Perceptron – A Running Example

The Algorithm

---------------------------------------------------------------------------------------------------------------------

For each data point ***x***

Predict the label *y*’= sgn{***w***T***x***}

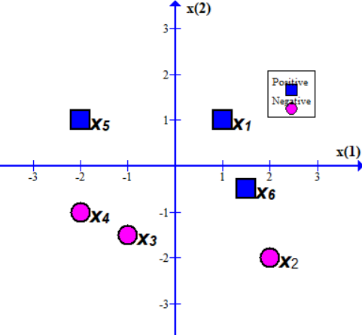
if y’!= *y,* update the weight vector

***w****k*+1 🡨 ***w****k* + *a yi****x****i* (*a* - a constant, learning rate)

else

***w****k*+1 🡨 ***w****k*

---------------------------------------------------------------------------------------------------------------------

We set *a*=0.2

Initial weight vector ***w****0*=[1, 0.5, 0] T

***x****1*=[1, 1, 1] T

***x****2*=[2, -2, 1] T

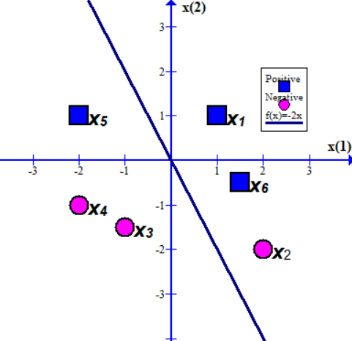
***x****3*=[-1, -1.5, 1] T

***x****4*=[-2, -1, 1] T

***x****5*=[-2, 1, 1] T

***x****6*=[1.5, -0.5, 1] T

---------------------------------------------------------------------------------------------------------------------

***w****0*=[1, 0.5, 0] T

The decision boundary is defined by ***w****0*T ***x***=0

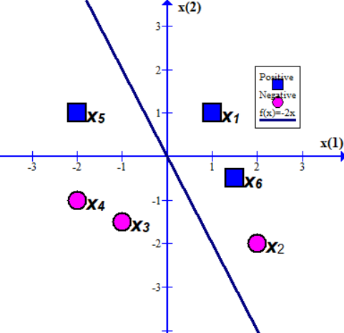
=> ***x*** (1) + 0.5 ***x*** (2) + 0 = 0

=> ***x*** (2) = -2 ***x*** (1)

---------------------------------------------------------------------------------------------------------------------

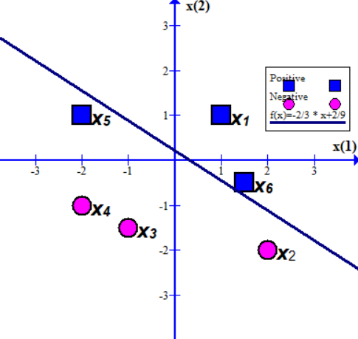
***w****0*=[1, 0.5, 0] T

Now we take the first example ***x****1*=[1, 1, 1] T and *y1*=+1

***w****0*T ***x****1* = 1\*1 + 0.5\*1 + 0\*1 = 1.5 > 0

Correct classification

---------------------------------------------------------------------------------------------------------------------

