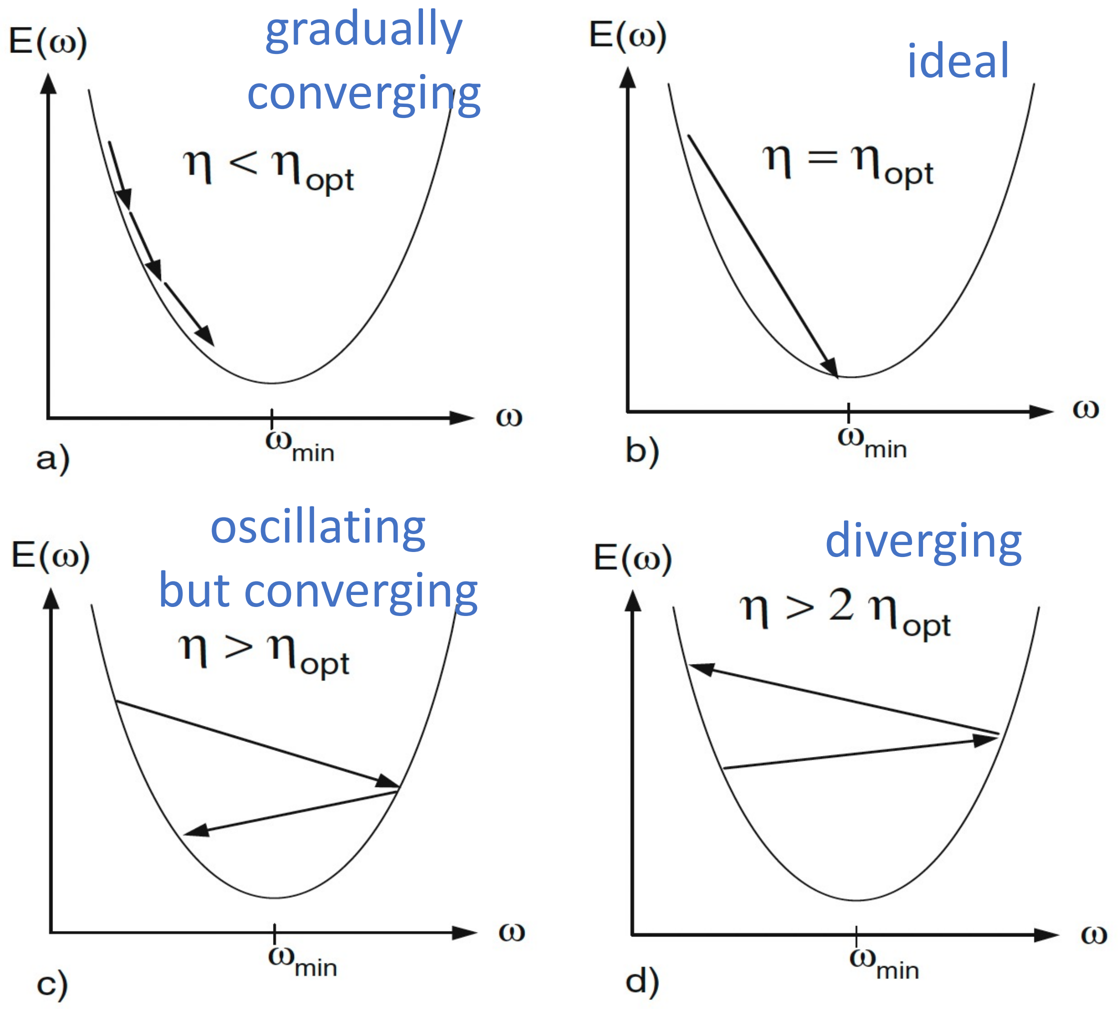
Normalization Modules

Learning Rate in Gradient Descent

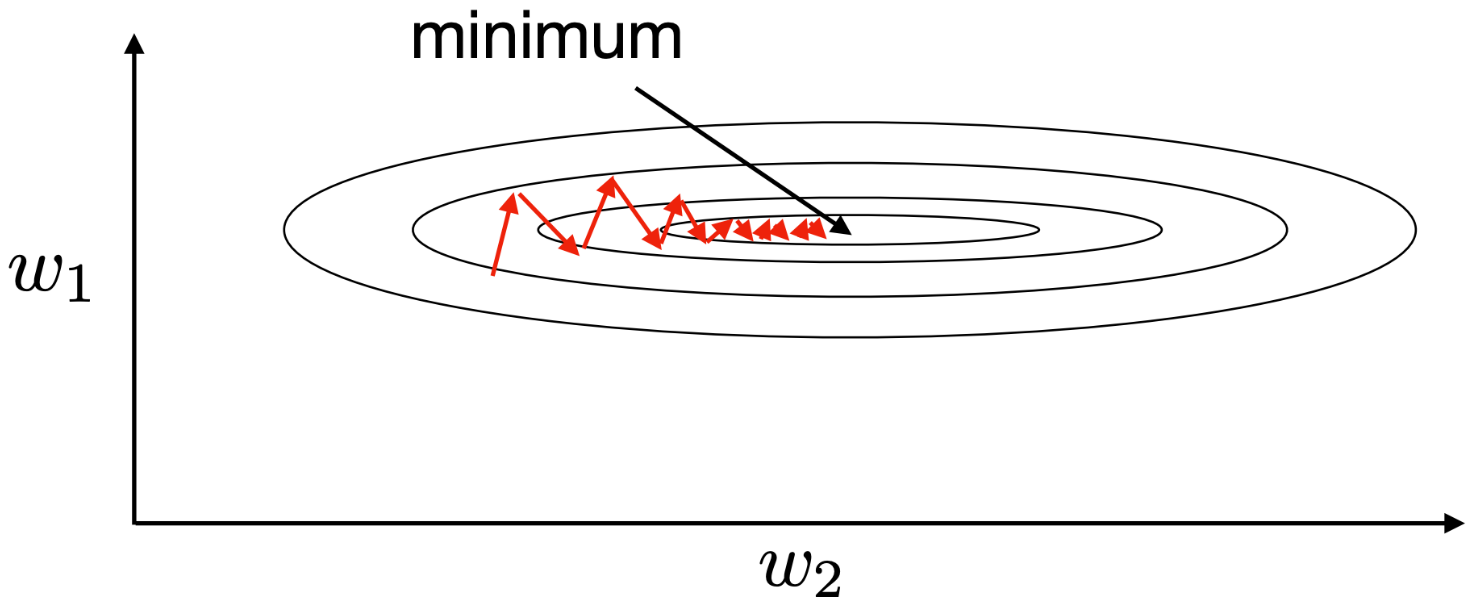
$$W := W - \eta \frac{\partial \mathcal{E}}{\partial W}$$

