

=====

The summary of the 4-layer CNN network 4Conv2FC

=====

```
Classifier_1d_4c_2fc_sub_qr(
(raw): Sequential(
  (0): MaxPool2d(kernel_size=1, stride=2, padding=0, dilation=1, ceil_mode=False)
  (1): SepConv1d_v4(
    (layers): Sequential(
      (0): Conv2d(2, 2, kernel_size=(1, 8), stride=(1, 4), padding=(0, 3), groups=2)
      (1): Conv2d(2, 32, kernel_size=(1, 1), stride=(1, 1))
      (2): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (3): ReLU(inplace=True)
      (4): Dropout(p=0.5, inplace=False)
    )
  )
  (2): SepConv1d_v4(
    (layers): Sequential(
      (0): Conv2d(32, 32, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=32)
      (1): Conv2d(32, 64, kernel_size=(1, 1), stride=(1, 1))
      (2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (3): ReLU(inplace=True)
      (4): Dropout(p=0.5, inplace=False)
    )
  )
  (3): SepConv1d_v4(
    (layers): Sequential(
      (0): Conv2d(64, 64, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=64)
      (1): Conv2d(64, 128, kernel_size=(1, 1), stride=(1, 1))
      (2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (3): ReLU(inplace=True)
      (4): Dropout(p=0.5, inplace=False)
    )
  )
  (4): SepConv1d_v4(
    (layers): Sequential(
      (0): Conv2d(128, 128, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=128)
      (1): Conv2d(128, 256, kernel_size=(1, 1), stride=(1, 1))
      (2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (3): ReLU(inplace=True)
      (4): Dropout(p=0.5, inplace=False)
    )
  )
)
(FC): Sequential(
  (0): Flatten()
  (1): Linear(in_features=1024, out_features=128, bias=True)
  (2): ReLU(inplace=True)
  (3): Dropout(p=0.5, inplace=False)
)
(out): Sequential(
  (0): Linear(in_features=128, out_features=2, bias=True)
)
(quant): QuantStub()
(dequant): DeQuantStub()
)
```

