

=====

The summary of the 6-layer CNN network:

=====

```
Classifier_1d_6_conv_v2(  
  (raw): Sequential(  
    (0): SepConv1d_v4(  
      (layers): Sequential(  
        (0): Conv2d(2, 2, kernel_size=(1, 8), stride=(1, 2), 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)  
      )  
    )  
    (1): 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)  
      )  
    )  
    (2): 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)  
      )  
    )  
    (3): 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)  
      )  
    )  
    (4): SepConv1d_v4(  
      (layers): Sequential(  
        (0): Conv2d(256, 256, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=256)  
        (1): Conv2d(256, 512, kernel_size=(1, 1), stride=(1, 1))  
        (2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (3): ReLU(inplace=True)  
        (4): Dropout(p=0.5, inplace=False)  
      )  
    )  
    (5): SepConv1d_v4(  
      (layers): Sequential(  
        (0): Conv2d(512, 512, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=512)  
        (1): Conv2d(512, 1024, kernel_size=(1, 1), stride=(1, 1))  
        (2): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (3): ReLU(inplace=True)  
        (4): Dropout(p=0.2, 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)  
  (4): Linear(in_features=128, out_features=128, bias=True)  
  (5): ReLU(inplace=True)  
  (6): 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:

=====

Layer (type)	Output Shape	Param #
QuantStub-1	[512, 2, 2048]	0
Conv2d-2	[512, 2, 1, 1024]	18
Conv2d-3	[512, 32, 1, 1024]	96
BatchNorm2d-4	[512, 32, 1, 1024]	64
ReLU-5	[512, 32, 1, 1024]	0
Dropout-6	[512, 32, 1, 1024]	0
SepConv1d_v4-7	[512, 32, 1, 1024]	0
Conv2d-8	[512, 32, 1, 256]	288
Conv2d-9	[512, 64, 1, 256]	2,112
BatchNorm2d-10	[512, 64, 1, 256]	128
ReLU-11	[512, 64, 1, 256]	0
Dropout-12	[512, 64, 1, 256]	0
SepConv1d_v4-13	[512, 64, 1, 256]	0
Conv2d-14	[512, 64, 1, 64]	576
Conv2d-15	[512, 128, 1, 64]	8,320
BatchNorm2d-16	[512, 128, 1, 64]	256
ReLU-17	[512, 128, 1, 64]	0
Dropout-18	[512, 128, 1, 64]	0
SepConv1d_v4-19	[512, 128, 1, 64]	0
Conv2d-20	[512, 128, 1, 16]	1,152
Conv2d-21	[512, 256, 1, 16]	33,024
BatchNorm2d-22	[512, 256, 1, 16]	512
ReLU-23	[512, 256, 1, 16]	0
Dropout-24	[512, 256, 1, 16]	0
SepConv1d_v4-25	[512, 256, 1, 16]	0
Conv2d-26	[512, 256, 1, 4]	2,304
Conv2d-27	[512, 512, 1, 4]	131,584
BatchNorm2d-28	[512, 512, 1, 4]	1,024
ReLU-29	[512, 512, 1, 4]	0
Dropout-30	[512, 512, 1, 4]	0
SepConv1d_v4-31	[512, 512, 1, 4]	0
Conv2d-32	[512, 512, 1, 1]	4,608
Conv2d-33	[512, 1024, 1, 1]	525,312
BatchNorm2d-34	[512, 1024, 1, 1]	2,048
ReLU-35	[512, 1024, 1, 1]	0
Dropout-36	[512, 1024, 1, 1]	0
SepConv1d_v4-37	[512, 1024, 1, 1]	0
Flatten-38	[512, 1024]	0
Linear-39	[512, 128]	131,200
ReLU-40	[512, 128]	0
Dropout-41	[512, 128]	0
Linear-42	[512, 128]	16,512
ReLU-43	[512, 128]	0
Dropout-44	[512, 128]	0
Linear-45	[512, 2]	258
DeQuantStub-46	[512, 2]	0

Total params: 861,396  
Trainable params: 861,396  
Non-trainable params: 0

-----

Input size (MB): 8.00  
Forward/backward pass size (MB): 1353.02  
Params size (MB): 3.29  
Estimated Total Size (MB): 1364.30

-----

=====

Detail of the network's per layer computations and parameters:

=====

```
Classifier_ld_6_conv_v2(
0.861 M, 100.000% Params, 3.254 MMac, 100.000% MACs,
(raw): Sequential(
0.713 M, 82.822% Params, 3.106 MMac, 95.453% MACs,
(0): SepConv1d_v4(
0.0 M, 0.021% Params, 0.215 MMac, 6.608% MACs,
(layers): Sequential(
0.0 M, 0.021% Params, 0.215 MMac, 6.608% MACs,
(0): Conv2d(0.0 M, 0.002% Params, 0.018 MMac, 0.566% MACs, 2, 2, kernel_size=(1, 8), stride=(1, 2), padding=(0, 3), groups=2)
(1): Conv2d(0.0 M, 0.011% Params, 0.098 MMac, 3.021% MACs, 2, 32, kernel_size=(1, 1), stride=(1, 1))
(2): BatchNorm2d(0.0 M, 0.007% Params, 0.066 MMac, 2.014% MACs, 32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(3): ReLU(0.0 M, 0.000% Params, 0.033 MMac, 1.007% MACs, inplace=True)
(4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
)
)
(1): SepConv1d_v4(
0.003 M, 0.293% Params, 0.664 MMac, 20.390% MACs,
(layers): Sequential(
0.003 M, 0.293% Params, 0.664 MMac, 20.390% MACs,
(0): Conv2d(0.0 M, 0.033% Params, 0.074 MMac, 2.266% MACs, 32, 32, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=32)
(1): Conv2d(0.002 M, 0.245% Params, 0.541 MMac, 16.614% MACs, 32, 64, kernel_size=(1, 1), stride=(1, 1))
(2): BatchNorm2d(0.0 M, 0.015% Params, 0.033 MMac, 1.007% MACs, 64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(3): ReLU(0.0 M, 0.000% Params, 0.016 MMac, 0.503% 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(
0.009 M, 1.062% Params, 0.594 MMac, 18.250% MACs,
(layers): Sequential(
0.009 M, 1.062% Params, 0.594 MMac, 18.250% MACs,
(0): Conv2d(0.001 M, 0.067% Params, 0.037 MMac, 1.133% MACs, 64, 64, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=64)
(1): Conv2d(0.008 M, 0.966% Params, 0.532 MMac, 16.362% MACs, 64, 128, kernel_size=(1, 1), stride=(1, 1))
(2): BatchNorm2d(0.0 M, 0.030% Params, 0.016 MMac, 0.503% MACs, 128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(3): ReLU(0.0 M, 0.000% Params, 0.008 MMac, 0.252% 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(
0.035 M, 4.027% Params, 0.559 MMac, 17.181% MACs,
(layers): Sequential(
0.035 M, 4.027% Params, 0.559 MMac, 17.181% MACs,
(0): Conv2d(0.001 M, 0.134% Params, 0.018 MMac, 0.566% MACs, 128, 128, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=128)
(1): Conv2d(0.033 M, 3.834% Params, 0.528 MMac, 16.237% MACs, 128, 256, kernel_size=(1, 1), stride=(1, 1))
(2): BatchNorm2d(0.001 M, 0.059% Params, 0.008 MMac, 0.252% MACs, 256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(3): ReLU(0.0 M, 0.000% Params, 0.004 MMac, 0.126% 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(
0.135 M, 15.662% Params, 0.542 MMac, 16.646% MACs,
(layers): Sequential(
0.135 M, 15.662% Params, 0.542 MMac, 16.646% MACs,
(0): Conv2d(0.002 M, 0.267% Params, 0.009 MMac, 0.283% MACs, 256, 256, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=256)
(1): Conv2d(0.132 M, 15.276% Params, 0.526 MMac, 16.174% MACs, 256, 512, kernel_size=(1, 1), stride=(1, 1))
(2): BatchNorm2d(0.001 M, 0.119% Params, 0.004 MMac, 0.126% MACs, 512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(3): ReLU(0.0 M, 0.000% Params, 0.002 MMac, 0.063% MACs, inplace=True)
(4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
)
)
(5): SepConv1d_v4(
0.532 M, 61.756% Params, 0.533 MMac, 16.378% MACs,
(layers): Sequential(
0.532 M, 61.756% Params, 0.533 MMac, 16.378% MACs,
(0): Conv2d(0.005 M, 0.535% Params, 0.005 MMac, 0.142% MACs, 512, 512, kernel_size=(1, 8), stride=(1, 4), padding=(0, 2), groups=512)
(1): Conv2d(0.525 M, 60.984% Params, 0.525 MMac, 16.142% MACs, 512, 1024, kernel_size=(1, 1), stride=(1, 1))
(2): BatchNorm2d(0.002 M, 0.238% Params, 0.002 MMac, 0.063% MACs, 1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(3): ReLU(0.0 M, 0.000% Params, 0.001 MMac, 0.031% MACs, inplace=True)
(4): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.2, inplace=False)
)
)
)
(FC): Sequential(
0.148 M, 17.148% Params, 0.148 MMac, 4.539% MACs,
(0): Flatten(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, )
(1): Linear(0.131 M, 15.231% Params, 0.131 MMac, 4.028% MACs, in_features=1024, out_features=128, bias=True)
(2): ReLU(0.0 M, 0.000% Params, 0.0 MMac, 0.004% MACs, inplace=True)
(3): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
(4): Linear(0.017 M, 1.917% Params, 0.016 MMac, 0.503% MACs, in_features=128, out_features=128, bias=True)
(5): ReLU(0.0 M, 0.000% Params, 0.0 MMac, 0.004% MACs, inplace=True)
(6): Dropout(0.0 M, 0.000% Params, 0.0 MMac, 0.000% MACs, p=0.5, inplace=False)
)
(out): Sequential(
0.0 M, 0.030% Params, 0.0 MMac, 0.008% MACs,
(0): Linear(0.0 M, 0.030% Params, 0.0 MMac, 0.008% 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 network's accuracy:

=====

The network has the following accuracy:  
TP: 99.10 , FP: 3.88 AF\_threshold = 3 (observing at least 3 AF signs to classify ECG as AF)  
TP: 98.77 , FP: 1.5 AF\_threshold = 7

