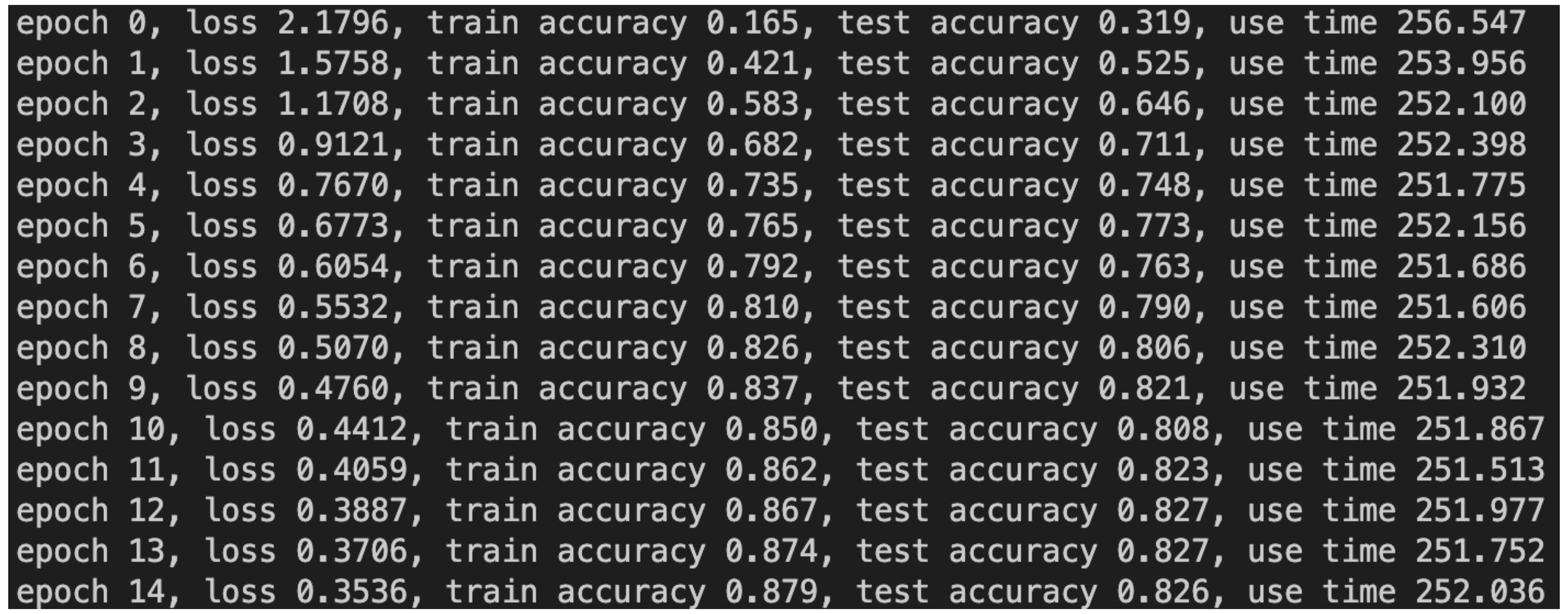
组会汇报

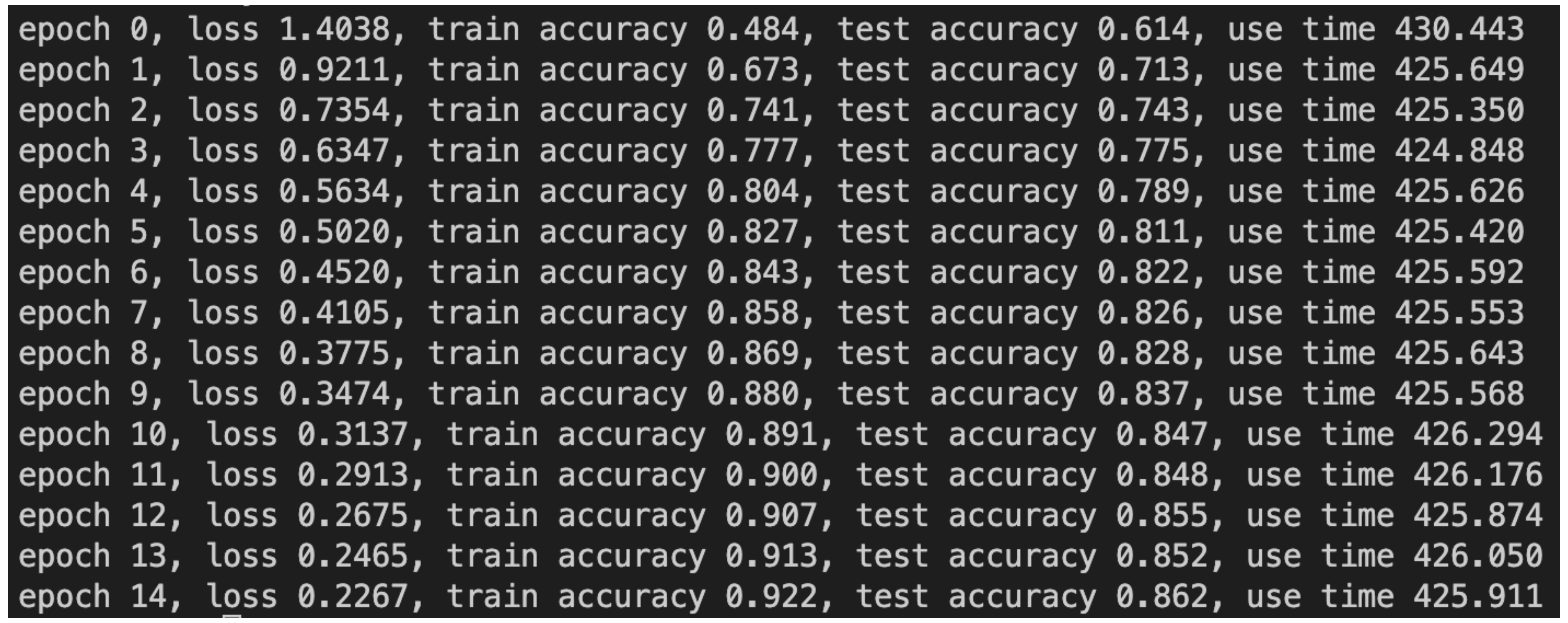
2.11

实验



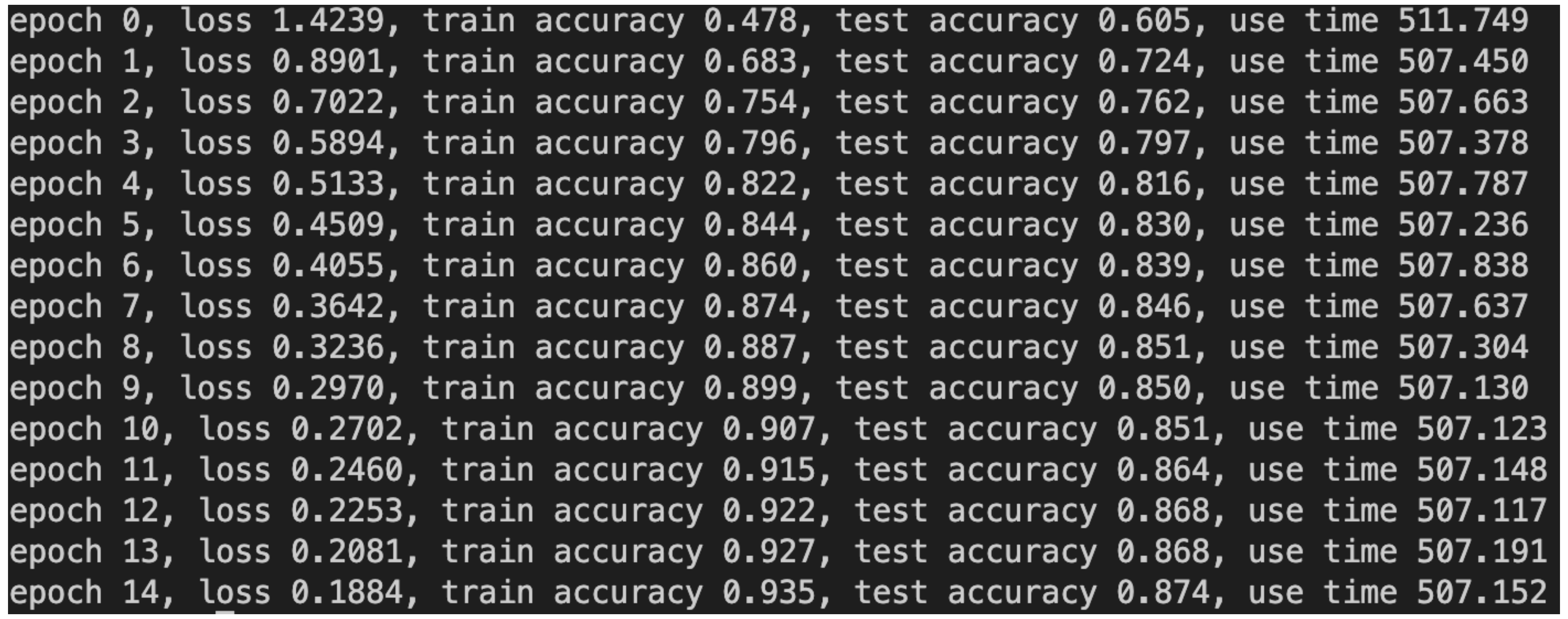
```
epoch 0, loss 2.1796, train accuracy 0.165, test accuracy 0.319, use time 256.547
epoch 1, loss 1.5758, train accuracy 0.421, test accuracy 0.525, use time 253.956
epoch 2, loss 1.1708, train accuracy 0.583, test accuracy 0.646, use time 252.100
epoch 3, loss 0.9121, train accuracy 0.682, test accuracy 0.711, use time 252.398
epoch 4, loss 0.7670, train accuracy 0.735, test accuracy 0.748, use time 251.775
epoch 5, loss 0.6773, train accuracy 0.765, test accuracy 0.773, use time 252.156
epoch 6, loss 0.6054, train accuracy 0.792, test accuracy 0.763, use time 251.686
epoch 7, loss 0.5532, train accuracy 0.810, test accuracy 0.790, use time 251.606
epoch 8, loss 0.5070, train accuracy 0.826, test accuracy 0.806, use time 252.310
epoch 9, loss 0.4760, train accuracy 0.837, test accuracy 0.821, use time 251.932
epoch 10, loss 0.4412, train accuracy 0.850, test accuracy 0.808, use time 251.867
epoch 11, loss 0.4059, train accuracy 0.862, test accuracy 0.823, use time 251.513
epoch 12, loss 0.3887, train accuracy 0.867, test accuracy 0.827, use time 251.977
epoch 13, loss 0.3706, train accuracy 0.874, test accuracy 0.827, use time 251.752
epoch 14, loss 0.3536, train accuracy 0.879, test accuracy 0.826, use time 252.036
```

