<https://brian2genn.readthedocs.io/en/stable/introduction/index.html>

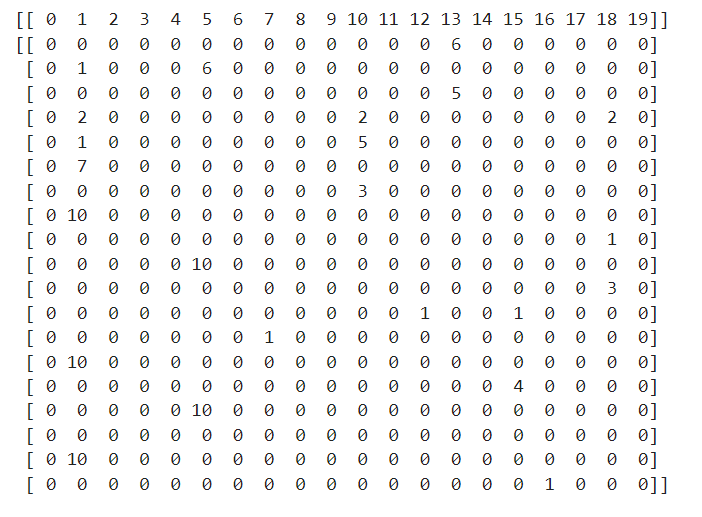
<https://github.com/brian-team/brian2cuda>

<https://brian2.readthedocs.io/en/stable/user/computation.html>

20 нейронов, 150 картинок - 30 мин

100 нейронов, 320 картинок - 5 часов

1. nn = 20 n\_train = 60 n\_test = 30, g\_const = 2.7, k\*0.9997



1. nn = 20 n\_train = 60 n\_test = 30, g\_const = 2.8, k\*0.9997

[[ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19]]

[[ 0 2 0 0 0 2 0 0 0 0 2 0 0 2 0 0 0 0 2 0]

[ 0 2 0 0 0 2 0 0 0 0 0 0 0 2 0 0 0 0 0 0]

[ 0 2 0 0 0 3 0 0 0 0 1 0 0 2 0 0 0 0 0 0]

[ 0 1 0 0 0 0 0 0 0 0 7 0 0 1 0 0 0 0 0 0]

[ 0 0 0 0 0 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0 0 0 0]

[ 0 0 0 0 0 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0 0 0 0]

[ 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]

[ 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 7 0]

[ 0 2 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0 0 0 0]

[ 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0 0 0 0]

[ 0 2 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0]

[ 0 2 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0]

[ 0 6 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0 0 0 0]

[ 0 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0 0 10 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0]

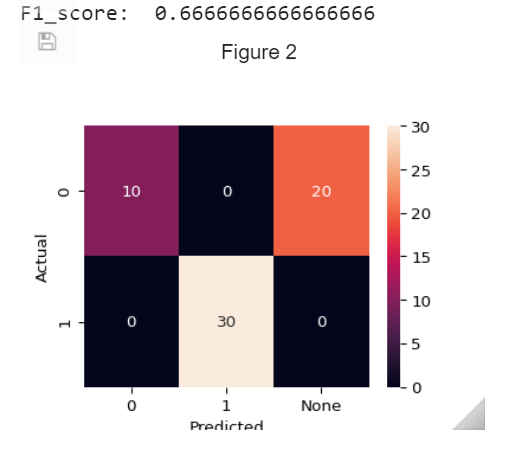
[ 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0]

[ 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0]

[ 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0]]



1. nn = 20 n\_train = 60 n\_test = 30, g\_const = 2.9, k\*0.9997

[[ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19]]

[[ 0 1 0 0 0 7 0 0 0 0 0 0 0 1 0 0 0 0 0 0]

[ 0 2 0 0 0 0 0 0 0 0 1 0 0 2 0 0 0 0 2 0]

[ 0 0 0 0 0 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[ 0 2 0 0 0 0 0 0 0 0 1 0 0 2 0 0 0 0 2 0]

[ 0 5 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0 0 0]

[ 0 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 2 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0 0 1 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 2 0 0 0 0 2 0 0 0 2 1 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0]

[ 0 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 2 1 0 0 0 2 0 0 0 2 1 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0]

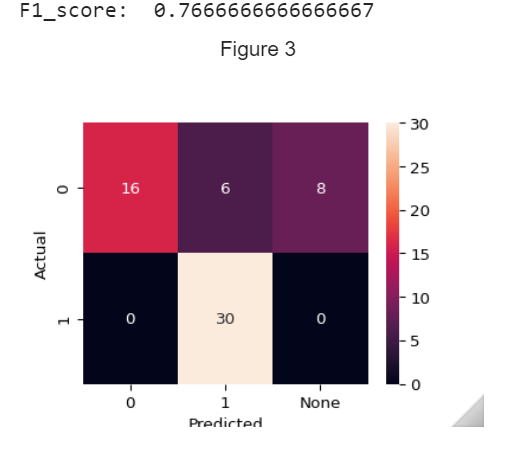
[ 0 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0]

[ 0 0 0 0 2 0 0 2 2 0 0 0 2 0 0 0 2 1 0 0]

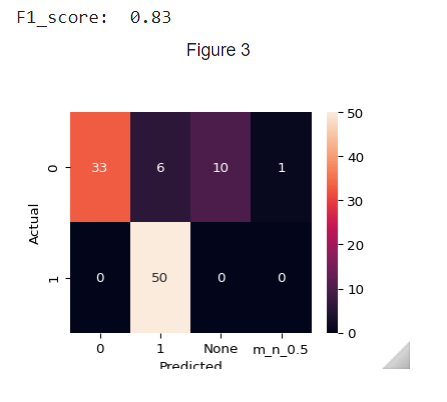
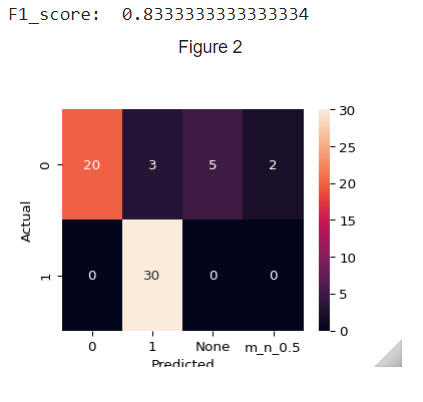
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0]

[ 0 0 0 0 0 0 0 1 5 0 0 0 1 0 0 0 1 0 0 0]]

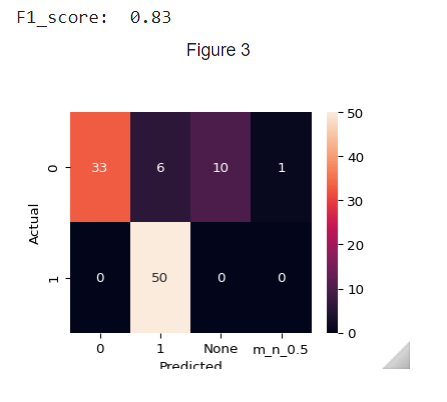
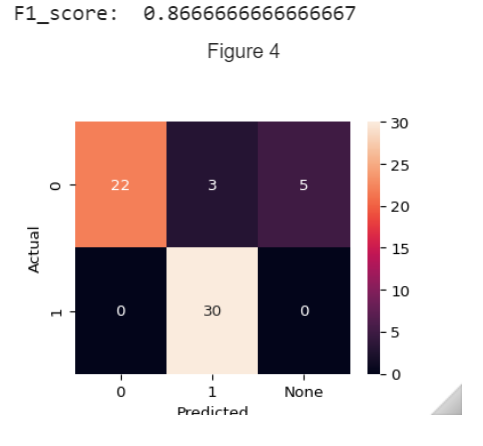


**Part 2!!!!!!!!!!!**

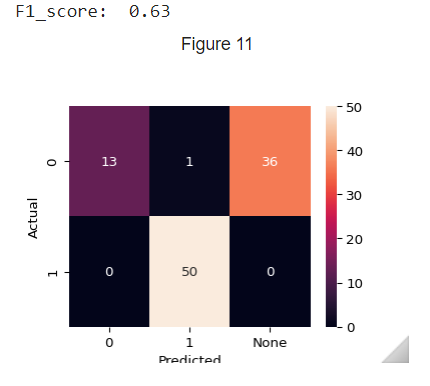
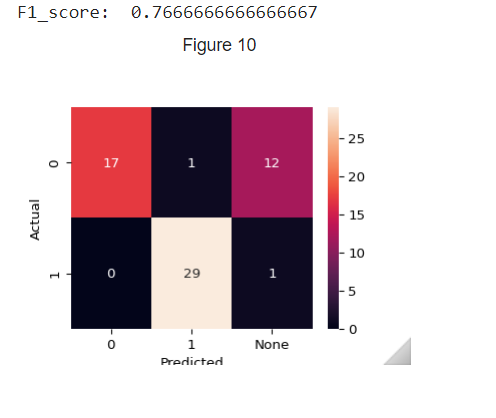
1. nn = 20 n\_train = 60 n\_test = 100, g\_const = 2.8(train+clf),3.0(test)



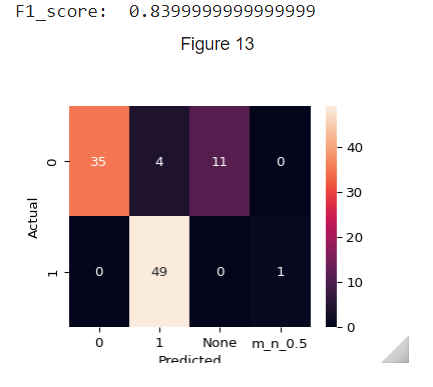
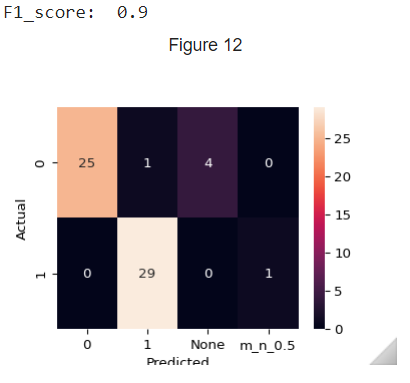
1. nn = 20 n\_train = 60 n\_test = 100, g\_const = 2.8(train),3.0(clf+test)



