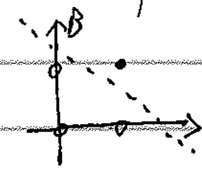
NN-HW

Pengfei Li

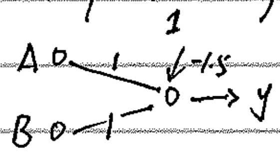
1. This is a possible decision surface of A and B. Empty circles mean ~~TRUE~~ FALSE, and solid ones TRUE.



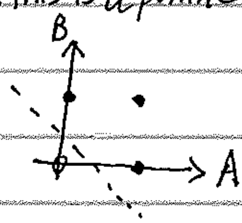
The output is $o(A,B) = sgn(w_0 + w_1 A + w_2 B)$, where the weight w satisfies

$$\begin{cases} w_0 + w_2 < 0 \\ w_0 + w_1 < 0 \\ w_0 + w_1 + w_2 > 0 \end{cases}$$

A ~~poss~~ perceptron with possible weights is

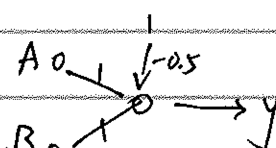


2. This is a possible decision surface of A OR B. The output is $o(A,B) = sgn(w_0 + w_1 A + w_2 B)$, where the weights meet

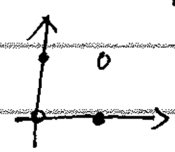


$$\begin{cases} w_0 + w_2 > 0 \\ w_0 + w_1 > 0 \\ w_0 < 0 \end{cases}$$

A perceptron with possible weights is

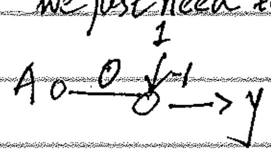


3. There is no possible decision surface for A XOR B, as we assume A is different from B.



If A is the same B. A XOR B = 0 always

we just need to let $w_0 < 0$

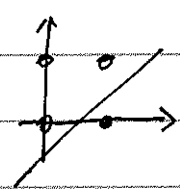


4. • Intialize each w_i in w to some random value
   • Until the termination is met. Do
      • Initialize ~~a~~ each Δw_i to zero
      • For each $<x^{(i)}, t>$ in traning samples. Do
         • Input the instance $x^{(i)}$ to the unit and compute the output
         • $w = w + \Delta w$, $\Delta w = -R\nabla L(w, x^{(i)})$ $\left(L(w,x) = \frac{1}{D}\sum_{i=1}^{D} L(w, x^{(i)}) \right)$

5. Since $1 + 2w_1 + w_2 > 0 \Leftarrow 2w_1 + w_2 > 0$

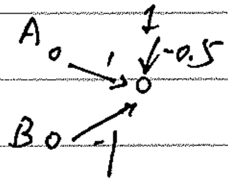
   So A is more general than B

6. I. The decision boundary for $A \wedge \neg B$ is

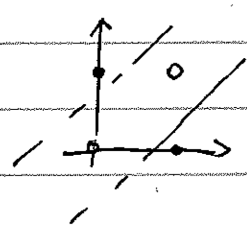


, with weights $\begin{cases} w_0 < 0 \\ w_0 + w_1 + w_2 < 0 \\ w_0 + w_1 > 0 \end{cases}$

A possible perceptron is

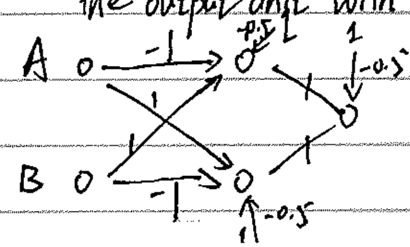


II. A 2-layer network can be drawn if we connect I with problem 2 by $A \text{ XOR } B = (A \wedge \neg B) \vee (\neg A \wedge B)$. ~~two~~ Two lines can be the decision boundaries.

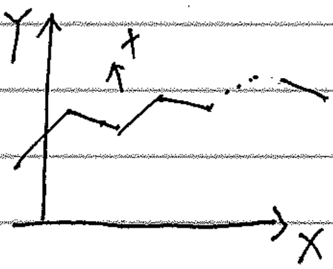


Two hidden units with weights $-0.5, -1, 1$ and $-0.5, 1, -1$

the output unit with weights $-0.5, 1, 1$



7. The problem asks for the simplest decision boundary such a NN can have.
Since we know each perceptron's decision surface is a line, we have n-line at hand.
So the simplest decison surface is an n-piecewise linear boundary.



Say for example above ~~Abov~~ is True, below in False.