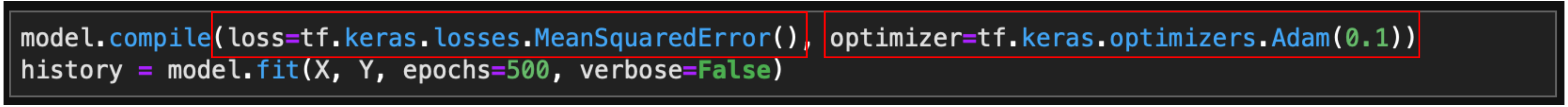
深度学习基础课程
Deep Learning Foundation Course

http://	https://www.streamingnology.com
	https://github.com/streamingnology
	@streamingnology
新浪微博 weibo.com	@streamingnology
YouTube	streamingnology

训练神经网络：loss function 损失函数

Loss function 损失函数

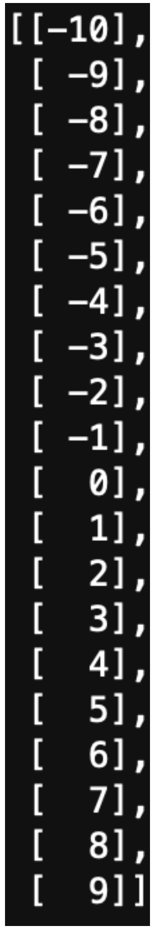
optimizer 优化器



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

训练神经网络：loss function 损失函数

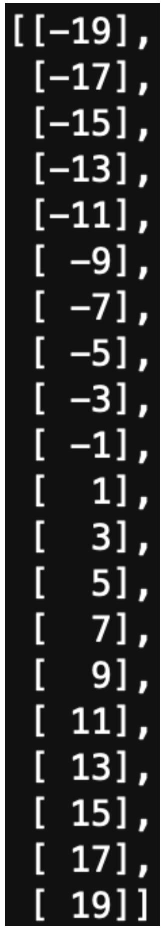
x



```
[[-10],
 [ -9],
 [ -8],
 [ -7],
 [ -6],
 [ -5],
 [ -4],
 [ -3],
 [ -2],
 [ -1],
 [  0],
 [  1],
 [  2],
 [  3],
 [  4],
 [  5],
 [  6],
 [  7],
 [  8],
 [  9]]
```

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

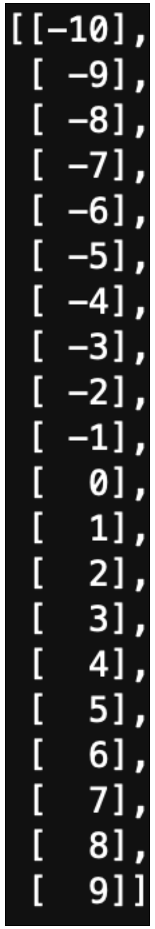
$f(x)$



```
[[-19],
 [-17],
 [-15],
 [-13],
 [-11],
 [ -9],
 [ -7],
 [ -5],
 [ -3],
 [ -1],
 [  1],
 [  3],
 [  5],
 [  7],
 [  9],
 [ 11],
 [ 13],
 [ 15],
 [ 17],
 [ 19]]
```

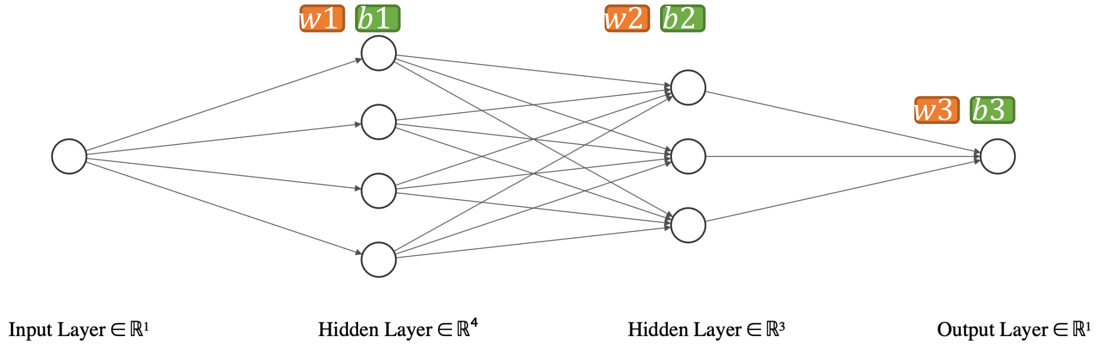
训练神经网络：loss function 损失函数

Epoch = 0

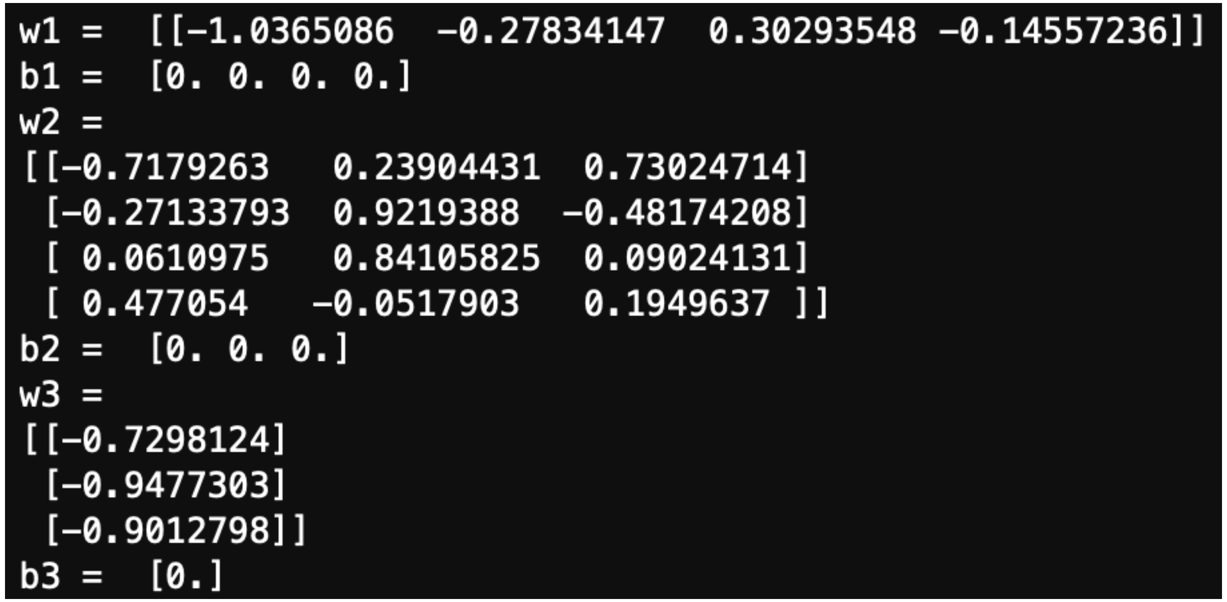


```
[[-10],
 [ -9],
 [ -8],
 [ -7],
 [ -6],
 [ -5],
 [ -4],
 [ -3],
 [ -2],
 [ -1],
 [  0],
 [  1],
 [  2],
 [  3],
 [  4],
 [  5],
 [  6],
 [  7],
 [  8],
 [  9]]
```