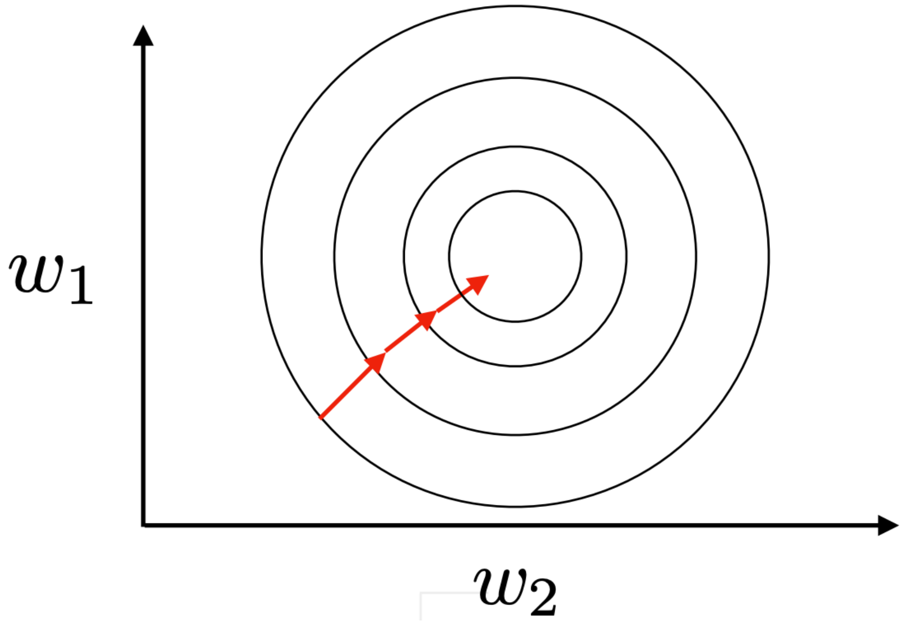
η: learning rate



LeCun et al 1998 “Efficient Backprop”

Normalization





- Same learning rate applied to all weights
- Large weights dominate updates
- Small weights oscillate (or diverge)

