Accuracy:    89.4769
Training SVM with lr=0.005, epochs=500, reg_const=0.05…
Epoch 0: Loss = 0.9497
Epoch 100: Loss = 0.3681
```

```
Epoch 200: Loss = 0.3232
Epoch 300: Loss = 0.2990
Epoch 400: Loss = 0.2835
Training Accuracy:   90.6032
Validation Accuracy: 89.2308
Testing Accuracy:    89.2923
Training SVM with lr=0.005, epochs=500, reg_const=0.1…
Epoch 0: Loss = 0.9636
Epoch 100: Loss = 0.3708
Epoch 200: Loss = 0.3269
Epoch 300: Loss = 0.3033
Epoch 400: Loss = 0.2882
Training Accuracy:   90.3365
Validation Accuracy: 89.1077
Testing Accuracy:    88.9846
Training SVM with lr=0.005, epochs=500, reg_const=0.5…
Epoch 0: Loss = 0.9933
Epoch 100: Loss = 0.3798
Epoch 200: Loss = 0.3427
Epoch 300: Loss = 0.3257
Epoch 400: Loss = 0.3162
Training Accuracy:   89.0439
Validation Accuracy: 87.5692
Testing Accuracy:    87.7538
Training SVM with lr=0.005, epochs=1000, reg_const=0.001…
Epoch 0: Loss = 0.9031
Epoch 100: Loss = 0.3654
Epoch 200: Loss = 0.3211
Epoch 300: Loss = 0.2965
Epoch 400: Loss = 0.2804
Epoch 500: Loss = 0.2687
Epoch 600: Loss = 0.2596
Epoch 700: Loss = 0.2520
Epoch 800: Loss = 0.2454
Epoch 900: Loss = 0.2394
Training Accuracy:   91.7932
Validation Accuracy: 90.8308
Testing Accuracy:    91.3231
Training SVM with lr=0.005, epochs=1000, reg_const=0.01…
Epoch 0: Loss = 0.9362
Epoch 100: Loss = 0.3704
Epoch 200: Loss = 0.3229
Epoch 300: Loss = 0.2973
Epoch 400: Loss = 0.2809
Epoch 500: Loss = 0.2692
Epoch 600: Loss = 0.2601
Epoch 700: Loss = 0.2527
Epoch 800: Loss = 0.2461
```

```
Epoch 900: Loss = 0.2403
Training Accuracy:   91.7932
Validation Accuracy: 90.8923
Testing Accuracy:    91.3846
Training SVM with lr=0.005, epochs=1000, reg_const=0.05…
Epoch 0: Loss = 0.9120
Epoch 100: Loss = 0.3691
Epoch 200: Loss = 0.3246
Epoch 300: Loss = 0.3006
Epoch 400: Loss = 0.2851
Epoch 500: Loss = 0.2740
Epoch 600: Loss = 0.2654
Epoch 700: Loss = 0.2583
Epoch 800: Loss = 0.2523
Epoch 900: Loss = 0.2470
Training Accuracy:   91.4649
Validation Accuracy: 90.7077
Testing Accuracy:    91.0769
Training SVM with lr=0.005, epochs=1000, reg_const=0.1…
Epoch 0: Loss = 0.9475
Epoch 100: Loss = 0.3765
Epoch 200: Loss = 0.3299
Epoch 300: Loss = 0.3052
Epoch 400: Loss = 0.2899
Epoch 500: Loss = 0.2790
Epoch 600: Loss = 0.2708
Epoch 700: Loss = 0.2643
Epoch 800: Loss = 0.2587
Epoch 900: Loss = 0.2540
Training Accuracy:   91.0546
Validation Accuracy: 90.0923
Testing Accuracy:    90.4000
Training SVM with lr=0.005, epochs=1000, reg_const=0.5…
Epoch 0: Loss = 0.9242
Epoch 100: Loss = 0.3821
Epoch 200: Loss = 0.3442
Epoch 300: Loss = 0.3266
Epoch 400: Loss = 0.3168
Epoch 500: Loss = 0.3106
Epoch 600: Loss = 0.3066
Epoch 700: Loss = 0.3039
Epoch 800: Loss = 0.3021
Epoch 900: Loss = 0.3008
Training Accuracy:   89.5158
Validation Accuracy: 88.2462
Testing Accuracy:    88.3077
Training SVM with lr=0.005, epochs=2000, reg_const=0.001…
Epoch 0: Loss = 1.0019
```

```
Epoch 100: Loss = 0.3756
Epoch 200: Loss = 0.3259
Epoch 300: Loss = 0.2996
Epoch 400: Loss = 0.2827
Epoch 500: Loss = 0.2707
Epoch 600: Loss = 0.2613
Epoch 700: Loss = 0.2537
Epoch 800: Loss = 0.2470
Epoch 900: Loss = 0.2409
Epoch 1000: Loss = 0.2354
Epoch 1100: Loss = 0.2303
Epoch 1200: Loss = 0.2255
Epoch 1300: Loss = 0.2210
Epoch 1400: Loss = 0.2167
Epoch 1500: Loss = 0.2126
Epoch 1600: Loss = 0.2087
Epoch 1700: Loss = 0.2050
Epoch 1800: Loss = 0.2014
Epoch 1900: Loss = 0.1980
Training Accuracy:   91.5880
Validation Accuracy: 90.8308
Testing Accuracy:    91.2615
Training SVM with lr=0.005, epochs=2000, reg_const=0.01…
Epoch 0: Loss = 0.9329
Epoch 100: Loss = 0.3660
Epoch 200: Loss = 0.3208
Epoch 300: Loss = 0.2964
Epoch 400: Loss = 0.2806
Epoch 500: Loss = 0.2691
Epoch 600: Loss = 0.2601
Epoch 700: Loss = 0.2527
Epoch 800: Loss = 0.2462
Epoch 900: Loss = 0.2403
Epoch 1000: Loss = 0.2350
Epoch 1100: Loss = 0.2301
Epoch 1200: Loss = 0.2255
Epoch 1300: Loss = 0.2212
Epoch 1400: Loss = 0.2171
Epoch 1500: Loss = 0.2132
Epoch 1600: Loss = 0.2096
Epoch 1700: Loss = 0.2060
Epoch 1800: Loss = 0.2027
Epoch 1900: Loss = 0.1995
Training Accuracy:   91.5059
Validation Accuracy: 90.7692
Testing Accuracy:    91.2000
Training SVM with lr=0.005, epochs=2000, reg_const=0.05…
Epoch 0: Loss = 0.9677
```

```
Epoch 100: Loss = 0.3687
Epoch 200: Loss = 0.3235
Epoch 300: Loss = 0.2994
Epoch 400: Loss = 0.2840
Epoch 500: Loss = 0.2730
Epoch 600: Loss = 0.2644
Epoch 700: Loss = 0.2575
Epoch 800: Loss = 0.2516
Epoch 900: Loss = 0.2463
Epoch 1000: Loss = 0.2416
Epoch 1100: Loss = 0.2372
Epoch 1200: Loss = 0.2332
Epoch 1300: Loss = 0.2295
Epoch 1400: Loss = 0.2260
Epoch 1500: Loss = 0.2227
Epoch 1600: Loss = 0.2197
Epoch 1700: Loss = 0.2168
Epoch 1800: Loss = 0.2142
Epoch 1900: Loss = 0.2117
Training Accuracy:   90.9315
Validation Accuracy: 90.2769
Testing Accuracy:    90.5846
Training SVM with lr=0.005, epochs=2000, reg_const=0.1…
Epoch 0: Loss = 0.9519
Epoch 100: Loss = 0.3698
Epoch 200: Loss = 0.3253
Epoch 300: Loss = 0.3018
Epoch 400: Loss = 0.2871
Epoch 500: Loss = 0.2766
Epoch 600: Loss = 0.2687
Epoch 700: Loss = 0.2623
Epoch 800: Loss = 0.2569
Epoch 900: Loss = 0.2523
Epoch 1000: Loss = 0.2483
Epoch 1100: Loss = 0.2446
Epoch 1200: Loss = 0.2413
Epoch 1300: Loss = 0.2383
Epoch 1400: Loss = 0.2356
Epoch 1500: Loss = 0.2331
Epoch 1600: Loss = 0.2307
Epoch 1700: Loss = 0.2286
Epoch 1800: Loss = 0.2268
Epoch 1900: Loss = 0.2250
Training Accuracy:   90.8904
Validation Accuracy: 89.9692
Testing Accuracy:    90.2769
Training SVM with lr=0.005, epochs=2000, reg_const=0.5…
Epoch 0: Loss = 0.9323
```

```
Epoch 100: Loss = 0.3800
Epoch 200: Loss = 0.3432
Epoch 300: Loss = 0.3263
Epoch 400: Loss = 0.3167
Epoch 500: Loss = 0.3106
Epoch 600: Loss = 0.3067
Epoch 700: Loss = 0.3041
Epoch 800: Loss = 0.3022
Epoch 900: Loss = 0.3009
Epoch 1000: Loss = 0.3000
Epoch 1100: Loss = 0.2993
Epoch 1200: Loss = 0.2988
Epoch 1300: Loss = 0.2984
Epoch 1400: Loss = 0.2981
Epoch 1500: Loss = 0.2978
Epoch 1600: Loss = 0.2977
Epoch 1700: Loss = 0.2975
Epoch 1800: Loss = 0.2974
Epoch 1900: Loss = 0.2973
Training Accuracy:   89.6184
Validation Accuracy: 88.1846
Testing Accuracy:    88.4308
Training SVM with lr=0.005, epochs=5000, reg_const=0.001…
Epoch 0: Loss = 0.9744
Epoch 100: Loss = 0.3720
Epoch 200: Loss = 0.3247
Epoch 300: Loss = 0.2988
Epoch 400: Loss = 0.2820
Epoch 500: Loss = 0.2698
Epoch 600: Loss = 0.2604
Epoch 700: Loss = 0.2527
Epoch 800: Loss = 0.2460
Epoch 900: Loss = 0.2399
Epoch 1000: Loss = 0.2344
Epoch 1100: Loss = 0.2293
Epoch 1200: Loss = 0.2246
Epoch 1300: Loss = 0.2200
Epoch 1400: Loss = 0.2158
Epoch 1500: Loss = 0.2117
Epoch 1600: Loss = 0.2078
Epoch 1700: Loss = 0.2041
Epoch 1800: Loss = 0.2005
Epoch 1900: Loss = 0.1971
Epoch 2000: Loss = 0.1939
Epoch 2100: Loss = 0.1907
Epoch 2200: Loss = 0.1877
Epoch 2300: Loss = 0.1850
Epoch 2400: Loss = 0.1823
```

```
Epoch 2500: Loss = 0.1798
Epoch 2600: Loss = 0.1774
Epoch 2700: Loss = 0.1751
Epoch 2800: Loss = 0.1730
Epoch 2900: Loss = 0.1710
Epoch 3000: Loss = 0.1690
Epoch 3100: Loss = 0.1672
Epoch 3200: Loss = 0.1654
Epoch 3300: Loss = 0.1637
Epoch 3400: Loss = 0.1621
Epoch 3500: Loss = 0.1606
Epoch 3600: Loss = 0.1590
Epoch 3700: Loss = 0.1576
Epoch 3800: Loss = 0.1562
Epoch 3900: Loss = 0.1548
Epoch 4000: Loss = 0.1536
Epoch 4100: Loss = 0.1523
Epoch 4200: Loss = 0.1512
Epoch 4300: Loss = 0.1501
Epoch 4400: Loss = 0.1491
Epoch 4500: Loss = 0.1504
Epoch 4600: Loss = 0.1484
Epoch 4700: Loss = 0.1476
Epoch 4800: Loss = 0.1466
Epoch 4900: Loss = 0.1460
Training Accuracy:   94.5630
Validation Accuracy: 93.9692
Testing Accuracy:    94.2154
Training SVM with lr=0.005, epochs=5000, reg_const=0.01…
Epoch 0: Loss = 0.9103
Epoch 100: Loss = 0.3699
Epoch 200: Loss = 0.3238
Epoch 300: Loss = 0.2983
Epoch 400: Loss = 0.2819
Epoch 500: Loss = 0.2700
Epoch 600: Loss = 0.2608
Epoch 700: Loss = 0.2533
Epoch 800: Loss = 0.2467
Epoch 900: Loss = 0.2408
Epoch 1000: Loss = 0.2355
Epoch 1100: Loss = 0.2306
Epoch 1200: Loss = 0.2260
Epoch 1300: Loss = 0.2217
Epoch 1400: Loss = 0.2176
Epoch 1500: Loss = 0.2138
Epoch 1600: Loss = 0.2101
Epoch 1700: Loss = 0.2067
Epoch 1800: Loss = 0.2033
```

```
Epoch 1900: Loss = 0.2001
Epoch 2000: Loss = 0.1971
Epoch 2100: Loss = 0.1942
Epoch 2200: Loss = 0.1914
Epoch 2300: Loss = 0.1888
Epoch 2400: Loss = 0.1863
Epoch 2500: Loss = 0.1839
Epoch 2600: Loss = 0.1817
Epoch 2700: Loss = 0.1796
Epoch 2800: Loss = 0.1776
Epoch 2900: Loss = 0.1757
Epoch 3000: Loss = 0.1739
Epoch 3100: Loss = 0.1723
Epoch 3200: Loss = 0.1707
Epoch 3300: Loss = 0.1692
Epoch 3400: Loss = 0.1677
Epoch 3500: Loss = 0.1663
Epoch 3600: Loss = 0.1651
Epoch 3700: Loss = 0.1639
Epoch 3800: Loss = 0.1625
Epoch 3900: Loss = 0.1614
Epoch 4000: Loss = 0.1600
Epoch 4100: Loss = 0.1590
Epoch 4200: Loss = 0.1579
Epoch 4300: Loss = 0.1569
Epoch 4400: Loss = 0.1559
Epoch 4500: Loss = 0.1549
Epoch 4600: Loss = 0.1540
Epoch 4700: Loss = 0.1531
Epoch 4800: Loss = 0.1528
Epoch 4900: Loss = 0.1520
Training Accuracy:   94.5425
Validation Accuracy: 94.0308
Testing Accuracy:    94.2154
Training SVM with lr=0.005, epochs=5000, reg_const=0.05…
Epoch 0: Loss = 0.8894
Epoch 100: Loss = 0.3726
Epoch 200: Loss = 0.3274
Epoch 300: Loss = 0.3028
Epoch 400: Loss = 0.2868
Epoch 500: Loss = 0.2753
Epoch 600: Loss = 0.2665
Epoch 700: Loss = 0.2593
Epoch 800: Loss = 0.2533
Epoch 900: Loss = 0.2479
Epoch 1000: Loss = 0.2430
Epoch 1100: Loss = 0.2386
Epoch 1200: Loss = 0.2345
```