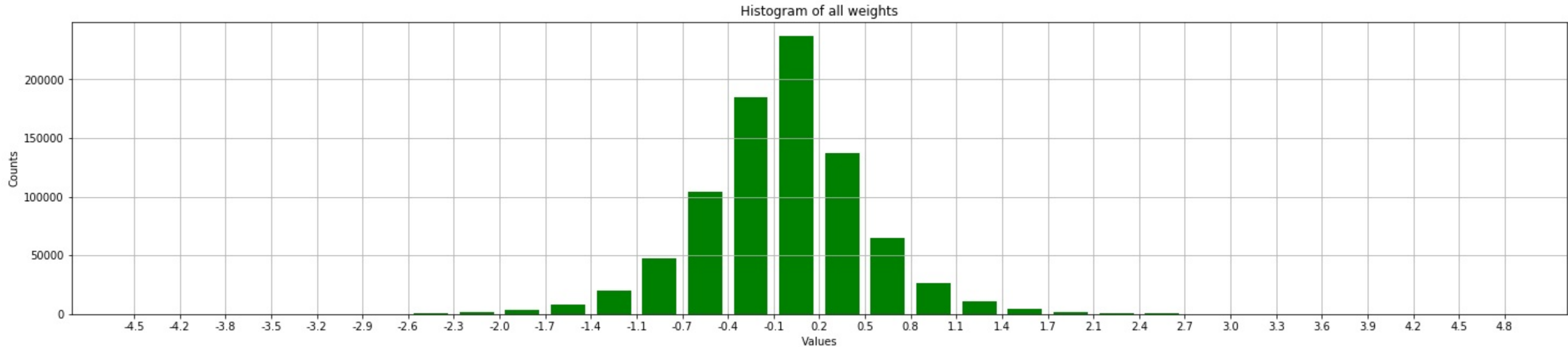
=====

Statistical analysis of the weights:

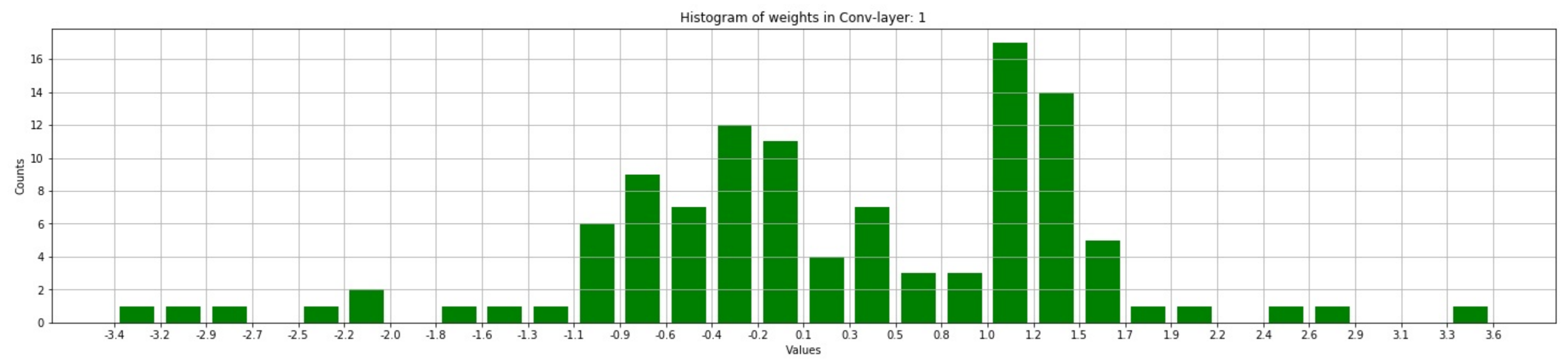
=====

===== All weights of the network

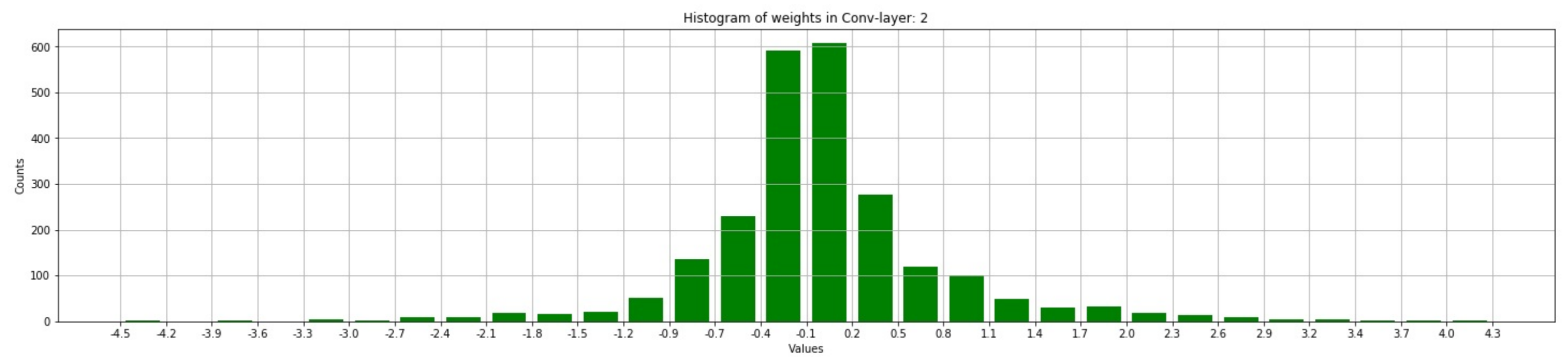
Minimum of weights: -4.4108548  
Maximum of weights: 4.8913217  
Average value of weights: -0.0088230  
Variance of weights: 0.5572891



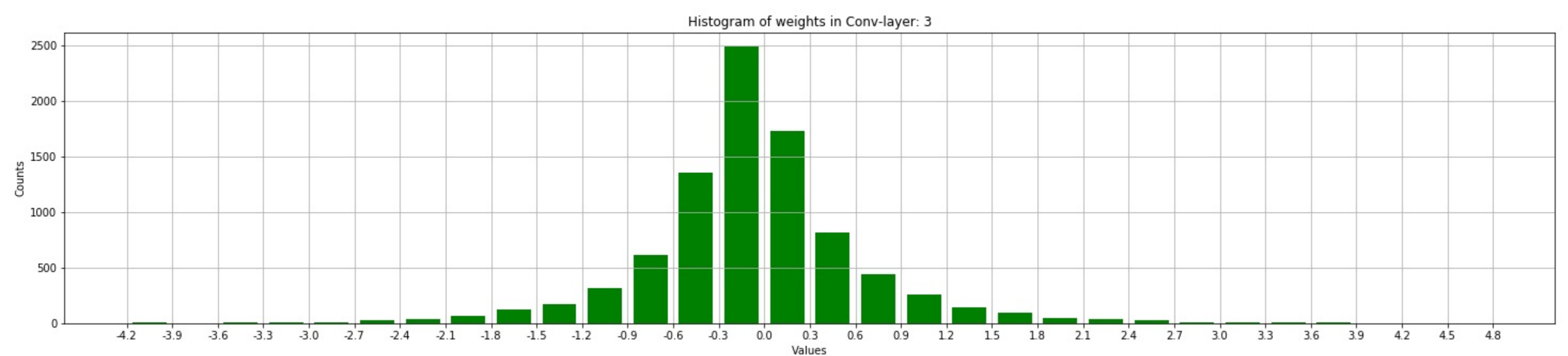
===== Conv. Layer: 1  
Minimum of weights: -3.3641193  
Maximum of weights: 3.6236877  
Average value of weights: 0.2965188  
Variance of weights: 1.1729762



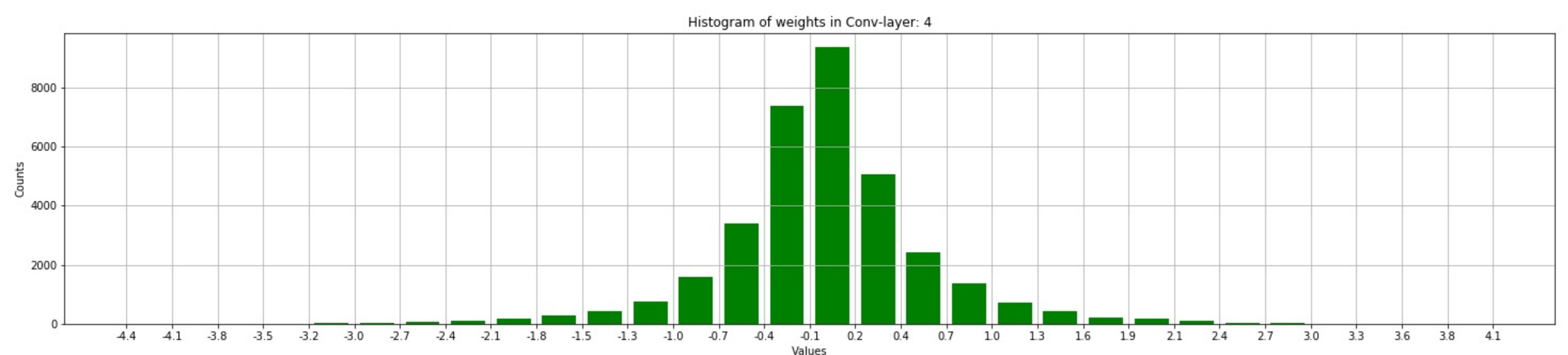
=====  
Conv. Layer: 2  
Minimum of weights: -4.4108548  
Maximum of weights: 4.3738699  
Average value of weights: 0.0736898  
Variance of weights: 0.7903544



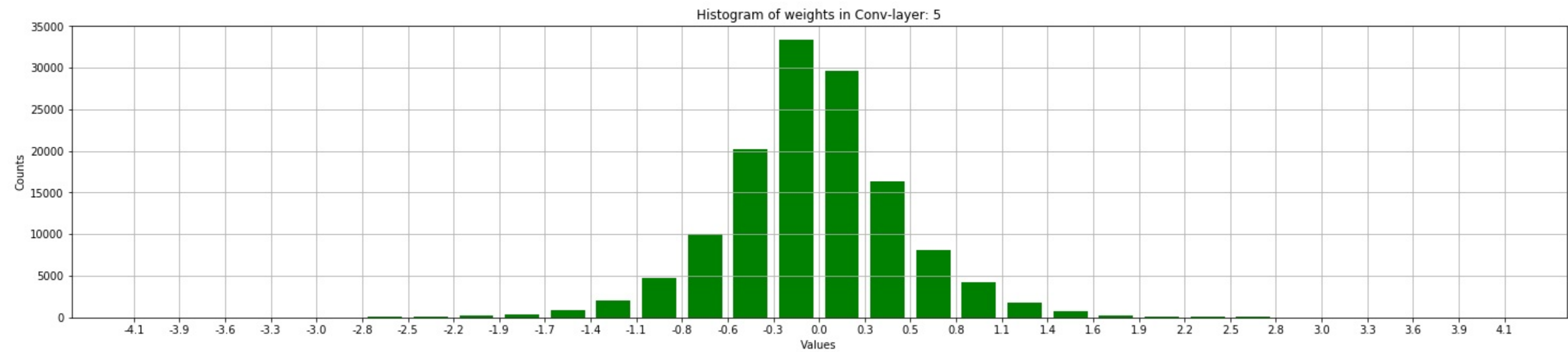
=====  
Conv. Layer: 3  
Minimum of weights: -4.1125565  
Maximum of weights: 4.8913217  
Average value of weights: 0.0162513  
Variance of weights: 0.7178196



