ID: 015755428
Kim
Gary

1) a.

Iteration	b	x1	x2	w0	w1	w2	x1 AND x2	z	a(z)	delta(w0)	delta(w1)	delta(w2)
1	1	0	0	1	1	1	0	1	1	-0.4	0	0
	1	1	0	0.6	1	1	0	1.6	1	-0.4	-0.4	0
	1	1	0	0.2	0.6	1	0	0.8	1	-0.4	-0.4	0
	1	1	1	-0.2	0.2	1	1	1	1	0	0	0

All delta(wi) are 0 at the end of iteration 1. Thus, our perceptron is correctly trained. If you try to train the perceptron further by moving on to iteration 2, it will perfectly classify the data and all delta(wi) will remain 0.

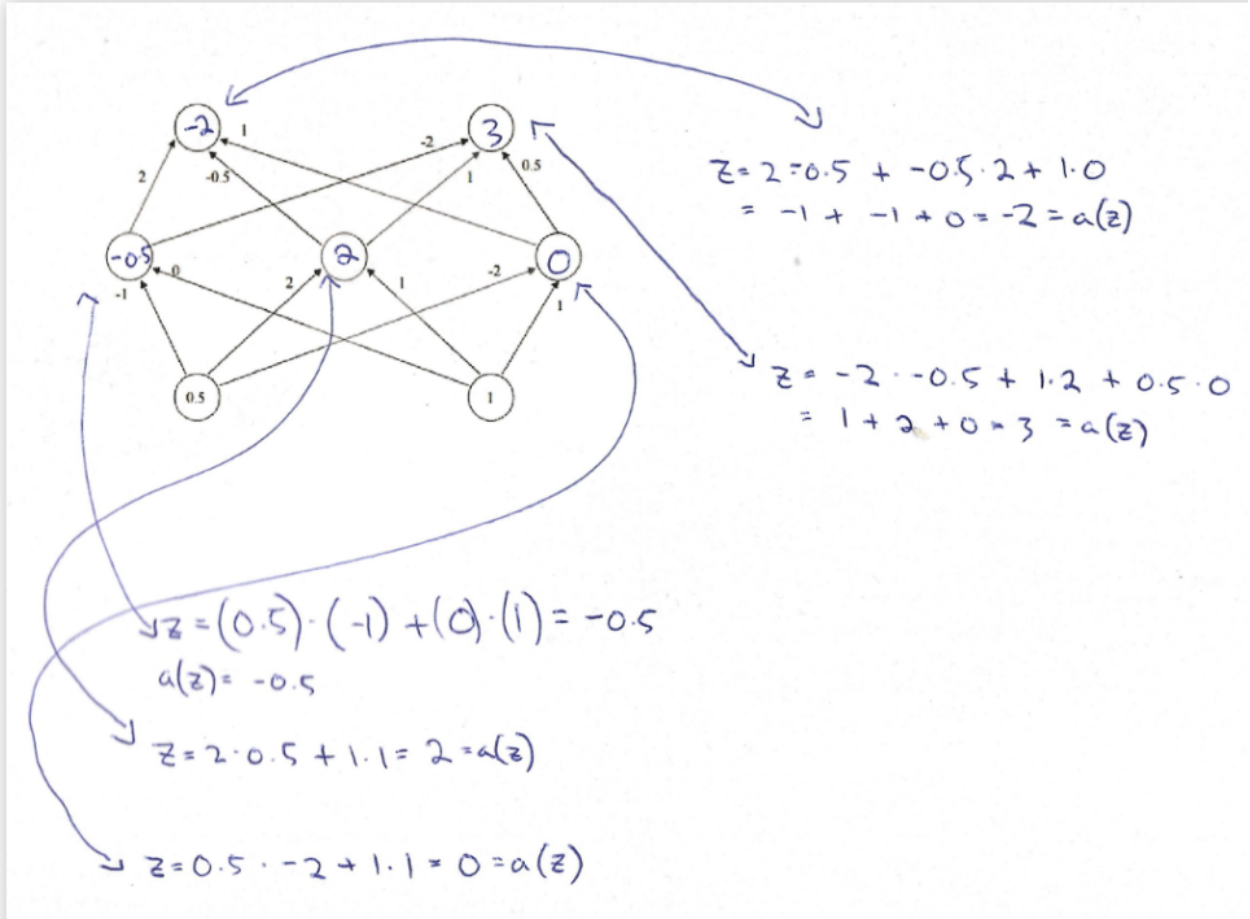
b.

Iteration	b	x1	NOT x1	w0	w1	z	a(z)	delta(w0)	delta(w1)
1	1	0	1	0	0	0	0	0.1	0
	1	1	0	0.1	0	0.1	1	-0.1	-0.1
2	1	0	1	0	-0.1	0	0	0.1	0
	1	1	0	0.1	-0.1	0	0	0	0

All delta(wi) are 0 after iteration 2. Thus, our perceptron is correctly trained.

2) https://github.com/garyjsk271/Machine-Learning.git

3) a.



The output at each node is written inside the node circle. The calculations are shown on the side with the corresponding edges.

b. The calculations do not need to be repeated as there the input is just doubled. So the output is just doubled.
For the 1st layer, going left to right, the outputs are -1, 4, 0.
For the 2nd layer, going left to right, the outputs are -4, 6.
If you repeat the calculations, you will get exactly these results. The reason for this is because of the linear activation function. Since at every layer, all you're doing is taking a sum of the product of the weights and inputs, if inputs are doubled, output is doubled.

4) https://github.com/garyjsk271/Machine-Learning.git

5)

Initial Population (C1=1001001, C2=0100101, C3=1011000, C4=1101100)

C1 = 100 100 1 (Sunny) && (Hot) → Yes
C2 = 010 010 1 (Overcast) && (Mild) → Yes
C3 = 101 100 0 (Sunny || Rain) && (Hot) → No
C4 = 110 110 0 (Sunny || Overcast) && (Hot || Mild) → No

Fitness Calculation

Correct classification of C1 = { D4 }, fitness(C1) = ⅕ = 0.2
Correct classification of C2 = { D1, D4, D5 }, fitness(C2) = ⅗ = 0.6
Correct classification of C3 = { D1, D2, D3, D5 }, fitness(C3) = ⅘ = 0.8
Correct classification of C4 = { D1, D2 }, fitness(C4)= ⅖ = 0.4

1st generation (C1=1001001, C2=0100101, C3=1011000, C4=1101100)

Pr(C1) = 0.2/(0.2 + 0.6 + 0.8 + 0.4) = 0.2/2 = 0.1
Pr(C2) = 0.6/2 = 0.3
Pr(C3) = 0.8/2 = 0.4
Pr(C4) = 0.4/2 = 0.2

Pr(C1) = 0.1 (4th)
Pr(C4) = 0.2 (3rd)
Pr(C2) = 0.3 (2nd)
Pr(C3) = 0.4 (1st)

1st and 3rd (C3, C4) chosen for crossover.
C3 = 101 | 1000
C4 = 110 | 1100
→ 1011100
→ 1101000

2 fittest and offsprings move on to next generation.

C1 = 101 100 0 (Sunny || Rain) && (Hot) → No
C2 = 010 010 1 (Overcast) && (Mild) → Yes
C3 = 101 110 0 (Sunny || Rain) && (Hot || Mild) → No
C4 = 110 100 0 (Sunny || Overcast) && (Hot) → No

Fitness Calculation

Correct classification of C1 = { D1, D2, D3, D5 }, fitness(C1) = ⅘ = 0.8
Correct classification of C2 = { D1, D4, D5 }, fitness(C2) = ⅗ = 0.6
Correct classification of C3 = { D1, D2, D3, D5}, fitness(C3) = ⅘ = 0.8
Correct classification of C4 = { D1, D2, D5}, fitness(C4)= ⅗ = 0.6

2nd Generation (C1 = 1011000, C2 = 0100101, C3 = 1011100, C4 = 1101000)
Pr(C1) = 0.8/(0.8 + 0.6 + 0.8 + 0.6) = 0.8/2.8 = 0.286
Pr(C2) = 0.6/2.8 = 0.214
Pr(C3) = 0.8/2.8 = 0.286
Pr(C4) = 0.6/2.8 = 0.214

Pr(C2) = 0.214 (4th)
Pr(C4) = 0.214 (3rd)
Pr(C3) = 0.286 (2nd)
Pr(C1) = 0.286 (1st)

1st and 2nd chosen for crossover for 2nd generation.
C1 = 101 | 10 | 00
C3 = 101 | 11 | 00
→ 1011000 (6th bit mutated) → 1011010
→ 1011100

2 fittest and offsprings move on to next generation.

C1 = 101 100 0	(sunny || rain) && (hot) → no
C2 = 101 110 0	(sunny || rain) && (hot || mild) → no
C3 = 101 101 0	(sunny || rain) && (hot || cool) → no
C4 = 101 110 0	(sunny || rain) && (hot || mild) → no

Fitness Calculation
Correct classification of C1 = {D1, D2, D3, D5}, fitness(C1) = 0.8
Correct classification of C2 = { D1, D2, D3, D5}, fitness(C2) = ⅘ = 0.8
Correct classification of C3 = { D4 }, fitness(C3) = ⅕ = 0.2
Correct classification of C4 = {D1, D2, D3, D4, D5}, fitness(C4) = 1.0

GA terminated since C4's accuracy is 1.0.

Model returned is (Sunny || Rain) && (hot || mild) → PlayTennis = no

6)

Initial population (000, 001, 010, 100)

Fitness Calculations
F(C1) = 570 * 0 + 710 * 0 + 640 * 0 = 0
F(C2) = 640 * 1 = 640
F(C3) = 710 * 1 = 710
F(C4) = 570 * 1 = 570

1st Generation (000, 001, 010, 100)

Pr(C1) = 0/(640+710+570) = 0/1920 = 0.0
Pr(C2) = 640/1920 = 0.33
Pr(C3) = 710/1920 = 0.37
Pr(C4) = 570/1920 = 0.30

Pr(C1) = 0.0 (4th)
Pr(C4) = 0.30 (3rd)
Pr(C2) = 0.33 (2nd)
Pr(C3) = 0.37 (1st)

2nd and 3rd crossover
C2 = 00 | 1
C4 = 10 | 0
→ 000
→ 101
Constraint check:
5 * 0 + 8 * 0 + 10 * 0 <= 15
1 * 5 + 10 * 1 = 15 <= 15

2nd and 1st crossover
C2 = 00 | 1
C3 = 01 | 0
→ 000
→ 011
Constraint check:
5 * 0 + 8 * 0 + 10 * 0 <= 15
1 * 8 + 10 * 1 = 18 !<= 15
011 replicates C3 = 010

C1 = 000, C2 = 101, C3 = 000, C4 = 010

F(C1) = 0
F(C2) = 570 * 1 + 640 * 1 = 1210
F(C3) = 0
F(C4) = 710 * 1 = 710

2nd Generation (C1 = 000, C2 = 101, C3 = 000, C4 = 010)

Pr(C1) = 0/1920 = 0.0 (4th)
Pr(C3) = 0/1920 = 0.0 (3rd)
Pr(C4) = 710/1920 = 0.37 (2nd)
Pr(C2) = 1210/1920 = 0.63 (1st)

1st and 1st crossover
C4 = 1 | 01
C4 = 1 | 01
→ 101 (3rd bit mutation) → 100
→ 101 (3rd bit mutation) → 100
Constraint check:
5 <= 15
5 <= 15

1st and 2nd crossover
C2 = 1 | 01
C4 = 0 | 10
→ 110
→ 001
Constraint check:
13 < 15
8 < 15

C1 = 100, C2 = 100, C3 = 110, C4 = 001

F(C1) = 570
F(C2) = 570
F(C3) = 570 + 710 = 1280
F(C4) = 640

3rd Generation (C1 = 010, C2 = 010, C3 = 001, C4 = 010)

Terminate in 3rd Generation.

Return best chromosome = C3 = 110