```
Epoch 1300: Loss = 0.2307
Epoch 1400: Loss = 0.2271
Epoch 1500: Loss = 0.2238
Epoch 1600: Loss = 0.2207
Epoch 1700: Loss = 0.2178
Epoch 1800: Loss = 0.2151
Epoch 1900: Loss = 0.2125
Epoch 2000: Loss = 0.2102
Epoch 2100: Loss = 0.2080
Epoch 2200: Loss = 0.2059
Epoch 2300: Loss = 0.2039
Epoch 2400: Loss = 0.2021
Epoch 2500: Loss = 0.2004
Epoch 2600: Loss = 0.1988
Epoch 2700: Loss = 0.1973
Epoch 2800: Loss = 0.1959
Epoch 2900: Loss = 0.1946
Epoch 3000: Loss = 0.1933
Epoch 3100: Loss = 0.1921
Epoch 3200: Loss = 0.1909
Epoch 3300: Loss = 0.1900
Epoch 3400: Loss = 0.1888
Epoch 3500: Loss = 0.1881
Epoch 3600: Loss = 0.1872
Epoch 3700: Loss = 0.1863
Epoch 3800: Loss = 0.1854
Epoch 3900: Loss = 0.1847
Epoch 4000: Loss = 0.1841
Epoch 4100: Loss = 0.1835
Epoch 4200: Loss = 0.1830
Epoch 4300: Loss = 0.1824
Epoch 4400: Loss = 0.1818
Epoch 4500: Loss = 0.1814
Epoch 4600: Loss = 0.1825
Epoch 4700: Loss = 0.1821
Epoch 4800: Loss = 0.1817
Epoch 4900: Loss = 0.1814
Training Accuracy:   93.5577
Validation Accuracy: 92.6154
Testing Accuracy:    92.8000
Training SVM with lr=0.005, epochs=5000, reg_const=0.1…
Epoch 0: Loss = 0.9463
Epoch 100: Loss = 0.3675
Epoch 200: Loss = 0.3243
Epoch 300: Loss = 0.3015
Epoch 400: Loss = 0.2870
Epoch 500: Loss = 0.2767
Epoch 600: Loss = 0.2689
```

```
Epoch 700: Loss = 0.2627
Epoch 800: Loss = 0.2573
Epoch 900: Loss = 0.2528
Epoch 1000: Loss = 0.2487
Epoch 1100: Loss = 0.2451
Epoch 1200: Loss = 0.2418
Epoch 1300: Loss = 0.2388
Epoch 1400: Loss = 0.2360
Epoch 1500: Loss = 0.2335
Epoch 1600: Loss = 0.2312
Epoch 1700: Loss = 0.2290
Epoch 1800: Loss = 0.2272
Epoch 1900: Loss = 0.2253
Epoch 2000: Loss = 0.2236
Epoch 2100: Loss = 0.2222
Epoch 2200: Loss = 0.2205
Epoch 2300: Loss = 0.2194
Epoch 2400: Loss = 0.2182
Epoch 2500: Loss = 0.2170
Epoch 2600: Loss = 0.2162
Epoch 2700: Loss = 0.2152
Epoch 2800: Loss = 0.2164
Epoch 2900: Loss = 0.2151
Epoch 3000: Loss = 0.2145
Epoch 3100: Loss = 0.2122
Epoch 3200: Loss = 0.2117
Epoch 3300: Loss = 0.2128
Epoch 3400: Loss = 0.2124
Epoch 3500: Loss = 0.2092
Epoch 3600: Loss = 0.2113
Epoch 3700: Loss = 0.2082
Epoch 3800: Loss = 0.2113
Epoch 3900: Loss = 0.2101
Epoch 4000: Loss = 0.2097
Epoch 4100: Loss = 0.2068
Epoch 4200: Loss = 0.2101
Epoch 4300: Loss = 0.2088
Epoch 4400: Loss = 0.2080
Epoch 4500: Loss = 0.2061
Epoch 4600: Loss = 0.2086
Epoch 4700: Loss = 0.2078
Epoch 4800: Loss = 0.2080
Epoch 4900: Loss = 0.2078
Training Accuracy:   91.0135
Validation Accuracy: 90.3385
Testing Accuracy:    90.7077
Training SVM with lr=0.005, epochs=5000, reg_const=0.5…
Epoch 0: Loss = 0.9359
```

```
Epoch 100: Loss = 0.3821
Epoch 200: Loss = 0.3445
Epoch 300: Loss = 0.3268
Epoch 400: Loss = 0.3170
Epoch 500: Loss = 0.3108
Epoch 600: Loss = 0.3067
Epoch 700: Loss = 0.3040
Epoch 800: Loss = 0.3021
Epoch 900: Loss = 0.3008
Epoch 1000: Loss = 0.2999
Epoch 1100: Loss = 0.2992
Epoch 1200: Loss = 0.2987
Epoch 1300: Loss = 0.2983
Epoch 1400: Loss = 0.2980
Epoch 1500: Loss = 0.2978
Epoch 1600: Loss = 0.2976
Epoch 1700: Loss = 0.2975
Epoch 1800: Loss = 0.2974
Epoch 1900: Loss = 0.2973
Epoch 2000: Loss = 0.2973
Epoch 2100: Loss = 0.2972
Epoch 2200: Loss = 0.2972
Epoch 2300: Loss = 0.2971
Epoch 2400: Loss = 0.2971
Epoch 2500: Loss = 0.2971
Epoch 2600: Loss = 0.2971
Epoch 2700: Loss = 0.2971
Epoch 2800: Loss = 0.2971
Epoch 2900: Loss = 0.2971
Epoch 3000: Loss = 0.2971
Epoch 3100: Loss = 0.2971
Epoch 3200: Loss = 0.2971
Epoch 3300: Loss = 0.2971
Epoch 3400: Loss = 0.2971
Epoch 3500: Loss = 0.2970
Epoch 3600: Loss = 0.2970
Epoch 3700: Loss = 0.2970
Epoch 3800: Loss = 0.2970
Epoch 3900: Loss = 0.2970
Epoch 4000: Loss = 0.2970
Epoch 4100: Loss = 0.2970
Epoch 4200: Loss = 0.2970
Epoch 4300: Loss = 0.2970
Epoch 4400: Loss = 0.2970
Epoch 4500: Loss = 0.2970
Epoch 4600: Loss = 0.2970
Epoch 4700: Loss = 0.2970
Epoch 4800: Loss = 0.2970
```

```
Epoch 4900: Loss = 0.2970
Training Accuracy:   89.6389
Validation Accuracy: 88.1846
Testing Accuracy:    88.4923
Training SVM with lr=0.01, epochs=100, reg_const=0.001…
Epoch 0: Loss = 0.8649
Training Accuracy:   87.8744
Validation Accuracy: 86.3385
Testing Accuracy:    86.5231
Training SVM with lr=0.01, epochs=100, reg_const=0.01…
Epoch 0: Loss = 0.9045
Training Accuracy:   87.8744
Validation Accuracy: 86.4000
Testing Accuracy:    86.5231
Training SVM with lr=0.01, epochs=100, reg_const=0.05…
Epoch 0: Loss = 0.8332
Training Accuracy:   87.9360
Validation Accuracy: 86.3385
Testing Accuracy:    86.4000
Training SVM with lr=0.01, epochs=100, reg_const=0.1…
Epoch 0: Loss = 0.8282
Training Accuracy:   87.4641
Validation Accuracy: 86.4000
Testing Accuracy:    86.2154
Training SVM with lr=0.01, epochs=100, reg_const=0.5…
Epoch 0: Loss = 0.8511
Training Accuracy:   86.3151
Validation Accuracy: 85.7846
Testing Accuracy:    84.6154
Training SVM with lr=0.01, epochs=500, reg_const=0.001…
Epoch 0: Loss = 0.9199
Epoch 100: Loss = 0.3420
Epoch 200: Loss = 0.3001
Epoch 300: Loss = 0.2772
Epoch 400: Loss = 0.2596
Training Accuracy:   91.3008
Validation Accuracy: 90.5846
Testing Accuracy:    90.7692
Training SVM with lr=0.01, epochs=500, reg_const=0.01…
Epoch 0: Loss = 0.8501
Epoch 100: Loss = 0.3402
Epoch 200: Loss = 0.2997
Epoch 300: Loss = 0.2748
Epoch 400: Loss = 0.2596
Training Accuracy:   91.2187
Validation Accuracy: 90.4615
Testing Accuracy:    90.7077
Training SVM with lr=0.01, epochs=500, reg_const=0.05…
```

```
Epoch 0: Loss = 0.8254
Epoch 100: Loss = 0.3427
Epoch 200: Loss = 0.3031
Epoch 300: Loss = 0.2844
Epoch 400: Loss = 0.2699
Training Accuracy:   91.5880
Validation Accuracy: 91.2000
Testing Accuracy:    91.2000
Training SVM with lr=0.01, epochs=500, reg_const=0.1…
Epoch 0: Loss = 0.8972
Epoch 100: Loss = 0.3461
Epoch 200: Loss = 0.3111
Epoch 300: Loss = 0.2918
Epoch 400: Loss = 0.2775
Training Accuracy:   91.2392
Validation Accuracy: 90.7077
Testing Accuracy:    90.7692
Training SVM with lr=0.01, epochs=500, reg_const=0.5…
Epoch 0: Loss = 0.8152
Epoch 100: Loss = 0.3691
Epoch 200: Loss = 0.3508
Epoch 300: Loss = 0.3466
Epoch 400: Loss = 0.3348
Training Accuracy:   87.9565
Validation Accuracy: 87.3846
Testing Accuracy:    86.8923
Training SVM with lr=0.01, epochs=1000, reg_const=0.001…
Epoch 0: Loss = 0.8431
Epoch 100: Loss = 0.3399
Epoch 200: Loss = 0.2960
Epoch 300: Loss = 0.2754
Epoch 400: Loss = 0.2584
Epoch 500: Loss = 0.2459
Epoch 600: Loss = 0.2353
Epoch 700: Loss = 0.2270
Epoch 800: Loss = 0.2198
Epoch 900: Loss = 0.2114
Training Accuracy:   91.1366
Validation Accuracy: 90.7077
Testing Accuracy:    90.7077
Training SVM with lr=0.01, epochs=1000, reg_const=0.01…
Epoch 0: Loss = 0.8554
Epoch 100: Loss = 0.3431
Epoch 200: Loss = 0.2976
Epoch 300: Loss = 0.2766
Epoch 400: Loss = 0.2606
Epoch 500: Loss = 0.2473
Epoch 600: Loss = 0.2374
```

```
Epoch 700: Loss = 0.2297
Epoch 800: Loss = 0.2226
Epoch 900: Loss = 0.2169
Training Accuracy:   91.0956
Validation Accuracy: 90.5846
Testing Accuracy:    90.5846
Training SVM with lr=0.01, epochs=1000, reg_const=0.05…
Epoch 0: Loss = 0.8941
Epoch 100: Loss = 0.3439
Epoch 200: Loss = 0.3071
Epoch 300: Loss = 0.2849
Epoch 400: Loss = 0.2694
Epoch 500: Loss = 0.2573
Epoch 600: Loss = 0.2490
Epoch 700: Loss = 0.2422
Epoch 800: Loss = 0.2365
Epoch 900: Loss = 0.2319
Training Accuracy:   90.8699
Validation Accuracy: 89.9077
Testing Accuracy:    90.3385
Training SVM with lr=0.01, epochs=1000, reg_const=0.1…
Epoch 0: Loss = 0.8485
Epoch 100: Loss = 0.3462
Epoch 200: Loss = 0.3117
Epoch 300: Loss = 0.2926
Epoch 400: Loss = 0.2816
Epoch 500: Loss = 0.2708
Epoch 600: Loss = 0.2630
Epoch 700: Loss = 0.2600
Epoch 800: Loss = 0.2554
Epoch 900: Loss = 0.2527
Training Accuracy:   91.7316
Validation Accuracy: 91.0154
Testing Accuracy:    91.0769
Training SVM with lr=0.01, epochs=1000, reg_const=0.5…
Epoch 0: Loss = 0.8372
Epoch 100: Loss = 0.3744
Epoch 200: Loss = 0.3560
Epoch 300: Loss = 0.3493
Epoch 400: Loss = 0.3506
Epoch 500: Loss = 0.3376
Epoch 600: Loss = 0.3357
Epoch 700: Loss = 0.3429
Epoch 800: Loss = 0.3363
Epoch 900: Loss = 0.3403
Training Accuracy:   88.4079
Validation Accuracy: 87.0769
Testing Accuracy:    87.0154
```

```
Training SVM with lr=0.01, epochs=2000, reg_const=0.001…
Epoch 0: Loss = 0.8610
Epoch 100: Loss = 0.3425
Epoch 200: Loss = 0.3000
Epoch 300: Loss = 0.2784
Epoch 400: Loss = 0.2623
Epoch 500: Loss = 0.2502
Epoch 600: Loss = 0.2376
Epoch 700: Loss = 0.2305
Epoch 800: Loss = 0.2224
Epoch 900: Loss = 0.2170
Epoch 1000: Loss = 0.2094
Epoch 1100: Loss = 0.2047
Epoch 1200: Loss = 0.2013
Epoch 1300: Loss = 0.1941
Epoch 1400: Loss = 0.1963
Epoch 1500: Loss = 0.1847
Epoch 1600: Loss = 0.1839
Epoch 1700: Loss = 0.1836
Epoch 1800: Loss = 0.1783
Epoch 1900: Loss = 0.1763
Training Accuracy:   95.0349
Validation Accuracy: 94.3385
Testing Accuracy:    94.5231
Training SVM with lr=0.01, epochs=2000, reg_const=0.01…
Epoch 0: Loss = 0.8587
Epoch 100: Loss = 0.3395
Epoch 200: Loss = 0.2994
Epoch 300: Loss = 0.2758
Epoch 400: Loss = 0.2624
Epoch 500: Loss = 0.2493
Epoch 600: Loss = 0.2398
Epoch 700: Loss = 0.2315
Epoch 800: Loss = 0.2244
Epoch 900: Loss = 0.2190
Epoch 1000: Loss = 0.2129
Epoch 1100: Loss = 0.2062
Epoch 1200: Loss = 0.2006
Epoch 1300: Loss = 0.2003
Epoch 1400: Loss = 0.1978
Epoch 1500: Loss = 0.1959
Epoch 1600: Loss = 0.1893
Epoch 1700: Loss = 0.1884
Epoch 1800: Loss = 0.1859
Epoch 1900: Loss = 0.1893
Training Accuracy:   94.9938
Validation Accuracy: 94.2154
Testing Accuracy:    94.4615
```

