Homework 4: Perceptron

Jef Harkay

November 17, 2011

1 Perceptron

The Perceptron algorithm I implemented was similar to Wikipedia's entry, but I had to first read in the x values and the deltas given in the homework assignment. The deltas are the correct classifications on each line (either a 1 or 0).

```
array(x) ← read(x_values)
array(deltas) ← read(delta_values)
array(weights) ← 0
threshold ← 0
while TRUE do
  adjust ← 0
  for i = 0 → size(x) do
    sum ← 0
    for j = 0 → size(x[i]) do
      sum ← sum + weights[j] * x[i][j]
    end for

    sum ← sum − 12
    if sum > threshold then
      gamma ← 1
    else
      gamma ← 0
    end if

    error ← deltas[i] − gamma
    if error ≠ 0 then
      adjust ← adjust + 1
    end if

    for j = 0 → size(x[i]) do
      weights[j] ← weights[j] + alpha * x[i][j] * error
    end for
  end for
  if adjust is 0 then
    exit
  end if
end while
```

This algorithm shows the linear separator method. For part 2 of the homework, I implemented a logistic regression equation. The only thing that really changes is what the sum is, which becomes:

$$sum = 1/(1 + e^{-(array(weights) * array(x))})$$

The sum is still checked against a threshold, and the weights are computed the same–we just have a new function we're working with.

By tweaking the learning rate, the results can vary quite a bit. To get weight values close to what the professor got, a 0.2 learning rate was used.

