

ORR

```
OrderedDict([('net.0.linear.weight', tensor([[ -0.8393, -0.0455,  0.1022,
  0.0614, -0.1148, -0.1206,  0.7439,  0.0911,
          -0.2051,  0.1094,  0.4079, -0.4002,  0.8954, -0.1835,  0.
0779],
          [ 0.2697,  0.4035,  0.1082, -0.1780, -0.3289, -0.0073,  0.0
324,  0.3732,
          -0.2218, -0.2258,  0.0560, -0.3470,  0.5542,  0.6065,  1.
2130],
          [ 0.5472,  0.3550,  0.0171, -0.5436,  0.2452, -0.1550,  0.
2094,  0.9828,
          0.4851,  0.0391, -0.8468, -0.1782,  0.0114,  0.4483, -1.
1195],
          [-0.8116,  0.4559, -0.0882,  0.3859, -0.2538, -0.3537, -0.0
893, -0.6286,
          0.2213,  0.1423,  0.3971,  0.1832,  0.4481, -0.1643, -0.
2083],
          [-0.4549,  0.1095, -0.0701,  0.5685,  0.1349, -0.0996, -0.4
463, -0.0570,
          -0.2087, -0.3780,  0.1199, -0.0688, -0.2131, -0.9682, -0.1
105],
          [-0.4670,  0.5797, -0.3416,  0.1590, -0.3778, -0.2368,  0.1
486, -0.3062,
```

-0.3284, 0.2763, -0.1981, -0.3427, 0.6484, 0.0755, 0.  
9327],

[-0.4277, 0.1822, 0.3613, -0.2516, -0.1479, -0.1276, 0.0  
624, -0.1571,

-0.7449, -0.2724, -0.7899, 0.0639, -0.4880, 0.5377, -0.7  
251],

[-0.3430, 0.3078, -0.4976, 0.2299, 0.6850, 0.1164, 0.  
3359, -0.2098,

-0.0054, -0.4260, -0.1360, 0.3958, 0.0999, 0.6010, 0.  
5937],

[ 0.4526, 0.0083, -0.1981, -0.0720, 0.1366, 0.3718, -0.4  
664, -0.9241,

0.2458, 0.0652, 0.6300, -0.1094, -0.3970, -0.1973, 1.  
0343],

[-0.1934, -0.2513, 0.2839, -0.3473, -0.0956, -0.1301, 0.5  
027, -0.3424,

-0.8224, 0.2758, 0.2259, -0.1178, -0.3921, -0.1477, -0.7  
204],

[-0.6542, 0.2426, -0.0636, 0.8750, 0.0227, 0.0452, -0.5  
561, -0.6689,

-0.7670, -0.5042, 0.5617, 0.3990, -0.0643, -0.0358, 0.4  
405],

[-1.0848, 0.2706, -0.4175, 0.0744, 0.2709, 0.3189, 0.  
4006, -0.2645,  
-0.3037, -0.1344, -0.0968, 0.0268, 0.2774, 0.1120, -0.1  
409],  
[-0.4152, 0.1568, 0.6052, -1.2651, 0.9304, -0.3554, 0.1  
112, 0.4761,  
-0.0896, -0.5201, 1.1382, 0.3419, 0.6181, 0.1975, 0.  
0743],  
[-0.2297, -0.6924, -0.1031, 0.1003, -0.4640, 0.9988, -0.0  
474, 0.1289,  
-0.1268, -0.0765, -0.0778, -0.3597, -0.2553, 0.7669, 0.8  
185],  
[-0.7961, -0.2912, 0.4429, -0.4902, -0.1096, 0.1406, 0.5  
441, -0.6158,  
-0.0578, -0.1913, 0.1979, 0.3792, -0.6417, -0.0643, -0.2  
027],  
[-0.5238, 0.2394, -0.1477, -0.2964, 0.0856, 0.1811, 0.4  
003, -0.1691,  
-0.1326, -0.1937, -0.5242, -0.1953, -0.0964, 0.2149, -0.4  
602]]), ('net.0.linear.bias', tensor([ 0.2351, 0.1046, -0.2425, -0.1257,  
-0.0152, 0.1421, -0.0893, -0.1652,