=====

Table of the network parameters: 4Conv2FC

=====

Layer (type)	Output Shape	Param #
QuantStub-1	[512, 2, 2048]	0
MaxPool2d-2	[512, 2, 1, 1024]	0
Conv2d-3	[512, 2, 1, 256]	18
Conv2d-4	[512, 32, 1, 256]	96
BatchNorm2d-5	[512, 32, 1, 256]	64
ReLU-6	[512, 32, 1, 256]	0
Dropout-7	[512, 32, 1, 256]	0
SepConv1d_v4-8	[512, 32, 1, 256]	0
Conv2d-9	[512, 32, 1, 64]	288
Conv2d-10	[512, 64, 1, 64]	2,112
BatchNorm2d-11	[512, 64, 1, 64]	128
ReLU-12	[512, 64, 1, 64]	0
Dropout-13	[512, 64, 1, 64]	0
SepConv1d_v4-14	[512, 64, 1, 64]	0
Conv2d-15	[512, 64, 1, 16]	576
Conv2d-16	[512, 128, 1, 16]	8,320
BatchNorm2d-17	[512, 128, 1, 16]	256
ReLU-18	[512, 128, 1, 16]	0
Dropout-19	[512, 128, 1, 16]	0
SepConv1d_v4-20	[512, 128, 1, 16]	0
Conv2d-21	[512, 128, 1, 4]	1,152
Conv2d-22	[512, 256, 1, 4]	33,024
BatchNorm2d-23	[512, 256, 1, 4]	512
ReLU-24	[512, 256, 1, 4]	0
Dropout-25	[512, 256, 1, 4]	0
SepConv1d_v4-26	[512, 256, 1, 4]	0
Flatten-27	[512, 1024]	0
Linear-28	[512, 128]	131,200
ReLU-29	[512, 128]	0
Dropout-30	[512, 128]	0
Linear-31	[512, 2]	258
DeQuantStub-32	[512, 2]	0

```
=====
Total params: 178,004
Trainable params: 178,004
Non-trainable params: 0
-----
```

```
Input size (MB): 8.00
Forward/backward pass size (MB): 345.52
Params size (MB): 0.68
Estimated Total Size (MB): 354.19
-----
```

```
=====
Detail of the network's per layer computations and parameters:      4Conv2FC
=====
```

```
Classifier_ld_4c_2fc_sub_qr(
  0.178 M, 100.000% Params, 0.643 MMac, 100.000% MACs,
  (raw): Sequential(
    0.047 M, 26.149% Params, 0.512 MMac, 79.570% MACs,
    (0): MaxPool2d(0.0 M, 0.000% Params, 0.004 MMac, 0.637% MACs, kernel_size=1, stride=2, padding=0, dilation=1, ceil_mode=False)
    (1): SepConv1d_v4(
      0.0 M, 0.100% Params, 0.054 MMac, 8.355% MACs,
      (layers): Sequential(
        0.0 M, 0.100% Params, 0.054 MMac, 8.355% MACs,
        (0): Conv2d(0.0 M, 0.010% Params, 0.005 MMac, 0.716% MACs, 2, 2, kernel_size=(1, 8), stride=(1, 4), padding=(0, 3), groups=2)
        (1): Conv2d(0.0 M, 0.054% Params, 0.025 MMac, 3.819% MACs, 2, 32, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(0.0 M, 0.036% Params, 0.016 MMac, 2.546% MACs, 32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (3): ReLU(0.0 M, 0.000% Params, 0.008 MMac, 1.273% MACs, inplace=True)
        (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
      )
    )
    (2): SepConv1d_v4(
      0.003 M, 1.420% Params, 0.166 MMac, 25.781% MACs,
      (layers): Sequential(
        0.003 M, 1.420% Params, 0.166 MMac, 25.781% MACs,
        (0): Conv2d(0.0 M, 0.162% Params, 0.018 MMac, 2.865% MACs, 32, 32, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),
groups=32)
        (1): Conv2d(0.002 M, 1.186% Params, 0.135 MMac, 21.007% MACs, 32, 64, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(0.0 M, 0.072% Params, 0.008 MMac, 1.273% MACs, 64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (3): ReLU(0.0 M, 0.000% Params, 0.004 MMac, 0.637% MACs, inplace=True)
        (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
      )
    )
    (3): SepConv1d_v4(
      0.009 M, 5.141% Params, 0.148 MMac, 23.075% MACs,
      (layers): Sequential(
        0.009 M, 5.141% Params, 0.148 MMac, 23.075% MACs,
        (0): Conv2d(0.001 M, 0.324% Params, 0.009 MMac, 1.432% MACs, 64, 64, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),
groups=64)
        (1): Conv2d(0.008 M, 4.674% Params, 0.133 MMac, 20.688% MACs, 64, 128, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(0.0 M, 0.144% Params, 0.004 MMac, 0.637% MACs, 128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (3): ReLU(0.0 M, 0.000% Params, 0.002 MMac, 0.318% MACs, inplace=True)
        (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
      )
    )
    (4): SepConv1d_v4(
      0.035 M, 19.487% Params, 0.14 MMac, 21.723% MACs,
      (layers): Sequential(
        0.035 M, 19.487% Params, 0.14 MMac, 21.723% MACs,
        (0): Conv2d(0.001 M, 0.647% Params, 0.005 MMac, 0.716% MACs, 128, 128, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),
groups=128)
        (1): Conv2d(0.033 M, 18.552% Params, 0.132 MMac, 20.529% MACs, 128, 256, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(0.001 M, 0.288% Params, 0.002 MMac, 0.318% MACs, 256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (3): ReLU(0.0 M, 0.000% Params, 0.001 MMac, 0.159% MACs, inplace=True)
        (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
      )
    )
  )
  (FC): Sequential(
    0.131 M, 73.706% Params, 0.131 MMac, 20.390% MACs,
    (0): Flatten(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )
    (1): Linear(0.131 M, 73.706% Params, 0.131 MMac, 20.370% MACs, in_features=1024, out_features=128, bias=True)
    (2): ReLU(0.0 M, 0.000% Params, 0.0 MMac, 0.020% MACs, inplace=True)
    (3): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
  )
  (out): Sequential(
    0.0 M, 0.145% Params, 0.0 MMac, 0.040% MACs,
    (0): Linear(0.0 M, 0.145% Params, 0.0 MMac, 0.040% MACs, in_features=128, out_features=2, bias=True)
  )
  (quant): QuantStub(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )
  (dequant): DeQuantStub(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )
)
```

```
=====
The summary of the 4-layer CNN network with conv. pruning:
model: 4Conv2FC-conv-pruned      Conv-channels: 706 to 421
=====
```

```
Classifier_ld_4c_2fc_sub_qr(
  (raw): Sequential(
    (0): MaxPool2d(kernel_size=1, stride=2, padding=0, dilation=1, ceil_mode=False)
    (1): SepConv1d_v4(
      (layers): Sequential(
        (0): Conv2d(2, 2, kernel_size=(1, 8), stride=(1, 4), padding=(0, 3), groups=2)
        (1): Conv2d(2, 29, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(29, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
  )
)
```

```
(3): ReLU(inplace=True)
(4): Dropout(p=0.5, inplace=False)
)
)
(2): SepConv1d_v4(
(layers): Sequential(
(0): Conv2d(29, 29, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=29)
(1): Conv2d(29, 32, kernel_size=(1, 1), stride=(1, 1))
(2): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(3): ReLU(inplace=True)
(4): Dropout(p=0.5, inplace=False)
)
)
(3): SepConv1d_v4(
(layers): Sequential(
(0): Conv2d(32, 32, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=32)
(1): Conv2d(32, 79, kernel_size=(1, 1), stride=(1, 1))
(2): BatchNorm2d(79, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(3): ReLU(inplace=True)
(4): Dropout(p=0.5, inplace=False)
)
)
(4): SepConv1d_v4(
(layers): Sequential(
(0): Conv2d(79, 79, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=79)
(1): Conv2d(79, 139, kernel_size=(1, 1), stride=(1, 1))
(2): BatchNorm2d(139, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(3): ReLU(inplace=True)
(4): Dropout(p=0.5, inplace=False)
)
)
)
(FC): Sequential(
(0): Flatten()
(1): Linear(in_features=556, out_features=128, bias=True)
(2): ReLU(inplace=True)
(3): Dropout(p=0.5, inplace=False)
)
(out): Sequential(
(0): Linear(in_features=128, out_features=2, bias=True)
)
(quant): QuantStub()
(dequant): DeQuantStub()
)
```