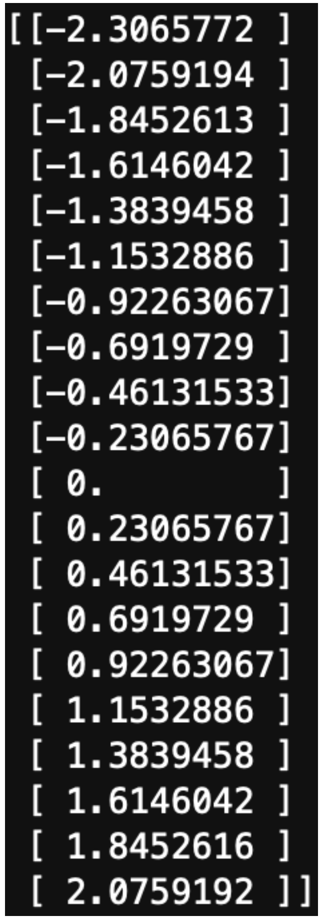
输入



神经网络网络参数

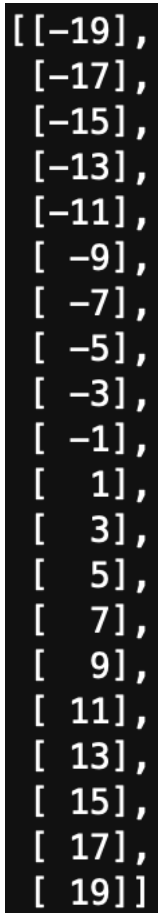


```
w1 =  [[-1.0365086  -0.27834147  0.30293548 -0.14557236]]
b1 =  [0. 0. 0. 0.]
w2 =
[[-0.7179263   0.23904431  0.73024714]
 [-0.27133793  0.9219388  -0.48174208]
 [ 0.0610975   0.84105825  0.09024131]
 [ 0.477054   -0.0517903   0.1949637 ]]
b2 =  [0. 0. 0.]
w3 =
[[-0.7298124]
 [-0.9477303]
 [-0.9012798]]
b3 =  [0.]
```



```
[[-2.3065772 ]
 [-2.0759194 ]
 [-1.8452613 ]
 [-1.6146042 ]
 [-1.3839458 ]
 [-1.1532886 ]
 [-0.92263067]
 [-0.6919729 ]
 [-0.46131533]
 [-0.23065767]
 [ 0.        ]
 [ 0.23065767]
 [ 0.46131533]
 [ 0.6919729 ]
 [ 0.92263067]
 [ 1.1532886 ]
 [ 1.3839458 ]
 [ 1.6146042 ]
 [ 1.8452616 ]
 [ 2.0759192 ]]
```

预测值

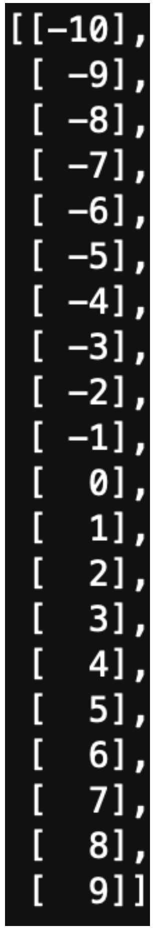


```
[[-19],
 [-17],
 [-15],
 [-13],
 [-11],
 [ -9],
 [ -7],
 [ -5],
 [ -3],
 [ -1],
 [  1],
 [  3],
 [  5],
 [  7],
 [  9],
 [ 11],
 [ 13],
 [ 15],
 [ 17],
 [ 19]]
```

真实值

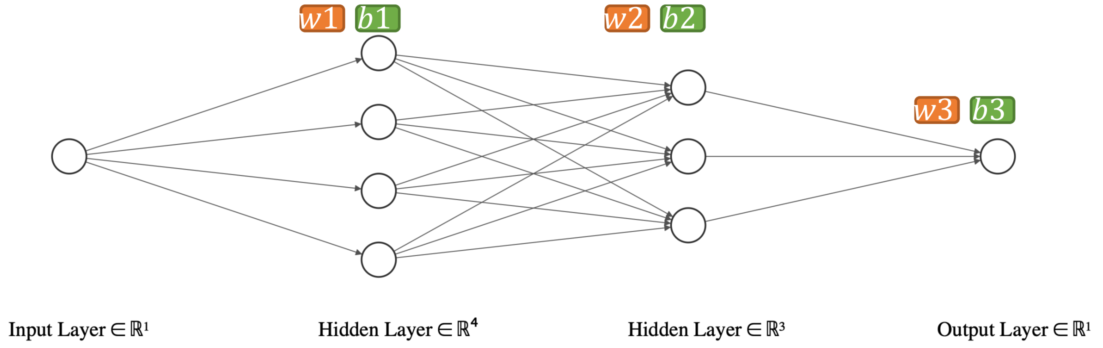
训练神经网络：loss function 损失函数

Epoch = 1

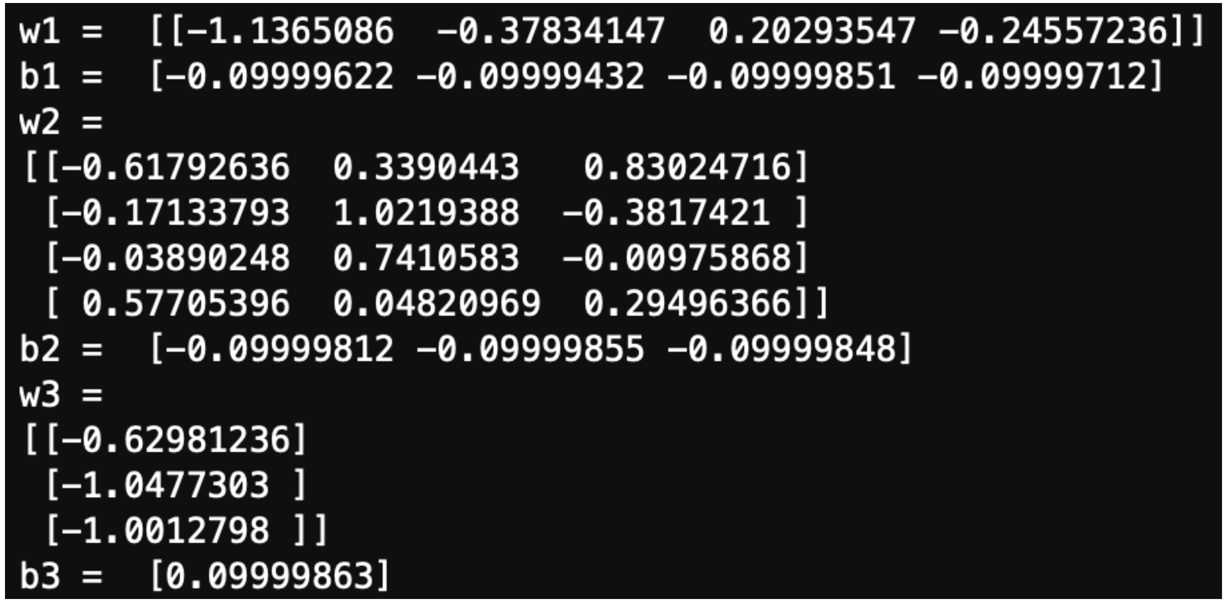


```
[[-10],
 [ -9],
 [ -8],
 [ -7],
 [ -6],
 [ -5],
 [ -4],
 [ -3],
 [ -2],
 [ -1],
 [  0],
 [  1],
 [  2],
 [  3],
 [  4],
 [  5],
 [  6],
 [  7],
 [  8],
 [  9]]
```

