Machine Learning Basics

A machine learning algorithm is an algorithm that is able to learn from data.

Mitchell (1997) provides a succinct definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E".

Task T

Machine learning tasks are usually described in terms of how the machine learning system should process an example. An example is a collection of features that have been quantitatively measured from some object or event that we want the machine learning system to process. We typically represent an example as a vector $\mathbf{x} \in \mathbb{R}^n$ where each entry x_i of the vector is another feature.

Common machine learning tasks include the following:

- Classification: In this type of task, the computer program is asked to specify which of k categories some input belongs to.
- Classification with missing inputs: Classification becomes more challenging if the computer program is not guaranteed that every measurement in its input vector will always be provided. To solve the classification task, the learning algorithm only has to define a single function mapping from a vector input to a categorical output. When some of the inputs may be missing,rather than providing a single classification function, the learning algorithm must learn a set of functions.
- Regression: In this type of task, the computer program is asked to predict a numerical value given some input.
- Transcription: In this type of task, the machine learning system is asked to observe a relatively unstructured representation of some kind of data and transcribe the information into discrete textual form.
- Machine translation: In a machine translation task, the input already consists of a sequence of symbols in some language, and the computer program must convert this into a sequence of symbols in another language.
- Synthesis and sampling: In this type of task, the machine learning algorithm is asked to generate new examples that are similar to those in the training data.

The Performance Measure P

This quantitative measure evaluates the performance of the machine learning algorithm.

i.e. For tasks such as classification and transcription, we often measure the accuracy of the model. Accuracy is just the proportion of examples for which the model produces the correct output. We can also obtain equivalent information by measuring the error rate, the proportion of examples for which the model produces an incorrect output.

We therefore evaluate these performance measures using a test set of data that is separate from the data used for training the machine learning system.

The Experience E

A dataset is a collection of many examples, and we call examples data points.

Unsupervised learning algorithms experience a dataset containing many features, then learn useful properties of the structure of this dataset.

Supervised learning algorithms experience a dataset containing features,but each example is also associated with a label or target.

Linear Regression

Let us consider the problem of Linear Regression. Linear regression solves a regression problem. In other words, the goal is to build a system that can take a vector $\mathbf{x} \in \mathbb{R}^n$ as input and predict the value of a scalar $y \in \mathbb{R}$ as its output. The output of linear regression is a linear function of the input.

Let $\hat{y}$ be the value that our model predicts that y should take on. We define the output to be

$$\hat{y} = w^T x$$

where $w \in \mathbb{R}$ is a vector of parameters. Parameters are values that control the behavior of the system. Each parameter w_i affects the feature x_i is either in a positive, negative, or does not affect all.

So far we have a definition of our task T: to predict y from x by out putting $\hat{y} = w^T x$.
Next, let us define a Performance Measure P for the task. How about we select the mean squared error(MSE) between the true value y and the predicted value $\hat{y}$ as out P? In this case, our goal is to select the set of w that minimizes the MSE.

Assuming that we have m number of data points, we can write the MSE as follows:

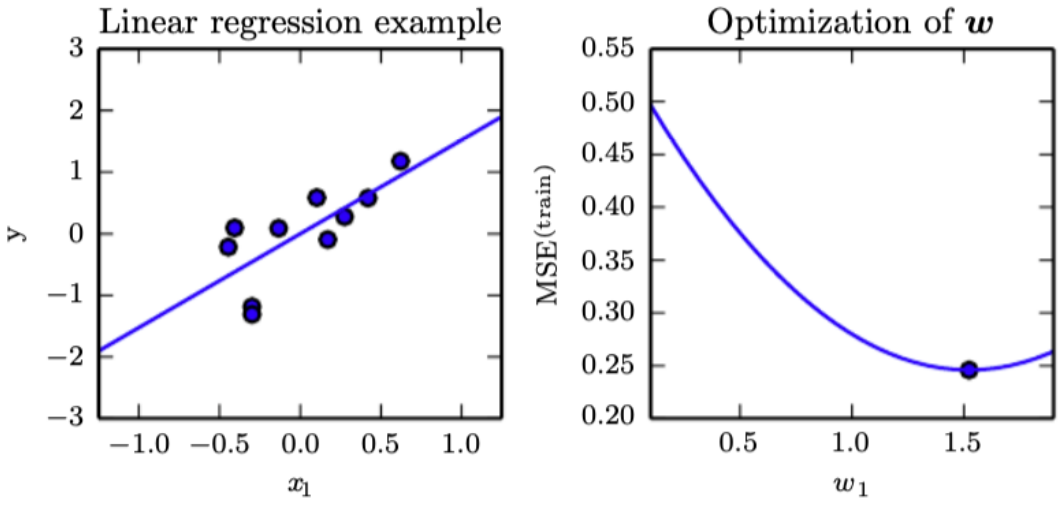
$$MSE = \frac{1}{m}\Sigma_i(\hat{y} - y)_i^2. \tag{1}$$

To make a machine learning algorithm, we need to design an algorithm that will improve weights w in a way that reduces MSE when the algorithm is allowed to gain experience by observing a training set (X, y). We can achieve this by minimize the mean squared error on the training set. To minimize MSE on the train, we can simply solve for where its gradient is 0. By doing some math, we arrive at the optimal set of weights for w as,

$$w = (XX^T)^{-1}X^T y \tag{2}$$

2 is known as the normal equations.

The image below visualizes our linear regression problem:



The following is a numerical example.

```
import numpy as np
import matplotlib.pyplot as plt

# Rainfall vs. Crop Yield
```

```
x = np.array([100, 150, 200, 250, 300, 350, 400, 450, 500, 550])  # Rainfall
    (mm)
y = np.array([2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5])  # Crop
   Yield (tons/ha)

# X as single feature
X = x.reshape(-1, 1)
y = y.reshape(-1, 1)

# w = (X^T X)^{-1} X^T y
XT = X.T
XTX = XT @ X
XTX_inv = np.linalg.inv(XTX)
XTy = XT @ y
w = XTX_inv @ XTy

w_optimal = w[0][0]  # Slope only

# Calculate the regression line
x_range = np.linspace(x.min(), x.max(), 100)
y_hat = w_optimal * x_range

# Save plot
plt.scatter(x, y, color='blue', label='Data')
plt.plot(x_range, y_hat, color='red', label=f'y = {w_optimal:.2f}x')
plt.xlabel('Rainfall (mm)')
plt.ylabel('Crop Yield (tons/ha)')
plt.title('Linear Regression: Rainfall vs. Crop Yield')
plt.legend()
plt.grid(True)
plt.savefig('rainfall_yield_regression.png')
plt.close()

print(f"Slope (w): {w_optimal:.2f}")
```