Layer (type)	Output Shape	Param #
QuantStub-1	[512, 2, 2048]	0
MaxPool2d-2	[512, 2, 1, 1024]	0
Conv2d-3	[512, 2, 1, 256]	18
Conv2d-4	[512, 29, 1, 256]	87
BatchNorm2d-5	[512, 29, 1, 256]	58
ReLU-6	[512, 29, 1, 256]	0
Dropout-7	[512, 29, 1, 256]	0
SepConv1d_v4-8	[512, 29, 1, 256]	0
Conv2d-9	[512, 29, 1, 64]	261
Conv2d-10	[512, 32, 1, 64]	960
BatchNorm2d-11	[512, 32, 1, 64]	64
ReLU-12	[512, 32, 1, 64]	0
Dropout-13	[512, 32, 1, 64]	0
SepConv1d_v4-14	[512, 32, 1, 64]	0
Conv2d-15	[512, 32, 1, 16]	288
Conv2d-16	[512, 79, 1, 16]	2,607
BatchNorm2d-17	[512, 79, 1, 16]	158
ReLU-18	[512, 79, 1, 16]	0
Dropout-19	[512, 79, 1, 16]	0
SepConv1d_v4-20	[512, 79, 1, 16]	0
Conv2d-21	[512, 79, 1, 4]	711
Conv2d-22	[512, 139, 1, 4]	11,120
BatchNorm2d-23	[512, 139, 1, 4]	278
ReLU-24	[512, 139, 1, 4]	0
Dropout-25	[512, 139, 1, 4]	0
SepConv1d_v4-26	[512, 139, 1, 4]	0
Flatten-27	[512, 556]	0
Linear-28	[512, 128]	71,296
ReLU-29	[512, 128]	0
Dropout-30	[512, 128]	0
Linear-31	[512, 2]	258
DeQuantStub-32	[512, 2]	0