```
Training SVM with lr=0.01, epochs=2000, reg_const=0.05…
Epoch 0: Loss = 0.8676
Epoch 100: Loss = 0.3459
Epoch 200: Loss = 0.3070
Epoch 300: Loss = 0.2851
Epoch 400: Loss = 0.2693
Epoch 500: Loss = 0.2569
Epoch 600: Loss = 0.2494
Epoch 700: Loss = 0.2424
Epoch 800: Loss = 0.2342
Epoch 900: Loss = 0.2323
Epoch 1000: Loss = 0.2259
Epoch 1100: Loss = 0.2219
Epoch 1200: Loss = 0.2208
Epoch 1300: Loss = 0.2173
Epoch 1400: Loss = 0.2170
Epoch 1500: Loss = 0.2084
Epoch 1600: Loss = 0.2038
Epoch 1700: Loss = 0.2189
Epoch 1800: Loss = 0.2001
Epoch 1900: Loss = 0.1989
Training Accuracy:   90.6237
Validation Accuracy: 89.6000
Testing Accuracy:    89.8462
Training SVM with lr=0.01, epochs=2000, reg_const=0.1…
Epoch 0: Loss = 0.8525
Epoch 100: Loss = 0.3461
Epoch 200: Loss = 0.3105
Epoch 300: Loss = 0.2903
Epoch 400: Loss = 0.2806
Epoch 500: Loss = 0.2698
Epoch 600: Loss = 0.2629
Epoch 700: Loss = 0.2609
Epoch 800: Loss = 0.2572
Epoch 900: Loss = 0.2528
Epoch 1000: Loss = 0.2514
Epoch 1100: Loss = 0.2461
Epoch 1200: Loss = 0.2475
Epoch 1300: Loss = 0.2422
Epoch 1400: Loss = 0.2428
Epoch 1500: Loss = 0.2423
Epoch 1600: Loss = 0.2429
Epoch 1700: Loss = 0.2393
Epoch 1800: Loss = 0.2366
Epoch 1900: Loss = 0.2393
Training Accuracy:   92.2856
Validation Accuracy: 91.5692
Testing Accuracy:    91.6308
```

```
Training SVM with lr=0.01, epochs=2000, reg_const=0.5…
Epoch 0: Loss = 0.8567
Epoch 100: Loss = 0.3725
Epoch 200: Loss = 0.3515
Epoch 300: Loss = 0.3382
Epoch 400: Loss = 0.3418
Epoch 500: Loss = 0.3444
Epoch 600: Loss = 0.3290
Epoch 700: Loss = 0.3387
Epoch 800: Loss = 0.3378
Epoch 900: Loss = 0.3349
Epoch 1000: Loss = 0.3460
Epoch 1100: Loss = 0.3300
Epoch 1200: Loss = 0.3381
Epoch 1300: Loss = 0.3361
Epoch 1400: Loss = 0.3303
Epoch 1500: Loss = 0.3424
Epoch 1600: Loss = 0.3304
Epoch 1700: Loss = 0.3372
Epoch 1800: Loss = 0.3349
Epoch 1900: Loss = 0.3423
Training Accuracy:   87.9975
Validation Accuracy: 87.2000
Testing Accuracy:    86.9538
Training SVM with lr=0.01, epochs=5000, reg_const=0.001…
Epoch 0: Loss = 0.8861
Epoch 100: Loss = 0.3412
Epoch 200: Loss = 0.3009
Epoch 300: Loss = 0.2779
Epoch 400: Loss = 0.2623
Epoch 500: Loss = 0.2495
Epoch 600: Loss = 0.2384
Epoch 700: Loss = 0.2301
Epoch 800: Loss = 0.2222
Epoch 900: Loss = 0.2144
Epoch 1000: Loss = 0.2086
Epoch 1100: Loss = 0.2039
Epoch 1200: Loss = 0.2021
Epoch 1300: Loss = 0.1963
Epoch 1400: Loss = 0.1926
Epoch 1500: Loss = 0.1909
Epoch 1600: Loss = 0.1883
Epoch 1700: Loss = 0.1823
Epoch 1800: Loss = 0.1780
Epoch 1900: Loss = 0.1752
Epoch 2000: Loss = 0.1756
Epoch 2100: Loss = 0.1743
Epoch 2200: Loss = 0.1706
```

```
Epoch 2300: Loss = 0.1674
Epoch 2400: Loss = 0.1697
Epoch 2500: Loss = 0.1634
Epoch 2600: Loss = 0.1624
Epoch 2700: Loss = 0.1583
Epoch 2800: Loss = 0.1627
Epoch 2900: Loss = 0.1574
Epoch 3000: Loss = 0.1561
Epoch 3100: Loss = 0.1562
Epoch 3200: Loss = 0.1557
Epoch 3300: Loss = 0.1553
Epoch 3400: Loss = 0.1542
Epoch 3500: Loss = 0.1524
Epoch 3600: Loss = 0.1532
Epoch 3700: Loss = 0.1518
Epoch 3800: Loss = 0.1490
Epoch 3900: Loss = 0.1508
Epoch 4000: Loss = 0.1465
Epoch 4100: Loss = 0.1454
Epoch 4200: Loss = 0.1499
Epoch 4300: Loss = 0.1486
Epoch 4400: Loss = 0.1441
Epoch 4500: Loss = 0.1389
Epoch 4600: Loss = 0.2123
Epoch 4700: Loss = 0.1360
Epoch 4800: Loss = 0.1553
Epoch 4900: Loss = 0.1816
Training Accuracy:   94.6040
Validation Accuracy: 94.0308
Testing Accuracy:    94.2154
Training SVM with lr=0.01, epochs=5000, reg_const=0.01…
Epoch 0: Loss = 0.8277
Epoch 100: Loss = 0.3386
Epoch 200: Loss = 0.3012
Epoch 300: Loss = 0.2775
Epoch 400: Loss = 0.2602
Epoch 500: Loss = 0.2468
Epoch 600: Loss = 0.2383
Epoch 700: Loss = 0.2297
Epoch 800: Loss = 0.2228
Epoch 900: Loss = 0.2170
Epoch 1000: Loss = 0.2102
Epoch 1100: Loss = 0.2056
Epoch 1200: Loss = 0.2002
Epoch 1300: Loss = 0.1937
Epoch 1400: Loss = 0.1925
Epoch 1500: Loss = 0.1876
Epoch 1600: Loss = 0.1855
```

```
Epoch 1700: Loss = 0.1824
Epoch 1800: Loss = 0.1801
Epoch 1900: Loss = 0.1746
Epoch 2000: Loss = 0.1759
Epoch 2100: Loss = 0.1724
Epoch 2200: Loss = 0.1701
Epoch 2300: Loss = 0.1678
Epoch 2400: Loss = 0.1645
Epoch 2500: Loss = 0.1672
Epoch 2600: Loss = 0.1603
Epoch 2700: Loss = 0.1633
Epoch 2800: Loss = 0.1630
Epoch 2900: Loss = 0.1609
Epoch 3000: Loss = 0.1602
Epoch 3100: Loss = 0.1563
Epoch 3200: Loss = 0.1542
Epoch 3300: Loss = 0.1570
Epoch 3400: Loss = 0.1568
Epoch 3500: Loss = 0.1525
Epoch 3600: Loss = 0.1560
Epoch 3700: Loss = 0.1506
Epoch 3800: Loss = 0.1554
Epoch 3900: Loss = 0.1483
Epoch 4000: Loss = 0.1495
Epoch 4100: Loss = 0.1543
Epoch 4200: Loss = 0.1524
Epoch 4300: Loss = 0.1465
Epoch 4400: Loss = 0.1460
Epoch 4500: Loss = 0.1463
Epoch 4600: Loss = 0.1506
Epoch 4700: Loss = 0.1505
Epoch 4800: Loss = 0.1469
Epoch 4900: Loss = 0.1451
Training Accuracy:   94.6245
Validation Accuracy: 93.8462
Testing Accuracy:    94.0923
Training SVM with lr=0.01, epochs=5000, reg_const=0.05…
Epoch 0: Loss = 0.8447
Epoch 100: Loss = 0.3434
Epoch 200: Loss = 0.3048
Epoch 300: Loss = 0.2838
Epoch 400: Loss = 0.2683
Epoch 500: Loss = 0.2575
Epoch 600: Loss = 0.2484
Epoch 700: Loss = 0.2415
Epoch 800: Loss = 0.2355
Epoch 900: Loss = 0.2301
Epoch 1000: Loss = 0.2314
```

```
Epoch 1100: Loss = 0.2253
Epoch 1200: Loss = 0.2184
Epoch 1300: Loss = 0.2154
Epoch 1400: Loss = 0.2124
Epoch 1500: Loss = 0.2143
Epoch 1600: Loss = 0.2100
Epoch 1700: Loss = 0.1995
Epoch 1800: Loss = 0.2049
Epoch 1900: Loss = 0.2094
Epoch 2000: Loss = 0.1945
Epoch 2100: Loss = 0.1961
Epoch 2200: Loss = 0.2128
Epoch 2300: Loss = 0.2040
Epoch 2400: Loss = 0.1988
Epoch 2500: Loss = 0.1923
Epoch 2600: Loss = 0.1885
Epoch 2700: Loss = 0.1873
Epoch 2800: Loss = 0.1891
Epoch 2900: Loss = 0.1954
Epoch 3000: Loss = 0.2029
Epoch 3100: Loss = 0.2002
Epoch 3200: Loss = 0.1899
Epoch 3300: Loss = 0.1853
Epoch 3400: Loss = 0.1872
Epoch 3500: Loss = 0.1944
Epoch 3600: Loss = 0.2001
Epoch 3700: Loss = 0.1968
Epoch 3800: Loss = 0.1894
Epoch 3900: Loss = 0.1854
Epoch 4000: Loss = 0.1856
Epoch 4100: Loss = 0.1909
Epoch 4200: Loss = 0.1898
Epoch 4300: Loss = 0.1892
Epoch 4400: Loss = 0.1890
Epoch 4500: Loss = 0.1918
Epoch 4600: Loss = 0.1911
Epoch 4700: Loss = 0.1922
Epoch 4800: Loss = 0.1898
Epoch 4900: Loss = 0.1898
Training Accuracy:   91.9163
Validation Accuracy: 91.5692
Testing Accuracy:    91.2615
Training SVM with lr=0.01, epochs=5000, reg_const=0.1…
Epoch 0: Loss = 0.9089
Epoch 100: Loss = 0.3513
Epoch 200: Loss = 0.3141
Epoch 300: Loss = 0.2948
Epoch 400: Loss = 0.2833
```

```
Epoch 500: Loss = 0.2721
Epoch 600: Loss = 0.2649
Epoch 700: Loss = 0.2572
Epoch 800: Loss = 0.2521
Epoch 900: Loss = 0.2485
Epoch 1000: Loss = 0.2462
Epoch 1100: Loss = 0.2436
Epoch 1200: Loss = 0.2375
Epoch 1300: Loss = 0.2394
Epoch 1400: Loss = 0.2368
Epoch 1500: Loss = 0.2366
Epoch 1600: Loss = 0.2368
Epoch 1700: Loss = 0.2353
Epoch 1800: Loss = 0.2355
Epoch 1900: Loss = 0.2337
Epoch 2000: Loss = 0.2310
Epoch 2100: Loss = 0.2363
Epoch 2200: Loss = 0.2354
Epoch 2300: Loss = 0.2274
Epoch 2400: Loss = 0.2250
Epoch 2500: Loss = 0.2346
Epoch 2600: Loss = 0.2286
Epoch 2700: Loss = 0.2249
Epoch 2800: Loss = 0.2331
Epoch 2900: Loss = 0.2218
Epoch 3000: Loss = 0.2359
Epoch 3100: Loss = 0.2214
Epoch 3200: Loss = 0.2350
Epoch 3300: Loss = 0.2236
Epoch 3400: Loss = 0.2312
Epoch 3500: Loss = 0.2269
Epoch 3600: Loss = 0.2240
Epoch 3700: Loss = 0.2317
Epoch 3800: Loss = 0.2216
Epoch 3900: Loss = 0.2356
Epoch 4000: Loss = 0.2208
Epoch 4100: Loss = 0.2342
Epoch 4200: Loss = 0.2226
Epoch 4300: Loss = 0.2307
Epoch 4400: Loss = 0.2262
Epoch 4500: Loss = 0.2234
Epoch 4600: Loss = 0.2317
Epoch 4700: Loss = 0.2212
Epoch 4800: Loss = 0.2355
Epoch 4900: Loss = 0.2207
Training Accuracy:   90.2749
Validation Accuracy: 89.3538
Testing Accuracy:    89.6000
```

```
Training SVM with lr=0.01, epochs=5000, reg_const=0.5…
Epoch 0: Loss = 0.9408
Epoch 100: Loss = 0.3723
Epoch 200: Loss = 0.3492
Epoch 300: Loss = 0.3491
Epoch 400: Loss = 0.3388
Epoch 500: Loss = 0.3300
Epoch 600: Loss = 0.3387
Epoch 700: Loss = 0.3298
Epoch 800: Loss = 0.3455
Epoch 900: Loss = 0.3302
Epoch 1000: Loss = 0.3446
Epoch 1100: Loss = 0.3303
Epoch 1200: Loss = 0.3457
Epoch 1300: Loss = 0.3300
Epoch 1400: Loss = 0.3455
Epoch 1500: Loss = 0.3299
Epoch 1600: Loss = 0.3448
Epoch 1700: Loss = 0.3359
Epoch 1800: Loss = 0.3348
Epoch 1900: Loss = 0.3424
Epoch 2000: Loss = 0.3299
Epoch 2100: Loss = 0.3425
Epoch 2200: Loss = 0.3346
Epoch 2300: Loss = 0.3423
Epoch 2400: Loss = 0.3320
Epoch 2500: Loss = 0.3449
Epoch 2600: Loss = 0.3293
Epoch 2700: Loss = 0.3447
Epoch 2800: Loss = 0.3311
Epoch 2900: Loss = 0.3405
Epoch 3000: Loss = 0.3300
Epoch 3100: Loss = 0.3436
Epoch 3200: Loss = 0.3312
Epoch 3300: Loss = 0.3379
Epoch 3400: Loss = 0.3369
Epoch 3500: Loss = 0.3318
Epoch 3600: Loss = 0.3401
Epoch 3700: Loss = 0.3302
Epoch 3800: Loss = 0.3455
Epoch 3900: Loss = 0.3298
Epoch 4000: Loss = 0.3447
Epoch 4100: Loss = 0.3301
Epoch 4200: Loss = 0.3403
Epoch 4300: Loss = 0.3375
Epoch 4400: Loss = 0.3324
Epoch 4500: Loss = 0.3443
Epoch 4600: Loss = 0.3297
```

```
Epoch 4700: Loss = 0.3382
Epoch 4800: Loss = 0.3364
Epoch 4900: Loss = 0.3323
Training Accuracy:   87.9565
Validation Accuracy: 87.0769
Testing Accuracy:    86.9538
Training SVM with lr=0.05, epochs=100, reg_const=0.001…
Epoch 0: Loss = 1.1306
Training Accuracy:   78.8059
Validation Accuracy: 77.5385
Testing Accuracy:    76.8615
Training SVM with lr=0.05, epochs=100, reg_const=0.01…
Epoch 0: Loss = 1.1950
Training Accuracy:   73.1022
Validation Accuracy: 72.7385
Testing Accuracy:    70.8923
Training SVM with lr=0.05, epochs=100, reg_const=0.05…
Epoch 0: Loss = 1.0200
Training Accuracy:   84.3865
Validation Accuracy: 84.3692
Testing Accuracy:    82.6462
Training SVM with lr=0.05, epochs=100, reg_const=0.1…
Epoch 0: Loss = 1.1100
Training Accuracy:   69.9015
Validation Accuracy: 69.7231
Testing Accuracy:    67.6923
Training SVM with lr=0.05, epochs=100, reg_const=0.5…
Epoch 0: Loss = 1.1272
Training Accuracy:   77.7390
Validation Accuracy: 75.7538
Testing Accuracy:    75.5077
Training SVM with lr=0.05, epochs=500, reg_const=0.001…
Epoch 0: Loss = 0.9178
Epoch 100: Loss = 0.7613
Epoch 200: Loss = 0.6087
Epoch 300: Loss = 0.2827
Epoch 400: Loss = 0.2623
Training Accuracy:   92.3266
Validation Accuracy: 91.6923
Testing Accuracy:    91.8154
Training SVM with lr=0.05, epochs=500, reg_const=0.01…
Epoch 0: Loss = 0.9662
Epoch 100: Loss = 0.4160
Epoch 200: Loss = 0.3520
Epoch 300: Loss = 0.3187
Epoch 400: Loss = 0.3382
Training Accuracy:   92.2651
Validation Accuracy: 91.6923
```