2 Results

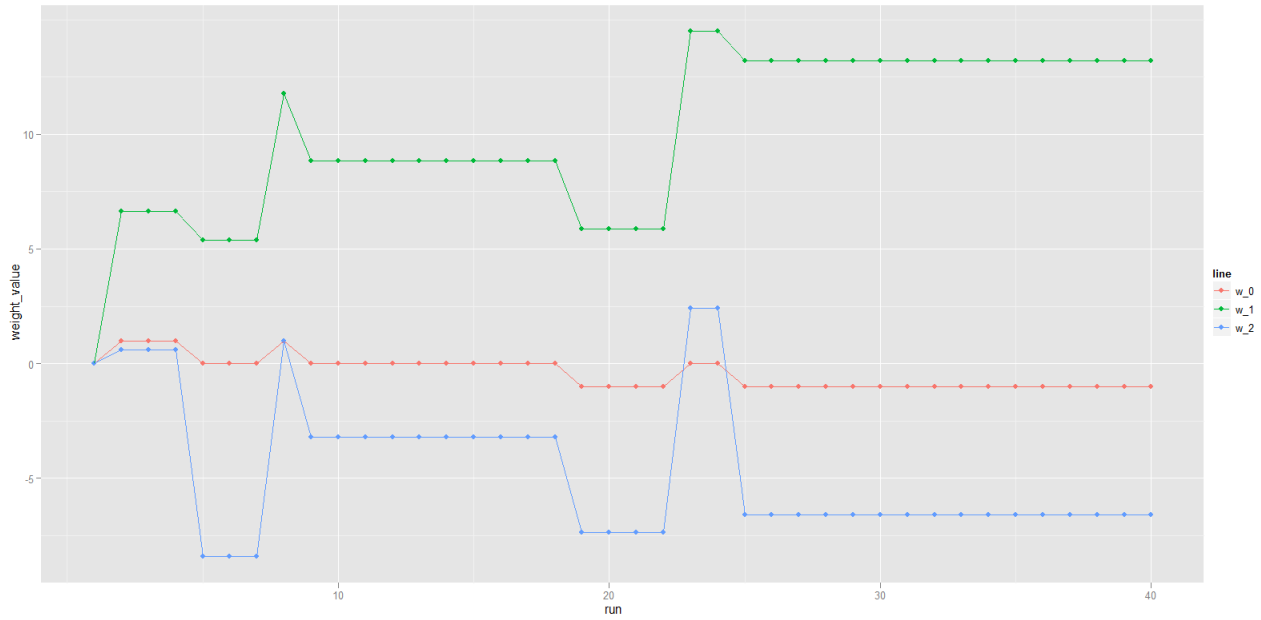

Figure 1: Linear: learning rate = 1

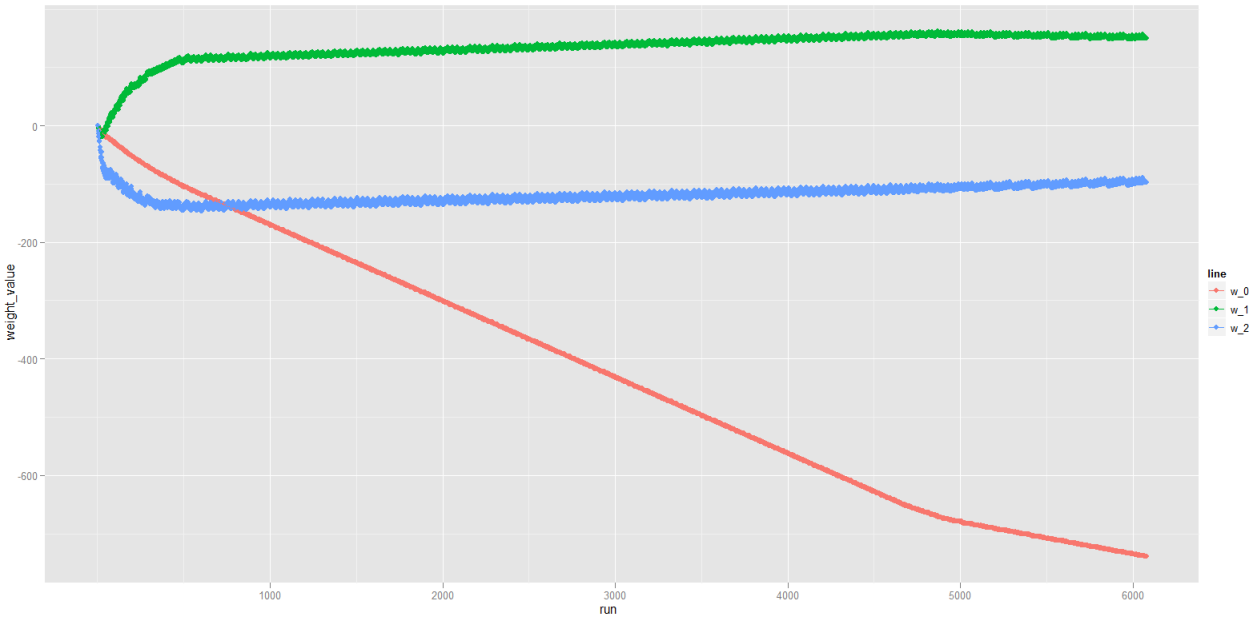

Figure 2: Logistic: learning rate = 1

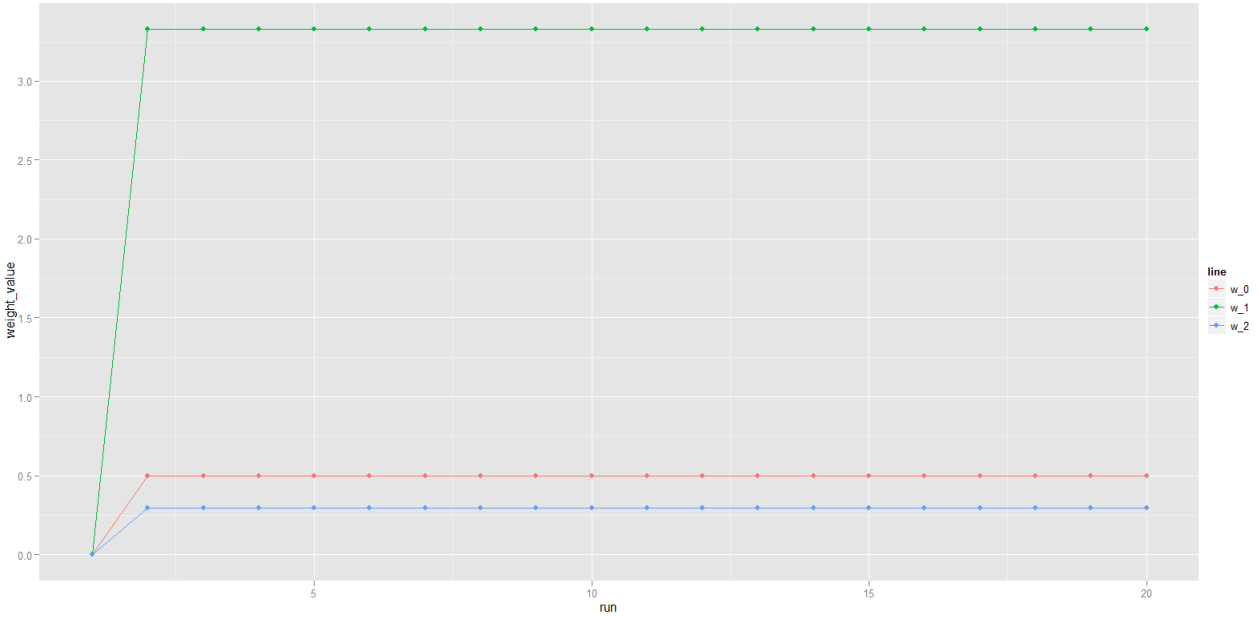

Figure 3: Linear: learning rate = 0.5

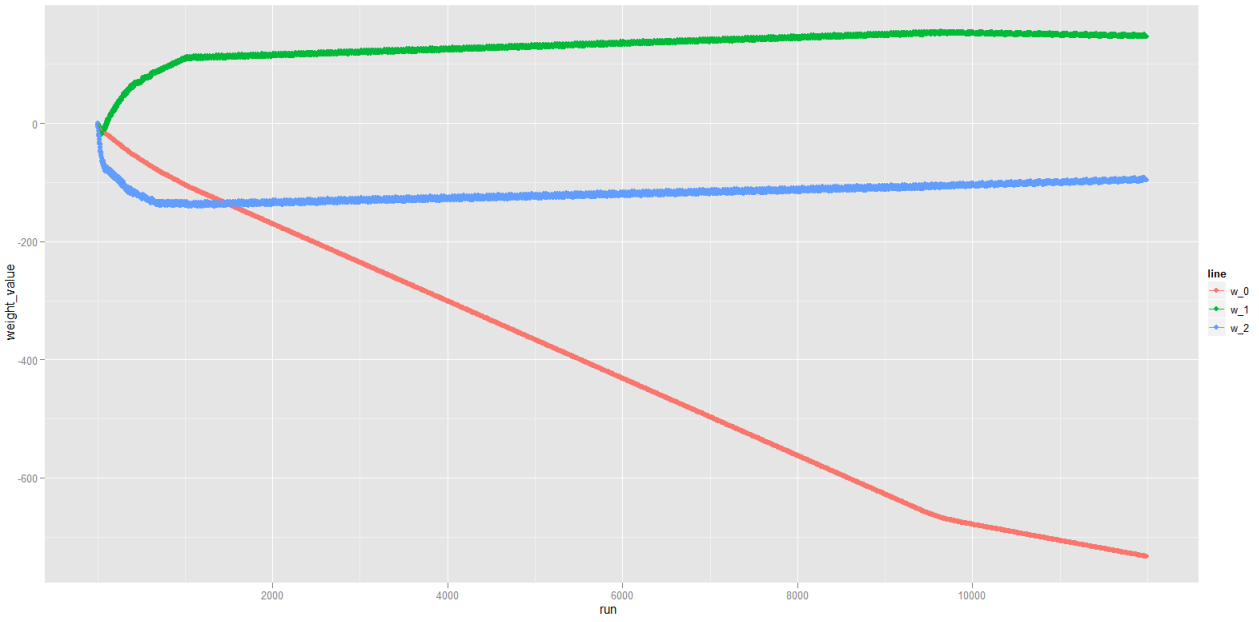