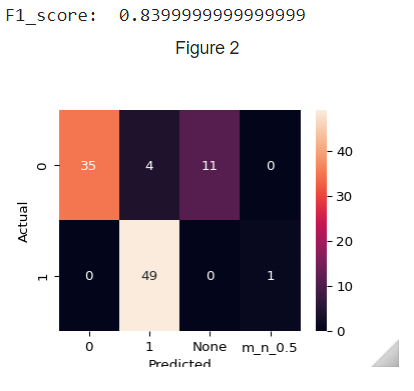
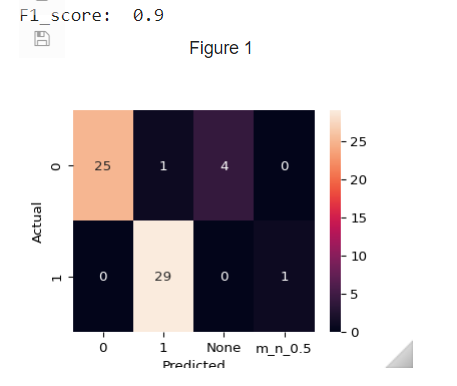
1. nn = 20 n\_train = 60 n\_test = 100, g\_const = 3.0(train),3.0(clf+test)



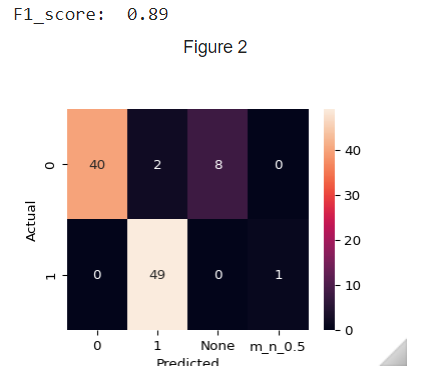
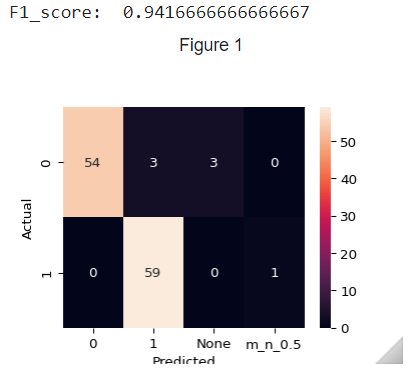
1. nn = 20 n\_train = 60 n\_test = 100, g\_const = 3.0(train+clf),3.1(test)



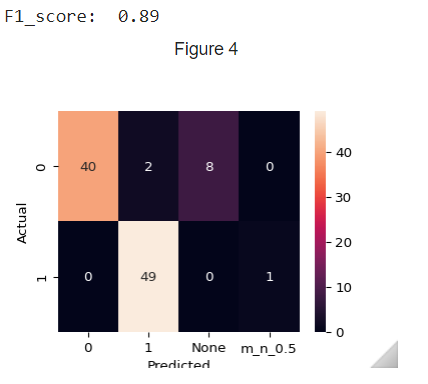
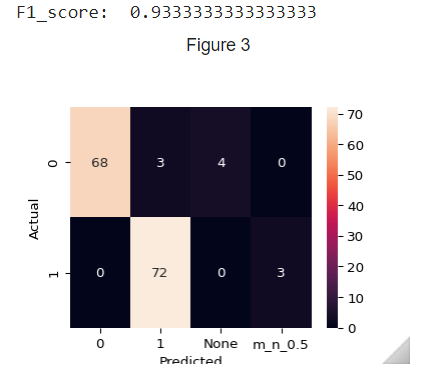
1. nn = 20 n\_train = 60 n\_test = 100, g\_const = 3.0(train),3.1(clf+test)



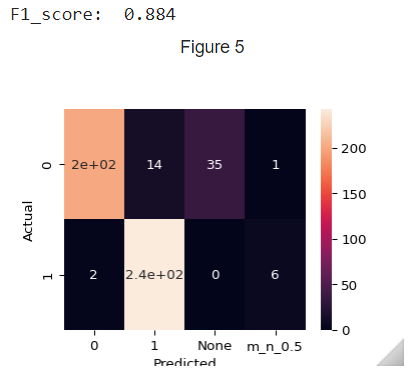
1. nn = 20 n\_train = 120 n\_test = 100, g\_const = 3.0(train),3.1(clf+test)



1. nn = 20 n\_train = 150 n\_test = 100, g\_const = 3.0(train),3.1(clf+test)

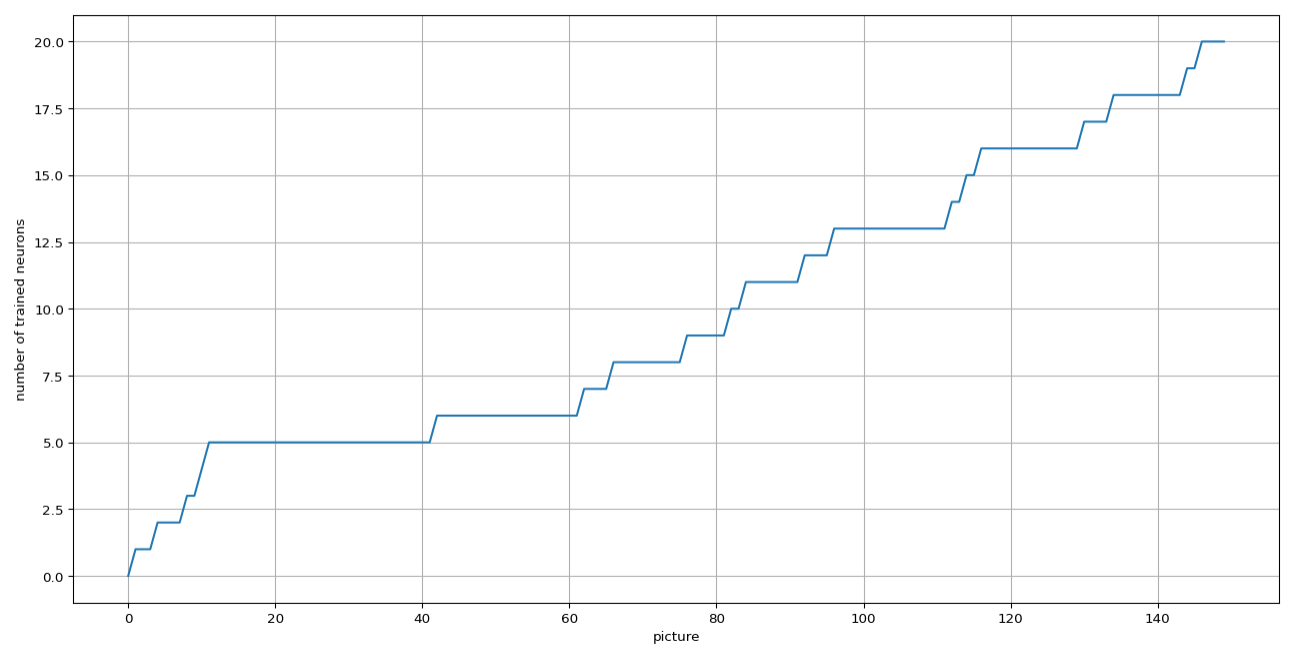


1. nn = 20 n\_train = 150 n\_test = 500, g\_const = 3.0(train),3.1(clf+test)

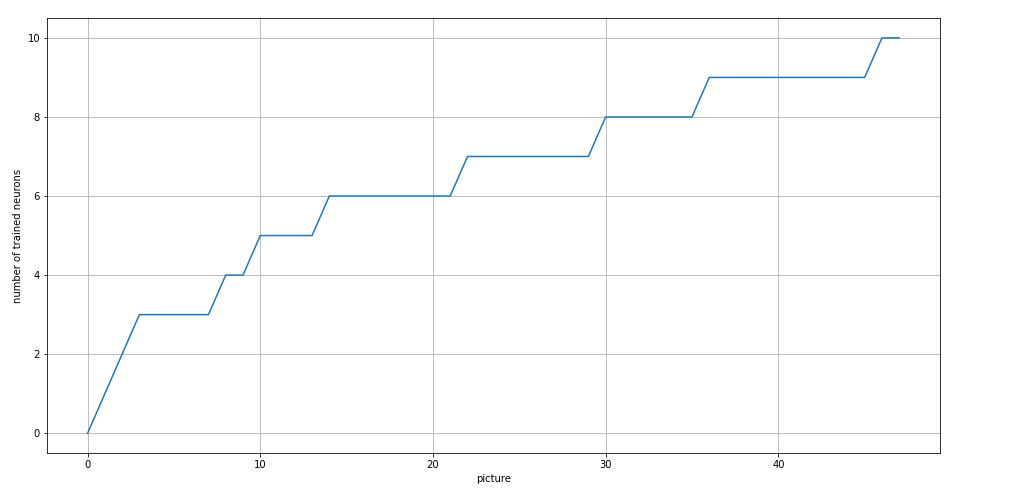
test|test:

1. nn = 100 n\_train = 320 g\_const = 3.0(train) time: 5h

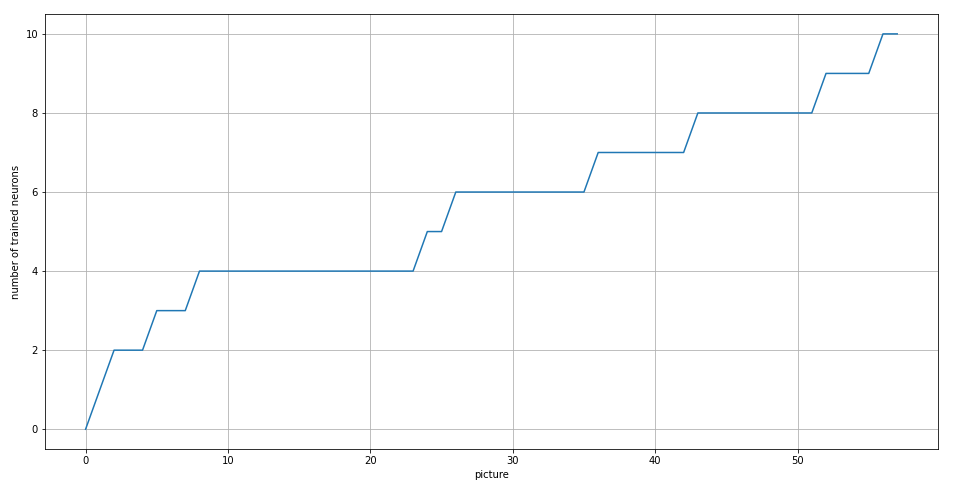
k\_reduce\_coef = 0.9997, nn = 20



k\_reduce\_coef = 0.9998, nn = 10



k\_reduce\_coef = 0.9999, nn = 10



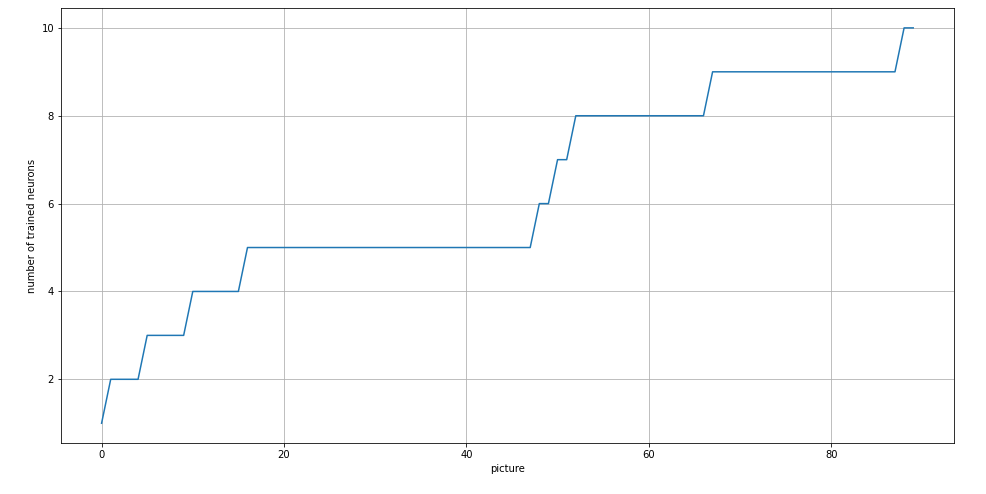
27.03.23

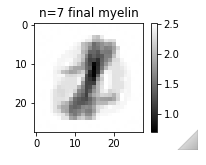
tau\_const = 3.5

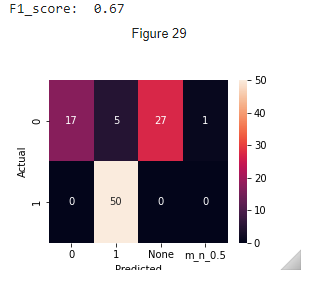
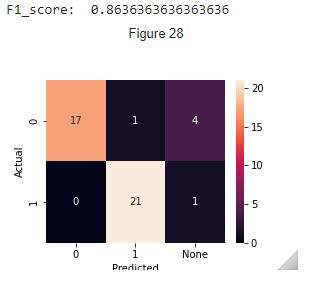
g\_const = 3.0 / ni

learning\_rate = 0.3

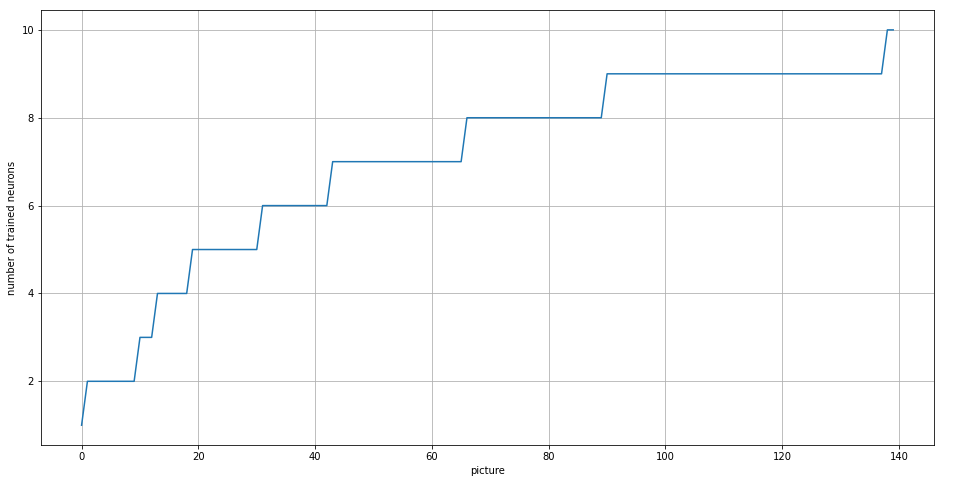
1. k\_reduce\_coef = 0.9999, nn = 10

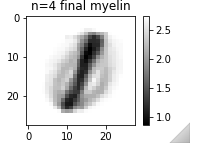


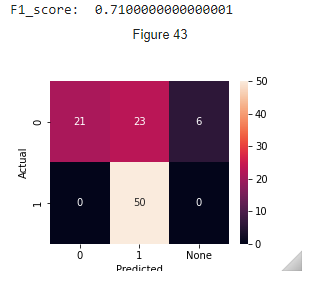
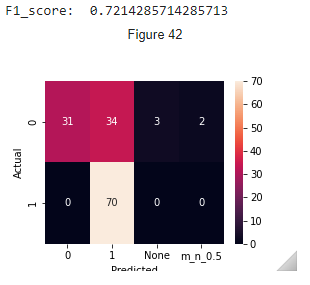
1 из 10 нейронов, реагирующий на единицу - было переобучение



1. k\_reduce\_coef = 0.99995, nn = 10



1 из 10 нейронов, реагирующий на единицу - было переобучение



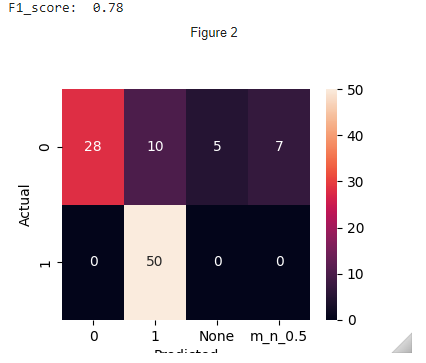
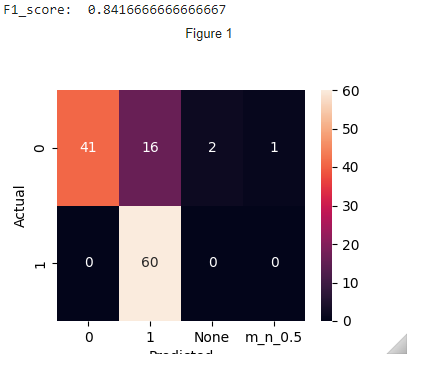
12.04.2023

1. nn = 20 n\_train = 120 n\_test = 100, k = 0.9998, g\_const = 3.0(train),3.1(clf+test)

Training time: 516.8402607440948

[ 8. 1. 6. 4. 6. 0. 1. 4. 0. 6. 6. 1. 16. 3. 2. 0. 5. 2. 0. 2.]

[ 0. 0. 60. 0. 0. 0. 0. 0. 0. 0. 0. 0. 35. 0. 17. 0. 0. 0. 0. 0.]

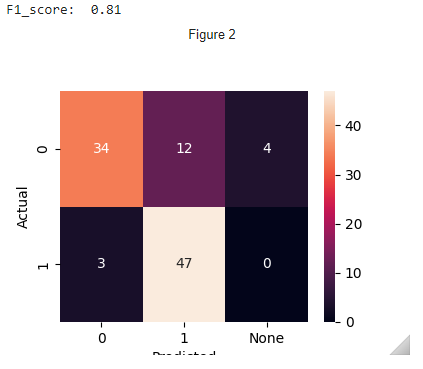
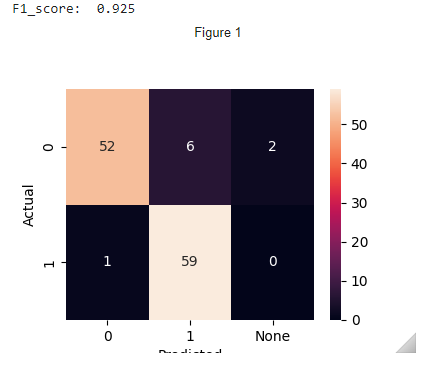


1. nn = 20 n\_train = 120 n\_test = 100, k = 0.9997, g\_const = 3.0(train),3.1(clf+test)

Training time: 837.0826835632324

[ 0. 0. 27. 0. 0. 0. 0. 6. 0. 11. 0. 6. 6. 2. 14. 0. 0. 0. 0. 1.]

[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 59. 0. 0. 1. 0. 0. 0. 0. 0.]

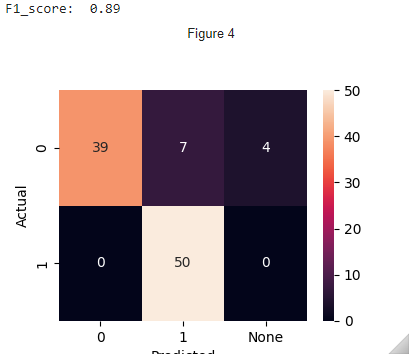
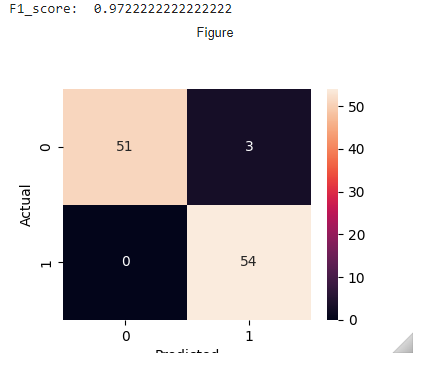


1. nn = 20 n\_train = 120 n\_test = 100, k = 0.9996, g\_const = 3.0(train),3.1(clf+test)

Training time: 452.24734234809875

[ 4. 4. 18. 4. 4. 1. 1. 5. 1. 13. 1. 1. 2. 2. 1. 1. 5. 2. 1. 2.]

[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 53. 0. 27. 0. 0. 0. 0. 0.]