```
Testing Accuracy:    91.3231
Training SVM with lr=0.05, epochs=500, reg_const=0.05…
Epoch 0: Loss = 1.1474
Epoch 100: Loss = 0.4238
Epoch 200: Loss = 0.4683
Epoch 300: Loss = 0.4217
Epoch 400: Loss = 0.3263
Training Accuracy:   79.5445
Validation Accuracy: 78.8308
Testing Accuracy:    77.2308
Training SVM with lr=0.05, epochs=500, reg_const=0.1…
Epoch 0: Loss = 1.0540
Epoch 100: Loss = 0.6624
Epoch 200: Loss = 0.9046
Epoch 300: Loss = 0.4449
Epoch 400: Loss = 0.3736
Training Accuracy:   69.6963
Validation Accuracy: 69.3538
Testing Accuracy:    68.3077
Training SVM with lr=0.05, epochs=500, reg_const=0.5…
Epoch 0: Loss = 1.1363
Epoch 100: Loss = 0.6951
Epoch 200: Loss = 0.6445
Epoch 300: Loss = 0.6386
Epoch 400: Loss = 0.6782
Training Accuracy:   78.0673
Validation Accuracy: 76.9231
Testing Accuracy:    75.0769
Training SVM with lr=0.05, epochs=1000, reg_const=0.001…
Epoch 0: Loss = 1.1356
Epoch 100: Loss = 0.9227
Epoch 200: Loss = 0.4069
Epoch 300: Loss = 0.2802
Epoch 400: Loss = 0.2620
Epoch 500: Loss = 0.3076
Epoch 600: Loss = 0.2888
Epoch 700: Loss = 0.2544
Epoch 800: Loss = 0.2357
Epoch 900: Loss = 0.2580
Training Accuracy:   94.9733
Validation Accuracy: 94.8308
Testing Accuracy:    94.2154
Training SVM with lr=0.05, epochs=1000, reg_const=0.01…
Epoch 0: Loss = 1.1680
Epoch 100: Loss = 1.3605
Epoch 200: Loss = 0.4161
Epoch 300: Loss = 0.4604
Epoch 400: Loss = 0.2859
```

```
Epoch 500: Loss = 0.2872
Epoch 600: Loss = 0.3196
Epoch 700: Loss = 0.2635
Epoch 800: Loss = 0.2873
Epoch 900: Loss = 0.6877
Training Accuracy:   92.2446
Validation Accuracy: 91.8154
Testing Accuracy:    91.3231
Training SVM with lr=0.05, epochs=1000, reg_const=0.05…
Epoch 0: Loss = 1.2223
Epoch 100: Loss = 0.4331
Epoch 200: Loss = 0.4645
Epoch 300: Loss = 0.7690
Epoch 400: Loss = 0.3132
Epoch 500: Loss = 0.8411
Epoch 600: Loss = 0.2724
Epoch 700: Loss = 0.3175
Epoch 800: Loss = 0.2815
Epoch 900: Loss = 0.3136
Training Accuracy:   88.8798
Validation Accuracy: 87.9385
Testing Accuracy:    87.2000
Training SVM with lr=0.05, epochs=1000, reg_const=0.1…
Epoch 0: Loss = 1.1180
Epoch 100: Loss = 0.6523
Epoch 200: Loss = 0.8584
Epoch 300: Loss = 0.4562
Epoch 400: Loss = 0.3730
Epoch 500: Loss = 1.7499
Epoch 600: Loss = 0.4357
Epoch 700: Loss = 0.3612
Epoch 800: Loss = 1.0518
Epoch 900: Loss = 0.3845
Training Accuracy:   69.3065
Validation Accuracy: 69.1077
Testing Accuracy:    67.6923
Training SVM with lr=0.05, epochs=1000, reg_const=0.5…
Epoch 0: Loss = 1.1018
Epoch 100: Loss = 1.1191
Epoch 200: Loss = 1.1488
Epoch 300: Loss = 1.0745
Epoch 400: Loss = 1.1371
Epoch 500: Loss = 1.1251
Epoch 600: Loss = 1.0893
Epoch 700: Loss = 1.0796
Epoch 800: Loss = 1.1437
Epoch 900: Loss = 1.1190
Training Accuracy:   69.3270
```

```
Validation Accuracy: 67.4462
Testing Accuracy:    66.1538
Training SVM with lr=0.05, epochs=2000, reg_const=0.001…
Epoch 0: Loss = 1.1160
Epoch 100: Loss = 0.7556
Epoch 200: Loss = 0.5628
Epoch 300: Loss = 0.2895
Epoch 400: Loss = 0.3204
Epoch 500: Loss = 0.3230
Epoch 600: Loss = 0.2445
Epoch 700: Loss = 0.2208
Epoch 800: Loss = 0.2179
Epoch 900: Loss = 0.2060
Epoch 1000: Loss = 0.2129
Epoch 1100: Loss = 0.2006
Epoch 1200: Loss = 0.1910
Epoch 1300: Loss = 0.1889
Epoch 1400: Loss = 0.2031
Epoch 1500: Loss = 0.1784
Epoch 1600: Loss = 0.7682
Epoch 1700: Loss = 0.1794
Epoch 1800: Loss = 0.1708
Epoch 1900: Loss = 0.2062
Training Accuracy:   94.0911
Validation Accuracy: 93.5385
Testing Accuracy:    93.4154
Training SVM with lr=0.05, epochs=2000, reg_const=0.01…
Epoch 0: Loss = 1.1944
Epoch 100: Loss = 0.5182
Epoch 200: Loss = 0.3098
Epoch 300: Loss = 0.2826
Epoch 400: Loss = 1.1451
Epoch 500: Loss = 0.3393
Epoch 600: Loss = 0.2622
Epoch 700: Loss = 0.5944
Epoch 800: Loss = 0.2392
Epoch 900: Loss = 0.2734
Epoch 1000: Loss = 0.2366
Epoch 1100: Loss = 0.2550
Epoch 1200: Loss = 0.2315
Epoch 1300: Loss = 0.2390
Epoch 1400: Loss = 0.2382
Epoch 1500: Loss = 0.2371
Epoch 1600: Loss = 0.2286
Epoch 1700: Loss = 0.2324
Epoch 1800: Loss = 0.2121
Epoch 1900: Loss = 0.2106
Training Accuracy:   94.6040
```

```
Validation Accuracy: 94.2154
Testing Accuracy:    93.9692
Training SVM with lr=0.05, epochs=2000, reg_const=0.05…
Epoch 0: Loss = 1.0984
Epoch 100: Loss = 0.4291
Epoch 200: Loss = 0.6969
Epoch 300: Loss = 1.3867
Epoch 400: Loss = 0.2989
Epoch 500: Loss = 0.7158
Epoch 600: Loss = 0.2905
Epoch 700: Loss = 0.7555
Epoch 800: Loss = 0.2806
Epoch 900: Loss = 0.3143
Epoch 1000: Loss = 0.4832
Epoch 1100: Loss = 0.3065
Epoch 1200: Loss = 0.8735
Epoch 1300: Loss = 0.2784
Epoch 1400: Loss = 1.5966
Epoch 1500: Loss = 0.3235
Epoch 1600: Loss = 0.7529
Epoch 1700: Loss = 0.2989
Epoch 1800: Loss = 0.3090
Epoch 1900: Loss = 0.4719
Training Accuracy:   89.5363
Validation Accuracy: 88.7385
Testing Accuracy:    88.0615
Training SVM with lr=0.05, epochs=2000, reg_const=0.1…
Epoch 0: Loss = 1.1000
Epoch 100: Loss = 1.3462
Epoch 200: Loss = 1.6956
Epoch 300: Loss = 0.6183
Epoch 400: Loss = 0.4017
Epoch 500: Loss = 0.7891
Epoch 600: Loss = 0.6107
Epoch 700: Loss = 0.3452
Epoch 800: Loss = 0.7915
Epoch 900: Loss = 0.4946
Epoch 1000: Loss = 0.3999
Epoch 1100: Loss = 1.3141
Epoch 1200: Loss = 0.4529
Epoch 1300: Loss = 0.3729
Epoch 1400: Loss = 1.3583
Epoch 1500: Loss = 0.3946
Epoch 1600: Loss = 0.5002
Epoch 1700: Loss = 0.6201
Epoch 1800: Loss = 0.3569
Epoch 1900: Loss = 1.0884
Training Accuracy:   86.3357
```

```
Validation Accuracy: 85.9692
Testing Accuracy:    84.9231
Training SVM with lr=0.05, epochs=2000, reg_const=0.5…
Epoch 0: Loss = 1.0788
Epoch 100: Loss = 0.7018
Epoch 200: Loss = 0.6887
Epoch 300: Loss = 0.6258
Epoch 400: Loss = 0.6344
Epoch 500: Loss = 0.6751
Epoch 600: Loss = 0.6840
Epoch 700: Loss = 0.6398
Epoch 800: Loss = 0.6312
Epoch 900: Loss = 0.6401
Epoch 1000: Loss = 0.6684
Epoch 1100: Loss = 0.6921
Epoch 1200: Loss = 0.6791
Epoch 1300: Loss = 0.6894
Epoch 1400: Loss = 0.6435
Epoch 1500: Loss = 0.6430
Epoch 1600: Loss = 0.6652
Epoch 1700: Loss = 0.6526
Epoch 1800: Loss = 0.6762
Epoch 1900: Loss = 0.6310
Training Accuracy:   77.3287
Validation Accuracy: 75.2000
Testing Accuracy:    74.7692
Training SVM with lr=0.05, epochs=5000, reg_const=0.001…
Epoch 0: Loss = 1.1276
Epoch 100: Loss = 0.6837
Epoch 200: Loss = 0.3217
Epoch 300: Loss = 0.3117
Epoch 400: Loss = 0.5542
Epoch 500: Loss = 0.2903
Epoch 600: Loss = 0.2299
Epoch 700: Loss = 0.2430
Epoch 800: Loss = 0.2663
Epoch 900: Loss = 0.2297
Epoch 1000: Loss = 0.3182
Epoch 1100: Loss = 0.2152
Epoch 1200: Loss = 1.4261
Epoch 1300: Loss = 0.1952
Epoch 1400: Loss = 0.2607
Epoch 1500: Loss = 0.1915
Epoch 1600: Loss = 0.1939
Epoch 1700: Loss = 0.2393
Epoch 1800: Loss = 0.1876
Epoch 1900: Loss = 0.5480
Epoch 2000: Loss = 0.1767
```

```
Epoch 2100: Loss = 0.1794
Epoch 2200: Loss = 1.1604
Epoch 2300: Loss = 0.1699
Epoch 2400: Loss = 0.1931
Epoch 2500: Loss = 0.1756
Epoch 2600: Loss = 0.2308
Epoch 2700: Loss = 0.1666
Epoch 2800: Loss = 0.1608
Epoch 2900: Loss = 0.1559
Epoch 3000: Loss = 0.1744
Epoch 3100: Loss = 0.1631
Epoch 3200: Loss = 0.1635
Epoch 3300: Loss = 0.1542
Epoch 3400: Loss = 0.1596
Epoch 3500: Loss = 0.5323
Epoch 3600: Loss = 0.1609
Epoch 3700: Loss = 0.1683
Epoch 3800: Loss = 0.2138
Epoch 3900: Loss = 0.5553
Epoch 4000: Loss = 0.1502
Epoch 4100: Loss = 0.1597
Epoch 4200: Loss = 0.1569
Epoch 4300: Loss = 0.1557
Epoch 4400: Loss = 0.1477
Epoch 4500: Loss = 0.1705
Epoch 4600: Loss = 0.3052
Epoch 4700: Loss = 0.1492
Epoch 4800: Loss = 0.1520
Epoch 4900: Loss = 0.1814
Training Accuracy:   95.8145
Validation Accuracy: 95.4462
Testing Accuracy:    94.9538
Training SVM with lr=0.05, epochs=5000, reg_const=0.01…
Epoch 0: Loss = 1.1900
Epoch 100: Loss = 1.2174
Epoch 200: Loss = 0.7439
Epoch 300: Loss = 1.3402
Epoch 400: Loss = 0.2772
Epoch 500: Loss = 0.2780
Epoch 600: Loss = 0.3973
Epoch 700: Loss = 0.2599
Epoch 800: Loss = 0.2743
Epoch 900: Loss = 0.4850
Epoch 1000: Loss = 1.7120
Epoch 1100: Loss = 1.5866
Epoch 1200: Loss = 0.4359
Epoch 1300: Loss = 0.2321
Epoch 1400: Loss = 0.2260
```

```
Epoch 1500: Loss = 0.2278
Epoch 1600: Loss = 0.2324
Epoch 1700: Loss = 0.5351
Epoch 1800: Loss = 0.7247
Epoch 1900: Loss = 0.3186
Epoch 2000: Loss = 1.4601
Epoch 2100: Loss = 1.6540
Epoch 2200: Loss = 0.9518
Epoch 2300: Loss = 0.3415
Epoch 2400: Loss = 0.2258
Epoch 2500: Loss = 0.2382
Epoch 2600: Loss = 0.2581
Epoch 2700: Loss = 0.4531
Epoch 2800: Loss = 0.2831
Epoch 2900: Loss = 0.2284
Epoch 3000: Loss = 0.2475
Epoch 3100: Loss = 0.2172
Epoch 3200: Loss = 0.2145
Epoch 3300: Loss = 0.2142
Epoch 3400: Loss = 0.2161
Epoch 3500: Loss = 0.2169
Epoch 3600: Loss = 0.2146
Epoch 3700: Loss = 0.2147
Epoch 3800: Loss = 0.2132
Epoch 3900: Loss = 0.2188
Epoch 4000: Loss = 0.2183
Epoch 4100: Loss = 0.2184
Epoch 4200: Loss = 0.2127
Epoch 4300: Loss = 0.2058
Epoch 4400: Loss = 0.2067
Epoch 4500: Loss = 0.2246
Epoch 4600: Loss = 0.2207
Epoch 4700: Loss = 0.1995
Epoch 4800: Loss = 0.2008
Epoch 4900: Loss = 0.2157
Training Accuracy:   93.0652
Validation Accuracy: 92.3692
Testing Accuracy:    92.5538
Training SVM with lr=0.05, epochs=5000, reg_const=0.05…
Epoch 0: Loss = 1.0854
Epoch 100: Loss = 0.5258
Epoch 200: Loss = 0.4562
Epoch 300: Loss = 1.3807
Epoch 400: Loss = 0.3174
Epoch 500: Loss = 0.6913
Epoch 600: Loss = 0.3070
Epoch 700: Loss = 1.5957
Epoch 800: Loss = 0.3307
```