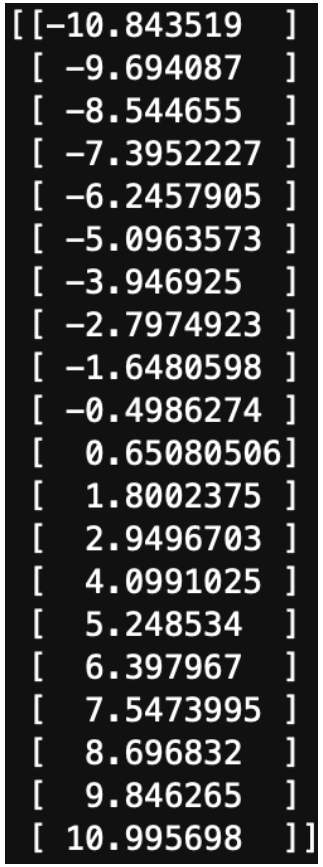
输入



神经网络网络参数

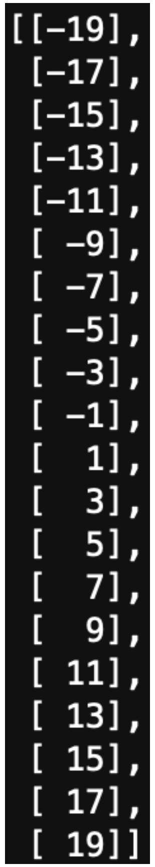


```
w1 =  [[-1.1365086  -0.37834147  0.20293547 -0.24557236]]
b1 =  [-0.09999622 -0.09999432 -0.09999851 -0.09999712]
w2 =
[[-0.61792636  0.3390443   0.83024716]
 [-0.17133793  1.0219388  -0.3817421 ]
 [-0.03890248  0.7410583  -0.00975868]
 [ 0.57705396  0.04820969  0.29496366]]
b2 =  [-0.09999812 -0.09999855 -0.09999848]
w3 =
[[-0.62981236]
 [-1.0477303 ]
 [-1.0012798 ]]
b3 =  [0.09999863]
```



```
[[-10.843519  ]
 [ -9.694087  ]
 [ -8.544655  ]
 [ -7.3952227 ]
 [ -6.2457905 ]
 [ -5.0963573 ]
 [ -3.946925  ]
 [ -2.7974923 ]
 [ -1.6480598 ]
 [ -0.4986274 ]
 [  0.65080506]
 [  1.8002375 ]
 [  2.9496703 ]
 [  4.0991025 ]
 [  5.248534  ]
 [  6.397967  ]
 [  7.5473995 ]
 [  8.696832  ]
 [  9.846265  ]
 [ 10.995698  ]]
```

预测值

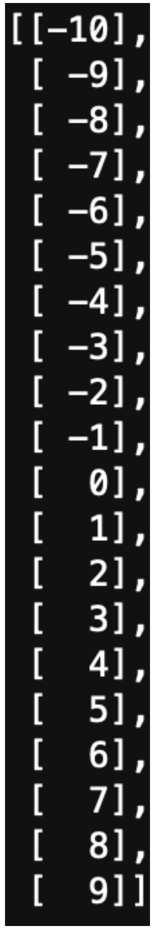


```
[[-19],
 [-17],
 [-15],
 [-13],
 [-11],
 [ -9],
 [ -7],
 [ -5],
 [ -3],
 [ -1],
 [  1],
 [  3],
 [  5],
 [  7],
 [  9],
 [ 11],
 [ 13],
 [ 15],
 [ 17],
 [ 19]]
```

真实值

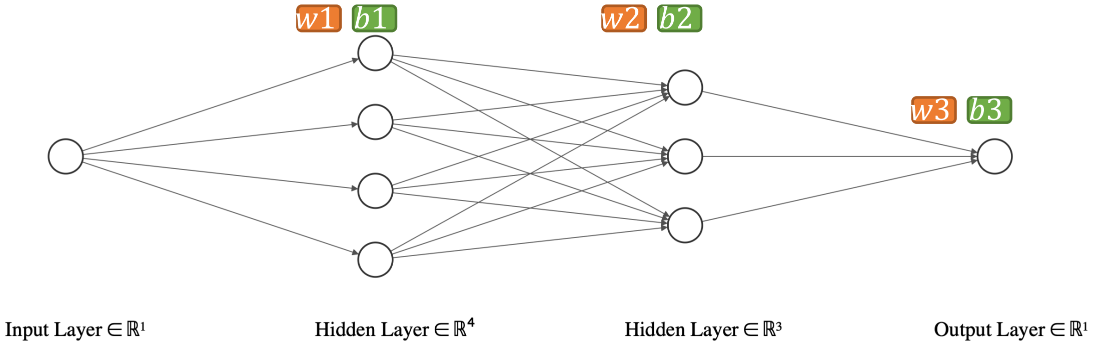
训练神经网络：loss function 损失函数

Epoch = 2

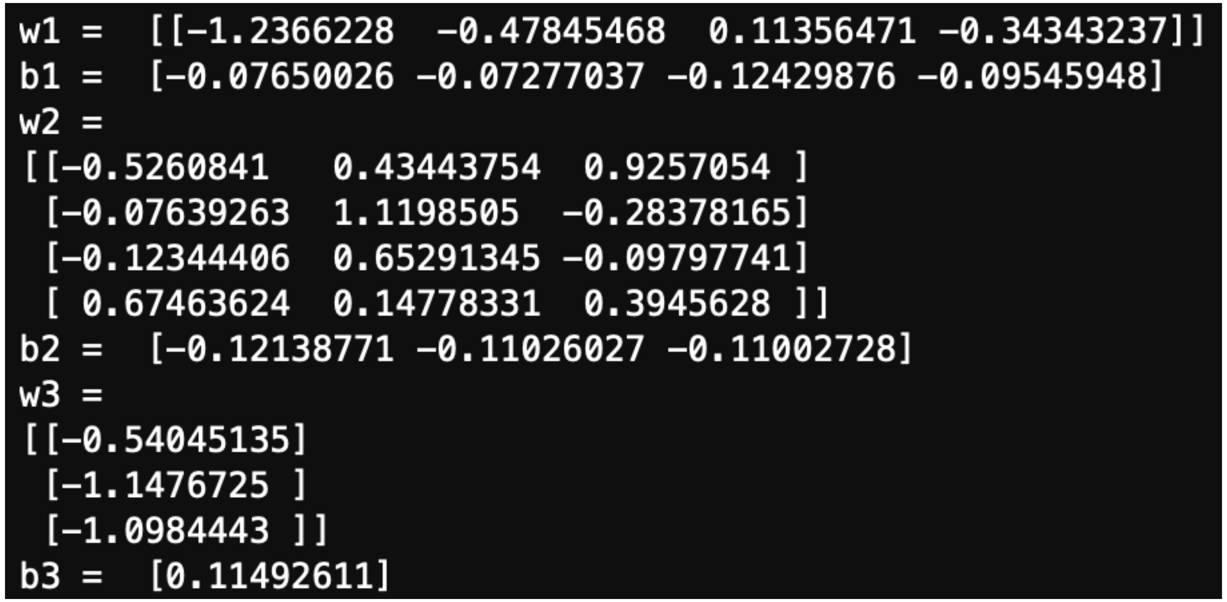


```
[[-10],
 [ -9],
 [ -8],
 [ -7],
 [ -6],
 [ -5],
 [ -4],
 [ -3],
 [ -2],
 [ -1],
 [  0],
 [  1],
 [  2],
 [  3],
 [  4],
 [  5],
 [  6],
 [  7],
 [  8],
 [  9]]
```

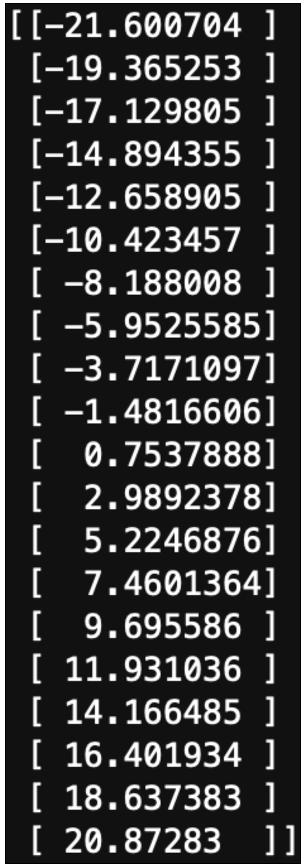
输入



神经网络网络参数

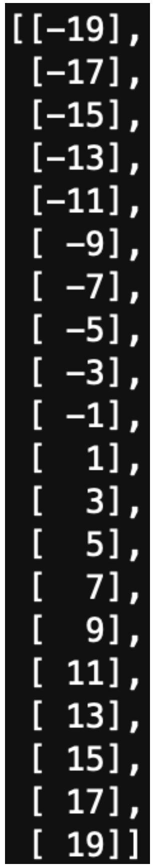


```
w1 =  [[-1.2366228  -0.47845468  0.11356471 -0.34343237]]
b1 =  [-0.07650026 -0.07277037 -0.12429876 -0.09545948]
w2 =
[[-0.5260841   0.43443754  0.9257054 ]
 [-0.07639263  1.1198505  -0.28378165]
 [-0.12344406  0.65291345 -0.09797741]
 [ 0.67463624  0.14778331  0.3945628 ]]
b2 =  [-0.12138771 -0.11026027 -0.11002728]
w3 =
[[-0.54045135]
 [-1.1476725 ]
 [-1.0984443 ]]
b3 =  [0.11492611]
```



```
[[-21.600704 ]
 [-19.365253 ]
 [-17.129805 ]
 [-14.894355 ]
 [-12.658905 ]
 [-10.423457 ]
 [ -8.188008 ]
 [ -5.9525585]
 [ -3.7171097]
 [ -1.4816606]
 [  0.7537888]
 [  2.9892378]
 [  5.2246876]
 [  7.4601364]
 [  9.695586 ]
 [ 11.931036 ]
 [ 14.166485 ]
 [ 16.401934 ]
 [ 18.637383 ]
 [ 20.87283  ]]
```