- Similar pace for all weights

Figures adapted from Sebastian Raschka: L11.1 Input Normalization

Input Normalization

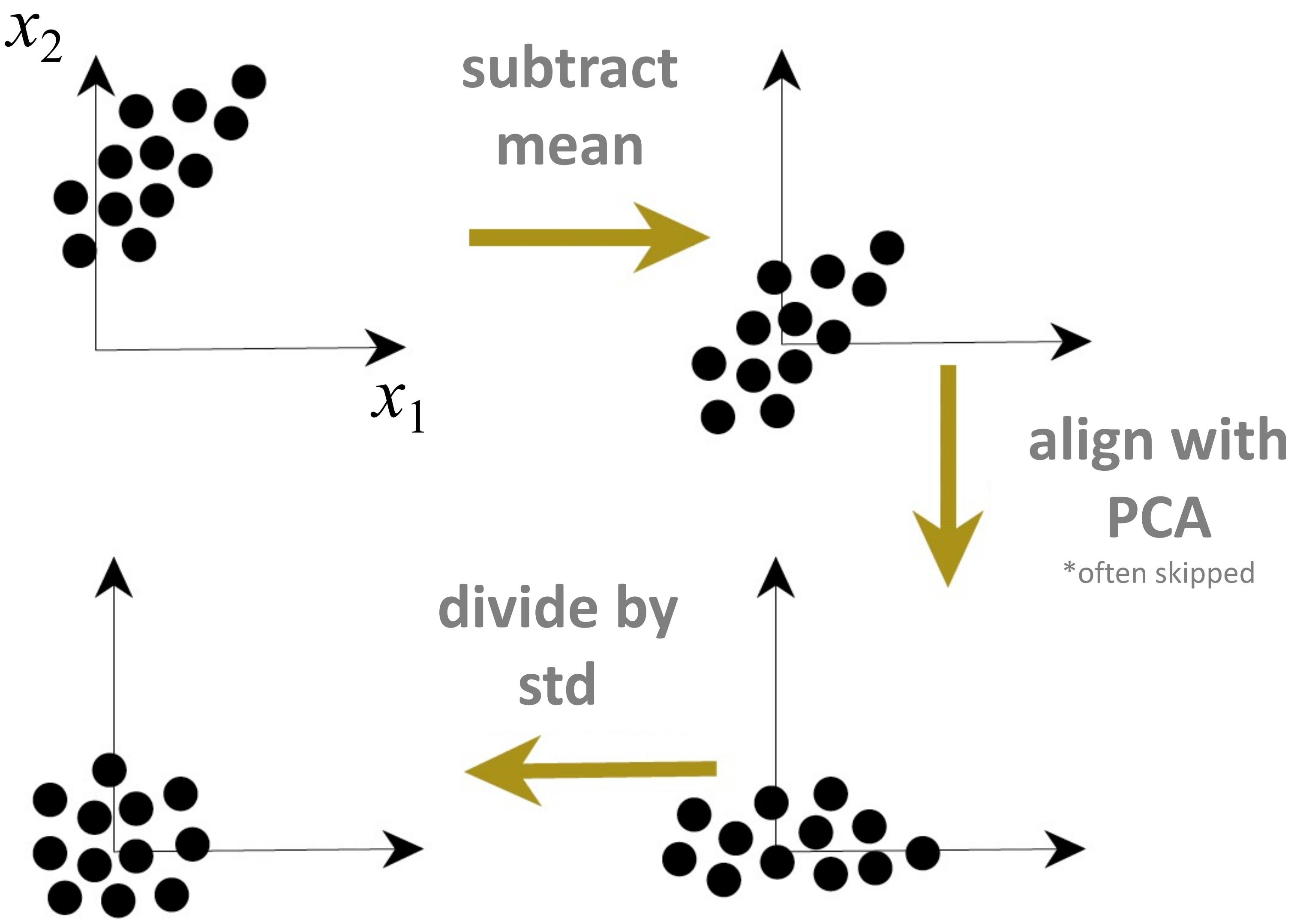
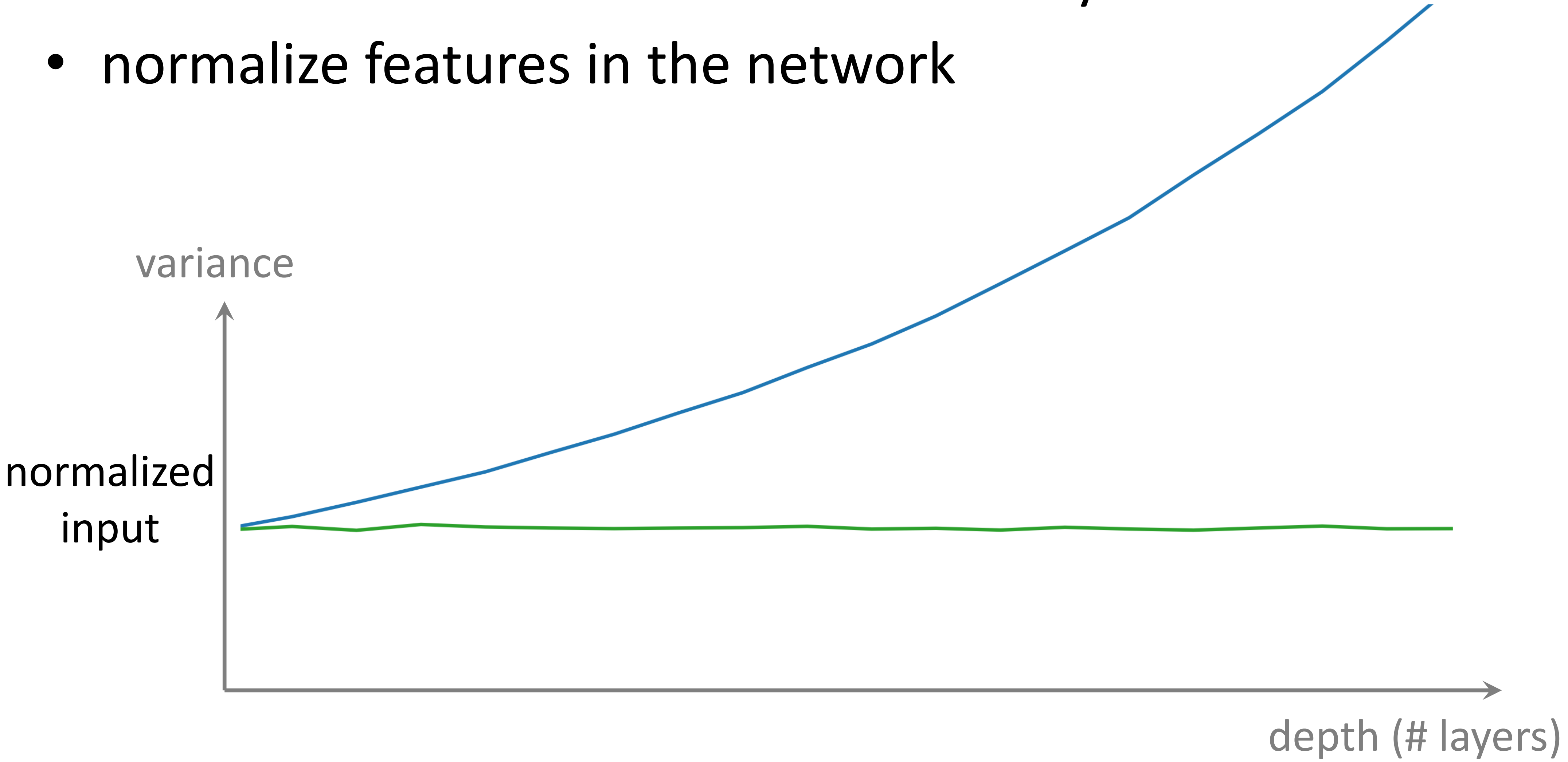


