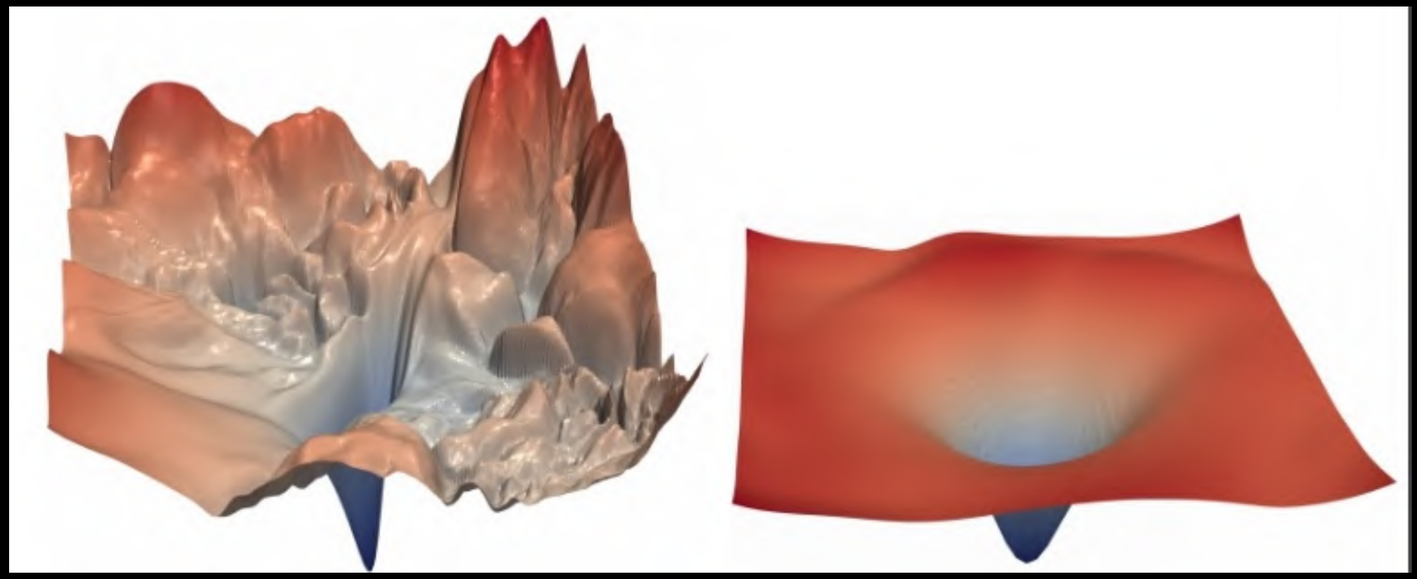
STATISTICAL METHODS IN AI

HOW DOES BATCH NORMALIZATION HELP OPTIMIZATION?

Team Name:ai-scream

Mentor TA: Tathagatho Roy

Github Repo Link:
https://github.com/Statistical-Methods-in-AI-Monsoon-2023/project-ai-scream/tree/main



WHAT IS BATCH NORMALIZATION

Batch normalization is a technique that makes training faster and more stable in deep neural networks.
Initially, it was thought that the reason for batch norms effectiveness was because it reduced the internal covariate shift.
What batch normalization does is it normalizes the inputs before passing them to a non linear activation function.
Because of this, the loss converges faster, and a larger learning rate can be used to train the model.

Link to the paper: https://arxiv.org/pdf/1805.11604.pdf

INTERNAL COVARIATE SHIFT

Internal covariate shift is the phenomenon wherein the distribution of inputs to a layer in the network changes due to an update of parameters of the previous layers. This change leads to a constant shift of the underlying training problem and is thus believed to have detrimental effect on the training process.

