**Practical assignment 1 (Hot and cold learning on page 54 of the Grokking Deep Learning textbook)**

Hot and cold learning is perhaps the simplest form of learning. After making a prediction, you predict two more times, once with a slightly higher weight and again with a slightly lower weight. You then move weight depending on which direction gave a smaller error. Repeating this enough times eventually reduces the error to 0.

In the Python code snippet below, you must determine a value for the step\_amount variable between and and a value for the number\_of\_iterations variable between 1 and 100000 so that the prediction value would be as close to the goal\_prediction value as possible. You must also write a Microsoft Word report of no more than 1000 words describing your strategy to determine these two values. Please cite reputable sources to motivate your strategy. Submit a single ZIP file on eFundi containing two documents: a Python (.py) text file with your two values for the step\_amount variable and the number\_of\_iterations variable entered into the code snippet below and a Microsoft Word document describing your strategy.

def hot\_and\_cold\_learning(): # Do not modify the function name

step\_amount = 0.001 # Choose a step\_amount here between 1e-6 and 1e-1

number\_of\_iterations = 1101 # Choose the number of iterations between 1 and 100000

### Do not modify the code from here ###

weight = 0.25

input = 0.75

goal\_prediction = 1.0

for iteration in range(number\_of\_iterations):

prediction = input \* weight

error = (prediction - goal\_prediction) \*\* 2

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

up\_prediction = input \* (weight + step\_amount)

up\_error = (goal\_prediction - up\_prediction) \*\* 2

down\_prediction = input \* (weight - step\_amount)

down\_error = (goal\_prediction - down\_prediction) \*\* 2

if(down\_error < up\_error):

weight = weight - step\_amount

if(down\_error > up\_error):

weight = weight + step\_amount

return abs(prediction - goal\_prediction)

### Do not modify the code to here ###

print(hot\_and\_cold\_learning())