Python实现的多层感知机实验报告与说明

> Author @陈屹峰20373860 全文共4字，阅读需要 分钟。

源代码列表

实验一：构建一个简单的反向传播神经网络

这里实现了一个3层的多层感知机。具体见`MLP_DIY.py`

1.1 实现反向传播获得梯度信息

本实验采用了反向传播算法的经典实现，如下

假设样本集为$\{x_n, t_n\}_{n=1}^N, x \in \mathbb{R}^D, t \in \mathbb{R}^M$

对每个样本都有网络对应输出$y(x_n, w, b)$，其中w, b为网络参数

采用最小均方误差原则设计损失函数函数

$$L(w,b) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n,w,b)-t_n)^2$$

寻找一个w, b使得L(w,b)最小，则有梯度下降方法

$$\nabla L(w,b) = 0$$

$$w^{(\tau+1)} = w^{(\tau)} - \alpha\frac{\partial L(w,b)}{\partial w}$$

$$b^{(\tau+1)} = b^{(\tau)} - \alpha\frac{\partial L(w,b)}{\partial b}$$

对于包含隐层的多层感知机，需要在前馈的过程中计算隐含层输入和输出，$h(\cdot)$为激活函数

$$z_j = h(a_j) = h(\sum_i w_{ji} z_i + b_{ji})$$

利用输出层输出、隐含层到输出层的连接权值、隐含层输出，就可以计算出误差函数对隐含层个神经元的偏导数

$$\frac{\partial L}{\partial w_{ji}} = \frac{\partial L}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}}$$
$$\frac{\partial L}{\partial b_{ji}} = \frac{\partial L}{\partial a_j}\frac{\partial a_j}{\partial b_{ji}}$$

在本实验中记，(k为j的下一层)

$$\delta_j \equiv \frac{\partial L}{\partial a_j} = \sum_k \frac{\partial L}{\partial a_k}\frac{\partial a_k}{\partial a_j} = h'(a_j)\sum_k w_{kj}\delta_k$$

又易知

$$\frac{\partial a_j}{\partial w_{ji}} = z_i, \frac{\partial a_j}{\partial b_{ji}} = 1$$

则有

$$\frac{\partial L}{\partial w_{ji}} = \delta_j z_i,$$
$$\frac{\partial L}{\partial b_{ji}} = \delta_j$$

1.2 使用梯度下降算法来更新参数

采用随机梯度下降对于相似的样本发生冗余，而采用批梯度下降计算量较大而且容易陷入局部极小阱。故

$$w^{(\tau+1)} = w^{(\tau)} - \alpha\frac{1}{m}\sum_{i=1}^{m}\frac{\partial L(x_i; w, b)}{\partial w}$$
$$b^{(\tau+1)} = b^{(\tau)} - \alpha\frac{1}{m}\sum_{i=1}^{m}\frac{\partial L(x_i; w, b)}{\partial b}$$

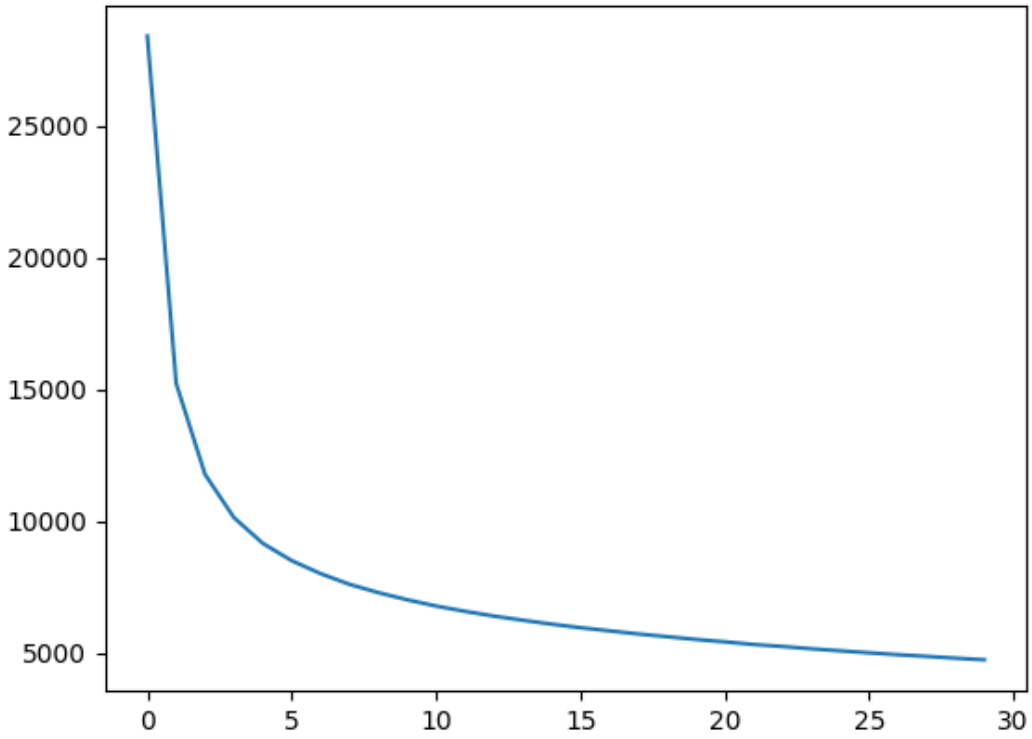
1.3 实际训练

激活函数采用$\sigma(x) = \frac{1}{1+\exp(-x)}$，学习率设置为0.3，损失为平方误差，采取小样本随机梯度下降方法进行参数更新，共进行30轮次更新。

以下是一次训练输出，(正确测试样本/总测试样本样本)

```
Epoch 0: 7716 / 10000 , lr = 0.3, loss = 28398.24299124524
Epoch 1: 8528 / 10000 , lr = 0.3, loss = 15225.257032650192
Epoch 2: 8753 / 10000 , lr = 0.3, loss = 11775.913169750436
Epoch 3: 8893 / 10000 , lr = 0.3, loss = 10140.55089713579
Epoch 4: 8952 / 10000 , lr = 0.3, loss = 9163.019415930743
Epoch 5: 9019 / 10000 , lr = 0.3, loss = 8507.954725905076
Epoch 6: 9050 / 10000 , lr = 0.3, loss = 8015.867094430018
Epoch 7: 9069 / 10000 , lr = 0.3, loss = 7610.846262320905
Epoch 8: 9124 / 10000 , lr = 0.3, loss = 7295.130000042745
```

```
Epoch 9: 9129 / 10000 , lr = 0.3, loss = 7024.728588921531
Epoch 10: 9170 / 10000 , lr = 0.3, loss = 6786.611984515212
Epoch 11: 9198 / 10000 , lr = 0.3, loss = 6585.694600407068
Epoch 12: 9208 / 10000 , lr = 0.3, loss = 6405.85155595964
Epoch 13: 9228 / 10000 , lr = 0.3, loss = 6248.8384234482255
Epoch 14: 9234 / 10000 , lr = 0.3, loss = 6099.826775911035
Epoch 15: 9246 / 10000 , lr = 0.3, loss = 5964.537814368393
Epoch 16: 9260 / 10000 , lr = 0.3, loss = 5845.7948898063405
Epoch 17: 9261 / 10000 , lr = 0.3, loss = 5727.611194910736
Epoch 18: 9282 / 10000 , lr = 0.3, loss = 5624.1152036273315
Epoch 19: 9279 / 10000 , lr = 0.3, loss = 5518.431295565117
Epoch 20: 9292 / 10000 , lr = 0.3, loss = 5428.453967176452
Epoch 21: 9302 / 10000 , lr = 0.3, loss = 5328.032730752089
Epoch 22: 9311 / 10000 , lr = 0.3, loss = 5252.336106319113
Epoch 23: 9324 / 10000 , lr = 0.3, loss = 5164.287252531484
Epoch 24: 9323 / 10000 , lr = 0.3, loss = 5088.747137739084
Epoch 25: 9324 / 10000 , lr = 0.3, loss = 5012.778581048862
Epoch 26: 9320 / 10000 , lr = 0.3, loss = 4940.431679087544
Epoch 27: 9341 / 10000 , lr = 0.3, loss = 4876.607985021676
Epoch 28: 9348 / 10000 , lr = 0.3, loss = 4808.844433810993
Epoch 29: 9355 / 10000 , lr = 0.3, loss = 4746.995679102938
```



