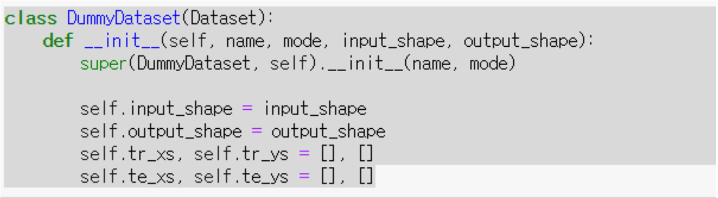
# 거대심층구조 신경망

Dummy Dataset을 가져와서 실습을 진행하였다.

- Inception Model 실행 과정 및 결과



```
class DummyDataset(Dataset):
    def __init__(self, name, mode, input_shape, output_shape):
        super(DummyDataset, self).__init__(name, mode)

        self.input_shape = input_shape
        self.output_shape = output_shape
        self.tr_xs, self.tr_ys = [], []
        self.te_xs, self.te_ys = [], []
```

```
%run ../chap09/cnn_ext_model.ipynb
%run ../chap09/dataset_dummy.ipynb

imagenet = DummyDataset('imagenet', 'select', [299,299,3], 200)

CnnExtModel.set_macro('v3_preproc',
    ['serial',
        ['conv', {'ksize':3, 'stride':2, 'chn':32, 'padding':'VALID'}],
        ['conv', {'ksize':3, 'chn':32, 'padding':'VALID'}],
        ['conv', {'ksize':3, 'chn':64, 'padding':'SAME'}],
        ['max', {'ksize':3, 'stride':2, 'padding':'VALID'}],
        ['conv', {'ksize':1, 'chn':80, 'padding':'VALID'}],
        ['conv', {'ksize':3, 'chn':192, 'padding':'VALID'}],
        ['max', {'ksize':3, 'stride':2, 'padding':'VALID'}]])

CnnExtModel.set_macro('v3_inception1',
    ['parallel',
        ['conv', {'ksize':1, 'chn':64}],
        ['serial',
            ['conv', {'ksize':1, 'chn':48}],
            ['conv', {'ksize':5, 'chn':64}]],
        ['serial',
            ['conv', {'ksize':1, 'chn':64}],
            ['conv', {'ksize':3, 'chn':96}],
            ['conv', {'ksize':3, 'chn':96}]],
        ['serial',
            ['avg', {'ksize':3, 'stride':1}],
            ['conv', {'ksize':1, 'chn':'#chn'}]]])

CnnExtModel.set_macro('v3_resize1',
    ['parallel',
        ['conv', {'ksize':3, 'stride':2, 'chn':384}],
        ['serial',
            ['conv', {'ksize':1, 'chn':64}],
            ['conv', {'ksize':3, 'chn':96}],
            ['conv', {'ksize':3, 'stride':2, 'chn':96}]],
        ['max', {'ksize':3, 'stride':2}]])

CnnExtModel.set_macro('v3_inception2',
    ['parallel',
        ['conv', {'ksize':1, 'chn':192}],
        ['serial',
            ['conv', {'ksize':[1,1], 'chn':'#chn'}],
            ['conv', {'ksize':[1,7], 'chn':'#chn'}],
            ['conv', {'ksize':[7,1], 'chn':192}]],
        ['serial',
            ['conv', {'ksize':[1,1], 'chn':'#chn'}],
            ['conv', {'ksize':[7,1], 'chn':'#chn'}],
            ['conv', {'ksize':[1,7], 'chn':'#chn'}],
            ['conv', {'ksize':[7,1], 'chn':'#chn'}],
            ['conv', {'ksize':[1,7], 'chn':192}]],
        ['serial',
            ['avg', {'ksize':3, 'stride':1}],
            ['conv', {'ksize':1, 'chn':192}]]])

CnnExtModel.set_macro('v3_resize2',
    ['parallel',
        ['serial',
            ['conv', {'ksize':1, 'chn':192}],
            ['conv', {'ksize':3, 'stride':2, 'chn':320}]],
        ['serial',
            ['conv', {'ksize':[1,1], 'chn':192}],
            ['conv', {'ksize':[1,7], 'chn':192}],
            ['conv', {'ksize':[7,1], 'chn':192}],
            ['conv', {'ksize':[3,3], 'stride':[2,2], 'chn':192}]],
        ['max', {'ksize':3, 'stride':2}]])

CnnExtModel.set_macro('v3_inception3',
    ['parallel',
        ['conv', {'ksize':1, 'chn':320}],
        ['serial',
            ['conv', {'ksize':[3,3], 'chn':384}],
            ['parallel',
                ['conv', {'ksize':[1,3], 'chn':384}],
                ['conv', {'ksize':[3,1], 'chn':384}]]],
        ['serial',
            ['conv', {'ksize':[1,1], 'chn':448}],
            ['conv', {'ksize':[3,3], 'chn':384}],
            ['parallel',
                ['conv', {'ksize':[1,3], 'chn':384}],
                ['conv', {'ksize':[3,1], 'chn':384}]]],
        ['serial',
            ['avg', {'ksize':3, 'stride':1}],
            ['conv', {'ksize':1, 'chn':192}]]]])

CnnExtModel.set_macro('v3_postproc',
    ['serial',
        ['avg', {'stride':8}],
        ['dropout', {'keep_prob':0.7}]])

CnnExtModel.set_macro('inception_v3',
    ['serial',
        ['custom', {'name':'v3_preproc'}],
        ['custom', {'name':'v3_inception1', 'args':{'#chn':32}}],
        ['custom', {'name':'v3_inception1', 'args':{'#chn':64}}],
        ['custom', {'name':'v3_inception1', 'args':{'#chn':64}}],
        ['custom', {'name':'v3_resize1'}],
        ['custom', {'name':'v3_inception2', 'args':{'#chn':128}}],
        ['custom', {'name':'v3_inception2', 'args':{'#chn':160}}],
        ['custom', {'name':'v3_inception2', 'args':{'#chn':160}}],
        ['custom', {'name':'v3_inception2', 'args':{'#chn':192}}],
        ['custom', {'name':'v3_resize2'}],
        ['custom', {'name':'v3_inception3'}],
        ['custom', {'name':'v3_inception3'}],
        ['custom', {'name':'v3_postproc'}]])

inception_v3 = CnnExtModel('inception_v3', imagenet,
                             [['custom', {'name':'inception_v3'}]], dump_structure=True)
```

```
custom inception_v3
  serial
    custom v3_preproc
      serial
        1: conv, [299, 299, 3]=>[148, 148, 32] pm:3x3x3x32+32=896
        2: conv, [148, 148, 32]=>[146, 146, 32] pm:3x3x32x32+32=9248
        3: conv, [146, 146, 32]=>[146, 146, 64] pm:3x3x32x64+64=18496
        4: max, [146, 146, 64]=>[72, 72, 64]
        5: conv, [72, 72, 64]=>[72, 72, 80] pm:1x1x64x80+80=5200
        6: conv, [72, 72, 80]=>[70, 70, 192] pm:3x3x80x192+192=138432
        7: max, [70, 70, 192]=>[34, 34, 192]
    custom v3_inception1
      parallel
        8: conv, [34, 34, 192]=>[34, 34, 64] pm:1x1x192x64+64=12352
        serial
          9: conv, [34, 34, 192]=>[34, 34, 48] pm:1x1x192x48+48=9264
          10: conv, [34, 34, 48]=>[34, 34, 64] pm:5x5x48x64+64=76864
        serial
          11: conv, [34, 34, 192]=>[34, 34, 64] pm:1x1x192x64+64=12352
          12: conv, [34, 34, 64]=>[34, 34, 96] pm:3x3x64x96+96=55392
          13: conv, [34, 34, 96]=>[34, 34, 96] pm:3x3x96x96+96=83040
        serial
          14: avg, [34, 34, 192]=>[34, 34, 192]
          15: conv, [34, 34, 192]=>[34, 34, 32] pm:1x1x192x32+32=6176
    custom v3_inception1
      parallel
        16: conv, [34, 34, 256]=>[34, 34, 64] pm:1x1x256x64+64=16448
        serial
          17: conv, [34, 34, 256]=>[34, 34, 48] pm:1x1x256x48+48=12336
          18: conv, [34, 34, 48]=>[34, 34, 64] pm:5x5x48x64+64=76864
        serial
          19: conv, [34, 34, 256]=>[34, 34, 64] pm:1x1x256x64+64=16448
          20: conv, [34, 34, 64]=>[34, 34, 96] pm:3x3x64x96+96=55392
          21: conv, [34, 34, 96]=>[34, 34, 96] pm:3x3x96x96+96=83040
        serial
          22: avg, [34, 34, 256]=>[34, 34, 256]
          23: conv, [34, 34, 256]=>[34, 34, 64] pm:1x1x256x64+64=16448
    custom v3_inception1
      parallel
        24: conv, [34, 34, 288]=>[34, 34, 64] pm:1x1x288x64+64=18496
        serial
          25: conv, [34, 34, 288]=>[34, 34, 48] pm:1x1x288x48+48=13872
          26: conv, [34, 34, 48]=>[34, 34, 64] pm:5x5x48x64+64=76864
        serial
          27: conv, [34, 34, 288]=>[34, 34, 64] pm:1x1x288x64+64=18496
          28: conv, [34, 34, 64]=>[34, 34, 96] pm:3x3x64x96+96=55392
          29: conv, [34, 34, 96]=>[34, 34, 96] pm:3x3x96x96+96=83040
        serial
          30: avg, [34, 34, 288]=>[34, 34, 288]
          31: conv, [34, 34, 288]=>[34, 34, 64] pm:1x1x288x64+64=18496
    custom v3_resize1
      parallel