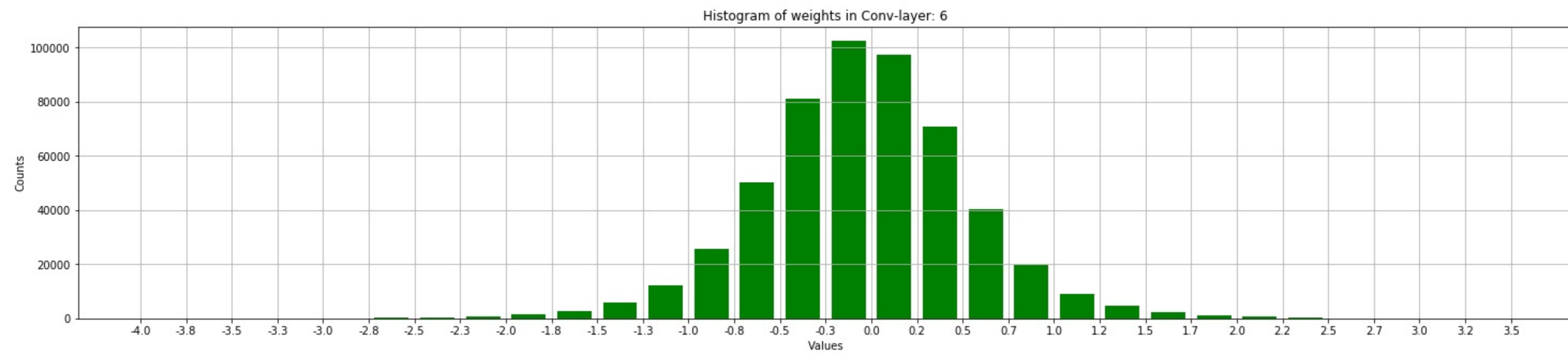
=====  
Conv. Layer: 4  
Minimum of weights: -4.3188796  
Maximum of weights: 4.1786413  
Average value of weights: 0.0214825  
Variance of weights: 0.6290445



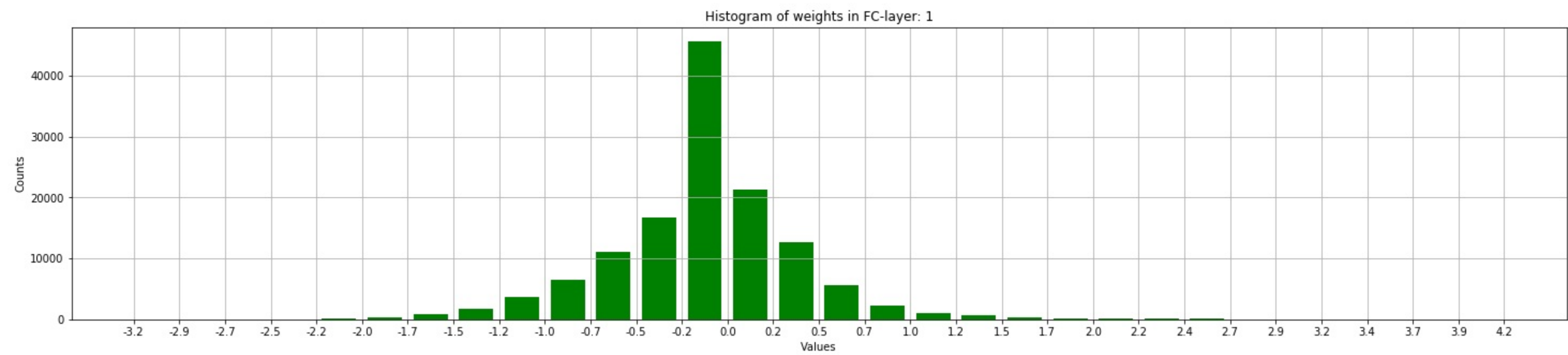
=====  
Conv. Layer: 5  
Minimum of weights: -4.0948877  
Maximum of weights: 4.1808672  
Average value of weights: 0.0055686  
Variance of weights: 0.5415530



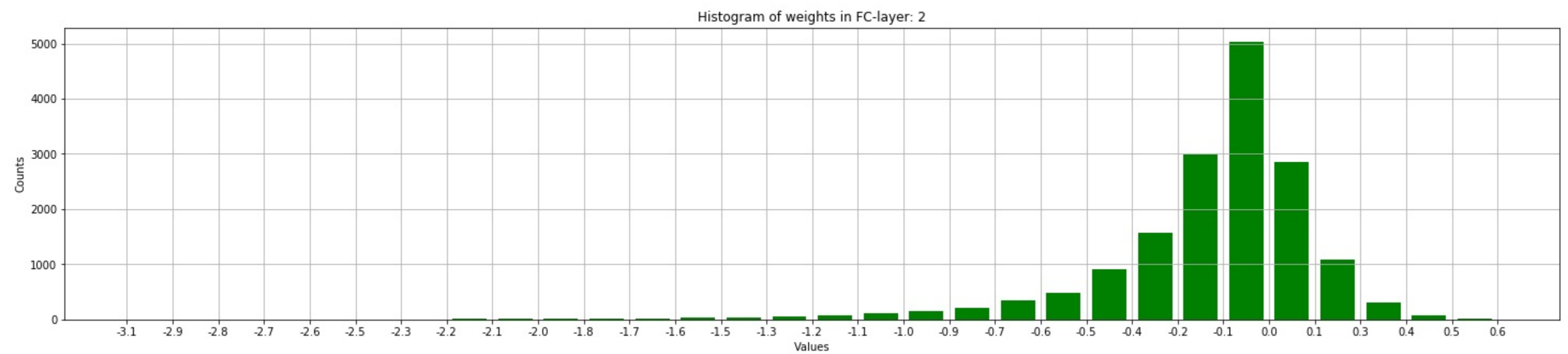
=====  
Conv. Layer: 6  
Minimum of weights: -3.9567604  
Maximum of weights: 3.5482860  
Average value of weights: 0.0016828  
Variance of weights: 0.5689645



=====  
Fully connected Layer: 1  
Minimum of weights: -3.1398053  
Maximum of weights: 4.2100883  
Average value of weights: -0.0663840  
Variance of weights: 0.5053805



=====  
Fully connected Layer: 2  
Minimum of weights: -3.0179267  
Maximum of weights: 0.6770992  
Average value of weights: -0.0958171  
Variance of weights: 0.2994871

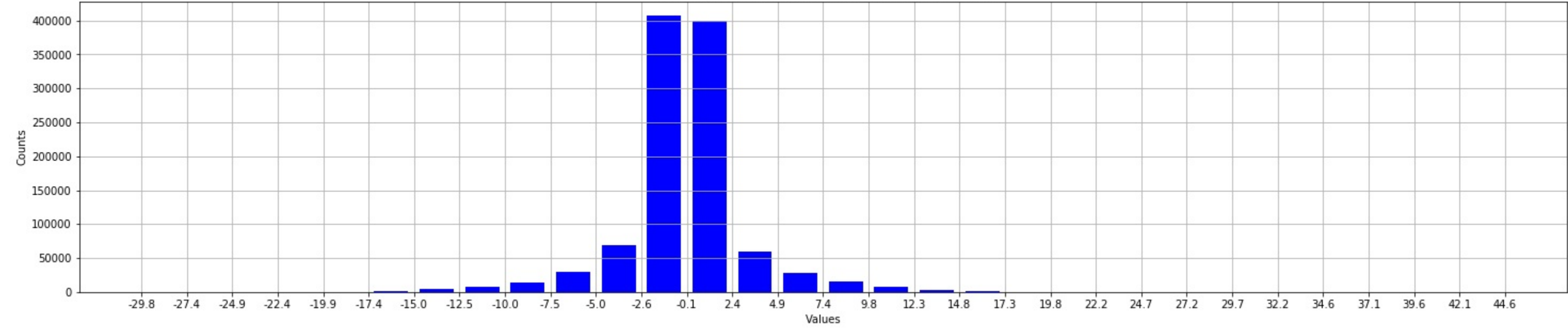


=====  
Statistical analysis of the intermediate signals:  
=====

=====  
Feature-values, layer 1 ,Conv. 1 output:  
Minimum of feature-values: -29.7912560  
Maximum of feature-values: 44.6093559  
Average value of feature-values: -0.1026003  
Variance of feature-values: 3.1774480

