report from Group 2

- **Kato** and **Sato** report **results and findings** from **Challenge A : Particle Classification**
- **Ninomiya** introduces his motivation to study ML

Y. Kato

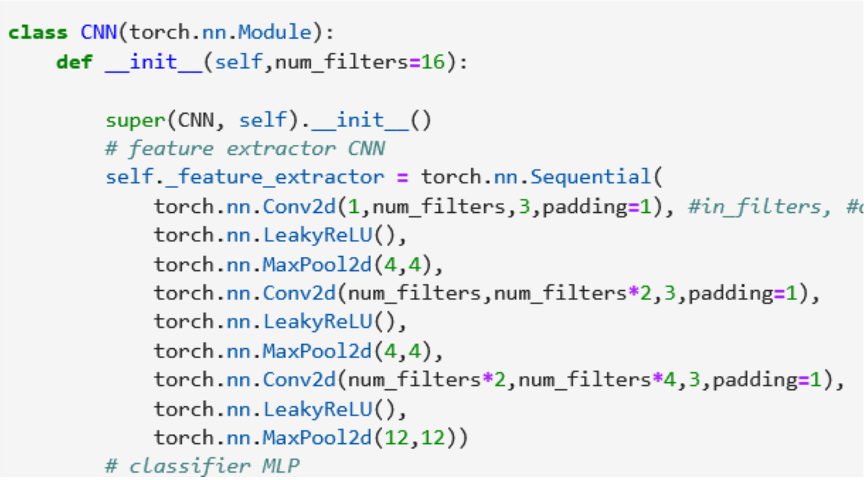
Challenge A: Particle classifier

- Starting from MNIST code, and adopted small difference of data structure.

- Checked how the performance changes according to change of hyper parameters

- Baseline: Num_filters=16 (and doubled for each MaxPooling), three convolution layers
  A: Num_filters = 32
  B: Put two Conv2d+LeakyReLU before MaxPool2d
  C: MaxPool (4,4) -> (2,2) and 7 convolution layers

Baseline



```
class CNN(torch.nn.Module):
    def __init__(self,num_filters=16):

        super(CNN, self).__init__()
        # feature extractor CNN
        self._feature_extractor = torch.nn.Sequential(
            torch.nn.Conv2d(1,num_filters,3,padding=1), #in_filters, #
            torch.nn.LeakyReLU(),
            torch.nn.MaxPool2d(4,4),
            torch.nn.Conv2d(num_filters,num_filters*2,3,padding=1),
            torch.nn.LeakyReLU(),
            torch.nn.MaxPool2d(4,4),
            torch.nn.Conv2d(num_filters*2,num_filters*4,3,padding=1),
            torch.nn.LeakyReLU(),
            torch.nn.MaxPool2d(12,12))
        # classifier MLP
```

Model	Accuracy (%)
Baseline	86.9
A	89.9
B	91.6
C	90.5

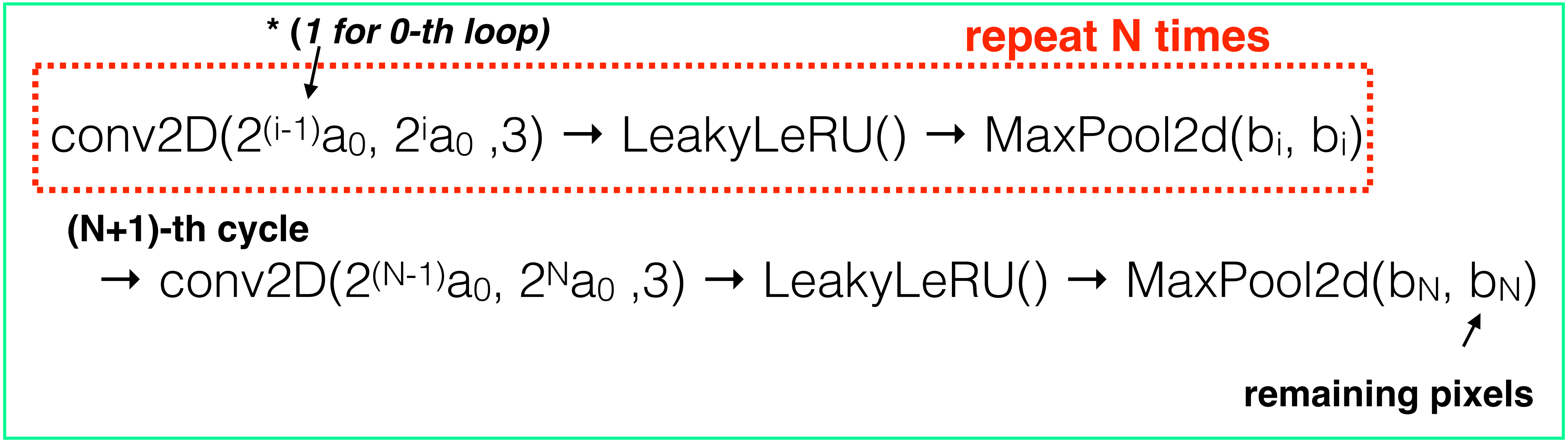
- All of changes improved the performance.
  - Putting two convolution layers has biggest improvement
- Tried combining A+B+C, but not finished within time..

Challenge A

K. Sato

Based on **IntroNeuralNetwork/Introduction03-MNIST-CNN**
I designed _***feature_extractor*** like …

_feature_extractor



***(1 for 0-th loop)**

repeat N times

conv2D($2^{(i-1)}a_0$, $2^i a_0$,3) → LeakyLeRU() → MaxPool2d(b_i, b_i)

(N+1)-th cycle

→ conv2D($2^{(N-1)}a_0$, $2^N a_0$,3) → LeakyLeRU() → MaxPool2d(b_N, b_N)

remaining pixels

What I changed

- # repetitions **N**
- combinations of $\mathbf{b_i}$
- initial # of filters : $\mathbf{a_0}$

What I checked

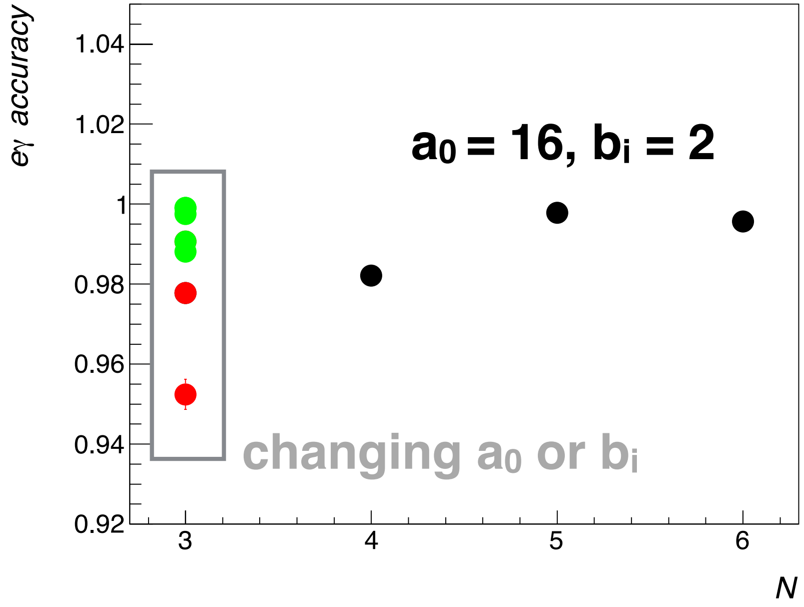
- accuracy of (e,γ)

  $(N_{ee}+N_{e\gamma}+N_{\gamma e}+N_{\gamma\gamma})\ /\ (N_e^{true}+N_\gamma^{true})$

***detector responses for e and γ are quite similar**

result

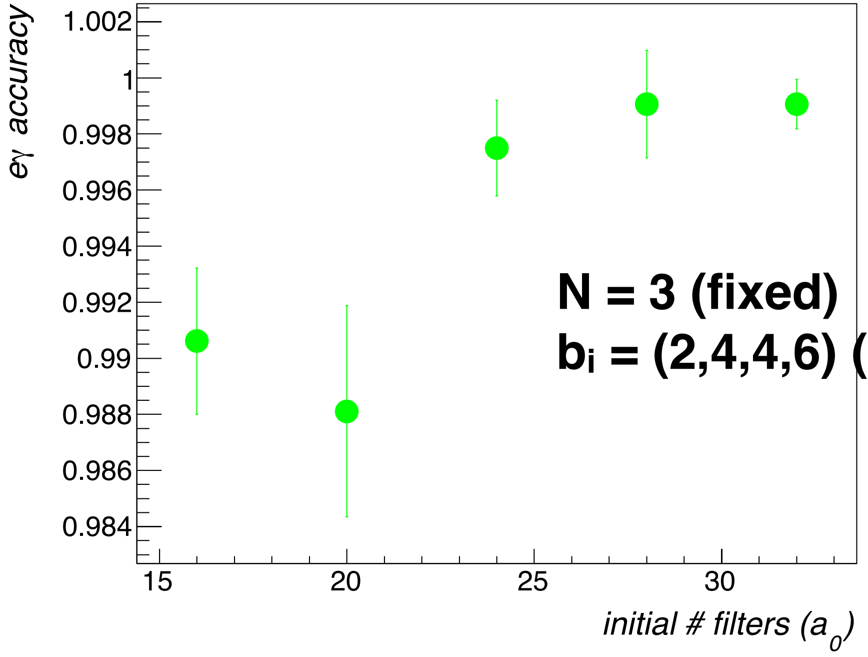
changing # of loop: N



$a_0 = 16$, $b_i = 2$

changing a_0 or b_i

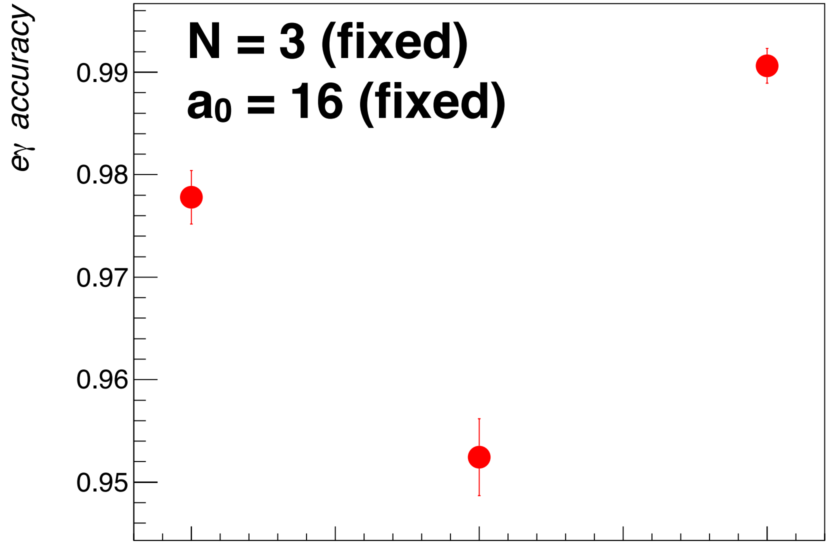
changing initial number of filters : a_0



N = 3 (fixed)

b_i = (2,4,4,6) (fixed)

changing b_i in MaxPool2d(b_i,b_i)



N = 3 (fixed)

a_0 = 16 (fixed)

b_0	2	2	2
b_1	2	2	4
b_2	2	4	4
b_3	24	12	6

- Larger a_0 (i.e., larger discriminants), better accuracy.

- MaxPool2d(b,b) with larger b reduces accuracy.
  - maybe some information are lost

K. Ninomiya

My B4 experiment

- **Main theme**
  Increase the accuracy to reconstructing events from image of water Cherenkov radiation at Super-Kamiokande.

- **Methods**
  First step Compare $\nu_\mu/\overline{\nu_\mu}$ identification accuracy of using Maximum likelihood estimation to Neural Network.
  Second step Challenging the main theme!

- **Leaning in KMI school**
  I learned basis of Neural Network. In particular, I will refer to code for images in MNIST-solution.