Detail of the network's per layer computations and parameters:
model: 4Conv2FC-conv-pruned Conv-channels: 706 to 421

```
=====  
Classifier_ld_4c_2fc_sub_qr(  
  0.088 M, 100.000% Params, 0.308 MMac, 100.000% MACs,  
  (raw): Sequential(  
    0.017 M, 18.840% Params, 0.237 MMac, 76.783% MACs,  
    (0): MaxPool2d(0.0 M, 0.000% Params, 0.004 MMac, 1.329% MACs, kernel_size=1, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (1): SepConv1d_v4(  
      0.0 M, 0.185% Params, 0.049 MMac, 15.948% MACs,  
      (layers): Sequential(  
        0.0 M, 0.185% Params, 0.049 MMac, 15.948% MACs,  
        (0): Conv2d(0.0 M, 0.020% Params, 0.005 MMac, 1.495% MACs, 2, 2, kernel_size=(1, 8), stride=(1, 4), padding=(0, 3), groups=2)  
        (1): Conv2d(0.0 M, 0.099% Params, 0.022 MMac, 7.227% MACs, 2, 29, kernel_size=(1, 1), stride=(1, 1))  
        (2): BatchNorm2d(0.0 M, 0.066% Params, 0.015 MMac, 4.818% MACs, 29, eps=1e-05, momentum=0.1, affine=True,  
track_running_stats=True)  
        (3): ReLU(0.0 M, 0.000% Params, 0.007 MMac, 2.409% MACs, inplace=True)  
        (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)  
      )  
    )  
    (2): SepConv1d_v4(  
      0.001 M, 1.458% Params, 0.084 MMac, 27.349% MACs,  
      (layers): Sequential(  
        0.001 M, 1.458% Params, 0.084 MMac, 27.349% MACs,  
        (0): Conv2d(0.0 M, 0.296% Params, 0.017 MMac, 5.420% MACs, 29, 29, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),  
groups=29)  
        (1): Conv2d(0.001 M, 1.089% Params, 0.061 MMac, 19.936% MACs, 29, 32, kernel_size=(1, 1), stride=(1, 1))  
        (2): BatchNorm2d(0.0 M, 0.073% Params, 0.004 MMac, 1.329% MACs, 32, eps=1e-05, momentum=0.1, affine=True,  
track_running_stats=True)  
        (3): ReLU(0.0 M, 0.000% Params, 0.002 MMac, 0.665% MACs, inplace=True)  
        (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)  
      )  
    )  
    (3): SepConv1d_v4(  
      0.003 M, 3.463% Params, 0.05 MMac, 16.260% MACs,  
      (layers): Sequential(  
        0.003 M, 3.463% Params, 0.05 MMac, 16.260% MACs,  
        (0): Conv2d(0.0 M, 0.327% Params, 0.005 MMac, 1.495% MACs, 32, 32, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),  
groups=32)  
        (1): Conv2d(0.003 M, 2.957% Params, 0.042 MMac, 13.534% MACs, 32, 79, kernel_size=(1, 1), stride=(1, 1))  
        (2): BatchNorm2d(0.0 M, 0.179% Params, 0.003 MMac, 0.820% MACs, 79, eps=1e-05, momentum=0.1, affine=True,  
track_running_stats=True)  
        (3): ReLU(0.0 M, 0.000% Params, 0.001 MMac, 0.410% MACs, inplace=True)  
        (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)  
      )  
    )  
    (4): SepConv1d_v4(  
      0.012 M, 13.735% Params, 0.049 MMac, 15.897% MACs,  
      (layers): Sequential(  
        0.012 M, 13.735% Params, 0.049 MMac, 15.897% MACs,  
        (0): Conv2d(0.001 M, 0.806% Params, 0.003 MMac, 0.923% MACs, 79, 79, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),  
groups=79)  
        (1): Conv2d(0.011 M, 12.613% Params, 0.044 MMac, 14.433% MACs, 79, 139, kernel_size=(1, 1), stride=(1, 1))  
        (2): BatchNorm2d(0.0 M, 0.315% Params, 0.001 MMac, 0.361% MACs, 139, eps=1e-05, momentum=0.1, affine=True,  
track_running_stats=True)  
        (3): ReLU(0.0 M, 0.000% Params, 0.001 MMac, 0.180% MACs, inplace=True)  
        (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)  
      )  
    )  
  )  
  (FC): Sequential(  
    0.071 M, 80.867% Params, 0.071 MMac, 23.134% MACs,  
    (0): Flatten(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )  
    (1): Linear(0.071 M, 80.867% Params, 0.071 MMac, 23.092% MACs, in_features=556, out_features=128, bias=True)  
    (2): ReLU(0.0 M, 0.000% Params, 0.0 MMac, 0.042% MACs, inplace=True)  
    (3): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)  
  )  
  (out): Sequential(  
    0.0 M, 0.293% Params, 0.0 MMac, 0.083% MACs,  
    (0): Linear(0.0 M, 0.293% Params, 0.0 MMac, 0.083% MACs, in_features=128, out_features=2, bias=True)  
  )  
  (quant): QuantStub(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )  
  (dequant): DeQuantStub(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )  
)  
=====  
The summary of the 4-layer CNN network with conv. pruning:  
model: 4Conv2FC-conv-pruned Conv-channels: 706 to 226  
=====
```

```
Classifier_ld_4c_2fc_sub_qr(  
  (raw): Sequential(  
    (0): MaxPool2d(kernel_size=1, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (1): SepConv1d_v4(  
      (layers): Sequential(  
        (0): Conv2d(2, 2, kernel_size=(1, 8), stride=(1, 4), padding=(0, 3), groups=2)  
        (1): Conv2d(2, 23, kernel_size=(1, 1), stride=(1, 1))  
        (2): BatchNorm2d(23, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (3): ReLU(inplace=True)  
        (4): Dropout(p=0.5, inplace=False)  
      )  
    )  
    (2): SepConv1d_v4(  
      (layers): Sequential(  
        (0): Conv2d(23, 23, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=23)  
        (1): Conv2d(23, 24, kernel_size=(1, 1), stride=(1, 1))  
        (2): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (3): ReLU(inplace=True)  
        (4): Dropout(p=0.5, inplace=False)  
      )  
    )  
  )  
  (quant): QuantStub(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )  
  (dequant): DeQuantStub(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )  
)
```

```

)
)
(3): SepConv1d_v4(
  (layers): Sequential(
    (0): Conv2d(24, 24, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=24)
    (1): Conv2d(24, 43, kernel_size=(1, 1), stride=(1, 1))
    (2): BatchNorm2d(43, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): ReLU(inplace=True)
    (4): Dropout(p=0.5, inplace=False)
  )
)
(4): SepConv1d_v4(
  (layers): Sequential(
    (0): Conv2d(43, 43, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=43)
    (1): Conv2d(43, 43, kernel_size=(1, 1), stride=(1, 1))
    (2): BatchNorm2d(43, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): ReLU(inplace=True)
    (4): Dropout(p=0.5, inplace=False)
  )
)
)
)
(FC): Sequential(
  (0): Flatten()
  (1): Linear(in_features=172, out_features=128, bias=True)
  (2): ReLU(inplace=True)
  (3): Dropout(p=0.5, inplace=False)
)
(out): Sequential(
  (0): Linear(in_features=128, out_features=2, bias=True)
)
(quant): QuantStub()
(dequant): DeQuantStub()
)

```

```

=====
Table of the network parameters:
model: 4Conv2FC-conv-pruned   Conv-channels: 706 to 226
=====

```

Layer (type)	Output Shape	Param #
QuantStub-1	[512, 2, 2048]	0
MaxPool2d-2	[512, 2, 1, 1024]	0
Conv2d-3	[512, 2, 1, 256]	18
Conv2d-4	[512, 23, 1, 256]	69
BatchNorm2d-5	[512, 23, 1, 256]	46
ReLU-6	[512, 23, 1, 256]	0
Dropout-7	[512, 23, 1, 256]	0
SepConv1d_v4-8	[512, 23, 1, 256]	0
Conv2d-9	[512, 23, 1, 64]	207
Conv2d-10	[512, 24, 1, 64]	576
BatchNorm2d-11	[512, 24, 1, 64]	48
ReLU-12	[512, 24, 1, 64]	0
Dropout-13	[512, 24, 1, 64]	0
SepConv1d_v4-14	[512, 24, 1, 64]	0
Conv2d-15	[512, 24, 1, 16]	216
Conv2d-16	[512, 43, 1, 16]	1,075
BatchNorm2d-17	[512, 43, 1, 16]	86
ReLU-18	[512, 43, 1, 16]	0
Dropout-19	[512, 43, 1, 16]	0
SepConv1d_v4-20	[512, 43, 1, 16]	0
Conv2d-21	[512, 43, 1, 4]	387
Conv2d-22	[512, 43, 1, 4]	1,892
BatchNorm2d-23	[512, 43, 1, 4]	86
ReLU-24	[512, 43, 1, 4]	0
Dropout-25	[512, 43, 1, 4]	0
SepConv1d_v4-26	[512, 43, 1, 4]	0
Flatten-27	[512, 172]	0
Linear-28	[512, 128]	22,144
ReLU-29	[512, 128]	0
Dropout-30	[512, 128]	0
Linear-31	[512, 2]	258
DeQuantStub-32	[512, 2]	0

```

=====
Total params: 27,108
Trainable params: 27,108
Non-trainable params: 0

