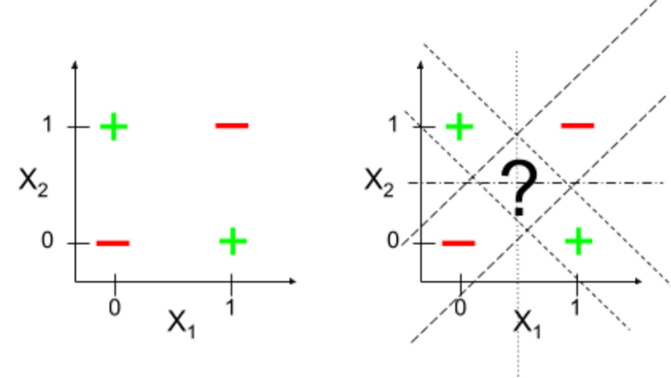
**Nama:** Limas Jaya Akeh | **NIM:** 2201763240 | **Kelas:** LA-01 | Artificial Neural Network

**XOR Problem**

1. Please solve XOR problem using BPNN, where the input patterns are (0,0), (0,1), (1,1), and (1,0) with target (0,1,0,1). XOR problem is not linearly separable. It should not be solved with single layer neural networks!



Please design BPNN and tune its hyperparameter to solve XOR problem by:

1. Showing your code
2. Screen shoot the result

**Answer**

Single-Layer Neural Network adalah sebuah jaringan terdiri dari satu layer input dan satu layer output, terdapat bobot dari tiap node dalam suatu layer input sehingga mencoba untuk “mendekati” jawaban yang diinginkan. Single-Layer Neural Network biasanya menyelesaikan masalah yang **Linearly separable.** (Dapat klasifikasi data dalam satu garis lurus).

A picture containing object, clock

Description automatically generated

Dalam Single-Layer Neural-Network, kita dapat menyelesaikan beberapa masalah menggunakan perubahan weight dari tiap perceptron.

Coding dari Single Layer Neural Network:

// Codingan Single Layer Neural Network

import numpy as np

def nonlin(x,deriv=False):

if(deriv==True):

return x\*(1-x)

return 1/(1+np.exp(-x))

X = np.array([ [0,0],

[0,1],

[1,1],

[1,0] ])

y = np.array([[0,1,0,1]]).T

np.random.seed(1)

syn0 = 2\*np.random.random((2,1)) - 1

for iter in range(10000):

l0 = X

l1 = nonlin(np.dot(l0,syn0))

l1\_error = y - l1

l1\_delta = l1\_error \* nonlin(l1,True)

syn0 += np.dot(l0.T,l1\_delta)

print("Output After Training:")

print(l1)

A screenshot of a cell phone

Description automatically generated

Dari contoh diatas, output setelah training adalah weight yang berbeda sehingga “semakin dekat” ke target. (Local Optimum), jawaban dari tiap weight sudah “mendekati” target yang diinginkan dan sudah cukup baik.

A screenshot of a cell phone

Description automatically generated

Dalam contoh ini, ketika semua output menghasilkan 1 semua, output yang dihasilkan masih cukup baik dan terupdate sesuai goal yang diinginkan.

Akan tetapi, apabila target diganti menjadi [0,1,0,1],

A screenshot of a cell phone

Description automatically generated

Maka bobot tiap perceptron tetap 0.5, sehingga model tidak dapat menjelaskan output yang diinginkan dari input yang diberikan, masalah ini disebut XOR Problem.

A screenshot of a cell phone

Description automatically generated

Dalam XOR Problem, output akan mengeluarkan true (atau 1) apabila kedua value (input) berbeda, dan dalam Single-layer Neural Network, **arsitektur model hanya dapat membagi data dalam sebuah garis.** Akan tetapi, XOR Problem merupakan sebuah masalah non-linearly separable (tidak dapat dibagi hanya dengan sebuah garis lurus).

XOR Problem tidak dapat dibuktikan menggunakan Single-Layer Neural Network, sehingga solusi yang dapat digunakan adalah menggunakan Multi-Layer Neural Network.

**Multi-Layer Neural Network**

A picture containing ball, room

Description automatically generated

Multi-Layer Neural Network adalah jaringan yang terdiri dari **setidaknya** 3 layer; satu layer input, satu layer output, dan terdapat layer diantaranya yang disebut **hidden layer(s).** Multi-Layer Neural Network dapat membedakan data yang not linearly separable. Dalam Multi-Layer Neural Network, ketika hasil tidak sesuai dari target yang diinginkan, hidden layer dapat memodifikasi weight dari tiap hidden node yang ada dalam hidden layer.

XOR Problem adalah salah satu masalah yang dapat diselesaikan dengan Multi-Layer Neural Network.

Coding dari Multi-Layer Neural Network sebagai berikut:

// Codingan Multi-Layer Neural Network

import numpy as np

def nonlin(x,deriv=False):

if(deriv==True):

return x\*(1-x)

return 1/(1+np.exp(-x))

X = np.array([ [0,0],

[0,1],

[1,1],

[1,0] ])

y = np.array([[0,1,0,1]]).T

np.random.seed(1)

syn0 = 2\*np.random.random((2,4)) - 1

syn1 = 2\*np.random.random((4,1)) - 1

for iter in range(50000):

l0 = X

l1 = nonlin(np.dot(l0,syn0))

l2 = nonlin(np.dot(l1, syn1))

l2\_error = y - l2

if(iter % 10000) == 0:

print ("Error : "+ str(np.mean(np.abs(l2\_error))))

l2\_delta = l2\_error \* nonlin(l2,True)

l1\_error = l2\_delta.dot(syn1.T)

l1\_delta = l1\_error \* nonlin(l1,True)

syn1 += np.dot(l1.T,l2\_delta)

syn0 += np.dot(l0.T,l1\_delta)

print("Output After Training:")

print(l2)

A screenshot of a cell phone

Description automatically generated