EARIN LAB 5

VARIANT 5

Paweł Uhma,
Carrasquero Maurizio

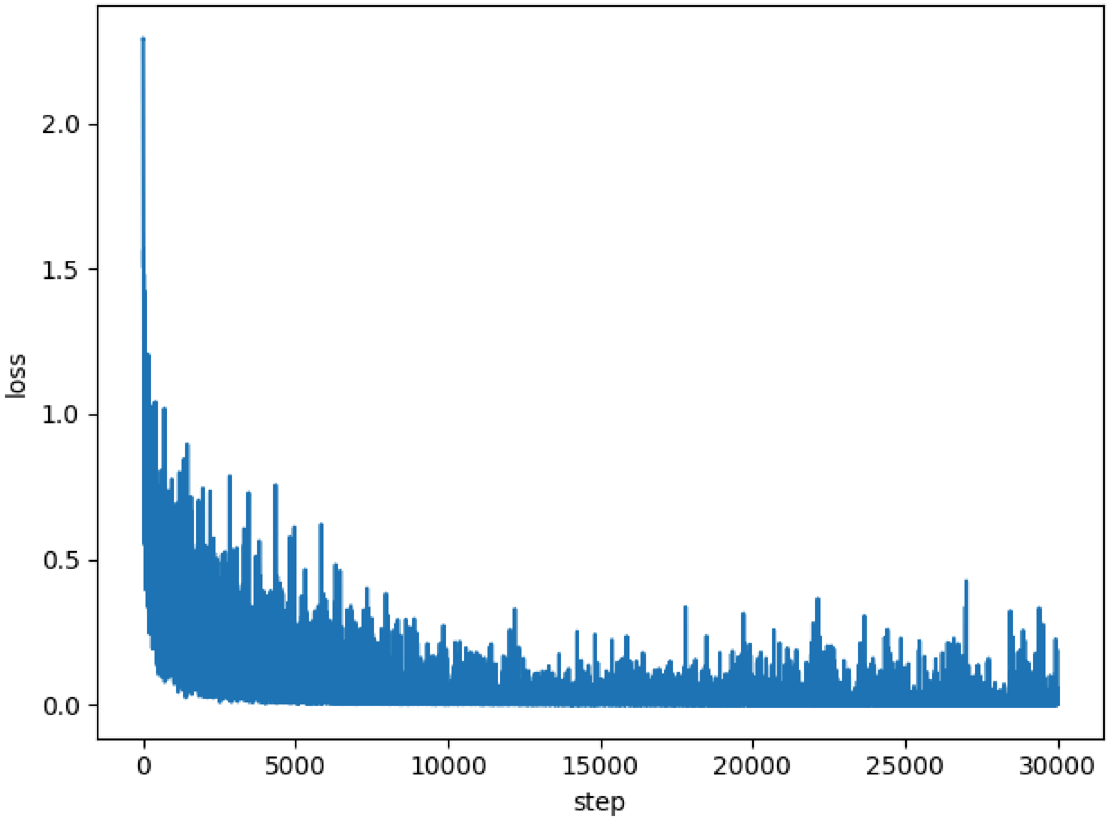
This short report sums up what we did during the lab where we trained a simple multi–layer perceptron (MLP) on the KMNIST dataset and tried a few hyper‑parameters.

