1. Consider a learning task where we want to fit a model to a set of 10,000 labeled training cases.

Assume that we have a database with data of the form

Table 1: example of data

x1	x2	y
4	1	2
2	8	-14
1	0	1
3	2	-1
1	4	7
6	7	-8

We want to use TensorFlow to create a classifier for this data. We will develop a model of the form:

$$\hat{y} = w_1 x_1 + w_2 x_2 + b.$$

Based on the data, our task is to compute the labels for the parameters W= [w_1 w_2] and b. Our objective is to train a neural network to minimise the loss function L= $(\hat{y} - y)^2$.

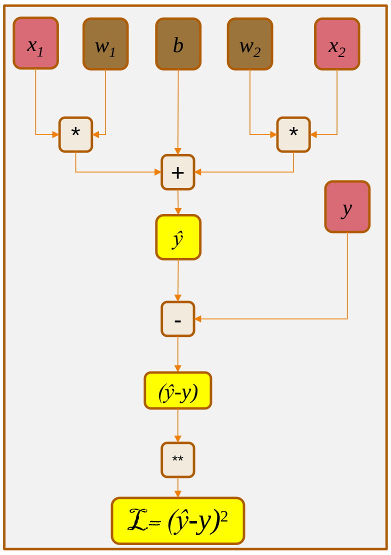
The update formula for any parameter p is given by $p' = p - \eta\ \partial L/\partial p$,

where

- p—current value
- p'—value after update
- η —learning rate, set to 0.05
- $\partial L/\partial p$—gradient, i.e. partial differential of L w.r.t. b

We use backpropagation to update the weights W= [w_1 w_2] and b. We initialize our weights randomly such that W= [w_1 w_2]= [-0.017 -0.048] and b=0.

a) Draw the computation graph for this model.



b) If we assign a CPU to every operator in the computation graph, state the assignment of the computation graph entities to CPUs.

We assign CPUs to the operations *, *, +, - **.

c) For the first data item (4, 1, 2) in Table 1, compute the loss.

$$
\begin{aligned}
\hat{y} &= w_1x_1 + w_2x_2 + b \\
&= (-0.017)\cdot 4 + (-0.048)\cdot 1 + 0.000 \\
&= -0.116
\end{aligned}
$$

$$
\begin{aligned}
L &= (\hat{y} - y)^2 \\
&= (-0.116 - 2)^2 \\
&= 4.48
\end{aligned}
$$

d) Compute the partial derivatives necessary to update the weights.

$$\hat{y} = w_1x_1 + w_2x_2 + b \qquad \frac{\partial \hat{y}}{\partial w_1} = x_1 \qquad \frac{\partial \hat{y}}{\partial w_2} = x_2 \qquad \frac{\partial \hat{y}}{\partial b} = 1$$

$$L = (\hat{y} - y)^2 \qquad \frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y)$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial b}$$

e) Use the update formula to update the weights W= [w_1 w_2] and b.

$$
\begin{aligned}
w_1' &= w_1 - \eta \frac{\partial L}{\partial w_1} \\
&= w_1 - \eta\left(\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_1}\right) \\
&= w_1 - \eta[2(\hat{y} - y)\cdot x_1] \\
&= -0.017 - 0.05[2(-0.116 - 2)\cdot 4] \\
&= 0.829
\end{aligned}
$$

$$
\begin{aligned}
w_2' &= w_2 - \eta \frac{\partial L}{\partial w_2} \\
&= w_2 - \eta\left(\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_2}\right) \\
&= w_2 - \eta[2(\hat{y} - y)\cdot x_2] \\
&= -0.048 - 0.05[2(-0.116 - 2)\cdot 1] \\
&= 0.164
\end{aligned}
$$

$$
\begin{aligned}
b' &= b - \eta \frac{\partial L}{\partial b} \\
&= b - \eta\left(\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial b}\right) \\
&= b - \eta[2(\hat{y} - y)\cdot 1] \\
&= 0.000 - 0.05[2(-0.116 - 2)\cdot 1] \\
&= 0.212
\end{aligned}
$$

1. Consider a medical diagnosis scenario where we have 100,000 brain scan images, each of 1Mx1M pixels, and we want to develop a classifier that identifies either normal scans or specifies 1 of 9 possible diseases (i.e., y is a 10-vector output). We assume a linear model, where y=WX+b, where W is a weight matrix and b a bias vector. We must decompose image analysis into image patches of size 10,000x10,000.
   a. Describe a computation-graph framework for image classification, defining the graph required.
   b. Define the assignment of CPU/GPU to this task. Assume that we have a data centre with a cluster of GPUs with 1,000 processors per GPU, and a cluster of 10,000CPUs.
   c. Compare and contrast how MapReduce might solve this problem.