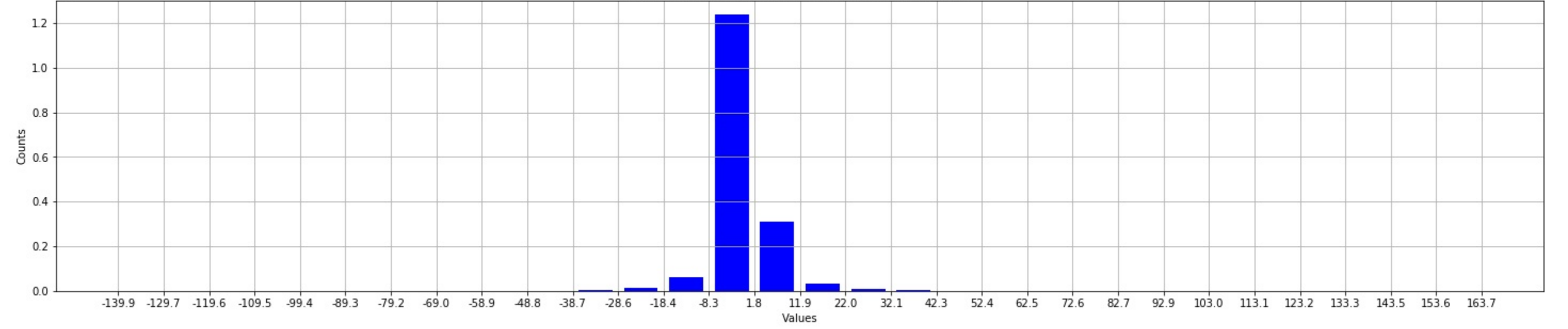


Histogram of feature-values, layer 1 , Conv. 1 output



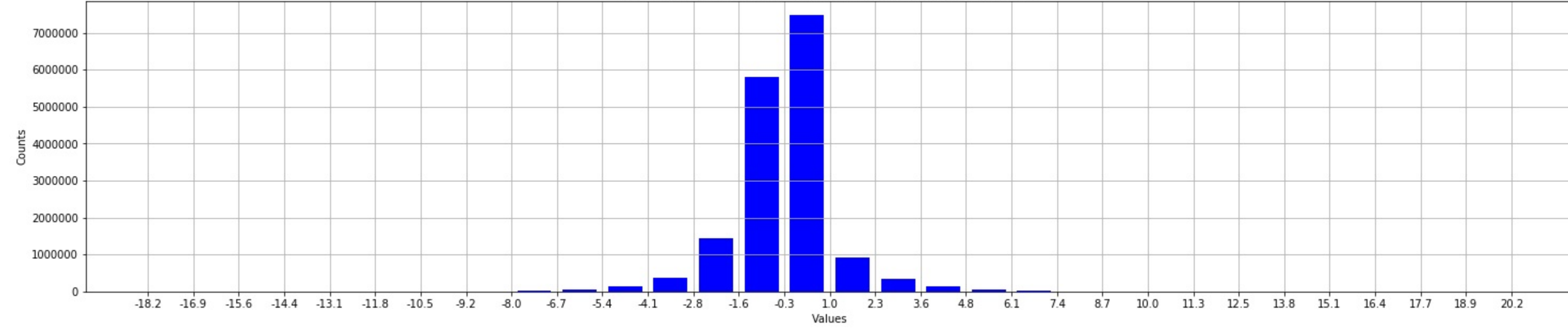
===== Feature-values, layer 1 ,Conv. 2 output:  
Minimum of feature-values: -139.8166656  
Maximum of feature-values: 163.7385101  
Average value of feature-values: -0.0175379  
Variance of feature-values: 5.9163141

Histogram of feature-values, layer 1 , Conv. 2 output



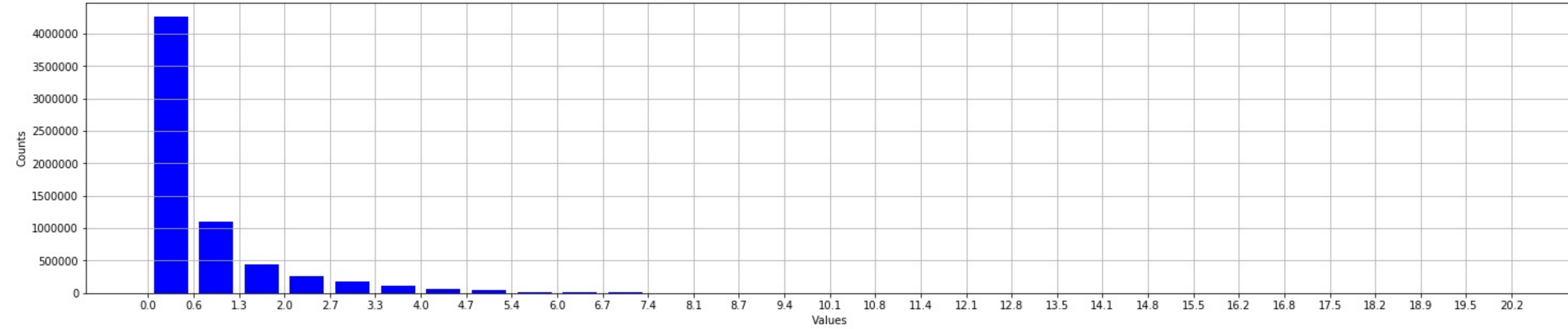
===== Feature-values, layer 1 ,batch normalization output:  
Minimum of feature-values: -18.1598396  
Maximum of feature-values: 20.2694416  
Average value of feature-values: -0.2499660  
Variance of feature-values: 1.3328955

Histogram of feature-values, layer 1 , batch normalization output



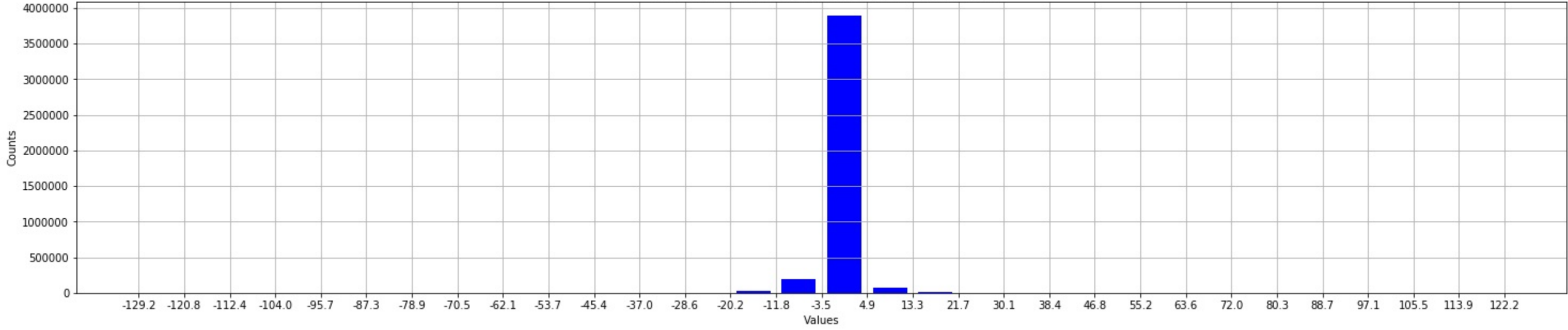
===== Feature-values, layer 1 ,ReLU output:  
Minimum of feature-values: 0.0000007  
Maximum of feature-values: 20.2694416  
Average value of feature-values: 0.7980773  
Variance of feature-values: 1.0657431

Histogram of feature-values, layer 1 , ReLU output



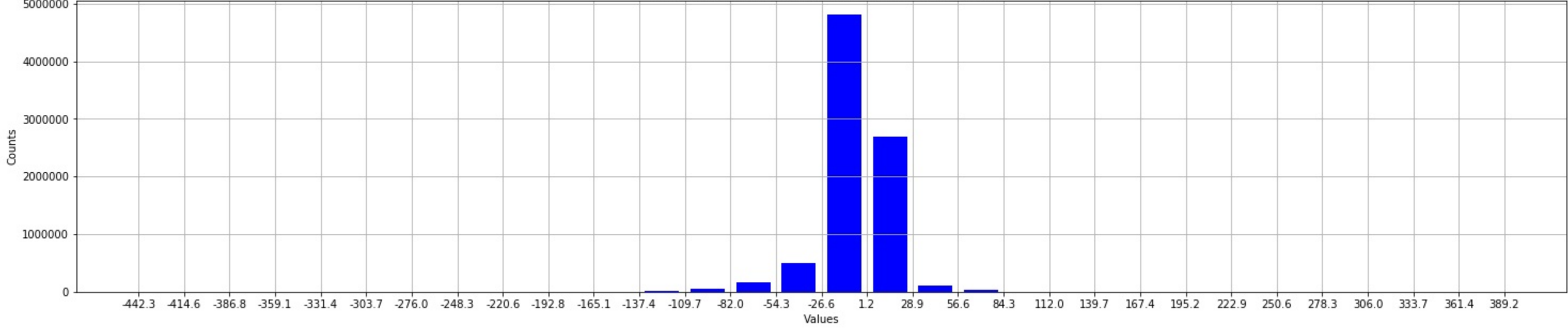
===== Feature-values, layer 2 ,Conv. 1 output:  
Minimum of feature-values: -129.1221466  
Maximum of feature-values: 122.2926865  
Average value of feature-values: -0.3301378  
Variance of feature-values: 2.5601006

Histogram of feature-values, layer 2 , Conv. 1 output



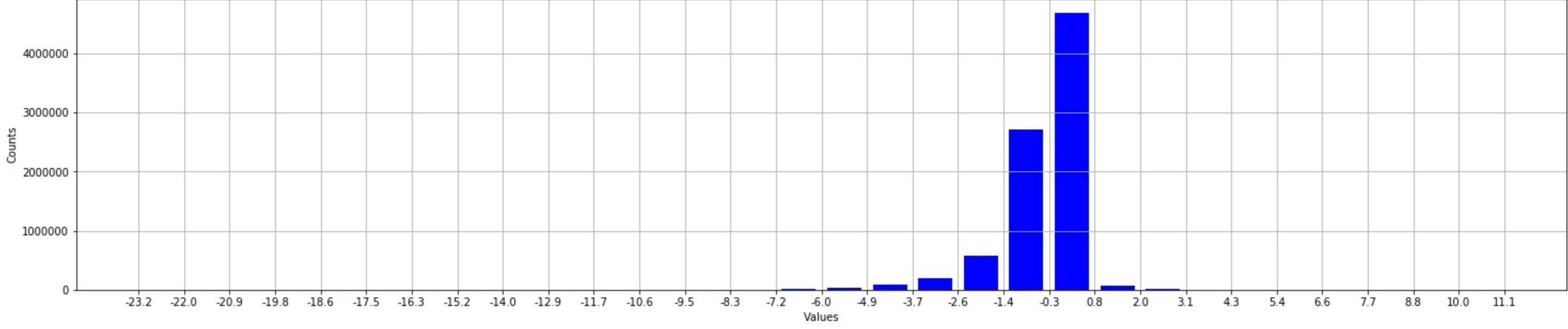
===== Feature-values, layer 2 ,Conv. 2 output:  
Minimum of feature-values: -442.2140808  
Maximum of feature-values: 389.2077942  
Average value of feature-values: -5.1846013  
Variance of feature-values: 18.8747444