Mathematically, internal covariate shift is defined as:

Definition 2.1. *Let $\mathcal{L}$ be the loss, $W_1^{(t)}, \ldots, W_k^{(t)}$ be the parameters of each of the k layers and $(x^{(t)}, y^{(t)})$ be the batch of input-label pairs used to train the network at time t. We define* internal covariate shift (ICS) *of activation i at time t to be the difference $||G_{t,i} - G'_{t,i}||_2$, where*

$$G_{t,i} = \nabla_{W_i^{(t)}} \mathcal{L}(W_1^{(t)}, \ldots, W_k^{(t)}; x^{(t)}, y^{(t)})$$

$$G'_{t,i} = \nabla_{W_i^{(t)}} \mathcal{L}(W_1^{(t+1)}, \ldots, W_{i-1}^{(t+1)}, W_i^{(t)}, W_{i+1}^{(t)}, \ldots, W_k^{(t)}; x^{(t)}, y^{(t)}).$$

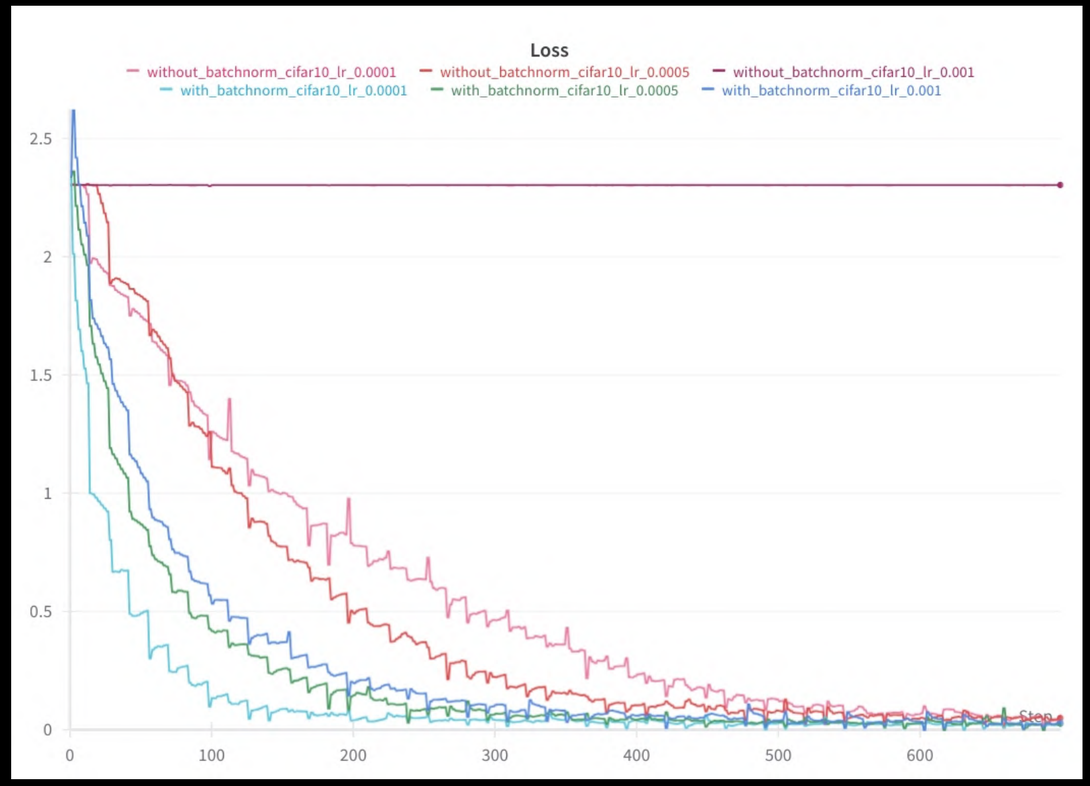
Here, $G_{t,i}$ corresponds to the gradient of the layer parameters that would be applied during a simultaneous update of all layers (as is typical). On the other hand, $G'_{t,i}$ is the same gradient *after* all the previous layers have been updated with their new values. The difference between G and G' thus reflects the change in the optimization landscape of W_i caused by the changes to its input. It thus captures precisely the effect of cross-layer dependencies that could be problematic for training.

EXPERIMENT 1

Training a standard VGG architecture on a dataset with and without BatchNorm

We built two standard VGG16 models, one with batch norm layers and the other without, and trained them on CIFAR10 and CIFAR100.
With both the datasets, the loss converged significantly faster with batchnorm, as is visible from the graphs.