        32: conv, [34, 34, 288]=>[17, 17, 384] pm:3x3x288x384+384=995712
        serial
          33: conv, [34, 34, 288]=>[34, 34, 64] pm:1x1x288x64+64=18496
          34: conv, [34, 34, 64]=>[34, 34, 96] pm:3x3x64x96+96=55392
          35: conv, [34, 34, 96]=>[17, 17, 96] pm:3x3x96x96+96=83040
        36: max, [34, 34, 288]=>[17, 17, 288]
    custom v3_inception2
      parallel
        37: conv, [17, 17, 768]=>[17, 17, 192] pm:1x1x768x192+192=147648
        serial
          38: conv, [17, 17, 768]=>[17, 17, 128] pm:1x1x768x128+128=98432
          39: conv, [17, 17, 128]=>[17, 17, 128] pm:1x7x128x128+128=114816
          40: conv, [17, 17, 128]=>[17, 17, 192] pm:7x1x128x192+192=172224
        serial
          41: conv, [17, 17, 768]=>[17, 17, 128] pm:1x1x768x128+128=98432
          42: conv, [17, 17, 128]=>[17, 17, 128] pm:7x1x128x128+128=114816
          43: conv, [17, 17, 128]=>[17, 17, 128] pm:1x7x128x128+128=114816
          44: conv, [17, 17, 128]=>[17, 17, 128] pm:7x1x128x128+128=114816
          45: conv, [17, 17, 128]=>[17, 17, 192] pm:1x7x128x192+192=172224
        serial
          46: avg, [17, 17, 768]=>[17, 17, 768]
          47: conv, [17, 17, 768]=>[17, 17, 192] pm:1x1x768x192+192=147648
    custom v3_inception2
      parallel
        48: conv, [17, 17, 768]=>[17, 17, 192] pm:1x1x768x192+192=147648
        serial
          49: conv, [17, 17, 768]=>[17, 17, 160] pm:1x1x768x160+160=123040
          50: conv, [17, 17, 160]=>[17, 17, 160] pm:1x7x160x160+160=179360
          51: conv, [17, 17, 160]=>[17, 17, 192] pm:7x1x160x192+192=215232
        serial
          52: conv, [17, 17, 768]=>[17, 17, 160] pm:1x1x768x160+160=123040
          53: conv, [17, 17, 160]=>[17, 17, 160] pm:7x1x160x160+160=179360
          54: conv, [17, 17, 160]=>[17, 17, 160] pm:1x7x160x160+160=179360
          55: conv, [17, 17, 160]=>[17, 17, 160] pm:7x1x160x160+160=179360
          56: conv, [17, 17, 160]=>[17, 17, 192] pm:1x7x160x192+192=215232
        serial
          57: avg, [17, 17, 768]=>[17, 17, 768]
          58: conv, [17, 17, 768]=>[17, 17, 192] pm:1x1x768x192+192=147648
    custom v3_inception2
      parallel
        59: conv, [17, 17, 768]=>[17, 17, 192] pm:1x1x768x192+192=147648
        serial
          60: conv, [17, 17, 768]=>[17, 17, 160] pm:1x1x768x160+160=123040
          61: conv, [17, 17, 160]=>[17, 17, 160] pm:1x7x160x160+160=179360
          62: conv, [17, 17, 160]=>[17, 17, 192] pm:7x1x160x192+192=215232
        serial
          63: conv, [17, 17, 768]=>[17, 17, 160] pm:1x1x768x160+160=123040
          64: conv, [17, 17, 160]=>[17, 17, 160] pm:7x1x160x160+160=179360
          65: conv, [17, 17, 160]=>[17, 17, 160] pm:1x7x160x160+160=179360
          66: conv, [17, 17, 160]=>[17, 17, 160] pm:7x1x160x160+160=179360
          67: conv, [17, 17, 160]=>[17, 17, 192] pm:1x7x160x192+192=215232
        serial
          68: avg, [17, 17, 768]=>[17, 17, 768]
          69: conv, [17, 17, 768]=>[17, 17, 192] pm:1x1x768x192+192=147648
    custom v3_inception2
      parallel
        70: conv, [17, 17, 768]=>[17, 17, 192] pm:1x1x768x192+192=147648
        serial
          71: conv, [17, 17, 768]=>[17, 17, 192] pm:1x1x768x192+192=147648
          72: conv, [17, 17, 192]=>[17, 17, 192] pm:1x7x192x192+192=258240
          73: conv, [17, 17, 192]=>[17, 17, 192] pm:7x1x192x192+192=258240
        serial
          74: conv, [17, 17, 768]=>[17, 17, 192] pm:1x1x768x192+192=147648
          75: conv, [17, 17, 192]=>[17, 17, 192] pm:7x1x192x192+192=258240
          76: conv, [17, 17, 192]=>[17, 17, 192] pm:1x7x192x192+192=258240
          77: conv, [17, 17, 192]=>[17, 17, 192] pm:7x1x192x192+192=258240
          78: conv, [17, 17, 192]=>[17, 17, 192] pm:1x7x192x192+192=258240
        serial
          79: avg, [17, 17, 768]=>[17, 17, 768]
          80: conv, [17, 17, 768]=>[17, 17, 192] pm:1x1x768x192+192=147648
    custom v3_resize2
      parallel
        serial
          81: conv, [17, 17, 768]=>[17, 17, 192] pm:1x1x768x192+192=147648
          82: conv, [17, 17, 192]=>[8, 8, 320] pm:3x3x192x320+320=553280
        serial
          83: conv, [17, 17, 768]=>[17, 17, 192] pm:1x1x768x192+192=147648
          84: conv, [17, 17, 192]=>[17, 17, 192] pm:1x7x192x192+192=258240
          85: conv, [17, 17, 192]=>[17, 17, 192] pm:7x1x192x192+192=258240
          86: conv, [17, 17, 192]=>[8, 8, 192] pm:3x3x192x192+192=331968
        87: max, [17, 17, 768]=>[8, 8, 768]
    custom v3_inception3
      parallel
        88: conv, [8, 8, 1280]=>[8, 8, 320] pm:1x1x1280x320+320=409920
        serial
          89: conv, [8, 8, 1280]=>[8, 8, 384] pm:3x3x1280x384+384=4424064
          parallel
            90: conv, [8, 8, 384]=>[8, 8, 384] pm:1x3x384x384+384=442752
            91: conv, [8, 8, 384]=>[8, 8, 384] pm:3x1x384x384+384=442752
        serial
          92: conv, [8, 8, 1280]=>[8, 8, 448] pm:1x1x1280x448+448=573888
          93: conv, [8, 8, 448]=>[8, 8, 384] pm:3x3x448x384+384=1548672
          parallel
            94: conv, [8, 8, 384]=>[8, 8, 384] pm:1x3x384x384+384=442752
            95: conv, [8, 8, 384]=>[8, 8, 384] pm:3x1x384x384+384=442752
        serial
          96: avg, [8, 8, 1280]=>[8, 8, 1280]
          97: conv, [8, 8, 1280]=>[8, 8, 192] pm:1x1x1280x192+192=245952
    custom v3_inception3
      parallel
        98: conv, [8, 8, 2048]=>[8, 8, 320] pm:1x1x2048x320+320=655680
        serial
          99: conv, [8, 8, 2048]=>[8, 8, 384] pm:3x3x2048x384+384=7078272
          parallel
            100: conv, [8, 8, 384]=>[8, 8, 384] pm:1x3x384x384+384=442752
            101: conv, [8, 8, 384]=>[8, 8, 384] pm:3x1x384x384+384=442752
        serial
          102: conv, [8, 8, 2048]=>[8, 8, 448] pm:1x1x2048x448+448=917952
          103: conv, [8, 8, 448]=>[8, 8, 384] pm:3x3x448x384+384=1548672
          parallel
            104: conv, [8, 8, 384]=>[8, 8, 384] pm:1x3x384x384+384=442752
          105: conv, [8, 8, 384]=>[8, 8, 384] pm:3x1x384x384+384=442752
        serial
         106: avg, [8, 8, 2048]=>[8, 8, 2048]
         107: conv, [8, 8, 2048]=>[8, 8, 192] pm:1x1x2048x192+192=393408
      custom v3_postproc
       serial
        108: avg, [8, 8, 2048]=>[1, 1, 2048]
        109: dropout, [1, 1, 2048]=>[1, 1, 2048]
    110: full, [1, 1, 2048]=>[200] pm:2048x200+200=409800
    Total parameter count: 32401768
```

```
%run ../chap05/dataset_flowers.ipynb

fd = FlowersDataset([96,96], [96,96,3])

CnnExtModel.set_macro('flower_preproc',
    ['serial',
        ['conv', {'ksize':3, 'stride':2, 'chn':6, 'actions':'#act'}]])

CnnExtModel.set_macro('flower_inception1',
    ['parallel',
        ['conv', {'ksize':1, 'chn':4, 'actions':'#act'}],
        ['conv', {'ksize':3, 'chn':6, 'actions':'#act'}],
        ['serial',
            ['conv', {'ksize':3, 'chn':6, 'actions':'#act'}],
            ['conv', {'ksize':3, 'chn':6, 'actions':'#act'}]],
        ['serial',
            ['avg', {'ksize':3, 'stride':1}],
            ['conv', {'ksize':1, 'chn':4, 'actions':'#act'}]]])

CnnExtModel.set_macro('flower_resize',
    ['parallel',
        ['conv', {'ksize':3, 'stride':2, 'chn':12, 'actions':'#act'}],
        ['serial',
            ['conv', {'ksize':3, 'chn':12, 'actions':'#act'}],