```
Epoch 900: Loss = 1.7553
Epoch 1000: Loss = 0.2710
Epoch 1100: Loss = 0.6920
Epoch 1200: Loss = 0.3583
Epoch 1300: Loss = 0.3268
Epoch 1400: Loss = 0.3927
Epoch 1500: Loss = 0.2928
Epoch 1600: Loss = 1.0609
Epoch 1700: Loss = 0.3150
Epoch 1800: Loss = 1.7622
Epoch 1900: Loss = 0.2715
Epoch 2000: Loss = 0.6928
Epoch 2100: Loss = 0.3599
Epoch 2200: Loss = 0.3169
Epoch 2300: Loss = 0.3910
Epoch 2400: Loss = 0.2910
Epoch 2500: Loss = 1.0396
Epoch 2600: Loss = 0.3243
Epoch 2700: Loss = 1.7598
Epoch 2800: Loss = 0.2720
Epoch 2900: Loss = 0.7007
Epoch 3000: Loss = 0.3508
Epoch 3100: Loss = 0.3151
Epoch 3200: Loss = 0.3930
Epoch 3300: Loss = 0.2909
Epoch 3400: Loss = 1.1935
Epoch 3500: Loss = 0.3292
Epoch 3600: Loss = 1.7678
Epoch 3700: Loss = 0.2703
Epoch 3800: Loss = 0.6824
Epoch 3900: Loss = 0.3553
Epoch 4000: Loss = 0.3160
Epoch 4100: Loss = 0.3890
Epoch 4200: Loss = 0.2915
Epoch 4300: Loss = 1.1264
Epoch 4400: Loss = 0.3150
Epoch 4500: Loss = 1.7625
Epoch 4600: Loss = 0.2710
Epoch 4700: Loss = 0.6898
Epoch 4800: Loss = 0.3533
Epoch 4900: Loss = 0.3163
Training Accuracy:   90.0903
Validation Accuracy: 89.4769
Testing Accuracy:    88.5538
Training SVM with lr=0.05, epochs=5000, reg_const=0.1…
Epoch 0: Loss = 1.2072
Epoch 100: Loss = 0.6465
Epoch 200: Loss = 0.8816
```

```
Epoch 300: Loss = 0.4557
Epoch 400: Loss = 0.3744
Epoch 500: Loss = 1.7571
Epoch 600: Loss = 0.4280
Epoch 700: Loss = 0.3742
Epoch 800: Loss = 1.7396
Epoch 900: Loss = 0.4341
Epoch 1000: Loss = 0.4500
Epoch 1100: Loss = 0.7896
Epoch 1200: Loss = 0.3374
Epoch 1300: Loss = 1.7115
Epoch 1400: Loss = 0.4419
Epoch 1500: Loss = 0.3606
Epoch 1600: Loss = 1.7352
Epoch 1700: Loss = 0.4526
Epoch 1800: Loss = 0.3673
Epoch 1900: Loss = 1.3209
Epoch 2000: Loss = 0.3223
Epoch 2100: Loss = 1.0234
Epoch 2200: Loss = 0.4538
Epoch 2300: Loss = 0.3610
Epoch 2400: Loss = 1.7040
Epoch 2500: Loss = 0.4225
Epoch 2600: Loss = 0.3640
Epoch 2700: Loss = 1.6569
Epoch 2800: Loss = 0.4251
Epoch 2900: Loss = 0.7579
Epoch 3000: Loss = 0.5620
Epoch 3100: Loss = 0.3592
Epoch 3200: Loss = 1.7489
Epoch 3300: Loss = 0.4164
Epoch 3400: Loss = 0.8034
Epoch 3500: Loss = 0.7543
Epoch 3600: Loss = 0.3906
Epoch 3700: Loss = 1.2280
Epoch 3800: Loss = 0.4596
Epoch 3900: Loss = 0.3617
Epoch 4000: Loss = 1.7267
Epoch 4100: Loss = 0.3947
Epoch 4200: Loss = 0.3592
Epoch 4300: Loss = 1.0639
Epoch 4400: Loss = 0.4109
Epoch 4500: Loss = 0.7974
Epoch 4600: Loss = 0.7529
Epoch 4700: Loss = 0.4168
Epoch 4800: Loss = 0.7875
Epoch 4900: Loss = 0.7584
Training Accuracy:   86.6434
```

```
Validation Accuracy: 86.5231
Testing Accuracy:    85.4154
Training SVM with lr=0.05, epochs=5000, reg_const=0.5…
Epoch 0: Loss = 1.1645
Epoch 100: Loss = 0.8129
Epoch 200: Loss = 0.6942
Epoch 300: Loss = 0.6104
Epoch 400: Loss = 0.6139
Epoch 500: Loss = 0.7004
Epoch 600: Loss = 0.6538
Epoch 700: Loss = 0.6550
Epoch 800: Loss = 0.6999
Epoch 900: Loss = 0.6119
Epoch 1000: Loss = 0.6805
Epoch 1100: Loss = 0.6104
Epoch 1200: Loss = 0.6207
Epoch 1300: Loss = 0.6543
Epoch 1400: Loss = 0.6480
Epoch 1500: Loss = 0.7106
Epoch 1600: Loss = 0.6808
Epoch 1700: Loss = 0.6635
Epoch 1800: Loss = 0.6131
Epoch 1900: Loss = 0.6226
Epoch 2000: Loss = 0.6886
Epoch 2100: Loss = 0.6805
Epoch 2200: Loss = 0.6329
Epoch 2300: Loss = 0.7103
Epoch 2400: Loss = 0.6570
Epoch 2500: Loss = 0.6518
Epoch 2600: Loss = 0.6207
Epoch 2700: Loss = 0.7028
Epoch 2800: Loss = 0.6876
Epoch 2900: Loss = 0.6647
Epoch 3000: Loss = 0.6809
Epoch 3100: Loss = 0.6654
Epoch 3200: Loss = 0.6473
Epoch 3300: Loss = 0.6798
Epoch 3400: Loss = 0.6469
Epoch 3500: Loss = 0.6856
Epoch 3600: Loss = 0.6118
Epoch 3700: Loss = 0.6521
Epoch 3800: Loss = 0.6273
Epoch 3900: Loss = 0.6783
Epoch 4000: Loss = 0.6116
Epoch 4100: Loss = 0.6890
Epoch 4200: Loss = 0.6120
Epoch 4300: Loss = 0.6214
Epoch 4400: Loss = 0.6119
```

```
Epoch 4500: Loss = 0.6888
Epoch 4600: Loss = 0.6136
Epoch 4700: Loss = 0.6117
Epoch 4800: Loss = 0.6120
Epoch 4900: Loss = 0.6395
Training Accuracy:   77.8416
Validation Accuracy: 76.1846
Testing Accuracy:    75.0769
Training SVM with lr=0.1, epochs=100, reg_const=0.001…
Epoch 0: Loss = 1.6365
Training Accuracy:   88.0181
Validation Accuracy: 86.8308
Testing Accuracy:    87.1385
Training SVM with lr=0.1, epochs=100, reg_const=0.01…
Epoch 0: Loss = 1.6439
Training Accuracy:   82.0886
Validation Accuracy: 82.2769
Testing Accuracy:    80.1231
Training SVM with lr=0.1, epochs=100, reg_const=0.05…
Epoch 0: Loss = 1.6320
Training Accuracy:   83.2786
Validation Accuracy: 82.8308
Testing Accuracy:    80.9231
Training SVM with lr=0.1, epochs=100, reg_const=0.1…
Epoch 0: Loss = 1.7212
Training Accuracy:   78.5392
Validation Accuracy: 77.9077
Testing Accuracy:    76.4923
Training SVM with lr=0.1, epochs=100, reg_const=0.5…
Epoch 0: Loss = 1.7263
Training Accuracy:   58.9249
Validation Accuracy: 57.2308
Testing Accuracy:    55.2000
Training SVM with lr=0.1, epochs=500, reg_const=0.001…
Epoch 0: Loss = 1.8222
Epoch 100: Loss = 0.5643
Epoch 200: Loss = 2.1455
Epoch 300: Loss = 0.5095
Epoch 400: Loss = 0.4185
Training Accuracy:   92.7985
Validation Accuracy: 92.1231
Testing Accuracy:    92.2462
Training SVM with lr=0.1, epochs=500, reg_const=0.01…
Epoch 0: Loss = 1.7808
Epoch 100: Loss = 0.7085
Epoch 200: Loss = 0.6687
Epoch 300: Loss = 1.1523
Epoch 400: Loss = 0.4220
```

```
Training Accuracy:   89.0644
Validation Accuracy: 88.8615
Testing Accuracy:    87.5077
Training SVM with lr=0.1, epochs=500, reg_const=0.05…
Epoch 0: Loss = 1.5634
Epoch 100: Loss = 0.5538
Epoch 200: Loss = 0.5646
Epoch 300: Loss = 0.6961
Epoch 400: Loss = 2.3875
Training Accuracy:   75.7284
Validation Accuracy: 74.0923
Testing Accuracy:    73.2923
Training SVM with lr=0.1, epochs=500, reg_const=0.1…
Epoch 0: Loss = 1.6465
Epoch 100: Loss = 2.6201
Epoch 200: Loss = 1.3494
Epoch 300: Loss = 0.6337
Epoch 400: Loss = 2.5996
Training Accuracy:   72.4456
Validation Accuracy: 71.2615
Testing Accuracy:    69.4154
Training SVM with lr=0.1, epochs=500, reg_const=0.5…
Epoch 0: Loss = 1.7546
Epoch 100: Loss = 2.4169
Epoch 200: Loss = 0.9394
Epoch 300: Loss = 0.9917
Epoch 400: Loss = 3.3195
Training Accuracy:   48.9331
Validation Accuracy: 49.6615
Testing Accuracy:    46.8308
Training SVM with lr=0.1, epochs=1000, reg_const=0.001…
Epoch 0: Loss = 1.8156
Epoch 100: Loss = 0.6735
Epoch 200: Loss = 0.5030
Epoch 300: Loss = 1.0525
Epoch 400: Loss = 0.3807
Epoch 500: Loss = 0.3430
Epoch 600: Loss = 3.3542
Epoch 700: Loss = 0.3345
Epoch 800: Loss = 0.3099
Epoch 900: Loss = 3.3919
Training Accuracy:   94.7066
Validation Accuracy: 94.7077
Testing Accuracy:    94.2154
Training SVM with lr=0.1, epochs=1000, reg_const=0.01…
Epoch 0: Loss = 1.8541
Epoch 100: Loss = 0.5907
Epoch 200: Loss = 0.5337
```

```
Epoch 300: Loss = 0.6715
Epoch 400: Loss = 1.3656
Epoch 500: Loss = 3.7982
Epoch 600: Loss = 0.6108
Epoch 700: Loss = 0.4149
Epoch 800: Loss = 1.3097
Epoch 900: Loss = 0.3975
Training Accuracy:   91.0956
Validation Accuracy: 90.5231
Testing Accuracy:    89.4769
Training SVM with lr=0.1, epochs=1000, reg_const=0.05…
Epoch 0: Loss = 1.8161
Epoch 100: Loss = 0.5838
Epoch 200: Loss = 1.3160
Epoch 300: Loss = 0.5311
Epoch 400: Loss = 0.6720
Epoch 500: Loss = 1.6011
Epoch 600: Loss = 1.2717
Epoch 700: Loss = 0.5539
Epoch 800: Loss = 0.6529
Epoch 900: Loss = 3.4024
Training Accuracy:   76.1592
Validation Accuracy: 74.5846
Testing Accuracy:    73.7231
Training SVM with lr=0.1, epochs=1000, reg_const=0.1…
Epoch 0: Loss = 1.8567
Epoch 100: Loss = 1.4212
Epoch 200: Loss = 0.8344
Epoch 300: Loss = 2.3836
Epoch 400: Loss = 1.3693
Epoch 500: Loss = 0.6114
Epoch 600: Loss = 1.0125
Epoch 700: Loss = 3.7218
Epoch 800: Loss = 0.5719
Epoch 900: Loss = 0.9001
Training Accuracy:   55.3549
Validation Accuracy: 55.0154
Testing Accuracy:    52.3692
Training SVM with lr=0.1, epochs=1000, reg_const=0.5…
Epoch 0: Loss = 1.8267
Epoch 100: Loss = 2.0041
Epoch 200: Loss = 2.4826
Epoch 300: Loss = 2.3107
Epoch 400: Loss = 3.3162
Epoch 500: Loss = 2.2405
Epoch 600: Loss = 3.4049
Epoch 700: Loss = 1.0230
Epoch 800: Loss = 4.2422
```

```
Epoch 900: Loss = 3.5796
Training Accuracy:   58.6582
Validation Accuracy: 56.8615
Testing Accuracy:    54.9538
Training SVM with lr=0.1, epochs=2000, reg_const=0.001…
Epoch 0: Loss = 1.7320
Epoch 100: Loss = 0.5467
Epoch 200: Loss = 0.4373
Epoch 300: Loss = 0.8337
Epoch 400: Loss = 0.3704
Epoch 500: Loss = 0.3309
Epoch 600: Loss = 2.3151
Epoch 700: Loss = 0.2730
Epoch 800: Loss = 0.2688
Epoch 900: Loss = 0.3204
Epoch 1000: Loss = 0.2421
Epoch 1100: Loss = 0.2487
Epoch 1200: Loss = 0.2474
Epoch 1300: Loss = 0.2495
Epoch 1400: Loss = 0.2295
Epoch 1500: Loss = 1.5378
Epoch 1600: Loss = 0.2356
Epoch 1700: Loss = 0.2165
Epoch 1800: Loss = 0.3323
Epoch 1900: Loss = 0.2294
Training Accuracy:   93.7833
Validation Accuracy: 92.9846
Testing Accuracy:    92.8615
Training SVM with lr=0.1, epochs=2000, reg_const=0.01…
Epoch 0: Loss = 1.6588
Epoch 100: Loss = 0.5518
Epoch 200: Loss = 0.5292
Epoch 300: Loss = 0.5051
Epoch 400: Loss = 0.3891
Epoch 500: Loss = 0.3503
Epoch 600: Loss = 0.3809
Epoch 700: Loss = 0.3319
Epoch 800: Loss = 0.3159
Epoch 900: Loss = 0.4049
Epoch 1000: Loss = 0.2832
Epoch 1100: Loss = 0.4134
Epoch 1200: Loss = 0.2737
Epoch 1300: Loss = 0.2809
Epoch 1400: Loss = 0.2890
Epoch 1500: Loss = 0.2948
Epoch 1600: Loss = 0.3559
Epoch 1700: Loss = 0.3104
Epoch 1800: Loss = 0.2771
```