0.1105, -0.0259, -0.1436, 0.3173, 0.0072, -0.4722, -0.40  
16, 0.0850]), ('net.0.batch\_norm.weight', tensor([0.7312, 1.2224, 1.2  
051, 0.7478, 0.8436, 1.0615, 0.9492, 1.0329, 1.0464,  
1.4702, 1.1644, 0.7817, 1.0234, 1.1938, 0.8728, 1.0455])),  
('net.0.batch\_norm.bias', tensor([-2.0105e-02, -1.6940e-01, 4.2450e-0  
2, 7.1827e-04, 1.6335e-01,  
1.2032e-01, 1.2300e-01, -3.8219e-02, -9.6014e-02, 6.031  
4e-02,  
-1.3299e-02, 8.9280e-05, 5.0680e-02, -1.2988e-01, -1.704  
1e-01,  
-1.1862e-01])), ('net.0.batch\_norm.running\_mean', tensor([2.7  
626e+00, 1.5985e+00, 2.8279e-01, 2.5888e+00, 1.3804e-12, 2.1835e+  
00,  
1.0308e-02, 5.7842e-01, 2.7212e-01, 1.9399e-05, 1.0827e-0  
1, 6.3382e-01,  
1.5939e+00, 1.0826e-01, 1.0323e-04, 1.0008e-02])), ('net.0.b  
atch\_norm.running\_var', tensor([1.1218e+00, 6.3962e-01, 2.0054e-01,  
9.4152e-01, 3.5860e-12, 8.4270e-01,  
4.9013e-03, 3.2804e-01, 1.5893e-01, 1.1466e-05, 9.9371e-0  
2, 6.4485e-01,  
1.0808e+00, 1.0942e-01, 6.0142e-05, 5.3393e-03])), ('net.0.b  
atch\_norm.num\_batches\_tracked', tensor(252)), ('net.1.linear.weight', te

nsor([[ 1.6525e-01, 6.7426e-01, -1.9293e-01, -2.5761e-01, -3.6555e-  
02,  
8.6560e-02, -1.8647e-02, -8.0940e-02, 7.1926e-01, -5.89  
63e-01,  
4.5432e-01, -2.0602e-01, -4.4301e-01, 7.6939e-01, -4.98  
59e-01,  
-4.8401e-01],  
[-4.9193e-01, -8.8527e-01, 4.1997e-01, 4.2334e-02, 6.15  
81e-02,  
-6.7496e-02, 2.4685e-01, -6.2281e-01, -8.6493e-01, -1.69  
97e-01,  
2.0557e-01, -2.0066e-01, -3.8295e-01, -2.5472e-01, -2.35  
54e-01,  
3.4244e-01],  
[-2.1124e-01, -4.1686e-01, 4.7788e-01, -2.9263e-01, -9.613  
4e-02,  
-7.9582e-02, 3.4482e-01, -7.3794e-02, -6.7243e-01, 4.57  
91e-02,  
3.0668e-02, 4.9877e-01, -9.9365e-01, -1.7205e-01, 1.05  
14e-01,  
1.7751e-01],

[-7.2899e-02, -2.4352e-01, 7.5088e-01, -4.9421e-01, -1.462  
1e-01,  
-5.4741e-01, -3.9055e-01, -6.1722e-01, -2.9312e-01, -4.431  
4e-01,  
-4.0046e-01, -3.4224e-01, 2.4217e-01, -2.7945e-01, 7.38  
72e-03,  
-1.2595e-01],  
[-1.9472e-01, 4.6701e-01, 3.4983e-02, 1.6210e-01, 3.55  
85e-01,  
3.2768e-01, -2.1492e-01, 4.4453e-01, -1.3115e-01, -7.85  
15e-01,  
3.8148e-01, -4.2568e-01, 1.8632e-02, 8.5167e-01, 7.5  
423e-01,  
-5.6742e-02],  
[ 5.9978e-01, -1.3607e-01, 4.4296e-01, -8.4701e-05, -1.87  
98e-01,  
-3.4729e-01, 4.2062e-01, -1.1904e-01, -1.2278e-01, -1.16  
66e-01,  
-1.6444e-01, -1.1233e+00, 1.4937e-01, -9.3663e-02, 4.34  
78e-01,  
7.3869e-01],

