A3 Report - Kevin Armbruster

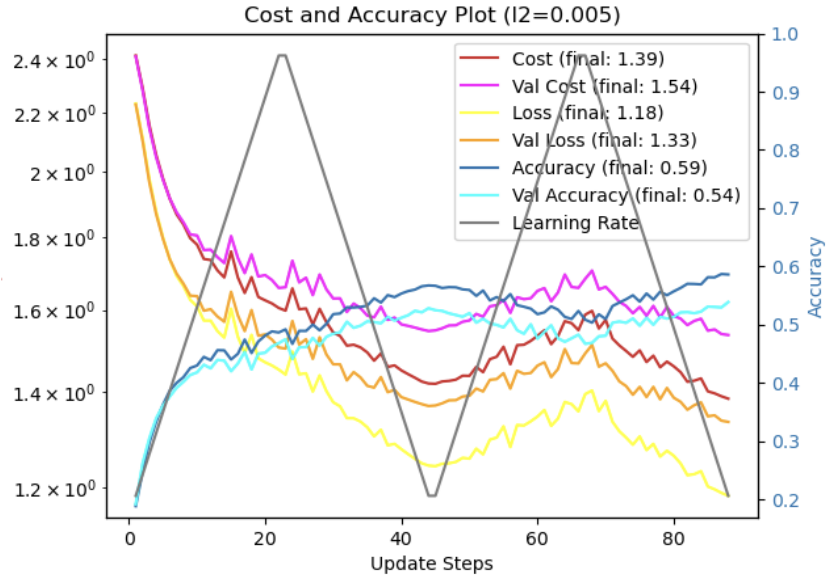
Gradient Computation Verification

I adapted the gradient computation for the two new, learned parameters gamma and beta. I verified the gradients as before by computing the analytical and numerical gradients (finite differences) and then using numpy's allclose function to verify the absolute and relative error. With sufficient small dimensions and sample size, the gradients for all k layers are equal up to 1e-6.

