

深度学习基础课程

Deep Learning Foundation Course



<https://www.streamingnology.com>



<https://github.com/streamingnology>



@streamingnology



@streamingnology



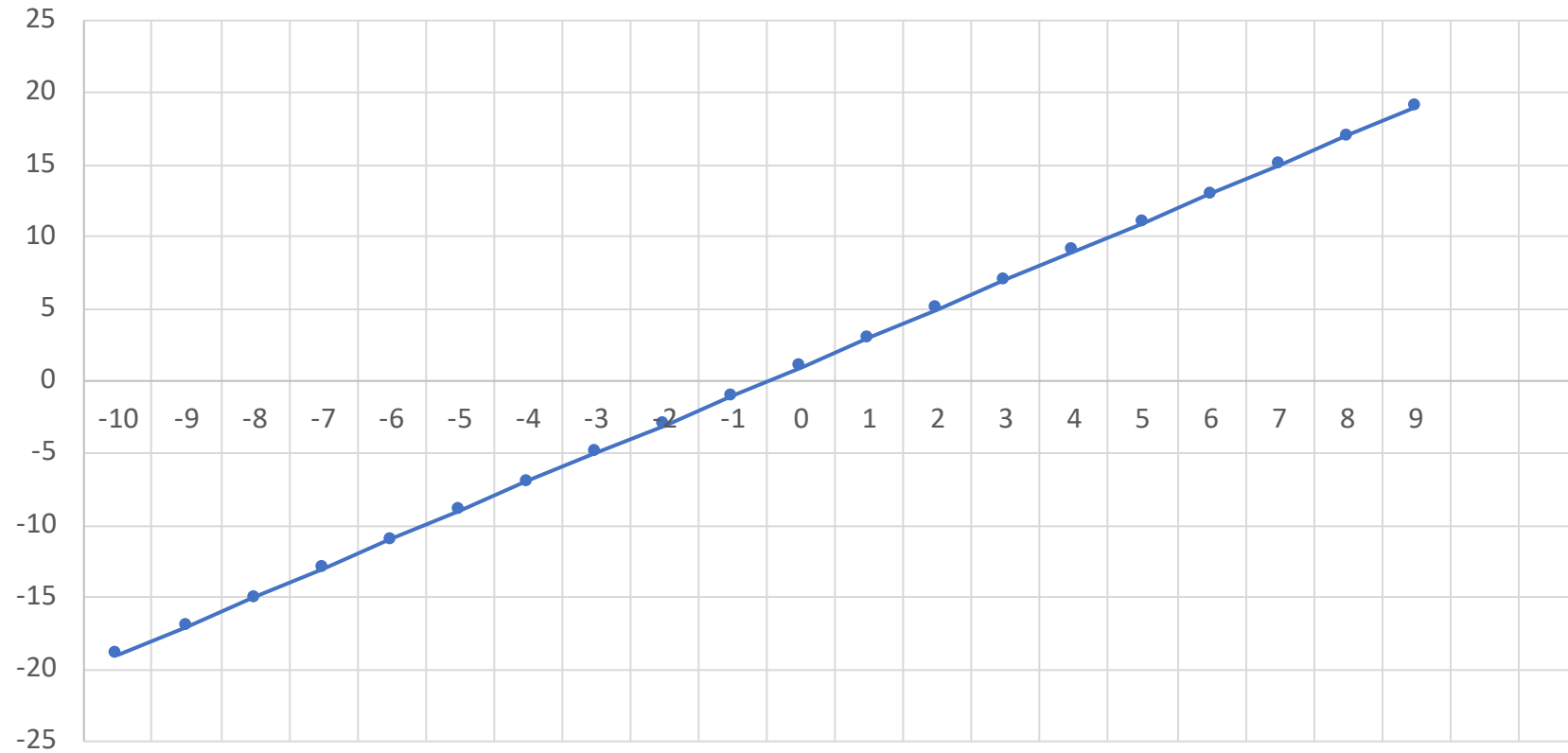
streamingnology

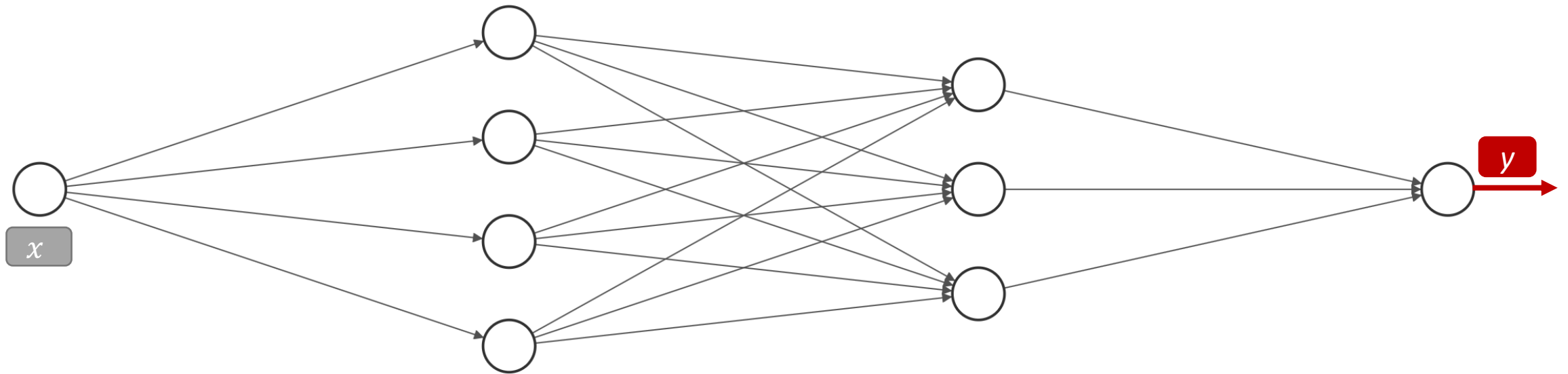
样本数：20个
输入数据：1个
输出数据：1个

输入数据：[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

输出数据：[-19, -17, -15, -13, -11, -9, -7, -5, -3, -1, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19]

$$f(x)=2x+1$$





Input Layer $\in \mathbb{R}^1$

Hidden Layer $\in \mathbb{R}^4$

Hidden Layer $\in \mathbb{R}^3$

Output Layer $\in \mathbb{R}^1$

```
import matplotlib.pyplot as plt
import tensorflow as tf
import numpy as np
```

```
def func(X):
    return 2*X+1
```

```
X = np.arange(-10, 10)