```
Epoch 1900: Loss = 0.2825
Training Accuracy:   91.4239
Validation Accuracy: 90.8308
Testing Accuracy:    90.4615
Training SVM with lr=0.1, epochs=2000, reg_const=0.05…
Epoch 0: Loss = 1.9727
Epoch 100: Loss = 3.2961
Epoch 200: Loss = 3.3145
Epoch 300: Loss = 1.2492
Epoch 400: Loss = 0.5626
Epoch 500: Loss = 0.6964
Epoch 600: Loss = 3.1751
Epoch 700: Loss = 1.1981
Epoch 800: Loss = 0.5181
Epoch 900: Loss = 0.6553
Epoch 1000: Loss = 1.3607
Epoch 1100: Loss = 1.3455
Epoch 1200: Loss = 0.5157
Epoch 1300: Loss = 0.6609
Epoch 1400: Loss = 3.3220
Epoch 1500: Loss = 1.1762
Epoch 1600: Loss = 0.4818
Epoch 1700: Loss = 0.6531
Epoch 1800: Loss = 1.7172
Epoch 1900: Loss = 1.3988
Training Accuracy:   85.8432
Validation Accuracy: 85.9692
Testing Accuracy:    84.4923
Training SVM with lr=0.1, epochs=2000, reg_const=0.1…
Epoch 0: Loss = 1.7736
Epoch 100: Loss = 3.4410
Epoch 200: Loss = 0.5989
Epoch 300: Loss = 0.9581
Epoch 400: Loss = 3.7864
Epoch 500: Loss = 0.5770
Epoch 600: Loss = 1.0368
Epoch 700: Loss = 3.8008
Epoch 800: Loss = 0.5919
Epoch 900: Loss = 0.9602
Epoch 1000: Loss = 3.7463
Epoch 1100: Loss = 1.0909
Epoch 1200: Loss = 1.3120
Epoch 1300: Loss = 0.8997
Epoch 1400: Loss = 1.0485
Epoch 1500: Loss = 3.6979
Epoch 1600: Loss = 0.5732
Epoch 1700: Loss = 1.0058
Epoch 1800: Loss = 3.7137
```

```
Epoch 1900: Loss = 0.5745
Training Accuracy:   79.4419
Validation Accuracy: 79.0154
Testing Accuracy:    77.8462
Training SVM with lr=0.1, epochs=2000, reg_const=0.5…
Epoch 0: Loss = 1.6470
Epoch 100: Loss = 2.3236
Epoch 200: Loss = 2.1022
Epoch 300: Loss = 2.6847
Epoch 400: Loss = 0.9973
Epoch 500: Loss = 3.7348
Epoch 600: Loss = 3.5609
Epoch 700: Loss = 2.1207
Epoch 800: Loss = 3.2498
Epoch 900: Loss = 4.1593
Epoch 1000: Loss = 1.9764
Epoch 1100: Loss = 3.1931
Epoch 1200: Loss = 1.7713
Epoch 1300: Loss = 3.3589
Epoch 1400: Loss = 2.0052
Epoch 1500: Loss = 4.4805
Epoch 1600: Loss = 2.6738
Epoch 1700: Loss = 4.1694
Epoch 1800: Loss = 3.1261
Epoch 1900: Loss = 2.3267
Training Accuracy:   61.5101
Validation Accuracy: 59.8154
Testing Accuracy:    57.9692
Training SVM with lr=0.1, epochs=5000, reg_const=0.001…
Epoch 0: Loss = 1.8667
Epoch 100: Loss = 0.5499
Epoch 200: Loss = 0.4262
Epoch 300: Loss = 0.4298
Epoch 400: Loss = 0.3578
Epoch 500: Loss = 0.3065
Epoch 600: Loss = 0.2920
Epoch 700: Loss = 0.2784
Epoch 800: Loss = 0.2760
Epoch 900: Loss = 0.8417
Epoch 1000: Loss = 0.2418
Epoch 1100: Loss = 0.2715
Epoch 1200: Loss = 0.2346
Epoch 1300: Loss = 0.2376
Epoch 1400: Loss = 0.2460
Epoch 1500: Loss = 0.2216
Epoch 1600: Loss = 1.9513
Epoch 1700: Loss = 0.2439
Epoch 1800: Loss = 0.2151
```

```
Epoch 1900: Loss = 0.2305
Epoch 2000: Loss = 0.3114
Epoch 2100: Loss = 0.2796
Epoch 2200: Loss = 0.2353
Epoch 2300: Loss = 0.2162
Epoch 2400: Loss = 1.0935
Epoch 2500: Loss = 0.2857
Epoch 2600: Loss = 0.2193
Epoch 2700: Loss = 0.2394
Epoch 2800: Loss = 0.2829
Epoch 2900: Loss = 0.2002
Epoch 3000: Loss = 0.4991
Epoch 3100: Loss = 0.2800
Epoch 3200: Loss = 0.2082
Epoch 3300: Loss = 0.2235
Epoch 3400: Loss = 0.2387
Epoch 3500: Loss = 0.2212
Epoch 3600: Loss = 0.2101
Epoch 3700: Loss = 0.2042
Epoch 3800: Loss = 0.4566
Epoch 3900: Loss = 0.1927
Epoch 4000: Loss = 1.3669
Epoch 4100: Loss = 0.2033
Epoch 4200: Loss = 0.2310
Epoch 4300: Loss = 0.3290
Epoch 4400: Loss = 0.2336
Epoch 4500: Loss = 0.1947
Epoch 4600: Loss = 0.4056
Epoch 4700: Loss = 0.2029
Epoch 4800: Loss = 0.1933
Epoch 4900: Loss = 0.2201
Training Accuracy:   93.3114
Validation Accuracy: 92.8615
Testing Accuracy:    92.6769
Training SVM with lr=0.1, epochs=5000, reg_const=0.01…
Epoch 0: Loss = 1.7428
Epoch 100: Loss = 0.6364
Epoch 200: Loss = 1.4863
Epoch 300: Loss = 0.4327
Epoch 400: Loss = 0.3910
Epoch 500: Loss = 0.4291
Epoch 600: Loss = 0.3197
Epoch 700: Loss = 0.3465
Epoch 800: Loss = 0.7848
Epoch 900: Loss = 0.3339
Epoch 1000: Loss = 0.2781
Epoch 1100: Loss = 0.4035
Epoch 1200: Loss = 0.2802
```

```
Epoch 1300: Loss = 0.4033
Epoch 1400: Loss = 0.3203
Epoch 1500: Loss = 0.2911
Epoch 1600: Loss = 0.2657
Epoch 1700: Loss = 0.2735
Epoch 1800: Loss = 0.3403
Epoch 1900: Loss = 0.3154
Epoch 2000: Loss = 0.2666
Epoch 2100: Loss = 2.5228
Epoch 2200: Loss = 0.2807
Epoch 2300: Loss = 2.6908
Epoch 2400: Loss = 0.3287
Epoch 2500: Loss = 0.2583
Epoch 2600: Loss = 0.3546
Epoch 2700: Loss = 0.3056
Epoch 2800: Loss = 0.2651
Epoch 2900: Loss = 0.4276
Epoch 3000: Loss = 0.3093
Epoch 3100: Loss = 0.2584
Epoch 3200: Loss = 2.5251
Epoch 3300: Loss = 0.3258
Epoch 3400: Loss = 0.2577
Epoch 3500: Loss = 0.2691
Epoch 3600: Loss = 0.3438
Epoch 3700: Loss = 0.3114
Epoch 3800: Loss = 0.2657
Epoch 3900: Loss = 0.3317
Epoch 4000: Loss = 0.2705
Epoch 4100: Loss = 0.2609
Epoch 4200: Loss = 0.2780
Epoch 4300: Loss = 0.3917
Epoch 4400: Loss = 0.3034
Epoch 4500: Loss = 0.2598
Epoch 4600: Loss = 0.3631
Epoch 4700: Loss = 0.3108
Epoch 4800: Loss = 0.2647
Epoch 4900: Loss = 2.2824
Training Accuracy:   90.3570
Validation Accuracy: 89.6615
Testing Accuracy:    89.2923
Training SVM with lr=0.1, epochs=5000, reg_const=0.05…
Epoch 0: Loss = 1.9036
Epoch 100: Loss = 1.2987
Epoch 200: Loss = 0.7086
Epoch 300: Loss = 2.6463
Epoch 400: Loss = 1.2797
Epoch 500: Loss = 0.5622
Epoch 600: Loss = 0.6191
```

```
Epoch 700: Loss = 1.4060
Epoch 800: Loss = 1.3097
Epoch 900: Loss = 0.5234
Epoch 1000: Loss = 0.7017
Epoch 1100: Loss = 3.2306
Epoch 1200: Loss = 1.1909
Epoch 1300: Loss = 0.4834
Epoch 1400: Loss = 0.6394
Epoch 1500: Loss = 1.4132
Epoch 1600: Loss = 1.3481
Epoch 1700: Loss = 0.5657
Epoch 1800: Loss = 0.6958
Epoch 1900: Loss = 2.9497
Epoch 2000: Loss = 1.3324
Epoch 2100: Loss = 0.5445
Epoch 2200: Loss = 0.6317
Epoch 2300: Loss = 1.6189
Epoch 2400: Loss = 1.3074
Epoch 2500: Loss = 0.5666
Epoch 2600: Loss = 0.6652
Epoch 2700: Loss = 3.3940
Epoch 2800: Loss = 1.3299
Epoch 2900: Loss = 0.5429
Epoch 3000: Loss = 0.6720
Epoch 3100: Loss = 1.3565
Epoch 3200: Loss = 1.2819
Epoch 3300: Loss = 0.5743
Epoch 3400: Loss = 0.6684
Epoch 3500: Loss = 3.3106
Epoch 3600: Loss = 1.1796
Epoch 3700: Loss = 0.4638
Epoch 3800: Loss = 0.6165
Epoch 3900: Loss = 1.8060
Epoch 4000: Loss = 1.3973
Epoch 4100: Loss = 0.5237
Epoch 4200: Loss = 0.7000
Epoch 4300: Loss = 2.7827
Epoch 4400: Loss = 1.3349
Epoch 4500: Loss = 0.5565
Epoch 4600: Loss = 0.6867
Epoch 4700: Loss = 2.7220
Epoch 4800: Loss = 1.3332
Epoch 4900: Loss = 0.5136
Training Accuracy:   85.3098
Validation Accuracy: 84.9231
Testing Accuracy:    83.0769
Training SVM with lr=0.1, epochs=5000, reg_const=0.1…
Epoch 0: Loss = 1.6459
```

```
Epoch 100: Loss = 1.1035
Epoch 200: Loss = 3.3689
Epoch 300: Loss = 0.5700
Epoch 400: Loss = 0.6459
Epoch 500: Loss = 2.1118
Epoch 600: Loss = 1.3960
Epoch 700: Loss = 0.6887
Epoch 800: Loss = 2.0425
Epoch 900: Loss = 1.3750
Epoch 1000: Loss = 0.6809
Epoch 1100: Loss = 0.9808
Epoch 1200: Loss = 3.8106
Epoch 1300: Loss = 1.3997
Epoch 1400: Loss = 0.6426
Epoch 1500: Loss = 0.9998
Epoch 1600: Loss = 3.9280
Epoch 1700: Loss = 0.5683
Epoch 1800: Loss = 1.0382
Epoch 1900: Loss = 3.7325
Epoch 2000: Loss = 0.5871
Epoch 2100: Loss = 1.0050
Epoch 2200: Loss = 3.6330
Epoch 2300: Loss = 0.5784
Epoch 2400: Loss = 1.0424
Epoch 2500: Loss = 2.1101
Epoch 2600: Loss = 1.3376
Epoch 2700: Loss = 0.8376
Epoch 2800: Loss = 1.4576
Epoch 2900: Loss = 3.9375
Epoch 3000: Loss = 0.5679
Epoch 3100: Loss = 1.0422
Epoch 3200: Loss = 3.5786
Epoch 3300: Loss = 0.5707
Epoch 3400: Loss = 0.6985
Epoch 3500: Loss = 2.2303
Epoch 3600: Loss = 1.3513
Epoch 3700: Loss = 0.6364
Epoch 3800: Loss = 2.0712
Epoch 3900: Loss = 1.3944
Epoch 4000: Loss = 0.6376
Epoch 4100: Loss = 1.8129
Epoch 4200: Loss = 1.6299
Epoch 4300: Loss = 1.3524
Epoch 4400: Loss = 0.6917
Epoch 4500: Loss = 2.0523
Epoch 4600: Loss = 1.3945
Epoch 4700: Loss = 0.6476
Epoch 4800: Loss = 2.0226
```

```
Epoch 4900: Loss = 1.3857
Training Accuracy:   83.4017
Validation Accuracy: 82.5231
Testing Accuracy:    81.6000
Training SVM with lr=0.1, epochs=5000, reg_const=0.5…
Epoch 0: Loss = 1.7875
Epoch 100: Loss = 2.5969
Epoch 200: Loss = 4.5166
Epoch 300: Loss = 2.0783
Epoch 400: Loss = 2.6791
Epoch 500: Loss = 2.4897
Epoch 600: Loss = 2.1234
Epoch 700: Loss = 3.3005
Epoch 800: Loss = 0.7966
Epoch 900: Loss = 4.2372
Epoch 1000: Loss = 3.5023
Epoch 1100: Loss = 2.0142
Epoch 1200: Loss = 2.5745
Epoch 1300: Loss = 1.0209
Epoch 1400: Loss = 4.3102
Epoch 1500: Loss = 4.8599
Epoch 1600: Loss = 3.5723
Epoch 1700: Loss = 3.7253
Epoch 1800: Loss = 2.2306
Epoch 1900: Loss = 1.8597
Epoch 2000: Loss = 3.9489
Epoch 2100: Loss = 2.2899
Epoch 2200: Loss = 1.4608
Epoch 2300: Loss = 3.0858
Epoch 2400: Loss = 1.4356
Epoch 2500: Loss = 1.9893
Epoch 2600: Loss = 1.0256
Epoch 2700: Loss = 4.1818
Epoch 2800: Loss = 2.5415
Epoch 2900: Loss = 3.1108
Epoch 3000: Loss = 4.5776
Epoch 3100: Loss = 2.0715
Epoch 3200: Loss = 3.2947
Epoch 3300: Loss = 2.5794
Epoch 3400: Loss = 1.0391
Epoch 3500: Loss = 4.2776
Epoch 3600: Loss = 3.7183
Epoch 3700: Loss = 3.9447
Epoch 3800: Loss = 4.5540
Epoch 3900: Loss = 2.0434
Epoch 4000: Loss = 3.5657
Epoch 4100: Loss = 1.0779
Epoch 4200: Loss = 2.3754
```

```
Epoch 4300: Loss = 1.8594
Epoch 4400: Loss = 3.1789
Epoch 4500: Loss = 4.2323
Epoch 4600: Loss = 4.2985
Epoch 4700: Loss = 3.3244
Epoch 4800: Loss = 4.6246
Epoch 4900: Loss = 3.3090
Training Accuracy:   52.0107
Validation Accuracy: 51.9385
Testing Accuracy:    49.7231

Best Hyperparameters Found:
{'lr': 0.05, 'epochs': 5000, 'reg_const': 0.001}
Best Training Accuracy:   95.8145
Best Validation Accuracy: 95.4462
Best Testing Accuracy:    94.9538
```

2.4 Plot Accuracy vs Learning Rate

[11]:
```
import matplotlib.pyplot as plt

def PlotAccuracyVsLr(best_params, lr_list, X_train_MR, y_train_MR, X_val_MR,␣
 ↪y_val_MR, X_test_MR, y_test_MR, n_class_MR):

    fixed_epochs = best_params['epochs']
    fixed_reg = best_params['reg_const']

    train_accs, val_accs, test_accs = [], [], []

    for lr in lr_list:
        model = SVM(n_class_MR, lr, fixed_epochs, fixed_reg)
        model.train(X_train_MR, y_train_MR)

        train_accs.append(get_acc(model.predict(X_train_MR), y_train_MR))
        val_accs.append(get_acc(model.predict(X_val_MR), y_val_MR))
        test_accs.append(get_acc(model.predict(X_test_MR), y_test_MR))