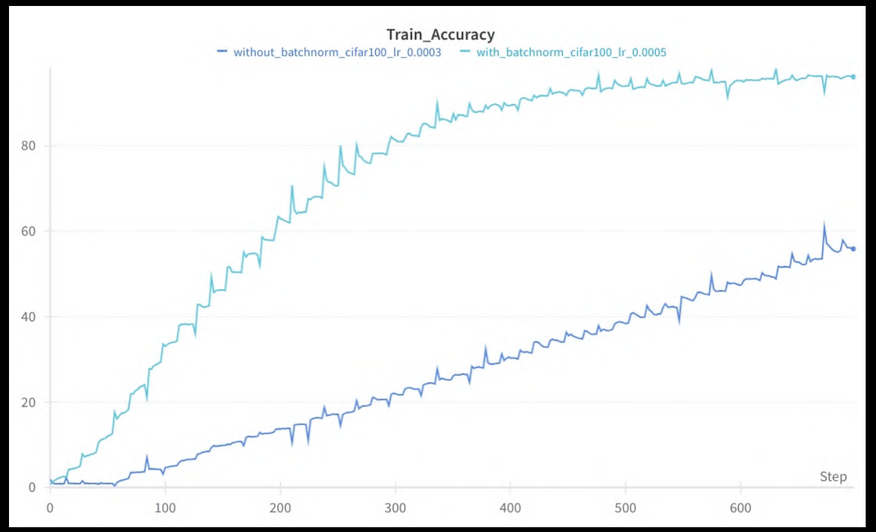
For a learning rate of 0.001, the model that is trained without batchnorm does not update its parameters at all.



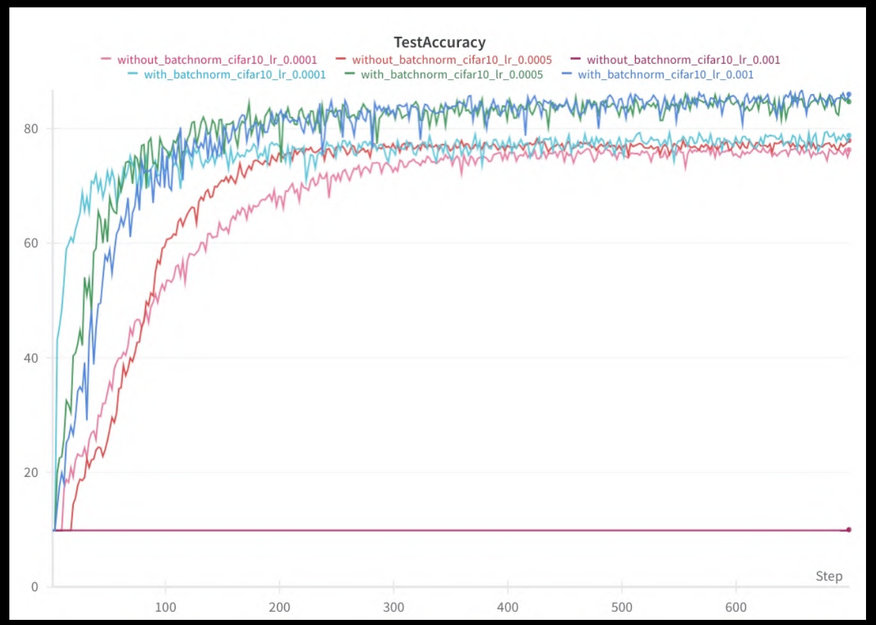
Loss vs steps graph for models with and without batchnorm for different learning rates