#[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Y = func(X)
#[-19, -17, -15, -13, -11, -9, -7, -5, -3, -1, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
```

Keras.layers.Dense层即是全连接层

units用来指定每层的神经元个数

没有使用非线性函数

```
h1 = tf.keras.layers.Dense(units=4, input_shape=[1])  
h2 = tf.keras.layers.Dense(units=3)
```

input_shape 输入层接收的数据矩阵

```
outputs = tf.keras.layers.Dense(units=1)
```

```
model = tf.keras.Sequential([h1, h2, outputs])
```

Keras.Sequential model即顺序模型
顺序模型是多个网络层的线性堆叠

Loss function 损失函数

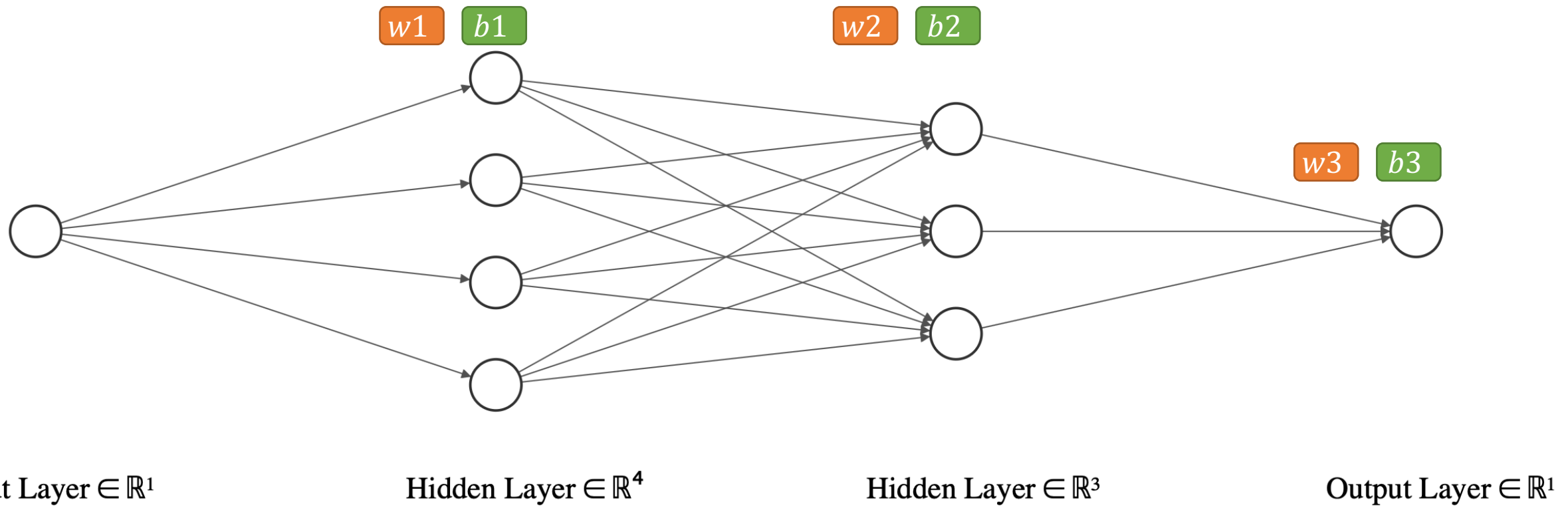
optimizer 优化器

```
model.compile(loss=tf.keras.losses.MeanSquaredError(), optimizer=tf.keras.optimizers.Adam(0.1))  
history = model.fit(X, Y, epochs=500, verbose=False)
```

