



IBM Developer SKILLS NETWORK

Objective

- How to make a prediction using multiple samples.

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

In [1]:

```
from torch import nn
import torch
```

Set the random seed:

In [2]:

```
torch.manual_seed(1)
```

Out[2]:

```
<torch._C.Generator at 0x214c9fd32d0>
```

Set the random seed:

In [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

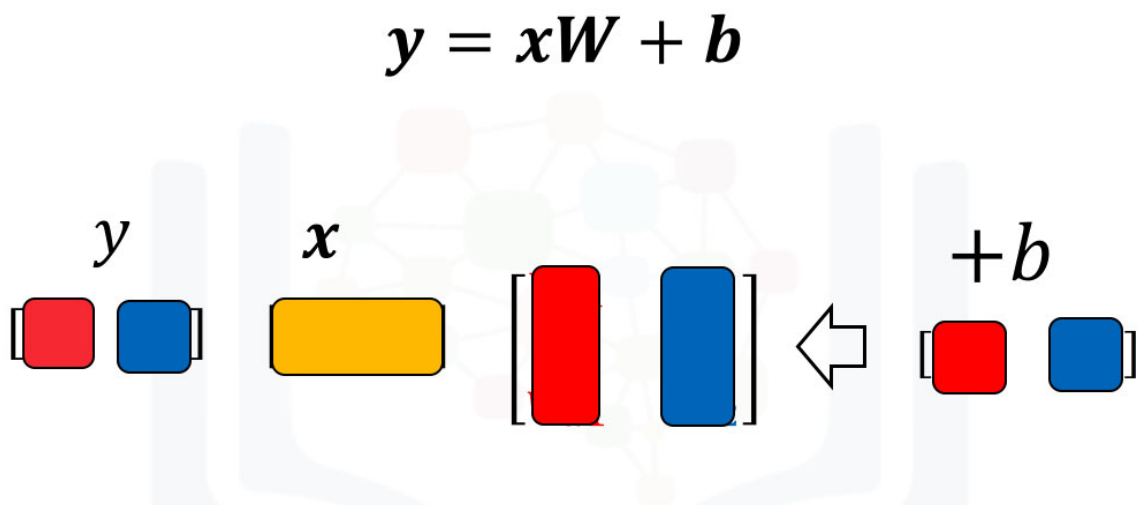
In [4]:

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

Out[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

In [5]:

```
list(model.parameters())
```

Out[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

In [6]:

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

we can make a prediction

In [7]:

```
yhat=model(x)
yhat
```

Out[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

In [8]:

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

we can make a prediction using multiple samples

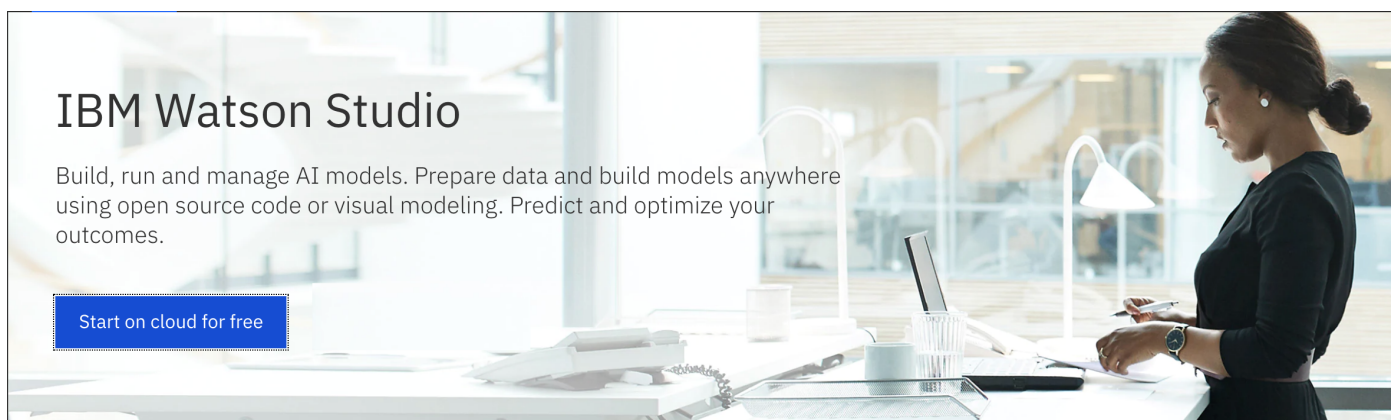
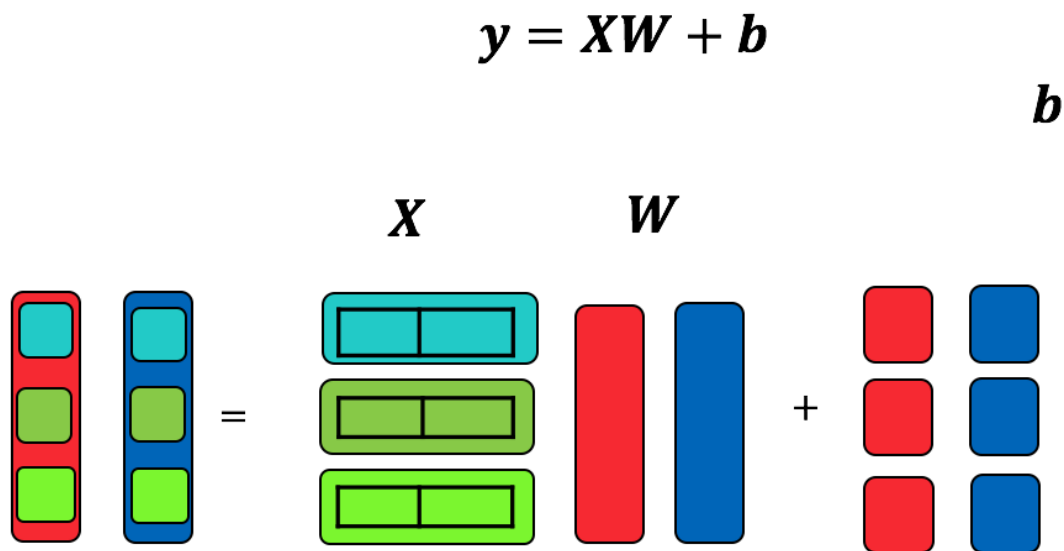
In [9]:

```
Yhat=model(X)
Yhat
```

Out[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.



(https://dataplatform.cloud.ibm.com/registration/stepone?context=cpdaas&apps=data_science_experience,watson_machine_learning)

About the Authors:

Other contributors: [Michelle Carey \(https://www.linkedin.com/in/michelleccarey?cm_mmc=Email_Newsletter-_Developer_Ed%2BTech-_WW_WW-_SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DL0110EN-SkillsNetwork-20647811&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvsrsrc=email.Newsletter.1\)](https://www.linkedin.com/in/michelleccarey?cm_mmc=Email_Newsletter-_Developer_Ed%2BTech-_WW_WW-_SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DL0110EN-SkillsNetwork-20647811&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvsrsrc=email.Newsletter.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.