预测值

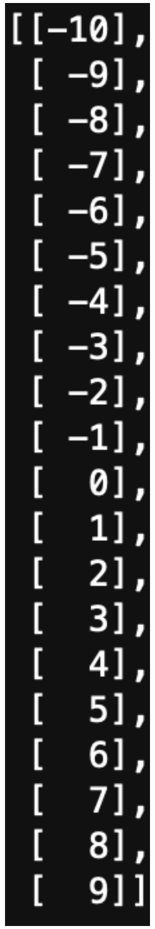


```
[[-19],
 [-17],
 [-15],
 [-13],
 [-11],
 [ -9],
 [ -7],
 [ -5],
 [ -3],
 [ -1],
 [  1],
 [  3],
 [  5],
 [  7],
 [  9],
 [ 11],
 [ 13],
 [ 15],
 [ 17],
 [ 19]]
```

真实值

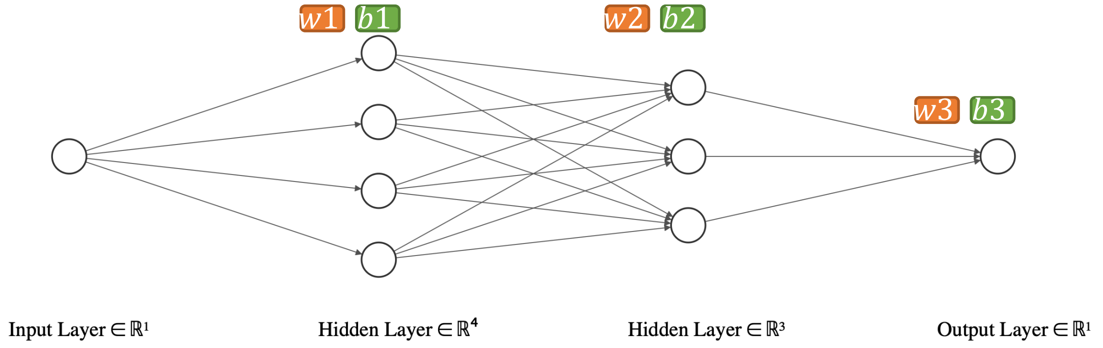
训练神经网络：loss function 损失函数

Epoch = 3

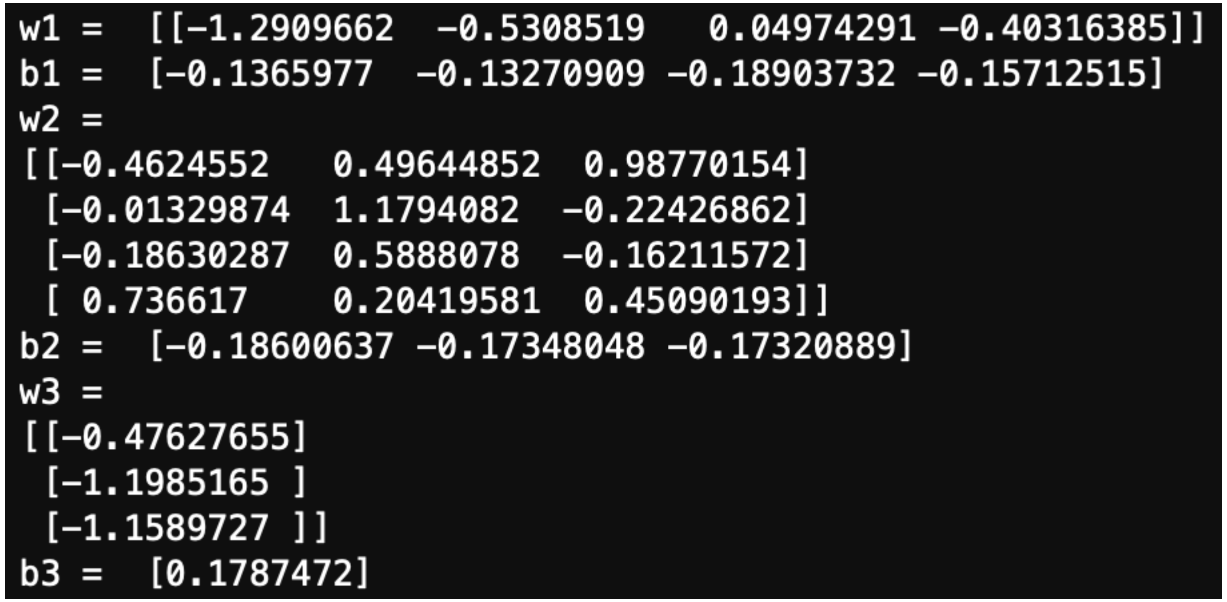


```
[[-10],
 [ -9],
 [ -8],
 [ -7],
 [ -6],
 [ -5],
 [ -4],
 [ -3],
 [ -2],
 [ -1],
 [  0],
 [  1],
 [  2],
 [  3],
 [  4],
 [  5],
 [  6],
 [  7],
 [  8],
 [  9]]
```

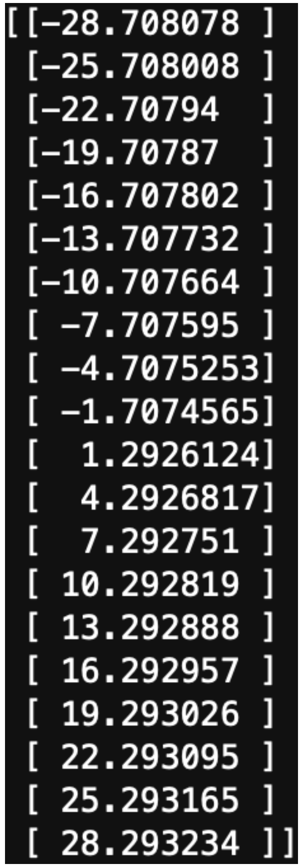
输入



神经网络网络参数

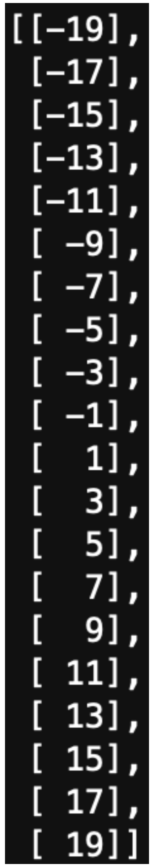


```
w1 =  [[-1.2909662  -0.5308519   0.04974291 -0.40316385]]
b1 =  [-0.1365977  -0.13270909 -0.18903732 -0.15712515]
w2 =
[[-0.4624552   0.49644852  0.98770154]
 [-0.01329874  1.1794082  -0.22426862]
 [-0.18630287  0.5888078  -0.16211572]
 [ 0.736617    0.20419581  0.45090193]]
b2 =  [-0.18600637 -0.17348048 -0.17320889]
w3 =
[[-0.47627655]
 [-1.1985165 ]
 [-1.1589727 ]]
b3 =  [0.1787472]
```



```
[[-28.708078 ]
 [-25.708008 ]
 [-22.70794  ]
 [-19.70787  ]
 [-16.707802 ]
 [-13.707732 ]
 [-10.707664 ]
 [ -7.707595 ]
 [ -4.7075253]
 [ -1.7074565]
 [  1.2926124]
 [  4.2926817]
 [  7.292751 ]
 [ 10.292819 ]
 [ 13.292888 ]
 [ 16.292957 ]
 [ 19.293026 ]
 [ 22.293095 ]
 [ 25.293165 ]
 [ 28.293234 ]]
```