EXPERIMENT 1

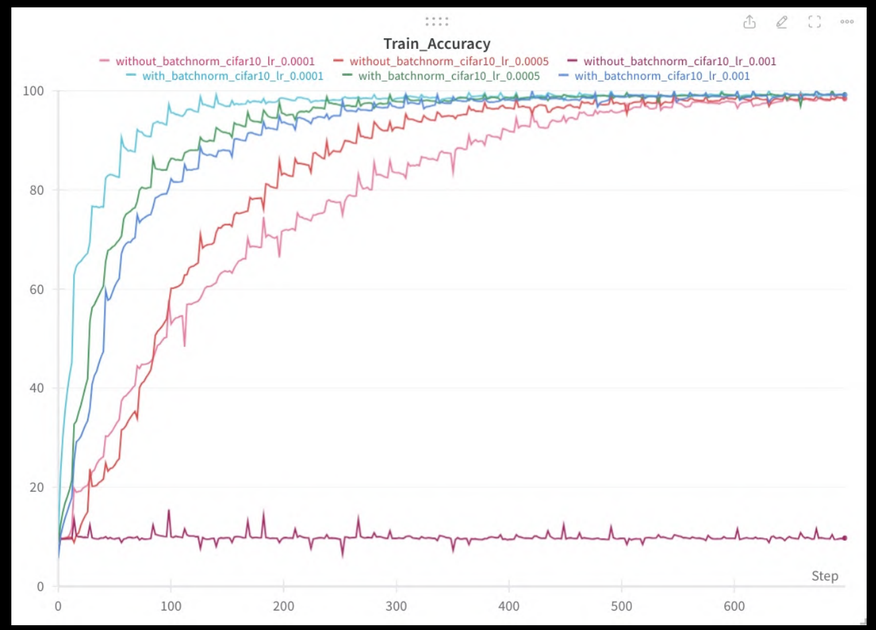
Training a standard VGG architecture on a dataset with and without BatchNorm