    # Plot
    plt.figure(figsize=(8, 5))
    plt.plot(lr_list, train_accs, label="Train Accuracy")
    plt.plot(lr_list, val_accs, label="Validation Accuracy")
    plt.plot(lr_list, test_accs, label="Test Accuracy")
    plt.xlabel("Learning Rate")
    plt.ylabel("Accuracy")
    plt.title("Accuracy vs Learning Rate (Fixed Epochs & Reg Const)")
    plt.grid(True)
    plt.legend()
```

```
    plt.show()
```

[12]:
```
PlotAccuracyVsLr(best_params, lr_list,
                              X_train_MR, y_train_MR,
                              X_val_MR, y_val_MR,
                              X_test_MR, y_test_MR,
                              n_class_MR)
```

```
Epoch 0: Loss = 1.0308
Epoch 100: Loss = 0.5029
Epoch 200: Loss = 0.4337
Epoch 300: Loss = 0.4014
Epoch 400: Loss = 0.3812
Epoch 500: Loss = 0.3660
Epoch 600: Loss = 0.3537
Epoch 700: Loss = 0.3433
Epoch 800: Loss = 0.3345
Epoch 900: Loss = 0.3268
Epoch 1000: Loss = 0.3201
Epoch 1100: Loss = 0.3141
Epoch 1200: Loss = 0.3086
Epoch 1300: Loss = 0.3037
Epoch 1400: Loss = 0.2993
Epoch 1500: Loss = 0.2952
Epoch 1600: Loss = 0.2915
Epoch 1700: Loss = 0.2881
Epoch 1800: Loss = 0.2849
Epoch 1900: Loss = 0.2819
Epoch 2000: Loss = 0.2792
Epoch 2100: Loss = 0.2766
Epoch 2200: Loss = 0.2742
Epoch 2300: Loss = 0.2719
Epoch 2400: Loss = 0.2697
Epoch 2500: Loss = 0.2676
Epoch 2600: Loss = 0.2657
Epoch 2700: Loss = 0.2638
Epoch 2800: Loss = 0.2620
Epoch 2900: Loss = 0.2603
Epoch 3000: Loss = 0.2586
Epoch 3100: Loss = 0.2571
Epoch 3200: Loss = 0.2555
Epoch 3300: Loss = 0.2541
Epoch 3400: Loss = 0.2526
Epoch 3500: Loss = 0.2512
Epoch 3600: Loss = 0.2498
Epoch 3700: Loss = 0.2485
Epoch 3800: Loss = 0.2472
```

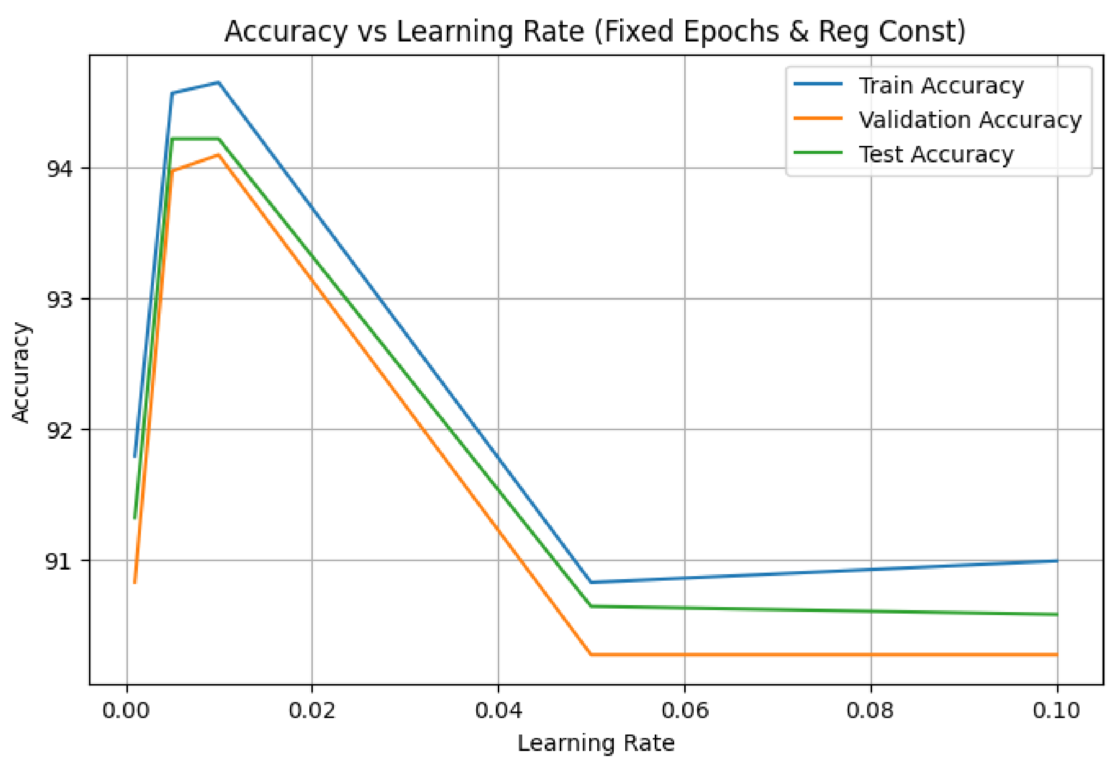
```
Epoch 3900: Loss = 0.2459
Epoch 4000: Loss = 0.2446
Epoch 4100: Loss = 0.2434
Epoch 4200: Loss = 0.2422
Epoch 4300: Loss = 0.2411
Epoch 4400: Loss = 0.2399
Epoch 4500: Loss = 0.2388
Epoch 4600: Loss = 0.2377
Epoch 4700: Loss = 0.2366
Epoch 4800: Loss = 0.2355
Epoch 4900: Loss = 0.2344
Epoch 0: Loss = 0.9277
Epoch 100: Loss = 0.3698
Epoch 200: Loss = 0.3236
Epoch 300: Loss = 0.2987
Epoch 400: Loss = 0.2825
Epoch 500: Loss = 0.2708
Epoch 600: Loss = 0.2616
Epoch 700: Loss = 0.2540
Epoch 800: Loss = 0.2473
Epoch 900: Loss = 0.2413
Epoch 1000: Loss = 0.2358
Epoch 1100: Loss = 0.2307
Epoch 1200: Loss = 0.2259
Epoch 1300: Loss = 0.2214
Epoch 1400: Loss = 0.2171
Epoch 1500: Loss = 0.2130
Epoch 1600: Loss = 0.2092
Epoch 1700: Loss = 0.2054
Epoch 1800: Loss = 0.2018
Epoch 1900: Loss = 0.1984
Epoch 2000: Loss = 0.1951
Epoch 2100: Loss = 0.1919
Epoch 2200: Loss = 0.1888
Epoch 2300: Loss = 0.1860
Epoch 2400: Loss = 0.1834
Epoch 2500: Loss = 0.1808
Epoch 2600: Loss = 0.1784
Epoch 2700: Loss = 0.1761
Epoch 2800: Loss = 0.1739
Epoch 2900: Loss = 0.1718
Epoch 3000: Loss = 0.1699
Epoch 3100: Loss = 0.1680
Epoch 3200: Loss = 0.1662
Epoch 3300: Loss = 0.1645
Epoch 3400: Loss = 0.1628
Epoch 3500: Loss = 0.1612
Epoch 3600: Loss = 0.1597
```

```
Epoch 3700: Loss = 0.1582
Epoch 3800: Loss = 0.1568
Epoch 3900: Loss = 0.1554
Epoch 4000: Loss = 0.1541
Epoch 4100: Loss = 0.1528
Epoch 4200: Loss = 0.1517
Epoch 4300: Loss = 0.1506
Epoch 4400: Loss = 0.1496
Epoch 4500: Loss = 0.1486
Epoch 4600: Loss = 0.1489
Epoch 4700: Loss = 0.1480
Epoch 4800: Loss = 0.1472
Epoch 4900: Loss = 0.1465
Epoch 0: Loss = 0.8591
Epoch 100: Loss = 0.3369
Epoch 200: Loss = 0.3001
Epoch 300: Loss = 0.2780
Epoch 400: Loss = 0.2614
Epoch 500: Loss = 0.2500
Epoch 600: Loss = 0.2384
Epoch 700: Loss = 0.2293
Epoch 800: Loss = 0.2226
Epoch 900: Loss = 0.2151
Epoch 1000: Loss = 0.2093
Epoch 1100: Loss = 0.2034
Epoch 1200: Loss = 0.2011
Epoch 1300: Loss = 0.1993
Epoch 1400: Loss = 0.1908
Epoch 1500: Loss = 0.1933
Epoch 1600: Loss = 0.1825
Epoch 1700: Loss = 0.1841
Epoch 1800: Loss = 0.1781
Epoch 1900: Loss = 0.1754
Epoch 2000: Loss = 0.1743
Epoch 2100: Loss = 0.1681
Epoch 2200: Loss = 0.1754
Epoch 2300: Loss = 0.1643
Epoch 2400: Loss = 0.1666
Epoch 2500: Loss = 0.1703
Epoch 2600: Loss = 0.1612
Epoch 2700: Loss = 0.1576
Epoch 2800: Loss = 0.1598
Epoch 2900: Loss = 0.1586
Epoch 3000: Loss = 0.1598
Epoch 3100: Loss = 0.1578
Epoch 3200: Loss = 0.1558
Epoch 3300: Loss = 0.1553
Epoch 3400: Loss = 0.1550
```

```
Epoch 3500: Loss = 0.1538
Epoch 3600: Loss = 0.1524
Epoch 3700: Loss = 0.1520
Epoch 3800: Loss = 0.1512
Epoch 3900: Loss = 0.1500
Epoch 4000: Loss = 0.1509
Epoch 4100: Loss = 0.1523
Epoch 4200: Loss = 0.1473
Epoch 4300: Loss = 0.1534
Epoch 4400: Loss = 0.1388
Epoch 4500: Loss = 0.1375
Epoch 4600: Loss = 0.1383
Epoch 4700: Loss = 0.1467
Epoch 4800: Loss = 0.1590
Epoch 4900: Loss = 0.1418
Epoch 0: Loss = 1.0780
Epoch 100: Loss = 0.7748
Epoch 200: Loss = 0.6582
Epoch 300: Loss = 0.3125
Epoch 400: Loss = 0.3167
Epoch 500: Loss = 0.5246
Epoch 600: Loss = 0.4035
Epoch 700: Loss = 0.2665
Epoch 800: Loss = 0.2313
Epoch 900: Loss = 0.2671
Epoch 1000: Loss = 0.2154
Epoch 1100: Loss = 0.2278
Epoch 1200: Loss = 0.2076
Epoch 1300: Loss = 0.1951
Epoch 1400: Loss = 0.2022
Epoch 1500: Loss = 0.2157
Epoch 1600: Loss = 0.1976
Epoch 1700: Loss = 0.6189
Epoch 1800: Loss = 0.1798
Epoch 1900: Loss = 0.1983
Epoch 2000: Loss = 0.1972
Epoch 2100: Loss = 0.1831
Epoch 2200: Loss = 0.2924
Epoch 2300: Loss = 0.1786
Epoch 2400: Loss = 0.1685
Epoch 2500: Loss = 0.1959
Epoch 2600: Loss = 0.3294
Epoch 2700: Loss = 0.2066
Epoch 2800: Loss = 0.2211
Epoch 2900: Loss = 0.1677
Epoch 3000: Loss = 0.1670
Epoch 3100: Loss = 0.1912
Epoch 3200: Loss = 0.1899
```

```
Epoch 3300: Loss = 0.1648
Epoch 3400: Loss = 0.2730
Epoch 3500: Loss = 0.1555
Epoch 3600: Loss = 0.1771
Epoch 3700: Loss = 0.1707
Epoch 3800: Loss = 0.1711
Epoch 3900: Loss = 0.1685
Epoch 4000: Loss = 0.1860
Epoch 4100: Loss = 0.1629
Epoch 4200: Loss = 0.1618
Epoch 4300: Loss = 0.1618
Epoch 4400: Loss = 0.1721
Epoch 4500: Loss = 0.1486
Epoch 4600: Loss = 0.1833
Epoch 4700: Loss = 0.1880
Epoch 4800: Loss = 0.1500
Epoch 4900: Loss = 0.1440
Epoch 0: Loss = 1.6811
Epoch 100: Loss = 0.5782
Epoch 200: Loss = 0.4346
Epoch 300: Loss = 0.4429
Epoch 400: Loss = 0.4310
Epoch 500: Loss = 0.3264
Epoch 600: Loss = 3.3743
Epoch 700: Loss = 0.4745
Epoch 800: Loss = 0.3064
Epoch 900: Loss = 0.2965
Epoch 1000: Loss = 0.2866
Epoch 1100: Loss = 0.2945
Epoch 1200: Loss = 0.2628
Epoch 1300: Loss = 0.2856
Epoch 1400: Loss = 0.2465
Epoch 1500: Loss = 0.2671
Epoch 1600: Loss = 0.2865
Epoch 1700: Loss = 0.2257
Epoch 1800: Loss = 0.2528
Epoch 1900: Loss = 1.1670
Epoch 2000: Loss = 0.2320
Epoch 2100: Loss = 0.2154
Epoch 2200: Loss = 1.0194
Epoch 2300: Loss = 1.6307
Epoch 2400: Loss = 0.2209
Epoch 2500: Loss = 0.2176
Epoch 2600: Loss = 0.2654
Epoch 2700: Loss = 0.2115
Epoch 2800: Loss = 0.1920
Epoch 2900: Loss = 0.2117
Epoch 3000: Loss = 0.2116
```

```
Epoch 3100: Loss = 0.2570
Epoch 3200: Loss = 0.7219
Epoch 3300: Loss = 0.2339
Epoch 3400: Loss = 0.2039
Epoch 3500: Loss = 0.7500
Epoch 3600: Loss = 0.2157
Epoch 3700: Loss = 0.2363
Epoch 3800: Loss = 0.2422
Epoch 3900: Loss = 0.3260
Epoch 4000: Loss = 0.2673
Epoch 4100: Loss = 2.2378
Epoch 4200: Loss = 0.1916
Epoch 4300: Loss = 0.2032
Epoch 4400: Loss = 0.2282
Epoch 4500: Loss = 0.1993
Epoch 4600: Loss = 0.1912
Epoch 4700: Loss = 0.1983
Epoch 4800: Loss = 0.2270
Epoch 4900: Loss = 0.2241
```



2.5 Plot Accuracy vs Regularization Constant

[13]:
```
def PlotAccuracyVsReg(best_params, reg_const_list, X_train_MR, y_train_MR,␣
↪X_val_MR, y_val_MR, X_test_MR, y_test_MR, n_class_MR):

    fixed_lr = best_params['lr']
    fixed_epochs = best_params['epochs']

    train_accs, val_accs, test_accs = [], [], []

    for reg in reg_const_list:
        model = SVM(n_class_MR, fixed_lr, fixed_epochs, reg)
        model.train(X_train_MR, y_train_MR)

        train_accs.append(get_acc(model.predict(X_train_MR), y_train_MR))
        val_accs.append(get_acc(model.predict(X_val_MR), y_val_MR))
        test_accs.append(get_acc(model.predict(X_test_MR), y_test_MR))