Figure 4: Logistic: learning rate = 0.5

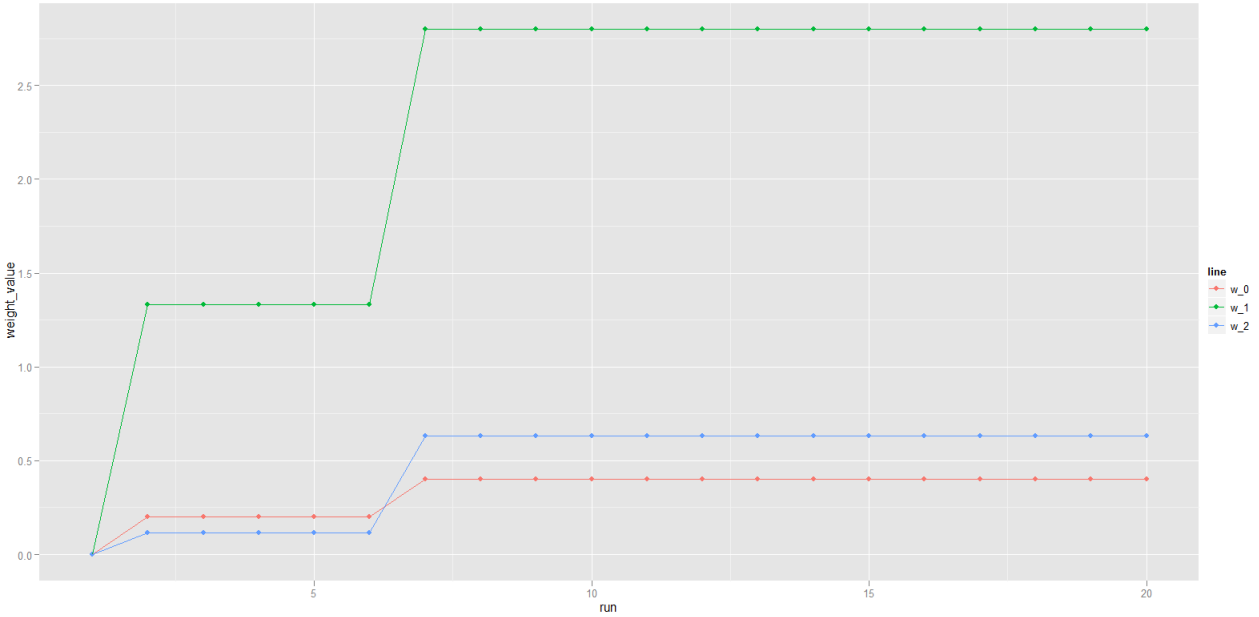

Figure 5: Linear: learning rate = 0.2

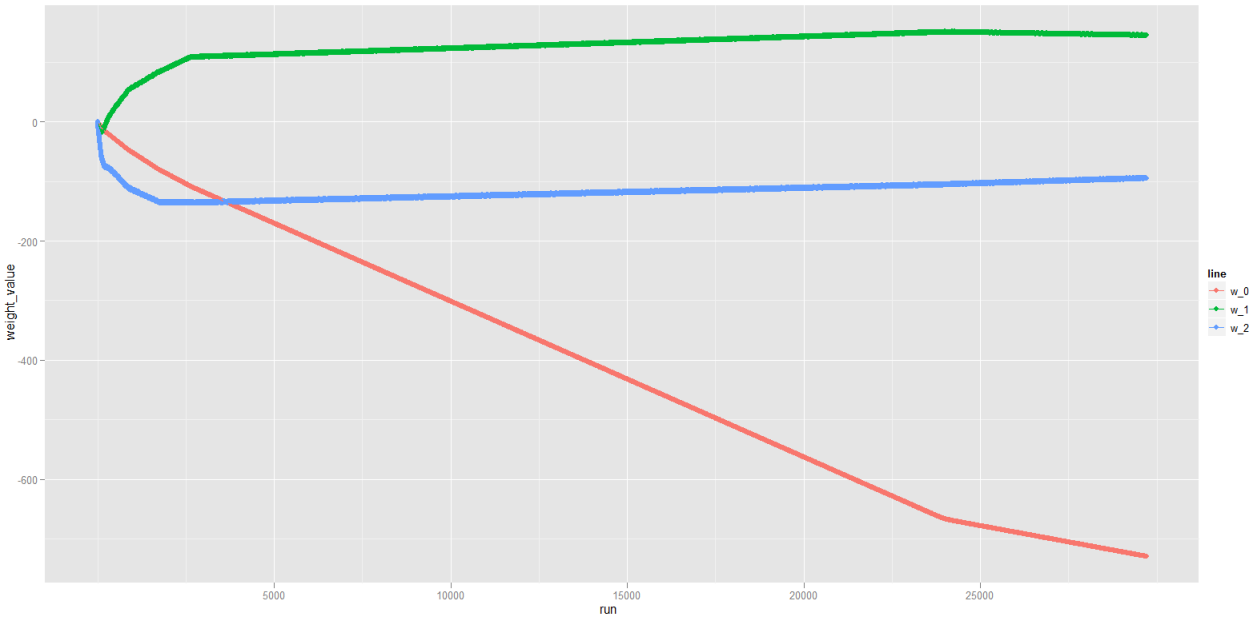

Figure 6: Logistic: learning rate = 0.2

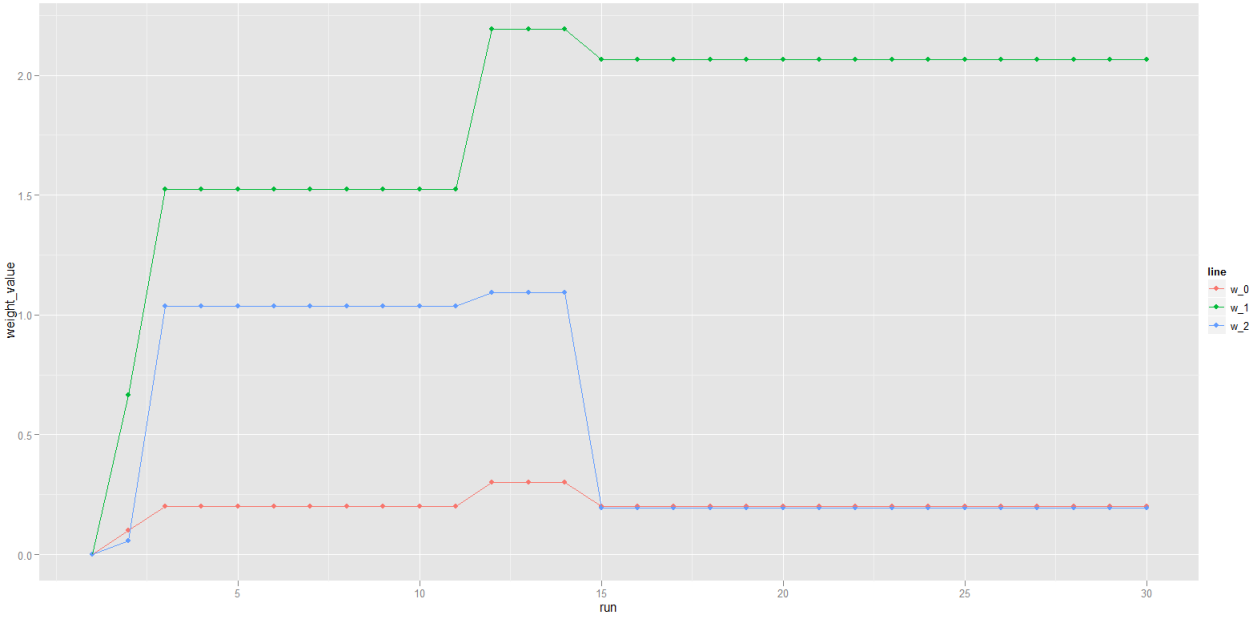

Figure 7: Linear: learning rate = 0.1