Vgg11



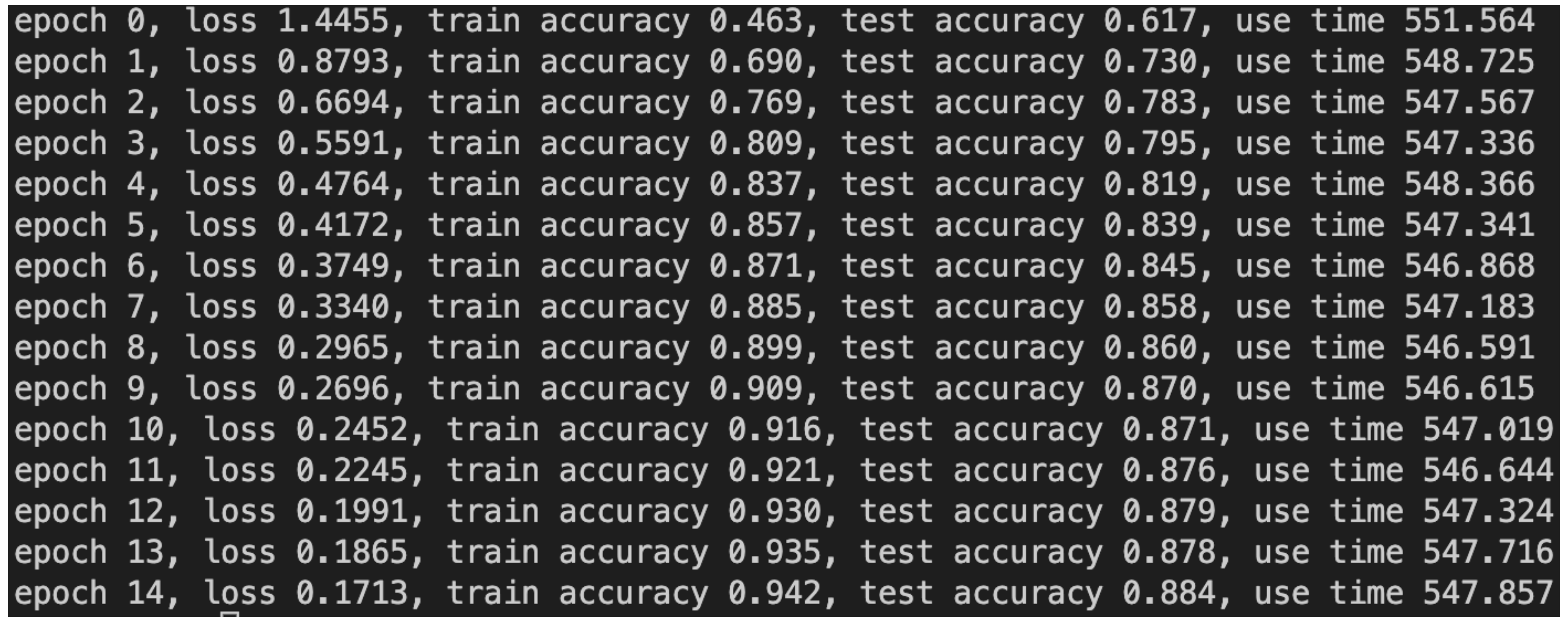
```
epoch 0, loss 1.4038, train accuracy 0.484, test accuracy 0.614, use time 430.443
epoch 1, loss 0.9211, train accuracy 0.673, test accuracy 0.713, use time 425.649
epoch 2, loss 0.7354, train accuracy 0.741, test accuracy 0.743, use time 425.350
epoch 3, loss 0.6347, train accuracy 0.777, test accuracy 0.775, use time 424.848
epoch 4, loss 0.5634, train accuracy 0.804, test accuracy 0.789, use time 425.626
epoch 5, loss 0.5020, train accuracy 0.827, test accuracy 0.811, use time 425.420
epoch 6, loss 0.4520, train accuracy 0.843, test accuracy 0.822, use time 425.592
epoch 7, loss 0.4105, train accuracy 0.858, test accuracy 0.826, use time 425.553
epoch 8, loss 0.3775, train accuracy 0.869, test accuracy 0.828, use time 425.643
epoch 9, loss 0.3474, train accuracy 0.880, test accuracy 0.837, use time 425.568
epoch 10, loss 0.3137, train accuracy 0.891, test accuracy 0.847, use time 426.294
epoch 11, loss 0.2913, train accuracy 0.900, test accuracy 0.848, use time 426.176
epoch 12, loss 0.2675, train accuracy 0.907, test accuracy 0.855, use time 425.874
epoch 13, loss 0.2465, train accuracy 0.913, test accuracy 0.852, use time 426.050
epoch 14, loss 0.2267, train accuracy 0.922, test accuracy 0.862, use time 425.911
```