1. Setup

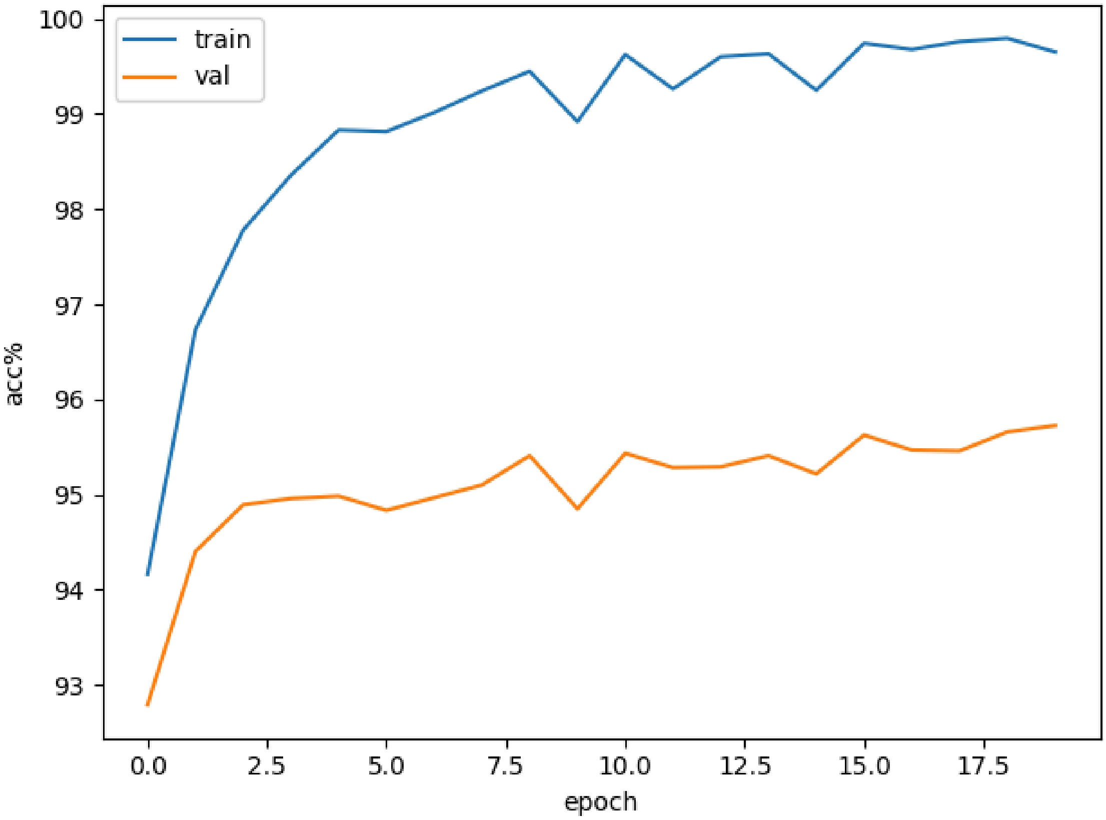
- Dataset: KMNIST (60 k train images of Japanese characters)
- Input: 28×28 grayscale, flattened to 784 features
- Baseline network: 1 hidden layer, 256 neurons, ReLU
- Optimiser: Adam
- Learning rate: 0.001
- Batch size: 32
- Epochs: 20 for full training
- Device: CPU/GPU depending on availability

2. Baseline training

The baseline run hit a best validation accuracy of **95.43 %** after 20 epochs. Below is the loss curve collected during that run:



We also tracked full-epoch train vs. validation accuracy:



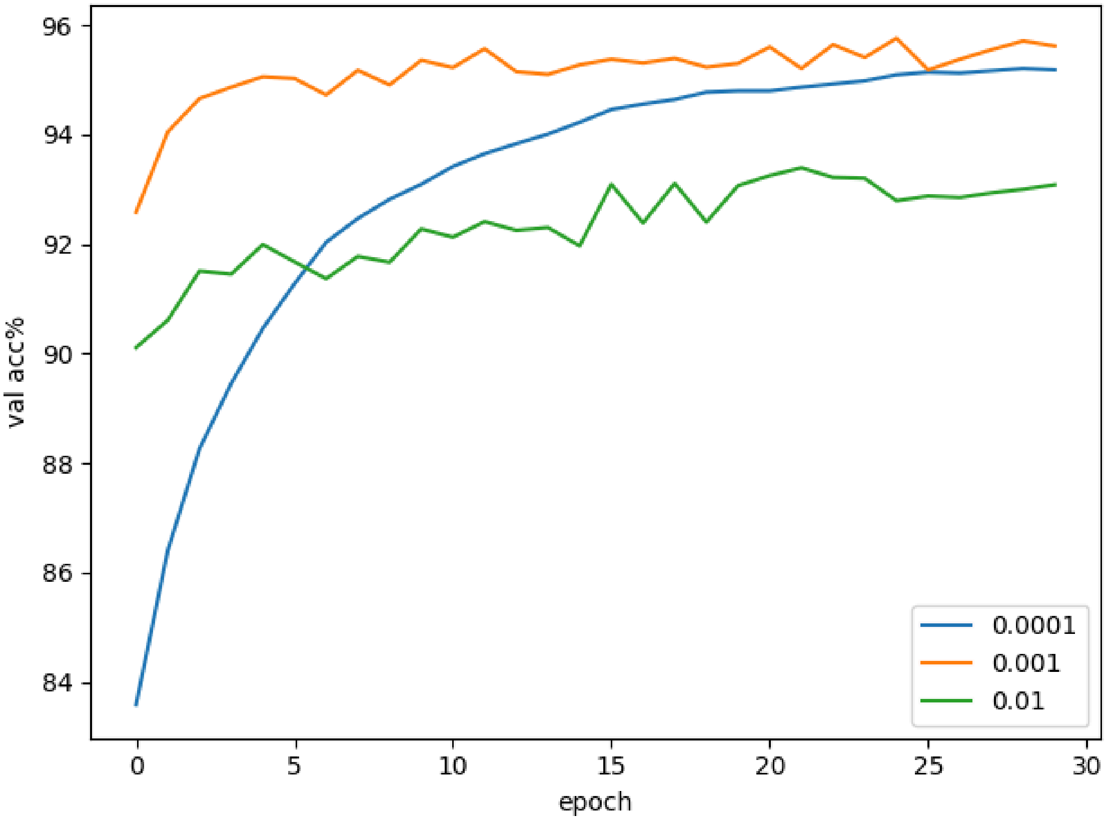
It can be observed that it stabilizes around 15 epoch.