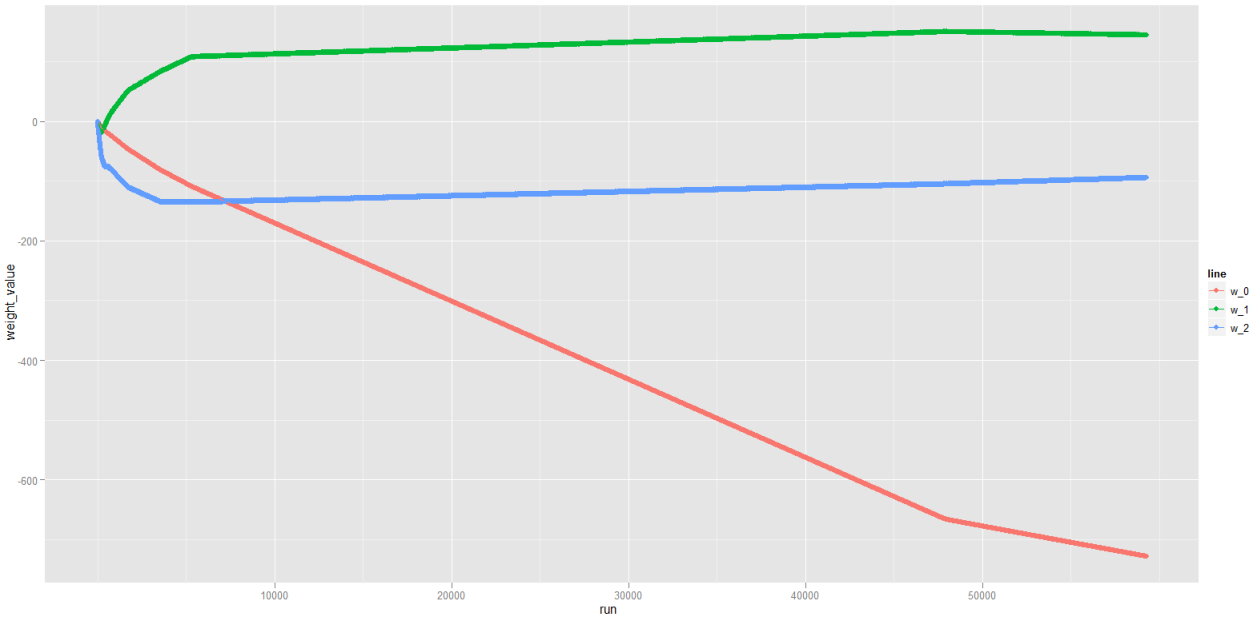

Figure 8: Logistic: learning rate = 0.1

3 Discussion

As you can see, the linear separator function converges a lot quicker than the logistic regression. However, when the learning rate is changed for the linear separator, the results can vary quite a