[-2.2400e-01, -5.4947e-01, 7.6777e-02, 1.1893e-01, 2.58  
01e-01,  
-3.5132e-01, 4.4057e-01, -3.4069e-01, -8.6527e-01, -1.49  
85e-01,  
3.9792e-02, 6.0084e-01, -8.6417e-01, -6.7962e-01, -4.28  
54e-01,  
1.1282e-01],  
[ 3.2987e-01, 1.6544e-01, -1.5679e-03, -2.4796e-01, 1.40  
49e-01,  
-1.8796e-01, 5.8703e-02, -4.1687e-01, 2.0840e-01, 4.54  
30e-01,  
6.7569e-01, -8.8422e-01, -5.6397e-02, -2.3841e-01, 2.97  
45e-01,  
3.2151e-01]]), ('net.1.linear.bias', tensor([-0.0513, 0.308  
8, 0.0356, 0.3774, -0.0040, 0.1649, 0.2468, -0.0041])), ('net.1.ba  
tch\_norm.weight', tensor([1.2236, 1.1533, 1.0977, 1.0693, 1.1277, 0.9  
930, 0.9161, 0.6519])), ('net.1.batch\_norm.bias', tensor([-0.2032, 0.2  
212, 0.6121, 0.3633, -0.4583, 0.4207, 0.4280, -0.4205])), ('net.1.  
batch\_norm.running\_mean', tensor([0.7265, 1.4362, 0.8588, 1.2245, 0.  
5361, 0.6550, 1.3287, 0.4814])), ('net.1.batch\_norm.running\_var', tens  
or([2.5507, 2.8423, 1.9457, 2.8786, 2.1127, 0.9930, 2.4972, 0.561  
9])), ('net.1.batch\_norm.num\_batches\_tracked', tensor(252)), ('net.2.wei

ght', tensor([[ -0.5329, 0.2355, 0.2850, 0.3761, -0.3019, 0.3423,  
0.3634, 0.0202]]))

DCR

```
OrderedDict([('net.0.linear.weight', tensor([[ 0.4103,  0.4802, -0.3575,
-0.0108,  0.0939,  0.5823, -0.4312,  0.3008,
          0.3302, -0.3976,  0.6730,  0.0017,  0.5069, -0.3433, -0.
1803],
          [ 0.0096, -0.1199, -0.3667,  0.3688, -0.1152, -0.4824, -0.4
949, -0.5499,
          0.1263,  0.6620,  0.4397,  0.2657,  0.0902, -0.1421,  0.
0281],
          [ 0.2765, -0.2659, -0.3458,  0.1032,  0.2032, -0.1919,  0.4
390,  0.2199,
          0.1486,  0.6287, -0.1141,  0.0386,  0.5357,  0.2316, -0.
4601],
          [-0.1883,  0.1584, -0.0220, -0.5876,  0.6140,  0.1059,  0.6
967, -0.6014,
          0.1180,  0.3132,  0.6498, -0.0145,  0.5526,  0.5028,  0.
5151]])), ('net.0.linear.bias', tensor([ 0.2940,  0.4012, -0.2402, -0.302
5])), ('net.0.batch_norm.weight', tensor([1.0734, 1.1909, 0.7024, 0.867
0])), ('net.0.batch_norm.bias', tensor([ 0.1839, -0.1125, -0.2452, -0.43
14])), ('net.0.batch_norm.running_mean', tensor([1.6440, 2.4807, 2.947
1, 3.4413])), ('net.0.batch_norm.running_var', tensor([0.7675, 0.6047,
0.7720, 0.8696])), ('net.0.batch_norm.num_batches_tracked', tensor(12
0)), ('net.1.linear.weight', tensor([[ -0.1067, -1.4193,  0.6960, -0.2460],
```

```
[-0.7609, 0.4822, 0.9258, 1.6678],  
[ 1.0075, -1.4479, -0.2113, 0.2078],  
[-0.2593, 0.2304, 0.0489, 0.2933]])), ('net.1.linear.bias', t  
ensor([-0.5716, 0.1457, 0.0681, -0.1421])), ('net.1.batch_norm.weig  
ht', tensor([0.6775, 0.6187, 0.3079, 0.3104])), ('net.1.batch_norm.bias',  
tensor([ 0.8798, -1.0616, -0.9060, -1.0985])), ('net.1.batch_norm.run  
ning_mean', tensor([0.5402, 0.5552, 1.1152, 0.0958])), ('net.1.batch_n  
orm.running_var', tensor([1.4197, 0.9728, 3.7899, 0.0432])), ('net.1.ba  
tch_norm.num_batches_tracked', tensor(120)), ('net.2.weight', tensor([[  
0.7709, -0.5515, -0.7008, -0.5478]])))]
```

