Learning by Gradient Descent

Introduction
Specific Example from Textbook
General Inputs and Outputs of Learning

Introduction
Useful Sources: Section 28.3 of textbook, Wikipedia articles on Gradient Descent and Machine Learning

Machine learning is concerned with algorithms that can learn from data (or experience) to improve performance.

Assume the performance measure is an error function (or loss, risk, or cost function).

Assume algorithm finds values for parameters and/or constructs a specific kind of data structure such as a neural network or Bayesian network.

Introduction Continued
A hypothesis is a specific assignment to the parameters/data structure.

The typical basic machine learning algorithm performs incremental improvement. That is, it has an initial hypothesis, then makes incremental changes to decrease its performance measure.

Gradient descent is one approach for implementing incremental improvement.

There are many more sophisticated variations, but they generally follow the above ideas.
Textbook Example

As an example of producing a least-squares fit, suppose that we have five data points:

\[
(x_1, y_1) = (-1, 2), \quad (x_2, y_2) = (1, 1), \quad (x_3, y_3) = (2, 1), \quad (x_4, y_4) = (3, 0), \quad (x_5, y_5) = (5, 3),
\]

shown as black dots in Figure 28.3. We wish to fit these points with a quadratic polynomial

\[
F(x) = c_1 + c_2x + c_3x^2.
\]

Matrix Representation

Minimize \( ||AC - Y||^2 \)

\[
A = \begin{pmatrix}
1 & x_1 & x_1^2 \\
1 & x_2 & x_2^2 \\
1 & x_3 & x_3^2 \\
1 & x_4 & x_4^2 \\
1 & x_5 & x_5^2
\end{pmatrix} = \begin{pmatrix}
1 & -1 & 1 \\
1 & 1 & 1 \\
1 & 2 & 4 \\
1 & 3 & 9 \\
1 & 5 & 25
\end{pmatrix}, \quad C = \begin{pmatrix}
1.200 \\
-0.757 \\
0.214
\end{pmatrix}
\]

\[
Y = \begin{pmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4 \\
y_5
\end{pmatrix} = \begin{pmatrix}
2 \\
1 \\
1 \\
0 \\
3
\end{pmatrix}
\]

Illustration of Solution

Gradient Descent Algorithm

\[
\text{Gradient-Descent}(D, W, p)
\]

Inputs \( D \): data
\( W \): weights/coefficients
\( p(D, W) \): performance measure

repeat until convergence

\[ \nabla p = \text{first derivatives of } p(D, W) \text{ wrt } W \]
\( \eta = \text{step size} \)
\( W = W - \eta \nabla p \)

return \( W \)
Gradient Descent Properties

Gradient descent is a greedy algorithm. Certain conditions must be true to converge to a global minimum (or even a local minimum).

More efficient algorithms (conjugate gradient, BFGS) use the gradient in more sophisticated ways.

The type of hypothesis (how the data and the weights are combined to make predictions) and the performance measure (measurement of error and complexity) are the critical elements.