```

```

-----
Input size (MB): 8.00
Forward/backward pass size (MB): 197.91
Params size (MB): 0.10
Estimated Total Size (MB): 206.01
-----

```

```

=====
Detail of the network's per layer computations and parameters:
model: 4Conv2FC-conv-pruned   Conv-channels: 706 to 226
=====

```

```

Classifier_1d_4c_2fc_sub_qr(
  0.027 M, 100.000% Params, 0.154 MMac, 100.000% MACs,
  (raw): Sequential(
    0.005 M, 17.360% Params, 0.131 MMac, 85.408% MACs,
    (0): MaxPool2d(0.0 M, 0.000% Params, 0.004 MMac, 2.668% MACs, kernel_size=1, stride=2, padding=0, dilation=1, ceil_mode=False)
    (1): SepConv1d_v4(
      0.0 M, 0.491% Params, 0.04 MMac, 26.016% MACs,

```

```

(layers): Sequential(
  0.0 M, 0.491% Params, 0.04 MMac, 26.016% MACs,
  (0): Conv2d(0.0 M, 0.066% Params, 0.005 MMac, 3.002% MACs, 2, 2, kernel_size=(1, 8), stride=(1, 4), padding=(0, 3), groups=2)
  (1): Conv2d(0.0 M, 0.255% Params, 0.018 MMac, 11.507% MACs, 2, 23, kernel_size=(1, 1), stride=(1, 1))
  (2): BatchNorm2d(0.0 M, 0.170% Params, 0.012 MMac, 7.671% MACs, 23, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (3): ReLU(0.0 M, 0.000% Params, 0.006 MMac, 3.836% MACs, inplace=True)
  (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
)
)
(2): SepConv1d_v4(
  0.001 M, 3.066% Params, 0.055 MMac, 35.647% MACs,
  (layers): Sequential(
    0.001 M, 3.066% Params, 0.055 MMac, 35.647% MACs,
    (0): Conv2d(0.0 M, 0.764% Params, 0.013 MMac, 8.630% MACs, 23, 23, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),
groups=23)
    (1): Conv2d(0.001 M, 2.125% Params, 0.037 MMac, 24.015% MACs, 23, 24, kernel_size=(1, 1), stride=(1, 1))
    (2): BatchNorm2d(0.0 M, 0.177% Params, 0.003 MMac, 2.001% MACs, 24, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (3): ReLU(0.0 M, 0.000% Params, 0.002 MMac, 1.001% MACs, inplace=True)
    (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
  )
)
(3): SepConv1d_v4(
  0.001 M, 5.080% Params, 0.023 MMac, 14.801% MACs,
  (layers): Sequential(
    0.001 M, 5.080% Params, 0.023 MMac, 14.801% MACs,
    (0): Conv2d(0.0 M, 0.797% Params, 0.003 MMac, 2.251% MACs, 24, 24, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),
groups=24)
    (1): Conv2d(0.001 M, 3.966% Params, 0.017 MMac, 11.205% MACs, 24, 43, kernel_size=(1, 1), stride=(1, 1))
    (2): BatchNorm2d(0.0 M, 0.317% Params, 0.001 MMac, 0.896% MACs, 43, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (3): ReLU(0.0 M, 0.000% Params, 0.001 MMac, 0.448% MACs, inplace=True)
    (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
  )
)
(4): SepConv1d_v4(
  0.002 M, 8.724% Params, 0.01 MMac, 6.275% MACs,
  (layers): Sequential(
    0.002 M, 8.724% Params, 0.01 MMac, 6.275% MACs,
    (0): Conv2d(0.0 M, 1.428% Params, 0.002 MMac, 1.008% MACs, 43, 43, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),
groups=43)
    (1): Conv2d(0.002 M, 6.979% Params, 0.008 MMac, 4.930% MACs, 43, 43, kernel_size=(1, 1), stride=(1, 1))
    (2): BatchNorm2d(0.0 M, 0.317% Params, 0.0 MMac, 0.224% MACs, 43, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (3): ReLU(0.0 M, 0.000% Params, 0.0 MMac, 0.112% MACs, inplace=True)
    (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
  )
)
)
)
(FC): Sequential(
  0.022 M, 81.688% Params, 0.022 MMac, 14.426% MACs,
  (0): Flatten(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )
  (1): Linear(0.022 M, 81.688% Params, 0.022 MMac, 14.342% MACs, in_features=172, out_features=128, bias=True)
  (2): ReLU(0.0 M, 0.000% Params, 0.0 MMac, 0.083% MACs, inplace=True)
  (3): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
)
(out): Sequential(
  0.0 M, 0.952% Params, 0.0 MMac, 0.167% MACs,
  (0): Linear(0.0 M, 0.952% Params, 0.0 MMac, 0.167% MACs, in_features=128, out_features=2, bias=True)
)
(quant): QuantStub(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )
(dequant): DeQuantStub(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )
)

