**Practical assignment 3 (Learning multiple weights at a time)**

Please modify the Python code snippet on page 86 of the Grokking Deep Learning textbook as follows. Add a fourth input *humidity* with the following four input values: [0.85, 0.60, 0.95, 0.75]. Also, add a fourth weight to the neural network with an initial value of 0.5. Learn the weights of the neural network not only with the first input of each input variable, but with each of the four inputs of each input variable. Finally, learn the weights for 10 iterations.

def neural\_network(input, weights):

out = 0

for i in range(len(input)):

out += (input[i] \* weights[i])

return out

def ele\_mul(scalar, vector):

out = [0,0,0]

for i in range(len(out)):

out[i] = vector[i] \* scalar

return out

toes = [8.5, 9.5, 9.9, 9.0]

wlrec = [0.65, 0.8, 0.8, 0.9]

nfans = [1.2, 1.3, 0.5, 1.0]

win\_or\_lose\_binary = [1, 1, 0, 1]

true = win\_or\_lose\_binary[0]

alpha = 0.01

weights = [0.1, 0.2, -.1]

input = [toes[0],wlrec[0],nfans[0]]

for iter in range(3):

pred = neural\_network(input,weights)

error = (pred - true) \*\* 2

delta = pred - true

weight\_deltas=ele\_mul(delta,input)

print("Iteration:" + str(iter+1))

print("Pred:" + str(pred))

print("Error:" + str(error))

print("Delta:" + str(delta))

print("Weights:" + str(weights))

print("Weight\_Deltas:")

print(str(weight\_deltas))

print()

for i in range(len(weights)):

weights[i]-=alpha\*weight\_deltas[i]