预测值

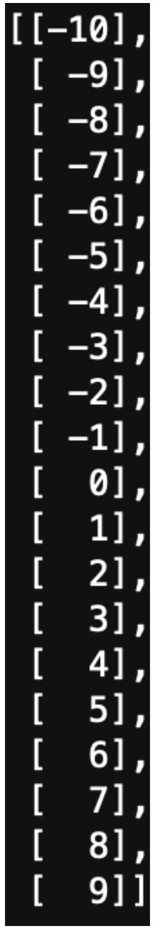


```
[[-19],
 [-17],
 [-15],
 [-13],
 [-11],
 [ -9],
 [ -7],
 [ -5],
 [ -3],
 [ -1],
 [  1],
 [  3],
 [  5],
 [  7],
 [  9],
 [ 11],
 [ 13],
 [ 15],
 [ 17],
 [ 19]]
```

真实值

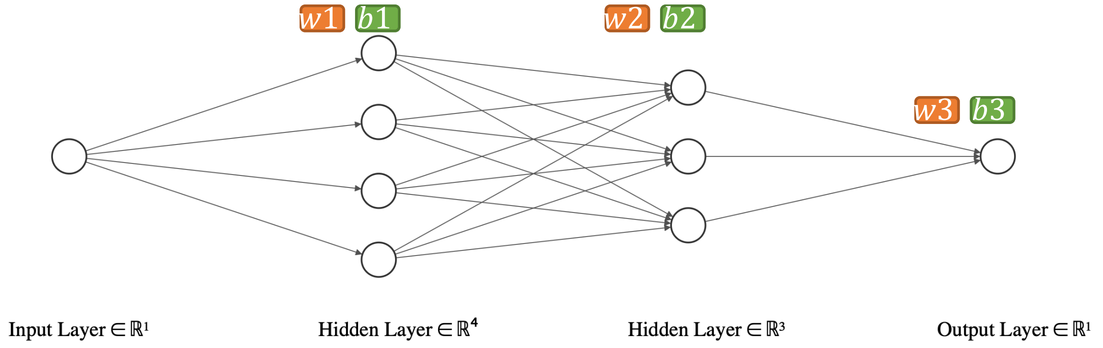
训练神经网络：loss function 损失函数

Epoch = 4

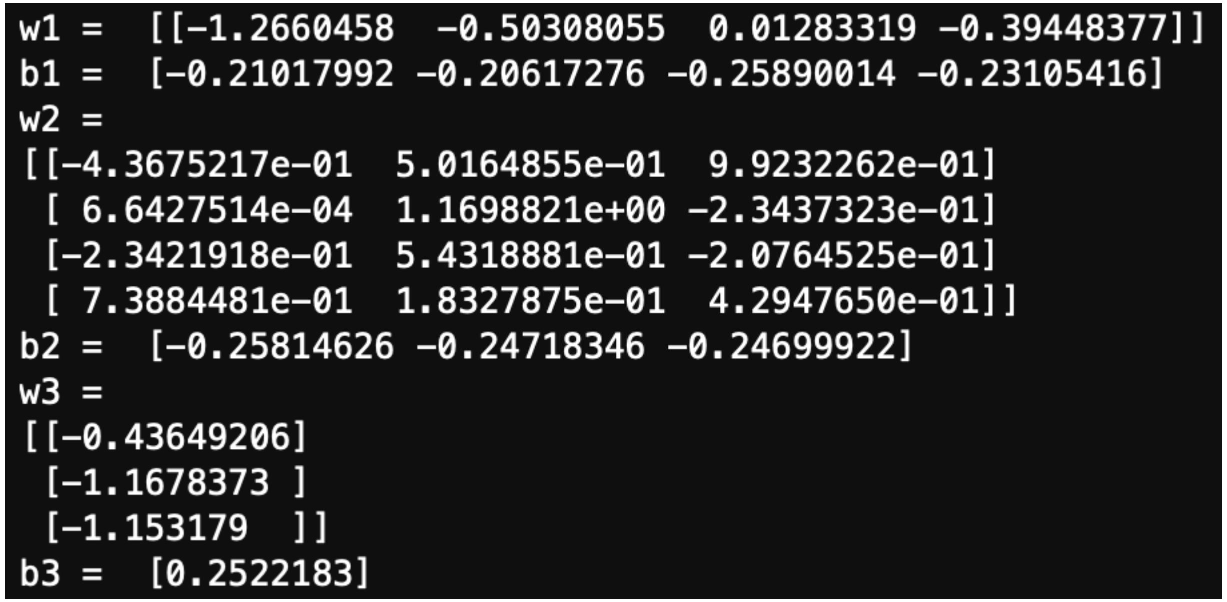


```
[[-10],
 [ -9],
 [ -8],
 [ -7],
 [ -6],
 [ -5],
 [ -4],
 [ -3],
 [ -2],
 [ -1],
 [  0],
 [  1],
 [  2],
 [  3],
 [  4],
 [  5],
 [  6],
 [  7],
 [  8],
 [  9]]
```

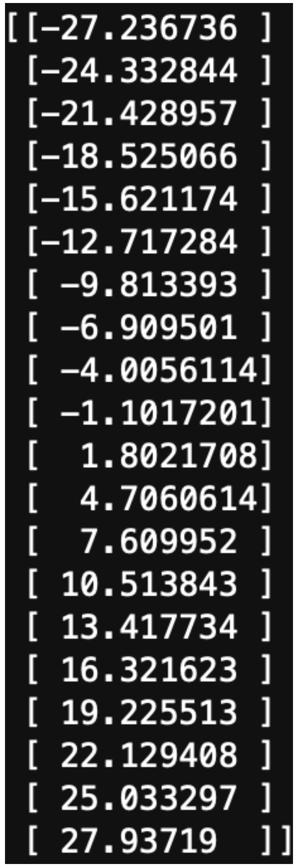
输入



神经网络网络参数

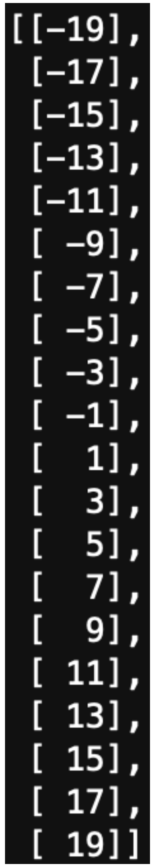


```
w1 =  [[-1.2660458   -0.50308055   0.01283319 -0.39448377]]
b1 =  [-0.21017992 -0.20617276 -0.25890014 -0.23105416]
w2 =
[[-4.3675217e-01   5.0164855e-01   9.9232262e-01]
 [ 6.6427514e-04   1.1698821e+00 -2.3437323e-01]
 [-2.3421918e-01   5.4318881e-01 -2.0764525e-01]
 [ 7.3884481e-01   1.8327875e-01   4.2947650e-01]]
b2 =  [-0.25814626 -0.24718346 -0.24699922]
w3 =
[[-0.43649206]
 [-1.1678373 ]
 [-1.153179  ]]
b3 =  [0.2522183]
```



```
[[-27.236736 ]
 [-24.332844 ]
 [-21.428957 ]
 [-18.525066 ]
 [-15.621174 ]
 [-12.717284 ]
 [ -9.813393 ]
 [ -6.909501 ]
 [ -4.0056114]
 [ -1.1017201]
 [  1.8021708]
 [  4.7060614]
 [  7.609952 ]
 [ 10.513843 ]
 [ 13.417734 ]
 [ 16.321623 ]
 [ 19.225513 ]
 [ 22.129408 ]
 [ 25.033297 ]
 [ 27.93719   ]]
```