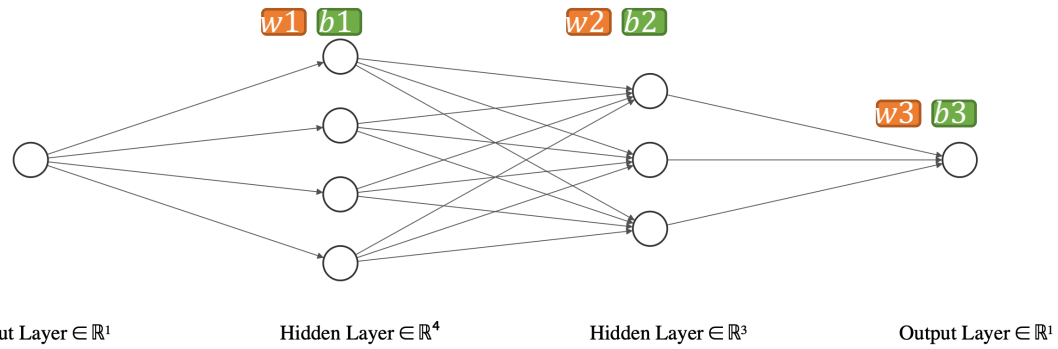
```
w1, b1 = h1.get_weights()
print("w1 = ", w1)
print("b1 = ", b1)

w2, b2 = h2.get_weights()
print("w2 = ")
print(w2)
print("b2 = ", b2)

w3, b3 = outputs.get_weights()
print("w3 = ")
print(w3)
print("b3 = ", b3)
```



Keras



$w1$ `[[-0.9617583 0.9393251 -0.9123167 0.15739506]]`

$b1$ `[0. 0.12659921 0. -0.13547021]`

$w2$ `[[0.3998077 0.9121256 -0.64119565]
[0.9239967 0.48231637 0.8087728]
[0.2642038 -0.11141396 0.0302161]
[0.23292127 -0.2934083 -0.73434186]]`

$b2$ `[0.15361612 0.15286203 0.14933605]`

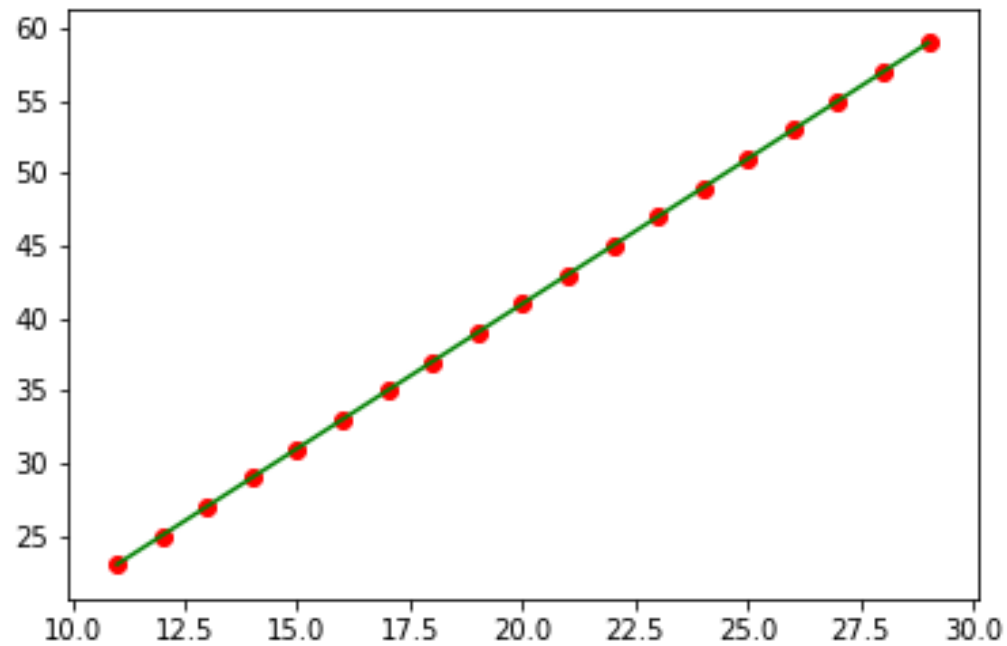
$w3$ `[[1.0965573]
[1.0701027]
[0.88724226]]`

$b3$ `[0.17090988]`

```
test_X = np.array([-20, 20])  
result = model.predict(test_X)  
print(result)
```

```
[[ -38.999996]  
 [ 40.999996]]
```

```
X_test = np.arange(11, 30)
Y_test = func(X_test)
Y_pred = model.predict(X_test)
plt.plot(X_test, Y_test, 'g')
plt.scatter(X_test, Y_pred, c='r')
plt.show()
```



Reference

1. Intro to TensorFlow for Deep Learning
<https://classroom.udacity.com/courses/ud187>
2. TensorFlow中文版
<https://www.youtube.com/playlist?list=PLQY2H8rRoyvwr-3IlvJXA1JyOlpcbIGa1>
3. Keras API docs
<https://keras.io/api/>