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*):
   a. for each training example (x, y):
      i. use NN to get prediction of h(x)
      ii. if h(x) differs from y, update all weights:
      iii. w[i] = w[i] + (y – h(x)) * x[i]
   b. compute accuracy over entire training data = (# predicted correctly)/(# of training examples)

Training data

Carry

x1	**x2**	**y**
0	0	0
0	1	0
1	0	0
1	1	1

$w + \text{error} \cdot x$

$\begin{pmatrix}1\\2\\3\end{pmatrix} + (-1)\begin{pmatrix}1\\0\\0\end{pmatrix}$

$\begin{pmatrix}1\\2\\3\end{pmatrix} + \begin{pmatrix}-1\\0\\0\end{pmatrix}$

error

$(-1)\begin{pmatrix}1\\0\\1\end{pmatrix}$

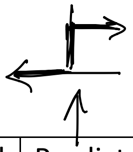
$\begin{pmatrix}0\\2\\3\end{pmatrix} + \begin{pmatrix}-1\\0\\-1\end{pmatrix}$

error

$(-1)\begin{pmatrix}1\\1\\0\end{pmatrix} = \begin{pmatrix}-1\\-1\\0\end{pmatrix}$

if weighted sum $\geq 0 \rightarrow h(x)=1$

$< 0 \rightarrow h(x)=0$



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	-1	0	2	3
1	0	2	3	1	0	1	0	3	1	-1	-1	2	2
1	-1	2	2	1	1	0	0	1	1	-1	-2	1	2
1	-2	1	2	1	1	1	1	1	1	0	"	"	"
2	-2	1	2	1	0	0	0						
2				1	0	1	0						
2				1	1	0	0						
2				1	1	1	1						
3	-2	2	2	1	0	0	0						
3				1	0	1	0						
3				1	1	0	0						
3				1	1	1	1						

4   -3   2   1 → 4/4 accuracy.