

# TP02-LinearRegression

September 18, 2019

## 1 Linear Regression

Observe and run the following code. It performs a linear regression over an artificial dataset.

```
[1]: import numpy as np
import tensorflow as tf

def generate_all_dataset():
    # --- Fake dataset ---

    np.random.seed(0)

    ntrain = 1000
    nvalid = 100
    ntest = 100

    # number of features
    nc = 4
    # number of outputs
    no = 2

    A = np.random.randn(nc, no)
    b = np.random.randn(no)

    print("True A\n", A)
    print("True b\n", b)

    def generate_one_dataset(A, b, n):
        nc, no = A.shape
        X = np.random.randn(n, nc)
        Y = X.dot(A)+b
        return np.array(X, dtype='float32'), np.array(Y, dtype='float32')

    Xtrain, Ytrain = generate_one_dataset(A, b, ntrain)
    Xvalid, Yvalid = generate_one_dataset(A, b, nvalid)
    Xtest, Ytest = generate_one_dataset(A, b, ntest)
```

```

return (Xtrain, Ytrain), (Xvalid, Yvalid), (Xtest, Ytest)

def linear_regression_v1(trainset, validset, testset):

    Xtrain, Ytrain = trainset
    Xvalid, Yvalid = validset
    Xtest, Ytest = testset

    # --- Linear Regression ---

    # NEW ! reset tensorflow
    tf.reset_default_graph()

    # Hyperparameters
    learning_rate = 5e-2
    training_epochs = 100

    # Xtrain (nxtrain,nc)
    nxtrain, nc = Xtrain.shape
    nptest, _ = Xtest.shape
    # Ytrain (nxtrain,no)
    no = Ytrain.shape[1]

    # Placeholders for exchanging between computer memory
    # and tensorflow compute engine.
    # We put None on teh shape where the value is unkoun
    # (here the number of exemples)
    x = tf.placeholder(tf.float32, [None, nc])
    y = tf.placeholder(tf.float32, [None, no])

    # Linear regression parameters
    A = tf.Variable(tf.zeros([nc, no]))
    b = tf.Variable(tf.zeros([no]))
    # All parameters are gathered into var_list
    var_list = [A, b]

    # Actual linear regression
    pred = tf.matmul(x, A) + b

    # Model loss
    loss = tf.losses.mean_squared_error(y, pred)

    # NEW ! Optimizer based on gradient descent
    # It computes gradients of loss over var_list items
    # and construct the updates for var_list items.

```

```

# We don't have to write the gradient step by ourselves
optimizer = tf.train.GradientDescentOptimizer(
    learning_rate).minimize(loss, var_list=var_list)

# NEW ! Initializer for all the variables
# Equivalent to A.initializer + b.initializer
init_global = tf.initializers.global_variables()

# NEW ! All the operations will be run in a unique
# session that will be closed at the end of the calculus
with tf.Session() as session:

    # We call the initialization of A and b
    session.run(init_global)

    # We compute the train and validation loss
    # Note that you just have to change the feed_dict to change the set
    trainloss, = session.run([loss], feed_dict={x: Xtrain, y: Ytrain})
    validloss, = session.run([loss], feed_dict={x: Xvalid, y: Yvalid})
    print("Init\t\t train loss %f\t valid loss %f" %
          (trainloss, validloss))

    # We cycle on epochs
    for epoch in range(training_epochs):
        # We do the gradient step and compute the train loss
        _, trainloss, = session.run(
            [optimizer, loss], feed_dict={x: Xtrain, y: Ytrain})
        # We compute the valid loss
        validloss, = session.run([loss], feed_dict={x: Xvalid, y: Yvalid})

        print("Epoch %03d\t train loss %f\t valid loss %f" %
              (epoch, trainloss, validloss))

    # We compute the test loss
    testloss, = session.run([loss], feed_dict={x: Xtest, y: Ytest})

    print("Test loss %f" % (testloss,))

    # Found parameters
    Aval, bval = session.run(var_list)
    print("Estimated A\n", Aval)
    print('Estimated b\n', bval)
# Here session is closed automatically

linear_regression_v1(*generate_all_dataset())

```

True A

```
[[ 1.76405235  0.40015721]
 [ 0.97873798  2.2408932 ]
 [ 1.86755799 -0.97727788]
 [ 0.95008842 -0.15135721]]
```

True b

```
[-0.10321885  0.4105985 ]
```

WARNING:tensorflow:From /home/rherault/.local/venvs/spyder/lib/python3.7/site-packages/tensorflow/python/framework/op\_def\_library.py:263: colocate\_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.

Instructions for updating:

Colocations handled automatically by placer.

WARNING:tensorflow:From /home/rherault/.local/venvs/spyder/lib/python3.7/site-packages/tensorflow/python/ops/losses/losses\_impl.py:667: to\_float (from tensorflow.python.ops.math\_ops) is deprecated and will be removed in a future version.

Instructions for updating:

Use tf.cast instead.