Histogram of feature-values, layer 2 , Conv. 2 output



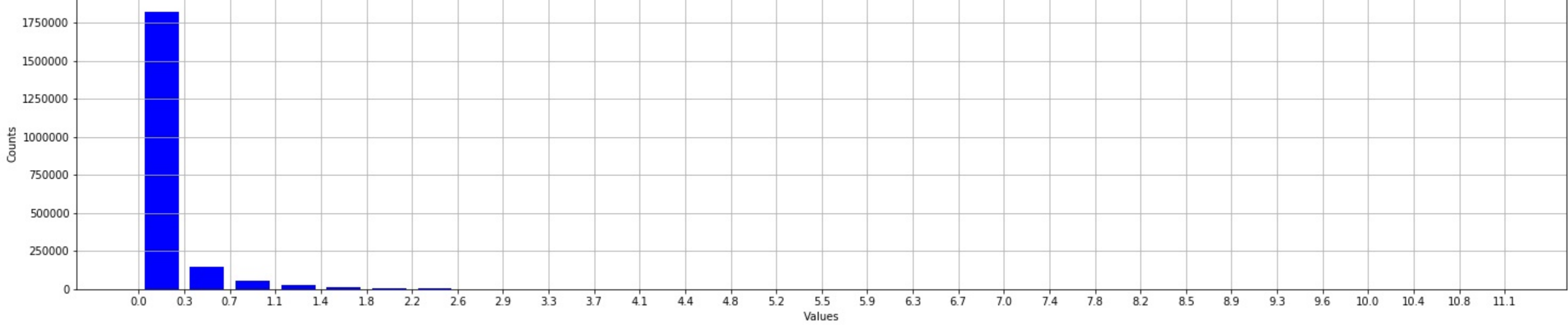
===== Feature-values, layer 2 ,batch normalization output:  
Minimum of feature-values: -23.1320820  
Maximum of feature-values: 11.1865282  
Average value of feature-values: -0.4765485  
Variance of feature-values: 0.9216570

Histogram of feature-values, layer 2 , batch normalization output

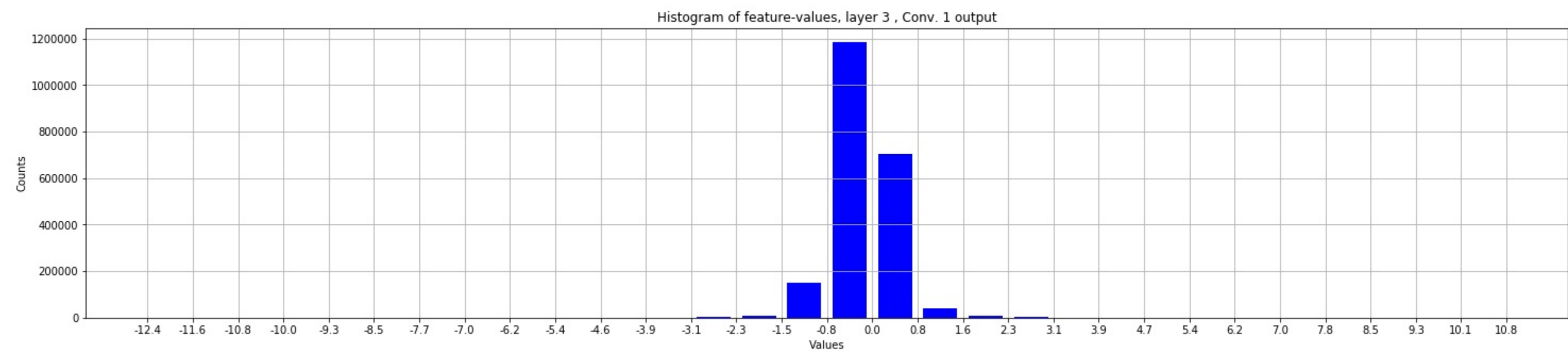


===== Feature-values, layer 2 ,ReLU output:  
Minimum of feature-values: 0.0000001  
Maximum of feature-values: 11.1865282  
Average value of feature-values: 0.1831227  
Variance of feature-values: 0.3105477

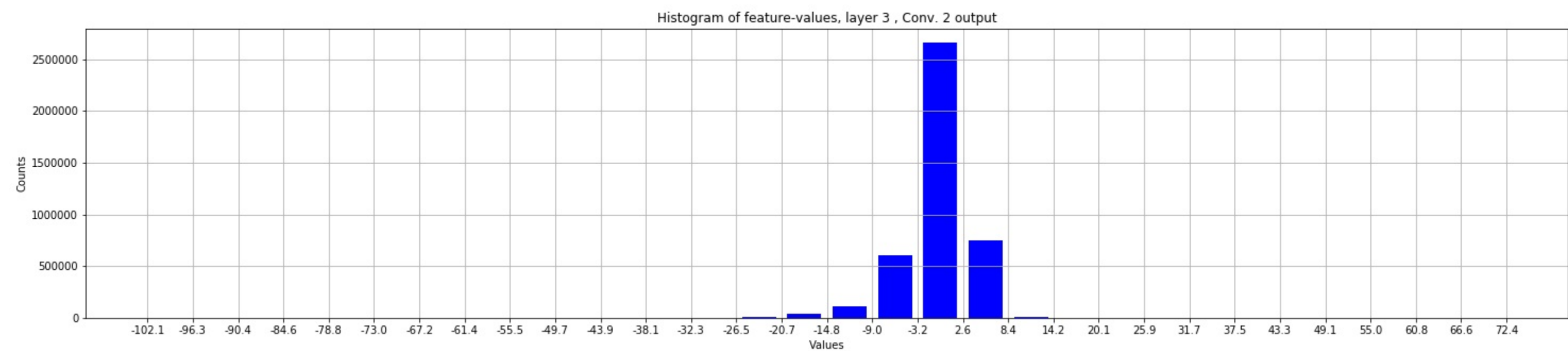
Histogram of feature-values, layer 2 , ReLU output



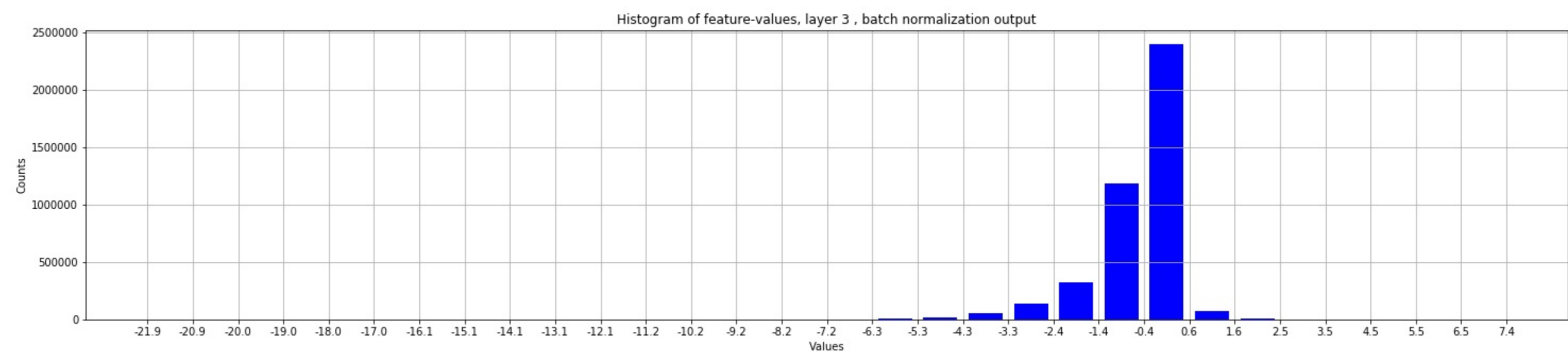
===== Feature-values, layer 3 ,Conv. 1 output:  
Minimum of feature-values: -12.3188953  
Maximum of feature-values: 10.8993044  
Average value of feature-values: -0.0852773  
Variance of feature-values: 0.5046849



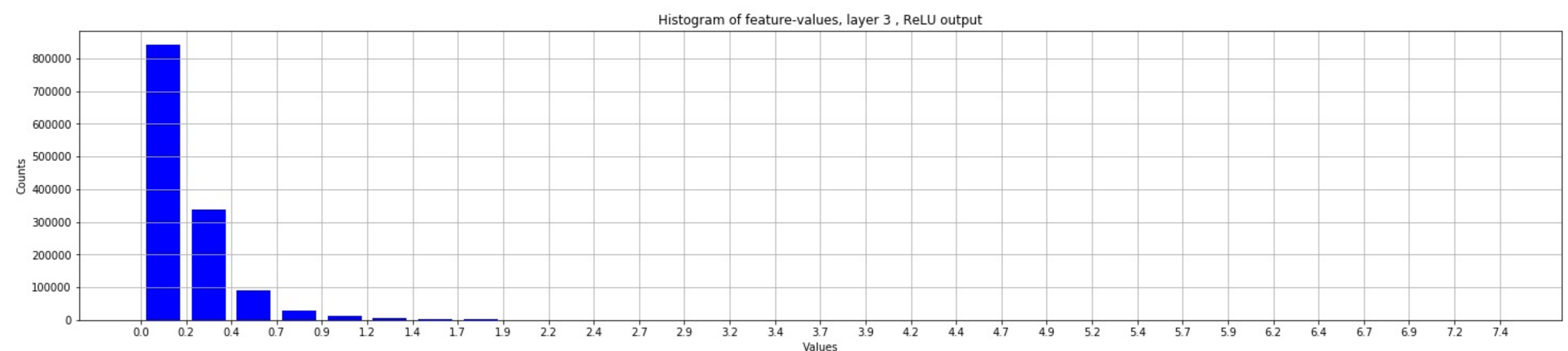
===== Feature-values, layer 3 ,Conv. 2 output:  
Minimum of feature-values: -102.0276871  
Maximum of feature-values: 72.4552383  
Average value of feature-values: -0.4983881  
Variance of feature-values: 4.0902452



===== Feature-values, layer 3 ,batch normalization output:  
Minimum of feature-values: -21.8711357  
Maximum of feature-values: 7.4836397  
Average value of feature-values: -0.4672673  
Variance of feature-values: 0.9083064