            ['conv', {'ksize':3, 'stride':2, 'chn':12, 'actions':'#act'}]],
        ['avg', {'ksize':3, 'stride':2}]])

CnnExtModel.set_macro('flower_inception2',
    ['parallel',
        ['conv', {'ksize':1, 'chn':8, 'actions':'#act'}],
        ['serial',
            ['conv', {'ksize':[3,3], 'chn':8, 'actions':'#act'}],
            ['parallel',
                ['conv', {'ksize':[1,3], 'chn':8, 'actions':'#act'}],
                ['conv', {'ksize':[3,1], 'chn':8, 'actions':'#act'}]]],
        ['serial',
            ['conv', {'ksize':[1,1], 'chn':8, 'actions':'#act'}],
            ['conv', {'ksize':[3,3], 'chn':8, 'actions':'#act'}],
            ['parallel',
                ['conv', {'ksize':[1,3], 'chn':8, 'actions':'#act'}],
                ['conv', {'ksize':[3,1], 'chn':8, 'actions':'#act'}]]],
        ['serial',
            ['avg', {'ksize':3, 'stride':1}],
            ['conv', {'ksize':1, 'chn':8, 'actions':'#act'}]]]])

CnnExtModel.set_macro('flower_postproc',
    ['serial',
        ['avg', {'stride':6}],
        ['dropout', {'keep_prob':0.7}]]])

CnnExtModel.set_macro('inception_flower',
    ['serial',
        ['custom', {'name':'flower_preproc', 'args':{'#act':'#act'}}],
        ['custom', {'name':'flower_inception1', 'args':{'#act':'#act'}}],
        ['custom', {'name':'flower_resize', 'args':{'#act':'#act'}}],
        ['custom', {'name':'flower_inception1', 'args':{'#act':'#act'}}],
        ['custom', {'name':'flower_resize', 'args':{'#act':'#act'}}],
        ['custom', {'name':'flower_inception2', 'args':{'#act':'#act'}}],
        ['custom', {'name':'flower_resize', 'args':{'#act':'#act'}}],
        ['custom', {'name':'flower_inception2', 'args':{'#act':'#act'}}],
        ['custom', {'name':'flower_postproc', 'args':{'#act':'#act'}}]]])

conf_flower_LA = ['custom', {'name':'inception_flower', 'args':{'#act':'LA'}}]
model_flower_LA = CnnExtModel('model_flower_LA', fd,
                         conf_flower_LA, dump_structure=True)
```

```
custom inception_flower
  serial
    custom flower_preproc
      serial
        1: conv, (96, 96, 3)=>[48, 48, 6] pm:3x3x3x6+6=168
    custom flower_inception1
      parallel
        2: conv, [48, 48, 6]=>[48, 48, 4] pm:1x1x6x4+4=28
        3: conv, [48, 48, 6]=>[48, 48, 6] pm:3x3x6x6+6=330
        serial
          4: conv, [48, 48, 6]=>[48, 48, 6] pm:3x3x6x6+6=330
          5: conv, [48, 48, 6]=>[48, 48, 6] pm:3x3x6x6+6=330
        serial
          6: avg, [48, 48, 6]=>[48, 48, 6]
          7: conv, [48, 48, 6]=>[48, 48, 4] pm:1x1x6x4+4=28
    custom flower_resize
      parallel
        8: conv, [48, 48, 20]=>[24, 24, 12] pm:3x3x20x12+12=2172
        serial
          9: conv, [48, 48, 20]=>[48, 48, 12] pm:3x3x20x12+12=2172
          10: conv, [48, 48, 12]=>[24, 24, 12] pm:3x3x12x12+12=1308
        11: avg, [48, 48, 20]=>[24, 24, 20]
    custom flower_inception1
      parallel
        12: conv, [24, 24, 44]=>[24, 24, 4] pm:1x1x44x4+4=180
        13: conv, [24, 24, 44]=>[24, 24, 6] pm:3x3x44x6+6=2382
        serial
          14: conv, [24, 24, 44]=>[24, 24, 6] pm:3x3x44x6+6=2382
          15: conv, [24, 24, 6]=>[24, 24, 6] pm:3x3x6x6+6=330
        serial
          16: avg, [24, 24, 44]=>[24, 24, 44]
          17: conv, [24, 24, 44]=>[24, 24, 4] pm:1x1x44x4+4=180
    custom flower_resize
      parallel
        18: conv, [24, 24, 20]=>[12, 12, 12] pm:3x3x20x12+12=2172
        serial
          19: conv, [24, 24, 20]=>[24, 24, 12] pm:3x3x20x12+12=2172
          20: conv, [24, 24, 12]=>[12, 12, 12] pm:3x3x12x12+12=1308
        21: avg, [24, 24, 20]=>[12, 12, 20]
    custom flower_inception2
      parallel
        22: conv, [12, 12, 44]=>[12, 12, 8] pm:1x1x44x8+8=360
        serial
          23: conv, [12, 12, 44]=>[12, 12, 8] pm:3x3x44x8+8=3176
          parallel
            24: conv, [12, 12, 8]=>[12, 12, 8] pm:1x3x8x8+8=200
            25: conv, [12, 12, 8]=>[12, 12, 8] pm:3x1x8x8+8=200
        serial
          26: conv, [12, 12, 44]=>[12, 12, 8] pm:1x1x44x8+8=360
          27: conv, [12, 12, 8]=>[12, 12, 8] pm:3x3x8x8+8=584
          parallel
            28: conv, [12, 12, 8]=>[12, 12, 8] pm:1x3x8x8+8=200
            29: conv, [12, 12, 8]=>[12, 12, 8] pm:3x1x8x8+8=200
        serial
          30: avg, [12, 12, 44]=>[12, 12, 44]
          31: conv, [12, 12, 44]=>[12, 12, 8] pm:1x1x44x8+8=360
    custom flower_resize
      parallel
        32: conv, [12, 12, 48]=>[6, 6, 12] pm:3x3x48x12+12=5196
        serial
          33: conv, [12, 12, 48]=>[12, 12, 12] pm:3x3x48x12+12=5196
          34: conv, [12, 12, 12]=>[6, 6, 12] pm:3x3x12x12+12=1308
        35: avg, [12, 12, 48]=>[6, 6, 48]
    custom flower_inception2
      parallel
```

```
    36: conv, [6, 6, 72]=>[6, 6, 8] pm:1x1x72x8+8=584
    serial
     37: conv, [6, 6, 72]=>[6, 6, 8] pm:3x3x72x8+8=5192
     parallel
      38: conv, [6, 6, 8]=>[6, 6, 8] pm:1x3x8x8+8=200
      39: conv, [6, 6, 8]=>[6, 6, 8] pm:3x1x8x8+8=200
    serial
     40: conv, [6, 6, 72]=>[6, 6, 8] pm:1x1x72x8+8=584
     41: conv, [6, 6, 8]=>[6, 6, 8] pm:3x3x8x8+8=584
     parallel
      42: conv, [6, 6, 8]=>[6, 6, 8] pm:1x3x8x8+8=200
      43: conv, [6, 6, 8]=>[6, 6, 8] pm:3x1x8x8+8=200
    serial
     44: avg, [6, 6, 72]=>[6, 6, 72]
     45: conv, [6, 6, 72]=>[6, 6, 8] pm:1x1x72x8+8=584
  custom flower_postproc
   serial
    46: avg, [6, 6, 48]=>[1, 1, 48]
    47: dropout, [1, 1, 48]=>[1, 1, 48]
48: full, [1, 1, 48]=>[5] pm:48x5+5=245
Total parameter count: 43885
```

```
model_flower_LA.exec_all(report=2)
```

```
Model model_flower_LA train started:
  Epoch 2: cost=1.601, accuracy=0.238/0.230 (841/841 secs)
  Epoch 4: cost=1.601, accuracy=0.244/0.220 (840/1681 secs)
  Epoch 6: cost=1.601, accuracy=0.244/0.230 (838/2519 secs)
  Epoch 8: cost=1.600, accuracy=0.245/0.220 (847/3366 secs)
  Epoch 10: cost=1.601, accuracy=0.244/0.230 (716/4082 secs)
Model model_flower_LA train ended in 4082 secs:
Model model_flower_LA test report: accuracy = 0.242, (19 secs)

Model model_flower_LA Visualization
```

```
추정확률분포 [18,24,18,17,22] => 추정 dandelion : 정답 dandelion => O
추정확률분포 [18,24,18,17,22] => 추정 dandelion : 정답 rose => X
추정확률분포 [18,24,18,17,22] => 추정 dandelion : 정답 tulip => X
```

```
conf_flower_LAB = ['custom', {'name':'inception_flower', 'args':{'#act':'LAB'}}]
```

```
model_flower_LAB = CnnExtModel('model_flower_LAB', fd,
                                        conf_flower_LAB, dump_structure=False)
model_flower_LAB.exec_all(epoch_count=10, report=2)
Model model_flower_LAB train started:
    Epoch 2: cost=1.529, accuracy=0.308/0.240 (570/570 secs)
    Epoch 4: cost=1.492, accuracy=0.313/0.230 (572/1142 secs)
    Epoch 6: cost=1.552, accuracy=0.294/0.250 (569/1711 secs)
    Epoch 8: cost=1.540, accuracy=0.306/0.230 (571/2282 secs)
    Epoch 10: cost=1.512, accuracy=0.330/0.290 (578/2860 secs)
Model model_flower_LAB train ended in 2860 secs:
Model model_flower_LAB test report: accuracy = 0.306, (22 secs)

Model model_flower_LAB Visualization
```

```
추정확률분포 [ 4,19,12,33,32] => 추정 sunflower : 정답 tulip => X
추정확률분포 [33,39,18, 2, 8] => 추정 dandelion : 정답 daisy => X
추정확률분포 [22,31,16,15,17] => 추정 dandelion : 정답 dandelion => O
```