Init	train loss	7.184080	valid loss	7.740232
Epoch 000	train loss	7.184080	valid loss	7.006136
Epoch 001	train loss	6.501032	valid loss	6.341899
Epoch 002	train loss	5.883163	valid loss	5.740851
Epoch 003	train loss	5.324234	valid loss	5.196962
Epoch 004	train loss	4.818598	valid loss	4.704776
Epoch 005	train loss	4.361157	valid loss	4.259362
Epoch 006	train loss	3.947302	valid loss	3.856260
Epoch 007	train loss	3.572863	valid loss	3.491436
Epoch 008	train loss	3.234073	valid loss	3.161245
Epoch 009	train loss	2.927526	valid loss	2.862387
Epoch 010	train loss	2.650142	valid loss	2.591878
Epoch 011	train loss	2.399136	valid loss	2.347021
Epoch 012	train loss	2.171991	valid loss	2.125374
Epoch 013	train loss	1.966431	valid loss	1.924729
Epoch 014	train loss	1.780396	valid loss	1.743091
Epoch 015	train loss	1.612025	valid loss	1.578652
Epoch 016	train loss	1.459636	valid loss	1.429779
Epoch 017	train loss	1.321705	valid loss	1.294993
Epoch 018	train loss	1.196856	valid loss	1.172956
Epoch 019	train loss	1.083843	valid loss	1.062458
Epoch 020	train loss	0.981541	valid loss	0.962405
Epoch 021	train loss	0.888930	valid loss	0.871806
Epoch 022	train loss	0.805089	valid loss	0.789764
Epoch 023	train loss	0.729184	valid loss	0.715469
Epoch 024	train loss	0.660462	valid loss	0.648187
Epoch 025	train loss	0.598241	valid loss	0.587253
Epoch 026	train loss	0.541902	valid loss	0.532067
Epoch 027	train loss	0.490889	valid loss	0.482084

Epoch 028	train loss 0.444695	valid loss 0.436812
Epoch 029	train loss 0.402864	valid loss 0.395806
Epoch 030	train loss 0.364982	valid loss 0.358663
Epoch 031	train loss 0.330676	valid loss 0.325016
Epoch 032	train loss 0.299605	valid loss 0.294537
Epoch 033	train loss 0.271465	valid loss 0.266925
Epoch 034	train loss 0.245977	valid loss 0.241911
Epoch 035	train loss 0.222891	valid loss 0.219248
Epoch 036	train loss 0.201979	valid loss 0.198716
Epoch 037	train loss 0.183037	valid loss 0.180112
Epoch 038	train loss 0.165877	valid loss 0.163257
Epoch 039	train loss 0.150332	valid loss 0.147983
Epoch 040	train loss 0.136249	valid loss 0.134144
Epoch 041	train loss 0.123490	valid loss 0.121603
Epoch 042	train loss 0.111930	valid loss 0.110238
Epoch 043	train loss 0.101456	valid loss 0.099939
Epoch 044	train loss 0.091966	valid loss 0.090605
Epoch 045	train loss 0.083366	valid loss 0.082146
Epoch 046	train loss 0.075574	valid loss 0.074479
Epoch 047	train loss 0.068512	valid loss 0.067530
Epoch 048	train loss 0.062113	valid loss 0.061232
Epoch 049	train loss 0.056314	valid loss 0.055522
Epoch 050	train loss 0.051058	valid loss 0.050347
Epoch 051	train loss 0.046294	valid loss 0.045656
Epoch 052	train loss 0.041976	valid loss 0.041403
Epoch 053	train loss 0.038062	valid loss 0.037548
Epoch 054	train loss 0.034515	valid loss 0.034053
Epoch 055	train loss 0.031299	valid loss 0.030884
Epoch 056	train loss 0.028384	valid loss 0.028011
Epoch 057	train loss 0.025742	valid loss 0.025406
Epoch 058	train loss 0.023346	valid loss 0.023045
Epoch 059	train loss 0.021174	valid loss 0.020903
Epoch 060	train loss 0.019205	valid loss 0.018961
Epoch 061	train loss 0.017419	valid loss 0.017200
Epoch 062	train loss 0.015800	valid loss 0.015603
Epoch 063	train loss 0.014333	valid loss 0.014155
Epoch 064	train loss 0.013001	valid loss 0.012842
Epoch 065	train loss 0.011794	valid loss 0.011651
Epoch 066	train loss 0.010700	valid loss 0.010570
Epoch 067	train loss 0.009707	valid loss 0.009591
Epoch 068	train loss 0.008807	valid loss 0.008702
Epoch 069	train loss 0.007990	valid loss 0.007896
Epoch 070	train loss 0.007250	valid loss 0.007165
Epoch 071	train loss 0.006578	valid loss 0.006501
Epoch 072	train loss 0.005969	valid loss 0.005900
Epoch 073	train loss 0.005416	valid loss 0.005354
Epoch 074	train loss 0.004915	valid loss 0.004859
Epoch 075	train loss 0.004460	valid loss 0.004409

Epoch 076	train loss 0.004048	valid loss 0.004002
Epoch 077	train loss 0.003674	valid loss 0.003632
Epoch 078	train loss 0.003334	valid loss 0.003296
Epoch 079	train loss 0.003026	valid loss 0.002992
Epoch 080	train loss 0.002746	valid loss 0.002716
Epoch 081	train loss 0.002493	valid loss 0.002465
Epoch 082	train loss 0.002263	valid loss 0.002238
Epoch 083	train loss 0.002054	valid loss 0.002031
Epoch 084	train loss 0.001864	valid loss 0.001844
Epoch 085	train loss 0.001693	valid loss 0.001674
Epoch 086	train loss 0.001537	valid loss 0.001520
Epoch 087	train loss 0.001395	valid loss 0.001380
Epoch 088	train loss 0.001266	valid loss 0.001253
Epoch 089	train loss 0.001150	valid loss 0.001137
Epoch 090	train loss 0.001044	valid loss 0.001033
Epoch 091	train loss 0.000948	valid loss 0.000938
Epoch 092	train loss 0.000861	valid loss 0.000851
Epoch 093	train loss 0.000781	valid loss 0.000773
Epoch 094	train loss 0.000710	valid loss 0.000702
Epoch 095	train loss 0.000644	valid loss 0.000637
Epoch 096	train loss 0.000585	valid loss 0.000579
Epoch 097	train loss 0.000531	valid loss 0.000526
Epoch 098	train loss 0.000483	valid loss 0.000477
Epoch 099	train loss 0.000438	valid loss 0.000434

