In [1]:

```python
import numpy as np
from sklearn.datasets import make_regression
from sklearn.utils import shuffle
import matplotlib.pyplot as plt
import time

class MyLinearRegression:
    def __init__(self, learning_rate=0.01, epochs=1000, batch_size=None):
        self.learning_rate = learning_rate
        self.epochs = epochs
        self.batch_size = batch_size
        self.weights = None
        self.validation_history = []

    def _generate_batches(self, X, y):
        n_samples = X.shape[0]
        if self.batch_size is None:
            self.batch_size = n_samples
        for start in range(0, n_samples, self.batch_size):
            end = min(start + self.batch_size, n_samples)
            yield X[start:end], y[start:end]

    def fit(self, X, y, X_val=None, y_val=None):
        n_samples, n_features = X.shape
        self.weights = np.random.randn(n_features) # Initialize weights

        for epoch in range(self.epochs):
            X, y = shuffle(X, y) # Shuffle the data
            for X_batch, y_batch in self._generate_batches(X, y):
                y_pred = X_batch.dot(self.weights)
                errors = y_pred - y_batch.flatten() # Flatten y_batch if
                gradient = X_batch.T.dot(errors) / X_batch.shape[0]
                self.weights -= self.learning_rate * gradient
            if X_val is not None and y_val is not None:
                y_val_pred = self.predict(X_val)
                mae = self.mean_absolute_error(y_val, y_val_pred)
                self.validation_history.append(mae)

    def predict(self, X):
        return X.dot(self.weights)

    def mean_absolute_error(self, y_true, y_pred):
        return np.mean(np.abs(y_true - y_pred))

    def plot_learning_curve(self):
        plt.figure(figsize=(10, 6))
        plt.plot(self.validation_history, label='MAE per epoch')
        plt.xlabel('Epochs')
        plt.ylabel('Mean Absolute Error')
        plt.title('Learning Curve')
        plt.legend()
        plt.grid(True)
        plt.show()

# Генерация синтетических данных
```

```python
X, y = make_regression(n_samples=100000, n_features=20, noise=0.1)

# Разделение данных на обучающую и валидационную выборки
n_train = int(0.8 * len(X))
X_train, X_val = X[:n_train], X[n_train:]
y_train, y_val = y[:n_train], y[n_train:]

# Размеры батчей для тестирования
batch_sizes = [10, 50, 100, 500, 1000]

# Количество эпох для тестирования
epochs = [10, 50, 100, 500, 1000]

# Тестирование различных размеров батчей и количества эпох
for batch_size in batch_sizes:
    for epoch in epochs:
        print(f"Batch size: {batch_size}, Epochs: {epoch}")
        model = MyLinearRegression(learning_rate=0.01, epochs=epoch, batc
        start_time = time.time()
        model.fit(X_train, y_train, X_val, y_val)
        end_time = time.time()
        training_time = end_time - start_time
        mae = model.mean_absolute_error(y_val, model.predict(X_val))
        print(f"Training time: {training_time:.2f} seconds, MAE: {mae:.4f
        print("--------------------------------------------------")
```

```
Batch size: 10, Epochs: 10
Training time: 0.68 seconds, MAE: 0.0800
--------------------------------------------------
Batch size: 10, Epochs: 50
Training time: 3.43 seconds, MAE: 0.0799
--------------------------------------------------
Batch size: 10, Epochs: 100
Training time: 6.96 seconds, MAE: 0.0802
--------------------------------------------------
Batch size: 10, Epochs: 500
Training time: 45.64 seconds, MAE: 0.0799
--------------------------------------------------
Batch size: 10, Epochs: 1000
Training time: 90.85 seconds, MAE: 0.0803
--------------------------------------------------
Batch size: 50, Epochs: 10
Training time: 0.47 seconds, MAE: 0.0797
--------------------------------------------------
Batch size: 50, Epochs: 50
Training time: 2.07 seconds, MAE: 0.0797
--------------------------------------------------
Batch size: 50, Epochs: 100
Training time: 4.10 seconds, MAE: 0.0797
--------------------------------------------------
Batch size: 50, Epochs: 500
Training time: 18.75 seconds, MAE: 0.0797
--------------------------------------------------
Batch size: 50, Epochs: 1000
Training time: 36.19 seconds, MAE: 0.0797
--------------------------------------------------
Batch size: 100, Epochs: 10
```

```
Training time: 0.30 seconds, MAE: 0.0796
--------------------------------------------------
Batch size: 100, Epochs: 50
Training time: 1.36 seconds, MAE: 0.0797
--------------------------------------------------
Batch size: 100, Epochs: 100
Training time: 2.97 seconds, MAE: 0.0797
--------------------------------------------------
Batch size: 100, Epochs: 500
Training time: 15.22 seconds, MAE: 0.0797
--------------------------------------------------
Batch size: 100, Epochs: 1000
Training time: 30.88 seconds, MAE: 0.0797
--------------------------------------------------
Batch size: 500, Epochs: 10
Training time: 0.76 seconds, MAE: 0.0796
--------------------------------------------------
Batch size: 500, Epochs: 50
Training time: 1.77 seconds, MAE: 0.0796
--------------------------------------------------
Batch size: 500, Epochs: 100
Training time: 1.73 seconds, MAE: 0.0796
--------------------------------------------------
Batch size: 500, Epochs: 500
Training time: 19.06 seconds, MAE: 0.0796
--------------------------------------------------
Batch size: 500, Epochs: 1000
Training time: 45.10 seconds, MAE: 0.0796
--------------------------------------------------
Batch size: 1000, Epochs: 10
Training time: 0.13 seconds, MAE: 0.1025
--------------------------------------------------
Batch size: 1000, Epochs: 50
Training time: 0.63 seconds, MAE: 0.0796
--------------------------------------------------
Batch size: 1000, Epochs: 100
Training time: 1.40 seconds, MAE: 0.0796
--------------------------------------------------
Batch size: 1000, Epochs: 500
Training time: 22.53 seconds, MAE: 0.0796
--------------------------------------------------
Batch size: 1000, Epochs: 1000
Training time: 38.66 seconds, MAE: 0.0796
--------------------------------------------------
```

In [ ]:
```
Вывод

Из результатов тестов видно, что размер батча и количество эпох влияют на
Несколько ключевых наблюдений
Увеличение размера батча приводит к уменьшению времени обучения, что може
Увеличение количества эпох увеличивает время обучения, поскольку модель т
поскольку он обновляет веса после каждого батча, а не после каждой эпохи.
MAE остается относительно стабильной и не сильно варьируется в зависимост
На основе результатов, оптимальным размером батча может быть 500, а колич
```