```

```

=====
The summary of the 4-layer CNN network with FC. pruning:
model: 4Conv2FC-FC-pruned    FC-nourons: 128 to 89
=====

```

```

Classifier_1d_4c_2fc_sub_qr(
  (raw): Sequential(
    (0): MaxPool2d(kernel_size=1, stride=2, padding=0, dilation=1, ceil_mode=False)
    (1): SepConv1d_v4(
      (layers): Sequential(
        (0): Conv2d(2, 2, kernel_size=(1, 8), stride=(1, 4), padding=(0, 3), groups=2)
        (1): Conv2d(2, 29, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(29, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (3): ReLU(inplace=True)
        (4): Dropout(p=0.5, inplace=False)
      )
    )
    (2): SepConv1d_v4(
      (layers): Sequential(
        (0): Conv2d(29, 29, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=29)
        (1): Conv2d(29, 32, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (3): ReLU(inplace=True)
        (4): Dropout(p=0.5, inplace=False)
      )
    )
    (3): SepConv1d_v4(
      (layers): Sequential(
        (0): Conv2d(32, 32, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=32)
        (1): Conv2d(32, 79, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(79, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (3): ReLU(inplace=True)
        (4): Dropout(p=0.5, inplace=False)
      )
    )
  )
)

```

```
(4): SepConv1d_v4(
  (layers): Sequential(
    (0): Conv2d(79, 79, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=79)
    (1): Conv2d(79, 139, kernel_size=(1, 1), stride=(1, 1))
    (2): BatchNorm2d(139, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): ReLU(inplace=True)
    (4): Dropout(p=0.5, inplace=False)
  )
)
(FC): Sequential(
  (0): Flatten()
  (1): Linear(in_features=556, out_features=87, bias=True)
  (2): ReLU(inplace=True)
  (3): Dropout(p=0.5, inplace=False)
)
(out): Sequential(
  (0): Linear(in_features=87, out_features=2, bias=True)
)
(quant): QuantStub()
(dequant): DeQuantStub()
)
```

Layer (type)	Output Shape	Param #
QuantStub-1	[512, 2, 2048]	0
MaxPool2d-2	[512, 2, 1, 1024]	0
Conv2d-3	[512, 2, 1, 256]	18
Conv2d-4	[512, 29, 1, 256]	87
BatchNorm2d-5	[512, 29, 1, 256]	58
ReLU-6	[512, 29, 1, 256]	0
Dropout-7	[512, 29, 1, 256]	0
SepConv1d_v4-8	[512, 29, 1, 256]	0
Conv2d-9	[512, 29, 1, 64]	261
Conv2d-10	[512, 32, 1, 64]	960
BatchNorm2d-11	[512, 32, 1, 64]	64
ReLU-12	[512, 32, 1, 64]	0
Dropout-13	[512, 32, 1, 64]	0
SepConv1d_v4-14	[512, 32, 1, 64]	0
Conv2d-15	[512, 32, 1, 16]	288
Conv2d-16	[512, 79, 1, 16]	2,607
BatchNorm2d-17	[512, 79, 1, 16]	158
ReLU-18	[512, 79, 1, 16]	0
Dropout-19	[512, 79, 1, 16]	0
SepConv1d_v4-20	[512, 79, 1, 16]	0
Conv2d-21	[512, 79, 1, 4]	711
Conv2d-22	[512, 139, 1, 4]	11,120
BatchNorm2d-23	[512, 139, 1, 4]	278
ReLU-24	[512, 139, 1, 4]	0
Dropout-25	[512, 139, 1, 4]	0
SepConv1d_v4-26	[512, 139, 1, 4]	0
Flatten-27	[512, 556]	0
Linear-28	[512, 87]	48,459
ReLU-29	[512, 87]	0
Dropout-30	[512, 87]	0
Linear-31	[512, 2]	176
DeQuantStub-32	[512, 2]	0