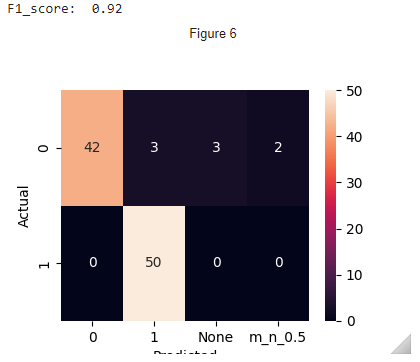
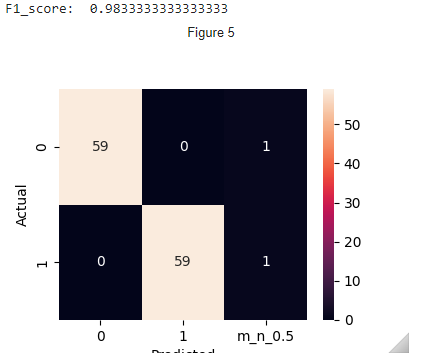


1. nn = 20 n\_train = 120 n\_test = 100, k = 0.9995, g\_const = 3.0(train),3.1(clf+test)

Training time: 513.8193271160126

[ 5. 0. 17. 2. 1. 0. 1. 3. 0. 11. 0. 1. 1. 1. 19. 0. 1. 3. 1. 4.]

[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 60. 0. 1. 0. 0. 0. 0. 0.]

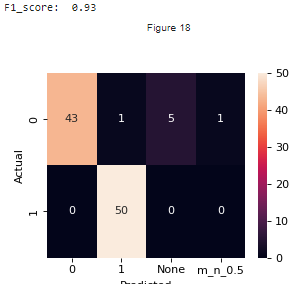
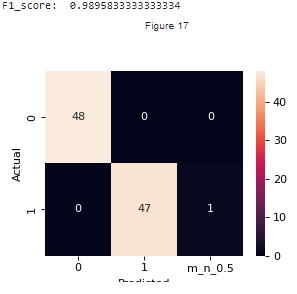


1. nn = 20 n\_train = 120 n\_test = 100, k = 0.99945, g\_const = 3.0(train),3.1(clf+test)

Training time: 402.336932182312

[ 1. 1. 13. 4. 1. 1. 1. 9. 1. 8. 2. 1. 1. 1. 13. 1. 7. 1. 2. 4.]

[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 48. 0. 1. 0. 0. 0. 0. 0.]

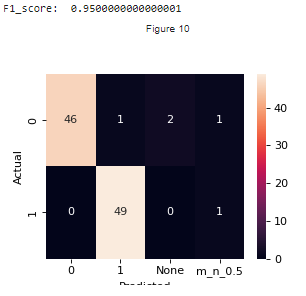
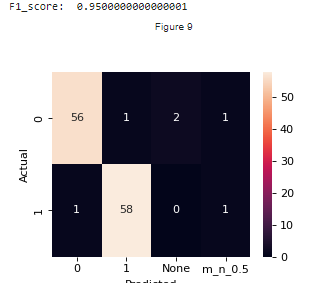


1. nn = 20 n\_train = 120 n\_test = 100, k = 0.9994, g\_const = 3.0(train),3.1(clf+test)

Training time: 509.4593915939331

[ 0. 0. 20. 3. 1. 0. 0. 3. 0. 14. 0. 1. 2. 0. 19. 0. 0. 1. 3. 1.]

[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 59. 0. 2. 0. 0. 0. 0. 0.]

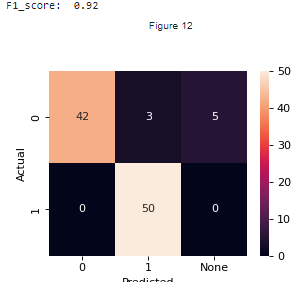
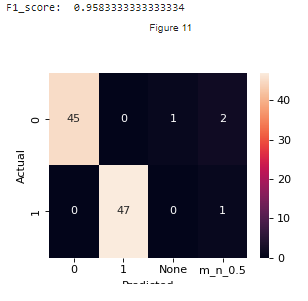


1. nn = 20 n\_train = 120 n\_test = 100, k = 0.9993, g\_const = 3.0(train),3.1(clf+test)

Training time: 411.2235026359558

[ 1. 1. 12. 3. 1. 1. 1. 9. 1. 7. 1. 1. 2. 2. 10. 1. 6. 2. 2. 1.]

[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 48. 0. 1. 0. 0. 0. 0. 0.]

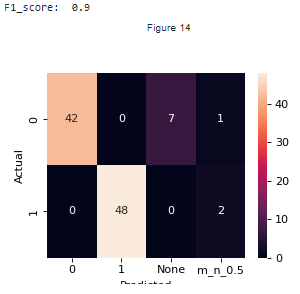
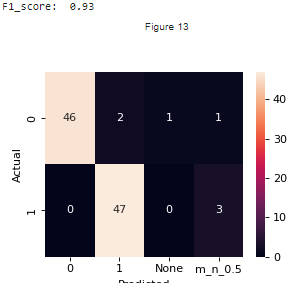


1. nn = 20 n\_train = 120 n\_test = 100, k = 0.9992, g\_const = 3.0(train),3.1(clf+test)

Training time: 425.7255325317383

[ 2. 1. 11. 3. 1. 1. 1. 5. 1. 5. 1. 3. 3. 1. 14. 1. 1. 1. 1. 3.]

[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 50. 0. 3. 0. 0. 0. 0. 0.]

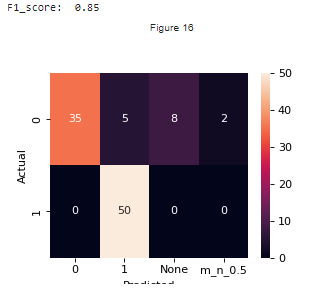
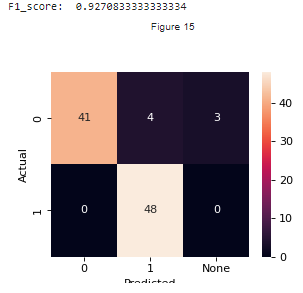


1. nn = 20 n\_train = 120 n\_test = 100, k = 0.9991, g\_const = 3.0(train),3.1(clf+test)

Training time: 407.02601289749146

[ 1. 2. 10. 3. 1. 1. 1. 7. 1. 4. 1. 2. 4. 1. 8. 1. 1. 0. 1. 3.]

[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 40. 0. 0. 0. 0. 11. 0. 0.]

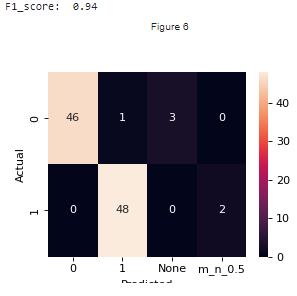
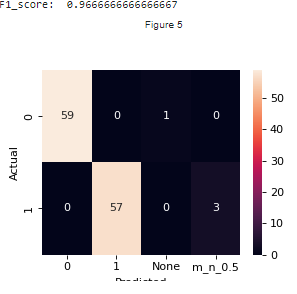


1. nn = 20 n\_train = 120 n\_test = 100, k = 0.99945, g\_const = 3.0(train), 3.1(clf+test) !shuffled!

Training time: 697.3508756160736

[ 3. 1. 8. 3. 2. 0. 0. 12. 0. 6. 0. 1. 26. 0. 13. 0. 4. 1. 0. 0.]

[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 0. 0. 0. 0. 0. 0. 60.]



**MNIST DATASET FULL**

1. nn = 20 n\_train = 120 n\_test = 100, k = 0.99945,

g\_const = 3.0(train), 3.1(clf+test)

Training time: 855.6375107765198

Actual number of training set: 120

{0: '2', 1: '3', 2: '3', 3: '4', 4: '4', 5: 'None', 6: 'None', 7: '6', 8: 'None', 9: '0', 10: 'None', 11: 'None', 12: '1', 13: '2', 14: '0', 15: 'None', 16: 'None', 17: 'None', 18: 'None', 19: '1'}

[[ 0. 0. 0. 0. 0. 0. 0. 1. 0. 2. 0. 0. 2. 0. 12. 0. 0. 0. 0. 0.]

[ 0. 0. 0. 0. 0. 0. 0. 10. 0. 0. 0. 0. 18. 0. 2. 0. 0. 0. 0. 8.]

[ 1. 0. 0. 0. 0. 0. 0. 2. 0. 0. 0. 0. 5. 1. 0. 0. 0. 0. 0. 0.]

[ 0. 1. 9. 0. 0. 0. 0. 0. 0. 0. 0. 0. 2. 0. 0. 0. 0. 0. 0. 0.]

[ 0. 0. 0. 1. 1. 0. 0. 5. 0. 0. 0. 0. 8. 0. 1. 0. 0. 0. 0. 8.]

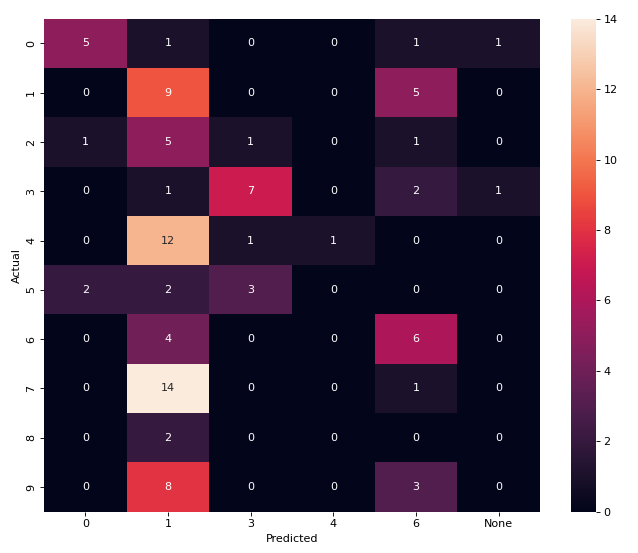
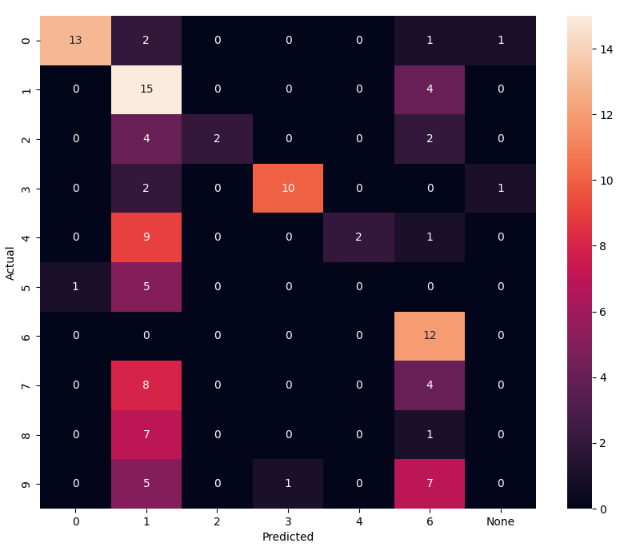
[ 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 5. 0. 3. 0. 0. 0. 0. 3.]

