Shreyas Angara

Period 7

4/21/17

|  |  |
| --- | --- |
| *#Shreyas Angara Period 7 4/21/17* import numpy as np class Percept:  def \_\_init\_\_(self, weights, threshold):  self.w = np.array(weights)  self.threshold = threshold  def step(self, dot):  if dot > self.threshold: return 1  return 0  def eval(self):  inputValues = []  for i in self.inputs:  inputValues.append(i.eval())  return self.step(np.dot(self.w, inputValues))  def set\_inputs(self, l):  self.inputs = np.array(l) class Input(Percept):  def \_\_init\_\_(self):  pass  def set\_value(self, i):  self.value = i  def eval(self):  return self.value  x1 = Input() x2 = Input() AND = Percept([1, 1], 1.5) OR = Percept([1, 1], .5) NAND = Percept([-1, -1], -1.5) NAND2 = Percept([-1, -1], -1.5) *# NAND.set\_inputs([x1, x2]) # OR.set\_inputs([x1, x2]) # AND.set\_inputs([NAND, OR]) # xor = AND* OR.set\_inputs([x1,x2]) NAND.set\_inputs([x1,x2]) NAND2.set\_inputs([OR, NAND]) xnor = NAND2 for a in range(2):  for b in range(2):  x1.set\_value(a)  x2.set\_value(b)  print(a, b, xnor.eval()) | XOR  0 0 0  0 1 1  1 0 1  1 1 0  XNOR  0 0 1  0 1 0  1 0 0  1 1 1 |