

4.3.multi-target_linear_regression

March 28, 2022

Objective

How to make a prediction using multiple samples.

1 Table of Contents

In this lab, we will review how to make a prediction for Linear Regression with Multiple Output.

Build Custom Modules

Estimated Time Needed: 15 min

Class Linear

```
[1]: from torch import nn
import torch
```

Set the random seed:

```
[2]: torch.manual_seed(1)
```

```
[2]: <torch._C.Generator at 0x7fbee138370>
```

Set the random seed:

```
[3]: class linear_regression(nn.Module):
    def __init__(self,input_size,output_size):
        super(linear_regression,self).__init__()
        self.linear=nn.Linear(input_size,output_size)
    def forward(self,x):
        yhat=self.linear(x)
        return yhat
```

create a linear regression object, as our input and output will be two we set the parameters accordingly

```
[4]: model=linear_regression(1,10)
model(torch.tensor([1.0]))
```

```
[4]: tensor([ 0.7926, -0.3920,  0.1714,  0.0797, -1.0143,  0.5097, -0.0608,  0.5047,
           1.0132,  0.1887], grad_fn=<AddBackward0>)
```

we can use the diagram to represent the model or object

we can see the parameters

```
[5]: list(model.parameters())
```

```
[5]: [Parameter containing:
      tensor([[ 0.5153],
              [-0.4414],
              [-0.1939],
              [ 0.4694],
              [-0.9414],
              [ 0.5997],
              [-0.2057],
              [ 0.5087],
              [ 0.1390],
              [-0.1224]], requires_grad=True),
      Parameter containing:
      tensor([ 0.2774,  0.0493,  0.3652, -0.3897, -0.0729, -0.0900,  0.1449, -0.0040,
              0.8742,  0.3112], requires_grad=True)]
```

we can create a tensor with two rows representing one sample of data

```
[6]: x=torch.tensor([[1.0]])
```

we can make a prediction

```
[7]: yhat=model(x)
      yhat
```

```
[7]: tensor([[ 0.7926, -0.3920,  0.1714,  0.0797, -1.0143,  0.5097, -0.0608,  0.5047,
              1.0132,  0.1887]], grad_fn=<AddmmBackward>)
```

each row in the following tensor represents a different sample

```
[8]: X=torch.tensor([[1.0],[1.0],[3.0]])
```

we can make a prediction using multiple samples

```
[9]: Yhat=model(X)
      Yhat
```

```
[9]: tensor([[ 0.7926, -0.3920,  0.1714,  0.0797, -1.0143,  0.5097, -0.0608,  0.5047,
              1.0132,  0.1887],
            [ 0.7926, -0.3920,  0.1714,  0.0797, -1.0143,  0.5097, -0.0608,  0.5047,
              1.0132,  0.1887],
            [ 1.8232, -1.2748, -0.2164,  1.0184, -2.8972,  1.7091, -0.4722,  1.5222,
              1.2912, -0.0561]], grad_fn=<AddmmBackward>)
```

the following figure represents the operation, where the red and blue represents the different parameters, and the different shades of green represent different samples.

2 About the Authors:

[Joseph Santarcangelo](#) has a PhD in Electrical Engineering, his research focused on using machine learning, signal processing, and computer vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Other contributors: [Michelle Carey](#)

2.1 Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-09-23	2.0	Shubham	Migrated Lab to Markdown and added to course repo in GitLab

##

© IBM Corporation 2020. All rights reserved.