Test loss 0.000440

Estimated A

```
[[ 1.7502961  0.39727157]
 [ 0.9725292  2.2264218 ]
 [ 1.8526078 -0.97195363]
 [ 0.94003755 -0.14778528]]
```

Estimated b

```
[-0.10920743  0.40893131]
```

## 1.1 Dataset

In the precedent example, all the dataset is used at each gradient step. In most of the case, you can't do that as it will not fit into the memory of the computation engine. You have to split your data into little batches. Tensorflow dataset are here to help you in that process.

Let's observe the next script. It presents how to create a Tensorflow dataset object and how to create batches to iterate over it. For each batch, the content of the batch is displayed as well as the computation of a phony operation.

```
[2]: def play_with_tensorflow_dataset_v1():

    # reset tensorflow
    tf.reset_default_graph()

    # create fake data
```

```

nsamples = 105

X = np.arange(nsamples, dtype='float32').reshape(nsamples, 1)
Y = np.ones((nsamples, 1), dtype='float32')

# loop / batch parameters
nepochs = 5
batchsize = 8
nbatches = int(np.ceil(nsamples/batchsize))

# Create a dataset from numpy array
dataset = tf.data.Dataset.from_tensor_slices((X, Y))
# Uncomment following line if you want to automatically shuffle the dataset
#dataset = dataset.shuffle(buffer_size=1000)
# Each iteration over the dataset will provide batchsize samples
dataset = dataset.batch(batchsize)

# Create an iterator over the dataset
dataset_iterator = dataset.make_initializable_iterator()

# Create placeholders for each dataset batch
xbatch, ybatch = dataset_iterator.get_next()

# Create an operation tree with these placeholders
f = tf.reduce_mean((xbatch-ybatch)**2)

init_global = tf.initializers.global_variables()

with tf.Session() as session:
    session.run(init_global)

    for epoch in range(nepochs):

        # Initialize the iterator
        session.run(dataset_iterator.initializer)
        # while True: would also work
        for b in range(nbatches):
            try:
                # get the next batch and run the f operation tree
                fval, xbatchval, ybatchval = session.run(
                    [f, xbatch, ybatch])
            except tf.errors.OutOfRangeError:
                break
            print("Epoch %d/%d Batch %d/%d " % (epoch+1, nepochs,
                b+1, nbatches), fval,
→xbatchval.T, ybatchval.T)

```

```
play_with_tensorflow_dataset_v1()
```

```
Epoch 1/5 Batch 1/14 11.5 [[0. 1. 2. 3. 4. 5. 6. 7.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 2/14 115.5 [[ 8.  9. 10. 11. 12. 13. 14. 15.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 3/14 347.5 [[16. 17. 18. 19. 20. 21. 22. 23.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 4/14 707.5 [[24. 25. 26. 27. 28. 29. 30. 31.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 5/14 1195.5 [[32. 33. 34. 35. 36. 37. 38. 39.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 6/14 1811.5 [[40. 41. 42. 43. 44. 45. 46. 47.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 7/14 2555.5 [[48. 49. 50. 51. 52. 53. 54. 55.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 8/14 3427.5 [[56. 57. 58. 59. 60. 61. 62. 63.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 9/14 4427.5 [[64. 65. 66. 67. 68. 69. 70. 71.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 10/14 5555.5 [[72. 73. 74. 75. 76. 77. 78. 79.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 11/14 6811.5 [[80. 81. 82. 83. 84. 85. 86. 87.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 12/14 8195.5 [[88. 89. 90. 91. 92. 93. 94. 95.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 13/14 9707.5 [[ 96.  97.  98.  99. 100. 101. 102. 103.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 14/14 10609.0 [[104.]] [[1.]]
Epoch 2/5 Batch 1/14 11.5 [[0. 1. 2. 3. 4. 5. 6. 7.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 2/5 Batch 2/14 115.5 [[ 8.  9. 10. 11. 12. 13. 14. 15.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 2/5 Batch 3/14 347.5 [[16. 17. 18. 19. 20. 21. 22. 23.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 2/5 Batch 4/14 707.5 [[24. 25. 26. 27. 28. 29. 30. 31.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 2/5 Batch 5/14 1195.5 [[32. 33. 34. 35. 36. 37. 38. 39.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 2/5 Batch 6/14 1811.5 [[40. 41. 42. 43. 44. 45. 46. 47.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 2/5 Batch 7/14 2555.5 [[48. 49. 50. 51. 52. 53. 54. 55.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
Epoch 2/5 Batch 8/14 3427.5 [[56. 57. 58. 59. 60. 61. 62. 63.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
```



Epoch 2/5 Batch 9/14 4427.5 [[64. 65. 66. 67. 68. 69. 70. 71.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 2/5 Batch 10/14 5555.5 [[72. 73. 74. 75. 76. 77. 78. 79.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 2/5 Batch 11/14 6811.5 [[80. 81. 82. 83. 84. 85. 86. 87.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 2/5 Batch 12/14 8195.5 [[88. 89. 90. 91. 92. 93. 94. 95.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 2/5 Batch 13/14 9707.5 [[ 96. 97. 98. 99. 100. 101. 102. 103.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 2/5 Batch 14/14 10609.0 [[104.]] [[1.]]  
 Epoch 3/5 Batch 1/14 11.5 [[0. 1. 2. 3. 4. 5. 6. 7.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 3/5 Batch 2/14 115.5 [[ 8. 9. 10. 11. 12. 13. 14. 15.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 3/5 Batch 3/14 347.5 [[16. 17. 18. 19. 20. 21. 22. 23.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 3/5 Batch 4/14 707.5 [[24. 25. 26. 27. 28. 29. 30. 31.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 3/5 Batch 5/14 1195.5 [[32. 33. 34. 35. 36. 37. 38. 39.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 3/5 Batch 6/14 1811.5 [[40. 41. 42. 43. 44. 45. 46. 47.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 3/5 Batch 7/14 2555.5 [[48. 49. 50. 51. 52. 53. 54. 55.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 3/5 Batch 8/14 3427.5 [[56. 57. 58. 59. 60. 61. 62. 63.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 3/5 Batch 9/14 4427.5 [[64. 65. 66. 67. 68. 69. 70. 71.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 3/5 Batch 10/14 5555.5 [[72. 73. 74. 75. 76. 77. 78. 79.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 3/5 Batch 11/14 6811.5 [[80. 81. 82. 83. 84. 85. 86. 87.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 3/5 Batch 12/14 8195.5 [[88. 89. 90. 91. 92. 93. 94. 95.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 3/5 Batch 13/14 9707.5 [[ 96. 97. 98. 99. 100. 101. 102. 103.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 3/5 Batch 14/14 10609.0 [[104.]] [[1.]]  
 Epoch 4/5 Batch 1/14 11.5 [[0. 1. 2. 3. 4. 5. 6. 7.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 4/5 Batch 2/14 115.5 [[ 8. 9. 10. 11. 12. 13. 14. 15.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 4/5 Batch 3/14 347.5 [[16. 17. 18. 19. 20. 21. 22. 23.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 4/5 Batch 4/14 707.5 [[24. 25. 26. 27. 28. 29. 30. 31.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 4/5 Batch 5/14 1195.5 [[32. 33. 34. 35. 36. 37. 38. 39.]] [[1. 1. 1. 1. 1. 1. 1. 1.]]

Epoch 4/5 Batch 6/14 1811.5 [[40. 41. 42. 43. 44. 45. 46. 47.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 4/5 Batch 7/14 2555.5 [[48. 49. 50. 51. 52. 53. 54. 55.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 4/5 Batch 8/14 3427.5 [[56. 57. 58. 59. 60. 61. 62. 63.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 4/5 Batch 9/14 4427.5 [[64. 65. 66. 67. 68. 69. 70. 71.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 4/5 Batch 10/14 5555.5 [[72. 73. 74. 75. 76. 77. 78. 79.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 4/5 Batch 11/14 6811.5 [[80. 81. 82. 83. 84. 85. 86. 87.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 4/5 Batch 12/14 8195.5 [[88. 89. 90. 91. 92. 93. 94. 95.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 4/5 Batch 13/14 9707.5 [[ 96. 97. 98. 99. 100. 101. 102. 103.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 4/5 Batch 14/14 10609.0 [[104.]] [[1.]]  
 Epoch 5/5 Batch 1/14 11.5 [[0. 1. 2. 3. 4. 5. 6. 7.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 5/5 Batch 2/14 115.5 [[ 8. 9. 10. 11. 12. 13. 14. 15.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 5/5 Batch 3/14 347.5 [[16. 17. 18. 19. 20. 21. 22. 23.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 5/5 Batch 4/14 707.5 [[24. 25. 26. 27. 28. 29. 30. 31.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 5/5 Batch 5/14 1195.5 [[32. 33. 34. 35. 36. 37. 38. 39.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 5/5 Batch 6/14 1811.5 [[40. 41. 42. 43. 44. 45. 46. 47.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 5/5 Batch 7/14 2555.5 [[48. 49. 50. 51. 52. 53. 54. 55.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 5/5 Batch 8/14 3427.5 [[56. 57. 58. 59. 60. 61. 62. 63.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 5/5 Batch 9/14 4427.5 [[64. 65. 66. 67. 68. 69. 70. 71.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 5/5 Batch 10/14 5555.5 [[72. 73. 74. 75. 76. 77. 78. 79.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 5/5 Batch 11/14 6811.5 [[80. 81. 82. 83. 84. 85. 86. 87.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 5/5 Batch 12/14 8195.5 [[88. 89. 90. 91. 92. 93. 94. 95.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 5/5 Batch 13/14 9707.5 [[ 96. 97. 98. 99. 100. 101. 102. 103.]] [[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]  
 Epoch 5/5 Batch 14/14 10609.0 [[104.]] [[1.]]

## 1.2 Multiple dataset

What should we do with multiple dataset ? For example how can we iterate over train and validation ?

```
[3]: def play_with_tensorflow_dataset_v2():

    # reset tensorflow
    tf.reset_default_graph()

    # create fake data
    nsamples = 105

    Xtrain = np.arange(nsamples, dtype='float32').reshape(nsamples, 1)
    Ytrain = np.ones((nsamples, 1), dtype='float32')

    Xvalid = np.arange(nsamples, dtype='float32').reshape(nsamples, 1)[::-1]
    Yvalid = -np.ones((nsamples, 1), dtype='float32')

    # loop / batch parameters
    nepochs = 5
    batchsize = 8
    nbatches = int(np.ceil(nsamples/batchsize))

    # Create a tensorflow dataset for each dataset
    traindataset = tf.data.Dataset.from_tensor_slices((Xtrain, Ytrain))
    traindataset = traindataset.batch(batchsize)

    validdataset = tf.data.Dataset.from_tensor_slices((Xvalid, Yvalid))
    validdataset = validdataset.batch(batchsize)

    # Create ONLY ONE iterator base on types and shapes of one of the dataset
    # both dataset should have the same types and shapes...
    dataset_iterator = tf.data.Iterator.from_structure(traindataset.
→output_types,
                                                    traindataset.
→output_shapes)

    # Create one initializer per dataset
    training_init_op = dataset_iterator.make_initializer(traindataset)
    validation_init_op = dataset_iterator .make_initializer(validdataset)

    # Create placeholders for each batch
    xbatch, ybatch = dataset_iterator.get_next()

    f = tf.reduce_mean((xbatch-ybatch)**2)

    init_global = tf.initializers.global_variables()
```

```

with tf.Session() as session:
    session.run(init_global)

    for epoch in range(nepochs):
        print("== Train loop ==")
        # Initialize the training dataset iterator
        # Now xbatch,ybatch will iterate over the train dataset
        session.run(training_init_op)
        for b in range(nbatches):
            try:
                fval, xbatchval, ybatchval = session.run(
                    [f, xbatch, ybatch])
            except tf.errors.OutOfRangeError:
                break
            print("Epoch %d/%d Batch %d/%d " % (epoch+1, nepochs,
                b+1, nbatches), fval,
→xbatchval.T, ybatchval.T)
        print("== Validation loop ==")
        # Initialize the validation dataset iterator
        # Now xbatch,ybatch will iterate over the validation dataset
        session.run(validation_init_op)
        for b in range(nbatches):
            try:
                fval, xbatchval, ybatchval = session.run(
                    [f, xbatch, ybatch])
            except tf.errors.OutOfRangeError:
                break
            print("Epoch %d/%d Batch %d/%d " % (epoch+1, nepochs,
                b+1, nbatches), fval,
→xbatchval.T, ybatchval.T)

play_with_tensorflow_dataset_v2()

```

```

== Train loop ==
Epoch 1/5 Batch 1/14  11.5 [[0.  1.  2.  3.  4.  5.  6.  7.]] [[1.  1.  1.  1.  1.  1.  1.  1.  1.  1.]]
Epoch 1/5 Batch 2/14 115.5 [[ 8.  9. 10. 11. 12. 13. 14. 15.]] [[1.  1.  1.  1.  1.  1.  1.  1.  1.  1.]]
Epoch 1/5 Batch 3/14 347.5 [[16. 17. 18. 19. 20. 21. 22. 23.]] [[1.  1.  1.  1.  1.  1.  1.  1.  1.  1.]]
Epoch 1/5 Batch 4/14 707.5 [[24. 25. 26. 27. 28. 29. 30. 31.]] [[1.  1.  1.  1.  1.  1.  1.  1.  1.  1.]]
Epoch 1/5 Batch 5/14 1195.5 [[32. 33. 34. 35. 36. 37. 38. 39.]] [[1.  1.  1.  1.  1.  1.  1.  1.  1.  1.]]
Epoch 1/5 Batch 6/14 1811.5 [[40. 41. 42. 43. 44. 45. 46. 47.]] [[1.  1.  1.  1.

```

```

1. 1. 1. 1.]]
Epoch 1/5 Batch 7/14 2555.5 [[48. 49. 50. 51. 52. 53. 54. 55.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 1/5 Batch 8/14 3427.5 [[56. 57. 58. 59. 60. 61. 62. 63.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 1/5 Batch 9/14 4427.5 [[64. 65. 66. 67. 68. 69. 70. 71.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 1/5 Batch 10/14 5555.5 [[72. 73. 74. 75. 76. 77. 78. 79.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 1/5 Batch 11/14 6811.5 [[80. 81. 82. 83. 84. 85. 86. 87.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 1/5 Batch 12/14 8195.5 [[88. 89. 90. 91. 92. 93. 94. 95.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 1/5 Batch 13/14 9707.5 [[ 96. 97. 98. 99. 100. 101. 102. 103.]] [[1.
1. 1. 1. 1. 1. 1.]]
Epoch 1/5 Batch 14/14 10609.0 [[104.]] [[1.]]
== Validation loop ==
Epoch 1/5 Batch 1/14 10307.5 [[104. 103. 102. 101. 100. 99. 98. 97.]] [[-1.
-1. -1. -1. -1. -1. -1.]]
Epoch 1/5 Batch 2/14 8747.5 [[96. 95. 94. 93. 92. 91. 90. 89.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 1/5 Batch 3/14 7315.5 [[88. 87. 86. 85. 84. 83. 82. 81.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 1/5 Batch 4/14 6011.5 [[80. 79. 78. 77. 76. 75. 74. 73.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 1/5 Batch 5/14 4835.5 [[72. 71. 70. 69. 68. 67. 66. 65.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 1/5 Batch 6/14 3787.5 [[64. 63. 62. 61. 60. 59. 58. 57.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 1/5 Batch 7/14 2867.5 [[56. 55. 54. 53. 52. 51. 50. 49.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 1/5 Batch 8/14 2075.5 [[48. 47. 46. 45. 44. 43. 42. 41.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 1/5 Batch 9/14 1411.5 [[40. 39. 38. 37. 36. 35. 34. 33.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 1/5 Batch 10/14 875.5 [[32. 31. 30. 29. 28. 27. 26. 25.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 1/5 Batch 11/14 467.5 [[24. 23. 22. 21. 20. 19. 18. 17.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 1/5 Batch 12/14 187.5 [[16. 15. 14. 13. 12. 11. 10. 9.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 1/5 Batch 13/14 35.5 [[8. 7. 6. 5. 4. 3. 2. 1.]] [[-1. -1. -1. -1. -1.
-1. -1. -1.]]
Epoch 1/5 Batch 14/14 1.0 [[0.]] [[-1.]]
== Train loop ==
Epoch 2/5 Batch 1/14 11.5 [[0. 1. 2. 3. 4. 5. 6. 7.]] [[1. 1. 1. 1. 1. 1. 1.
1.]]
Epoch 2/5 Batch 2/14 115.5 [[ 8. 9. 10. 11. 12. 13. 14. 15.]] [[1. 1. 1. 1. 1.