Vgg13



```
epoch 0, loss 1.4239, train accuracy 0.478, test accuracy 0.605, use time 511.749
epoch 1, loss 0.8901, train accuracy 0.683, test accuracy 0.724, use time 507.450
epoch 2, loss 0.7022, train accuracy 0.754, test accuracy 0.762, use time 507.663
epoch 3, loss 0.5894, train accuracy 0.796, test accuracy 0.797, use time 507.378
epoch 4, loss 0.5133, train accuracy 0.822, test accuracy 0.816, use time 507.787
epoch 5, loss 0.4509, train accuracy 0.844, test accuracy 0.830, use time 507.236
epoch 6, loss 0.4055, train accuracy 0.860, test accuracy 0.839, use time 507.838
epoch 7, loss 0.3642, train accuracy 0.874, test accuracy 0.846, use time 507.637
epoch 8, loss 0.3236, train accuracy 0.887, test accuracy 0.851, use time 507.304
epoch 9, loss 0.2970, train accuracy 0.899, test accuracy 0.850, use time 507.130
epoch 10, loss 0.2702, train accuracy 0.907, test accuracy 0.851, use time 507.123
epoch 11, loss 0.2460, train accuracy 0.915, test accuracy 0.864, use time 507.148
epoch 12, loss 0.2253, train accuracy 0.922, test accuracy 0.868, use time 507.117
epoch 13, loss 0.2081, train accuracy 0.927, test accuracy 0.868, use time 507.191
epoch 14, loss 0.1884, train accuracy 0.935, test accuracy 0.874, use time 507.152
```

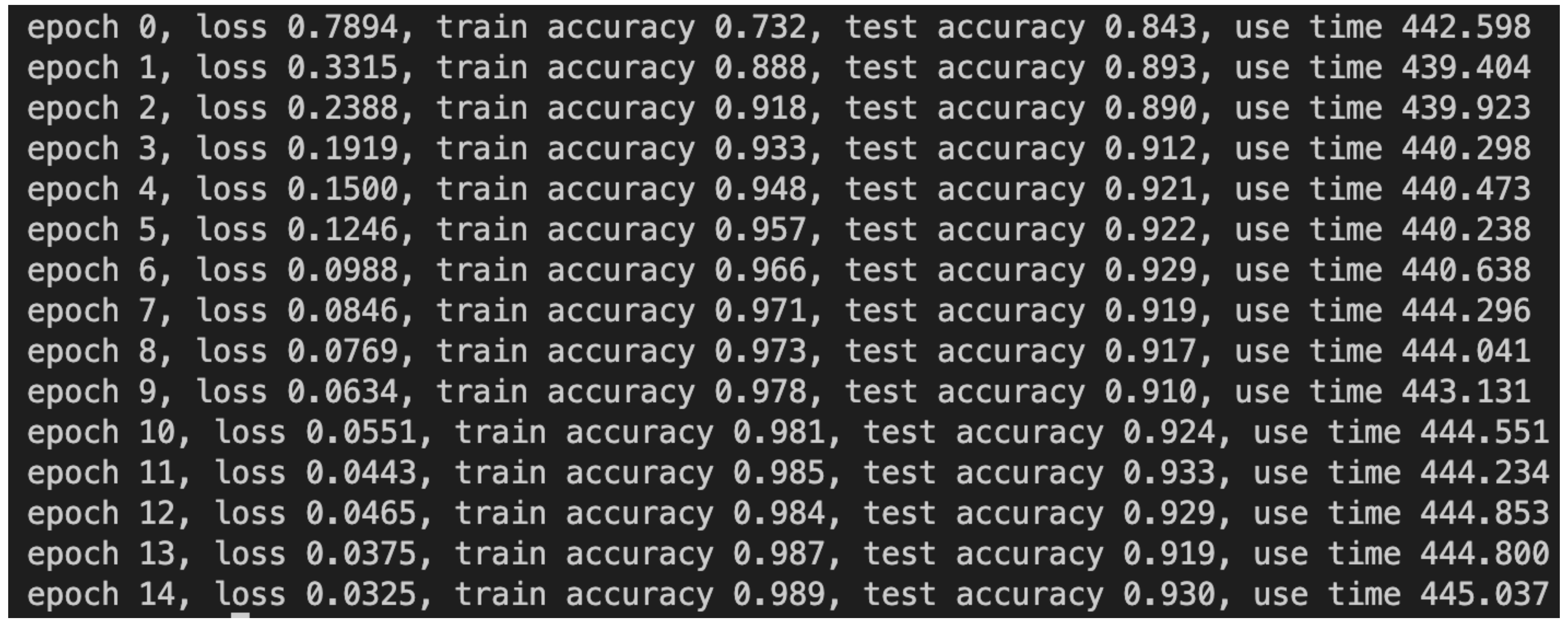
Vgg16



```
epoch 0, loss 1.4455, train accuracy 0.463, test accuracy 0.617, use time 551.564
epoch 1, loss 0.8793, train accuracy 0.690, test accuracy 0.730, use time 548.725
epoch 2, loss 0.6694, train accuracy 0.769, test accuracy 0.783, use time 547.567
epoch 3, loss 0.5591, train accuracy 0.809, test accuracy 0.795, use time 547.336
epoch 4, loss 0.4764, train accuracy 0.837, test accuracy 0.819, use time 548.366
epoch 5, loss 0.4172, train accuracy 0.857, test accuracy 0.839, use time 547.341
epoch 6, loss 0.3749, train accuracy 0.871, test accuracy 0.845, use time 546.868
epoch 7, loss 0.3340, train accuracy 0.885, test accuracy 0.858, use time 547.183
epoch 8, loss 0.2965, train accuracy 0.899, test accuracy 0.860, use time 546.591
epoch 9, loss 0.2696, train accuracy 0.909, test accuracy 0.870, use time 546.615
epoch 10, loss 0.2452, train accuracy 0.916, test accuracy 0.871, use time 547.019
epoch 11, loss 0.2245, train accuracy 0.921, test accuracy 0.876, use time 546.644
epoch 12, loss 0.1991, train accuracy 0.930, test accuracy 0.879, use time 547.324
epoch 13, loss 0.1865, train accuracy 0.935, test accuracy 0.878, use time 547.716
epoch 14, loss 0.1713, train accuracy 0.942, test accuracy 0.884, use time 547.857
```