预测值

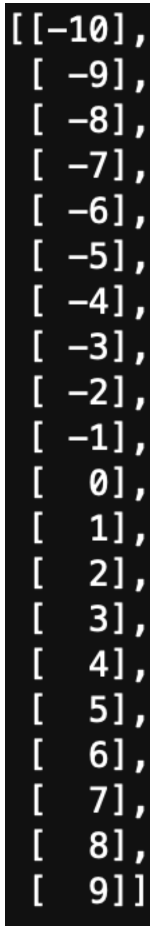


```
[[-19],
 [-17],
 [-15],
 [-13],
 [-11],
 [ -9],
 [ -7],
 [ -5],
 [ -3],
 [ -1],
 [  1],
 [  3],
 [  5],
 [  7],
 [  9],
 [ 11],
 [ 13],
 [ 15],
 [ 17],
 [ 19]]
```

真实值

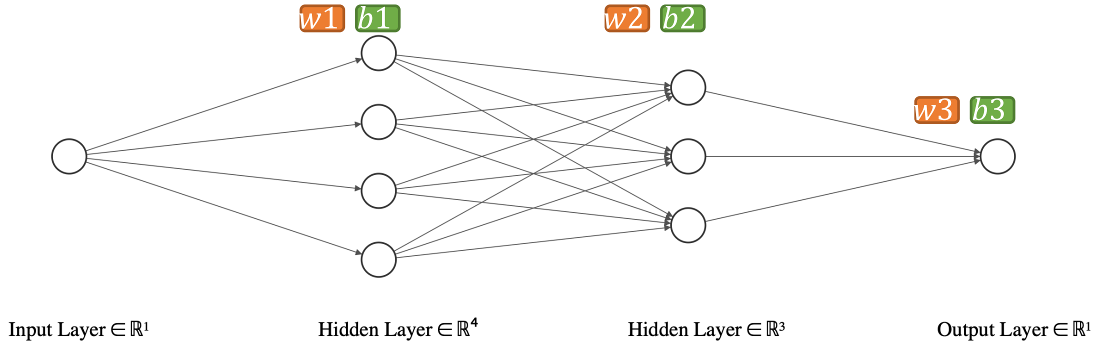
训练神经网络：loss function 损失函数

Epoch = 5



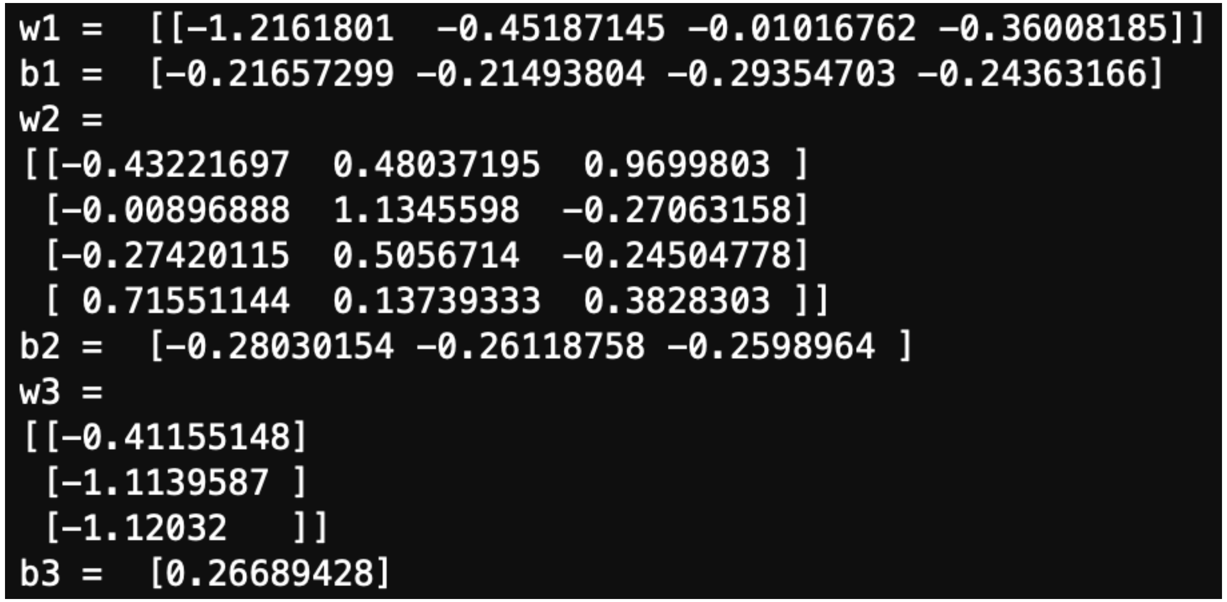
```
[[-10],
 [ -9],
 [ -8],
 [ -7],
 [ -6],
 [ -5],
 [ -4],
 [ -3],
 [ -2],
 [ -1],
 [  0],
 [  1],
 [  2],
 [  3],
 [  4],
 [  5],
 [  6],
 [  7],
 [  8],
 [  9]]
```

输入

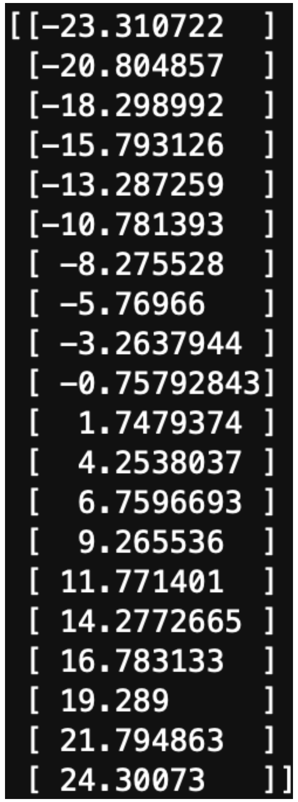


w1
b1
w2
b2
w3
b3
Input Layer ∈ $\mathbb{R}^1$
Hidden Layer ∈ $\mathbb{R}^4$
Hidden Layer ∈ $\mathbb{R}^3$
Output Layer ∈ $\mathbb{R}^1$

神经网络网络参数

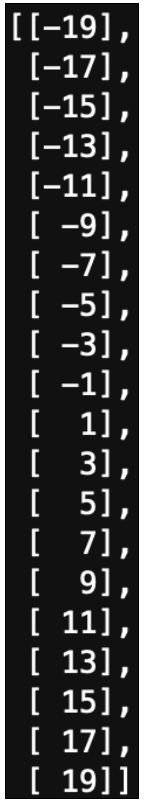


```
w1 =  [[-1.2161801  -0.45187145 -0.01016762 -0.36008185]]
b1 =  [-0.21657299 -0.21493804 -0.29354703 -0.24363166]
w2 =
[[-0.43221697  0.48037195  0.9699803 ]
 [-0.00896888  1.1345598  -0.27063158]
 [-0.27420115  0.5056714  -0.24504778]
 [ 0.71551144  0.13739333  0.3828303 ]]
b2 =  [-0.28030154 -0.26118758 -0.2598964 ]
w3 =
[[-0.41155148]
 [-1.1139587 ]
 [-1.12032   ]]
b3 =  [0.26689428]
```



```
[[-23.310722  ]
 [-20.804857  ]
 [-18.298992  ]
 [-15.793126  ]
 [-13.287259  ]
 [-10.781393  ]
 [ -8.275528  ]
 [ -5.76966   ]
 [ -3.2637944 ]
 [ -0.75792843]
 [  1.7479374 ]
 [  4.2538037 ]
 [  6.7596693 ]
 [  9.265536  ]
 [ 11.771401  ]
 [ 14.2772665 ]
 [ 16.783133  ]
 [ 19.289     ]
 [ 21.794863  ]
 [ 24.30073   ]]
```

预测值

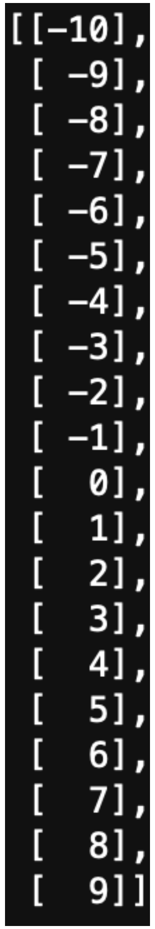


```
[[-19],
 [-17],
 [-15],
 [-13],
 [-11],
 [ -9],
 [ -7],
 [ -5],
 [ -3],
 [ -1],
 [  1],
 [  3],
 [  5],
 [  7],
 [  9],
 [ 11],
 [ 13],
 [ 15],
 [ 17],
 [ 19]]
```