Detail of the network's per layer computations and parameters:
model: 4Conv2FC-FC-pruned FC-neurons: 128 to 89

```

0.001 M, 1.969% Params, 0.084 MMac, 29.546% MACs,
(layers): Sequential(
  0.001 M, 1.969% Params, 0.084 MMac, 29.546% MACs,
  (0): Conv2d(0.0 M, 0.400% Params, 0.017 MMac, 5.855% MACs, 29, 29, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),
groups=29)
  (1): Conv2d(0.001 M, 1.471% Params, 0.061 MMac, 21.537% MACs, 29, 32, kernel_size=(1, 1), stride=(1, 1))
  (2): BatchNorm2d(0.0 M, 0.098% Params, 0.004 MMac, 1.436% MACs, 32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (3): ReLU(0.0 M, 0.000% Params, 0.002 MMac, 0.718% MACs, inplace=True)
  (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
)
)
(3): SepConv1d_v4(
  0.003 M, 4.679% Params, 0.05 MMac, 17.566% MACs,
  (layers): Sequential(
    0.003 M, 4.679% Params, 0.05 MMac, 17.566% MACs,
    (0): Conv2d(0.0 M, 0.441% Params, 0.005 MMac, 1.615% MACs, 32, 32, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),
groups=32)
    (1): Conv2d(0.003 M, 3.996% Params, 0.042 MMac, 14.622% MACs, 32, 79, kernel_size=(1, 1), stride=(1, 1))
    (2): BatchNorm2d(0.0 M, 0.242% Params, 0.003 MMac, 0.886% MACs, 79, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (3): ReLU(0.0 M, 0.000% Params, 0.001 MMac, 0.443% MACs, inplace=True)
    (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
  )
)
(4): SepConv1d_v4(
  0.012 M, 18.559% Params, 0.049 MMac, 17.174% MACs,
  (layers): Sequential(
    0.012 M, 18.559% Params, 0.049 MMac, 17.174% MACs,
    (0): Conv2d(0.001 M, 1.090% Params, 0.003 MMac, 0.997% MACs, 79, 79, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),
groups=79)
    (1): Conv2d(0.011 M, 17.043% Params, 0.044 MMac, 15.592% MACs, 79, 139, kernel_size=(1, 1), stride=(1, 1))
    (2): BatchNorm2d(0.0 M, 0.426% Params, 0.001 MMac, 0.390% MACs, 139, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (3): ReLU(0.0 M, 0.000% Params, 0.001 MMac, 0.195% MACs, inplace=True)
    (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
  )
)
)
(FC): Sequential(
  0.048 M, 74.272% Params, 0.048 MMac, 16.987% MACs,
  (0): Flatten(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )
  (1): Linear(0.048 M, 74.272% Params, 0.048 MMac, 16.956% MACs, in_features=556, out_features=87, bias=True)
  (2): ReLU(0.0 M, 0.000% Params, 0.0 MMac, 0.030% MACs, inplace=True)
  (3): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
)
(out): Sequential(
  0.0 M, 0.270% Params, 0.0 MMac, 0.061% MACs,
  (0): Linear(0.0 M, 0.270% Params, 0.0 MMac, 0.061% MACs, in_features=87, out_features=2, bias=True)
)
(quant): QuantStub(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )
(dequant): DeQuantStub(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )
)

```

```

=====
The summary of the 4-layer CNN network with FC. pruning:
model: 4Conv2FC-FC-pruned    FC-nourons: 128 to 42
=====

```

```

Classifier_1d_4c_2fc_sub_qr(
  (raw): Sequential(
    (0): MaxPool2d(kernel_size=1, stride=2, padding=0, dilation=1, ceil_mode=False)
    (1): SepConv1d_v4(
      (layers): Sequential(
        (0): Conv2d(2, 2, kernel_size=(1, 8), stride=(1, 4), padding=(0, 3), groups=2)
        (1): Conv2d(2, 29, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(29, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (3): ReLU(inplace=True)
        (4): Dropout(p=0.5, inplace=False)
      )
    )
    (2): SepConv1d_v4(
      (layers): Sequential(
        (0): Conv2d(29, 29, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=29)
        (1): Conv2d(29, 32, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (3): ReLU(inplace=True)
        (4): Dropout(p=0.5, inplace=False)
      )
    )
    (3): SepConv1d_v4(
      (layers): Sequential(
        (0): Conv2d(32, 32, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=32)
        (1): Conv2d(32, 79, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(79, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (3): ReLU(inplace=True)
        (4): Dropout(p=0.5, inplace=False)
      )
    )
    (4): SepConv1d_v4(
      (layers): Sequential(
        (0): Conv2d(79, 79, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=79)
        (1): Conv2d(79, 139, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(139, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (3): ReLU(inplace=True)
        (4): Dropout(p=0.5, inplace=False)
      )
    )
  )
  (FC): Sequential(

```



```

(0): Flatten()
(1): Linear(in_features=556, out_features=42, bias=True)
(2): ReLU(inplace=True)
(3): Dropout(p=0.5, inplace=False)
)
(out): Sequential(
  (0): Linear(in_features=42, out_features=2, bias=True)
)
(quant): QuantStub()
(dequant): DeQuantStub()
)

```

```

=====
Table of the network parameters:
model: 4Conv2FC-FC-pruned    FC-nourons: 128 to 42
=====

```

Layer (type)	Output Shape	Param #
QuantStub-1	[512, 2, 2048]	0
MaxPool2d-2	[512, 2, 1, 1024]	0
Conv2d-3	[512, 2, 1, 256]	18
Conv2d-4	[512, 29, 1, 256]	87
BatchNorm2d-5	[512, 29, 1, 256]	58
ReLU-6	[512, 29, 1, 256]	0
Dropout-7	[512, 29, 1, 256]	0
SepConv1d_v4-8	[512, 29, 1, 256]	0
Conv2d-9	[512, 29, 1, 64]	261
Conv2d-10	[512, 32, 1, 64]	960
BatchNorm2d-11	[512, 32, 1, 64]	64
ReLU-12	[512, 32, 1, 64]	0
Dropout-13	[512, 32, 1, 64]	0
SepConv1d_v4-14	[512, 32, 1, 64]	0
Conv2d-15	[512, 32, 1, 16]	288
Conv2d-16	[512, 79, 1, 16]	2,607
BatchNorm2d-17	[512, 79, 1, 16]	158
ReLU-18	[512, 79, 1, 16]	0
Dropout-19	[512, 79, 1, 16]	0
SepConv1d_v4-20	[512, 79, 1, 16]	0
Conv2d-21	[512, 79, 1, 4]	711
Conv2d-22	[512, 139, 1, 4]	11,120
BatchNorm2d-23	[512, 139, 1, 4]	278
ReLU-24	[512, 139, 1, 4]	0
Dropout-25	[512, 139, 1, 4]	0
SepConv1d_v4-26	[512, 139, 1, 4]	0
Flatten-27	[512, 556]	0
Linear-28	[512, 42]	23,394
ReLU-29	[512, 42]	0
Dropout-30	[512, 42]	0
Linear-31	[512, 2]	86
DeQuantStub-32	[512, 2]	0

```

=====
Total params: 40,090
Trainable params: 40,090
Non-trainable params: 0
=====

