Why Batch-Norm Parameters Should Not Have Weight Decay

1 Batch-Norm Recap

For a mini-batch $B = \{x_i\}_{i=1}^m$ and weights W of the preceding linear or convolutional layer, the Batch Normalization (BN) process is defined as:

$$\begin{aligned} z_i &= Wx_i, \\ \hat{z}_i &= \frac{z_i - \mu_B}{\sigma_B}, \\ y_i &= \gamma \hat{z}_i + \beta, \end{aligned}$$

where μ_B and σ_B are the batch mean and standard deviation, and γ and β are trainable scale and shift parameters, respectively.

2 Scale-Invariance Introduced by BN

Consider scaling all rows of W by a positive scalar c:

$$W' = cW \implies \hat{z}'_i = \frac{cWx_i - \mu'_B}{\sigma'_B} = \hat{z}_i,$$

since $\mu'_B = c\mu_B$ and $\sigma'_B = c\sigma_B$. Thus, the networks output is invariant to the norm of W; only the direction of the weights matters. Regularizers penalizing the norm of W do not constrain the function implemented by the network.

3 What Weight Decay Does in This Setting

Weight decay (L2 regularization) adds the following term to the loss:

$$\mathcal{L}_{\text{WD}} = \frac{\lambda}{2} \left(\|W\|_2^2 + \gamma^2 + \beta^2 \right).$$

Since the data loss is invariant to $\|W\|$, the true optimum lies on a ray $\{(cW, \gamma, \beta) \mid c > 0\}$. Adding $\mathcal{L}_{\text{WD}}$ breaks this symmetry, driving the optimizer to satisfy:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial W} &= 0, \\ \frac{\partial \mathcal{L}}{\partial \gamma} + \lambda\gamma &= 0, \\ \frac{\partial \mathcal{L}}{\partial \beta} + \lambda\beta &= 0. \end{aligned}$$

Since $\frac{\partial \mathcal{L}}{\partial \gamma}$ and $\frac{\partial \mathcal{L}}{\partial \beta}$ are often small near convergence, these equations force $\gamma, \beta \to 0$. This collapses the post-BN activations (since $\text{Var}(y) = \gamma^2$), reducing representational power. The network may rescale earlier weights to compensate, leaving performance unchanged but worsening training dynamics.

4 Effective-Learning-Rate Distortion

With weight decay, each SGD or Adam update for a BN parameter is:

$$\Delta\gamma = -\eta \left(\frac{\partial \mathcal{L}}{\partial \gamma} + \lambda\gamma \right).$$

When $\lambda\gamma$ dominates, the effective learning rate becomes:

$$\eta_{\text{eff}} = \eta \frac{\frac{\partial \mathcal{L}}{\partial \gamma}}{\frac{\partial \mathcal{L}}{\partial \gamma} + \lambda\gamma}.$$

This behaves like a much smaller learning rate. Similarly, for W, the term λW affects only the norm, not the direction, competing with the gradient and harming convergence without regularizing the function.

5 Practical Takeaway

To avoid these issues:

- Place all BN parameters (γ, β) and often the weights feeding directly into BN into a separate optimizer group with $\lambda = 0$.
- Apply weight decay only to parameters whose scale affects the networks output (e.g., convolution or linear weights not immediately normalized).

This preserves the intended regularization effect of weight decay while avoiding meaningless shrinkage of scale/shift terms and associated optimization pathologies.