[ 0. 0. 0. 0. 0. 0. 0. 12. 0. 0. 0. 0. 0. 0. 6. 0. 0. 0. 0. 1.]

[ 0. 0. 0. 0. 0. 0. 0. 5. 0. 0. 0. 0. 12. 0. 2. 0. 0. 0. 0. 3.]

[ 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 8. 0. 1. 0. 0. 0. 0. 0.]

[ 0. 0. 1. 0. 0. 0. 0. 9. 0. 0. 0. 0. 12. 0. 1. 0. 0. 0. 0. 3.]]

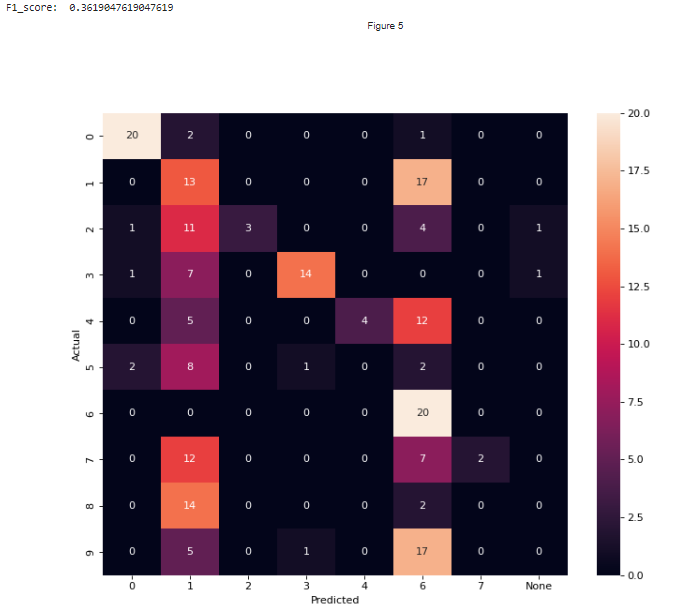
F1\_score: 0.45 F1\_score: 0.28

1. nn = 20 n\_train = 500 n\_test = 100, k = 0.99945,

g\_const = 3.0(train), 3.1(clf+test)

Training time: 1212.0478882789612

Actual number of training set: 210



F1-score (train): 0.3619 F1-score (test): 0.27

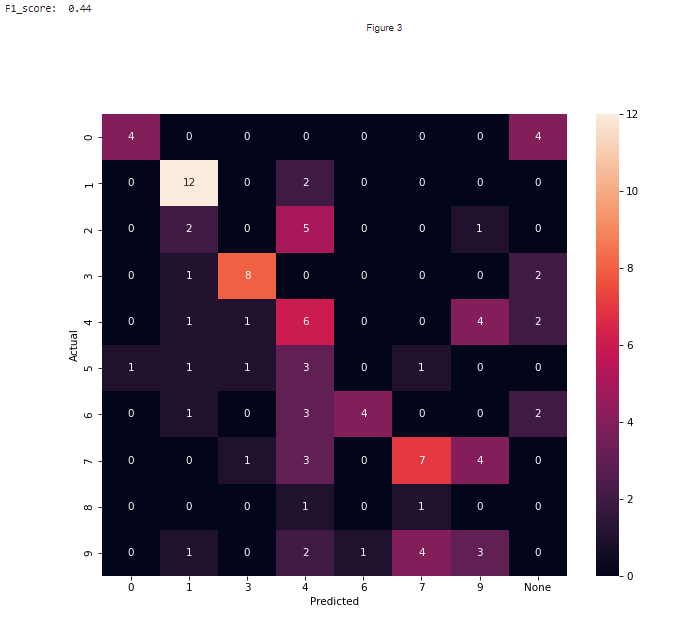
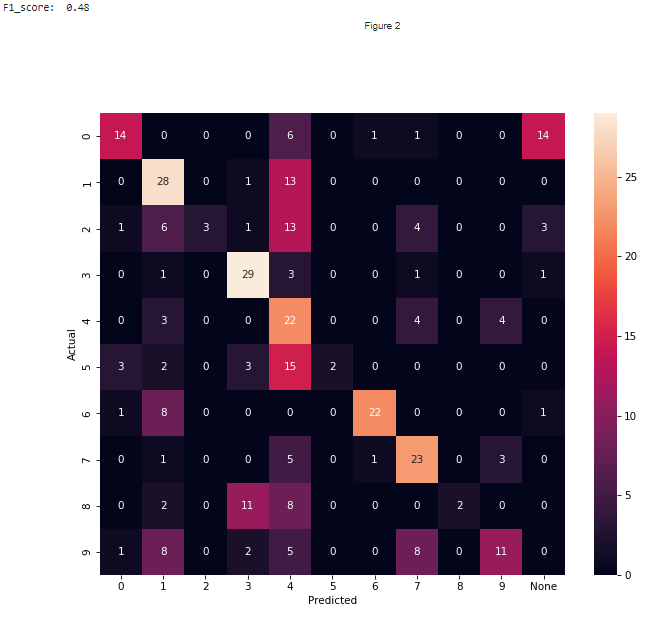
1. nn = 30 n\_train = 500 n\_test = 100, k = 0.99945,

g\_const = 3.0(train), 3.1(clf+test)

Training time: 2757.884792327881

Actual number of training set: 325

{0: '4', 1: '7', 2: '9', 3: '8', 4: '0', 5: '2', 6: '2', 7: '3', 8: '9', 9: '2', 10: '6', 11: '0', 12: '1', 13: '1', 14: '0', 15: '3', 16: '4', 17: '0', 18: '0', 19: '8', 20: '4', 21: '7', 22: '5', 23: '7', 24: '1', 25: '3', 26: '0', 27: '5', 28: '6', 29: '7'}



F1\_score\_train = 0.48 F1\_score\_train = 0.44

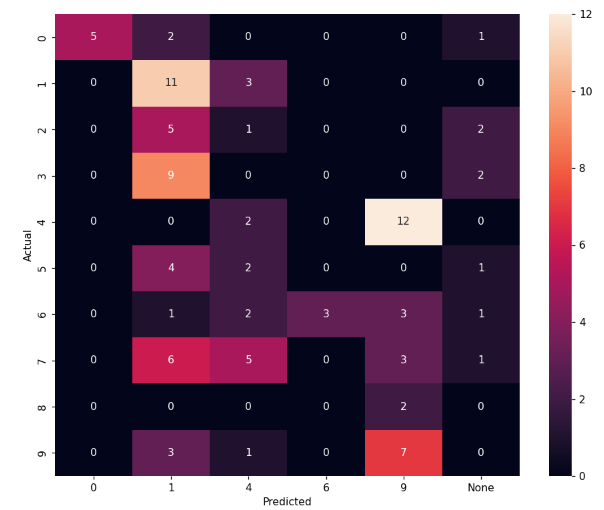
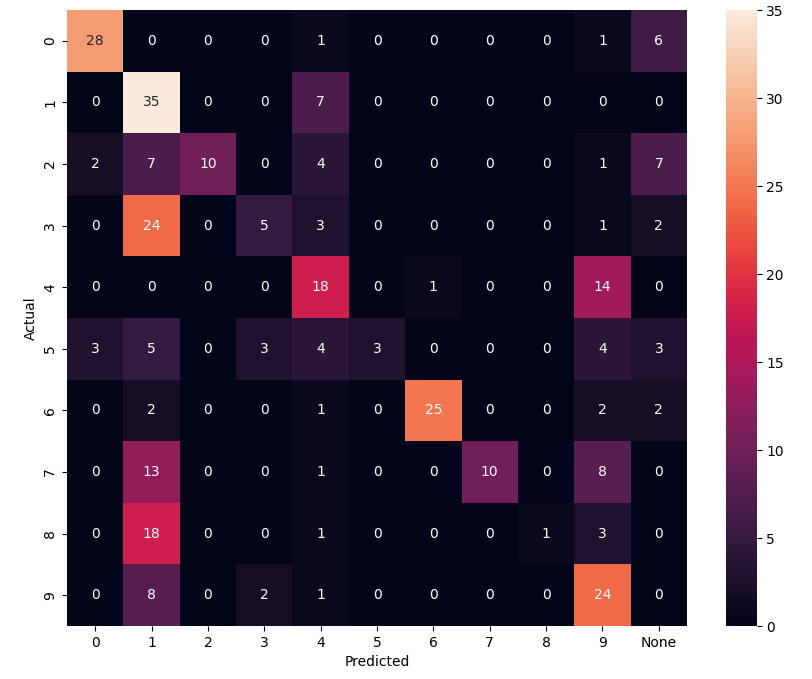
1. nn = 40 n\_train = 500 n\_test = 100, k = 0.99945,

g\_const = 3.0(train), 3.1(clf+test)

Training time: 3186.6706631183624

Actual number of training set: 324

{0: '3', 1: '0', 2: 'None', 3: '7', 4: '5', 5: '0', 6: 'None', 7: '2', 8: '5', 9: '6', 10: '2', 11: '9', 12: '3', 13: '4', 14: '4', 15: '9', 16: '8', 17: '0', 18: '2', 19: '0', 20: 'None', 21: '2', 22: '0', 23: '6', 24: '0', 25: '4', 26: '6', 27: '2', 28: '2', 29: '1', 30: '4', 31: '1', 32: '3', 33: 'None', 34: '6', 35: '5', 36: '4', 37: '0', 38: '0', 39: '0'}