- ResNet

```
%run ../chap09/cnn_ext_model.ipynb
%run ../chap09/dataset_dummy.ipynb

imagenet = DummyDataset('imagenet', 'select', [224,224,3], 1000)
CnnExtModel.set_macro('p24',
    ['serial',
        ['loop', {'repeat':'#repeat'}, ['conv', {'ksize':3, 'chn':'#chn'}]],
        ['max', {'stride':2}]])

CnnExtModel.set_macro('vgg_19',
    ['serial',
        ['custom', {'name':'p24', 'args':{'#repeat':2, '#chn':64}}],
        ['custom', {'name':'p24', 'args':{'#repeat':2, '#chn':128}}],
        ['custom', {'name':'p24', 'args':{'#repeat':4, '#chn':256}}],
        ['custom', {'name':'p24', 'args':{'#repeat':4, '#chn':512}}],
        ['custom', {'name':'p24', 'args':{'#repeat':4, '#chn':512}}],
        ['loop', {'repeat':2}, ['full', {'width':4096}]]])

vgg19 = CnnExtModel('vgg_19', imagenet,
        ['custom', {'name':'vgg_19'}], dump_structure=True)
```

```
custom vgg_19
  serial
    custom p24
      serial
        loop
          1: conv, [224, 224, 3]=>[224, 224, 64] pm:3x3x3x64+64=1792
          2: conv, [224, 224, 64]=>[224, 224, 64] pm:3x3x64x64+64=36928
        3: max, [224, 224, 64]=>[112, 112, 64]
    custom p24
      serial
        loop
          4: conv, [112, 112, 64]=>[112, 112, 128] pm:3x3x64x128+128=73856
          5: conv, [112, 112, 128]=>[112, 112, 128] pm:3x3x128x128+128=147584
        6: max, [112, 112, 128]=>[56, 56, 128]
    custom p24
      serial
        loop
          7: conv, [56, 56, 128]=>[56, 56, 256] pm:3x3x128x256+256=295168
          8: conv, [56, 56, 256]=>[56, 56, 256] pm:3x3x256x256+256=590080
          9: conv, [56, 56, 256]=>[56, 56, 256] pm:3x3x256x256+256=590080
          10: conv, [56, 56, 256]=>[56, 56, 256] pm:3x3x256x256+256=590080
        11: max, [56, 56, 256]=>[28, 28, 256]
    custom p24
      serial
        loop
          12: conv, [28, 28, 256]=>[28, 28, 512] pm:3x3x256x512+512=1180160
          13: conv, [28, 28, 512]=>[28, 28, 512] pm:3x3x512x512+512=2359808
          14: conv, [28, 28, 512]=>[28, 28, 512] pm:3x3x512x512+512=2359808
          15: conv, [28, 28, 512]=>[28, 28, 512] pm:3x3x512x512+512=2359808
        16: max, [28, 28, 512]=>[14, 14, 512]
    custom p24
      serial
        loop
          17: conv, [14, 14, 512]=>[14, 14, 512] pm:3x3x512x512+512=2359808
          18: conv, [14, 14, 512]=>[14, 14, 512] pm:3x3x512x512+512=2359808
          19: conv, [14, 14, 512]=>[14, 14, 512] pm:3x3x512x512+512=2359808
          20: conv, [14, 14, 512]=>[14, 14, 512] pm:3x3x512x512+512=2359808
        21: max, [14, 14, 512]=>[7, 7, 512]
    loop
      22: full, [7, 7, 512]=>[4096] pm:25088x4096+4096=102764544
      23: full, [4096]=>[4096] pm:4096x4096+4096=16781312
24: full, [4096]=>[1000] pm:4096x1000+1000=4097000
```

```
Total parameter count: 143667240
```

```
CnnExtModel.set_macro('pn',
    ['serial',
        ['conv', {'ksize':3, 'stride':2, 'chn':'#n', 'actions':'#act'}],
        ['loop', {'repeat':'#cnt1'},
                  ['conv', {'ksize':3, 'chn':'#n', 'actions':'#act'}]]])

CnnExtModel.set_macro('plain_34',
    ['serial',
        ['conv', {'ksize':7, 'stride':2, 'chn':64, 'actions':'#act'}],
        ['max', {'stride':2}],
        ['loop', {'repeat':6}, ['conv', {'ksize':3, 'chn':64, 'actions':'#act'}]],
        ['custom', {'name':'pn', 'args':{'#cnt1':7, '#n':128, '#act':'#act'}}],
        ['custom', {'name':'pn', 'args':{'#cnt1':11, '#n':256, '#act':'#act'}}],
        ['custom', {'name':'pn', 'args':{'#cnt1':5, '#n':512, '#act':'#act'}}],
        ['avg', {'stride':7}]])

plain_34 = CnnExtModel('plain_34', imagenet,
       ['custom', {'name':'plain_34', 'args':{'#act':'LA'}}], dump_structure=True)
```

```
custom plain_34
  serial
    1: conv, [224, 224, 3]=>[112, 112, 64] pm:7x7x3x64+64=9472
    2: max, [112, 112, 64]=>[56, 56, 64]
    loop
      3: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
      4: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
      5: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
      6: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
      7: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
      8: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
    custom pn
      serial
        9: conv, [56, 56, 64]=>[28, 28, 128] pm:3x3x64x128+128=73856
        loop
          10: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
          11: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
          12: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
          13: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
          14: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
          15: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
          16: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
    custom pn
      serial
        17: conv, [28, 28, 128]=>[14, 14, 256] pm:3x3x128x256+256=295168
        loop
          18: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
```

```
     19: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
     20: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
     21: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
     22: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
     23: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
     24: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
     25: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
     26: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
     27: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
     28: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
  custom pn
   serial
    29: conv, [14, 14, 256]=>[7, 7, 512] pm:3x3x256x512+512=1180160
    loop
     30: conv, [7, 7, 512]=>[7, 7, 512] pm:3x3x512x512+512=2359808
     31: conv, [7, 7, 512]=>[7, 7, 512] pm:3x3x512x512+512=2359808
     32: conv, [7, 7, 512]=>[7, 7, 512] pm:3x3x512x512+512=2359808
     33: conv, [7, 7, 512]=>[7, 7, 512] pm:3x3x512x512+512=2359808
     34: conv, [7, 7, 512]=>[7, 7, 512] pm:3x3x512x512+512=2359808
  35: avg, [7, 7, 512]=>[1, 1, 512]
36: full, [1, 1, 512]=>[1000] pm:512x1000+1000=513000
Total parameter count: 21616232
```

```
CnnExtModel.set_macro('rf',
    ['add', {'x':True},
        ['serial', ['conv', {'ksize':3, 'chn':'#n', 'actions':'#act'}],
                   ['conv', {'ksize':3, 'chn':'#n', 'actions':'#act'}]]])

CnnExtModel.set_macro('rh',
    ['add', {'x':False},
        ['serial', ['conv', {'ksize':3, 'stride':2, 'chn':'#n', 'actions':'#act'}],
                   ['conv', {'ksize':3, 'chn':'#n', 'actions':'#act'}]],
        ['avg', {'stride':2}]])

CnnExtModel.set_macro('rfull',
    ['serial',
        ['loop', {'repeat':'#cnt'},
                 ['custom', {'name':'rf', 'args':{'#n':'#n', '#act':'#act'}}]]])

CnnExtModel.set_macro('rhalf',
    ['serial',
        ['custom', {'name':'rh', 'args':{'#n':'#n', '#act':'#act'}}],
        ['loop', {'repeat':'#cnt1'},
                 ['custom', {'name':'rf', 'args':{'#n':'#n', '#act':'#act'}}]]])

CnnExtModel.set_macro('residual_34',
    ['serial',
        ['conv', {'ksize':7, 'stride':2, 'chn':64, 'actions':'#act'}],
        ['max', {'stride':2}],
        ['custom', {'name':'rfull', 'args':{'#cnt':3, '#n':64, '#act':'#act'}}],
        ['custom', {'name':'rhalf', 'args':{'#cnt1':3, '#n':128, '#act':'#act'}}],
        ['custom', {'name':'rhalf', 'args':{'#cnt1':5, '#n':256, '#act':'#act'}}],
        ['custom', {'name':'rhalf', 'args':{'#cnt1':2, '#n':512, '#act':'#act'}}],
        ['avg', {'stride':7}]])
```

```
residual_34 = CnnExtModel('residual_34', imagenet,
    ['custom', {'name':'residual_34', 'args':{'#act':'LA'}}], dump_structure=True)
custom residual_34
  serial
    1: conv, [224, 224, 3]=>[112, 112, 64] pm:7x7x3x64+64=9472
    2: max, [112, 112, 64]=>[56, 56, 64]
    custom rfull
      serial
        loop
          custom rf
            add
              serial
                3: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
                4: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
          custom rf
            add
              serial
                5: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
                6: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
          custom rf
            add
              serial
                7: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
                8: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
    custom rhalf
      serial
        custom rh
          add
            serial
              9: conv, [56, 56, 64]=>[28, 28, 128] pm:3x3x64x128+128=73856
              10: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
            11: avg, [56, 56, 64]=>[28, 28, 64]
        loop
          custom rf
            add
              serial
                12: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
                13: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
          custom rf
            add
              serial
                14: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
                15: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
          custom rf
            add
              serial
                16: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
                17: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
    custom rhalf
      serial
        custom rh
          add
            serial
              18: conv, [28, 28, 128]=>[14, 14, 256] pm:3x3x128x256+256=295168
              19: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
            20: avg, [28, 28, 128]=>[14, 14, 128]
        loop
          custom rf
            add
              serial
                21: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
                22: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
          custom rf
            add
              serial
                23: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
                24: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
          custom rf
            add
              serial
                25: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
                26: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
          custom rf
            add
              serial
                27: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
                28: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
          custom rf
            add
              serial
                29: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
                30: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
    custom rhalf
      serial
        custom rh
          add
          serial
           31: conv, [14, 14, 256]=>[7, 7, 512] pm:3x3x256x512+512=1180160
           32: conv, [7, 7, 512]=>[7, 7, 512] pm:3x3x512x512+512=2359808
          33: avg, [14, 14, 256]=>[7, 7, 256]
        loop
         custom rf
          add
           serial
            34: conv, [7, 7, 512]=>[7, 7, 512] pm:3x3x512x512+512=2359808
            35: conv, [7, 7, 512]=>[7, 7, 512] pm:3x3x512x512+512=2359808
         custom rf
          add
           serial
            36: conv, [7, 7, 512]=>[7, 7, 512] pm:3x3x512x512+512=2359808
            37: conv, [7, 7, 512]=>[7, 7, 512] pm:3x3x512x512+512=2359808
      38: avg, [7, 7, 512]=>[1, 1, 512]
    39: full, [1, 1, 512]=>[1000] pm:512x1000+1000=513000
    Total parameter count: 21616232
```

```
CnnExtModel.set_macro('bf',
    ['add', {'x':True},
        ['serial',
            ['conv', {'ksize':1, 'chn':'#n1', 'actions':'#act'}],
            ['conv', {'ksize':3, 'chn':'#n1', 'actions':'#act'}],
            ['conv', {'ksize':1, 'chn':'#n4', 'actions':'#act'}]]])

CnnExtModel.set_macro('bh',
    ['add', {'x':False},
        ['serial',
            ['conv', {'ksize':1, 'stride':2, 'chn':'#n1', 'actions':'#act'}],
            ['conv', {'ksize':3, 'chn':'#n1', 'actions':'#act'}],
            ['conv', {'ksize':1, 'chn':'#n4', 'actions':'#act'}]],
        ['avg', {'stride':2}]])

CnnExtModel.set_macro('bfull',
    ['serial',
        ['loop', {'repeat':'#cnt'},
            ['custom', {'name':'bf', 'args':{'#n1':'#n1', '#n4':'#n4',
                '#act':'#act'}}]]])

CnnExtModel.set_macro('bhalf',
    ['serial',
        ['custom', {'name':'bh', 'args':{'#n1':'#n1', '#n4':'#n4',
                '#act':'#act'}}],
        ['loop', {'repeat':'#cnt1'},
            ['custom', {'name':'bf', 'args':{'#n1':'#n1', '#n4':'#n4',
                '#act':'#act'}}]]])
```

```
CnnExtModel.set_macro('bottleneck_152',
    ['serial',
        ['conv', {'ksize':7, 'stride':2, 'chn':64, 'actions':'#act'}],
        ['max', {'ksize':3, 'stride':2}],
        ['custom', {'name':'bfull','args':{'#cnt':3,'#n1':64,'#n4':256,'#act':'#act'}}],