3. Hyper-parameter experiments

3.1 Learning rate

Tried 0.0001, 0.001, and 0.01 (30 epochs each).

- 0.001 performed best at **95.76 %**
- 0.0001 was almost as good **95.21 %**.
- 0.01 was clearly too high **93.39 %**.



It can be concluded that 0.01 lr generalizes too much and overshoots not being able to find the optimum.

lr of 0.0001 starts with the worst accuracy but steadily over time is reaching really good accuracy, it probably would surpass 0.0001 after more epochs

lr of 0.001 starts with the highest accuracy and quickly reaches the highest level, it is the best compromise between speed and accuracy

3.2 Batch size

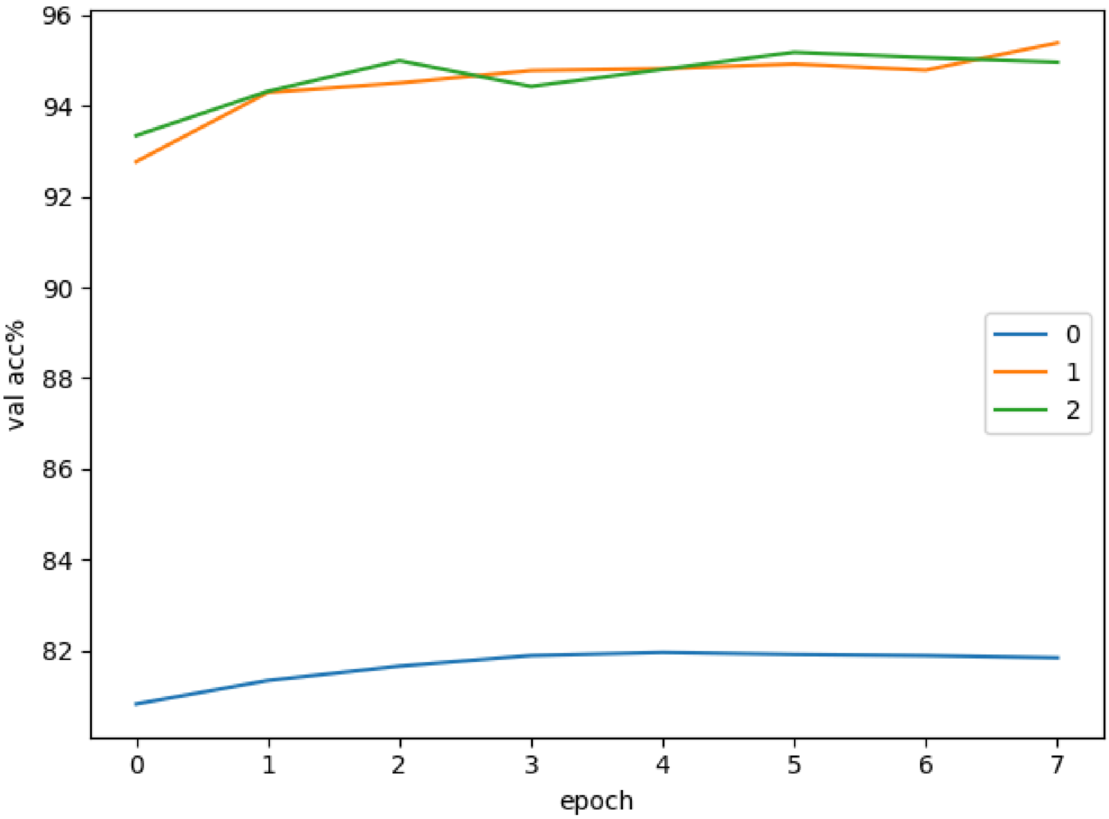
To avoid long runtimes we ran just one epoch for bs = 1/32/128.
Even that single pass shows the trend: smaller batches were more accurate, but also slow in training.