实验二：进一步探索不同设置下模型的损失与正确率

为了控制变量，这里将以实验一为基线，每次变动一个因素，探索不同网络设置对于网络模型的影响。

2.1 Xavier权值初始化

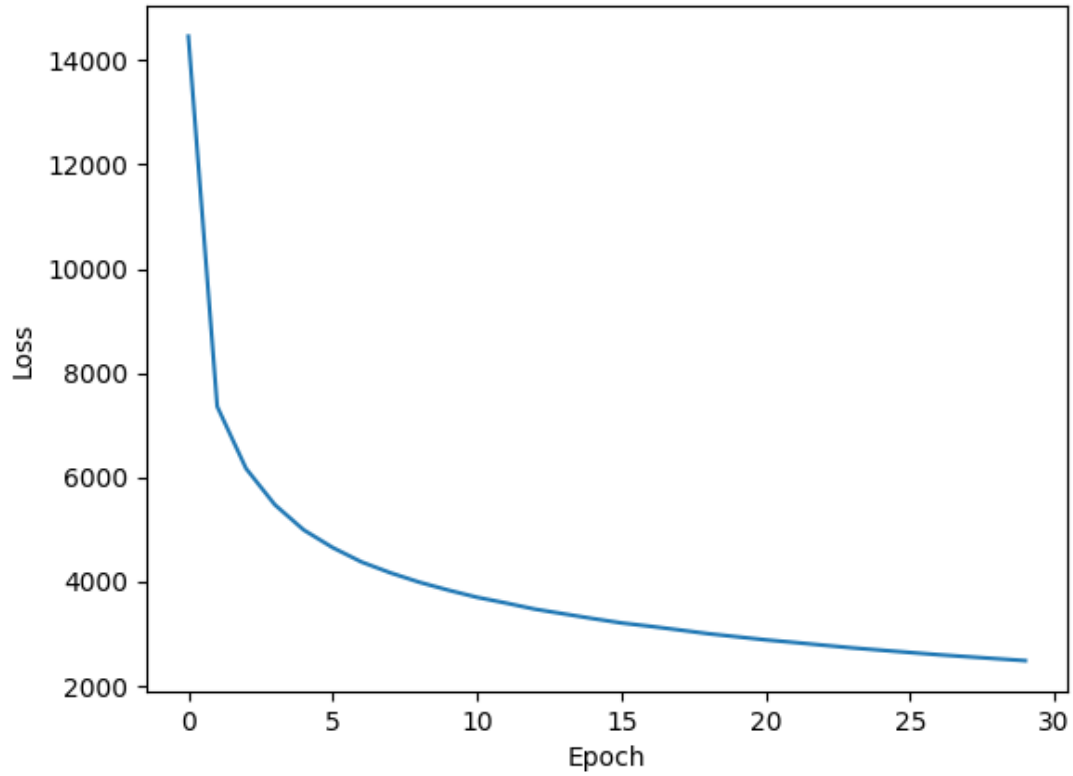
对于多层感知机，一种常见的权值初始化方式是正态分布。但是随着神经网络层数的增加，在参数更新的过程中，权值的均值不变，而方差急剧变小，且权值直方图在0点处收缩成一条线，这样的神经网络会导致梯度更新无法进行。

阅读文献可知，一种权值初始化的方法是Xavier权值初始化，希望使得每个元素的初始化采用均值为零的分布，且方差相同

$$W_i \sim U(-a, a), a = \sqrt{\frac{6}{n_1 + n_2}}$$

以下是实验结果。可以看到在Xavier初始化的条件下，明显加速了模型参数的迭代速度。

```
Epoch 0: 9157 / 10000 , lr = 0.3, loss = 14682.46610755577
Epoch 1: 9303 / 10000 , lr = 0.3, loss = 7372.535616610821
Epoch 2: 9346 / 10000 , lr = 0.3, loss = 6204.344497794502
Epoch 3: 9405 / 10000 , lr = 0.3, loss = 5550.029930630258
Epoch 4: 9459 / 10000 , lr = 0.3, loss = 5083.088563991321
Epoch 5: 9492 / 10000 , lr = 0.3, loss = 4732.610201021123
Epoch 6: 9505 / 10000 , lr = 0.3, loss = 4463.31775895462
Epoch 7: 9525 / 10000 , lr = 0.3, loss = 4239.264082086092
Epoch 8: 9539 / 10000 , lr = 0.3, loss = 4030.6002830935395
Epoch 9: 9553 / 10000 , lr = 0.3, loss = 3876.8491095668446
Epoch 10: 9545 / 10000 , lr = 0.3, loss = 3733.667991074618
Epoch 11: 9564 / 10000 , lr = 0.3, loss = 3607.0649628856913
Epoch 12: 9578 / 10000 , lr = 0.3, loss = 3497.7990235571137
Epoch 13: 9591 / 10000 , lr = 0.3, loss = 3394.413470135938
Epoch 14: 9592 / 10000 , lr = 0.3, loss = 3303.2512968308133
Epoch 15: 9593 / 10000 , lr = 0.3, loss = 3219.0761465601818
Epoch 16: 9595 / 10000 , lr = 0.3, loss = 3138.5274529328153
Epoch 17: 9603 / 10000 , lr = 0.3, loss = 3063.737526355156
Epoch 18: 9610 / 10000 , lr = 0.3, loss = 3002.2542595761197
Epoch 19: 9607 / 10000 , lr = 0.3, loss = 2931.6941233281964
Epoch 20: 9611 / 10000 , lr = 0.3, loss = 2871.2672290154806
Epoch 21: 9616 / 10000 , lr = 0.3, loss = 2823.275304899208
Epoch 22: 9618 / 10000 , lr = 0.3, loss = 2771.0431290697024
Epoch 23: 9622 / 10000 , lr = 0.3, loss = 2722.3562039886174
Epoch 24: 9629 / 10000 , lr = 0.3, loss = 2670.237697744435
Epoch 25: 9620 / 10000 , lr = 0.3, loss = 2643.072418085762
Epoch 26: 9622 / 10000 , lr = 0.3, loss = 2593.9156505373376
Epoch 27: 9613 / 10000 , lr = 0.3, loss = 2558.2155776358763
Epoch 28: 9631 / 10000 , lr = 0.3, loss = 2521.891766374496
Epoch 29: 9624 / 10000 , lr = 0.3, loss = 2482.10424031091
```



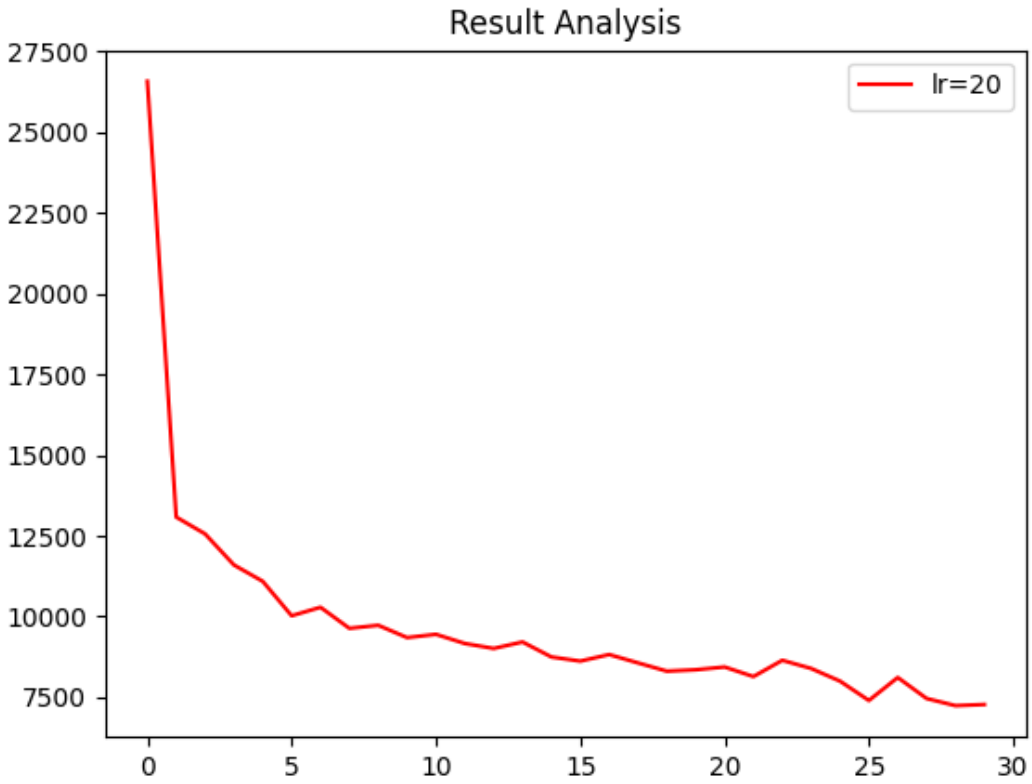
2.2 学习率

固定学习率

lr=20

```
Epoch 0: 8261 / 10000 , lr = 20, loss = 26575.42947590192
Epoch 1: 8725 / 10000 , lr = 20, loss = 13082.362247328892
Epoch 2: 8809 / 10000 , lr = 20, loss = 12561.888152023264
Epoch 3: 8809 / 10000 , lr = 20, loss = 11600.097104929446
Epoch 4: 8843 / 10000 , lr = 20, loss = 11091.695782459476
Epoch 5: 8885 / 10000 , lr = 20, loss = 10026.65672266425
Epoch 6: 8901 / 10000 , lr = 20, loss = 10289.565580093758
Epoch 7: 9012 / 10000 , lr = 20, loss = 9637.808898219862
Epoch 8: 9022 / 10000 , lr = 20, loss = 9731.675352908416
Epoch 9: 9015 / 10000 , lr = 20, loss = 9350.820167555035
Epoch 10: 9007 / 10000 , lr = 20, loss = 9451.956902436497
Epoch 11: 9026 / 10000 , lr = 20, loss = 9167.558368169255
Epoch 12: 9126 / 10000 , lr = 20, loss = 9014.655576761268
Epoch 13: 9057 / 10000 , lr = 20, loss = 9215.456974382412
Epoch 14: 9102 / 10000 , lr = 20, loss = 8750.74723783139
Epoch 15: 9127 / 10000 , lr = 20, loss = 8622.435443755023
Epoch 16: 9082 / 10000 , lr = 20, loss = 8825.738152481797
Epoch 17: 9029 / 10000 , lr = 20, loss = 8567.00593421823
Epoch 18: 9089 / 10000 , lr = 20, loss = 8311.513145424198
Epoch 19: 9071 / 10000 , lr = 20, loss = 8354.6649094772
Epoch 20: 9083 / 10000 , lr = 20, loss = 8440.571482451322
Epoch 21: 9116 / 10000 , lr = 20, loss = 8144.333681391886
Epoch 22: 9062 / 10000 , lr = 20, loss = 8648.072216853061
Epoch 23: 9177 / 10000 , lr = 20, loss = 8394.35933615639
```