```

```

1. 1. 1.]]
Epoch 2/5 Batch 3/14 347.5 [[16. 17. 18. 19. 20. 21. 22. 23.]] [[1. 1. 1. 1. 1.
1. 1. 1.]]
Epoch 2/5 Batch 4/14 707.5 [[24. 25. 26. 27. 28. 29. 30. 31.]] [[1. 1. 1. 1. 1.
1. 1. 1.]]
Epoch 2/5 Batch 5/14 1195.5 [[32. 33. 34. 35. 36. 37. 38. 39.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 2/5 Batch 6/14 1811.5 [[40. 41. 42. 43. 44. 45. 46. 47.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 2/5 Batch 7/14 2555.5 [[48. 49. 50. 51. 52. 53. 54. 55.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 2/5 Batch 8/14 3427.5 [[56. 57. 58. 59. 60. 61. 62. 63.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 2/5 Batch 9/14 4427.5 [[64. 65. 66. 67. 68. 69. 70. 71.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 2/5 Batch 10/14 5555.5 [[72. 73. 74. 75. 76. 77. 78. 79.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 2/5 Batch 11/14 6811.5 [[80. 81. 82. 83. 84. 85. 86. 87.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 2/5 Batch 12/14 8195.5 [[88. 89. 90. 91. 92. 93. 94. 95.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 2/5 Batch 13/14 9707.5 [[ 96. 97. 98. 99. 100. 101. 102. 103.]] [[1.
1. 1. 1. 1. 1. 1.]]
Epoch 2/5 Batch 14/14 10609.0 [[104.]] [[1.]]
== Validation loop ==
Epoch 2/5 Batch 1/14 10307.5 [[104. 103. 102. 101. 100. 99. 98. 97.]] [[-1.
-1. -1. -1. -1. -1. -1.]]
Epoch 2/5 Batch 2/14 8747.5 [[96. 95. 94. 93. 92. 91. 90. 89.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 2/5 Batch 3/14 7315.5 [[88. 87. 86. 85. 84. 83. 82. 81.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 2/5 Batch 4/14 6011.5 [[80. 79. 78. 77. 76. 75. 74. 73.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 2/5 Batch 5/14 4835.5 [[72. 71. 70. 69. 68. 67. 66. 65.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 2/5 Batch 6/14 3787.5 [[64. 63. 62. 61. 60. 59. 58. 57.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 2/5 Batch 7/14 2867.5 [[56. 55. 54. 53. 52. 51. 50. 49.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 2/5 Batch 8/14 2075.5 [[48. 47. 46. 45. 44. 43. 42. 41.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 2/5 Batch 9/14 1411.5 [[40. 39. 38. 37. 36. 35. 34. 33.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 2/5 Batch 10/14 875.5 [[32. 31. 30. 29. 28. 27. 26. 25.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 2/5 Batch 11/14 467.5 [[24. 23. 22. 21. 20. 19. 18. 17.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 2/5 Batch 12/14 187.5 [[16. 15. 14. 13. 12. 11. 10. 9.]] [[-1. -1. -1.

```

```

-1. -1. -1. -1. -1.]]
Epoch 2/5 Batch 13/14 35.5 [[8. 7. 6. 5. 4. 3. 2. 1.]] [[-1. -1. -1. -1. -1.
-1. -1. -1.]]
Epoch 2/5 Batch 14/14 1.0 [[0.]] [[-1.]]
== Train loop ==
Epoch 3/5 Batch 1/14 11.5 [[0. 1. 2. 3. 4. 5. 6. 7.]] [[1. 1. 1. 1. 1. 1. 1.
1.]]
Epoch 3/5 Batch 2/14 115.5 [[ 8. 9. 10. 11. 12. 13. 14. 15.]] [[1. 1. 1. 1. 1.
1. 1. 1.]]
Epoch 3/5 Batch 3/14 347.5 [[16. 17. 18. 19. 20. 21. 22. 23.]] [[1. 1. 1. 1. 1.
1. 1. 1.]]
Epoch 3/5 Batch 4/14 707.5 [[24. 25. 26. 27. 28. 29. 30. 31.]] [[1. 1. 1. 1. 1.
1. 1. 1.]]
Epoch 3/5 Batch 5/14 1195.5 [[32. 33. 34. 35. 36. 37. 38. 39.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 3/5 Batch 6/14 1811.5 [[40. 41. 42. 43. 44. 45. 46. 47.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 3/5 Batch 7/14 2555.5 [[48. 49. 50. 51. 52. 53. 54. 55.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 3/5 Batch 8/14 3427.5 [[56. 57. 58. 59. 60. 61. 62. 63.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 3/5 Batch 9/14 4427.5 [[64. 65. 66. 67. 68. 69. 70. 71.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 3/5 Batch 10/14 5555.5 [[72. 73. 74. 75. 76. 77. 78. 79.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 3/5 Batch 11/14 6811.5 [[80. 81. 82. 83. 84. 85. 86. 87.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 3/5 Batch 12/14 8195.5 [[88. 89. 90. 91. 92. 93. 94. 95.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 3/5 Batch 13/14 9707.5 [[ 96. 97. 98. 99. 100. 101. 102. 103.]] [[1.
1. 1. 1. 1. 1. 1.]]
Epoch 3/5 Batch 14/14 10609.0 [[104.]] [[1.]]
== Validation loop ==
Epoch 3/5 Batch 1/14 10307.5 [[104. 103. 102. 101. 100. 99. 98. 97.]] [[-1.
-1. -1. -1. -1. -1. -1.]]
Epoch 3/5 Batch 2/14 8747.5 [[96. 95. 94. 93. 92. 91. 90. 89.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 3/5 Batch 3/14 7315.5 [[88. 87. 86. 85. 84. 83. 82. 81.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 3/5 Batch 4/14 6011.5 [[80. 79. 78. 77. 76. 75. 74. 73.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 3/5 Batch 5/14 4835.5 [[72. 71. 70. 69. 68. 67. 66. 65.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 3/5 Batch 6/14 3787.5 [[64. 63. 62. 61. 60. 59. 58. 57.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 3/5 Batch 7/14 2867.5 [[56. 55. 54. 53. 52. 51. 50. 49.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 3/5 Batch 8/14 2075.5 [[48. 47. 46. 45. 44. 43. 42. 41.]] [[-1. -1. -1.