After 1 epoch the accuracy was:
bs 1 ≈ **92.9 %**,
bs 32 ≈ **92.7 %**,
bs 128 ≈ **89.6 %**

3.3 Hidden layers

Zero, one, or two hidden layers for 8 epochs.
• 0 layers (i.e. logistic reg) bottomed out at ~**82 %**

- 1 layer was best (**95.38 %**).
- 2 layers did not really improve (**95.17 %**).

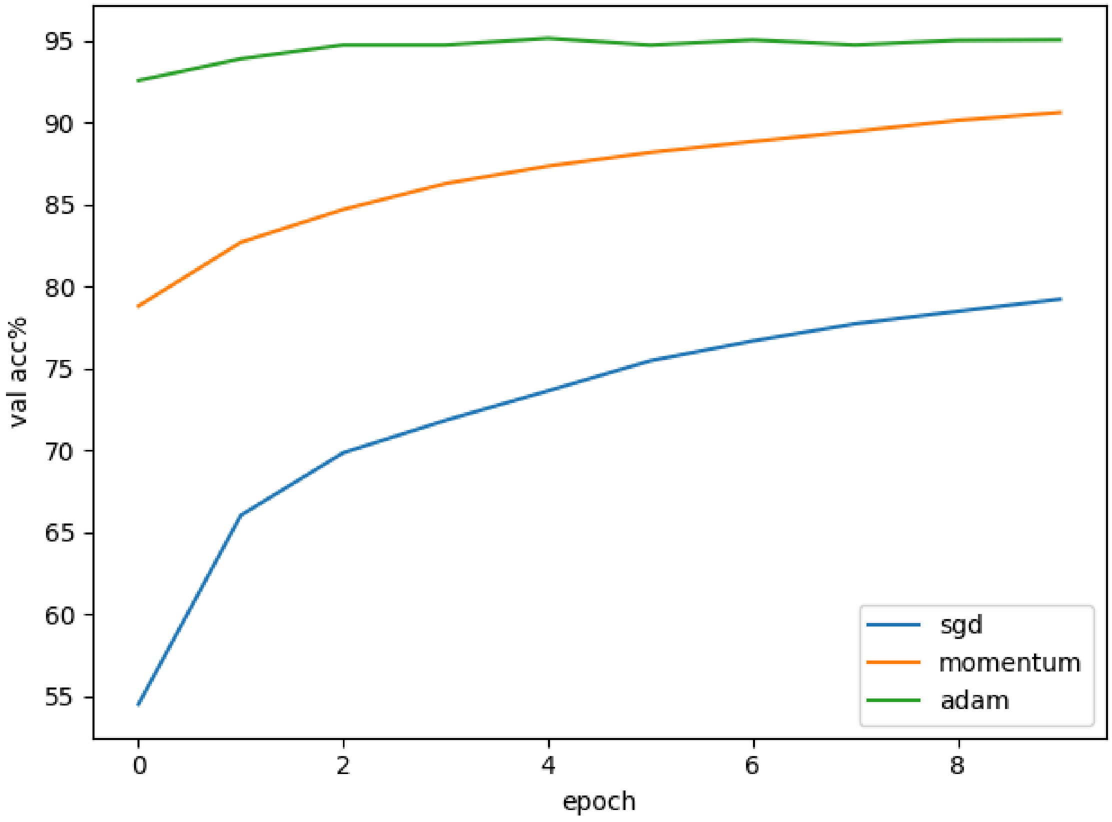


It can be concluded that this dataset does not require deep network for correct classification and 1 hidden layer was just enough

3.4 Optimiser

Ten-epoch runs with SGD, Momentum, and Adam.

- Adam still wins at **95.15 %**
- Momentum helped a lot over plain SGD (**90.6 %** vs **79.2 %**).



4. Take-aways

- Adam with lr 0.001 has the most promising outcomes.
- Going deeper (past one hidden layer) does add much on KMNIST.
- Extremely small batches can work but make epochs slow.

5. Code run

The code should be run by running the main.py file.
The testing values can be tweaked in compare() function