3 Layer Network

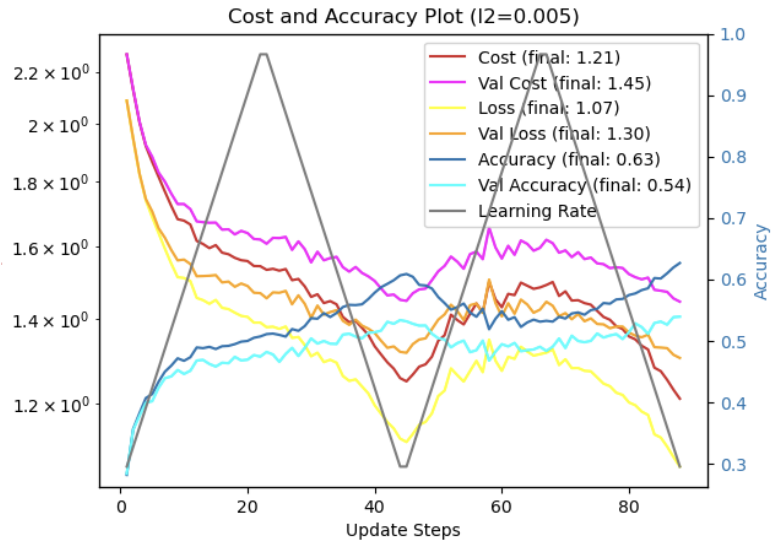
Without Batch Normalization

Test Accuracy = 0.5312



With Batch Normalization

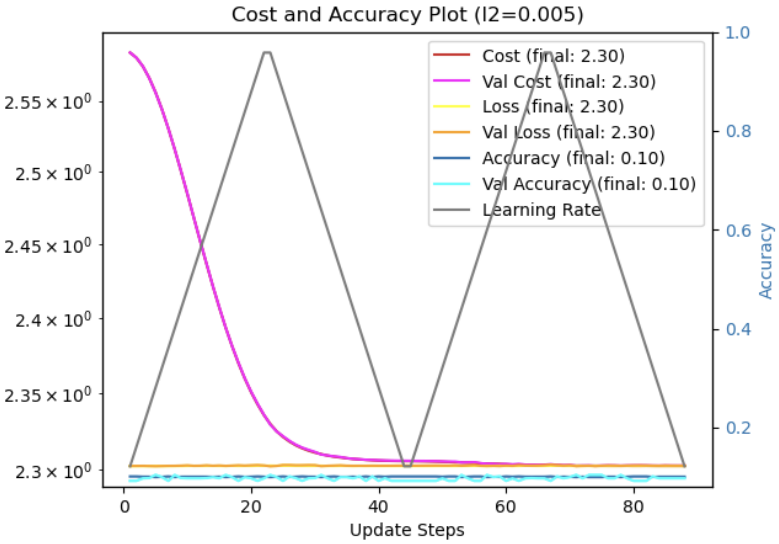
Test Accuracy = 0.5437



9 Layer Network

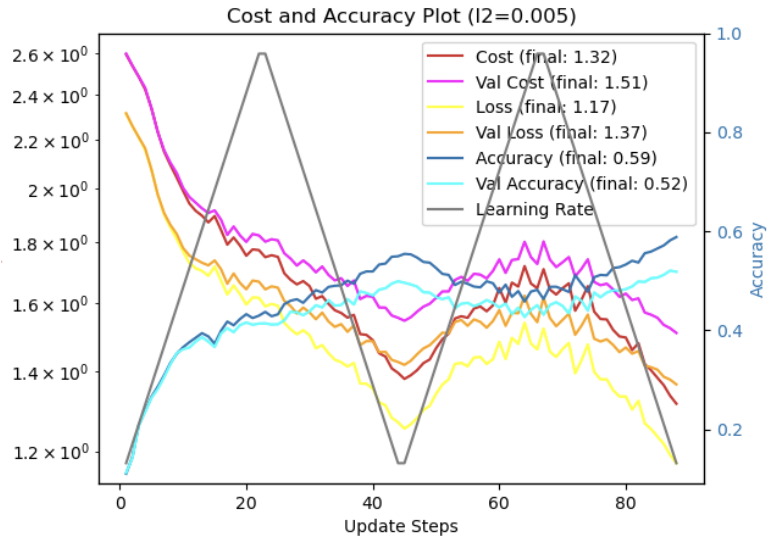
Without Batch Normalization

Test Accuracy = 0.1



With Batch Normalization

Test Accuracy = 0.526



Coarse to Fine Hyperparameter Search

I tested 50 steps in logarithmic space from 1e-2 to 1e-8 using np.logspace. As the differences are small and widely spread across space, I chose the best and skipped a finer granularity.

Results (L2: Validation Accuracy):

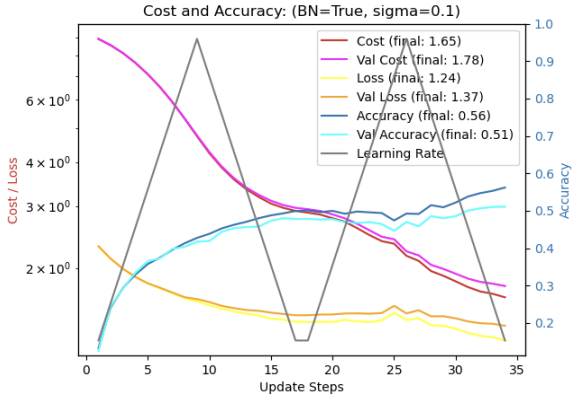
1. 0.00429193: 0.547
2. 0.01000000: 0.543
3. 0.00000002: 0.5424
4. 0.00323746: 0.541
5. 0.00000010: 0.5406
6. 0.00184207: 0.5404
7. 0.00079060: 0.5394
8. 0.00568987: 0.5394
9. 0.00244205: 0.5392
10. 0.00000004: 0.5388
11. … next is 50th …
12. 0.00000494: 0.5224

