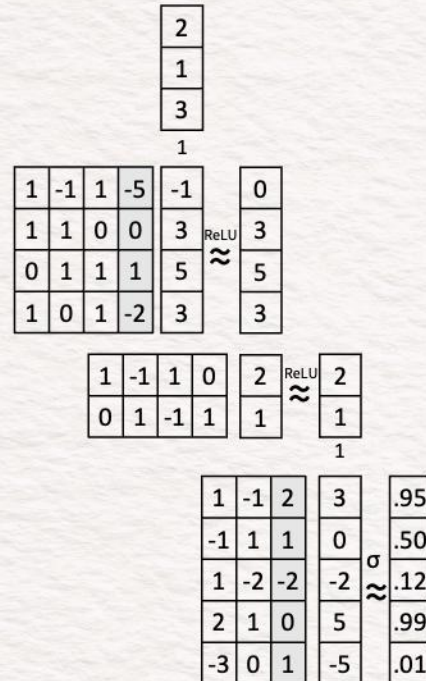


Multi Layer Perceptron in PyTorch

1

```

1 mlp_model = nn.Sequential(
2     nn.____(____,____, bias = ____),
3     nn.____(),
4     nn.____(____,____, bias = ____),
5     nn.____(),
6     nn.____(____,____, bias = ____),
7     nn.____()
8 )
    
```



Hints:

Linear Layer: { Identity | Linear | Bilinear }

Activation Function: { ReLU | Tanh | Sigmoid }

in_features: { int }

out_features: { int }

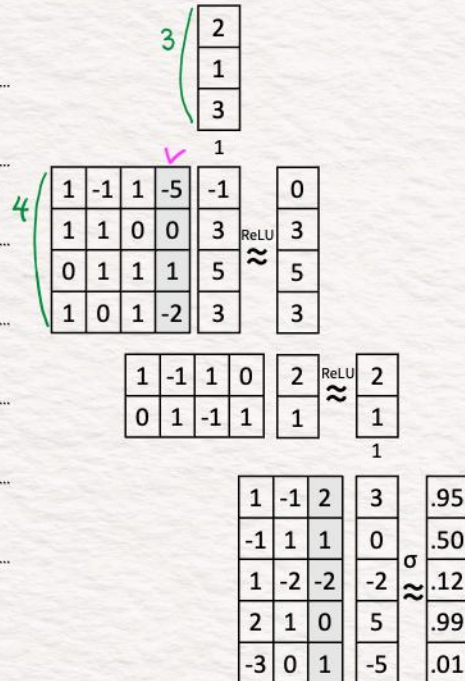
bias: { T | F }

Multi Layer Perceptron in PyTorch

2

```

1 mlp_model = nn.Sequential(
2     nn.Linear( 3, 4, bias = T ),
3     nn._____,
4     nn._____( _____, bias = _____ ),
5     nn._____,
6     nn._____( _____, bias = _____ ),
7     nn._____()
8 )
    
```



Hints:

Linear Layer: { Identity | Linear | Bilinear }

Activation Function: { ReLU | Tanh | Sigmoid }

in_features: { int }

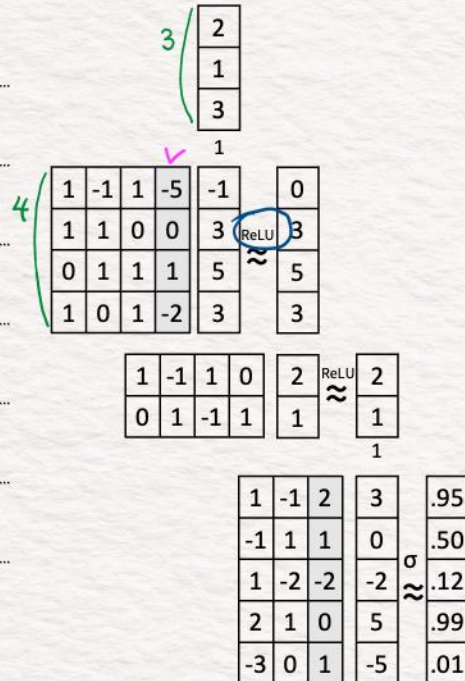
out_features: { int }

bias: { T | F }

Multi Layer Perceptron in PyTorch

3

```
1 mlp_model = nn.Sequential(  
2     nn.Linear( 3, 4, bias = T ),  
3     nn.ReLU(),  
4     nn._____( ____, ____, bias = ____ ),  
5     nn._____( ),  
6     nn._____( ____, ____, bias = ____ ),  
7     nn._____( )  
8 )
```



Hints:

Linear Layer: { Identity | Linear | Bilinear }

Activation Function: { ReLU | Tanh | Sigmoid }

in_features: { int }

out_features: { int }

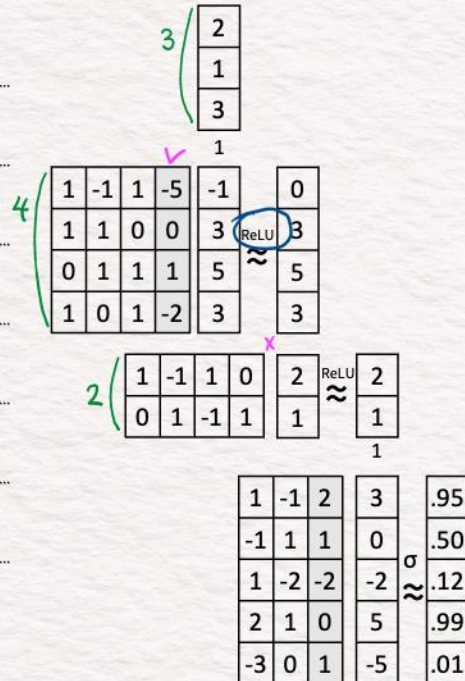
bias: { T | F }

Multi Layer Perceptron in PyTorch 🔥

4

```

1 mlp_model = nn.Sequential(
2     nn.Linear( 3, 4, bias = T ),
3     nn.ReLU(),
4     nn.Linear( 4, 2, bias = F ),
5     nn._____,
6     nn._____( _____, _____, bias = _____ ),
7     nn._____)
8 )
    
```