Accuracy graph for CIFAR100



As we can see, the models with batchnorm perform significantly better

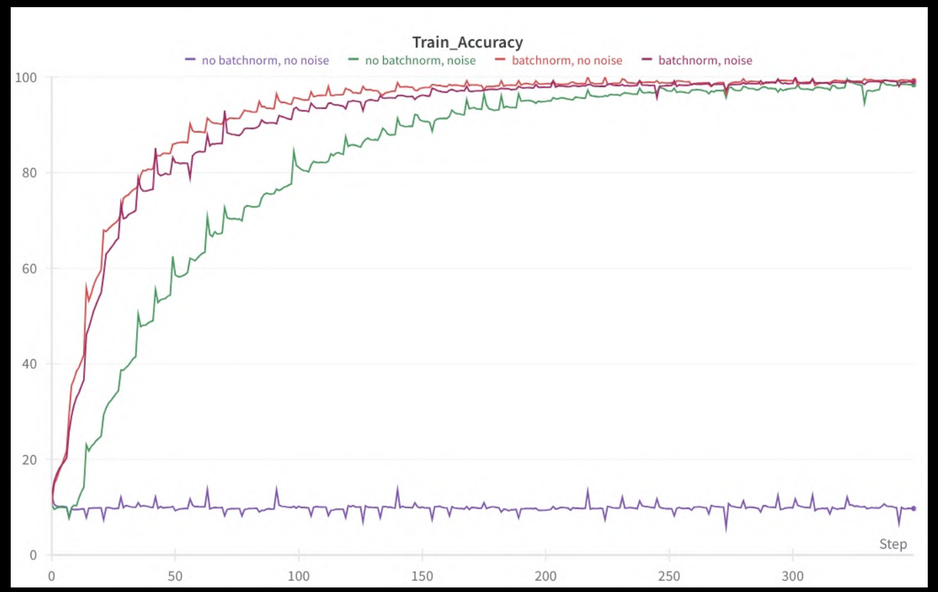


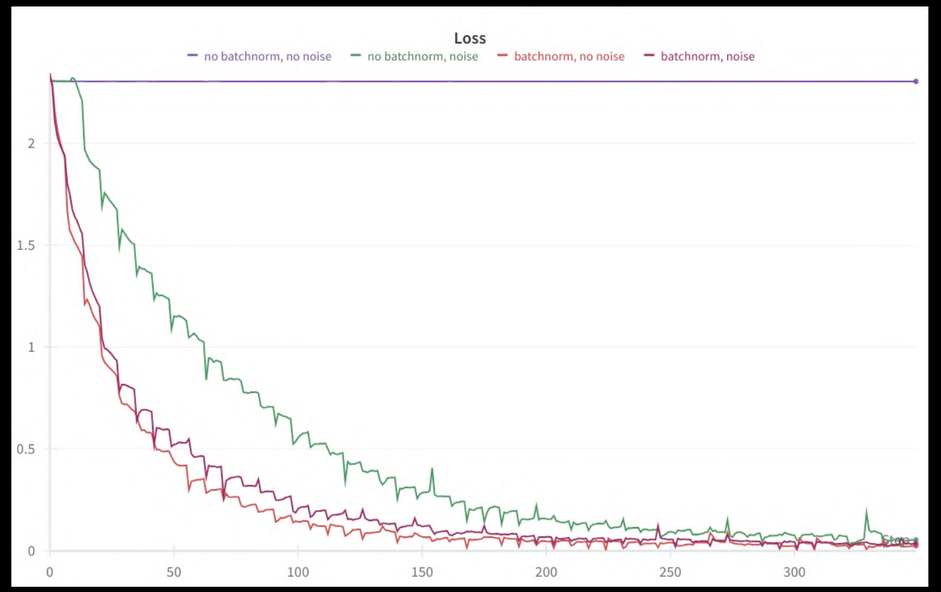
Even here, for train accuracy, the models with batchnorm outperform the others

EXPERIMENT 2

Checking if Batchnorm reduces ICS

For the first part of this experiment, we trained the same model used previously, but this time noise was added after the batchnorm layers.
This effectively increases the "shift" in the inputs to the activation function, and depending on the loss and accuracy curve can be used to determine whether this shift affects the performance of the model.