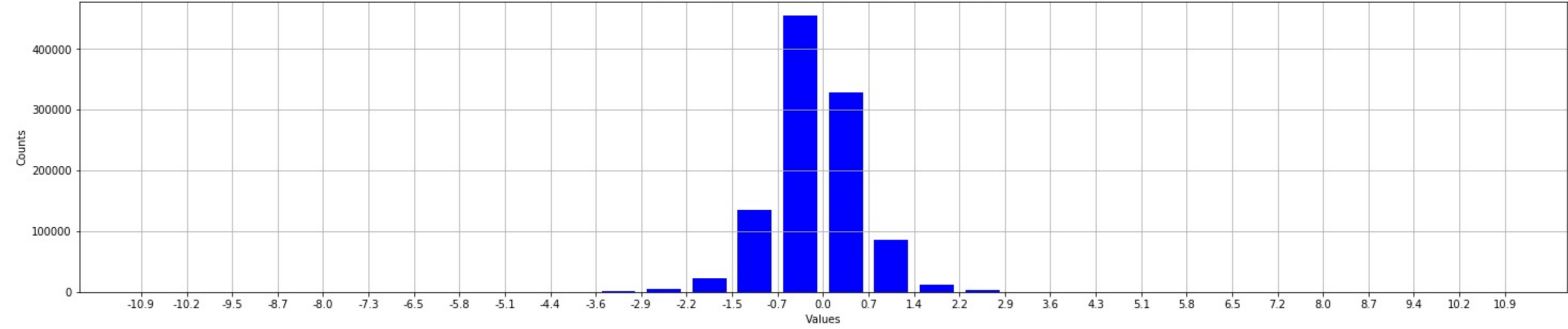


===== Feature-values, layer 3 ,ReLU output:  
Minimum of feature-values: 0.0000005  
Maximum of feature-values: 7.4836397  
Average value of feature-values: 0.2437660  
Variance of feature-values: 0.2365261



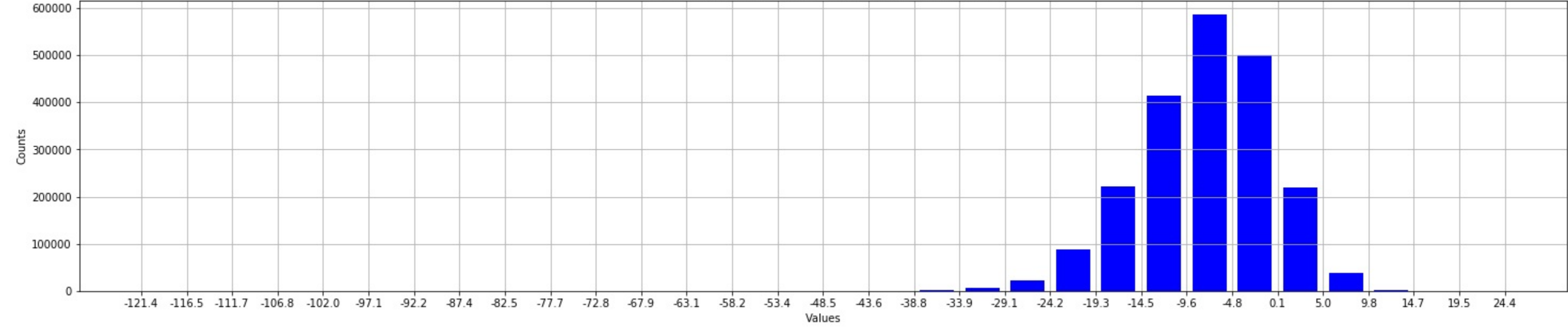
===== Feature-values, layer 4 ,Conv. 1 output:  
Minimum of feature-values: -10.8536139  
Maximum of feature-values: 10.9304829  
Average value of feature-values: -0.0843354  
Variance of feature-values: 0.6781013

Histogram of feature-values, layer 4 , Conv. 1 output



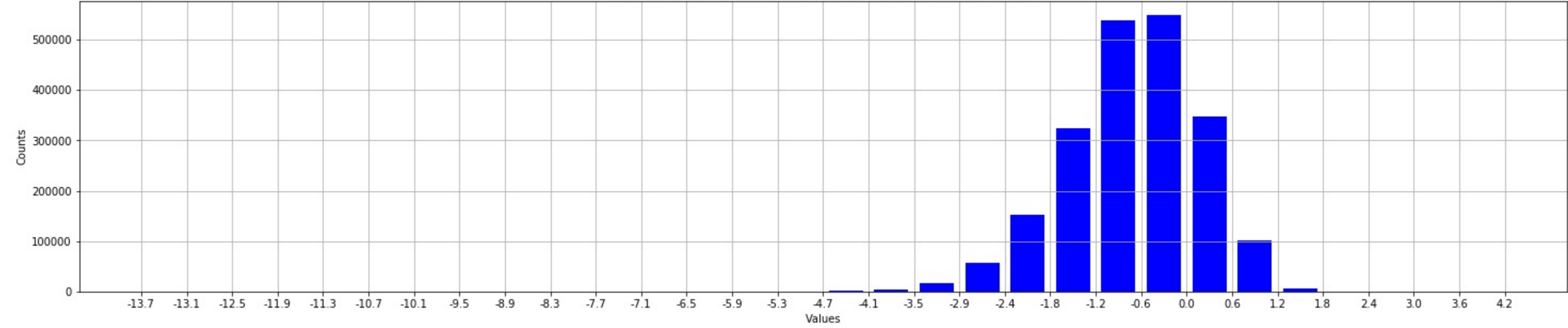
===== Feature-values, layer 4 ,Conv. 2 output:  
Minimum of feature-values: -121.3520813  
Maximum of feature-values: 24.4509182  
Average value of feature-values: -7.5850410  
Variance of feature-values: 6.9239483

Histogram of feature-values, layer 4 , Conv. 2 output



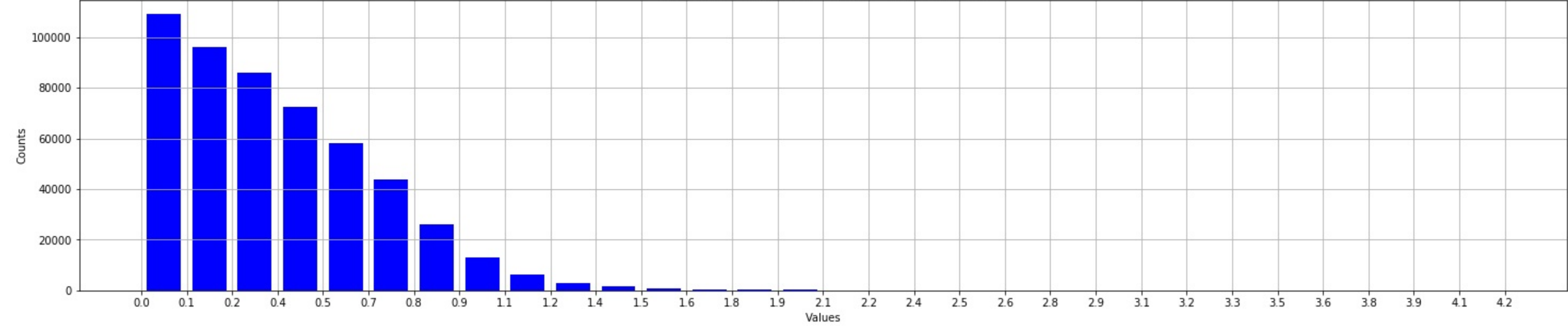
===== Feature-values, layer 4 ,batch normalization output:  
Minimum of feature-values: -13.6144218  
Maximum of feature-values: 4.2477922  
Average value of feature-values: -0.6210883  
Variance of feature-values: 0.8689694

Histogram of feature-values, layer 4 , batch normalization output



===== Feature-values, layer 4 ,ReLU output:  
Minimum of feature-values: 0.0000022  
Maximum of feature-values: 4.2477922  
Average value of feature-values: 0.4244087  
Variance of feature-values: 0.3138603

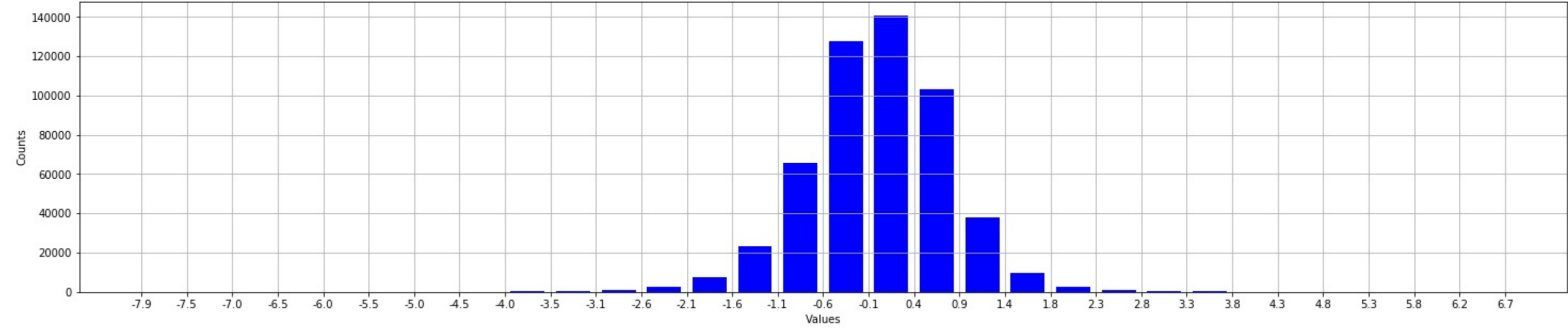
Histogram of feature-values, layer 4 , ReLU output



===== Feature-values, layer 5 ,Conv. 1 output:  
Minimum of feature-values: -7.8931351  
Maximum of feature-values: 6.7803674  
Average value of feature-values: 0.0313304  
Variance of feature-values: 0.7493501