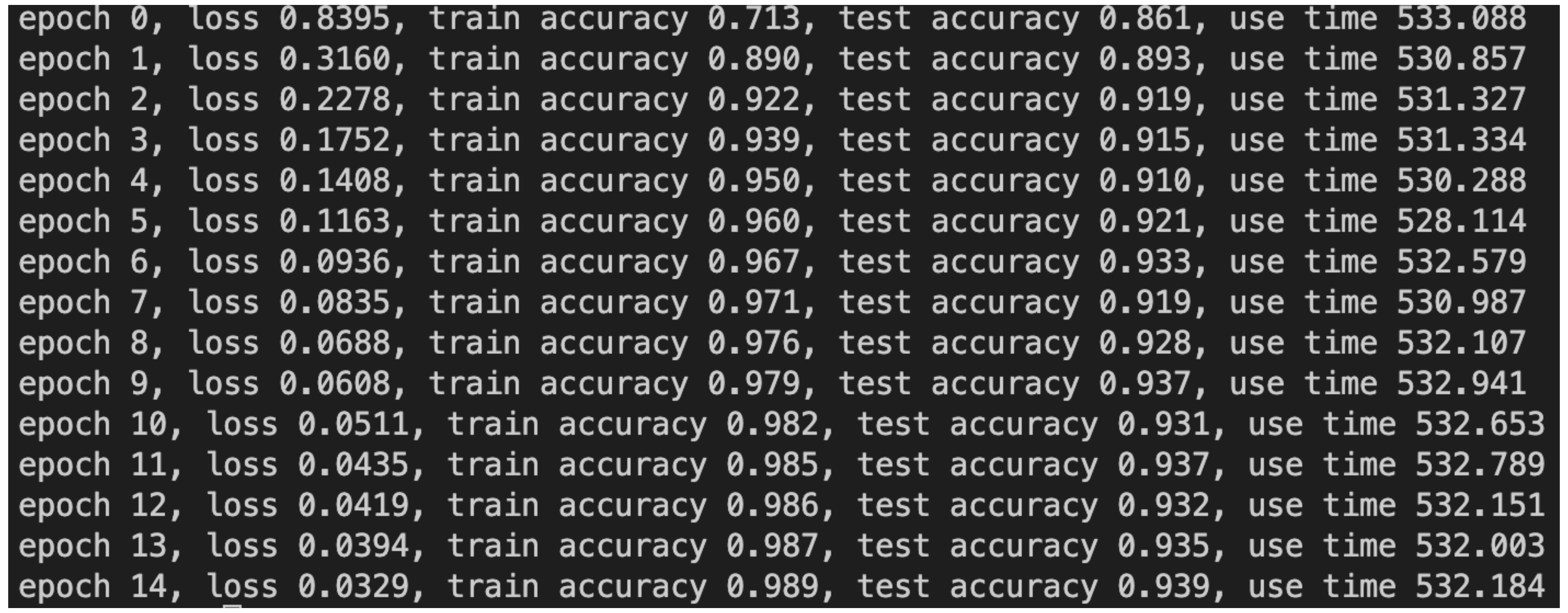
Vgg19

预训练



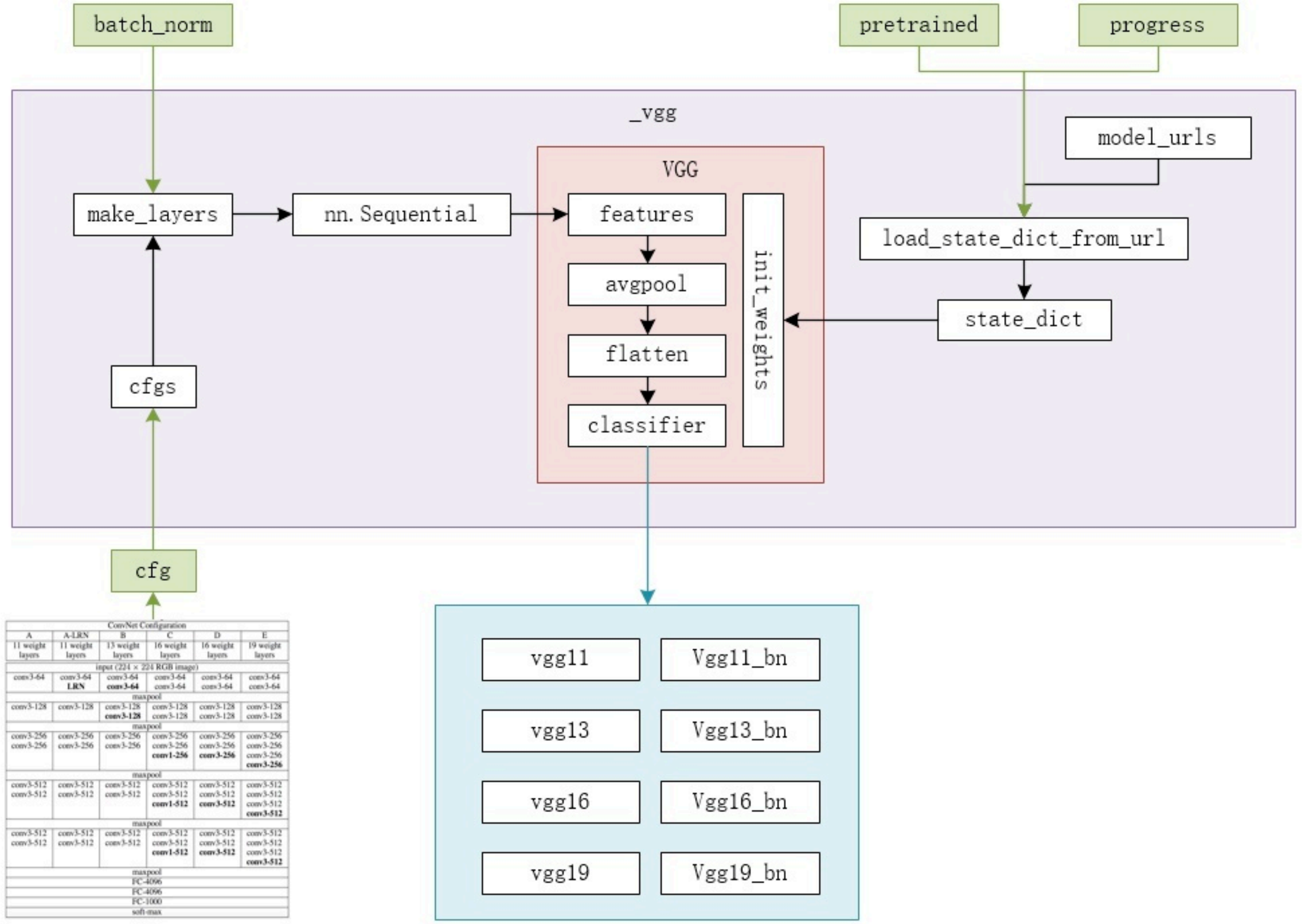
```
epoch 0, loss 0.7894, train accuracy 0.732, test accuracy 0.843, use time 442.598
epoch 1, loss 0.3315, train accuracy 0.888, test accuracy 0.893, use time 439.404
epoch 2, loss 0.2388, train accuracy 0.918, test accuracy 0.890, use time 439.923
epoch 3, loss 0.1919, train accuracy 0.933, test accuracy 0.912, use time 440.298
epoch 4, loss 0.1500, train accuracy 0.948, test accuracy 0.921, use time 440.473
epoch 5, loss 0.1246, train accuracy 0.957, test accuracy 0.922, use time 440.238
epoch 6, loss 0.0988, train accuracy 0.966, test accuracy 0.929, use time 440.638
epoch 7, loss 0.0846, train accuracy 0.971, test accuracy 0.919, use time 444.296
epoch 8, loss 0.0769, train accuracy 0.973, test accuracy 0.917, use time 444.041
epoch 9, loss 0.0634, train accuracy 0.978, test accuracy 0.910, use time 443.131
epoch 10, loss 0.0551, train accuracy 0.981, test accuracy 0.924, use time 444.551
epoch 11, loss 0.0443, train accuracy 0.985, test accuracy 0.933, use time 444.234
epoch 12, loss 0.0465, train accuracy 0.984, test accuracy 0.929, use time 444.853
epoch 13, loss 0.0375, train accuracy 0.987, test accuracy 0.919, use time 444.800
epoch 14, loss 0.0325, train accuracy 0.989, test accuracy 0.930, use time 445.037
```

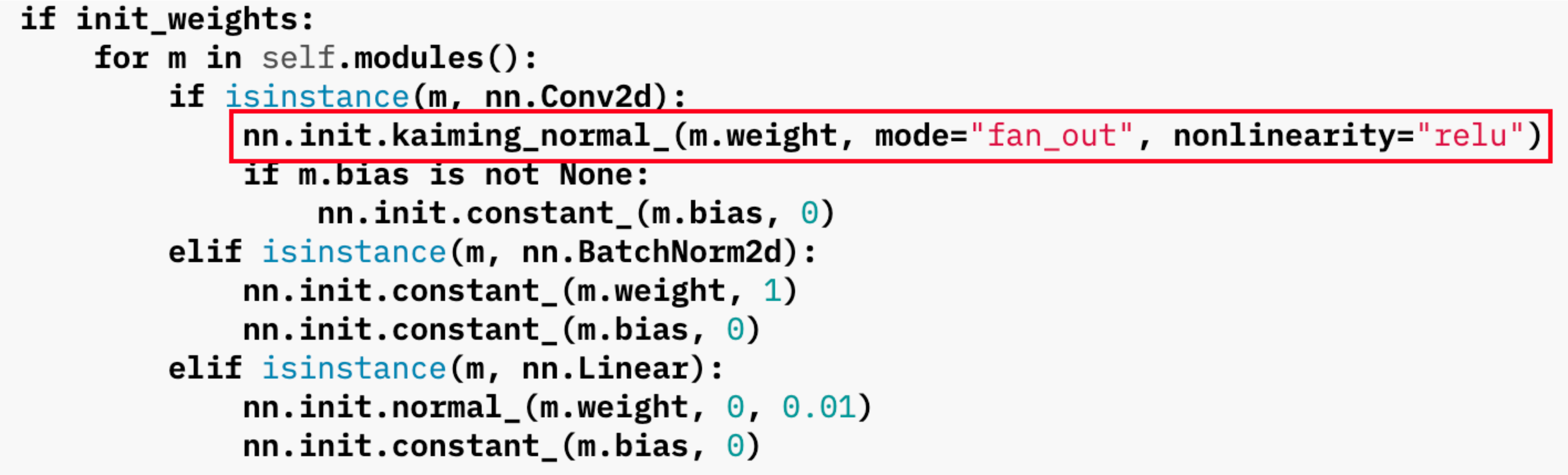
Vgg16



```
epoch 0, loss 0.8395, train accuracy 0.713, test accuracy 0.861, use time 533.088
epoch 1, loss 0.3160, train accuracy 0.890, test accuracy 0.893, use time 530.857
epoch 2, loss 0.2278, train accuracy 0.922, test accuracy 0.919, use time 531.327
epoch 3, loss 0.1752, train accuracy 0.939, test accuracy 0.915, use time 531.334
epoch 4, loss 0.1408, train accuracy 0.950, test accuracy 0.910, use time 530.288
epoch 5, loss 0.1163, train accuracy 0.960, test accuracy 0.921, use time 528.114
epoch 6, loss 0.0936, train accuracy 0.967, test accuracy 0.933, use time 532.579
epoch 7, loss 0.0835, train accuracy 0.971, test accuracy 0.919, use time 530.987
epoch 8, loss 0.0688, train accuracy 0.976, test accuracy 0.928, use time 532.107
epoch 9, loss 0.0608, train accuracy 0.979, test accuracy 0.937, use time 532.941
epoch 10, loss 0.0511, train accuracy 0.982, test accuracy 0.931, use time 532.653
epoch 11, loss 0.0435, train accuracy 0.985, test accuracy 0.937, use time 532.789
epoch 12, loss 0.0419, train accuracy 0.986, test accuracy 0.932, use time 532.151
epoch 13, loss 0.0394, train accuracy 0.987, test accuracy 0.935, use time 532.003
epoch 14, loss 0.0329, train accuracy 0.989, test accuracy 0.939, use time 532.184
```

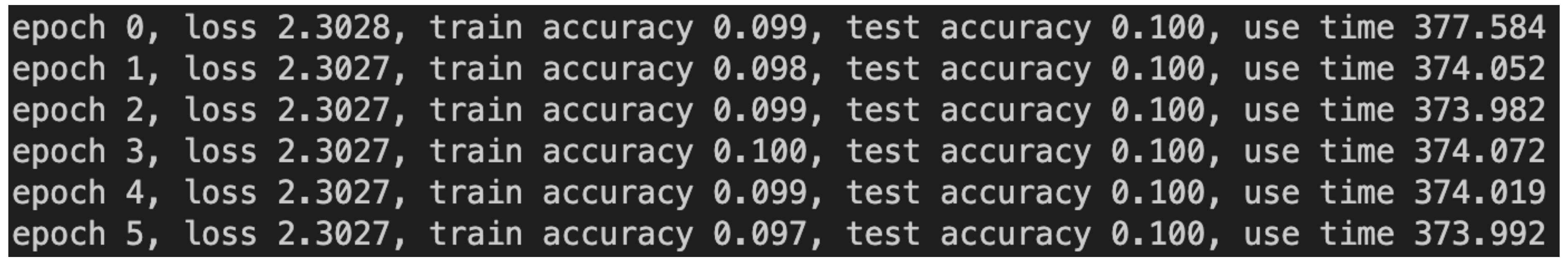
Vgg19



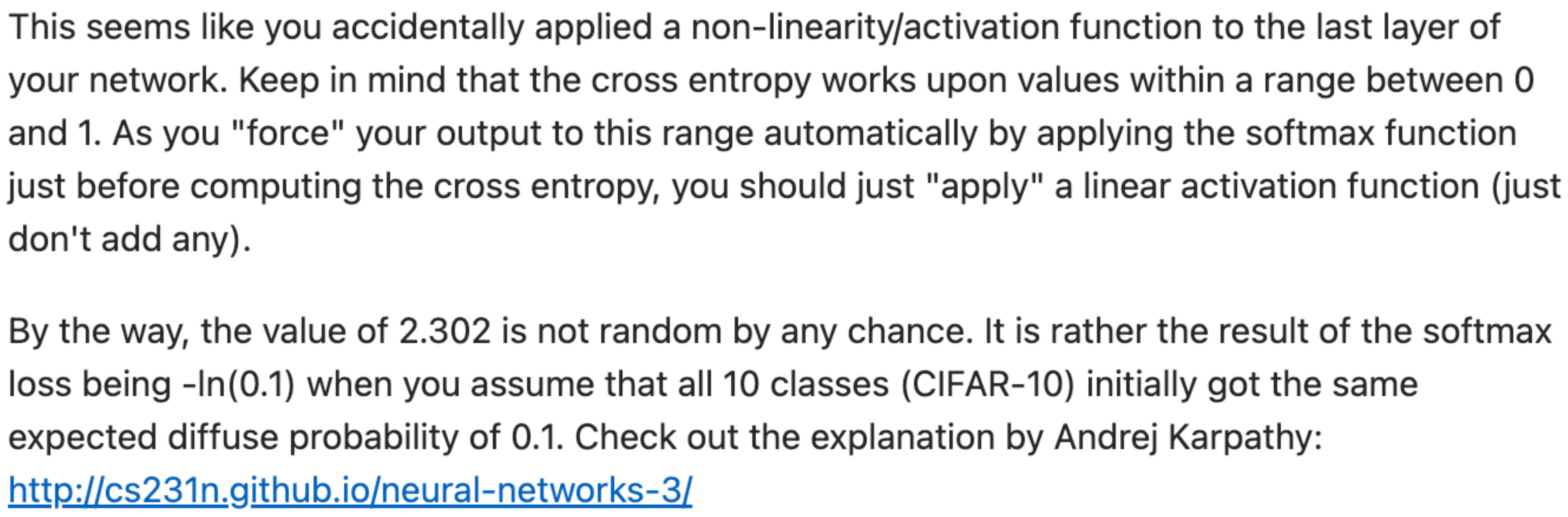


```
if init_weights:
    for m in self.modules():
        if isinstance(m, nn.Conv2d):
            nn.init.kaiming_normal_(m.weight, mode="fan_out", nonlinearity="relu")
            if m.bias is not None:
                nn.init.constant_(m.bias, 0)
        elif isinstance(m, nn.BatchNorm2d):
            nn.init.constant_(m.weight, 1)
            nn.init.constant_(m.bias, 0)
        elif isinstance(m, nn.Linear):
            nn.init.normal_(m.weight, 0, 0.01)
            nn.init.constant_(m.bias, 0)
```