真实值

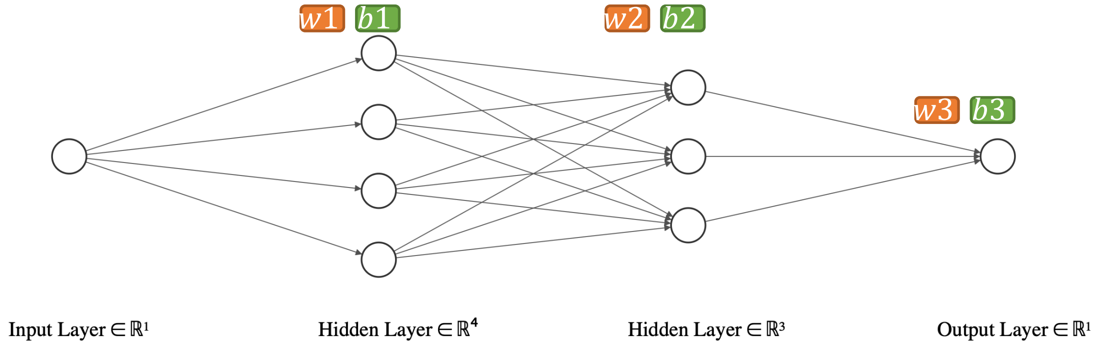
训练神经网络：loss function 损失函数

Epoch = 500

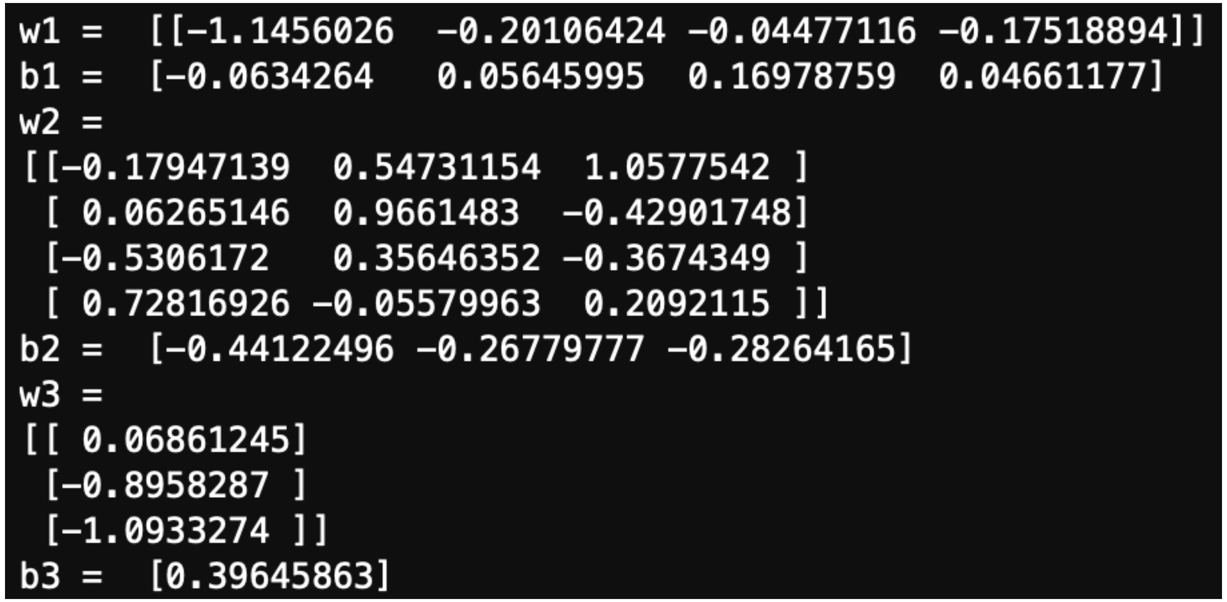


```
[[-10],
 [ -9],
 [ -8],
 [ -7],
 [ -6],
 [ -5],
 [ -4],
 [ -3],
 [ -2],
 [ -1],
 [  0],
 [  1],
 [  2],
 [  3],
 [  4],
 [  5],
 [  6],
 [  7],
 [  8],
 [  9]]
```

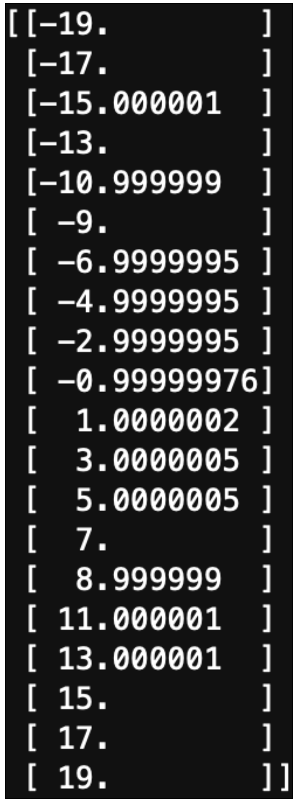
输入



神经网络网络参数

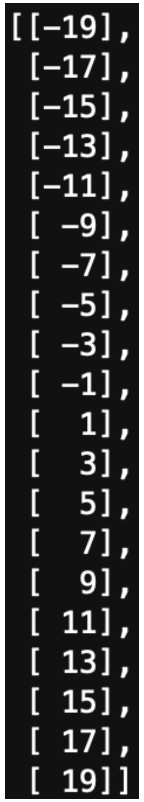


```
w1 =  [[-1.1456026  -0.20106424 -0.04477116 -0.17518894]]
b1 =  [-0.0634264   0.05645995  0.16978759  0.04661177]
w2 =
[[-0.17947139  0.54731154  1.0577542 ]
 [ 0.06265146  0.9661483  -0.42901748]
 [-0.5306172   0.35646352 -0.3674349 ]
 [ 0.72816926 -0.05579963  0.2092115 ]]
b2 =  [-0.44122496 -0.26779777 -0.28264165]
w3 =
[[ 0.06861245]
 [-0.8958287 ]
 [-1.0933274 ]]
b3 =  [0.39645863]
```



```
[[-19.        ]
 [-17.        ]
 [-15.000001  ]
 [-13.        ]
 [-10.999999  ]
 [ -9.        ]
 [ -6.9999995 ]
 [ -4.9999995 ]
 [ -2.9999995 ]
 [ -0.99999976]
 [  1.0000002 ]
 [  3.0000005 ]
 [  5.0000005 ]
 [  7.        ]
 [  8.999999  ]
 [ 11.000001  ]
 [ 13.000001  ]
 [ 15.        ]
 [ 17.        ]
 [ 19.        ]]
```