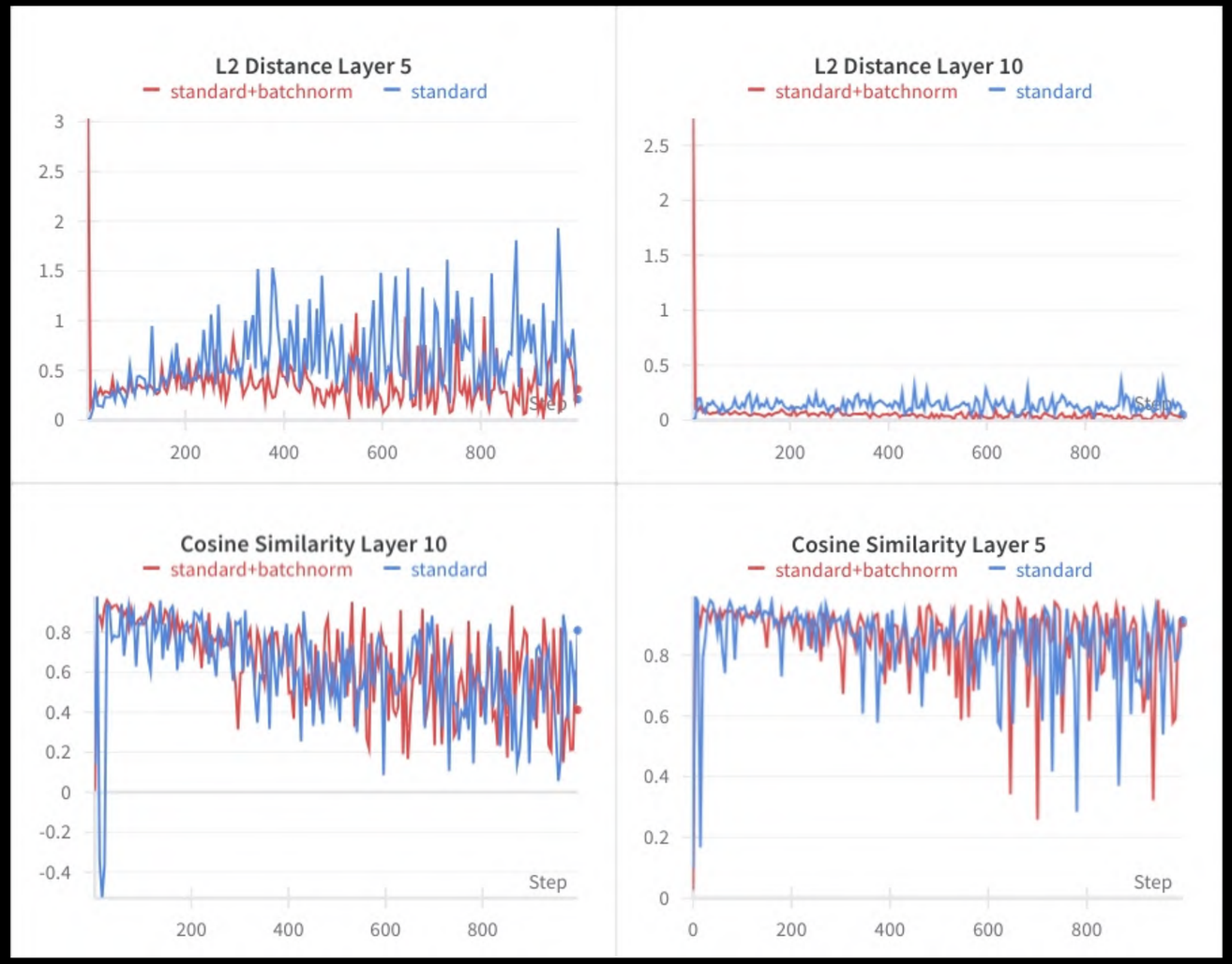


With added noise, the models with batchnorm and batchnorm+noise perform better than the models without. The model with just noise does not converge at all

EXPERIMENT 2

Checking if BatchNorm reduces ICS

For the second part, we measured ICS, for layer 5 and 10. For a layer the cosine angle and the l2 difference of the gradients was measured, before and after updates to the preceding layers.
Models with batchnorm are observed to have similar, or in some cases worse internal covariate shift, even though they converge faster.
This shows that batchnorm doesnt affect the internal covariate shift.

