DD2424

Assignment 3

Isac Lorentz

June 10, 2022

1 Analytical Gradient Computation

I believe that my analytical gradients were correct. Running a python conversion of gradNumSlow, I managed to achieve to achieve sufficiently low errors for the norms of the gradients. The formula below was used:

$$\frac{|\, f'_a - f'_n \,|}{\max(10e-6, |\, f'_a \,| + |\, f'_n \,|)}$$

Using 20 samples and 10 dimensions and h=1e-6 for gradNumSlow, the following relative errors of the norms was obtained, which are all at an acceptable threshold. I tried a few different configurations, as suggested in the lab

Number of layers	Hidden layers	Gradient	Value	batchNorm
2	[50]	W1	1.49e-9	No
2	[50]	W2	8.03e-10	No
2	[50]	b1	1.77e-9	No
2	[50]	b2	7.46e-10	No
2	[50]	W1	2.50e-9	Yes
2	[50]	W2	1.07e-9	Yes
2	[50]	b1	6.39e-12	Yes
2	[50]	b2	1.00e-9	Yes
2	[50]	gamma1	1.48e-9	Yes
2	[50]	beta1	1.40e-9	Yes
4	[50,50,50]	W1	1.99e-9	No
4	[50,50,50]	W2	2.43e-9	No
4	[50,50,50]	W3	2.00e-9	No
4	[50,50,50]	W4	6.88e-10	No
4	[50,50,50]	b1	1.40e-9	No
4	[50,50,50]	b2	1.67e-9	No
4	[50,50,50]	b3	1.71e-9	No
4	[50,50,50]	b4	1.12e-9	No
4	[50,50,50]	W1	8.59e-10	Yes
4	[50,50,50]	W2	2.34e-9	Yes
4	[50,50,50]	W3	3.18e-9	Yes
4	[50,50,50]	W4	1.34e-9	Yes
4	[50,50,50]	b1	1.11e-11	Yes
4	[50,50,50]	b2	7.97e-12	Yes
4	[50,50,50]	b3	9.38e-12	Yes
4	[50,50,50]	b4	1.13e-9	Yes
4	[50,50,50]	gamma1	2.01e-9	Yes
4	[50,50,50]	gamma2	2.39e-9	Yes
4	[50,50,50]	gamma3	1.75e-9	Yes
4	[50,50,50]	beta1	2.81e-9	Yes
4	[50,50,50]	beta2	2.14e-9	Yes
4	[50,50,50]	beta3	1.93e-9	Yes

Table 1: Errors for the gradient norms

2 Batch / no batch loss functions

Loss plots running with the default parameters: n_batch = 100, eta_min = 1e-5, eta_max = 1e-1, lambda =.005, nCycles = 2, n_s = 5 * 45000 / n_batch

2.1 3 layer network

hidden layers = [50,50]

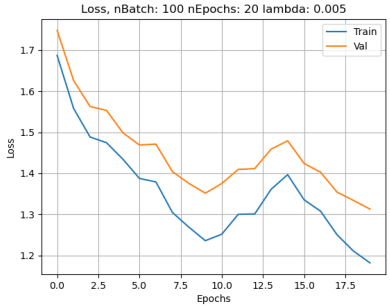

Figure 1: No batch normalization loss

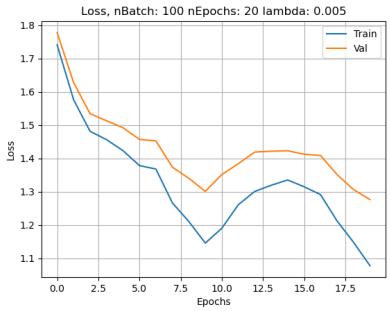

Figure 2: Batch normalization loss

Running the batch normalization config gives smoother plots and a lower loss in the end

2.2 9 layer network

hidden layers = [50,30,20,20,10,10,10,10]

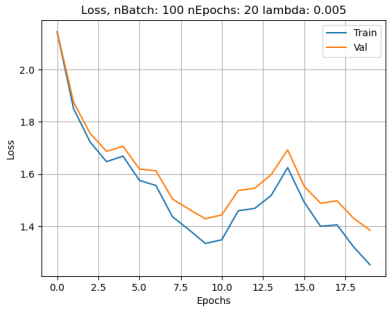

Figure 3: No batch normalization loss

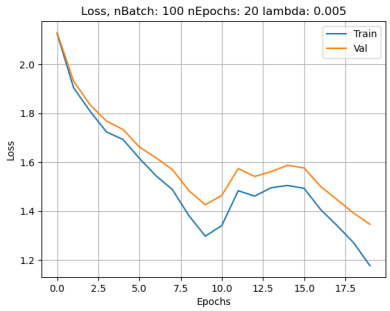

Figure 4: Batch normalization loss

Running the batch normalization config gives smoother plots and a lower loss in the end