Figure adapted from: LeCun et al. “Efficient BackProp”. 1998.

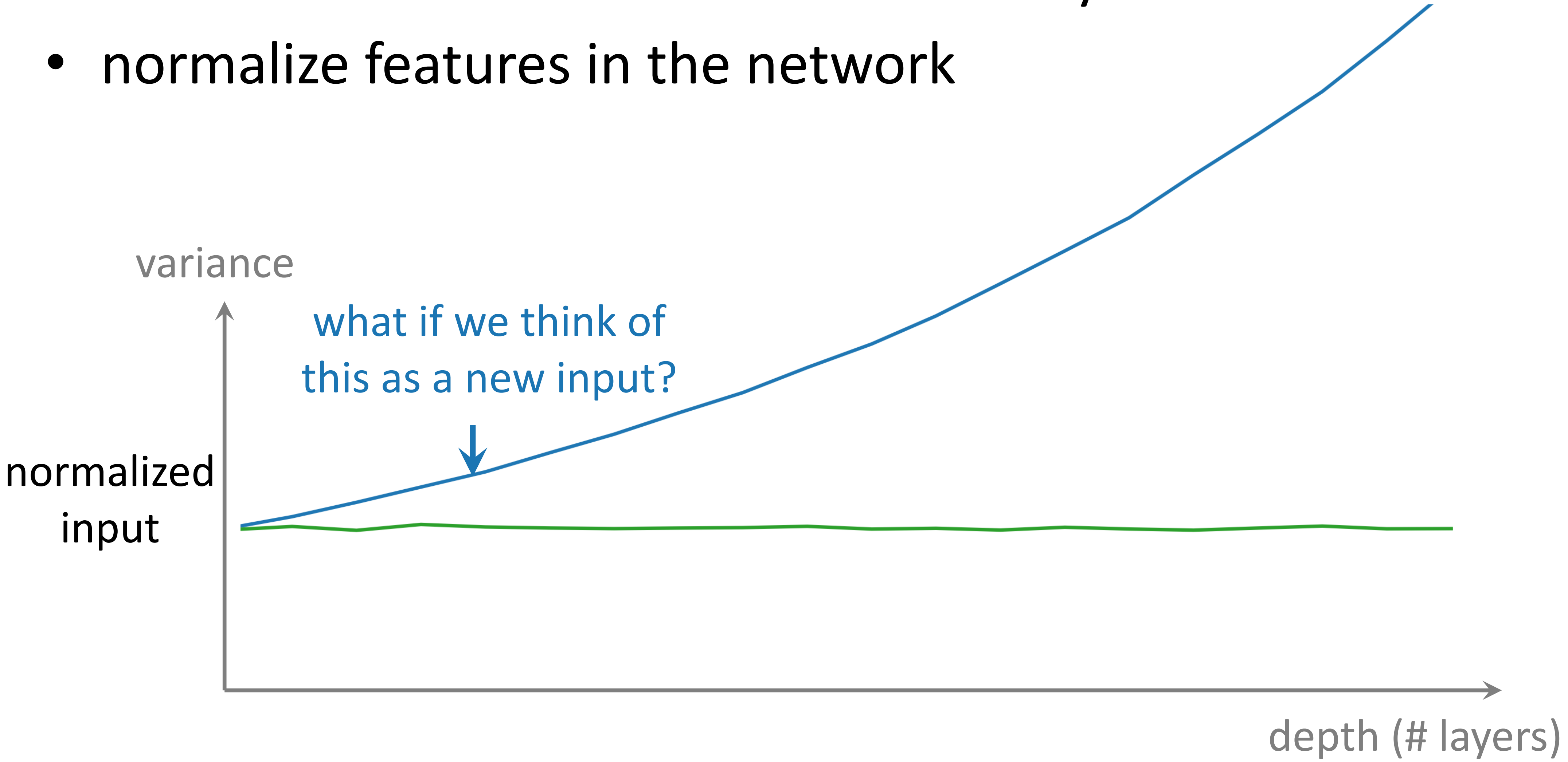
Normalization Modules

- We want to maintain variance for all layers
- normalize features in the network



