**Perceptron Learning Worksheet**

Algorithm:

1. Initialize starting weights randomly
2. Do until you want to stop (*typically when accuracy is good enough or weights stop changing*):
   1. for each training example (x, y):
      1. use NN to get prediction of h(x)
      2. if h(x) differs from y, update all weights:
      3. w[i] = w[i] + (y – h(x)) \* x[i]
   2. compute accuracy over entire training data = (# predicted correctly)/(# of training examples)

Training data (XOR)

**x1 x2 y**

0 0 0

0 1 1

1 0 1

1 1 0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Epoch | Starting weights | | | Example | | | | Weighted  sum | Predict  h(x) | Error  y – h(x) | Updated weights | | |
|  | w0 | w1 | w2 | x0 (bias) | x1 | x2 | y |  |  |  | w0 | w1 | w2 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 1 | 2 | 3 | 1 | 0 | 0 | 0 |  |  |  |  |  |  |
| 1 |  |  |  | 1 | 0 | 1 | 1 |  |  |  |  |  |  |
| 1 |  |  |  | 1 | 1 | 0 | 1 |  |  |  |  |  |  |
| 1 |  |  |  | 1 | 1 | 1 | 0 |  |  |  |  |  |  |
| 2 |  |  |  | 1 | 0 | 0 | 0 |  |  |  |  |  |  |
| 2 |  |  |  | 1 | 0 | 1 | 1 |  |  |  |  |  |  |
| 2 |  |  |  | 1 | 1 | 0 | 1 |  |  |  |  |  |  |
| 2 |  |  |  | 1 | 1 | 1 | 0 |  |  |  |  |  |  |
| 3 |  |  |  | 1 | 0 | 0 | 0 |  |  |  |  |  |  |
| 3 |  |  |  | 1 | 0 | 1 | 1 |  |  |  |  |  |  |
| 3 |  |  |  | 1 | 1 | 0 | 1 |  |  |  |  |  |  |
| 3 |  |  |  | 1 | 1 | 1 | 0 |  |  |  |  |  |  |