```

```

-1. -1. -1. -1. -1.]]
Epoch 3/5 Batch 9/14 1411.5 [[40. 39. 38. 37. 36. 35. 34. 33.]] [[-1. -1. -1.
-1. -1. -1. -1. -1.]]
Epoch 3/5 Batch 10/14 875.5 [[32. 31. 30. 29. 28. 27. 26. 25.]] [[-1. -1. -1.
-1. -1. -1. -1. -1.]]
Epoch 3/5 Batch 11/14 467.5 [[24. 23. 22. 21. 20. 19. 18. 17.]] [[-1. -1. -1.
-1. -1. -1. -1. -1.]]
Epoch 3/5 Batch 12/14 187.5 [[16. 15. 14. 13. 12. 11. 10. 9.]] [[-1. -1. -1.
-1. -1. -1. -1. -1.]]
Epoch 3/5 Batch 13/14 35.5 [[8. 7. 6. 5. 4. 3. 2. 1.]] [[-1. -1. -1. -1. -1.
-1. -1. -1.]]
Epoch 3/5 Batch 14/14 1.0 [[0.]] [[-1.]]
== Train loop ==
Epoch 4/5 Batch 1/14 11.5 [[0. 1. 2. 3. 4. 5. 6. 7.]] [[1. 1. 1. 1. 1. 1. 1.
1.]]
Epoch 4/5 Batch 2/14 115.5 [[ 8. 9. 10. 11. 12. 13. 14. 15.]] [[1. 1. 1. 1. 1.
1. 1. 1.]]
Epoch 4/5 Batch 3/14 347.5 [[16. 17. 18. 19. 20. 21. 22. 23.]] [[1. 1. 1. 1. 1.
1. 1. 1.]]
Epoch 4/5 Batch 4/14 707.5 [[24. 25. 26. 27. 28. 29. 30. 31.]] [[1. 1. 1. 1. 1.
1. 1. 1.]]
Epoch 4/5 Batch 5/14 1195.5 [[32. 33. 34. 35. 36. 37. 38. 39.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 4/5 Batch 6/14 1811.5 [[40. 41. 42. 43. 44. 45. 46. 47.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 4/5 Batch 7/14 2555.5 [[48. 49. 50. 51. 52. 53. 54. 55.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 4/5 Batch 8/14 3427.5 [[56. 57. 58. 59. 60. 61. 62. 63.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 4/5 Batch 9/14 4427.5 [[64. 65. 66. 67. 68. 69. 70. 71.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 4/5 Batch 10/14 5555.5 [[72. 73. 74. 75. 76. 77. 78. 79.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 4/5 Batch 11/14 6811.5 [[80. 81. 82. 83. 84. 85. 86. 87.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 4/5 Batch 12/14 8195.5 [[88. 89. 90. 91. 92. 93. 94. 95.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 4/5 Batch 13/14 9707.5 [[ 96. 97. 98. 99. 100. 101. 102. 103.]] [[1.
1. 1. 1. 1. 1. 1.]]
Epoch 4/5 Batch 14/14 10609.0 [[104.]] [[1.]]
== Validation loop ==
Epoch 4/5 Batch 1/14 10307.5 [[104. 103. 102. 101. 100. 99. 98. 97.]] [[-1.
-1. -1. -1. -1. -1. -1.]]
Epoch 4/5 Batch 2/14 8747.5 [[96. 95. 94. 93. 92. 91. 90. 89.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 4/5 Batch 3/14 7315.5 [[88. 87. 86. 85. 84. 83. 82. 81.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 4/5 Batch 4/14 6011.5 [[80. 79. 78. 77. 76. 75. 74. 73.]] [[-1. -1. -1.