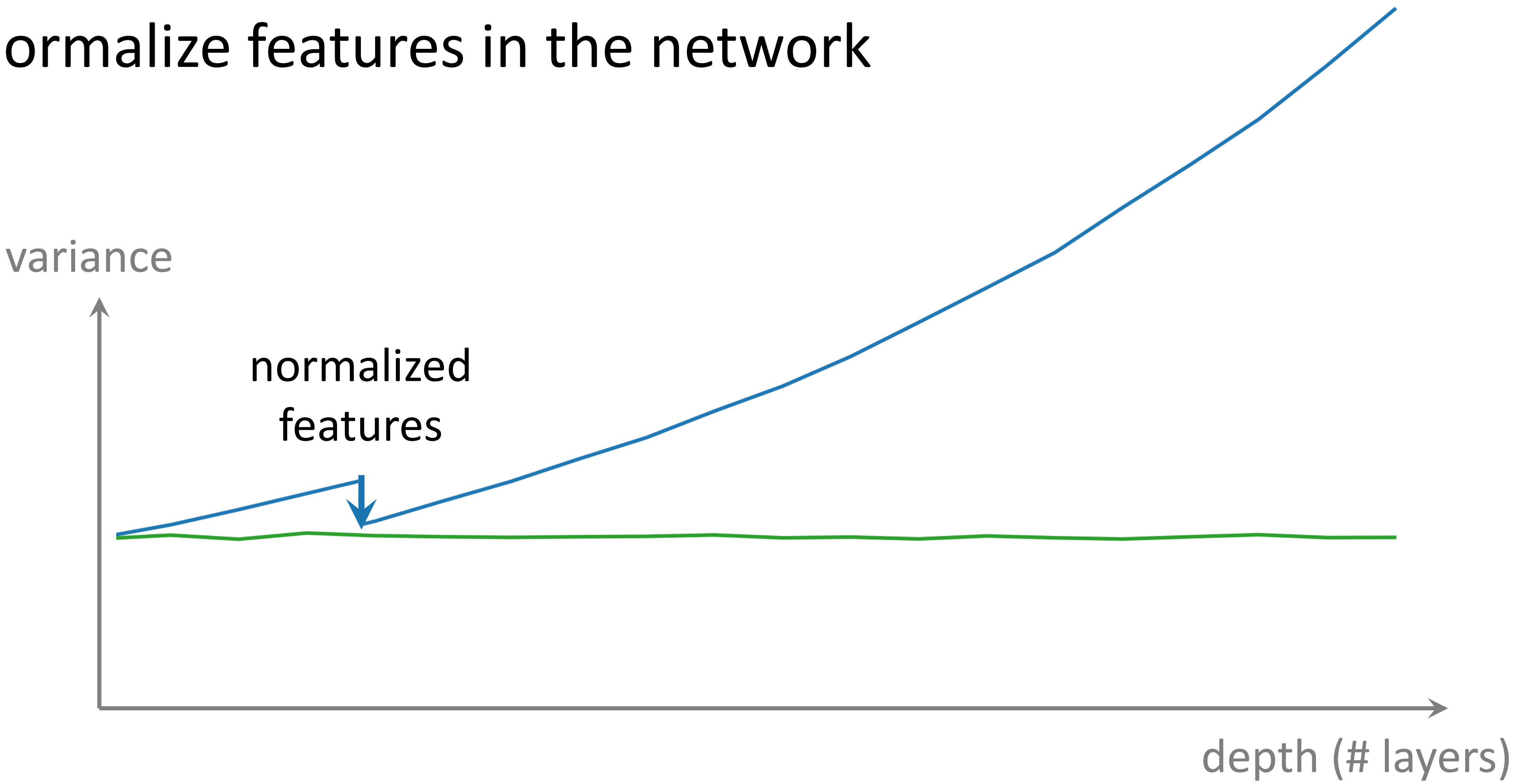
Normalization Modules

- We want to maintain variance for all layers
- normalize features in the network