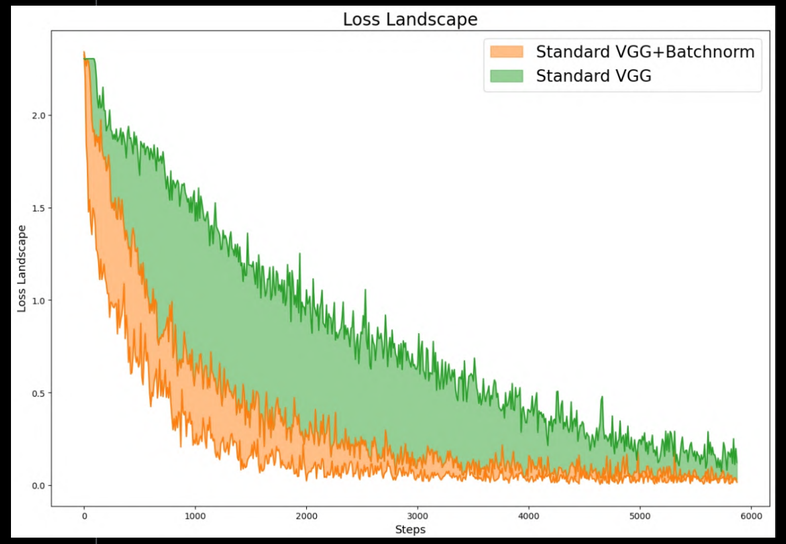


Cosine and L2 similarity of gradients for layer 5 and 10

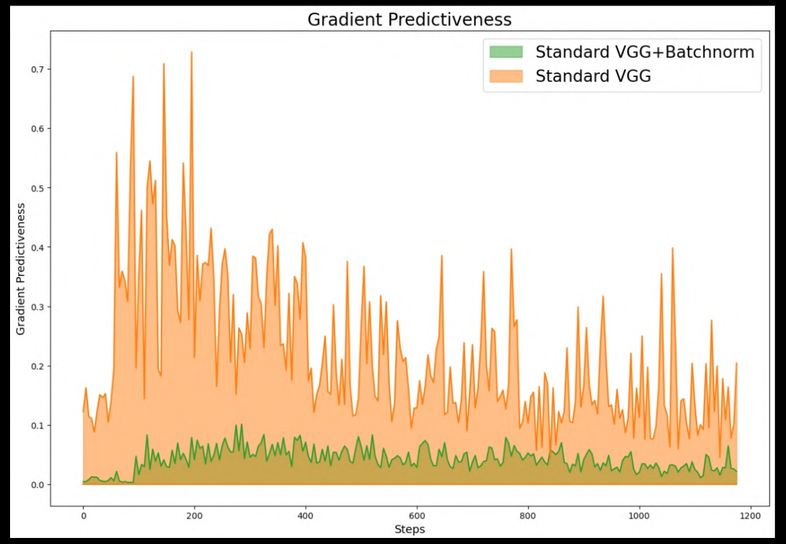
EXPERIMENT 3

Checking what BatchNorm actually reduces

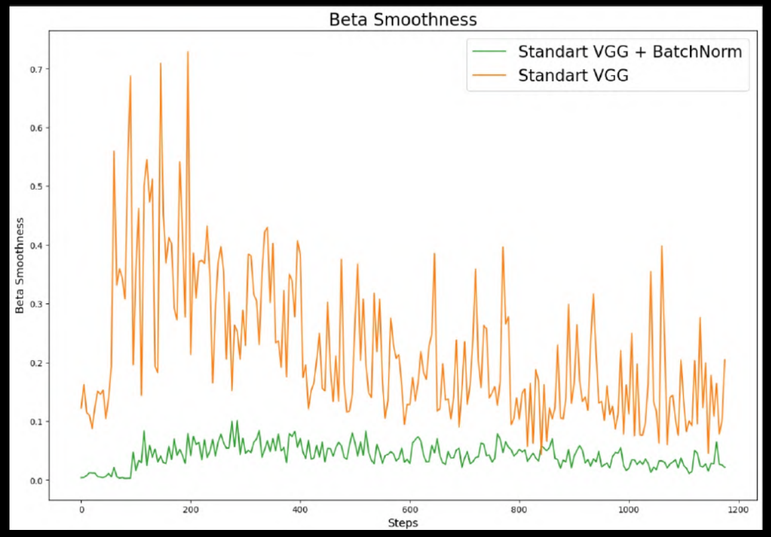
For the first part of this experiment, we trained the same model used in experiment 1, and measured the loss and the gradient of the loss function with respect to the last layers parameters for 2 different learning rates to measure how BatchNorm affects the loss landscape and the "predictiveness" of the gradient.



Loss Landscape vs Steps



Gradient Predictiveness vs Steps



Beta Smoothness vs Steps

EXPERIMENT 3

Checking what BatchNorm actually reduces

Beta Smoothness:
Beta Smoothness refers to the maximum difference (in l2-norm) in gradient over distance moved in that direction.

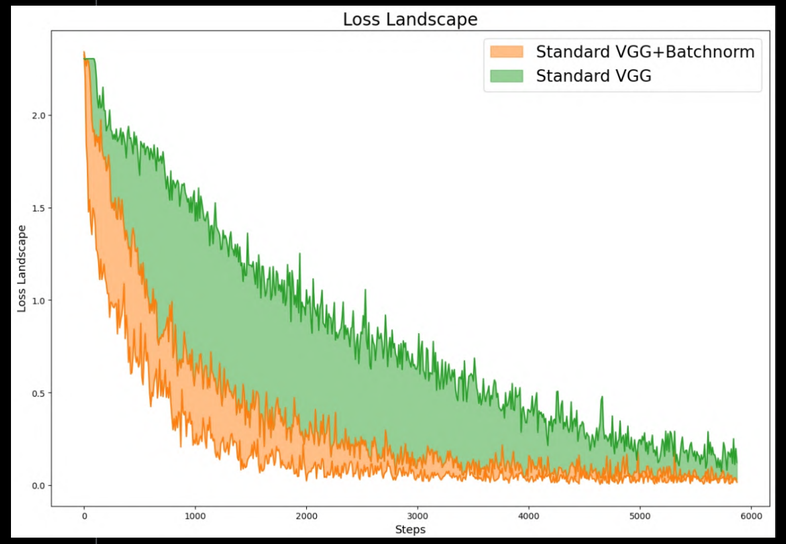
Gradient Predictiveness:
Gradient Predictiveness refers to the l2 changes in the gradient as we move in the gradient direction.

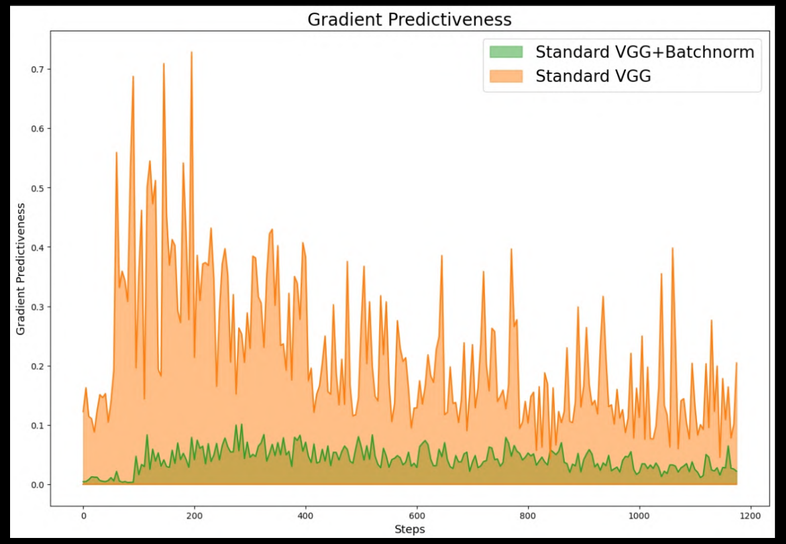
Loss Landscape:
At a particular training step, the variation in loss .

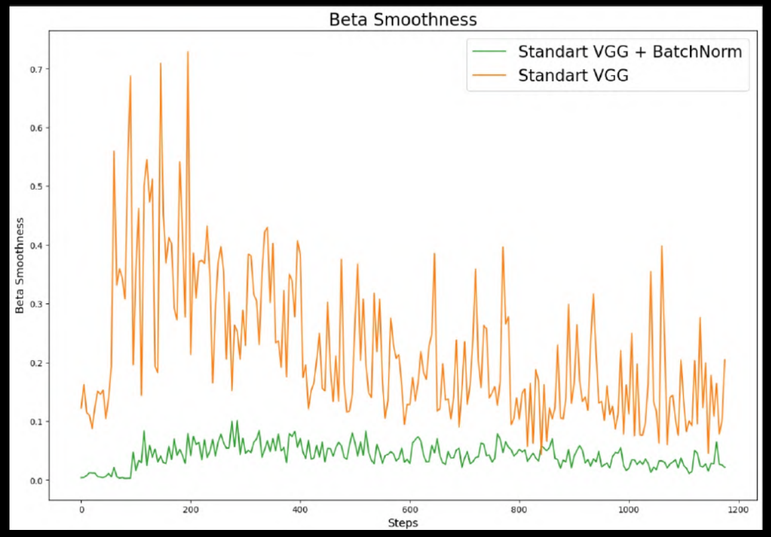
EXPERIMENT 3

Checking what BatchNorm actually reduces

From the graphs we can see that the models trained with BatchNorm has smaller variations in gradients when compared to a model trained without BatchNorm.
The loss landscape of the model trained with BatchNorm is also smoother than the model trained without Batchnorm.







EFFECTIVENESS OF BATCHNORM

Checking what BatchNorm actually reduces

Beta Smoothness:
Beta Smoothness refers to the maximum difference (in l2-norm) in gradient over distance moved in that direction.

Gradient Predictiveness:
Gradient Predictiveness refers to the l2 changes in the gradient as we move in the gradient direction.

Loss Landscape:
At a particular training step, the variation in loss .

TEAM MEMBERS

Maanasa Kovuru

Sanika Damle

Contributions:

Experiment 1: Maanasa and Sanika

Experiment 2: Sanika

Experiment 3: Maanasa

Pitch

Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It's free)