Hints:

Linear Layer: { Identity | Linear | Bilinear }

Activation Function: { ReLU | Tanh | Sigmoid }

in_features: { int }

out_features: { int }

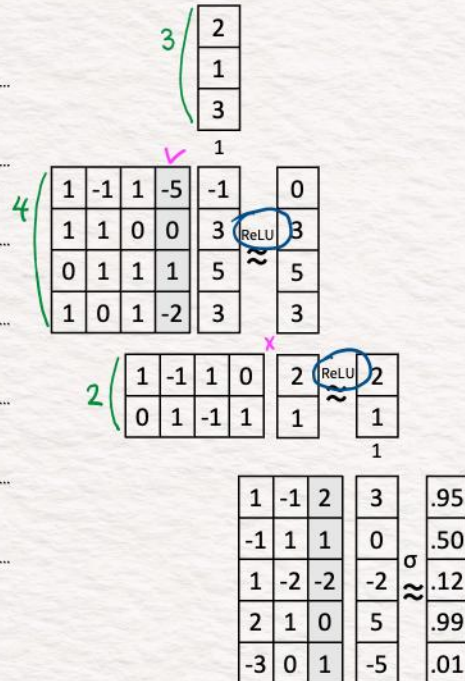
bias: { T | F }

Multi Layer Perceptron in PyTorch

5

```

1 mlp_model = nn.Sequential(
2     nn.Linear( 3, 4, bias = T ),
3     nn.ReLU(),
4     nn.Linear( 4, 2, bias = F ),
5     nn.ReLU(),
6     nn._____( ____, ____, bias = ____ ),
7     nn._____( )
8 )
    
```



Hints:

Linear Layer: { Identity | Linear | Bilinear }

Activation Function: { ReLU | Tanh | Sigmoid }

in_features: { int }

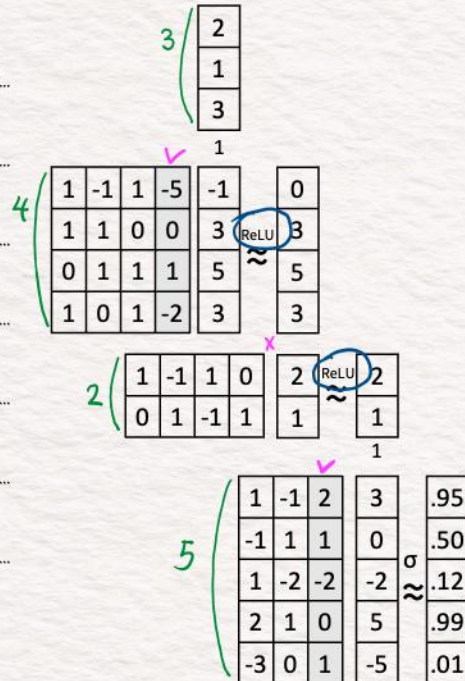
out_features: { int }

bias: { T | F }

Multi Layer Perceptron in PyTorch

6

```
1 mlp_model = nn.Sequential(  
2     nn.Linear( 3, 4, bias = T ),  
3     nn.ReLU(),  
4     nn.Linear( 4, 2, bias = F ),  
5     nn.ReLU(),  
6     nn.Linear( 2, 5, bias = T ),  
7     nn.____(),  
8 )
```



Hints:

Linear Layer: { Identity | Linear | Bilinear }

Activation Function: { ReLU | Tanh | Sigmoid }

in_features: { int }

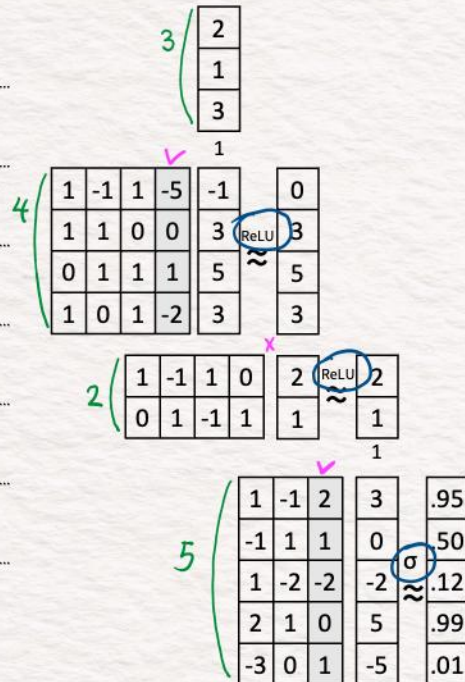
out_features: { int }

bias: { T | F }

Multi Layer Perceptron in PyTorch

7

```
1 mlp_model = nn.Sequential(  
2     nn.Linear( 3, 4, bias = T ),  
3     nn.ReLU(),  
4     nn.Linear( 4, 2, bias = F ),  
5     nn.ReLU(),  
6     nn.Linear( 2, 5, bias = T ),  
7     nn.Sigmoid(),  
8 )
```



Hints:

Linear Layer: { Identity | Linear | Bilinear }

Activation Function: { ReLU | Tanh | Sigmoid }

in_features: { int }

out_features: { int }

bias: { T | F }