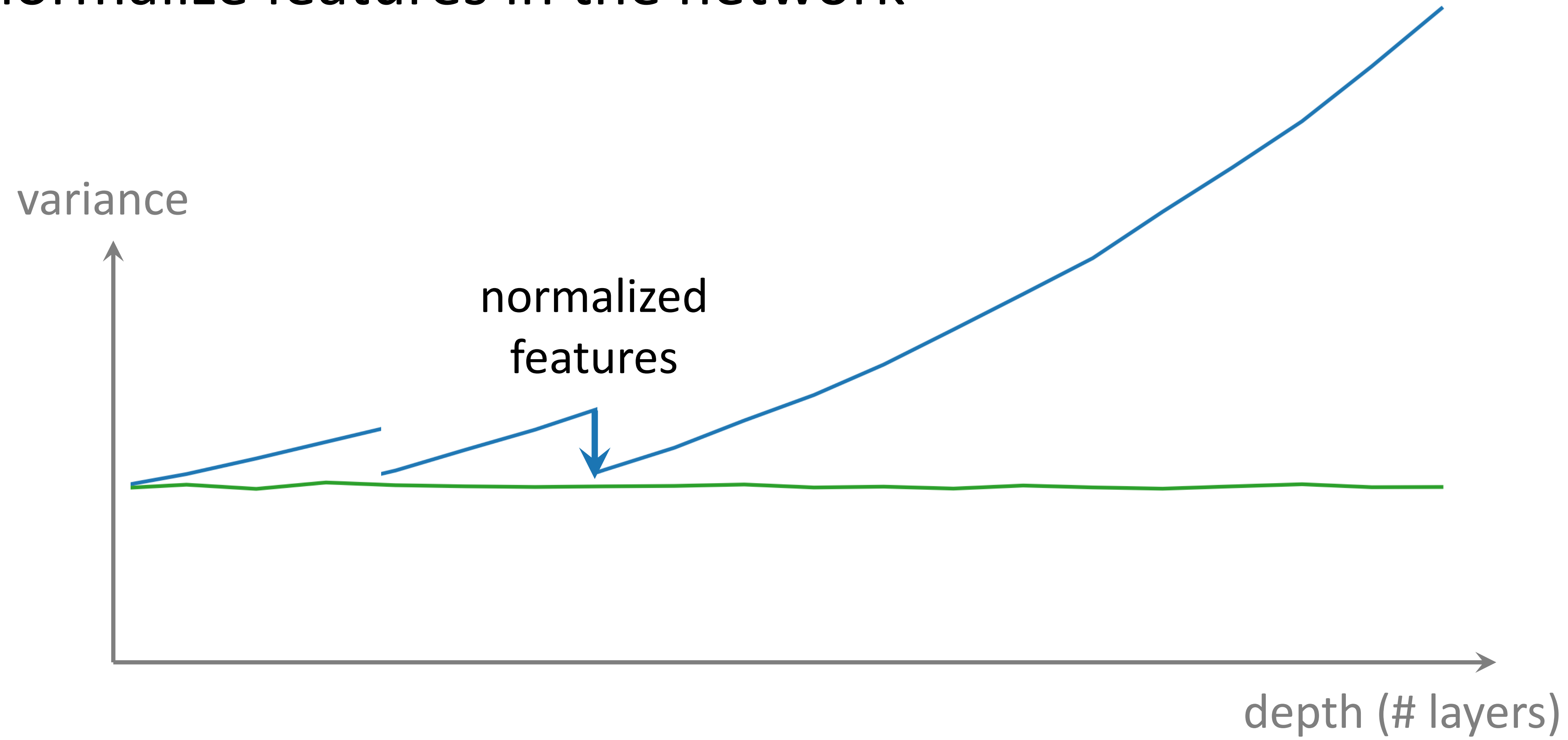
Normalization Modules

- We want to maintain variance for all layers
- normalize features in the network