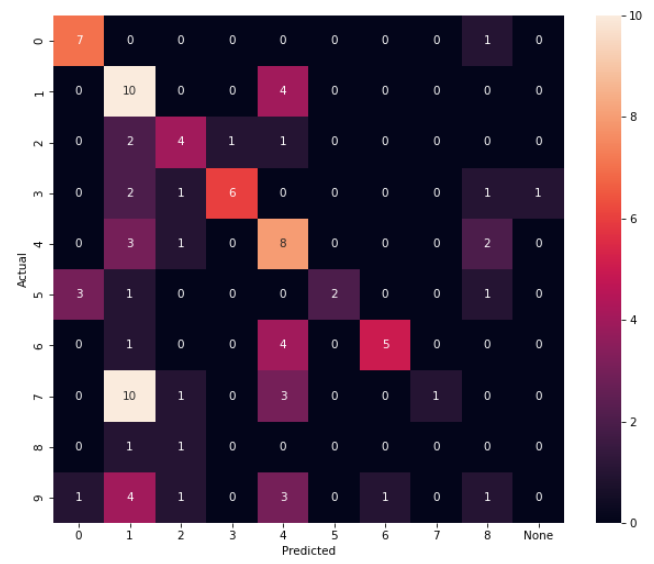
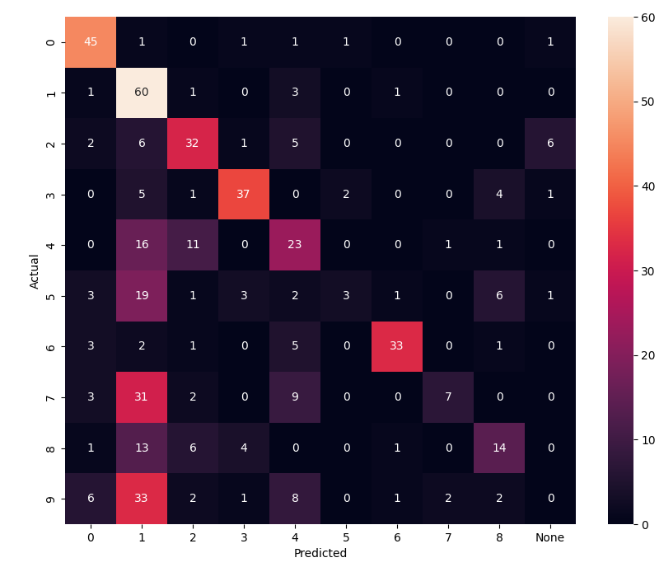
F1\_score (train): 0.49 F1\_score (test): 0.28

1. nn = 50 n\_train = 500 n\_test = 100, k = 0.99945,

g\_const = 3.0(train), 3.1(clf+test)

Training time: 6504.648355484009

Actual number of training set: 500



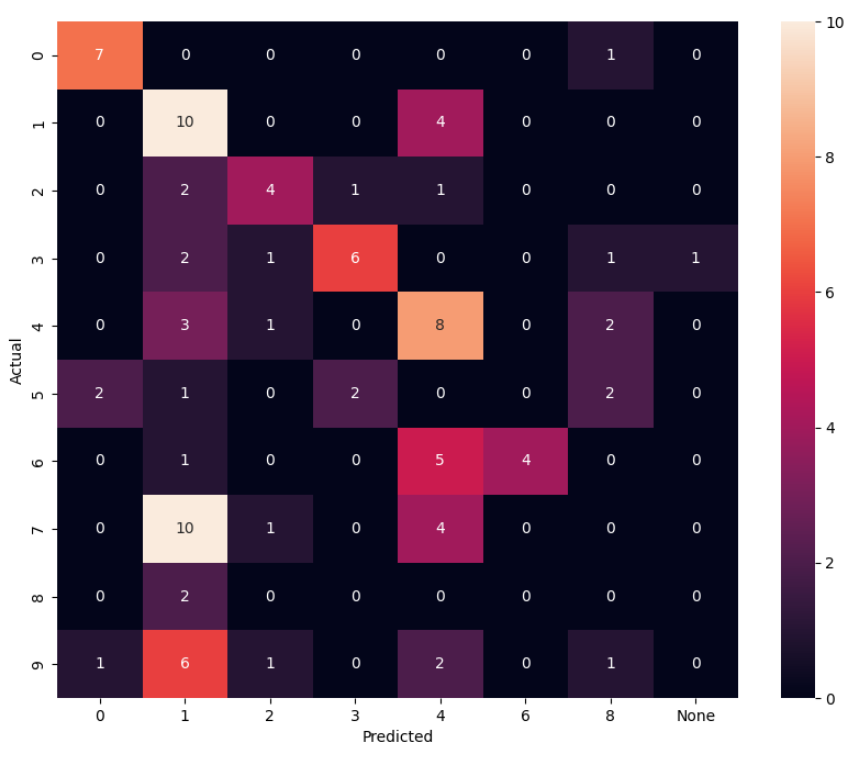
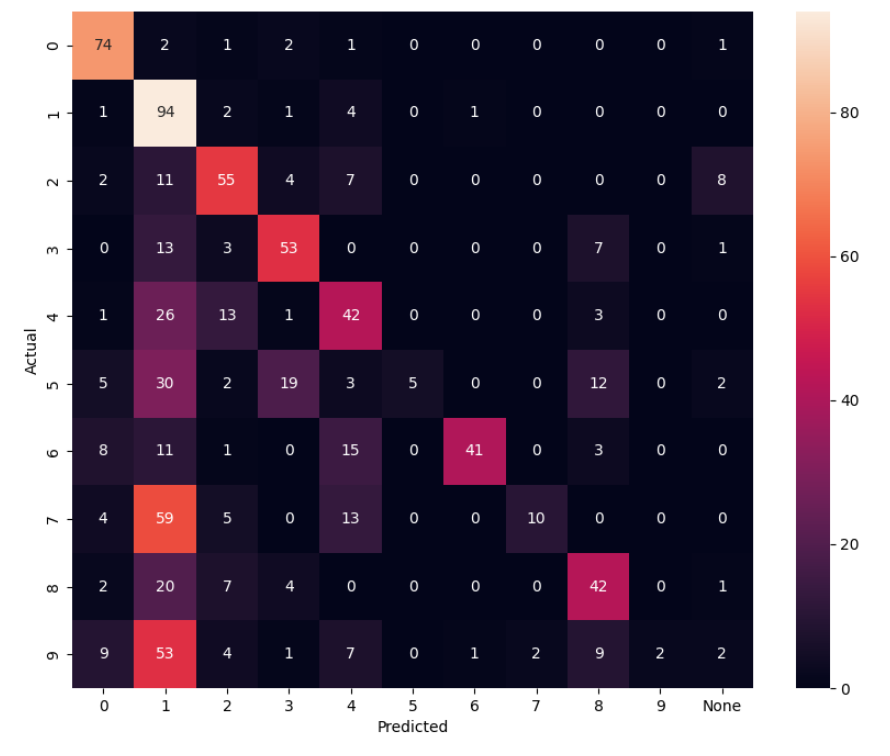
F1\_score (train): 0.508 F1\_score (test): 0.43

1. nn = 50 n\_train = 1000 n\_test = 100, k = 0.99945,

g\_const = 3.0(train), 3.1(clf+test)

Training time: 12791.355892896652

Actual number of training set: 848



F1\_score (train): 0.4929 F1\_score (test): 0.39

1. nn = 30 n\_train = 350 n\_test = 100, k = 0.99945,

g\_const = 3.0(train), 3.1(clf+test)

Training time: 1394.543408870697

Actual number of training set: 179

F1\_score (train): 0.536 F1\_score (test): 0.419

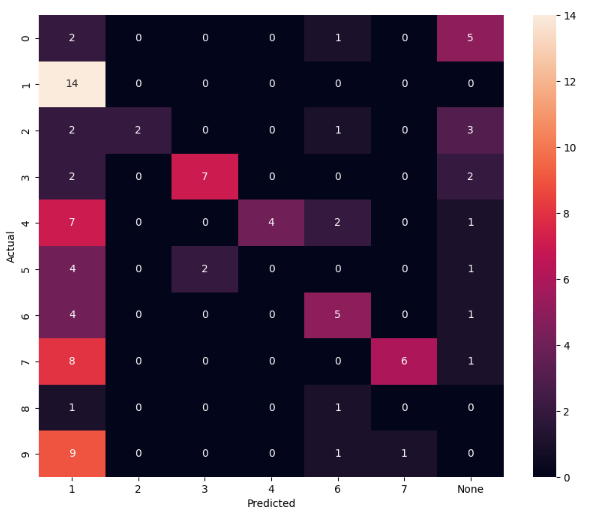
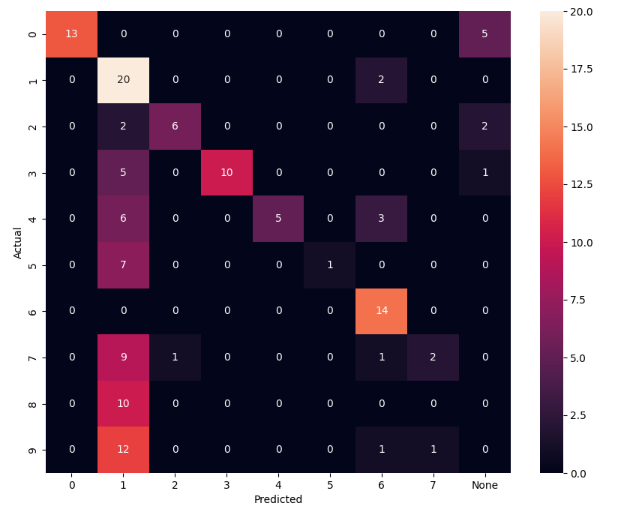
15.04.2023

1. nn = 30 n\_train = 500 n\_test = 100, k = 0.99945,

g\_const = 3.0(train), 3.1(clf+test)

Training time: 1301.0149984359741

Actual number of training set: 139



F1\_score (train): 0.51 F1\_score (test): 0.38

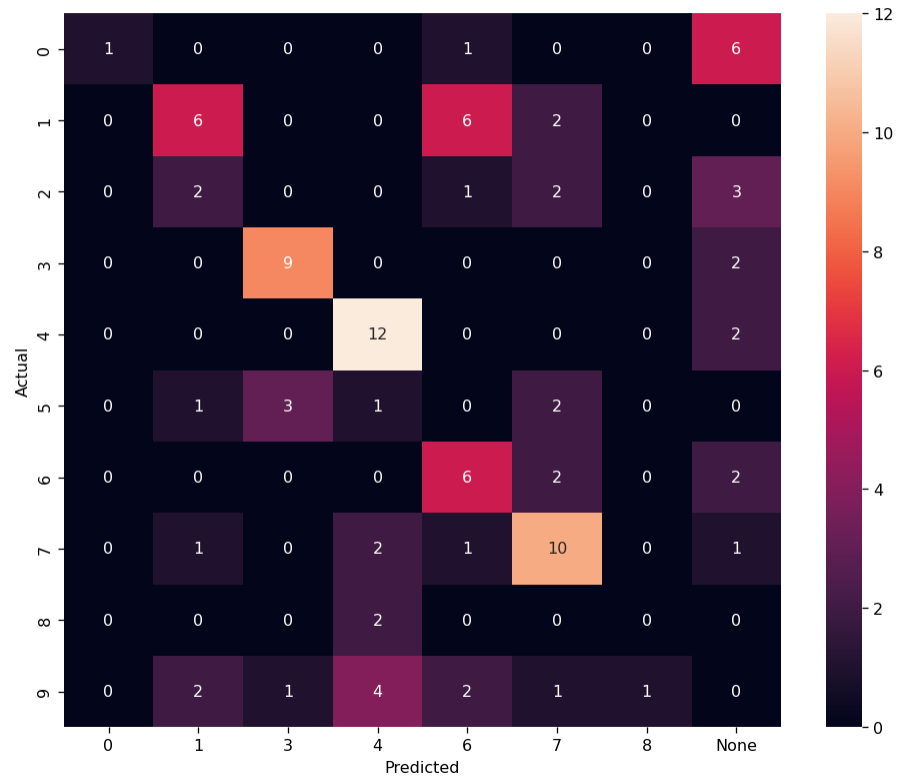
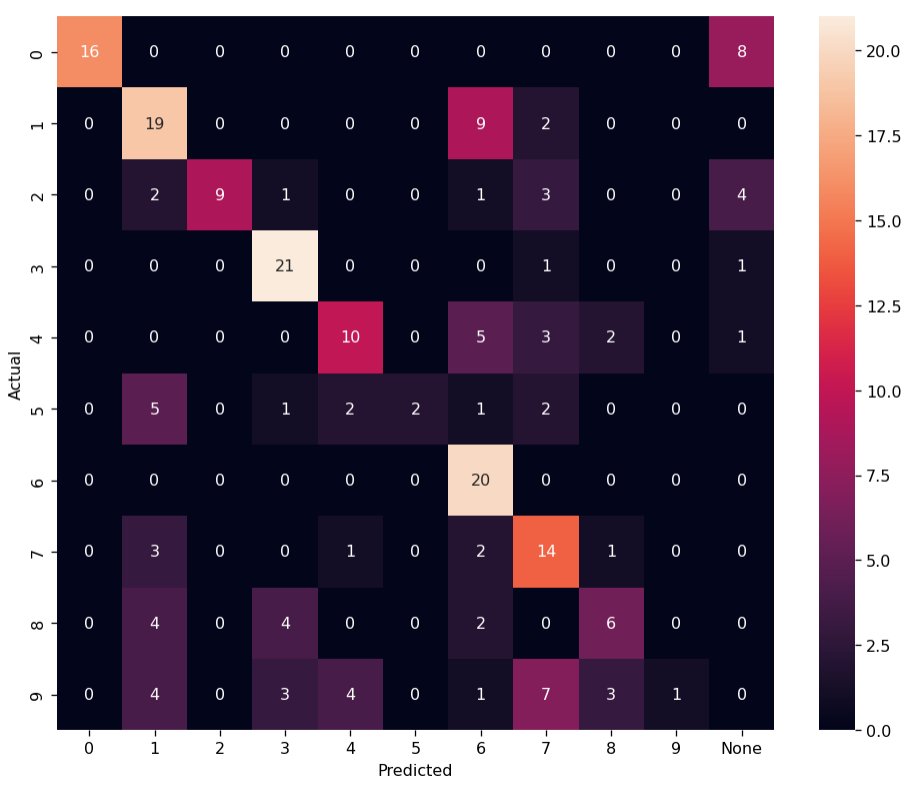
21.04.2023 (cooler comp)

1. nn = 40 n\_train = 500 n\_test = 100, k = 0.99945,

g\_const = 3.0(train), 3.1(clf+test)

Training time: 1654.8499121665955

Actual number of training set: 211



F1\_score (train): 0.559 F1\_score (test): 0.44

23.04.2023 (google collab)

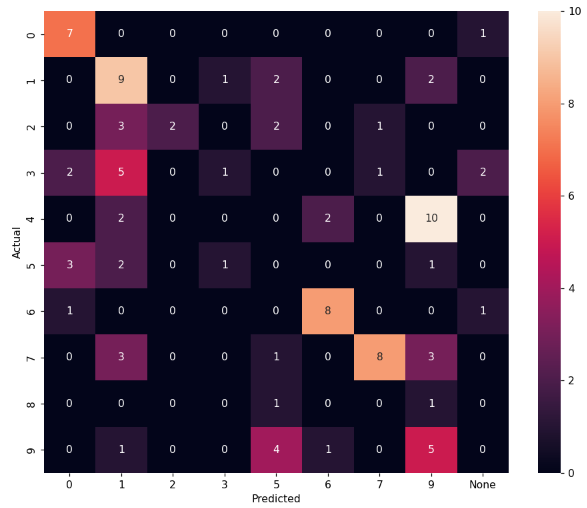
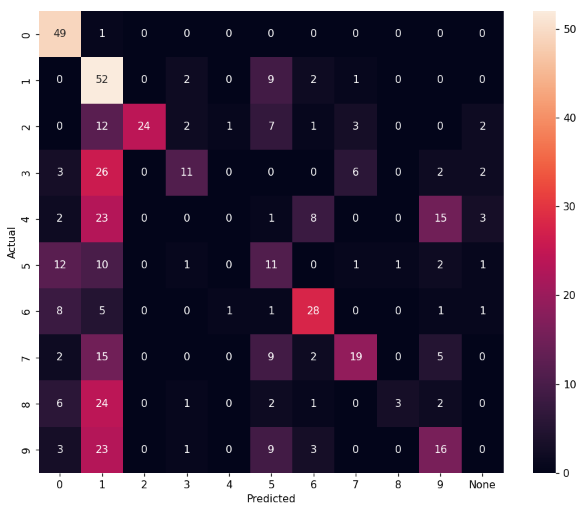
1. nn = 60 n\_train = 500 n\_test = 100, k = 0.99945,

g\_const = 3.0(train), 3.1(clf+test)

Training time: 7292.7466604709625

Actual number of training set: 500

Number of neurons, reacting to the corresponding image: {'None': 38, '0': 7, '2': 5, '5': 1, '7': 1, '8': 1, '1': 2, '6': 1, '3': 2, '9': 1, '4': 1}



F1\_score (train): 0.426 F1\_score (test): 0.40

23.04.2023 (my comp)

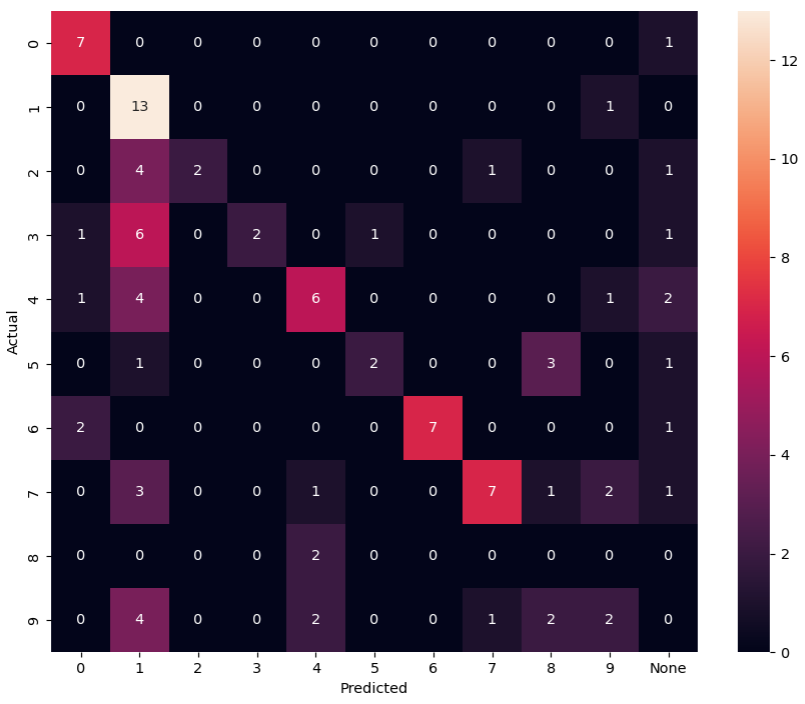
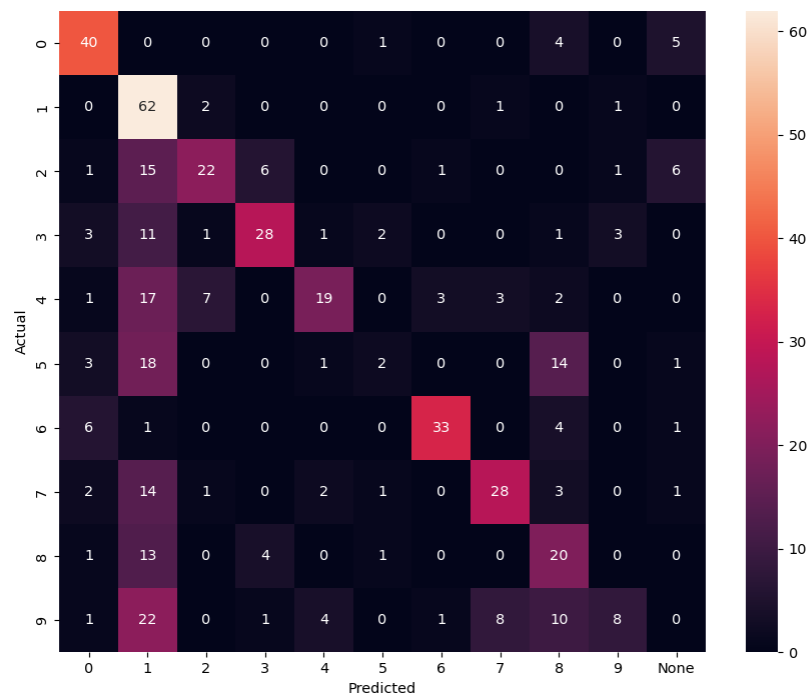
1. nn = 80 n\_train = 500 n\_test = 100, k = 0.99945,

g\_const = 3.0(train), 3.1(clf+test)

