

Objetivo

- Realizar una predicción usando múltiples muestras

Tabla de contenido

- Construir módulos personalizados

Clase Linear

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

Establecemos la semilla aleatoria:

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

Out[2]: <torch._C.Generator at 0x1fc7904d250>

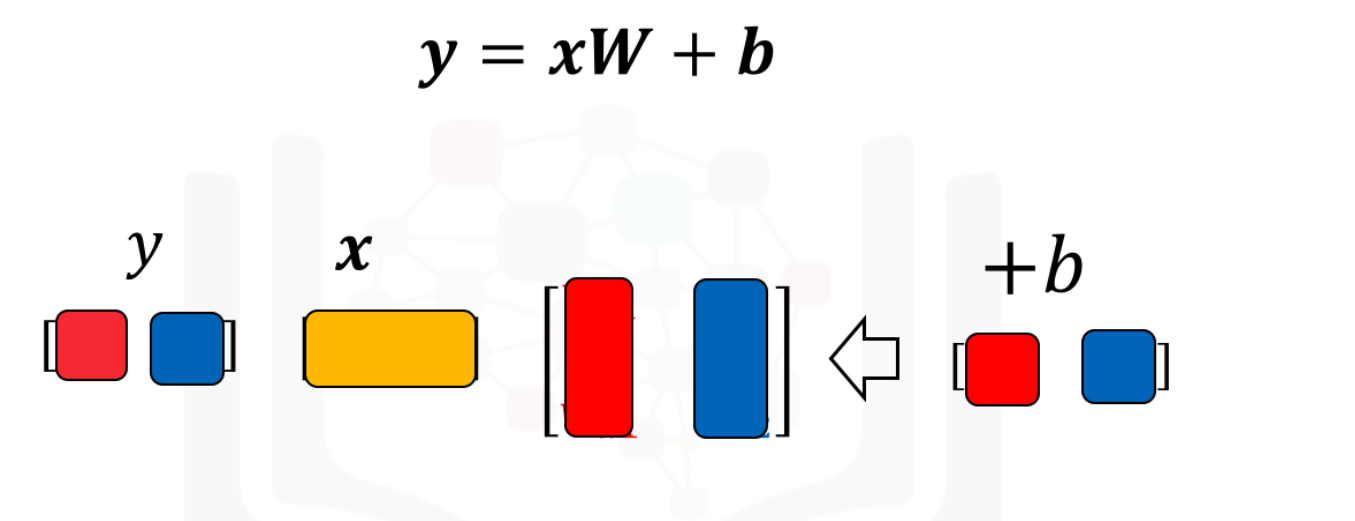
```
In [4]: class linear_regression(nn.Module):
def __init__(self,input_size,output_size):
super(linear_regression,self).__init__()
self.linear=nn.Linear(input_size,output_size)
def forward(self,x):
yhat=self.linear(x)
return yhat
```

Creamos un objeto regresión lineal.

```
In [5]: model=linear_regression(1,10)
model(torch.tensor([1.0]))
```

Out[5]: tensor([0.7926, -0.3920, 0.1714, 0.0797, -1.0143, 0.5097, -0.0608, 0.5047, 1.0132, 0.1887], grad_fn=<AddBackward0>)

Podemos usar el siguiente diagrama para representar el modelo:



podemos ver los parámetros:

```
In [6]: list(model.parameters())
```

Out[6]: [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)]

podemos crear un tensor con 2 filas representando una muestra de datos

```
In [7]: x=torch.tensor([[1.0]])
```

realizamos una predicción:

```
In [8]: yhat=model(x)
yhat
```

Out[8]: tensor([[0.7926, -0.3920, 0.1714, 0.0797, -1.0143, 0.5097, -0.0608, 0.5047, 1.0132, 0.1887]], grad_fn=<AddmmBackward>)

cada fila en el siguiente tensor representa una muestra diferente

```
In [13]: X=torch.tensor([[1.0],[1.0],[3.0]])
X.size()
```

Out[13]: torch.Size([3, 1])

realizamos una predicción usando múltiples muestras

```
In [14]: Yhat=model(X)
Yhat
```

Out[14]: 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>)

la figura siguiente representa la operación, el rojo y azul representan los diferentes parámetros, y los diferentes tonos de verde representan muestras diferentes.