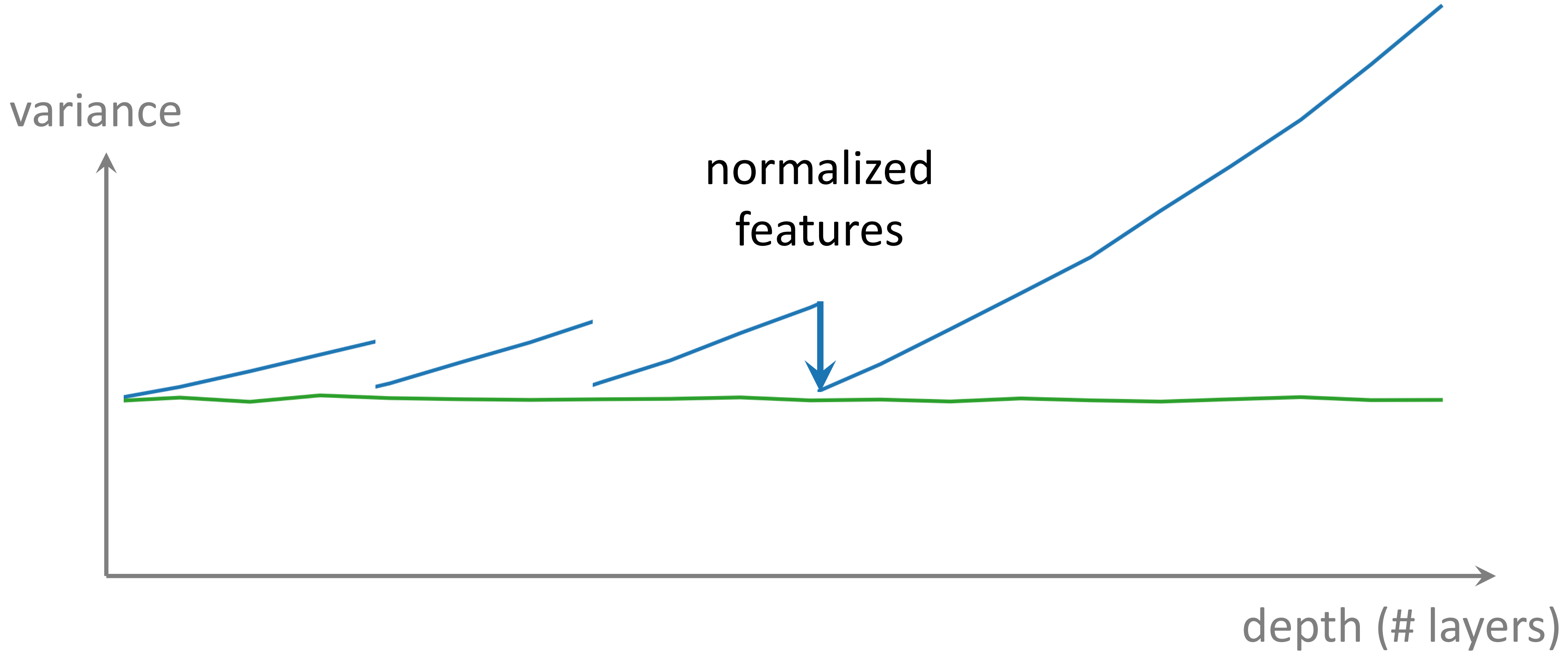
Normalization Modules

- We want to maintain variance for all layers
- normalize features in the network



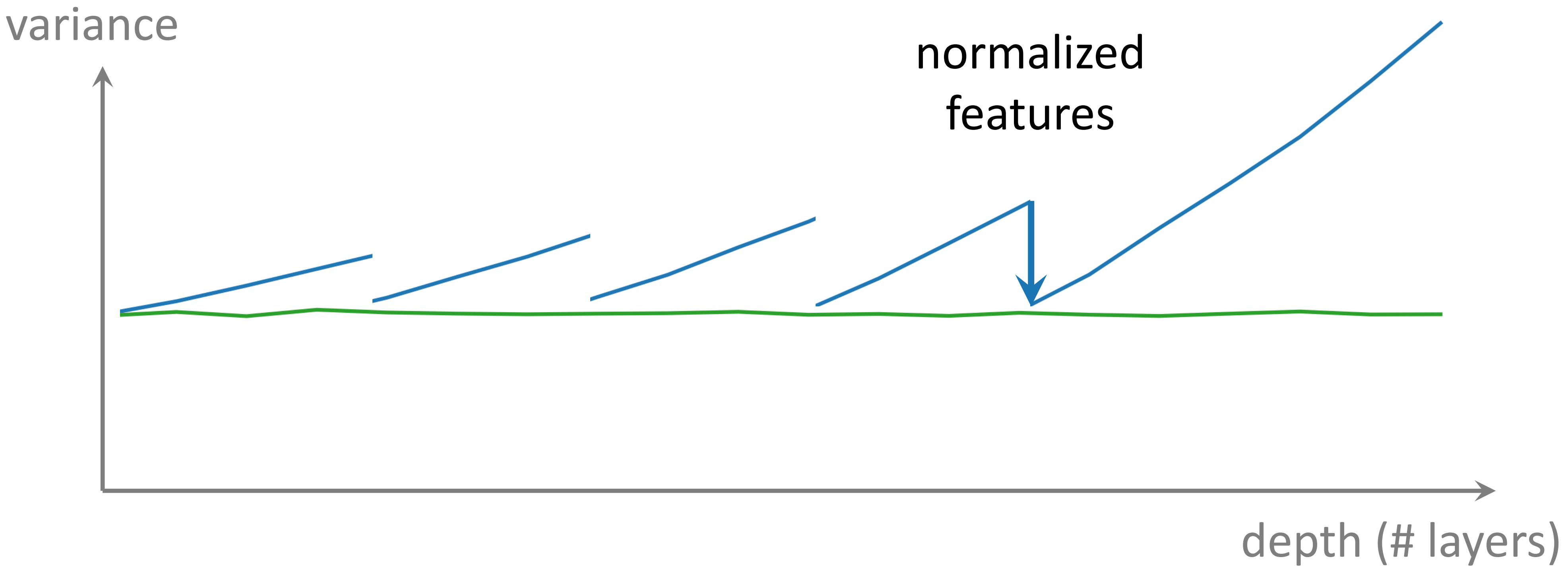
Normalization Modules

- We want to maintain variance for all layers
- normalize features in the network