```



```

-1. -1. -1. -1. -1.]]
Epoch 4/5 Batch 5/14 4835.5 [[72. 71. 70. 69. 68. 67. 66. 65.]] [[-1. -1. -1.
-1. -1. -1. -1. -1.]]
Epoch 4/5 Batch 6/14 3787.5 [[64. 63. 62. 61. 60. 59. 58. 57.]] [[-1. -1. -1.
-1. -1. -1. -1. -1.]]
Epoch 4/5 Batch 7/14 2867.5 [[56. 55. 54. 53. 52. 51. 50. 49.]] [[-1. -1. -1.
-1. -1. -1. -1. -1.]]
Epoch 4/5 Batch 8/14 2075.5 [[48. 47. 46. 45. 44. 43. 42. 41.]] [[-1. -1. -1.
-1. -1. -1. -1. -1.]]
Epoch 4/5 Batch 9/14 1411.5 [[40. 39. 38. 37. 36. 35. 34. 33.]] [[-1. -1. -1.
-1. -1. -1. -1. -1.]]
Epoch 4/5 Batch 10/14 875.5 [[32. 31. 30. 29. 28. 27. 26. 25.]] [[-1. -1. -1.
-1. -1. -1. -1. -1.]]
Epoch 4/5 Batch 11/14 467.5 [[24. 23. 22. 21. 20. 19. 18. 17.]] [[-1. -1. -1.
-1. -1. -1. -1. -1.]]
Epoch 4/5 Batch 12/14 187.5 [[16. 15. 14. 13. 12. 11. 10. 9.]] [[-1. -1. -1.
-1. -1. -1. -1. -1.]]
Epoch 4/5 Batch 13/14 35.5 [[8. 7. 6. 5. 4. 3. 2. 1.]] [[-1. -1. -1. -1. -1.
-1. -1. -1.]]
Epoch 4/5 Batch 14/14 1.0 [[0.]] [[-1.]]
== Train loop ==
Epoch 5/5 Batch 1/14 11.5 [[0. 1. 2. 3. 4. 5. 6. 7.]] [[1. 1. 1. 1. 1. 1. 1.
1.]]
Epoch 5/5 Batch 2/14 115.5 [[ 8. 9. 10. 11. 12. 13. 14. 15.]] [[1. 1. 1. 1. 1.
1. 1. 1.]]
Epoch 5/5 Batch 3/14 347.5 [[16. 17. 18. 19. 20. 21. 22. 23.]] [[1. 1. 1. 1. 1.
1. 1. 1.]]
Epoch 5/5 Batch 4/14 707.5 [[24. 25. 26. 27. 28. 29. 30. 31.]] [[1. 1. 1. 1. 1.
1. 1. 1.]]
Epoch 5/5 Batch 5/14 1195.5 [[32. 33. 34. 35. 36. 37. 38. 39.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 5/5 Batch 6/14 1811.5 [[40. 41. 42. 43. 44. 45. 46. 47.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 5/5 Batch 7/14 2555.5 [[48. 49. 50. 51. 52. 53. 54. 55.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 5/5 Batch 8/14 3427.5 [[56. 57. 58. 59. 60. 61. 62. 63.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 5/5 Batch 9/14 4427.5 [[64. 65. 66. 67. 68. 69. 70. 71.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 5/5 Batch 10/14 5555.5 [[72. 73. 74. 75. 76. 77. 78. 79.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 5/5 Batch 11/14 6811.5 [[80. 81. 82. 83. 84. 85. 86. 87.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 5/5 Batch 12/14 8195.5 [[88. 89. 90. 91. 92. 93. 94. 95.]] [[1. 1. 1. 1.
1. 1. 1. 1.]]
Epoch 5/5 Batch 13/14 9707.5 [[ 96. 97. 98. 99. 100. 101. 102. 103.]] [[1.
1. 1. 1. 1. 1. 1.]]
Epoch 5/5 Batch 14/14 10609.0 [[104.]] [[1.]]

```

```

== Validation loop ==
Epoch 5/5 Batch 1/14 10307.5 [[104. 103. 102. 101. 100. 99. 98. 97.]] [[-1.
-1. -1. -1. -1. -1. -1.]]
Epoch 5/5 Batch 2/14 8747.5 [[96. 95. 94. 93. 92. 91. 90. 89.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 5/5 Batch 3/14 7315.5 [[88. 87. 86. 85. 84. 83. 82. 81.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 5/5 Batch 4/14 6011.5 [[80. 79. 78. 77. 76. 75. 74. 73.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 5/5 Batch 5/14 4835.5 [[72. 71. 70. 69. 68. 67. 66. 65.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 5/5 Batch 6/14 3787.5 [[64. 63. 62. 61. 60. 59. 58. 57.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 5/5 Batch 7/14 2867.5 [[56. 55. 54. 53. 52. 51. 50. 49.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 5/5 Batch 8/14 2075.5 [[48. 47. 46. 45. 44. 43. 42. 41.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 5/5 Batch 9/14 1411.5 [[40. 39. 38. 37. 36. 35. 34. 33.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 5/5 Batch 10/14 875.5 [[32. 31. 30. 29. 28. 27. 26. 25.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 5/5 Batch 11/14 467.5 [[24. 23. 22. 21. 20. 19. 18. 17.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 5/5 Batch 12/14 187.5 [[16. 15. 14. 13. 12. 11. 10. 9.]] [[-1. -1. -1.
-1. -1. -1. -1.]]
Epoch 5/5 Batch 13/14 35.5 [[8. 7. 6. 5. 4. 3. 2. 1.]] [[-1. -1. -1. -1. -1.
-1. -1. -1.]]
Epoch 5/5 Batch 14/14 1.0 [[0.]] [[-1.]]

```

### 1.3 Your turn !

Modify the initial linear regression function to introduce mini batches.

The typical code for computing the validation loss will be something like

```

def compute_loss(session,iterator_init,loss,nbatches):
    session.run(iterator_init)
    lossval = 0.
    for b in range(nbatches):
        batch_lossval, = session.run([loss])
        lossval += batch_lossval
    lossval /= nbatches
    return lossval

```