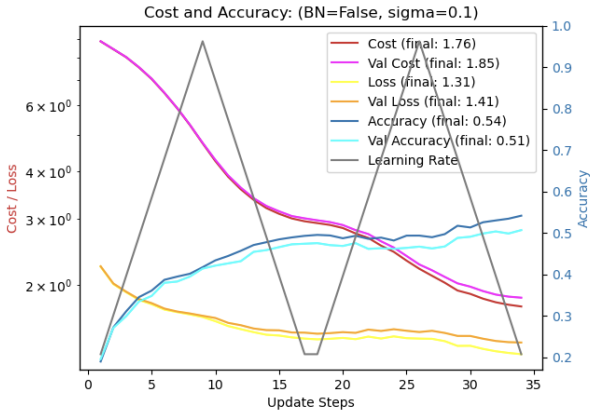
Test Accuracy of best L2 = 0.5386

Sensitivity to initialization

Sigma 1e-1

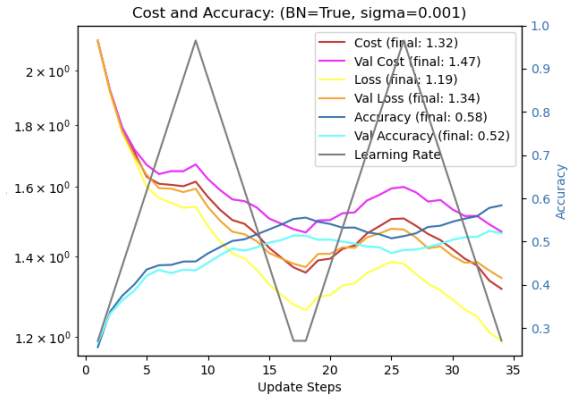
Sigma 0.1: BN: False: Final test accuracy 0.5069
Sigma 0.1: BN: True: Final test accuracy 0.5167

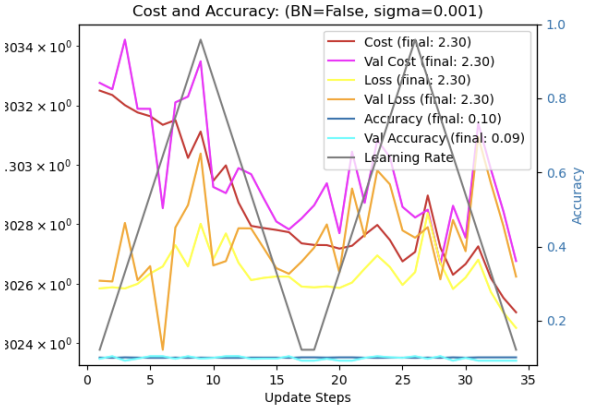




Sigma 1e-3

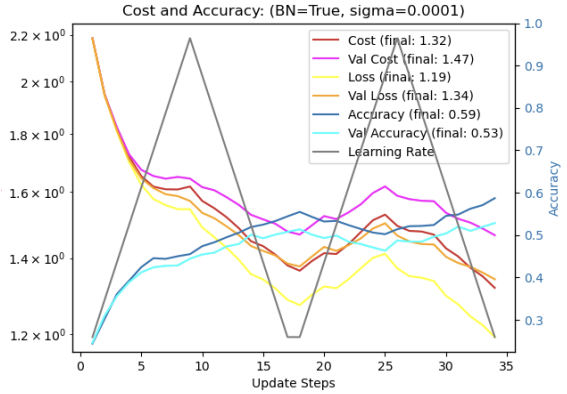
Sigma 0.001: BN: False: Final test accuracy 0.1
Sigma 0.001: BN: True: Final test accuracy 0.5295

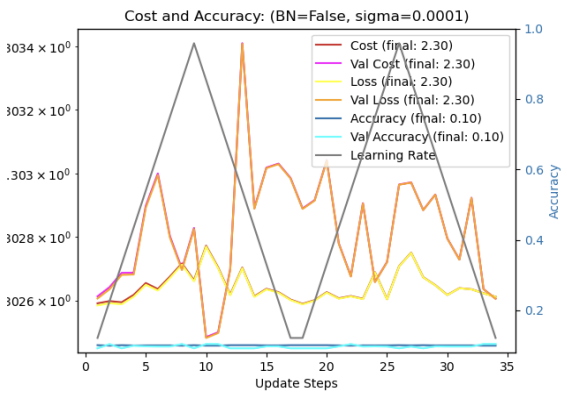




Sigma 1e-4

Sigma 0.0001: BN: False: Final test accuracy 0.1
Sigma 0.0001: BN: True: Final test accuracy 0.5289





Final comments

Batch normalization allows the usage of deeper networks and reduces the importance of the network initialization, thus improving the learning process. For both the 9 layer network and the initialization with sigma < 1e-1, we observed without BN a diverged learning process, resulting in fluctuating cost/loss curves and accuracy values around the random chance of 1/10 classes => 10%. BN solves these issues.