        ['custom', {'name':'bhalf','args':{'#cnt1':7,'#n1':128,'#n4':512,
            '#act':'#act'}}],
        ['custom', {'name':'bhalf','args':{'#cnt1':35,'#n1':256,'#n4':1024,
            '#act':'#act'}}],
        ['custom', {'name':'bhalf','args':{'#cnt1':2,'#n1':512,'#n4':2048,
            '#act':'#act'}}],
        ['avg', {'stride':7}]])

bottleneck_152 = CnnExtModel('bottleneck_152', imagenet,
        ['custom', {'name':'bottleneck_152', 'args':{'#act':'LAB'}}],
                    dump_structure=True)
```

```
custom bottleneck_152
  serial
    1: conv, [224, 224, 3]=>[112, 112, 64] pm:7x7x3x64+64=9472
    2: max, [112, 112, 64]=>[56, 56, 64]
    custom bfull
      serial
        loop
          custom bf
            add
              serial
                3: conv, [56, 56, 64]=>[56, 56, 64] pm:1x1x64x64+64=4160
                4: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
                5: conv, [56, 56, 64]=>[56, 56, 256] pm:1x1x64x256+256=16640
          custom bf
            add
              serial
                6: conv, [56, 56, 256]=>[56, 56, 64] pm:1x1x256x64+64=16448
                7: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
                8: conv, [56, 56, 64]=>[56, 56, 256] pm:1x1x64x256+256=16640
          custom bf
            add
              serial
                9: conv, [56, 56, 256]=>[56, 56, 64] pm:1x1x256x64+64=16448
                10: conv, [56, 56, 64]=>[56, 56, 64] pm:3x3x64x64+64=36928
                11: conv, [56, 56, 64]=>[56, 56, 256] pm:1x1x64x256+256=16640
    custom bhalf
      serial
        custom bh
          add
            serial
              12: conv, [56, 56, 256]=>[28, 28, 128] pm:1x1x256x128+128=32896
              13: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
              14: conv, [28, 28, 128]=>[28, 28, 512] pm:1x1x128x512+512=66048
            15: avg, [56, 56, 256]=>[28, 28, 256]
        loop
          custom bf
            add
              serial
                16: conv, [28, 28, 512]=>[28, 28, 128] pm:1x1x512x128+128=65664
                17: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
                18: conv, [28, 28, 128]=>[28, 28, 512] pm:1x1x128x512+512=66048
          custom bf
            add
              serial
                19: conv, [28, 28, 512]=>[28, 28, 128] pm:1x1x512x128+128=65664
                20: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
                21: conv, [28, 28, 128]=>[28, 28, 512] pm:1x1x128x512+512=66048
          custom bf
            add
              serial
                22: conv, [28, 28, 512]=>[28, 28, 128] pm:1x1x512x128+128=65664
                23: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
                24: conv, [28, 28, 128]=>[28, 28, 512] pm:1x1x128x512+512=66048
          custom bf
            add
              serial
                25: conv, [28, 28, 512]=>[28, 28, 128] pm:1x1x512x128+128=65664
                26: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
                27: conv, [28, 28, 128]=>[28, 28, 512] pm:1x1x128x512+512=66048
          custom bf
            add
              serial
                28: conv, [28, 28, 512]=>[28, 28, 128] pm:1x1x512x128+128=65664
                29: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
                30: conv, [28, 28, 128]=>[28, 28, 512] pm:1x1x128x512+512=66048
          custom bf
            add
              serial
                31: conv, [28, 28, 512]=>[28, 28, 128] pm:1x1x512x128+128=65664
                32: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
                33: conv, [28, 28, 128]=>[28, 28, 512] pm:1x1x128x512+512=66048
          custom bf
            add
              serial
                34: conv, [28, 28, 512]=>[28, 28, 128] pm:1x1x512x128+128=65664
                35: conv, [28, 28, 128]=>[28, 28, 128] pm:3x3x128x128+128=147584
                36: conv, [28, 28, 128]=>[28, 28, 512] pm:1x1x128x512+512=66048
```

```
custom bhalf
  serial
    custom bh
      add
        serial
          37: conv, [28, 28, 512]=>[14, 14, 256] pm:1x1x512x256+256=131328
          38: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
          39: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
        40: avg, [28, 28, 512]=>[14, 14, 512]
    loop
      custom bf
        add
          serial
            41: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
            42: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
            43: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
      custom bf
        add
          serial
            44: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
            45: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
            46: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
      custom bf
        add
          serial
            47: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
            48: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
            49: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
      custom bf
        add
          serial
            50: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
            51: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
            52: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
      custom bf
        add
          serial
            53: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
            54: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
            55: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
      custom bf
        add
          serial
            56: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
            57: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
            58: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
      custom bf
  add
    serial
      59: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      60: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      61: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      62: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      63: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      64: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      65: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      66: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      67: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      68: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      69: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      70: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      71: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      72: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      73: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      74: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      75: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      76: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      77: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      78: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      79: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      80: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      81: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      82: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
```

```
custom bf
  add
    serial
      83: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      84: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      85: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      86: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      87: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      88: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      89: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      90: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      91: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      92: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      93: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      94: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      95: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      96: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      97: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      98: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      99: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      100: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      101: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      102: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      103: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      104: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      105: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      106: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      107: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      108: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      109: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      110: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      111: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      112: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      113: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      114: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      115: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      116: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      117: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      118: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      119: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      120: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      121: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      122: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      123: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      124: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      125: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
      126: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
      127: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bf
  add
    serial
      128: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
```

```
            129: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
            130: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
      custom bf
        add
          serial
            131: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
            132: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
            133: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
      custom bf
        add
          serial
            134: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
            135: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
            136: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
      custom bf
        add
          serial
            137: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
            138: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
            139: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
      custom bf
        add
          serial
            140: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
            141: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
            142: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
      custom bf
        add
          serial
            143: conv, [14, 14, 1024]=>[14, 14, 256] pm:1x1x1024x256+256=262400
            144: conv, [14, 14, 256]=>[14, 14, 256] pm:3x3x256x256+256=590080
            145: conv, [14, 14, 256]=>[14, 14, 1024] pm:1x1x256x1024+1024=263168
custom bhalf
  serial
    custom bh
      add
        serial
          146: conv, [14, 14, 1024]=>[7, 7, 512] pm:1x1x1024x512+512=524800
          147: conv, [7, 7, 512]=>[7, 7, 512] pm:3x3x512x512+512=2359808
          148: conv, [7, 7, 512]=>[7, 7, 2048] pm:1x1x512x2048+2048=1050624
        149: avg, [14, 14, 1024]=>[7, 7, 1024]
    loop
      custom bf
        add
          serial
            150: conv, [7, 7, 2048]=>[7, 7, 512] pm:1x1x2048x512+512=1049088
            151: conv, [7, 7, 512]=>[7, 7, 512] pm:3x3x512x512+512=2359808
          152: conv, [7, 7, 512]=>[7, 7, 2048] pm:1x1x512x2048+2048=1050624
     custom bf
      add
        serial
          153: conv, [7, 7, 2048]=>[7, 7, 512] pm:1x1x2048x512+512=1049088
          154: conv, [7, 7, 512]=>[7, 7, 512] pm:3x3x512x512+512=2359808
          155: conv, [7, 7, 512]=>[7, 7, 2048] pm:1x1x512x2048+2048=1050624
  156: avg, [7, 7, 2048]=>[1, 1, 2048]
157: full, [1, 1, 2048]=>[1000] pm:2048x1000+1000=2049000
Total parameter count: 57344360
```

In [6]:

```
%run ../chap05/dataset_flowers.ipynb

fd = FlowersDataset([64,64], [64,64,3])
```

In [7]:

```
CnnExtModel.set_macro('plain_flower',
    ['serial',
        ['conv', {'ksize':7, 'stride':2, 'chn':16, 'actions':'#act'}],
        ['max', {'stride':2}],
        ['loop', {'repeat':4}, ['conv', {'ksize':3, 'chn':16, 'actions':'#act'}]],
        ['custom', {'name':'pn', 'args':{'#cnt1':3, '#n':32, '#act':'#act'}}],
        ['custom', {'name':'pn', 'args':{'#cnt1':3, '#n':64, '#act':'#act'}}],
        ['avg', {'stride':4}]])

plain_flower = CnnExtModel('plain_flower', fd,
        ['custom', {'name':'plain_flower', 'args':{'#act':'LAB'}}],
                    dump_structure=True)
plain_flower.exec_all(epoch_count=10, report=2)
```

```
custom plain_flower
  serial
    1: conv, (64, 64, 3)=>[32, 32, 16] pm:7x7x3x16+16=2368
    2: max, [32, 32, 16]=>[16, 16, 16]
    loop
      3: conv, [16, 16, 16]=>[16, 16, 16] pm:3x3x16x16+16=2320
      4: conv, [16, 16, 16]=>[16, 16, 16] pm:3x3x16x16+16=2320
      5: conv, [16, 16, 16]=>[16, 16, 16] pm:3x3x16x16+16=2320
      6: conv, [16, 16, 16]=>[16, 16, 16] pm:3x3x16x16+16=2320
    custom pn
      serial
```

```
        7: conv, [16, 16, 16]=>[8, 8, 32] pm:3x3x16x32+32=4640
        loop
          8: conv, [8, 8, 32]=>[8, 8, 32] pm:3x3x32x32+32=9248
          9: conv, [8, 8, 32]=>[8, 8, 32] pm:3x3x32x32+32=9248
          10: conv, [8, 8, 32]=>[8, 8, 32] pm:3x3x32x32+32=9248
    custom pn
      serial
        11: conv, [8, 8, 32]=>[4, 4, 64] pm:3x3x32x64+64=18496
        loop
          12: conv, [4, 4, 64]=>[4, 4, 64] pm:3x3x64x64+64=36928
          13: conv, [4, 4, 64]=>[4, 4, 64] pm:3x3x64x64+64=36928
          14: conv, [4, 4, 64]=>[4, 4, 64] pm:3x3x64x64+64=36928
    15: avg, [4, 4, 64]=>[1, 1, 64]
16: full, [1, 1, 64]=>[5] pm:64x5+5=325
Total parameter count: 173637
Model plain_flower train started:
    Epoch 2: cost=1.375, accuracy=0.380/0.360 (243/243 secs)
    Epoch 4: cost=1.393, accuracy=0.390/0.260 (241/484 secs)
    Epoch 6: cost=1.325, accuracy=0.406/0.360 (245/729 secs)
    Epoch 8: cost=1.311, accuracy=0.415/0.350 (246/975 secs)
    Epoch 10: cost=1.305, accuracy=0.410/0.440 (243/1218 secs)
Model plain_flower train ended in 1218 secs:
Model plain_flower test report: accuracy = 0.358, (7 secs)

Model plain_flower Visualization
```

```
추정확률분포 [ 4,96, 0, 0, 0] => 추정 dandelion : 정답 rose => X
추정확률분포 [26,33,17, 9,15] => 추정 dandelion : 정답 tulip => X
추정확률분포 [ 0, 0,72, 0,28] => 추정 rose : 정답 tulip => X
```

