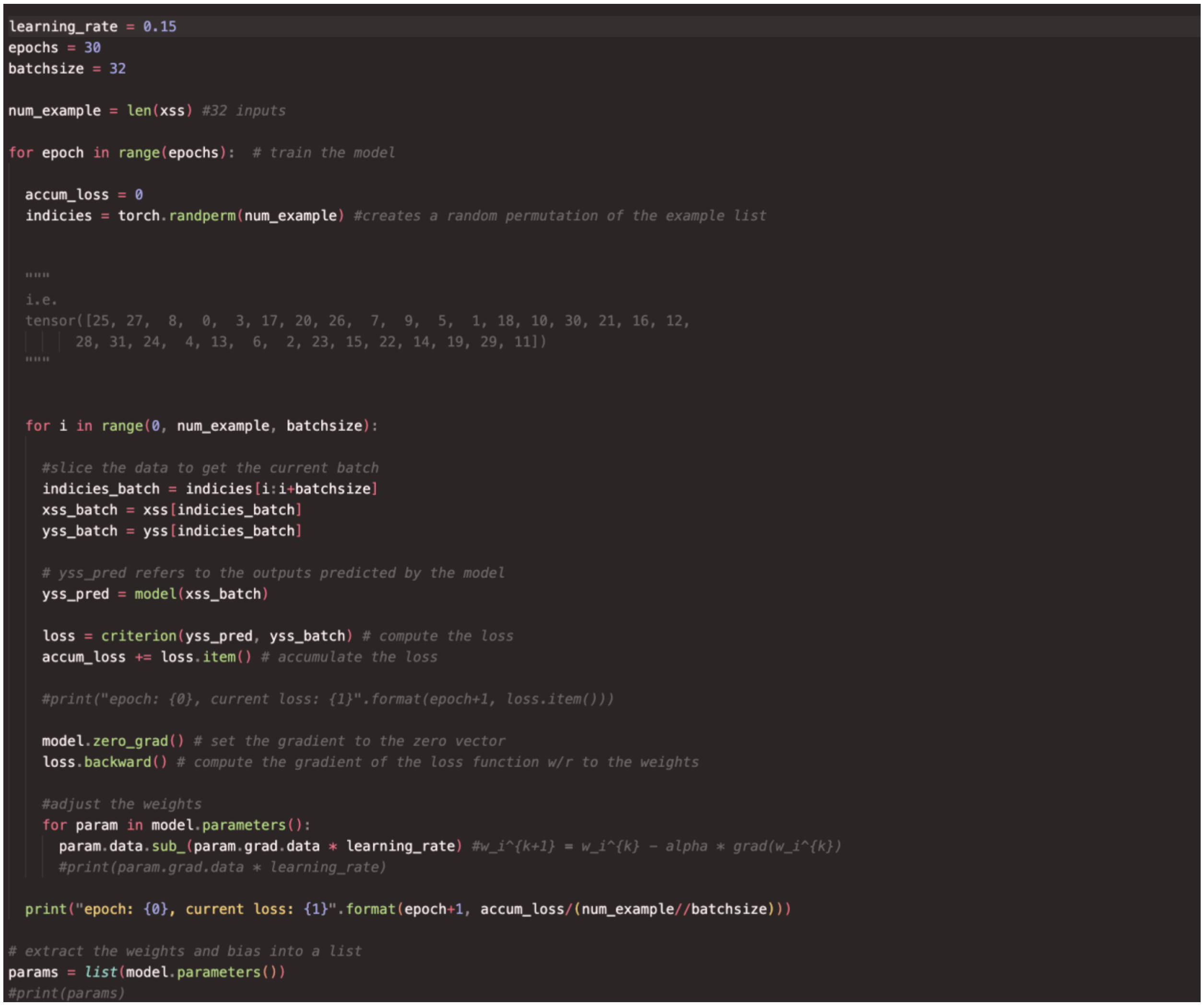
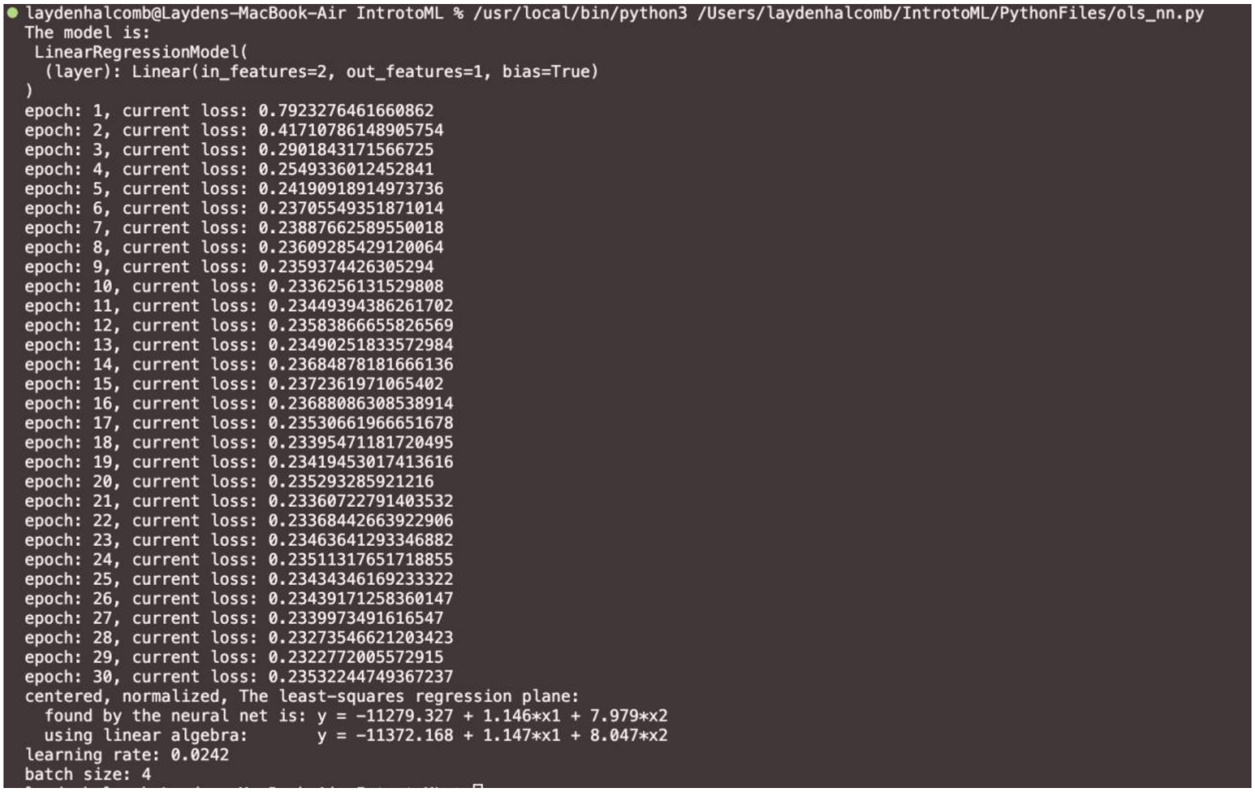
Part 1

Code Loop:



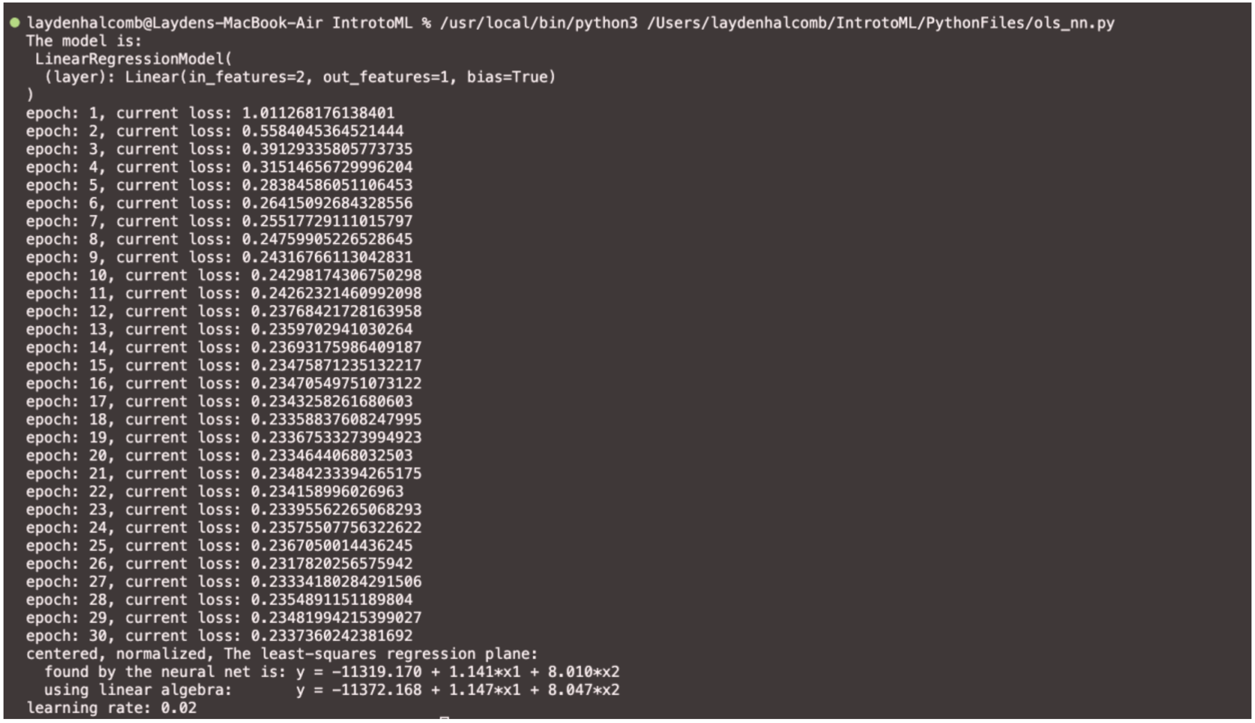
```
learning_rate = 0.15
epochs = 30
batchsize = 32

num_example = len(xss) #32 inputs

for epoch in range(epochs):  # train the model

  accum_loss = 0
  indicies = torch.randperm(num_example) #creates a random permutation of the example list


  """
  i.e.
  tensor([25, 27,  8,  0,  3, 17, 20, 26,  7,  9,  5,  1, 18, 10, 30, 21, 16, 12,
        28, 31, 24,  4, 13,  6,  2, 23, 15, 22, 14, 19, 29, 11])
  """


  for i in range(0, num_example, batchsize):

    #slice the data to get the current batch
    indicies_batch = indicies[i:i+batchsize]
    xss_batch = xss[indicies_batch]
    yss_batch = yss[indicies_batch]

    # yss_pred refers to the outputs predicted by the model
    yss_pred = model(xss_batch)

    loss = criterion(yss_pred, yss_batch) # compute the loss