    # Plot
    plt.figure(figsize=(8, 5))
    plt.plot(reg_const_list, train_accs, label="Train Accuracy")
    plt.plot(reg_const_list, val_accs, label="Validation Accuracy")
    plt.plot(reg_const_list, test_accs, label="Test Accuracy")
    plt.xlabel("Regularization Constant")
    plt.ylabel("Accuracy")
    plt.title("Accuracy vs Regularization Constant (Fixed LR & Epochs)")
    plt.grid(True)
    plt.legend()
    plt.show()
```

[14]:
```
PlotAccuracyVsReg(best_params, reg_const_list,
                               X_train_MR, y_train_MR,
                               X_val_MR, y_val_MR,
                               X_test_MR, y_test_MR,
                               n_class_MR)
```

```
Epoch 0: Loss = 1.1183
Epoch 100: Loss = 0.7295
Epoch 200: Loss = 0.3311
Epoch 300: Loss = 0.2921
Epoch 400: Loss = 0.2671
Epoch 500: Loss = 0.2484
Epoch 600: Loss = 0.2356
Epoch 700: Loss = 0.2192
Epoch 800: Loss = 0.2181
Epoch 900: Loss = 0.2111
Epoch 1000: Loss = 0.2000
Epoch 1100: Loss = 0.4413
```

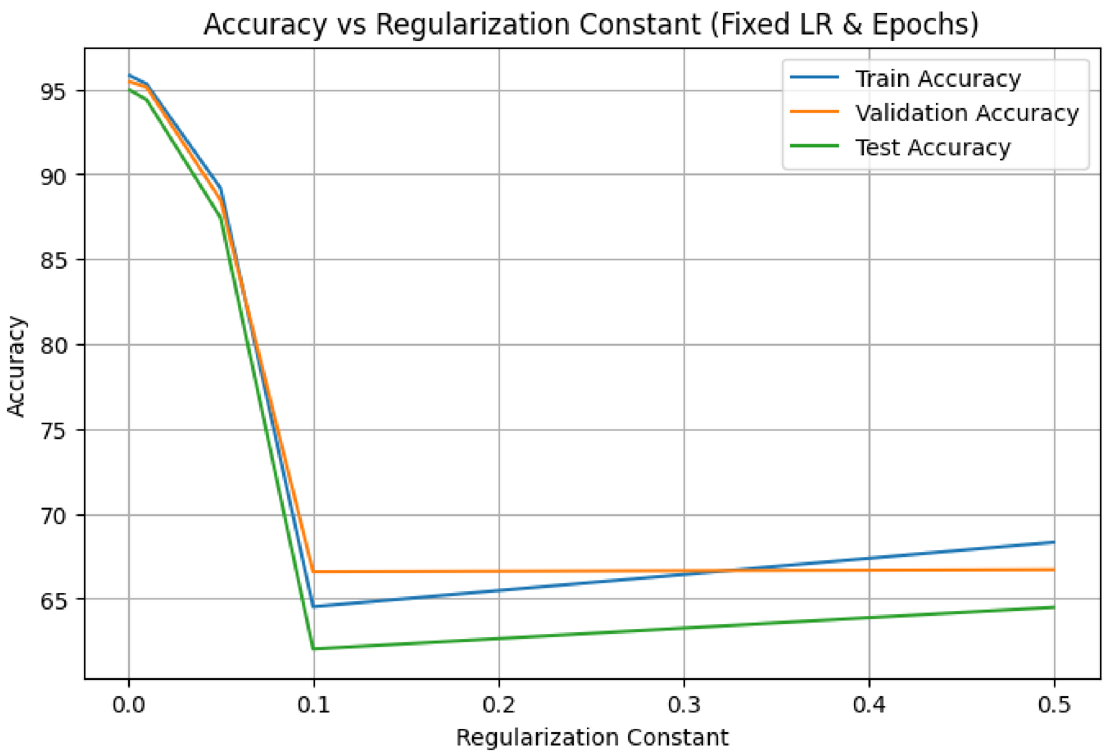
```
Epoch 1200: Loss = 0.1864
Epoch 1300: Loss = 0.6807
Epoch 1400: Loss = 0.2023
Epoch 1500: Loss = 0.4919
Epoch 1600: Loss = 0.1848
Epoch 1700: Loss = 0.1887
Epoch 1800: Loss = 0.6819
Epoch 1900: Loss = 0.1830
Epoch 2000: Loss = 0.1994
Epoch 2100: Loss = 0.1872
Epoch 2200: Loss = 0.1923
Epoch 2300: Loss = 0.1764
Epoch 2400: Loss = 0.1808
Epoch 2500: Loss = 0.1671
Epoch 2600: Loss = 0.1942
Epoch 2700: Loss = 0.1961
Epoch 2800: Loss = 0.1652
Epoch 2900: Loss = 0.1660
Epoch 3000: Loss = 0.4243
Epoch 3100: Loss = 0.1596
Epoch 3200: Loss = 0.1572
Epoch 3300: Loss = 0.1759
Epoch 3400: Loss = 0.2265
Epoch 3500: Loss = 0.6267
Epoch 3600: Loss = 0.2239
Epoch 3700: Loss = 0.1506
Epoch 3800: Loss = 0.1571
Epoch 3900: Loss = 0.1573
Epoch 4000: Loss = 0.1977
Epoch 4100: Loss = 0.1592
Epoch 4200: Loss = 0.1612
Epoch 4300: Loss = 0.5098
Epoch 4400: Loss = 0.1553
Epoch 4500: Loss = 0.2189
Epoch 4600: Loss = 0.1848
Epoch 4700: Loss = 0.1754
Epoch 4800: Loss = 0.1503
Epoch 4900: Loss = 0.1482
Epoch 0: Loss = 1.0075
Epoch 100: Loss = 0.4090
Epoch 200: Loss = 0.7477
Epoch 300: Loss = 0.3327
Epoch 400: Loss = 0.3217
Epoch 500: Loss = 0.2925
Epoch 600: Loss = 0.2720
Epoch 700: Loss = 0.2941
Epoch 800: Loss = 0.6276
Epoch 900: Loss = 0.2596
```

```
Epoch 1000: Loss = 0.2642
Epoch 1100: Loss = 0.2322
Epoch 1200: Loss = 0.2318
Epoch 1300: Loss = 0.2376
Epoch 1400: Loss = 0.2294
Epoch 1500: Loss = 0.2369
Epoch 1600: Loss = 0.2314
Epoch 1700: Loss = 0.2283
Epoch 1800: Loss = 0.2237
Epoch 1900: Loss = 0.2222
Epoch 2000: Loss = 0.2234
Epoch 2100: Loss = 0.2158
Epoch 2200: Loss = 0.2366
Epoch 2300: Loss = 0.2048
Epoch 2400: Loss = 0.2076
Epoch 2500: Loss = 0.2154
Epoch 2600: Loss = 0.4601
Epoch 2700: Loss = 0.5261
Epoch 2800: Loss = 1.7321
Epoch 2900: Loss = 1.0877
Epoch 3000: Loss = 0.3421
Epoch 3100: Loss = 0.2751
Epoch 3200: Loss = 0.2164
Epoch 3300: Loss = 0.2114
Epoch 3400: Loss = 0.2181
Epoch 3500: Loss = 0.2211
Epoch 3600: Loss = 0.2086
Epoch 3700: Loss = 0.2197
Epoch 3800: Loss = 0.2058
Epoch 3900: Loss = 0.2013
Epoch 4000: Loss = 0.2027
Epoch 4100: Loss = 0.2173
Epoch 4200: Loss = 0.2847
Epoch 4300: Loss = 0.5651
Epoch 4400: Loss = 0.9135
Epoch 4500: Loss = 0.3642
Epoch 4600: Loss = 0.2214
Epoch 4700: Loss = 0.2155
Epoch 4800: Loss = 0.2227
Epoch 4900: Loss = 0.2239
Epoch 0: Loss = 1.1978
Epoch 100: Loss = 0.7763
Epoch 200: Loss = 0.3728
Epoch 300: Loss = 0.3251
Epoch 400: Loss = 0.7534
Epoch 500: Loss = 0.3938
Epoch 600: Loss = 0.3028
Epoch 700: Loss = 0.8389
```

```
Epoch 800: Loss = 0.2818
Epoch 900: Loss = 1.5623
Epoch 1000: Loss = 0.3254
Epoch 1100: Loss = 0.7497
Epoch 1200: Loss = 0.2835
Epoch 1300: Loss = 0.3103
Epoch 1400: Loss = 0.4729
Epoch 1500: Loss = 0.3081
Epoch 1600: Loss = 0.8634
Epoch 1700: Loss = 0.2776
Epoch 1800: Loss = 1.6065
Epoch 1900: Loss = 0.3242
Epoch 2000: Loss = 0.7477
Epoch 2100: Loss = 0.3028
Epoch 2200: Loss = 0.3088
Epoch 2300: Loss = 0.4771
Epoch 2400: Loss = 0.3059
Epoch 2500: Loss = 0.8755
Epoch 2600: Loss = 0.2774
Epoch 2700: Loss = 1.6011
Epoch 2800: Loss = 0.3254
Epoch 2900: Loss = 0.7473
Epoch 3000: Loss = 0.2992
Epoch 3100: Loss = 0.3085
Epoch 3200: Loss = 0.5027
Epoch 3300: Loss = 0.3083
Epoch 3400: Loss = 0.8711
Epoch 3500: Loss = 0.2787
Epoch 3600: Loss = 1.6356
Epoch 3700: Loss = 0.3225
Epoch 3800: Loss = 0.7439
Epoch 3900: Loss = 0.3166
Epoch 4000: Loss = 0.3088
Epoch 4100: Loss = 0.4882
Epoch 4200: Loss = 0.3066
Epoch 4300: Loss = 0.8741
Epoch 4400: Loss = 0.2783
Epoch 4500: Loss = 1.6310
Epoch 4600: Loss = 0.3231
Epoch 4700: Loss = 0.7490
Epoch 4800: Loss = 0.2988
Epoch 4900: Loss = 0.3086
Epoch 0: Loss = 1.1088
Epoch 100: Loss = 1.3952
Epoch 200: Loss = 1.7006
Epoch 300: Loss = 0.5739
Epoch 400: Loss = 0.3736
Epoch 500: Loss = 1.2065
```

```
Epoch 600: Loss = 0.6053
Epoch 700: Loss = 0.3748
Epoch 800: Loss = 0.9775
Epoch 900: Loss = 0.4562
Epoch 1000: Loss = 0.3603
Epoch 1100: Loss = 1.6364
Epoch 1200: Loss = 0.3931
Epoch 1300: Loss = 0.7870
Epoch 1400: Loss = 0.6995
Epoch 1500: Loss = 0.3434
Epoch 1600: Loss = 1.0063
Epoch 1700: Loss = 0.4068
Epoch 1800: Loss = 0.3611
Epoch 1900: Loss = 1.6517
Epoch 2000: Loss = 0.3866
Epoch 2100: Loss = 0.7862
Epoch 2200: Loss = 0.4768
Epoch 2300: Loss = 0.3553
Epoch 2400: Loss = 1.7156
Epoch 2500: Loss = 0.4181
Epoch 2600: Loss = 0.3601
Epoch 2700: Loss = 0.7060
Epoch 2800: Loss = 0.3584
Epoch 2900: Loss = 0.7912
Epoch 3000: Loss = 0.5870
Epoch 3100: Loss = 0.3615
Epoch 3200: Loss = 1.7482
Epoch 3300: Loss = 0.3744
Epoch 3400: Loss = 0.7872
Epoch 3500: Loss = 0.7405
Epoch 3600: Loss = 0.3964
Epoch 3700: Loss = 1.3746
Epoch 3800: Loss = 0.4586
Epoch 3900: Loss = 0.3940
Epoch 4000: Loss = 1.7646
Epoch 4100: Loss = 0.4332
Epoch 4200: Loss = 1.1030
Epoch 4300: Loss = 0.4030
Epoch 4400: Loss = 0.3622
Epoch 4500: Loss = 1.7400
Epoch 4600: Loss = 0.4411
Epoch 4700: Loss = 0.6504
Epoch 4800: Loss = 0.7029
Epoch 4900: Loss = 0.3724
Epoch 0: Loss = 0.9932
Epoch 100: Loss = 1.1407
Epoch 200: Loss = 1.0224
Epoch 300: Loss = 1.0492
```

```
Epoch 400: Loss = 1.0563
Epoch 500: Loss = 1.1255
Epoch 600: Loss = 1.0599
Epoch 700: Loss = 1.1295
Epoch 800: Loss = 1.1093
Epoch 900: Loss = 1.0789
Epoch 1000: Loss = 1.0877
Epoch 1100: Loss = 1.0664
Epoch 1200: Loss = 1.1519
Epoch 1300: Loss = 1.0797
Epoch 1400: Loss = 1.0926
Epoch 1500: Loss = 1.0235
Epoch 1600: Loss = 1.0518
Epoch 1700: Loss = 1.0296
Epoch 1800: Loss = 1.0538
Epoch 1900: Loss = 1.0563
Epoch 2000: Loss = 1.0386
Epoch 2100: Loss = 1.0430
Epoch 2200: Loss = 1.0562
Epoch 2300: Loss = 1.0230
Epoch 2400: Loss = 1.0300
Epoch 2500: Loss = 1.0249
Epoch 2600: Loss = 1.0233
Epoch 2700: Loss = 1.0235
Epoch 2800: Loss = 1.0275
Epoch 2900: Loss = 1.0350
Epoch 3000: Loss = 1.0272
Epoch 3100: Loss = 1.0250
Epoch 3200: Loss = 1.0507
Epoch 3300: Loss = 1.0555
Epoch 3400: Loss = 1.0482
Epoch 3500: Loss = 1.0263
Epoch 3600: Loss = 1.0282
Epoch 3700: Loss = 1.0220
Epoch 3800: Loss = 1.0320
Epoch 3900: Loss = 1.0509
Epoch 4000: Loss = 1.0601
Epoch 4100: Loss = 1.0223
Epoch 4200: Loss = 1.0225
Epoch 4300: Loss = 1.0366
Epoch 4400: Loss = 1.0224
Epoch 4500: Loss = 1.0514
Epoch 4600: Loss = 1.0223
Epoch 4700: Loss = 1.0422
Epoch 4800: Loss = 1.0360
Epoch 4900: Loss = 1.1226
```


Accuracy vs Regularization Constant (Fixed LR & Epochs)
Train Accuracy
Validation Accuracy
Test Accuracy
Accuracy
95
90
85
80
75
70
65
0.0
0.1
0.2
0.3
0.4
0.5
Regularization Constant