Gradient norms and Gradient vanashing

- During backpropigation the model computes the gradients (see gradient notes) the gradient norm is a single scalar that summarizes the magnitude of all gradients accross the model, typically computed as the L2 (Euclidean) norm: ($\|g\|_2 = \sqrt{\sum_i g_i^2}$) where g_i are the indivisual gradient components. the subscript of "2" denotes L2 norm (standard way to measure size or vector) there are also other norms (L1, L∞, etc) but L_2 is most common.

- interpritation of Gradient norm: Healthy gradient norms are typically neither too close too zero nor excessively large. Very small norms (ex: <1e-5) indicate gradient vanishing problems, basically when grad shrink to near 0 values when this happens gradients needed to update early layers almost become 0 when that happens learning stops in those early layers they freeze while later layers do the learning (see pg 119 of how sigmoid activation function can cause this). Very large norms (ex: >100 or exploding to NaN) indicate exploding gradients when this happens weights explode as they update by huge ammounts causing loss to spike or be NaN, model outputs to overflow and training to be unstable. this often requires gradient clipping, gradient clipping is a technique that caps the gradient at a threshold (or clip grad norm to max val of 1.0 or 5.0) to prevent exploding gradients (commonly used in RNN/transformers) monitor gradient norms to find issues in training.

NN specific Evaluations (like Grad norms)

- Parameter count: Higher means more model capacity, but too many may lead to model overfitting or inefficient models, balance param count with dataset size & task complexity
- Throughput: Higher throughput (samples processed per sec) is better as it indicates faster training or inference, watch for bottlenecks in hardware or datapipelines if Low.
- inference latency: lower latency (time per sample) is better, especially for real life applications, High val means inefficiencies in model or hardware
- weight distribution: healthy weigh dist should not be overly concentrated (eg all weights near zero) or expssively spred out. Saturated weights (very large or small) indicate dead neurons or poor init. ideal case is that the weight distribution follows a bell curve (norm dist) on a weight count vs weight value graph
- training/validation curves: training loss should decrease steadily, and validation loss should follow a similar trend. Diverging curves may be overfitting or poor generalization (on a graph of Loss vs Epoch for train and validation loss)