    accum_loss += loss.item() # accumulate the loss

    #print("epoch: {0}, current loss: {1}".format(epoch+1, loss.item()))

    model.zero_grad() # set the gradient to the zero vector
    loss.backward() # compute the gradient of the loss function w/r to the weights

    #adjust the weights
    for param in model.parameters():
      param.data.sub_(param.grad.data * learning_rate) #w_i^{k+1} = w_i^{k} - alpha * grad(w_i^{k})
      #print(param.grad.data * learning_rate)

  print("epoch: {0}, current loss: {1}".format(epoch+1, accum_loss/(num_example//batchsize)))

# extract the weights and bias into a list
params = list(model.parameters())
#print(params)
```

Output With learning rate = 0.0242 and batchsize = 4:



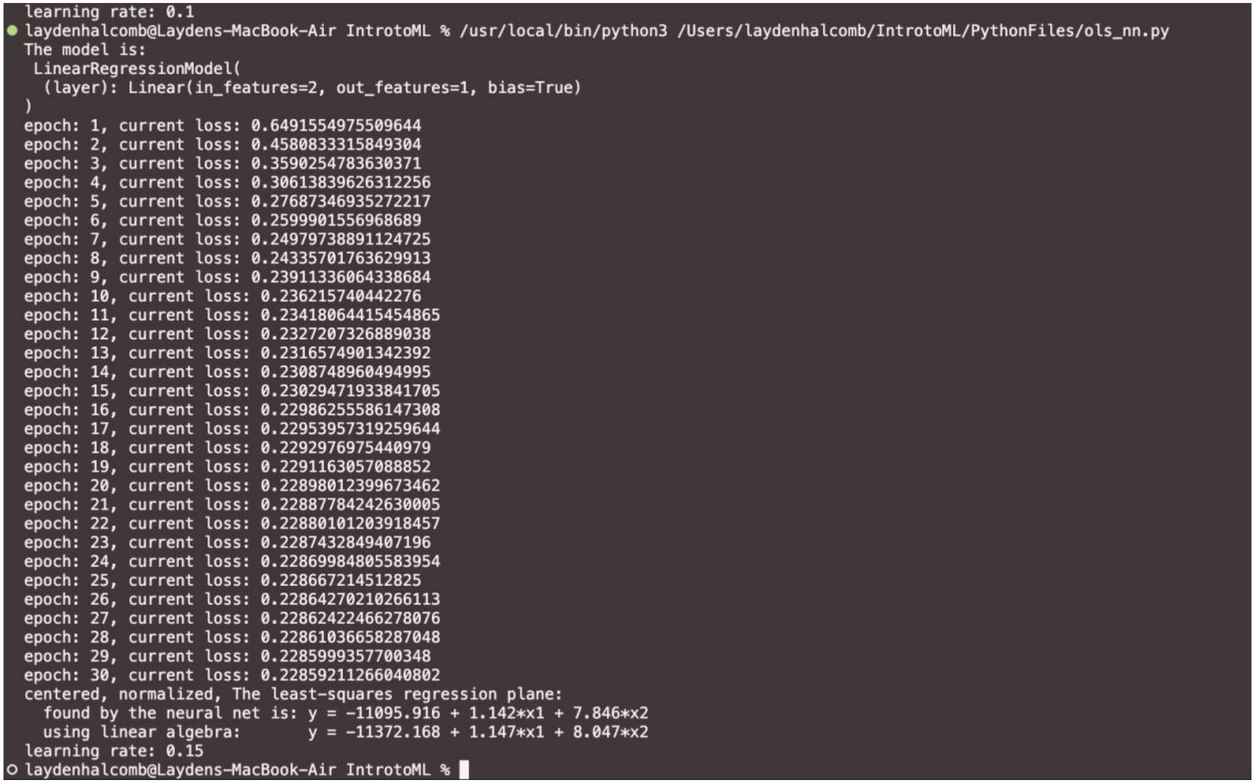
```
● laydenhalcomb@Laydens-MacBook-Air IntrotoML % /usr/local/bin/python3 /Users/laydenhalcomb/IntrotoML/PythonFiles/ols_nn.py
  The model is:
   LinearRegressionModel(
    (layer): Linear(in_features=2, out_features=1, bias=True)
  )
  epoch: 1, current loss: 0.7923276461660862
  epoch: 2, current loss: 0.41710786148905754
  epoch: 3, current loss: 0.2901843171566725
  epoch: 4, current loss: 0.2549336012452841
  epoch: 5, current loss: 0.24190918914973736
  epoch: 6, current loss: 0.23705549351871014
  epoch: 7, current loss: 0.23887662589550018
  epoch: 8, current loss: 0.23609285429120064
  epoch: 9, current loss: 0.2359374426305294
  epoch: 10, current loss: 0.2336256131529808
  epoch: 11, current loss: 0.23449394386261702
  epoch: 12, current loss: 0.23583866655826569
  epoch: 13, current loss: 0.23490251833572984
  epoch: 14, current loss: 0.23684878181666136
  epoch: 15, current loss: 0.2372361971065402
  epoch: 16, current loss: 0.23688086308538914
  epoch: 17, current loss: 0.23530661966651678
  epoch: 18, current loss: 0.23395471181720495
  epoch: 19, current loss: 0.23419453017413616
  epoch: 20, current loss: 0.235293285921216
  epoch: 21, current loss: 0.23360722791403532
  epoch: 22, current loss: 0.23368442663922906
  epoch: 23, current loss: 0.23463641293346882
  epoch: 24, current loss: 0.23511317651718855
  epoch: 25, current loss: 0.23434346169233322
  epoch: 26, current loss: 0.23439171258360147
  epoch: 27, current loss: 0.2339973491616547
  epoch: 28, current loss: 0.23273546621203423
  epoch: 29, current loss: 0.2322772005572915
  epoch: 30, current loss: 0.23532244749367237
  centered, normalized, The least-squares regression plane:
    found by the neural net is: y = -11279.327 + 1.146*x1 + 7.979*x2
    using linear algebra:       y = -11372.168 + 1.147*x1 + 8.047*x2
  learning rate: 0.0242
  batch size: 4
```

With Stochastic gradient descent. Otherwords, batchsize = 1.



```
● laydenhalcomb@Laydens-MacBook-Air IntrotoML % /usr/local/bin/python3 /Users/laydenhalcomb/IntrotoML/PythonFiles/ols_nn.py
  The model is:
   LinearRegressionModel(
    (layer): Linear(in_features=2, out_features=1, bias=True)
  )
  epoch: 1, current loss: 1.011268176138401
  epoch: 2, current loss: 0.5584045364521444
  epoch: 3, current loss: 0.39129335805773735
  epoch: 4, current loss: 0.31514656729996204
  epoch: 5, current loss: 0.28384586051106453
  epoch: 6, current loss: 0.26415092684328556
  epoch: 7, current loss: 0.25517729111015797
  epoch: 8, current loss: 0.24759905226528645
  epoch: 9, current loss: 0.24316766113042831
  epoch: 10, current loss: 0.24298174306750298
  epoch: 11, current loss: 0.24262321460992098
  epoch: 12, current loss: 0.23768421728163958
  epoch: 13, current loss: 0.2359702941030264
  epoch: 14, current loss: 0.23693175986409187
  epoch: 15, current loss: 0.23475871235132217
  epoch: 16, current loss: 0.23470549751073122
  epoch: 17, current loss: 0.2343258261680603
  epoch: 18, current loss: 0.23358837608247995
  epoch: 19, current loss: 0.23367533273994923
  epoch: 20, current loss: 0.2334644068032503
  epoch: 21, current loss: 0.23484233394265175
  epoch: 22, current loss: 0.234158996026963
  epoch: 23, current loss: 0.23395562265068293
  epoch: 24, current loss: 0.23575507756322622
  epoch: 25, current loss: 0.2367050014436245
  epoch: 26, current loss: 0.2317820256575942
  epoch: 27, current loss: 0.23334180284291506
  epoch: 28, current loss: 0.2354891151189804
  epoch: 29, current loss: 0.23481994215399027
  epoch: 30, current loss: 0.2337360242381692
  centered, normalized, The least-squares regression plane:
    found by the neural net is: y = -11319.170 + 1.141*x1 + 8.010*x2
    using linear algebra:       y = -11372.168 + 1.147*x1 + 8.047*x2
  learning rate: 0.02
```

With batchsize = 32. All inputs. And learning rate = 0.15



```
  learning rate: 0.1
● laydenhalcomb@Laydens-MacBook-Air IntrotoML % /usr/local/bin/python3 /Users/laydenhalcomb/IntrotoML/PythonFiles/ols_nn.py
  The model is:
   LinearRegressionModel(
    (layer): Linear(in_features=2, out_features=1, bias=True)
  )
  epoch: 1, current loss: 0.6491554975509644
  epoch: 2, current loss: 0.4580833315849304
  epoch: 3, current loss: 0.3590254783630371
  epoch: 4, current loss: 0.30613839626312256
  epoch: 5, current loss: 0.27687346935272217
  epoch: 6, current loss: 0.2599901556968689
  epoch: 7, current loss: 0.24979738891124725
  epoch: 8, current loss: 0.24335701763629913
  epoch: 9, current loss: 0.23911336064338684
  epoch: 10, current loss: 0.236215740442276
  epoch: 11, current loss: 0.23418064415454865
  epoch: 12, current loss: 0.2327207326889038
  epoch: 13, current loss: 0.2316574901342392
  epoch: 14, current loss: 0.2308748960494995
  epoch: 15, current loss: 0.23029471933841705
  epoch: 16, current loss: 0.22986255586147308
  epoch: 17, current loss: 0.22953957319259644
  epoch: 18, current loss: 0.2292976975440979
  epoch: 19, current loss: 0.2291163057088852
  epoch: 20, current loss: 0.22898012399673462
  epoch: 21, current loss: 0.22887784242630005
  epoch: 22, current loss: 0.22880101203918457
  epoch: 23, current loss: 0.2287432849407196
  epoch: 24, current loss: 0.22869984805583954
  epoch: 25, current loss: 0.228667214512825
  epoch: 26, current loss: 0.22864270210266113
  epoch: 27, current loss: 0.22862422466278076
  epoch: 28, current loss: 0.22861036658287048
  epoch: 29, current loss: 0.2285999357700348
  epoch: 30, current loss: 0.22859211266040802
  centered, normalized, The least-squares regression plane:
    found by the neural net is: y = -11095.916 + 1.142*x1 + 7.846*x2
    using linear algebra:       y = -11372.168 + 1.147*x1 + 8.047*x2
  learning rate: 0.15
○ laydenhalcomb@Laydens-MacBook-Air IntrotoML % █
```

Part 2