In [8]:

```
CnnExtModel.set_macro('residual_flower',
    ['serial',
        ['conv', {'ksize':7, 'stride':2, 'chn':16, 'actions':'#act'}],
        ['max', {'stride':2}],
        ['custom', {'name':'rfull', 'args':{'#cnt':2, '#n':16, '#act':'#act'}}],
        ['custom', {'name':'rhalf', 'args':{'#cnt1':1, '#n':32, '#act':'#act'}}],
            ['custom', {'name':'rhalf', 'args':{'#cnt1':1, '#n':64, '#act':'#act'}}],
            ['avg', {'stride':4}]])

residual_flower = CnnExtModel('residual_flower', fd,
          ['custom', {'name':'residual_flower', 'args':{'#act':'LAB'}}],
                          dump_structure=True)
residual_flower.exec_all(epoch_count=10, report=2)
```

```
custom residual_flower
 serial
  1: conv, (64, 64, 3)=>[32, 32, 16] pm:7x7x3x16+16=2368
  2: max, [32, 32, 16]=>[16, 16, 16]
  custom rfull
   serial
    loop
     custom rf
      add
       serial
        3: conv, [16, 16, 16]=>[16, 16, 16] pm:3x3x16x16+16=2320
        4: conv, [16, 16, 16]=>[16, 16, 16] pm:3x3x16x16+16=2320
     custom rf
      add
       serial
        5: conv, [16, 16, 16]=>[16, 16, 16] pm:3x3x16x16+16=2320
        6: conv, [16, 16, 16]=>[16, 16, 16] pm:3x3x16x16+16=2320
  custom rhalf
   serial
    custom rh
     add
      serial
       7: conv, [16, 16, 16]=>[8, 8, 32] pm:3x3x16x32+32=4640
       8: conv, [8, 8, 32]=>[8, 8, 32] pm:3x3x32x32+32=9248
      9: avg, [16, 16, 16]=>[8, 8, 16]
    loop
     custom rf
      add
       serial
        10: conv, [8, 8, 32]=>[8, 8, 32] pm:3x3x32x32+32=9248
        11: conv, [8, 8, 32]=>[8, 8, 32] pm:3x3x32x32+32=9248
  custom rhalf
   serial
    custom rh
     add
      serial
       12: conv, [8, 8, 32]=>[4, 4, 64] pm:3x3x32x64+64=18496
       13: conv, [4, 4, 64]=>[4, 4, 64] pm:3x3x64x64+64=36928
      14: avg, [8, 8, 32]=>[4, 4, 32]
    loop
```

```
          custom rf
            add
              serial
                15: conv, [4, 4, 64]=>[4, 4, 64] pm:3x3x64x64+64=36928
                16: conv, [4, 4, 64]=>[4, 4, 64] pm:3x3x64x64+64=36928
    17: avg, [4, 4, 64]=>[1, 1, 64]
18: full, [1, 1, 64]=>[5] pm:64x5+5=325
Total parameter count: 173637
Model residual_flower train started:
    Epoch 2: cost=1.257, accuracy=0.473/0.250 (239/239 secs)
    Epoch 4: cost=1.174, accuracy=0.521/0.230 (244/483 secs)
    Epoch 6: cost=1.103, accuracy=0.554/0.420 (241/724 secs)
    Epoch 8: cost=1.031, accuracy=0.590/0.490 (245/969 secs)
    Epoch 10: cost=0.986, accuracy=0.608/0.500 (247/1216 secs)
Model residual_flower train ended in 1216 secs:
Model residual_flower test report: accuracy = 0.463, (6 secs)

Model residual_flower Visualization
```

```
추정확률분포 [ 3,59, 7,21,10] => 추정 dandelion : 정답 dandelion => O
추정확률분포 [ 0, 0,30, 0,69] => 추정 tulip : 정답 rose => X
추정확률분포 [ 0,83, 1, 9, 8] => 추정 dandelion : 정답 sunflower => X
```

```
CnnExtModel.set_macro('bottleneck_flower',
    ['serial',
        ['conv', {'ksize':7, 'stride':2, 'chn':16, 'actions':'#act'}],
        ['max', {'ksize':3, 'stride':2}],
        ['custom', {'name':'bfull', 'args':{'#cnt':1,'#n1':16,'#n4': 64,
            '#act':'#act'}}],
        ['custom', {'name':'bhalf', 'args':{'#cnt1':2,'#n1':32,'#n4':128
            '#act':'#act'}}],
        ['custom', {'name':'bhalf', 'args':{'#cnt1':1,'#n1':64,'#n4':256,
            '#act':'#act'}}],
        ['avg', {'stride':4}]])

bottleneck_flower = CnnExtModel('bottleneck_flower', fd,
    ['custom', {'name':'bottleneck_flower', 'args':{'#act':'LAB'}}],
        dump_structure=True)
bottleneck_flower.exec_all(epoch_count=10, report=2)
```

```
custom bottleneck_flower
```

```
serial
  1: conv, (64, 64, 3)=>[32, 32, 16] pm:7x7x3x16+16=2368
  2: max, [32, 32, 16]=>[16, 16, 16]
  custom bfull
    serial
      loop
        custom bf
          add
            serial
              3: conv, [16, 16, 16]=>[16, 16, 16] pm:1x1x16x16+16=272
              4: conv, [16, 16, 16]=>[16, 16, 16] pm:3x3x16x16+16=2320
              5: conv, [16, 16, 16]=>[16, 16, 64] pm:1x1x16x64+64=1088
  custom bhalf
    serial
      custom bh
        add
          serial
            6: conv, [16, 16, 64]=>[8, 8, 32] pm:1x1x64x32+32=2080
            7: conv, [8, 8, 32]=>[8, 8, 32] pm:3x3x32x32+32=9248
            8: conv, [8, 8, 32]=>[8, 8, 128] pm:1x1x32x128+128=4224
          9: avg, [16, 16, 64]=>[8, 8, 64]
      loop
        custom bf
          add
            serial
              10: conv, [8, 8, 128]=>[8, 8, 32] pm:1x1x128x32+32=4128
              11: conv, [8, 8, 32]=>[8, 8, 32] pm:3x3x32x32+32=9248
              12: conv, [8, 8, 32]=>[8, 8, 128] pm:1x1x32x128+128=4224
        custom bf
          add
            serial
              13: conv, [8, 8, 128]=>[8, 8, 32] pm:1x1x128x32+32=4128
              14: conv, [8, 8, 32]=>[8, 8, 32] pm:3x3x32x32+32=9248
              15: conv, [8, 8, 32]=>[8, 8, 128] pm:1x1x32x128+128=4224
  custom bhalf
    serial
      custom bh
        add
          serial
            16: conv, [8, 8, 128]=>[4, 4, 64] pm:1x1x128x64+64=8256
            17: conv, [4, 4, 64]=>[4, 4, 64] pm:3x3x64x64+64=36928
            18: conv, [4, 4, 64]=>[4, 4, 256] pm:1x1x64x256+256=16640
          19: avg, [8, 8, 128]=>[4, 4, 128]
      loop
        custom bf
          add
            serial
```

```
             20: conv, [4, 4, 256]=>[4, 4, 64] pm:1x1x256x64+64=16448
             21: conv, [4, 4, 64]=>[4, 4, 64] pm:3x3x64x64+64=36928
             22: conv, [4, 4, 64]=>[4, 4, 256] pm:1x1x64x256+256=16640
   23: avg, [4, 4, 256]=>[1, 1, 256]
24: full, [1, 1, 256]=>[5] pm:256x5+5=1285
Total parameter count: 189925
Model bottleneck_flower train started:
   Epoch 2: cost=1.221, accuracy=0.502/0.300 (279/279 secs)
   Epoch 4: cost=1.080, accuracy=0.578/0.550 (281/560 secs)
   Epoch 6: cost=0.985, accuracy=0.609/0.430 (278/838 secs)
   Epoch 8: cost=0.956, accuracy=0.634/0.350 (280/1118 secs)
   Epoch 10: cost=0.874, accuracy=0.664/0.410 (285/1403 secs)
Model bottleneck_flower train ended in 1403 secs:
Model bottleneck_flower test report: accuracy = 0.420, (8 secs)

Model bottleneck_flower Visualization
```

```
추정확률분포 [ 1,96, 2, 1, 1] => 추정 dandelion : 정답 dandelion => O
추정확률분포 [ 0,98, 0, 2, 0] => 추정 dandelion : 정답 sunflower => X
추정확률분포 [ 1,21,24, 9,44] => 추정 tulip : 정답 rose => X
```