Responder

```
OrderedDict([('net.0.linear.weight', tensor([[ -0.0809, -0.0795,  0.1795,
  0.0418, -0.1517, -0.7644, -0.0250,  1.0789,
    -0.2142, -0.1237, -0.7041,  0.4009, -0.0502, -0.2291, -0.0
512],
  [ 0.5147, -0.5862, -0.0103,  0.3739, -0.5364,  0.0704, -0.2
829, -0.9063,
    0.2137,  0.1741, -0.4399, -0.3887,  0.2876, -0.7271, -0.5
617],
  [-0.0898,  0.0542, -0.1222,  0.3138,  0.1093, -0.1611, -0.3
371, -0.3977,
    -0.1485, -0.2840, -0.2922,  0.2183, -0.0846, -0.2236,  0.4
234],
  [-0.2072,  0.0361,  0.4021,  0.4635,  0.1606,  0.6168,  0.
3328, -0.0443,
    0.0643,  0.3048, -0.2873, -0.1667, -0.4465,  0.5364, -0.1
347],
  [ 0.1041, -0.7965,  0.2502,  0.3336,  0.7849, -0.4881, -0.5
794, -1.2234,
    -0.1952,  0.1656, -0.1434, -0.1469,  0.5828, -0.8002, -0.1
676],
  [-0.0903,  0.0186, -0.8024,  0.2451,  0.3429,  0.3294, -0.0
365,  0.0998,
```

-0.2161, 0.2580, -0.1379, -0.0107, -0.5985, -0.3632, 0.1  
559],  
[-0.3309, -1.1762, -0.1649, -0.6400, -0.2062, -0.1673, 0.39  
77, -0.1730,  
-0.2402, -0.0732, -0.2255, -0.4033, 0.4980, 0.2243, 0.1  
552],  
[ 0.3666, 0.1776, -0.0888, -0.2974, -0.0314, -0.7288, -0.0  
339, -0.0311,  
-0.1869, -0.2934, -0.0403, 0.6866, 0.7424, -0.0508, -0.1  
703],  
[ 0.5348, 0.4715, 0.0345, 0.8360, 0.0569, 0.0524, 0.  
0685, -0.4291,  
-0.1532, -0.3911, -0.1431, 0.2844, -0.1924, 0.0628, -0.6  
395],  
[-0.1395, 0.0557, 0.0606, 0.9464, 0.4938, 0.0331, 0.  
4472, 0.0207,  
-0.1598, -0.0298, -0.3810, 0.0804, 0.4541, 1.5929, 0.  
0066],  
[-0.2185, -0.0522, -0.4931, 0.0417, 1.0544, -0.1208, -0.2  
756, 0.3080,  
-0.0401, -0.1150, -0.1590, 0.0464, -0.7883, -0.0600, -0.4  
136],

[-0.0170, 0.3727, 0.4981, -0.5791, -0.1891, 0.5324, 0.1  
744, 0.6083,  
0.1145, 0.5592, -0.3824, -0.2755, -0.1427, 0.3600, 0.  
0957],  
[ 0.0778, -0.3289, -0.4172, -0.2462, 0.2275, -0.1397, 0.1  
179, 0.4069,  
-0.2859, -0.6001, -0.6674, -0.4962, 0.4061, 0.0816, 0.2  
971],  
[-0.0213, 0.6351, 0.2320, -0.5524, -0.4702, 0.0682, -0.6  
138, 0.3539,  
0.5307, 0.3474, 0.1275, -0.2588, 0.6806, 0.3603, -0.  
0706],  
[-0.1924, -0.2437, 0.6285, 0.1325, 0.0280, 0.3230, 0.  
7625, -0.4221,  
0.0931, 0.6833, -0.0214, 0.2380, -0.0399, 0.2060, 0.  
4679],  
[-0.3746, -0.1292, 0.4932, 0.1327, -0.0309, 0.0941, 0.5  
298, 0.0549,  
-0.7280, -0.0457, -0.2753, -0.1583, -0.1984, -1.2612, -0.18  
40]]), ('net.0.linear.bias', tensor([-0.1875, 0.2535, 0.0475, 0.2728,  
0.1301, 0.0808, -0.0562, 0.4615,