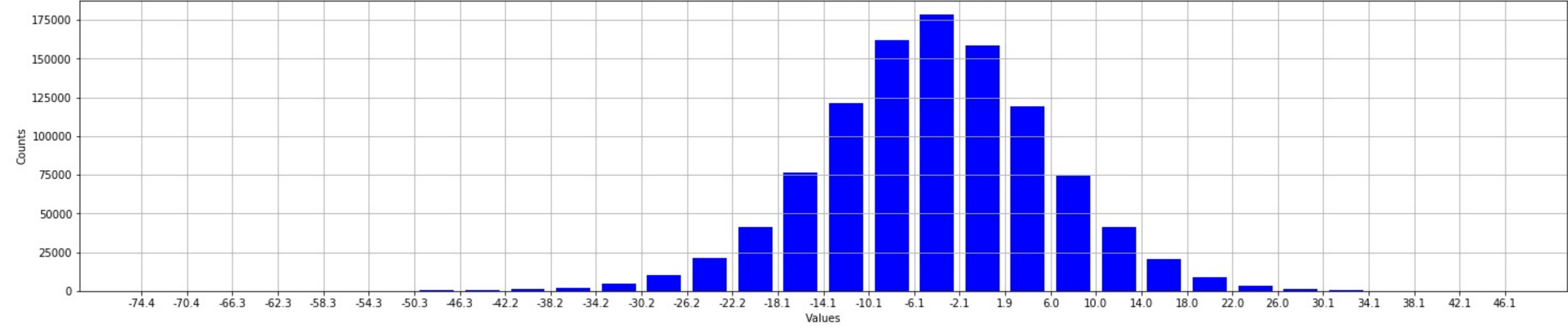


Histogram of feature-values, layer 5 , Conv. 1 output



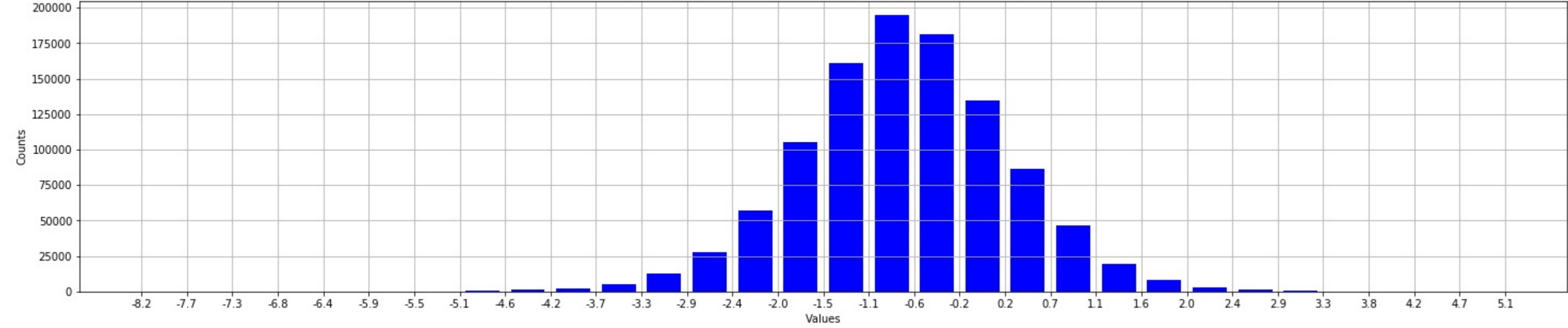
===== Feature-values, layer 5 ,Conv. 2 output:  
Minimum of feature-values: -74.3217239  
Maximum of feature-values: 46.1651535  
Average value of feature-values: -4.2845340  
Variance of feature-values: 10.0076628

Histogram of feature-values, layer 5 , Conv. 2 output



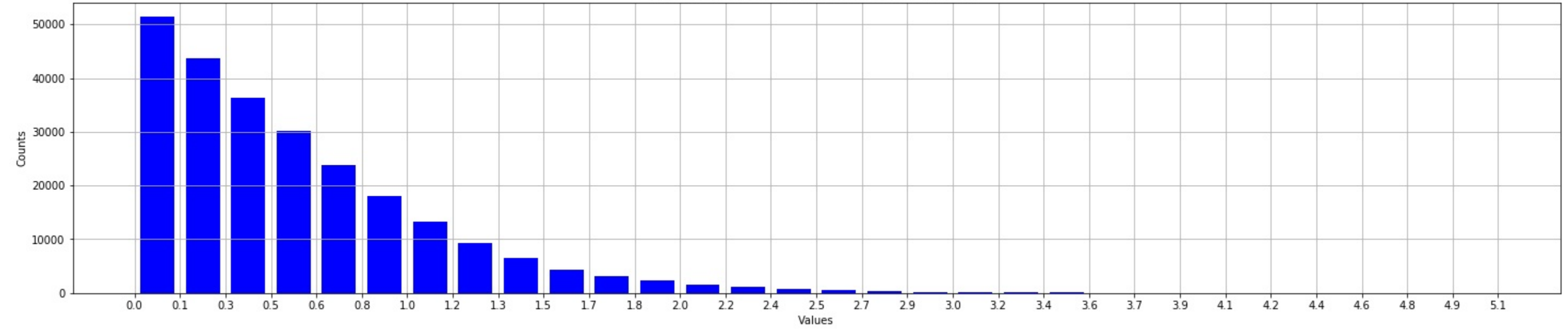
===== Feature-values, layer 5 ,batch normalization output:  
Minimum of feature-values: -8.1038589  
Maximum of feature-values: 5.1482258  
Average value of feature-values: -0.6873527  
Variance of feature-values: 1.0059863

Histogram of feature-values, layer 5 , batch normalization output



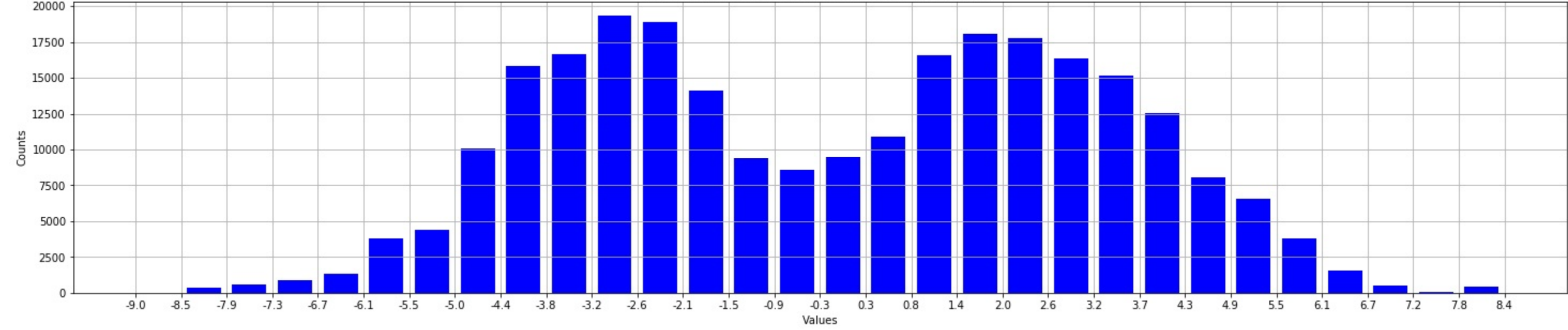
===== Feature-values, layer 5 ,ReLU output:  
Minimum of feature-values: 0.0000010  
Maximum of feature-values: 5.1482258  
Average value of feature-values: 0.6094315  
Variance of feature-values: 0.5290701

Histogram of feature-values, layer 5 , ReLU output



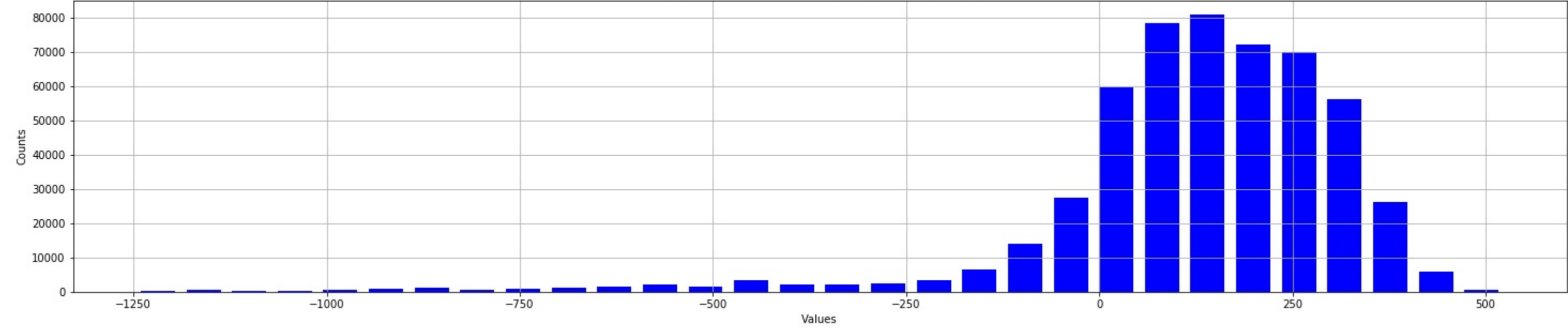
===== Feature-values, layer 6 ,Conv. 1 output:  
Minimum of feature-values: -8.9857683  
Maximum of feature-values: 8.4469433  
Average value of feature-values: -0.0404695  
Variance of feature-values: 3.2198601

Histogram of feature-values, layer 6 , Conv. 1 output



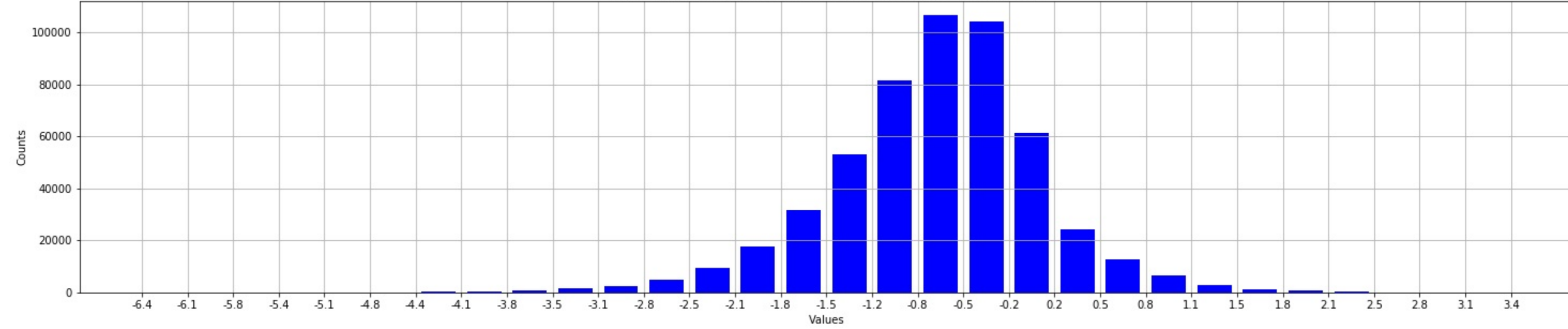
===== Feature-values, layer 6 ,Conv. 2 output:  
Minimum of feature-values: -1248.0252686  
Maximum of feature-values: 524.3761597  
Average value of feature-values: 123.7501526  
Variance of feature-values: 202.5832520