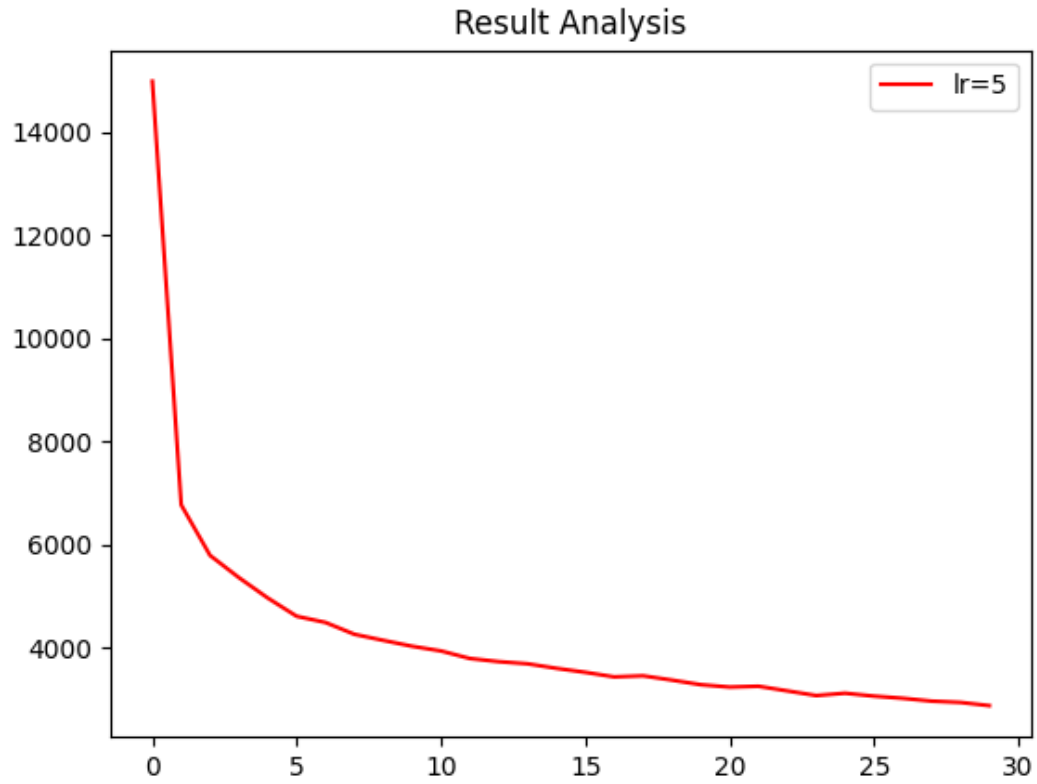
```
Epoch 24: 9207 / 10000 , lr = 20, loss = 8002.386319001124
Epoch 25: 9235 / 10000 , lr = 20, loss = 7404.260146398512
Epoch 26: 9159 / 10000 , lr = 20, loss = 8115.601131034691
Epoch 27: 9235 / 10000 , lr = 20, loss = 7464.891477672419
Epoch 28: 9188 / 10000 , lr = 20, loss = 7243.774205851251
Epoch 29: 9189 / 10000 , lr = 20, loss = 7278.863273422187
```



lr=5

```
Epoch 0: 9122 / 10000 , lr = 5, loss = 14983.78622368072
Epoch 1: 9291 / 10000 , lr = 5, loss = 6765.840079979906
Epoch 2: 9307 / 10000 , lr = 5, loss = 5785.614093552099
Epoch 3: 9414 / 10000 , lr = 5, loss = 5354.910399995234
Epoch 4: 9419 / 10000 , lr = 5, loss = 4959.500351048167
Epoch 5: 9438 / 10000 , lr = 5, loss = 4605.299652657747
Epoch 6: 9461 / 10000 , lr = 5, loss = 4485.133486616547
Epoch 7: 9467 / 10000 , lr = 5, loss = 4255.730853798753
Epoch 8: 9453 / 10000 , lr = 5, loss = 4137.9538126544085
Epoch 9: 9447 / 10000 , lr = 5, loss = 4023.467009479016
Epoch 10: 9467 / 10000 , lr = 5, loss = 3932.5185708419212
Epoch 11: 9490 / 10000 , lr = 5, loss = 3786.3704738109577
Epoch 12: 9442 / 10000 , lr = 5, loss = 3725.0113607515163
Epoch 13: 9481 / 10000 , lr = 5, loss = 3683.3693259078264
Epoch 14: 9508 / 10000 , lr = 5, loss = 3595.4831527196106
Epoch 15: 9504 / 10000 , lr = 5, loss = 3517.867928825622
Epoch 16: 9512 / 10000 , lr = 5, loss = 3429.437495827732
Epoch 17: 9495 / 10000 , lr = 5, loss = 3450.3439064302593
Epoch 18: 9498 / 10000 , lr = 5, loss = 3367.3663258479005
Epoch 19: 9495 / 10000 , lr = 5, loss = 3277.7255621129448
Epoch 20: 9528 / 10000 , lr = 5, loss = 3231.8620813158723
```

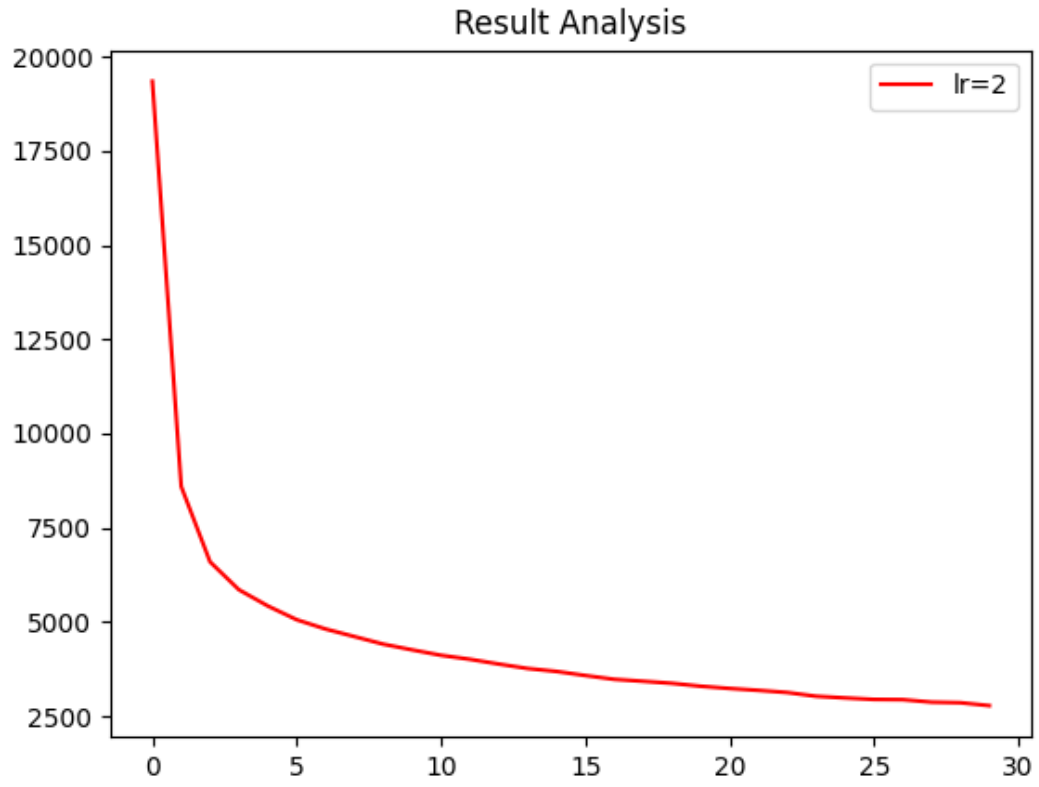
```
Epoch 21: 9515 / 10000 , lr = 5, loss = 3247.6463451841287
Epoch 22: 9507 / 10000 , lr = 5, loss = 3157.7655837675115
Epoch 23: 9505 / 10000 , lr = 5, loss = 3067.483648385053
Epoch 24: 9528 / 10000 , lr = 5, loss = 3112.3169699284554
Epoch 25: 9516 / 10000 , lr = 5, loss = 3056.1391117441526
Epoch 26: 9515 / 10000 , lr = 5, loss = 3013.944868099883
Epoch 27: 9518 / 10000 , lr = 5, loss = 2959.194112534059
Epoch 28: 9499 / 10000 , lr = 5, loss = 2934.013766485089
Epoch 29: 9501 / 10000 , lr = 5, loss = 2871.405137114586
```



lr=2

```
Epoch 0: 8063 / 10000 , lr = 2, loss = 19344.744029184378
Epoch 1: 9155 / 10000 , lr = 2, loss = 8593.557984862256
Epoch 2: 9234 / 10000 , lr = 2, loss = 6592.172495709144
Epoch 3: 9318 / 10000 , lr = 2, loss = 5852.753542512306
Epoch 4: 9305 / 10000 , lr = 2, loss = 5422.698806023302
Epoch 5: 9344 / 10000 , lr = 2, loss = 5057.787364601609
Epoch 6: 9379 / 10000 , lr = 2, loss = 4810.606315516037
Epoch 7: 9370 / 10000 , lr = 2, loss = 4610.109252453241
Epoch 8: 9416 / 10000 , lr = 2, loss = 4408.997579021576
Epoch 9: 9392 / 10000 , lr = 2, loss = 4257.420856607887
Epoch 10: 9422 / 10000 , lr = 2, loss = 4113.516361750415
Epoch 11: 9408 / 10000 , lr = 2, loss = 4007.687776804142
Epoch 12: 9410 / 10000 , lr = 2, loss = 3881.3206303652273
Epoch 13: 9419 / 10000 , lr = 2, loss = 3763.4812347256016
Epoch 14: 9418 / 10000 , lr = 2, loss = 3685.139095599333
Epoch 15: 9435 / 10000 , lr = 2, loss = 3577.231414124613
Epoch 16: 9405 / 10000 , lr = 2, loss = 3476.0482762064526
Epoch 17: 9458 / 10000 , lr = 2, loss = 3424.2368177910453
```

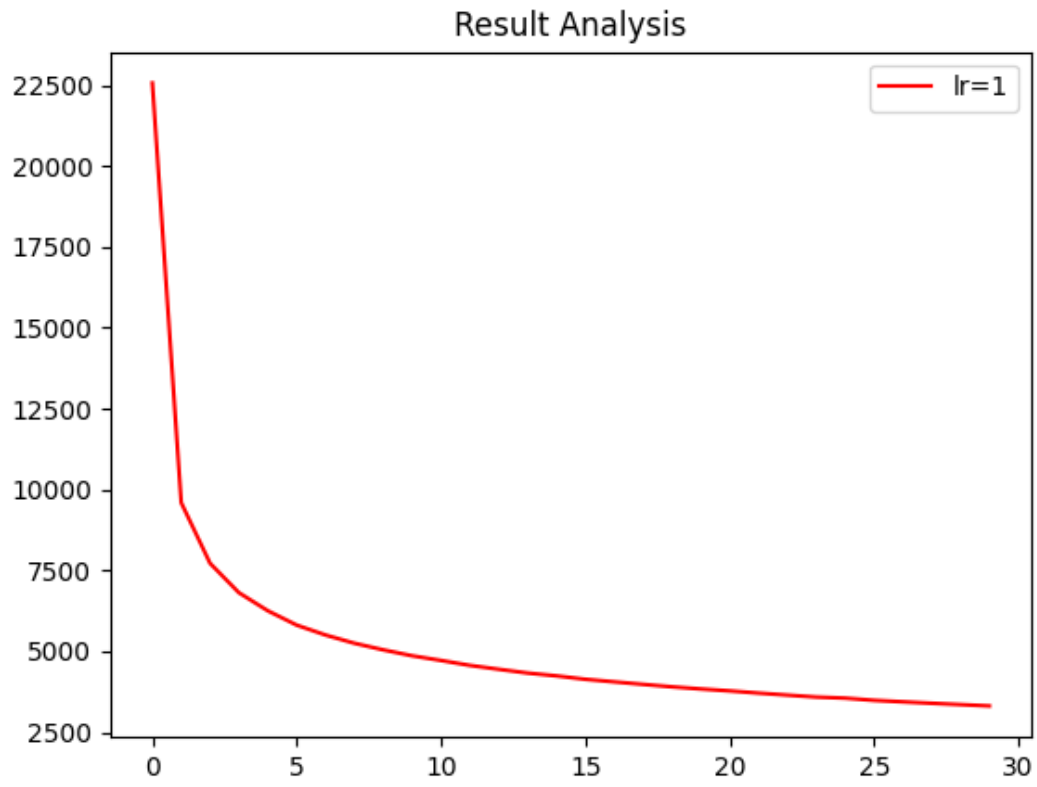
```
Epoch 18: 9459 / 10000 , lr = 2, loss = 3370.1757416395535
Epoch 19: 9426 / 10000 , lr = 2, loss = 3292.5135425550784
Epoch 20: 9445 / 10000 , lr = 2, loss = 3233.2531884399527
Epoch 21: 9458 / 10000 , lr = 2, loss = 3182.9749730896615
Epoch 22: 9468 / 10000 , lr = 2, loss = 3125.889964384232
Epoch 23: 9454 / 10000 , lr = 2, loss = 3025.8401711916003
Epoch 24: 9482 / 10000 , lr = 2, loss = 2981.9896070769664
Epoch 25: 9458 / 10000 , lr = 2, loss = 2942.361395818436
Epoch 26: 9422 / 10000 , lr = 2, loss = 2936.156804842875
Epoch 27: 9435 / 10000 , lr = 2, loss = 2869.835105244645
Epoch 28: 9477 / 10000 , lr = 2, loss = 2854.7984518738563
Epoch 29: 9454 / 10000 , lr = 2, loss = 2777.2305551915797
```



lr=1

```
Epoch 0: 8686 / 10000 , lr = 1, loss = 22563.317191037073
Epoch 1: 8984 / 10000 , lr = 1, loss = 9593.06896397636
Epoch 2: 9105 / 10000 , lr = 1, loss = 7717.911022415159
Epoch 3: 9182 / 10000 , lr = 1, loss = 6809.158012903695
Epoch 4: 9227 / 10000 , lr = 1, loss = 6251.120655056348
Epoch 5: 9249 / 10000 , lr = 1, loss = 5807.907859260497
Epoch 6: 9286 / 10000 , lr = 1, loss = 5501.386190917039
Epoch 7: 9304 / 10000 , lr = 1, loss = 5247.220046647261
Epoch 8: 9310 / 10000 , lr = 1, loss = 5043.027347353836
Epoch 9: 9354 / 10000 , lr = 1, loss = 4859.785486155152
Epoch 10: 9346 / 10000 , lr = 1, loss = 4715.436810621446
Epoch 11: 9358 / 10000 , lr = 1, loss = 4562.968338796924
Epoch 12: 9368 / 10000 , lr = 1, loss = 4442.808406249182
Epoch 13: 9379 / 10000 , lr = 1, loss = 4326.18623819474
Epoch 14: 9379 / 10000 , lr = 1, loss = 4238.901395435499
Epoch 15: 9395 / 10000 , lr = 1, loss = 4133.581844553996
```

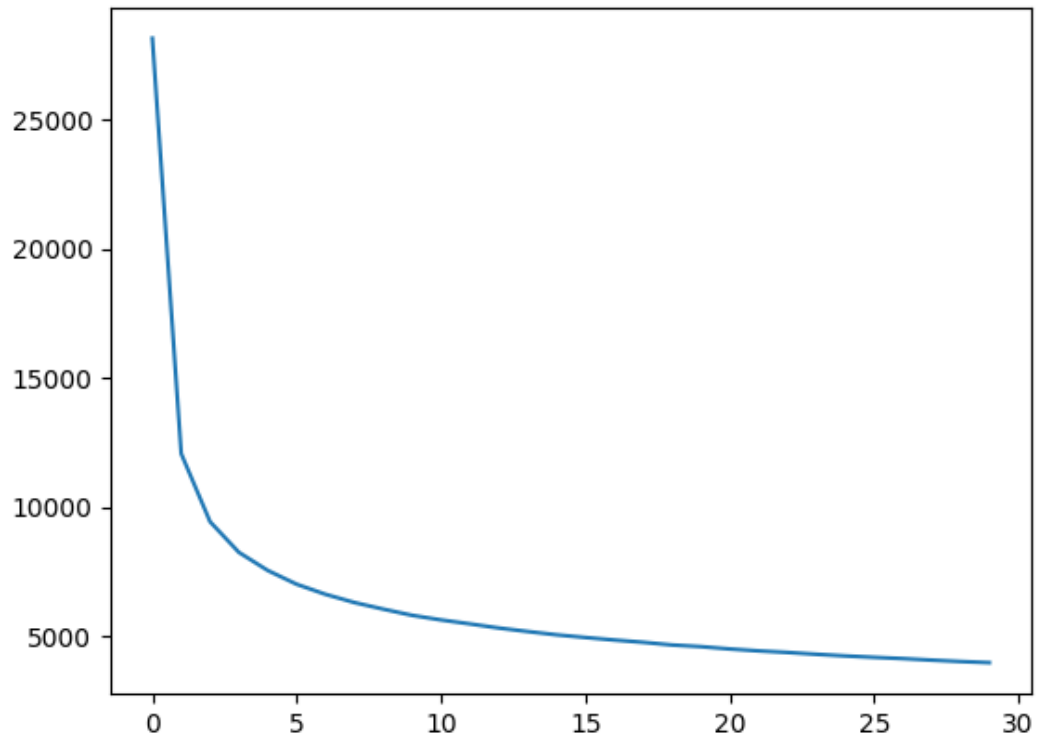
```
Epoch 16: 9402 / 10000 , lr = 1, loss = 4054.3178008389086
Epoch 17: 9396 / 10000 , lr = 1, loss = 3976.352941408937
Epoch 18: 9392 / 10000 , lr = 1, loss = 3900.385731458402
Epoch 19: 9408 / 10000 , lr = 1, loss = 3834.385966816921
Epoch 20: 9404 / 10000 , lr = 1, loss = 3774.6021816869493
Epoch 21: 9398 / 10000 , lr = 1, loss = 3706.510207865997
Epoch 22: 9411 / 10000 , lr = 1, loss = 3644.4970806451706
Epoch 23: 9413 / 10000 , lr = 1, loss = 3583.7206925246583
Epoch 24: 9399 / 10000 , lr = 1, loss = 3550.9442229358133
Epoch 25: 9415 / 10000 , lr = 1, loss = 3482.5264451491607
Epoch 26: 9427 / 10000 , lr = 1, loss = 3436.6748945511963
Epoch 27: 9419 / 10000 , lr = 1, loss = 3392.2822571845077
Epoch 28: 9410 / 10000 , lr = 1, loss = 3350.9052718949642
Epoch 29: 9396 / 10000 , lr = 1, loss = 3309.6543541299434
```



lr=0.5

```
Epoch 0: 8245 / 10000 , lr = 0.5, loss = 28133.165829161822
Epoch 1: 8783 / 10000 , lr = 0.5, loss = 12079.180360100814
Epoch 2: 8967 / 10000 , lr = 0.5, loss = 9448.45566014392
Epoch 3: 9057 / 10000 , lr = 0.5, loss = 8262.767851160772
Epoch 4: 9100 / 10000 , lr = 0.5, loss = 7563.4521238218
Epoch 5: 9155 / 10000 , lr = 0.5, loss = 7031.045592270474
Epoch 6: 9162 / 10000 , lr = 0.5, loss = 6641.859271189254
Epoch 7: 9204 / 10000 , lr = 0.5, loss = 6324.990879343313
Epoch 8: 9226 / 10000 , lr = 0.5, loss = 6069.724929982176
Epoch 9: 9230 / 10000 , lr = 0.5, loss = 5830.2834742730265
Epoch 10: 9254 / 10000 , lr = 0.5, loss = 5652.332873671789
Epoch 11: 9269 / 10000 , lr = 0.5, loss = 5499.54661940375
Epoch 12: 9278 / 10000 , lr = 0.5, loss = 5340.150797779407
Epoch 13: 9285 / 10000 , lr = 0.5, loss = 5205.099931731564
```

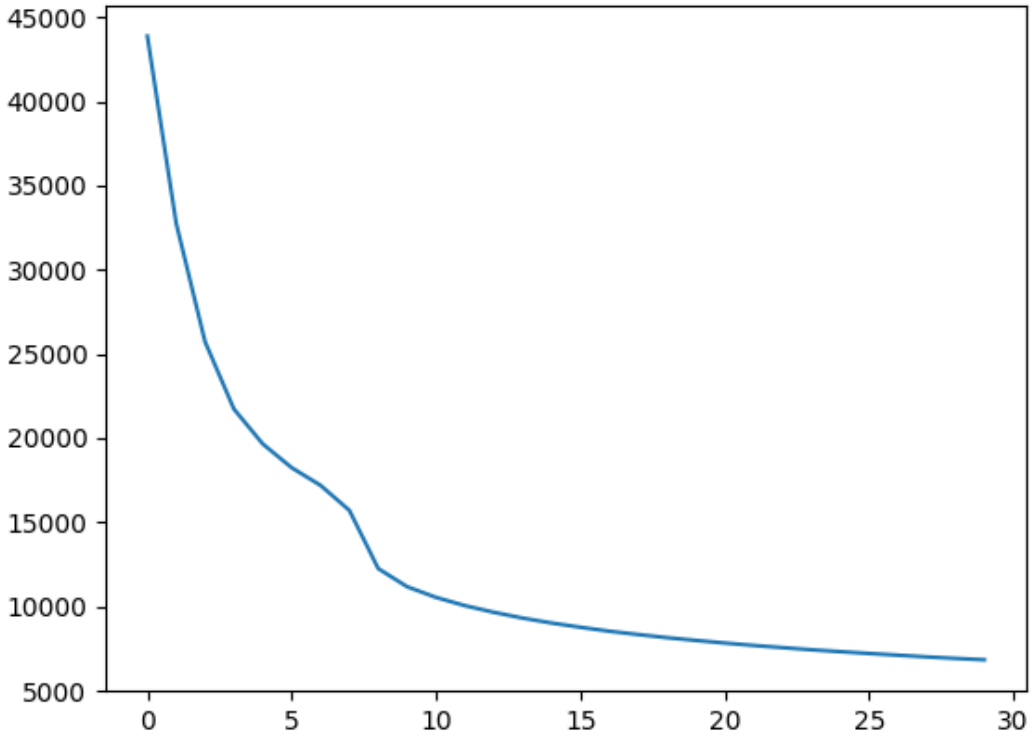
```
Epoch 14: 9305 / 10000 , lr = 0.5, loss = 5077.482629511953
Epoch 15: 9333 / 10000 , lr = 0.5, loss = 4973.493865941528
Epoch 16: 9328 / 10000 , lr = 0.5, loss = 4875.995553576773
Epoch 17: 9328 / 10000 , lr = 0.5, loss = 4788.000924253244
Epoch 18: 9341 / 10000 , lr = 0.5, loss = 4684.701443349562
Epoch 19: 9329 / 10000 , lr = 0.5, loss = 4620.823978759262
Epoch 20: 9336 / 10000 , lr = 0.5, loss = 4530.034218967368
Epoch 21: 9349 / 10000 , lr = 0.5, loss = 4455.954904585768
Epoch 22: 9346 / 10000 , lr = 0.5, loss = 4396.235151226579
Epoch 23: 9343 / 10000 , lr = 0.5, loss = 4324.950410608716
Epoch 24: 9354 / 10000 , lr = 0.5, loss = 4261.412980563262
Epoch 25: 9372 / 10000 , lr = 0.5, loss = 4206.920905987486
Epoch 26: 9367 / 10000 , lr = 0.5, loss = 4158.043161059869
Epoch 27: 9361 / 10000 , lr = 0.5, loss = 4098.987780546952
Epoch 28: 9359 / 10000 , lr = 0.5, loss = 4044.85493916775
Epoch 29: 9372 / 10000 , lr = 0.5, loss = 4001.101289092296
```



lr=0.1

```
Epoch 0: 4177 / 10000 , lr = 0.1, loss = 43883.94601844133
Epoch 1: 5705 / 10000 , lr = 0.1, loss = 32764.040138979162
Epoch 2: 6820 / 10000 , lr = 0.1, loss = 25727.54474423453
Epoch 3: 7190 / 10000 , lr = 0.1, loss = 21729.142135925234
Epoch 4: 7421 / 10000 , lr = 0.1, loss = 19656.091179796545
Epoch 5: 7581 / 10000 , lr = 0.1, loss = 18245.784947611333
Epoch 6: 7679 / 10000 , lr = 0.1, loss = 17196.310352513065
Epoch 7: 8445 / 10000 , lr = 0.1, loss = 15710.606450412926
Epoch 8: 8628 / 10000 , lr = 0.1, loss = 12252.42161096784
Epoch 9: 8702 / 10000 , lr = 0.1, loss = 11174.144401012574
Epoch 10: 8768 / 10000 , lr = 0.1, loss = 10540.8491031145
Epoch 11: 8819 / 10000 , lr = 0.1, loss = 10051.572862891675
```

```
Epoch 12: 8868 / 10000 , lr = 0.1, loss = 9654.808876802173
Epoch 13: 8903 / 10000 , lr = 0.1, loss = 9311.72988474084
Epoch 14: 8938 / 10000 , lr = 0.1, loss = 9019.145415225217
Epoch 15: 8962 / 10000 , lr = 0.1, loss = 8765.876529382438
Epoch 16: 8979 / 10000 , lr = 0.1, loss = 8532.078128288576
Epoch 17: 9005 / 10000 , lr = 0.1, loss = 8335.066902939949
Epoch 18: 9023 / 10000 , lr = 0.1, loss = 8147.655142866198
Epoch 19: 9036 / 10000 , lr = 0.1, loss = 7985.416722146247
Epoch 20: 9047 / 10000 , lr = 0.1, loss = 7829.174110936863
Epoch 21: 9076 / 10000 , lr = 0.1, loss = 7686.0222229274705
Epoch 22: 9076 / 10000 , lr = 0.1, loss = 7556.394031955392
Epoch 23: 9082 / 10000 , lr = 0.1, loss = 7428.758982013682
Epoch 24: 9106 / 10000 , lr = 0.1, loss = 7317.346789587532
Epoch 25: 9114 / 10000 , lr = 0.1, loss = 7209.148748867881
Epoch 26: 9132 / 10000 , lr = 0.1, loss = 7107.7810388963135
Epoch 27: 9131 / 10000 , lr = 0.1, loss = 7005.91604088349
Epoch 28: 9139 / 10000 , lr = 0.1, loss = 6915.585961337499
Epoch 29: 9145 / 10000 , lr = 0.1, loss = 6832.588091923184
```



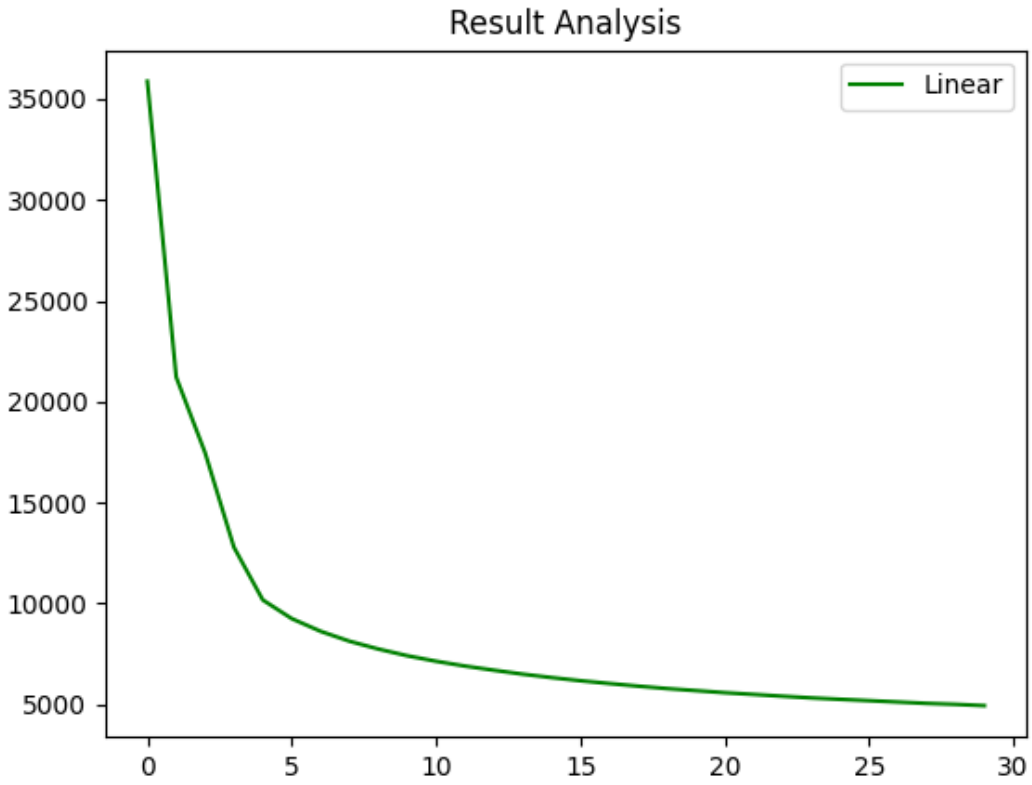
总结：对于固定学习率，较大的学习率收敛速度较快，但是到一定程度会发生Loss震荡，难以找到全局极小点；较小的学习率虽然能够避免越过极小点，但是收敛速度过慢，总之需要进行大量的验证和实验，才能找到较合适的学习率。

可变学习率

Linear

$$\alpha^{(\tau)} = k\alpha^{(\tau-1)}$$

```
Epoch 0: 6559 / 10000 , loss = 35882.503722095564
Epoch 1: 7442 / 10000 , loss = 21222.176965847302
Epoch 2: 7739 / 10000 , loss = 17478.383642499557
Epoch 3: 8716 / 10000 , loss = 12790.051812629024
Epoch 4: 8855 / 10000 , loss = 10178.37225980348
Epoch 5: 8938 / 10000 , loss = 9255.576396604083
Epoch 6: 8975 / 10000 , loss = 8625.936088329665
Epoch 7: 9013 / 10000 , loss = 8133.915571561642
Epoch 8: 9052 / 10000 , loss = 7747.441068991741
Epoch 9: 9083 / 10000 , loss = 7415.397324737706
Epoch 10: 9113 / 10000 , loss = 7140.913146137555
Epoch 11: 9131 / 10000 , loss = 6902.98848665373
Epoch 12: 9149 / 10000 , loss = 6697.2500439224605
Epoch 13: 9171 / 10000 , loss = 6502.219574195122
Epoch 14: 9194 / 10000 , loss = 6332.2343523643785
Epoch 15: 9205 / 10000 , loss = 6175.558965650472
Epoch 16: 9210 / 10000 , loss = 6038.113388535207
Epoch 17: 9223 / 10000 , loss = 5907.742131779419
Epoch 18: 9224 / 10000 , loss = 5787.156547399345
Epoch 19: 9233 / 10000 , loss = 5685.828388608208
Epoch 20: 9241 / 10000 , loss = 5582.4730105460485
Epoch 21: 9254 / 10000 , loss = 5496.944982119867
Epoch 22: 9264 / 10000 , loss = 5407.69190756148
Epoch 23: 9259 / 10000 , loss = 5324.214865426825
Epoch 24: 9269 / 10000 , loss = 5254.332170770473
Epoch 25: 9275 / 10000 , loss = 5188.586510926823
Epoch 26: 9289 / 10000 , loss = 5120.781316451178
Epoch 27: 9292 / 10000 , loss = 5053.096444277559
Epoch 28: 9297 / 10000 , loss = 5004.914503152329
Epoch 29: 9296 / 10000 , loss = 4940.218905383333
```

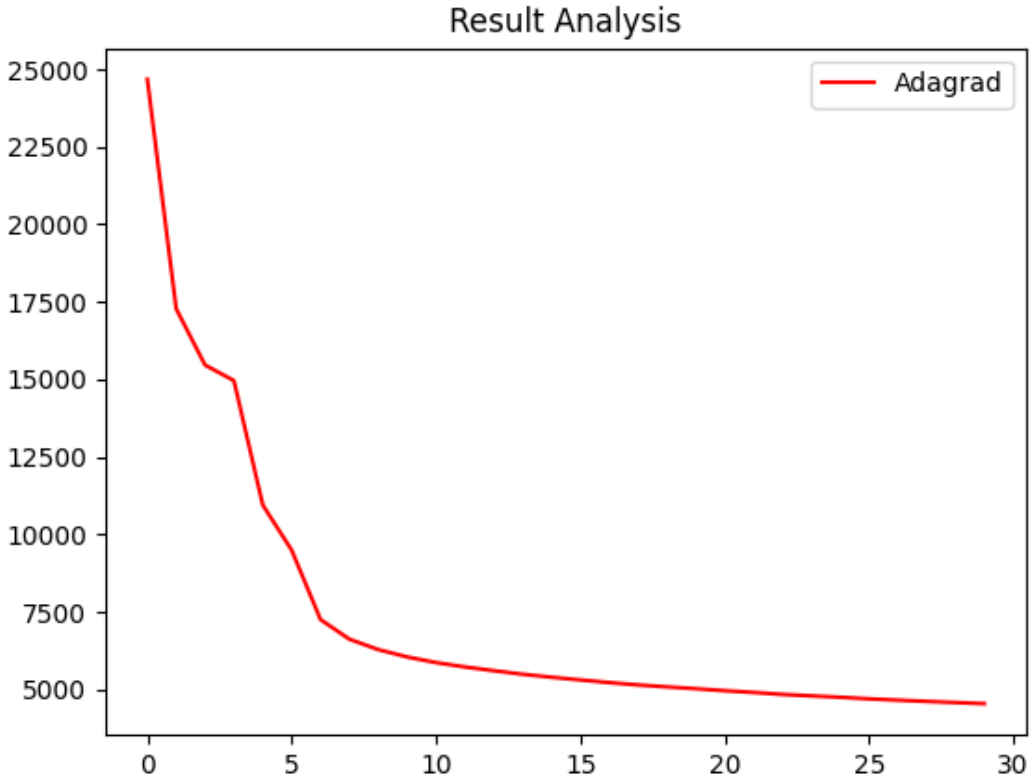


Adagrad

$$\alpha^{(\tau)} = \frac{1}{\sqrt{\sum_{t=1}^{\tau} \nabla^2 L(w^{(t)}, b^{(t)}) + \epsilon}} \alpha^{(0)}$$

```
Epoch 0: 6581 / 10000 , loss = 24678.6027006375
Epoch 1: 7450 / 10000 , loss = 17275.0203631023
Epoch 2: 7532 / 10000 , loss = 15462.012855750516
Epoch 3: 7804 / 10000 , loss = 14958.99145547533
Epoch 4: 8469 / 10000 , loss = 10947.347396154282
Epoch 5: 9078 / 10000 , loss = 9493.75922771869
Epoch 6: 9131 / 10000 , loss = 7254.025768755406
Epoch 7: 9187 / 10000 , loss = 6622.036722201418
Epoch 8: 9213 / 10000 , loss = 6285.6218045275955
Epoch 9: 9219 / 10000 , loss = 6048.79480996238
Epoch 10: 9242 / 10000 , loss = 5866.93046851529
Epoch 11: 9243 / 10000 , loss = 5726.572579446943
Epoch 12: 9268 / 10000 , loss = 5605.273367380281
Epoch 13: 9253 / 10000 , loss = 5491.383253722598
Epoch 14: 9291 / 10000 , loss = 5394.161780871379
Epoch 15: 9289 / 10000 , loss = 5307.002142625327
Epoch 16: 9300 / 10000 , loss = 5221.318009680166
Epoch 17: 9299 / 10000 , loss = 5145.573073506758
Epoch 18: 9290 / 10000 , loss = 5082.580036175778
Epoch 19: 9287 / 10000 , loss = 5024.368833807431
Epoch 20: 9303 / 10000 , loss = 4959.954203881221
Epoch 21: 9301 / 10000 , loss = 4904.608777437189
Epoch 22: 9307 / 10000 , loss = 4839.476369854631
Epoch 23: 9309 / 10000 , loss = 4793.56093555187
Epoch 24: 9305 / 10000 , loss = 4748.515954922164
```

```
Epoch 25: 9315 / 10000 , loss = 4697.779343577753
Epoch 26: 9318 / 10000 , loss = 4655.54250957998
Epoch 27: 9318 / 10000 , loss = 4615.123630980914
Epoch 28: 9321 / 10000 , loss = 4577.794052140049
Epoch 29: 9324 / 10000 , loss = 4542.216551765991
```

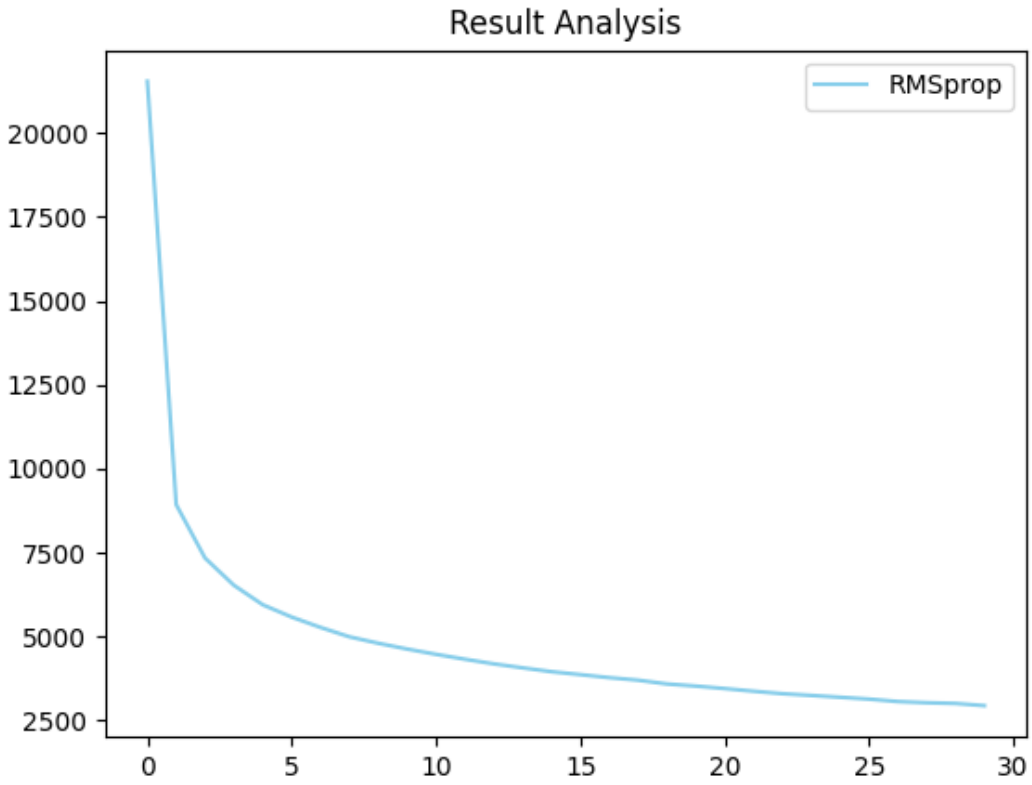


RMSprop

$$g^{(\tau)} = \gamma g^{(\tau-1)} + (1-\gamma)\nabla L^2(w^{(\tau)}, b^{(\tau)})$$

$$\alpha^{(\tau)} = \frac{1}{\sqrt{g^{(\tau)} + \epsilon}}\alpha^{(0)}$$

```
...Epoch 29: 9446 / 10000 , loss = 2937.063524722071
```



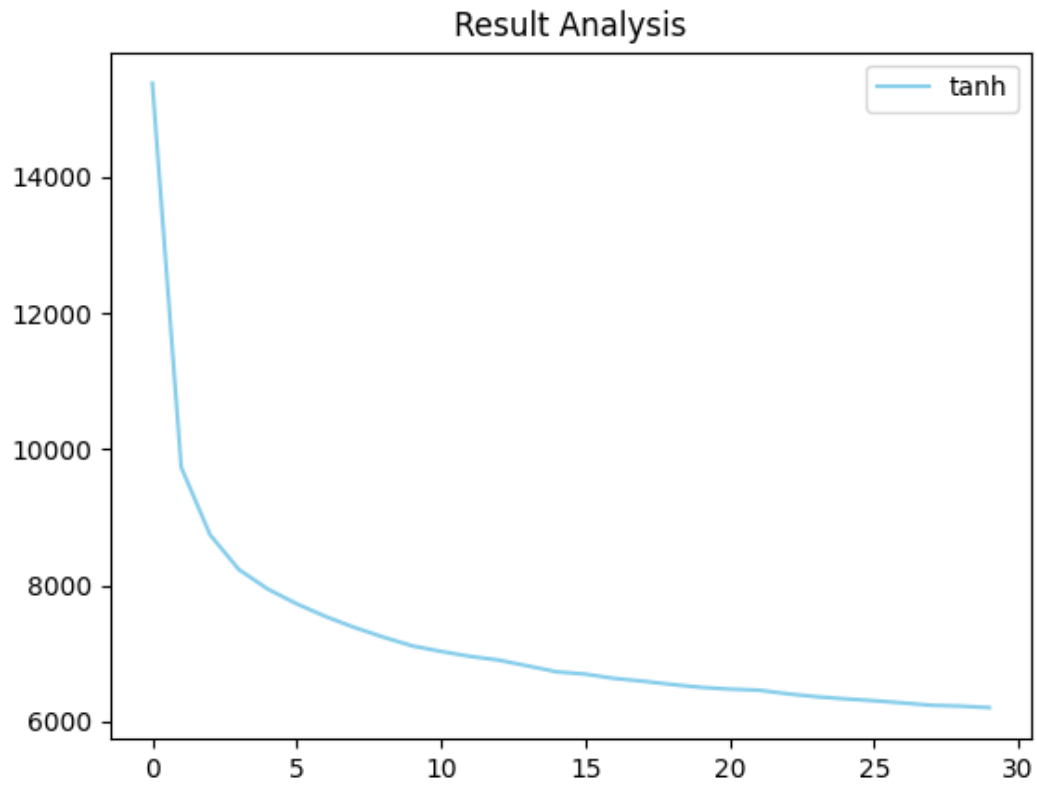
总结：从上一个实验可以看出，一个固定的学习率是很难先验的得知，需要一些经验，故需要使用各种可变动的学习率技巧。其中线性衰减学习率希望在一开始的时候加速收敛，在后面需要寻找极小点的时候进行惊喜查找。Adagrad思路类似，希望积累以往的梯度大小信息，对于稀疏的系数快速收敛，稠密的系数减慢搜索步长。RMSprop在此基础上削弱以往的梯度对于搜索步长的影响，中和当前信息和以往信息，减少震荡，加快收敛。

2.3 使用不同的激活函数

使用$tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

```
Epoch 0: 9139 / 10000 , lr = 0.1, loss = 15382.135652736562
Epoch 1: 9256 / 10000 , lr = 0.1, loss = 9735.850446045422
Epoch 2: 9239 / 10000 , lr = 0.1, loss = 8742.57723356086
Epoch 3: 9280 / 10000 , lr = 0.1, loss = 8231.169795315784
Epoch 4: 9304 / 10000 , lr = 0.1, loss = 7944.4836627037585
Epoch 5: 9285 / 10000 , lr = 0.1, loss = 7727.56119883
Epoch 6: 9253 / 10000 , lr = 0.1, loss = 7541.504343007939
Epoch 7: 9289 / 10000 , lr = 0.1, loss = 7379.884468484366
Epoch 8: 9314 / 10000 , lr = 0.1, loss = 7238.32061869674
Epoch 9: 9349 / 10000 , lr = 0.1, loss = 7109.534022983143
Epoch 10: 9263 / 10000 , lr = 0.1, loss = 7028.033697551473
Epoch 11: 9262 / 10000 , lr = 0.1, loss = 6954.933141932033
Epoch 12: 9302 / 10000 , lr = 0.1, loss = 6900.1610255415835
Epoch 13: 9286 / 10000 , lr = 0.1, loss = 6814.997830098305
Epoch 14: 9300 / 10000 , lr = 0.1, loss = 6728.124291223308
Epoch 15: 9349 / 10000 , lr = 0.1, loss = 6694.17289855181
Epoch 16: 9363 / 10000 , lr = 0.1, loss = 6630.80248447726
Epoch 17: 9379 / 10000 , lr = 0.1, loss = 6589.366584924665
Epoch 18: 9331 / 10000 , lr = 0.1, loss = 6541.459244887242
Epoch 19: 9329 / 10000 , lr = 0.1, loss = 6499.482889014795
```

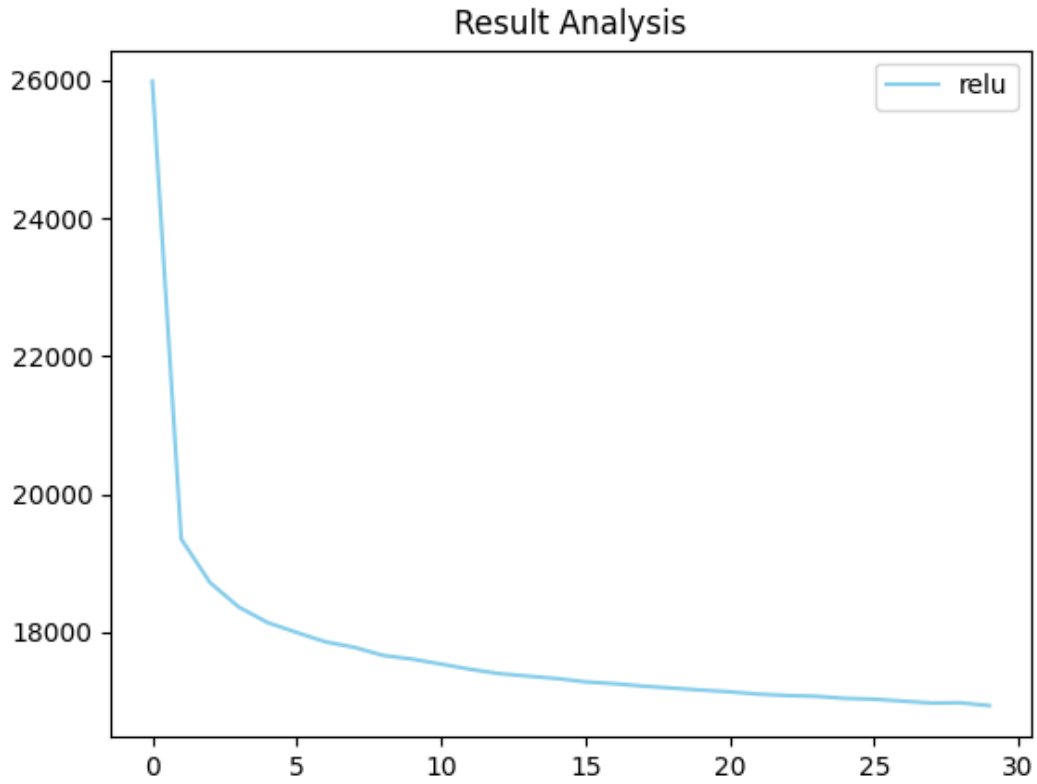
```
Epoch 20: 9362 / 10000 , lr = 0.1, loss = 6473.771437776647
Epoch 21: 9365 / 10000 , lr = 0.1, loss = 6457.715899614046
Epoch 22: 9292 / 10000 , lr = 0.1, loss = 6405.049690794703
Epoch 23: 9366 / 10000 , lr = 0.1, loss = 6361.598657145688
Epoch 24: 9360 / 10000 , lr = 0.1, loss = 6330.0422714583465
Epoch 25: 9375 / 10000 , lr = 0.1, loss = 6303.088523476691
Epoch 26: 9352 / 10000 , lr = 0.1, loss = 6270.234597674914
Epoch 27: 9337 / 10000 , lr = 0.1, loss = 6235.569775661904
Epoch 28: 9318 / 10000 , lr = 0.1, loss = 6223.317103086967
Epoch 29: 9343 / 10000 , lr = 0.1, loss = 6200.595036702501
```



使用$relu = max(x, 0)$

```
Epoch 0: 6673 / 10000 , lr = 0.1, loss = 25985.816445850603
Epoch 1: 6747 / 10000 , lr = 0.1, loss = 19348.11511535276
Epoch 2: 6739 / 10000 , lr = 0.1, loss = 18714.208641737314
Epoch 3: 6780 / 10000 , lr = 0.1, loss = 18359.03770142299
Epoch 4: 6770 / 10000 , lr = 0.1, loss = 18134.906313988955
Epoch 5: 6813 / 10000 , lr = 0.1, loss = 17992.22268757448
Epoch 6: 6803 / 10000 , lr = 0.1, loss = 17856.125493042113
Epoch 7: 6806 / 10000 , lr = 0.1, loss = 17776.53534084255
Epoch 8: 6820 / 10000 , lr = 0.1, loss = 17658.726675814014
Epoch 9: 6822 / 10000 , lr = 0.1, loss = 17606.12906187218
Epoch 10: 6814 / 10000 , lr = 0.1, loss = 17533.901022652914
Epoch 11: 6821 / 10000 , lr = 0.1, loss = 17459.203372825916
Epoch 12: 6826 / 10000 , lr = 0.1, loss = 17396.842873935966
Epoch 13: 6818 / 10000 , lr = 0.1, loss = 17359.5145235123
Epoch 14: 6820 / 10000 , lr = 0.1, loss = 17324.77725349647
Epoch 15: 6827 / 10000 , lr = 0.1, loss = 17276.00154529964
Epoch 16: 6829 / 10000 , lr = 0.1, loss = 17249.72648643931
Epoch 17: 6820 / 10000 , lr = 0.1, loss = 17214.180937818404
```

```
Epoch 18: 6824 / 10000 , lr = 0.1, loss = 17185.943808450473
Epoch 19: 6821 / 10000 , lr = 0.1, loss = 17157.025126176584
Epoch 20: 6820 / 10000 , lr = 0.1, loss = 17131.8055970417
Epoch 21: 6827 / 10000 , lr = 0.1, loss = 17097.429535501527
Epoch 22: 6824 / 10000 , lr = 0.1, loss = 17078.312248357735
Epoch 23: 6821 / 10000 , lr = 0.1, loss = 17066.80665658254
Epoch 24: 6832 / 10000 , lr = 0.1, loss = 17036.287426711617
Epoch 25: 6826 / 10000 , lr = 0.1, loss = 17024.444217726515
Epoch 26: 6819 / 10000 , lr = 0.1, loss = 16996.823608198556
Epoch 27: 6827 / 10000 , lr = 0.1, loss = 16968.62798065416
Epoch 28: 6829 / 10000 , lr = 0.1, loss = 16972.231603481294
Epoch 29: 6831 / 10000 , lr = 0.1, loss = 16930.812467658463
```



总结：使用不同的激活函数，几乎需要完全重新探索学习率以及参数初始化的方法。实验过程中，RELU函数在使用的时候对于初始值极度敏感，稍有不慎则无法更新参数。

2.4 数据总结

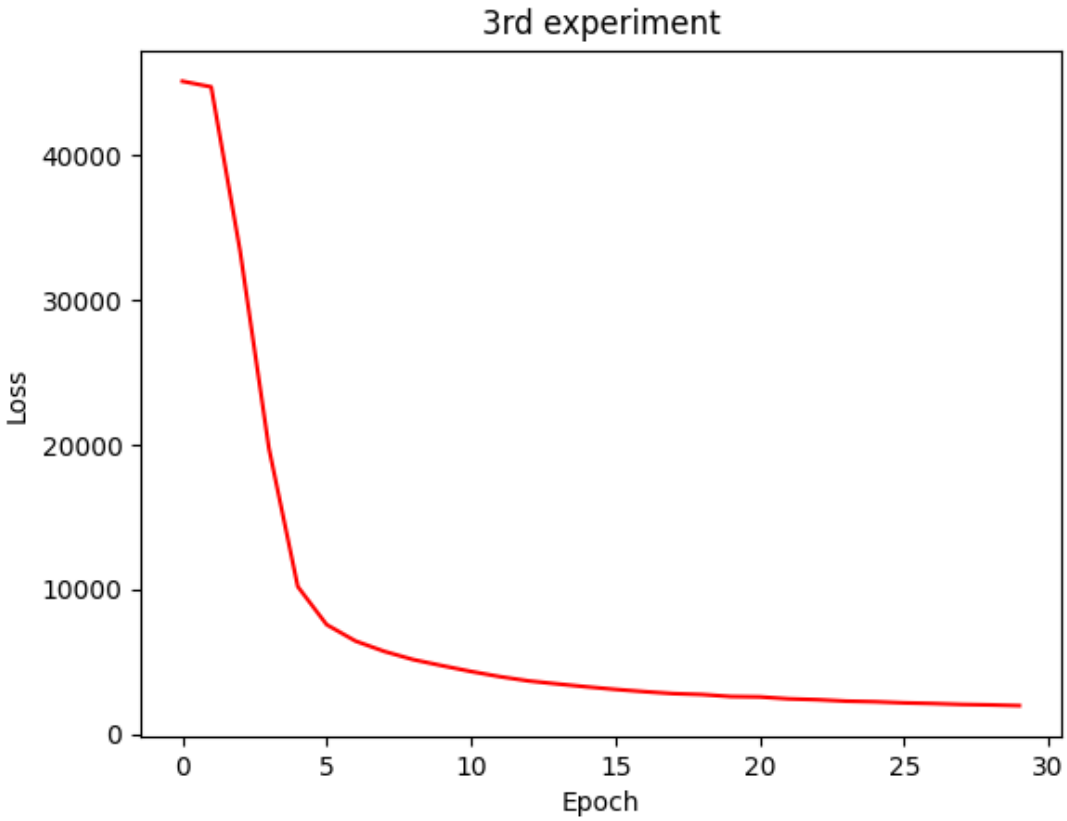
均迭代30轮次	Loss	Accurate Rate(%)
基线	4747	93.55
Xavier权值初始化	2482	96.24
learning rate=20	7279	91.80
lr=5	2871	95.01
lr=1	3309	93.96
lr=0.5	4001	93.72

均迭代30轮次	Loss	Accurate Rate(%)
lr=0.1	4940	92.96
线性衰减学习率	4940	93.92
Adagrad	4542	93.24
RMSprop	2934	94.46
tanh	6200	93.43
relu	16930	68.31

实验三：实现95%正确率的六层以上的多层感知机

网络结构为[784, 30, 30, 30, 30, 10]，激活函数为sigmoid，采用Xavier初始化，小样本批处理梯度下降。见`实验三95%正确率.py`

```
Epoch 0: 1032 / 10000 , lr = 0.3, loss = 45113.79812587879
Epoch 1: 2117 / 10000 , lr = 0.3, loss = 44725.325179226595
Epoch 2: 6355 / 10000 , lr = 0.3, loss = 33407.014118336396
Epoch 3: 8554 / 10000 , lr = 0.3, loss = 19762.1773539181
Epoch 4: 8967 / 10000 , lr = 0.3, loss = 10205.884008161924
Epoch 5: 9127 / 10000 , lr = 0.3, loss = 7576.118984945956
Epoch 6: 9254 / 10000 , lr = 0.3, loss = 6444.247755448362
Epoch 7: 9294 / 10000 , lr = 0.3, loss = 5727.757088393604
Epoch 8: 9306 / 10000 , lr = 0.3, loss = 5161.437578053755
Epoch 9: 9329 / 10000 , lr = 0.3, loss = 4742.126384150059
Epoch 10: 9405 / 10000 , lr = 0.3, loss = 4342.969864225763
Epoch 11: 9447 / 10000 , lr = 0.3, loss = 3985.3359443627714
Epoch 12: 9471 / 10000 , lr = 0.3, loss = 3688.0478430036837
Epoch 13: 9483 / 10000 , lr = 0.3, loss = 3480.621005009056
Epoch 14: 9469 / 10000 , lr = 0.3, loss = 3283.065232412878
Epoch 15: 9472 / 10000 , lr = 0.3, loss = 3104.2549205399264
Epoch 16: 9500 / 10000 , lr = 0.3, loss = 2949.4389254202233
Epoch 17: 9520 / 10000 , lr = 0.3, loss = 2814.30227128724
Epoch 18: 9516 / 10000 , lr = 0.3, loss = 2750.029817788935
Epoch 19: 9519 / 10000 , lr = 0.3, loss = 2611.0638576895117
Epoch 20: 9532 / 10000 , lr = 0.3, loss = 2585.459925366044
Epoch 21: 9500 / 10000 , lr = 0.3, loss = 2452.3164137453746
Epoch 22: 9541 / 10000 , lr = 0.3, loss = 2393.1762664507633
Epoch 23: 9520 / 10000 , lr = 0.3, loss = 2293.2667088473177
Epoch 24: 9535 / 10000 , lr = 0.3, loss = 2250.8668046927355
Epoch 25: 9528 / 10000 , lr = 0.3, loss = 2173.593433815489
Epoch 26: 9554 / 10000 , lr = 0.3, loss = 2129.8664606918383
Epoch 27: 9538 / 10000 , lr = 0.3, loss = 2064.873726536674
Epoch 28: 9529 / 10000 , lr = 0.3, loss = 2031.4417050510904
Epoch 29: 9564 / 10000 , lr = 0.3, loss = 1981.3869640424164
```


3rd experiment
40000
30000
20000
10000
0
Loss
0
5
10
15
20
25
30
Epoch