***w****0*=[1, 0.5, 0] T

Now we take ***x****2*=[2, -2, 1] T and *y2*=-1

***w****0*T ***x****2* = 1\*2 + 0.5\*-2 + 0\*1 = 1 > 0 => Misclassification

We apply the learning rule ***w****k*+1 🡨 ***w****k* + *a yi****x****i*

***w****1*= [1, 0.5, 0] T+0.2\*(-1)\*[2, -2, 1] T

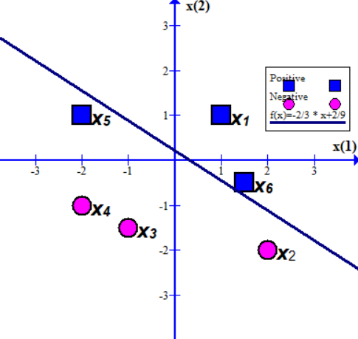
= [0.6, 0.9, -0.2] T

The decision boundary is defined by ***w****1*T ***x***=0

=> 0.6***x*** (1) + 0.9 ***x*** (2) - 0.2 = 0

=> ***x*** (2) = -2/3 ***x*** (1)+2/9

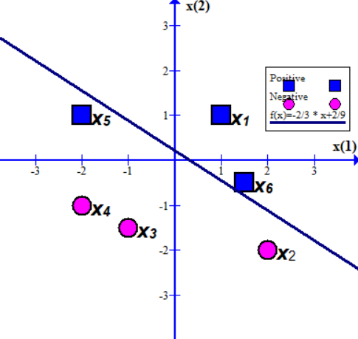
---------------------------------------------------------------------------------------------------------------------

***w****1*= [0.6, 0.9, -0.2] T

Now we take ***x****3*=[-1, -1.5, 1] T and *y3*=-1

***w****1*T ***x****3* = 0.6\*-1 + 0.9\*-1.5 -0.2\*1 < 0 => Correct classification

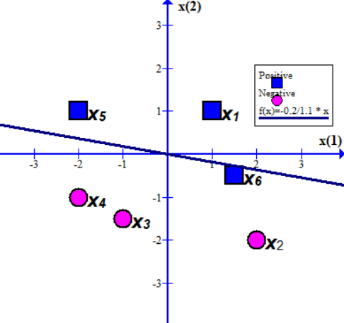
---------------------------------------------------------------------------------------------------------------------

***w****1*= [0.6, 0.9, -0.2] T

Now we take ***x****4*=[-2, -1, 1] T and *y4*=-1

***w****1*T ***x****4* = 0.6\*-2 + 0.9\*-1 -0.2\*1 < 0 => Correct classification

---------------------------------------------------------------------------------------------------------------------

***w****1*= [0.6, 0.9, -0.2] T

Now we take ***x****5*=[-2, 1, 1] Tand *y5*=1

***w****1*T ***x****5* = 0.6\*-2 + 0.9\*1 -0.2\*1 < 0 => Misclassification

We apply the learning rule ***w****k*+1 🡨 ***w****k* + *a yi****x****i*

***w****2*= [0.6, 0.9, -0.2] T+0.2\*1\*[-2, 1, 1] T

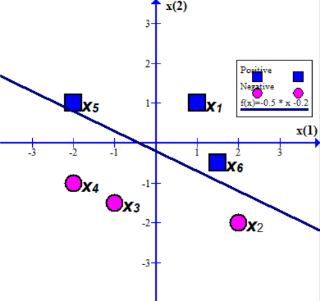
= [0.2, 1.1, 0] T

The decision boundary is defined by ***w****2*T ***x***=0

=> 0.2***x*** (1) + 1.1 ***x*** (2) + 0 = 0

=> ***x*** (2) = -0.2/1.1 ***x*** (1)

---------------------------------------------------------------------------------------------------------------------

***w****2* = [0.2, 1.1, 0] T

Now we take ***x****6*=[1.5, -0.5, 1] T and *y6*=1

***w****2*T ***x****6* = 0.2\*1.5 + 1.1\*-0.5 + 0\*1 = 0.3 – 0.55 < 0 => Misclassification

We apply the learning rule ***w****k*+1 🡨 ***w****k* + *a yi****x****i*

***w****3*= [0.2, 1.1, 0] T+0.2\*1\*[1.5, -0.5, 1] T

= [0.5, 1.0, 0.2] T

The decision boundary is defined by ***w****3*T ***x***=0

=> 0.5***x*** (1) + 1.0 ***x*** (2) + 0.2 = 0

=> ***x*** (2) = -0.5 ***x*** (1) - 0.2

---------------------------------------------------------------------------------------------------------------------

You can try different learning rate and different sequence of examples.