bit. Whereas logistic regression seems to just hone in on one answer. Also, when the learning rate gets smaller for logistic regression, there are a lot more runs involved. I'm assuming that's because it's trying to create more precise weights.

I don't really see the advantage of one over the other with this simple example. I'm thinking if the equation was more complex, logistic regression could solve it no problem, but the linear separator might have a hard time converging to an answer.

4 Perl script

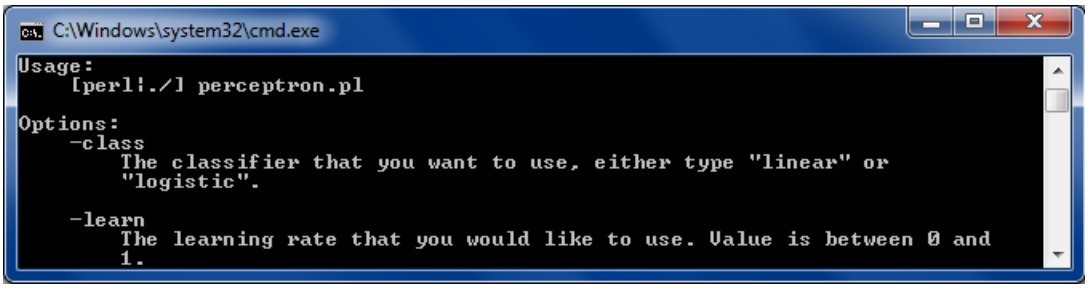

Figure 9: perceptron.pl usage

5 Code