[결론 및 정리]

Inception Model 및 ResNet Model을 적용하여 진행과정을 살펴보면서 학습 과정을 파악할 수 있었다.

Dataset이 부족하기 때문에 Inception Model 이나 ResNet Model을 적용하였을 때에도 익히 알려진 것과는 다르게 좋은 성능이 나타나지 않았으며 50% 언저리에서 별볼일 없는 성능을 보이는 것을 확인할 수 있었다.

### - Bottle neck의 장단점

1 x 1 convolution이면 convolution이 의미가 없다고 생각할 수 있으나, filter 수를 조절하기 위해서 사용된다.

입력하는 채널의 수와 출력하는 채널의 수가 완전히 동일하다면 convolution은 큰 의미가 없으나, 이 과정에서 차원의 수를 바꿔줄 수 있다. 한 예로 차원을 축소시켜준다면 연산 량을 크게 줄일 수 있어 뒤로 갈수록 **연산 량의 수와 파라미터 수가 줄어들게 되어 같은 computing, 시간으로 더 Deep한 Network를 설계하고 학습할 수 있다**. Bottleneck이라는 구조를 활용하여 다음 Layer에서 학습할 만한 양을 남겨둔 적당한 차원으로 만들어준다.

3x3, 5x5 convolution을 하기 전에 1x1 convolution을 수행하여 차원의 수를 감소시킨 후에 3x3, 5x5를 수행하여 앞서 말했던 computing, 시간 면에서 이득을 보면서 더 Deep한 Network를 설계하였다. 이를 Inception Model에 사용하여 파라미터 수는 줄이고 Layer를 더 깊게하여 Resnet에서 눈에 띄는 성과를 나타내었다.

가장 큰 단점은 **Input과 Output의 차원을 동일하게 만들어줘야 한다는 점**이다.

### - Resnet과 Inception Model의 차이점

Inception Model은 Bottleneck 구조를 통해서 Parameter를 획기적으로 줄인 모델이다. 같은 receptive field를 만들지 않기 위해 여러 갈래길을 이용한다.

ResNet : short-cut을 이용한다. (output에 Input을 더해줌으로써 최종 출력단에서 얻어내는 receptive field가 다양하게 형성된다.