0.3479, -0.2502, 0.0811, 0.5616, -0.0825, 0.0126, 0.2723, -0.0631)), ('net.0.batch\_norm.weight', tensor([1.0000, 0.9796, 1.0000, 1.3295, 1.1274, 1.0000, 1.1123, 1.1554, 1.0607, 1.1837, 1.0000, 1.1061, 0.8349, 0.9107, 0.9365, 0.8930])), ('net.0.batch\_norm.bias', tensor([0.1062, 0.0800, 0.0700, -0.2030, -0.1355, -0.1892, -0.1046, 0.1998, 0.0256, -0.2712, -0.1179, 0.2745, -0.0097, 0.0397, -0.5497, -0.1027])), ('net.0.batch\_norm.running\_mean', tensor([0.0000e+00, 2.8731e-01, 0.0000e+00, 1.3628e+00, 5.9712e-01, 0.0000e+00, 3.6306e-02, 1.3917e+00, 1.7101e-01, 3.4342e+00, 0.0000e+00, 1.9092e+00, 3.2593e-18, 4.0053e+00, 3.0359e+00, 2.0203e-07])), ('net.0.batch\_norm.running\_var', tensor([7.8227e-17, 2.3632e-01, 7.8227e-17, 3.5071e-01, 2.9482e-01, 7.8227e-17, 2.8496e-02, 4.4110e-01, 1.8245e-01, 6.2787e-01, 7.8227e-17, 5.2439e-01, 7.9199e-17, 9.0294e-01, 7.7338e-01, 8.9641e-08])), ('net.0.batch\_norm.num\_batches\_tracked', tensor(352)), ('net.1.linear.weight', tensor([[0.1809, -0.1523, -0.1854, 0.3446, -0.1983, -0.1807, -0.4087, -0.4657, 0.0276, 0.6830, 0.6967, 0.0526, 0.0563, 0.6120, 0.6683, 0.3254],

[-0.3949, 0.3565, -0.6347, -0.0442, 0.3550, 0.0972, 0.6  
976, 0.2563,  
0.3192, 0.4385, -0.2279, -0.3401, -0.6874, -0.4223, -0.1  
325, -0.2595],  
[ 0.3660, 0.5440, 0.1553, 0.3594, -0.3406, -0.6700, -0.4  
278, -0.2630,  
-0.4371, 0.4540, 0.0411, 0.0977, 0.1347, -0.2749, 0.  
6787, -0.3602],  
[-0.1736, -0.2470, 0.3085, 0.4464, -0.3650, 0.8129, -0.5  
133, -0.6761,  
-0.0695, 0.5470, -0.2664, -0.0383, 0.5905, -0.2973, 0.3  
522, 0.1632]])), ('net.1.linear.bias', tensor([-0.2017, 0.0670, 0.222  
6, -0.0305])), ('net.1.batch\_norm.weight', tensor([1.1399, 1.1082, 0.85  
53, 0.8135])), ('net.1.batch\_norm.bias', tensor([ 0.3010, -0.1620, 0.3  
743, 0.3551])), ('net.1.batch\_norm.running\_mean', tensor([0.5684, 0.5  
670, 0.7493, 0.6150])), ('net.1.batch\_norm.running\_var', tensor([1.277  
8, 1.6483, 1.2780, 1.3734])), ('net.1.batch\_norm.num\_batches\_tracked',  
tensor(352)), ('net.2.weight', tensor([[[-0.2359, 0.5702, -0.2433, -0.3  
592]]]))))