3 Lambda optimization for 3-layer network with batch-norm

The lambda value was optimized based on the validation accuracy (5000 images). Similarly to lab2, loglamda (log 10) was first searched from -5 to -1 with 8 values randomly drawn. I ran the training for two cycles. The best configurations were the following:

Loglambda	Validation accuracy
-2.41	0.5518
-2.47	0.5480
-3.10	0.5376

Table 2: Coarse search

Based on the coarse search result, a fine search was conducted in the interval (-3.40, -2.11). The best performing loglamda was -2.3434 at 55.28% accuracy. Training this configuration for 3 cycles using the same parameters, I reached a 54.57% test accuracy and a 55.3% validation accuracy.

4 Sensitivity to initialization

Loss plots running with the default parameters for the 3-layer network: n_batch = 100, eta_min = 1e-5, eta_max = 1e-1, lambda =.005, nCycles = 2, n_s = 5 * 45000 / n_batch, hidden layers = [50,50].

The trend in accuracy for both batchNorm and no batchNorm is that the test and validation accuracy decreases as σ grows smaller, with batchNorm performing better for all three σ-values. Notably, batchNorm does not suffer the same catastrophic accuracy decrease as no batchNorm when $\sigma = 1e{-}4$ (10% test accuracy for no batchNorm).

For batchnorm, the lossplots look ok for all three σ-values.

For the no batchnorm case, it is clear that the model is a lot less robust to poor initialization and it performs worse as σ decreases. At the $\sigma = 1e-4$-level, no meaningful learning seems to take place judging by the loss plot for no batchnorm.

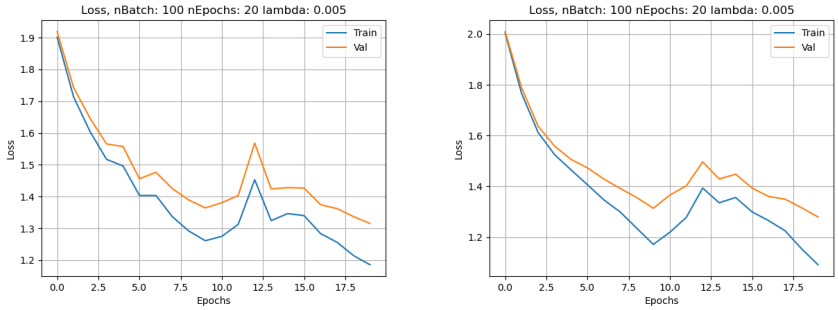

Figure 5: $\sigma = 1e-1$, no batchnorm left, batchnorm right

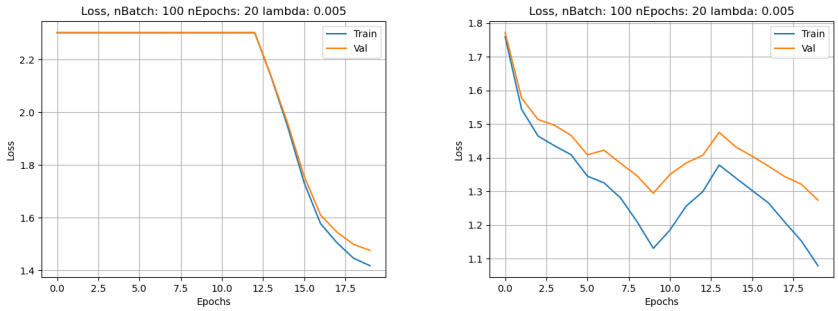

Figure 6: $\sigma = 1e-3$, no batchnorm left, batchnorm right

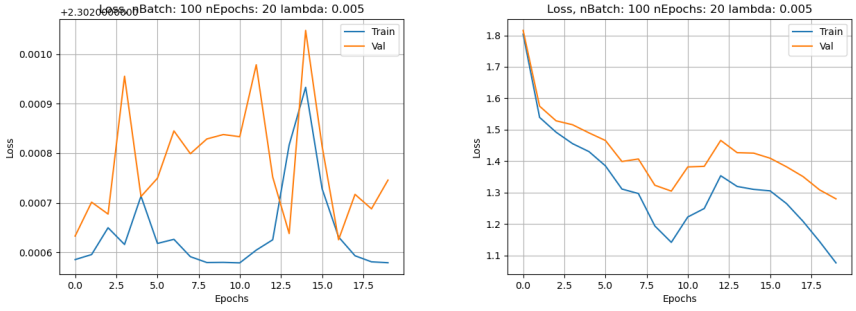

Figure 7: $\sigma = 1e-4$, no batchnorm left, batchnorm right