```

```

-----
Input size (MB): 8.00
Forward/backward pass size (MB): 259.71
Params size (MB): 0.15
Estimated Total Size (MB): 267.86
-----

```

```

=====
Detail of the network's per layer computations and parameters:
model: 4Conv2FC-FC-pruned    FC-nourons: 128 to 42
=====

```

```

Classifier 1d_4c_2fc_sub_qr(
  0.04 M, 100.000% Params, 0.26 MMac, 100.000% MACs,
  (raw): Sequential(
    0.017 M, 41.432% Params, 0.237 MMac, 90.974% MACs,
    (0): MaxPool2d(0.0 M, 0.000% Params, 0.004 MMac, 1.575% MACs, kernel_size=1, stride=2, padding=0, dilation=1, ceil_mode=False)
    (1): SepConv1d_v4(
      0.0 M, 0.407% Params, 0.049 MMac, 18.896% MACs,
      (layers): Sequential(
        0.0 M, 0.407% Params, 0.049 MMac, 18.896% MACs,
        (0): Conv2d(0.0 M, 0.045% Params, 0.005 MMac, 1.772% MACs, 2, 2, kernel_size=(1, 8), stride=(1, 4), padding=(0, 3), groups=2)
        (1): Conv2d(0.0 M, 0.217% Params, 0.022 MMac, 8.562% MACs, 2, 29, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(0.0 M, 0.145% Params, 0.015 MMac, 5.708% MACs, 29, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (3): ReLU(0.0 M, 0.000% Params, 0.007 MMac, 2.854% MACs, inplace=True)
        (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
      )
    )
    (2): SepConv1d_v4(
      0.001 M, 3.205% Params, 0.084 MMac, 32.404% MACs,
      (layers): Sequential(
        0.001 M, 3.205% Params, 0.084 MMac, 32.404% MACs,
        (0): Conv2d(0.0 M, 0.651% Params, 0.017 MMac, 6.422% MACs, 29, 29, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),
groups=29)
        (1): Conv2d(0.001 M, 2.395% Params, 0.061 MMac, 23.620% MACs, 29, 32, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(0.0 M, 0.160% Params, 0.004 MMac, 1.575% MACs, 32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (3): ReLU(0.0 M, 0.000% Params, 0.002 MMac, 0.787% MACs, inplace=True)
        (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
      )
    )
  )
)

```

```

)
(3): SepConv1d_v4(
  0.003 M, 7.615% Params, 0.05 MMac, 19.265% MACs,
  (layers): Sequential(
    0.003 M, 7.615% Params, 0.05 MMac, 19.265% MACs,
    (0): Conv2d(0.0 M, 0.718% Params, 0.005 MMac, 1.772% MACs, 32, 32, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),
groups=32)
    (1): Conv2d(0.003 M, 6.503% Params, 0.042 MMac, 16.036% MACs, 32, 79, kernel_size=(1, 1), stride=(1, 1))
    (2): BatchNorm2d(0.0 M, 0.394% Params, 0.003 MMac, 0.972% MACs, 79, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (3): ReLU(0.0 M, 0.000% Params, 0.001 MMac, 0.486% MACs, inplace=True)
    (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
  )
)
(4): SepConv1d_v4(
  0.012 M, 30.205% Params, 0.049 MMac, 18.835% MACs,
  (layers): Sequential(
    0.012 M, 30.205% Params, 0.049 MMac, 18.835% MACs,
    (0): Conv2d(0.001 M, 1.774% Params, 0.003 MMac, 1.093% MACs, 79, 79, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2),
groups=79)
    (1): Conv2d(0.011 M, 27.738% Params, 0.044 MMac, 17.100% MACs, 79, 139, kernel_size=(1, 1), stride=(1, 1))
    (2): BatchNorm2d(0.0 M, 0.693% Params, 0.001 MMac, 0.427% MACs, 139, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (3): ReLU(0.0 M, 0.000% Params, 0.001 MMac, 0.214% MACs, inplace=True)
    (4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
  )
)
)
(FC): Sequential(
  0.023 M, 58.354% Params, 0.023 MMac, 8.994% MACs,
  (0): Flatten(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )
  (1): Linear(0.023 M, 58.354% Params, 0.023 MMac, 8.977% MACs, in_features=556, out_features=42, bias=True)
  (2): ReLU(0.0 M, 0.000% Params, 0.0 MMac, 0.016% MACs, inplace=True)
  (3): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
)
(out): Sequential(
  0.0 M, 0.215% Params, 0.0 MMac, 0.032% MACs,
  (0): Linear(0.0 M, 0.215% Params, 0.0 MMac, 0.032% MACs, in_features=42, out_features=2, bias=True)
)
(quant): QuantStub(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )
(dequant): DeQuantStub(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )
)

```

In [31]: