ensemble\_size = 4

    alpha\_gamma\_init = 0.5

    num\_epochs = 5

    batch\_size = 128

    lr = 0.01

**Originalcode**

Ein Bild, das Text, Screenshot, Diagramm enthält.

Automatisch generierte Beschreibung

**With IVON:**

optimizer = ivon.IVON(model.parameters(), lr=0.1, ess=len(x\_train), weight\_decay=1e-4, beta1=0.9)

Ein Bild, das Text, Screenshot, Diagramm, Schrift enthält.

Automatisch generierte Beschreibung

**MLP\_CNN\_BatchEns\_MNIST**

Adam

optimizer = torch.optim.Adam(model.parameters(), lr=lr, betas=(0.9, 0.999), weight\_decay=1e-4)

MLP

Epoch [1/5], Simple Model Test Accuracy: 94.87%

Epoch [2/5], Simple Model Test Accuracy: 96.52%

Epoch [3/5], Simple Model Test Accuracy: 96.20%

Epoch [4/5], Simple Model Test Accuracy: 97.04%

Epoch [5/5], Simple Model Test Accuracy: 96.62%

Epoch [1/5], Batchensemble Model Test Accuracy: 96.19%

Epoch [2/5], Batchensemble Model Test Accuracy: 97.09%

Epoch [3/5], Batchensemble Model Test Accuracy: 96.64%

Epoch [4/5], Batchensemble Model Test Accuracy: 97.00%

Epoch [5/5], Batchensemble Model Test Accuracy: 96.99%

Ein Bild, das Text, Diagramm, Reihe, parallel enthält.

Automatisch generierte Beschreibung

CNN

Epoch [1/5], Simple Model Test Accuracy: 98.34%

Epoch [2/5], Simple Model Test Accuracy: 98.50%

Epoch [3/5], Simple Model Test Accuracy: 97.89%

Epoch [4/5], Simple Model Test Accuracy: 98.65%

Epoch [5/5], Simple Model Test Accuracy: 98.98%

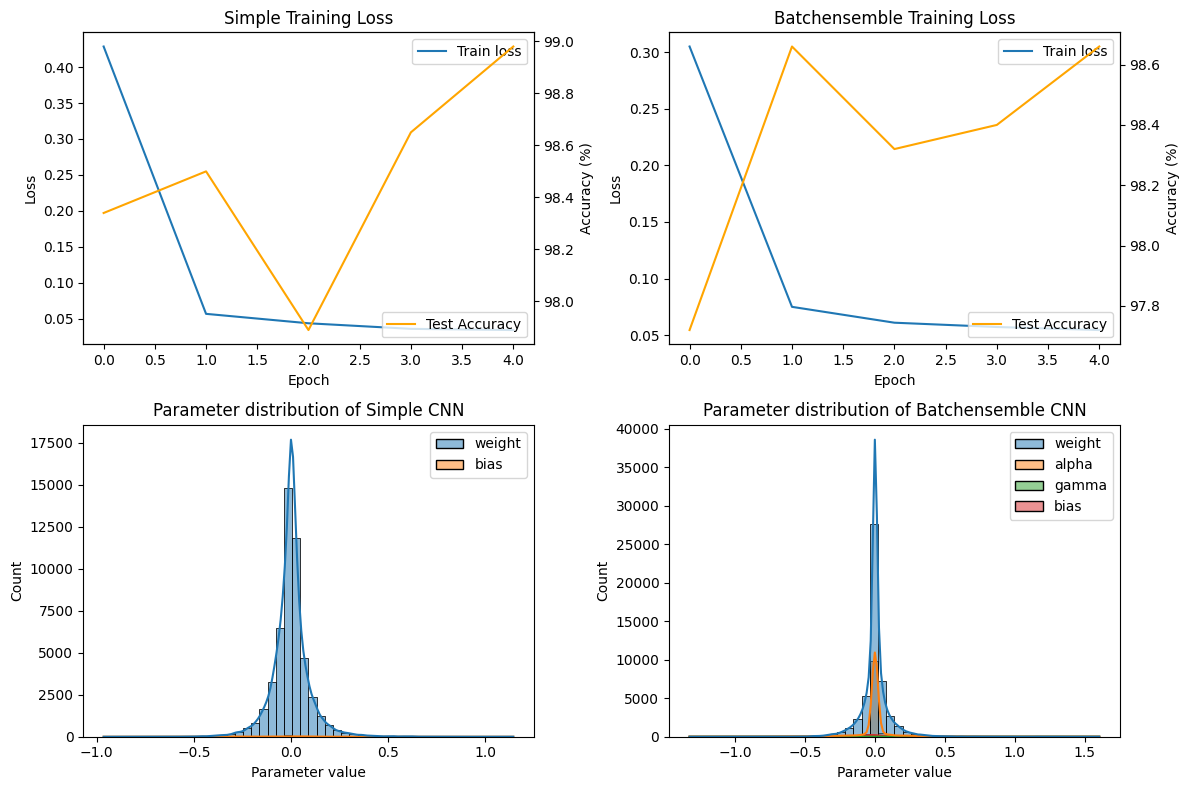
Epoch [1/5], Batchensemble Model Test Accuracy: 97.72%

Epoch [2/5], Batchensemble Model Test Accuracy: 98.66%

Epoch [3/5], Batchensemble Model Test Accuracy: 98.32%

Epoch [4/5], Batchensemble Model Test Accuracy: 98.40%

Epoch [5/5], Batchensemble Model Test Accuracy: 98.66%



IVON

optimizer = ivon.IVON(model.parameters(), lr=lr, ess=len(train\_loader), weight\_decay=1e-4, beta1=0.9)

MLP -> Fast

Epoch [1/5], Simple Model Test Accuracy: 63.47%

Epoch [2/5], Simple Model Test Accuracy: 66.44%

Epoch [3/5], Simple Model Test Accuracy: 75.19%

Epoch [4/5], Simple Model Test Accuracy: 84.25%

Epoch [5/5], Simple Model Test Accuracy: 86.02%

Epoch [1/5], Batchensemble Model Test Accuracy: 46.15%

Epoch [2/5], Batchensemble Model Test Accuracy: 57.62%

Epoch [3/5], Batchensemble Model Test Accuracy: 59.10%

Epoch [4/5], Batchensemble Model Test Accuracy: 59.53%

Epoch [5/5], Batchensemble Model Test Accuracy: 56.40%

Ein Bild, das Text, Diagramm, Reihe, parallel enthält.

Automatisch generierte Beschreibung

CNN -> Simple little bit longer, Ensemble very long

Epoch [1/5], Simple Model Test Accuracy: 93.45%

Epoch [2/5], Simple Model Test Accuracy: 95.68%

Epoch [3/5], Simple Model Test Accuracy: 96.29%

Epoch [4/5], Simple Model Test Accuracy: 96.86%

Epoch [5/5], Simple Model Test Accuracy: 97.09%

Epoch [1/5], Batchensemble Model Test Accuracy: 85.98%

Epoch [2/5], Batchensemble Model Test Accuracy: 90.45%

Epoch [3/5], Batchensemble Model Test Accuracy: 92.27%

Epoch [4/5], Batchensemble Model Test Accuracy: 93.48%

Epoch [5/5], Batchensemble Model Test Accuracy: 94.01%

Ein Bild, das Text, Diagramm, Reihe, parallel enthält.

Automatisch generierte Beschreibung

Overall 30 min

Ein Bild, das Screenshot, Text, Kreis, Elektronik enthält.

Automatisch generierte Beschreibung

[1] <https://ysngshn.github.io/research/why-ivon/>

IVON tries to fit a Gaussian distribution over the most probable parameter configurations instead of looking for a specific set of parameters. [1]

**Steps & plots:**

Different learning rate

Different epochs

ensemble\_size = 4

alpha\_gamma\_init = 0.5

num\_epochs = 5

batch\_size = 128

lr = 0.01

Different weight decay

Different beta

optimizer = ivon.IVON(model.parameters(), lr=lr, ess=len(train\_loader), weight\_decay=1e-4, beta1=0.9)

Im Notebook der Autoren gefunden (für MNIST)  
learning\_rate = 1e-1

batch\_size = 128

epochs = 100

weight\_decay = 1e-4

momentum = 0.9