ENPM703- Assignment-2

Part2: Batchnorm

Architecture

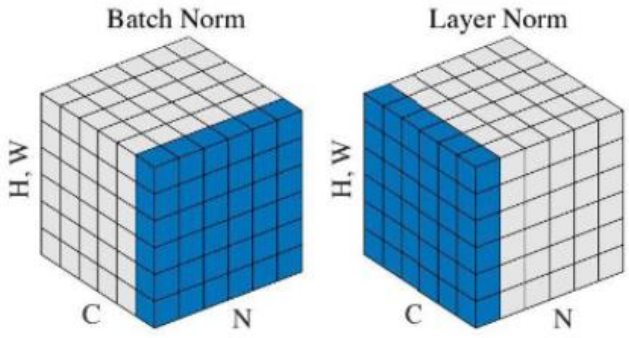
Affine: Each layer starts with an affine transformation, multiplying the input by a weight matrix and adding a bias, resulting in a linear combination that transforms the input into a new feature space.

BN: Normalization layers stabilize and speed up training by adjusting the inputs to each layer, making the activations more consistent. Batch Normalization normalizes over the entire mini-batch, while LayerNorm normalizes across the features of each individual data point. LayerNorm is often used when BN is not suitable, like in models with small batches.

ReLU: The ReLU activation adds non-linearity, passing positive inputs unchanged and outputting zero for negatives, allowing the network to learn complex, non-linear patterns.

This layer is repeated for the first L−1 layers, refining and abstracting the data through multiple hidden layers. The last affine layer computes class scores, and the softmax function converts them into probabilities, ensuring a proper distribution across classes.

Batchnorm



Ref: Zaki, George. Module 07B: Training Neural Networks Part 2. University of Maryland.

Batch Normalization normalizes across the whole mini-batch by calculating the mean and variance for each feature from all the samples in the batch. This helps reduce internal covariate shift. But, it relies on the batch size, which can make it less effective when the batches are small.

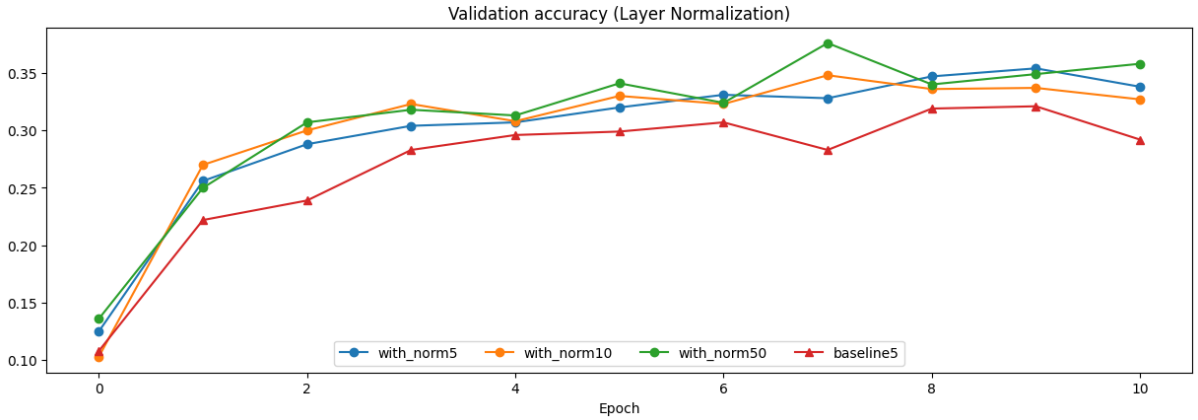
We also compare the results of vanilla batchnorm and simplified batch norm. Here, simplified batchnorm gives better results since it reduces the number of calculations and simplifies the operations required to compute gradients especially when processing large batches of data during training.

Layernorm

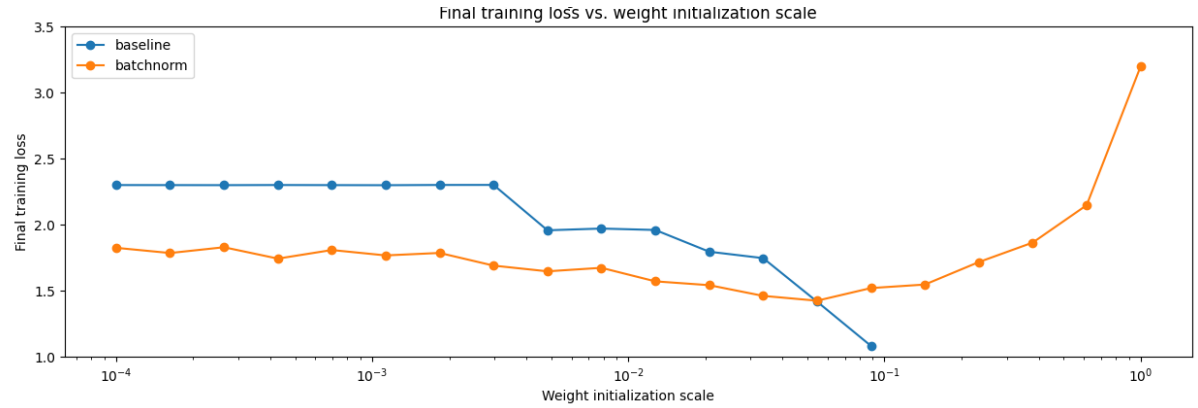
Layer Normalization normalizes the features of a single data point instead of the whole batch. This makes it independent of batch size, making it better suited for tasks when using small batches.

Why batch norm works?

By calculating the mean and variance for each feature across the mini-batch, BatchNorm helps keep the inputs to the activation functions at a consistent distribution which prevents problems like vanishing gradients. This normalization reduces internal covariate shift, where the distribution of inputs to the layers change during training helping the model learn better. BatchNorm also adds learnable parameters to scale and shift the normalized outputs, allowing the network to adjust the features dynamically. As a result, BatchNorm often results in fast convergence, better performance, and ability to use higher learning rates.



- Also having larger batch size corresponds to improved accuracy since it provides a better statistical representation of the data.



- A general trend is batchnorm is less sensitive to weight initialization than the baseline model.