loss 2.302?

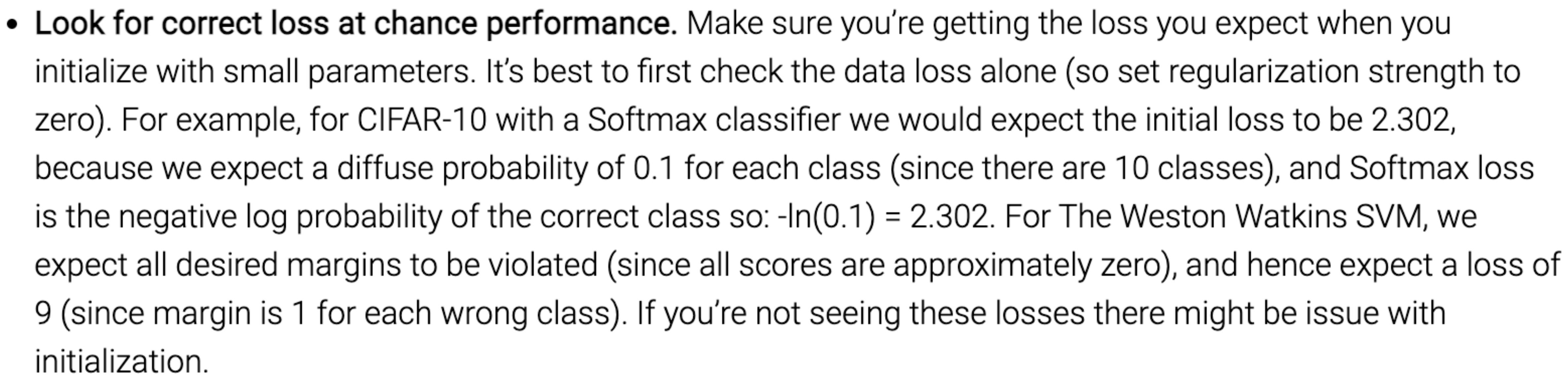


```
epoch 0, loss 2.3028, train accuracy 0.099, test accuracy 0.100, use time 377.584
epoch 1, loss 2.3027, train accuracy 0.098, test accuracy 0.100, use time 374.052
epoch 2, loss 2.3027, train accuracy 0.099, test accuracy 0.100, use time 373.982
epoch 3, loss 2.3027, train accuracy 0.100, test accuracy 0.100, use time 374.072
epoch 4, loss 2.3027, train accuracy 0.099, test accuracy 0.100, use time 374.019
epoch 5, loss 2.3027, train accuracy 0.097, test accuracy 0.100, use time 373.992
```



This seems like you accidentally applied a non-linearity/activation function to the last layer of your network. Keep in mind that the cross entropy works upon values within a range between 0 and 1. As you "force" your output to this range automatically by applying the softmax function just before computing the cross entropy, you should just "apply" a linear activation function (just don't add any).

By the way, the value of 2.302 is not random by any chance. It is rather the result of the softmax loss being -ln(0.1) when you assume that all 10 classes (CIFAR-10) initially got the same expected diffuse probability of 0.1. Check out the explanation by Andrej Karpathy: http://cs231n.github.io/neural-networks-3/



- **Look for correct loss at chance performance.** Make sure you're getting the loss you expect when you initialize with small parameters. It's best to first check the data loss alone (so set regularization strength to zero). For example, for CIFAR-10 with a Softmax classifier we would expect the initial loss to be 2.302, because we expect a diffuse probability of 0.1 for each class (since there are 10 classes), and Softmax loss is the negative log probability of the correct class so: -ln(0.1) = 2.302. For The Weston Watkins SVM, we expect all desired margins to be violated (since all scores are approximately zero), and hence expect a loss of 9 (since margin is 1 for each wrong class). If you're not seeing these losses there might be issue with initialization.