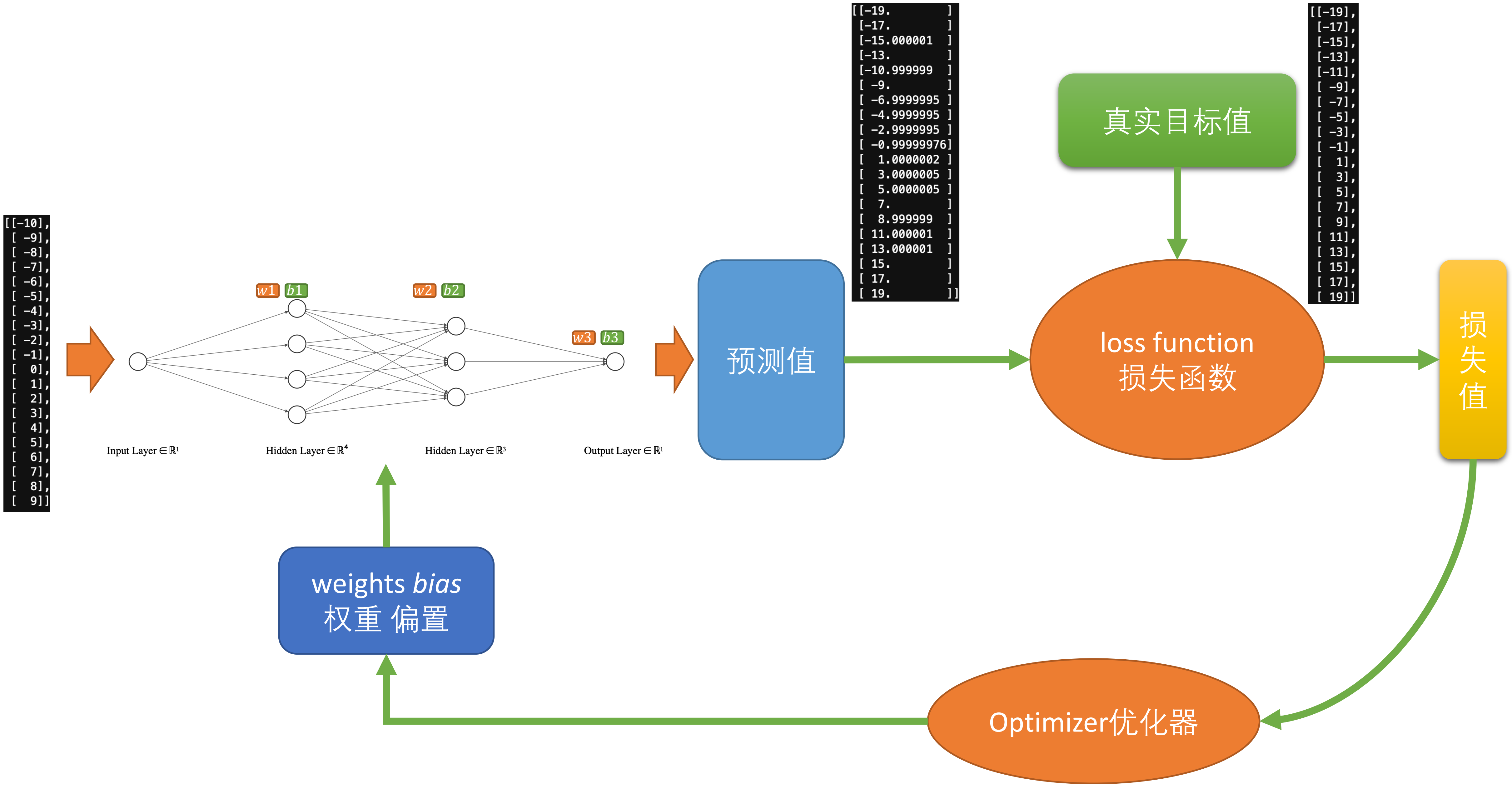
预测值



```
[[-19],
 [-17],
 [-15],
 [-13],
 [-11],
 [ -9],
 [ -7],
 [ -5],
 [ -3],
 [ -1],
 [  1],
 [  3],
 [  5],
 [  7],
 [  9],
 [ 11],
 [ 13],
 [ 15],
 [ 17],
 [ 19]]
```

真实值

训练神经网络：loss function 损失函数



```
[[-10],
 [ -9],
 [ -8],
 [ -7],
 [ -6],
 [ -5],
 [ -4],
 [ -3],
 [ -2],
 [ -1],
 [  0],
 [  1],
 [  2],
 [  3],
 [  4],
 [  5],
 [  6],
 [  7],
 [  8],
 [  9]]
```

预测值

```
[[-19.        ]
 [-17.        ]
 [-15.000001  ]
 [-13.        ]
 [-10.999999  ]
 [ -9.        ]
 [ -6.9999995 ]
 [ -4.9999995 ]
 [ -2.9999995 ]
 [ -0.99999976]
 [  1.0000002 ]
 [  3.0000005 ]
 [  5.0000005 ]
 [  7.        ]
 [  8.999999  ]
 [ 11.000001  ]
 [ 13.000001  ]
 [ 15.        ]
 [ 17.        ]
 [ 19.        ]]
```

真实目标值

```
[[-19],
 [-17],
 [-15],
 [-13],
 [-11],
 [ -9],
 [ -7],
 [ -5],
 [ -3],
 [ -1],
 [  1],
 [  3],
 [  5],
 [  7],
 [  9],
 [ 11],
 [ 13],
 [ 15],
 [ 17],
 [ 19]]
```

loss function
损失函数

损失值

Optimizer优化器

weights *bias*
权重 偏置

训练神经网络：loss function 损失函数

mean squared error
均方误差

$$\mathrm{m}se = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2$$

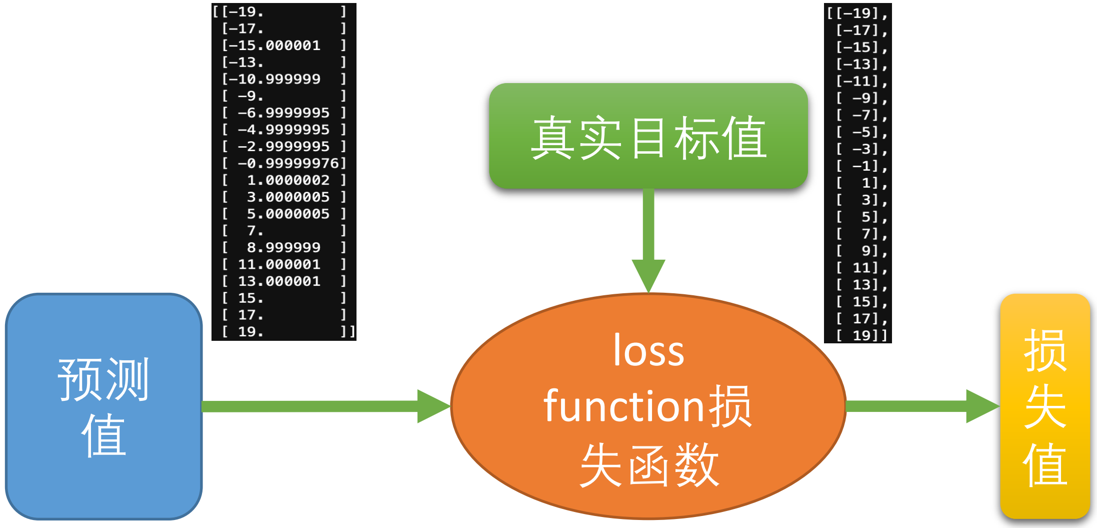
y：真实目标值
$\hat{y}$：预测值

越小越好

训练神经网络：loss function 损失函数

$$mse = \frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2$$

y：真实目标值
$\hat{y}$：预测值



```
[[-19.        ]
 [-17.        ]
 [-15.000001  ]
 [-13.        ]
 [-10.999999  ]
 [ -9.        ]
 [ -6.9999995 ]
 [ -4.9999995 ]
 [ -2.9999995 ]
 [ -0.99999976]
 [  1.0000002 ]
 [  3.0000005 ]
 [  5.0000005 ]
 [  7.        ]
 [  8.999999  ]
 [ 11.000001  ]
 [ 13.000001  ]
 [ 15.        ]
 [ 17.        ]
 [ 19.        ]]
```

```
[[-19],
 [-17],
 [-15],
 [-13],
 [-11],
 [ -9],
 [ -7],
 [ -5],
 [ -3],
 [ -1],
 [  1],
 [  3],
 [  5],
 [  7],
 [  9],
 [ 11],
 [ 13],
 [ 15],
 [ 17],
 [ 19]]
```

$$\frac{1}{20}\sum_{i=1}^{20}\big((-19-(-19))^2+(-17-(-17))^2+(-15.000001-(-15))^2+(-13-(-13))^2$$
$$+(-10.999999-(-11))^2+(-9-(-9))^2+(-6.9999995-(-7))^2+(-4.9999995-(-5))^2$$
$$+(-2.9999995-(-3))^2+(-0.99999976-(-1))^2+(1.0000002-1)^2+(3.0000005-3)^2$$
$$+(5.0000005-5)^2+(7-7)^2+(8.999999-9)^2+(11.00001-11)^2+(13.00001-13)^2+(15-15)^2$$
$$+(17-17)^2+(19-19)^2\big)$$

= 2.043476e-10

Reference

1. Python深度学习 [美] 弗朗索瓦·肖莱 著 张亮 译