Training time: 9751.672762393951

Actual number of training set: 500

Number of neurons, reacting to the corresponding image: {'None': 52, '9': 1, '2': 2, '8': 1, '4': 3, '0': 5, '3': 5, '1': 4, '6': 2, '7': 3, '5': 2}



F1\_score (train): 0.524 F1\_score (test): 0.48

24.04.2023 (ipu comp)

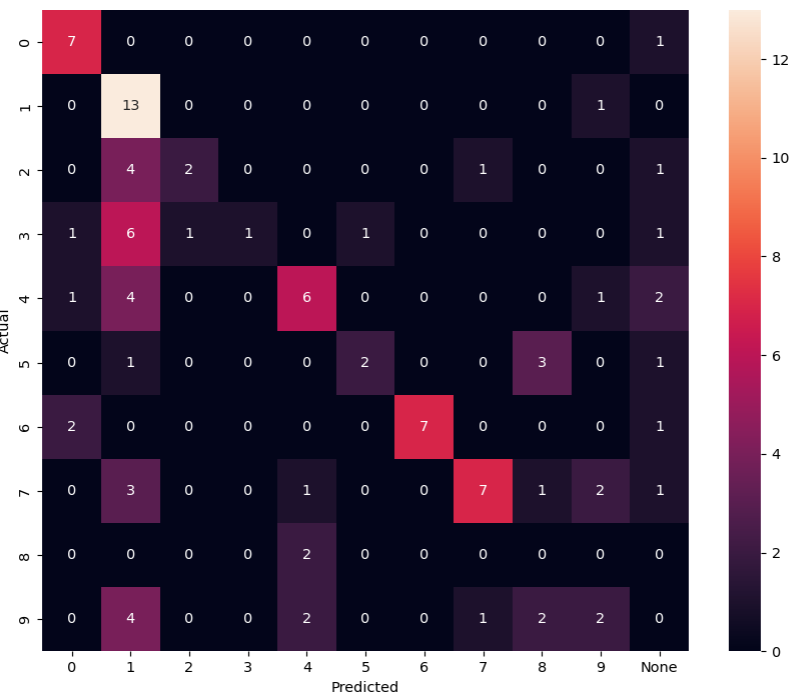
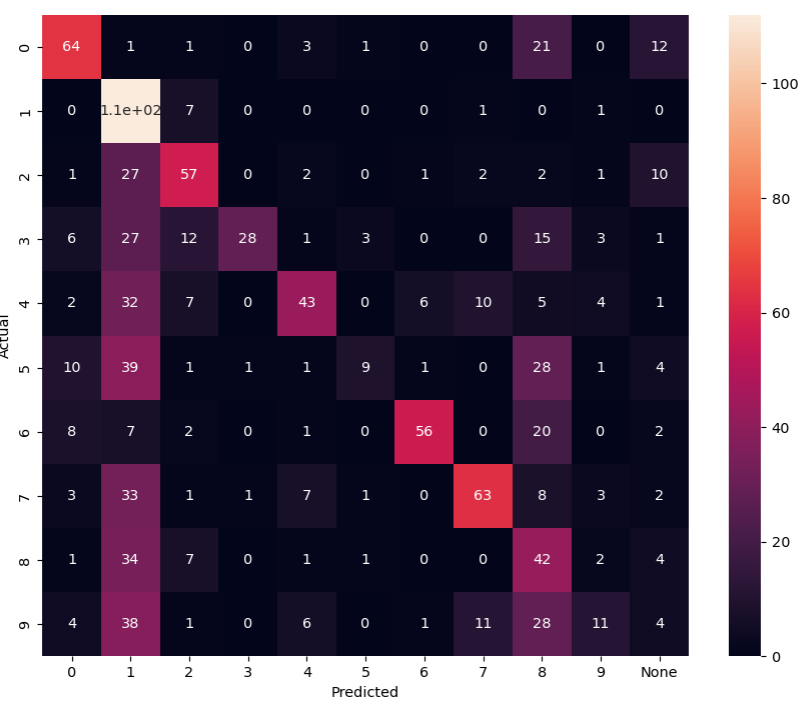
1. nn = 80 n\_train = 2000 n\_test = 100, k = 0.99945,

g\_const = 3.0(train), 3.1(clf+test)

Training time: 22015.084827661514

Actual number of training set: 1042

Number of neurons, reacting to the corresponding image: {'None': 52, '9': 1, '2': 3, '8': 1, '4': 3, '0': 5, '3': 4, '1': 4, '6': 2, '7': 3, '5': 2}



F1\_score (train): 0.465 F1\_score (test): 0.47

23.04.2023 (cooler comp)

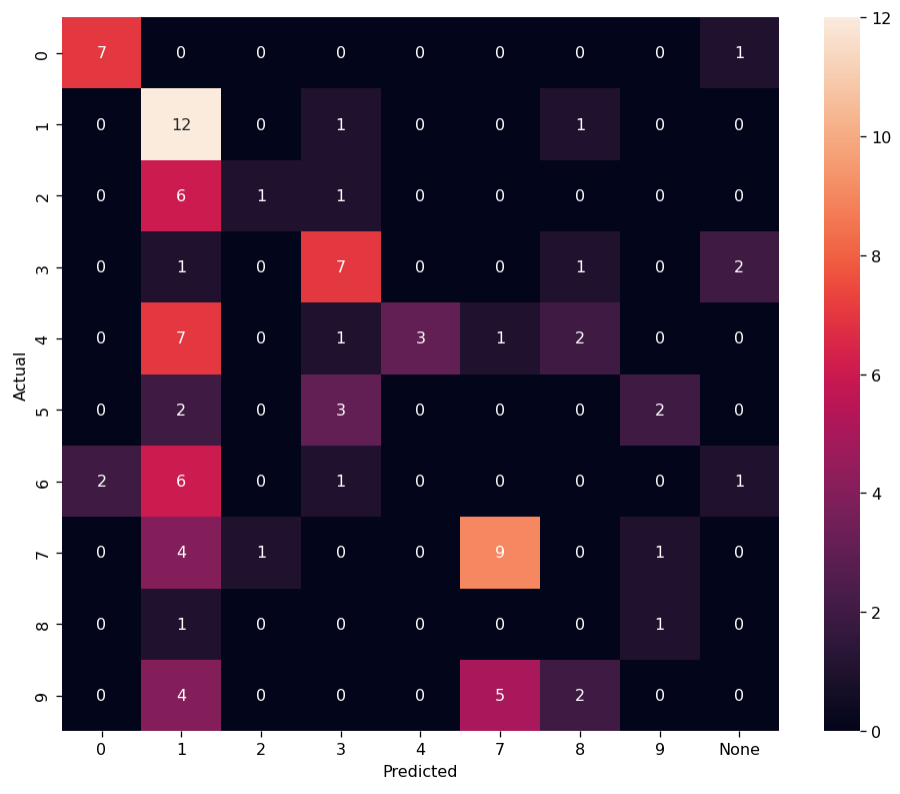
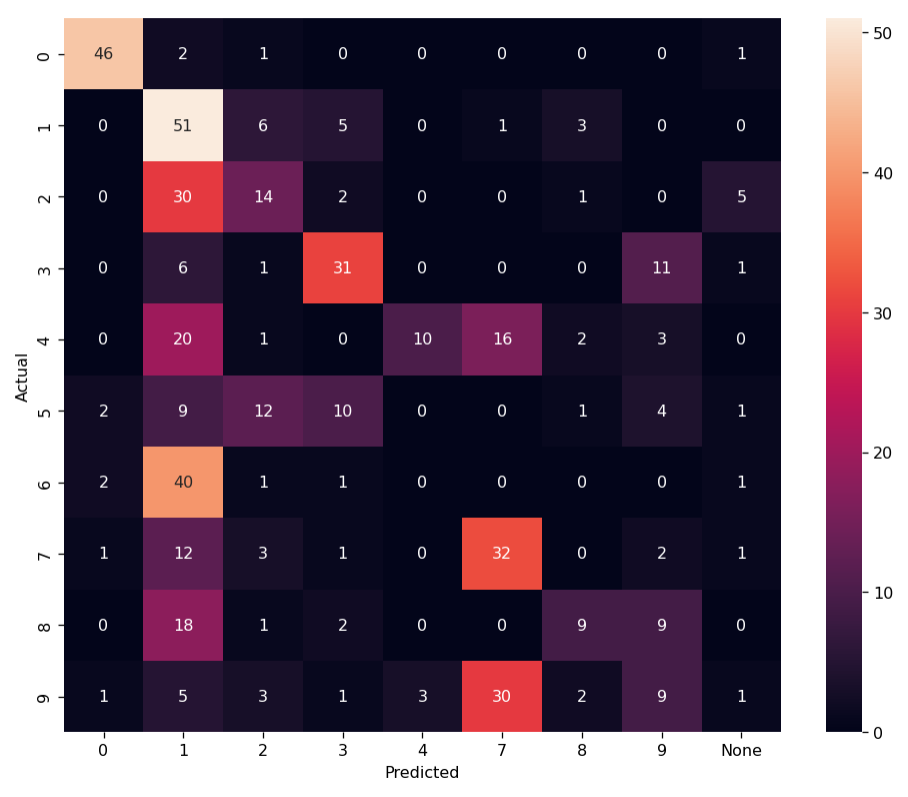
1. nn = 100 n\_train = 500 n\_test = 100, k = 0.99945,

g\_const = 3.0(train), 3.1(clf+test)

Training time: 9332.753213644028

Actual number of training set: 500

Number of neurons, reacting to the corresponding image: {'None': 85, '4': 1, '0': 5, '8': 2, '3': 2, '2': 1, '1': 2, '7': 1, '9': 1}



F1\_score (train): 0.404 F1\_score (test): 0.39

Задачи:

1. Подключить пакет для подбора гиперпараметров
2. Добавить (кросс)валидацию и\или увеличить обучающую выборку, чтобы удовлетворять условиям VC-размерности
3. Исследовать сеть на разреженность
4. Попробовать на датасете USPS
5. Сравнить с сетью на STDP
6. Сравнить со стандартным алгоритмом кластеризации (k-means)
7. Доработать актуальность: предоставить больше нейробиологических данных.
8. Доработать наглядность: подумать о представлении правила в матричном виде; подумать над представлением процесса обучения в виде графиков изменения некоторых ключевых характеристик.
9. Выложить код на GitHub с файлом requirements.