Histogram of feature-values, layer 6 , Conv. 2 output



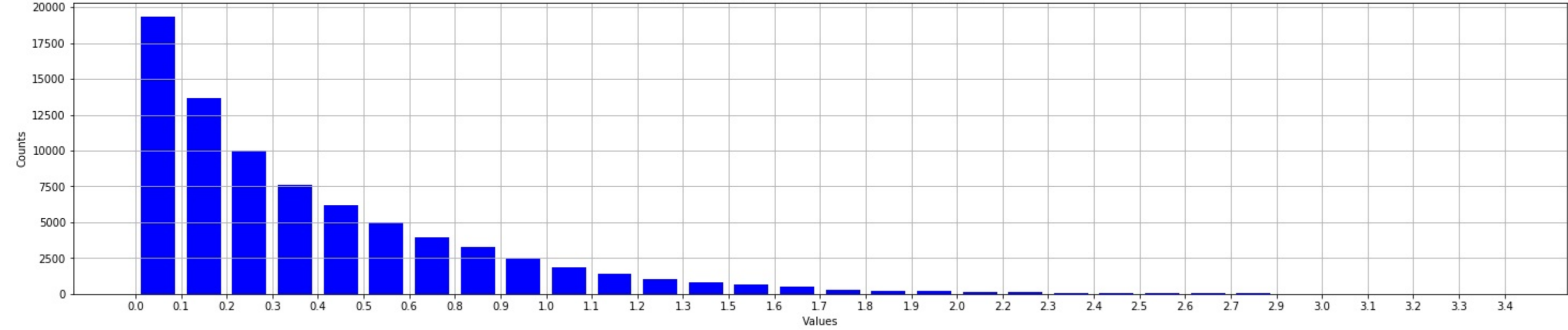
===== Feature-values, layer 6 ,batch normalization output:  
Minimum of feature-values: -6.3641953  
Maximum of feature-values: 3.4977722  
Average value of feature-values: -0.6462038  
Variance of feature-values: 0.7518793

Histogram of feature-values, layer 6 , batch normalization output



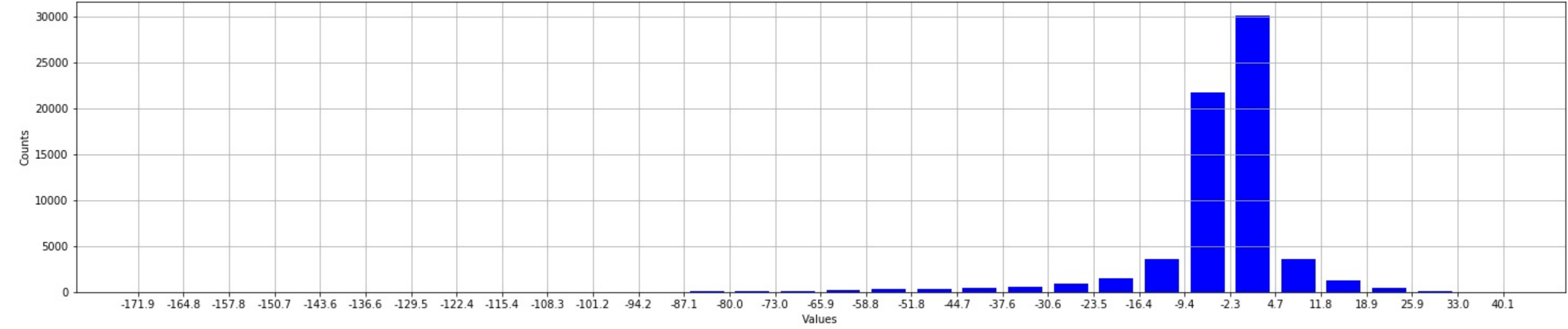
===== Feature-values, layer 6 ,ReLU output:  
Minimum of feature-values: 0.0000062  
Maximum of feature-values: 3.4977722  
Average value of feature-values: 0.4430953  
Variance of feature-values: 0.4345212

Histogram of feature-values, layer 6 , ReLU output



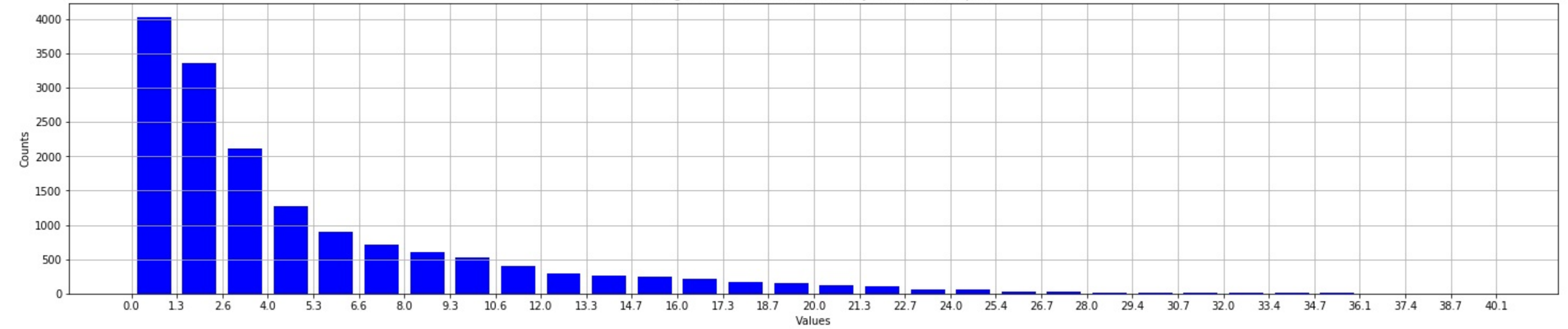
===== Feature-values, FC layer 1 ,Linear layer output:  
Minimum of feature-values: -171.8381653  
Maximum of feature-values: 40.1236763  
Average value of feature-values: -3.8969889  
Variance of feature-values: 11.6884527

Histogram of feature-values, FC layer 1 , Linear layer output



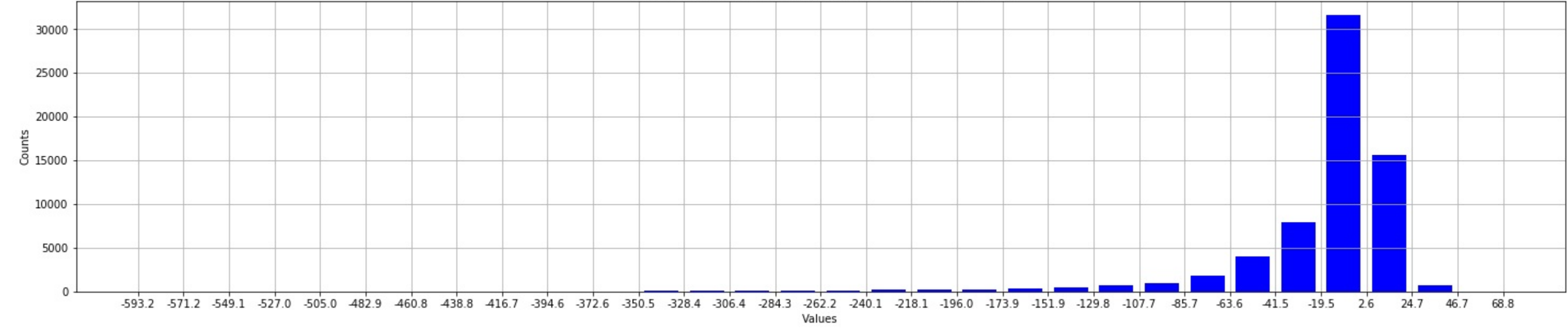
===== Feature-values, FC layer 1 ,ReLU output:  
Minimum of feature-values: 0.0002006  
Maximum of feature-values: 40.1236763  
Average value of feature-values: 5.1100726  
Variance of feature-values: 5.6298699

Histogram of feature-values, FC layer 1 , ReLU output



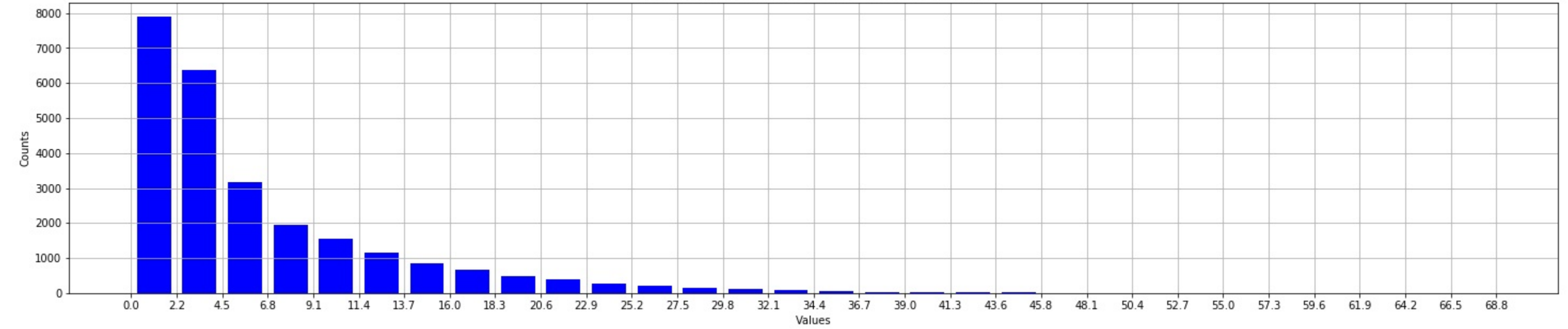
===== Feature-values, FC layer 4 ,Linear layer output:  
Minimum of feature-values: -593.1761475  
Maximum of feature-values: 68.8441162  
Average value of feature-values: -17.6684818  
Variance of feature-values: 43.7324715

Histogram of feature-values, FC layer 4 , Linear layer output



===== Feature-values, FC layer 4 ,ReLU output:  
Minimum of feature-values: 0.0003807  
Maximum of feature-values: 68.8441162  
Average value of feature-values: 6.4858184  
Variance of feature-values: 7.0906758

Histogram of feature-values, FC layer 4 , ReLU output



=====

we performed a fix-point quantization to int8 format.

There is only one de-quantization unit in the network in the last layer to get back the output

In each layer a scale factor is determined for each output channel.  
For example, a conv-layer with 32 inputs and 64 outputs has 64 different scale-factors.

Accuracy after using training-aware fixpoint quantization:  
TP: 99.02    FP: 6.03    AF\_threshold = 3  
TP: 98.77    FP: 2.81    AF\_threshold = 7

```
=====
Replacing ReLU activation functions with Sigmoid functions (floating point):
=====
```

Experiment 1:  
All functions in convolution and the fully connected layers are replaced by sigmoids:  
TP: 90.84    FP: 25.76    AF\_threshold = 3

Experiment 2:  
Only functions in the fully connected layers are replaced by sigmoids:  
TP: 99.02    FP: 3.14    AF\_threshold = 3

In [100]: