Perceptron Algorithm
By: Eryn Hopps and Natalie Smith

History

• Warren McCulloch and Walter Pitts paper on nervous activity
• Rosenblatt in 1958
  • Study the brain
  • Simple mathematical model
  • Model OR/AND/NOT
• Test by Minsky and Papert in 1969
How It Works

• Dataset = [[2.7,2.5,0], [1.4,2.3,0], [3.3,4.4,0]]

\[ y = \omega_0 + \sum_{i=1}^{n} x_i \omega_i \]

\[ y = 0 + 0 \times 2.7 + 0 \times 2.5 = 0 \]

• If prediction is greater than or equal to 0 then =1 else 0

• Calculate the error

\[ \text{Error} = \text{predicted} - \text{actual} \]

\[ \text{Error} = 0 - 1 = -1 \]

• Recalculate weights
  • \( W_0 = w_0 + \text{lrate} \times \text{error} \)
  • \( W_i = w_i + \text{lrate} \times \text{error} \times x_i \)

\[ W_0 = 0 + 0.1 \times -1 = -0.1 \]

\[ W_i = 0 + 0.1 \times -1 \times 2.7 = -0.27 \]
Python

1. Set $x_0 = 1$; move the branch history into $x_1, ..., x_n$ where $x_n$ is the most recent.
2. If the dot product of $w$ and $x$ is positive, predict 1; else, predict $-1$.
3. When the actual result is determined, add it to $w_0$.
4. For $j = 1, ..., n$, if the actual result is equal to $x_j$, add 1 to $w_j$; else, add $-1$.

Python

```
11 pattern = [-1, 1, -1, -1, 1, 1]
12 history = pattern
13 repeats = 3
14 for i in range(repeats): history.extend(pattern)
```
Python

```python
16 weight = [0 for i in range(train_length + 1)]
17 prediction = [0 for i in range(len(history))]

for i in range(train_length, len(history)):
    x = [1]
    x.extend([history[j] for j in range(i-train_length, i)])
    prediction[i] = 1 if sum(np.multiply(weight, x)) > 0 else -1
    if weight[0] < threshold and history[i] == 1:
        weight[0] += 1
    elif weight[0] > -threshold and history[i] == -1:
        weight[0] += -1
    for j in range(1, len(weight)):
        if history[i] == x[j] and weight[j] < threshold:
            weight[j] += 1
        elif history[i] != x[j] and weight[j] > -threshold:
            weight[j] += -1
```
Python

- High accuracy
- Many patterns: 100%
- Some patterns less
VHDL

• Use switches to input history with an enable button
• Initialized weights to 0
• Multiplying history with weights
• Compare and adjust

Comparison

**Advantages**
• Easy to understand
• Learns quickly
• More accurate than other predictors

**Disadvantages**
• Works for linearly separable data sets only
• Not always 100% accurate
Resources

- https://www.jilp.org/cbp/Daniel-slides.PDF
- Computer Organization and Design by David A Patterson and John L. Hennessy
- https://work.caltech.edu/library/images1/perceptron.png
- https://i.stack.imgur.com/Rxf8G.png