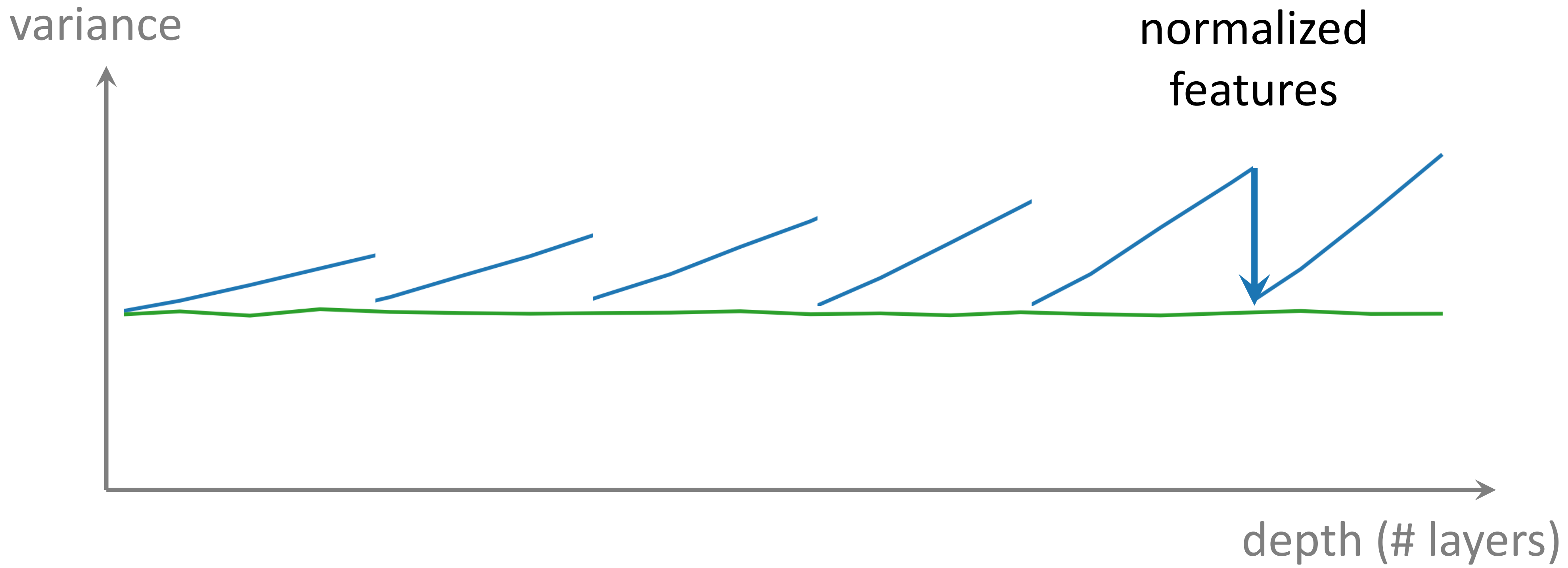
Normalization Modules

- We want to maintain variance for all layers
- normalize features in the network



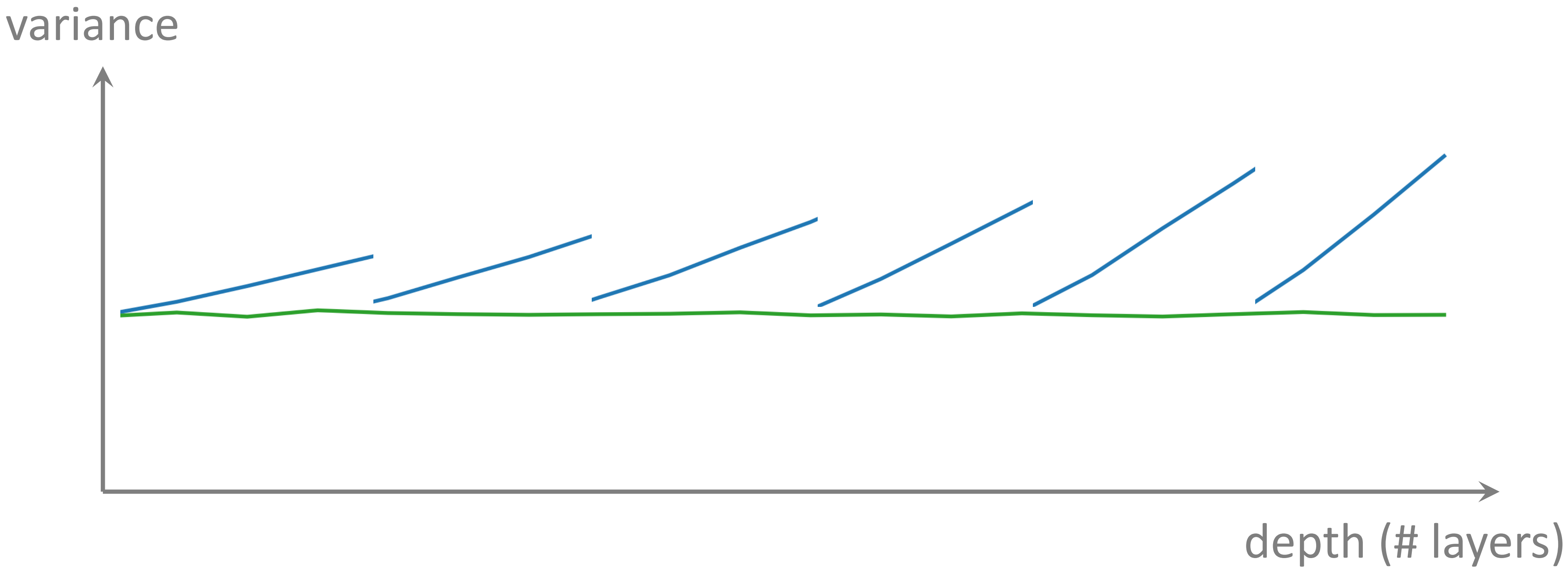
Normalization Modules

- We want to maintain variance for all layers
- normalize features in the network



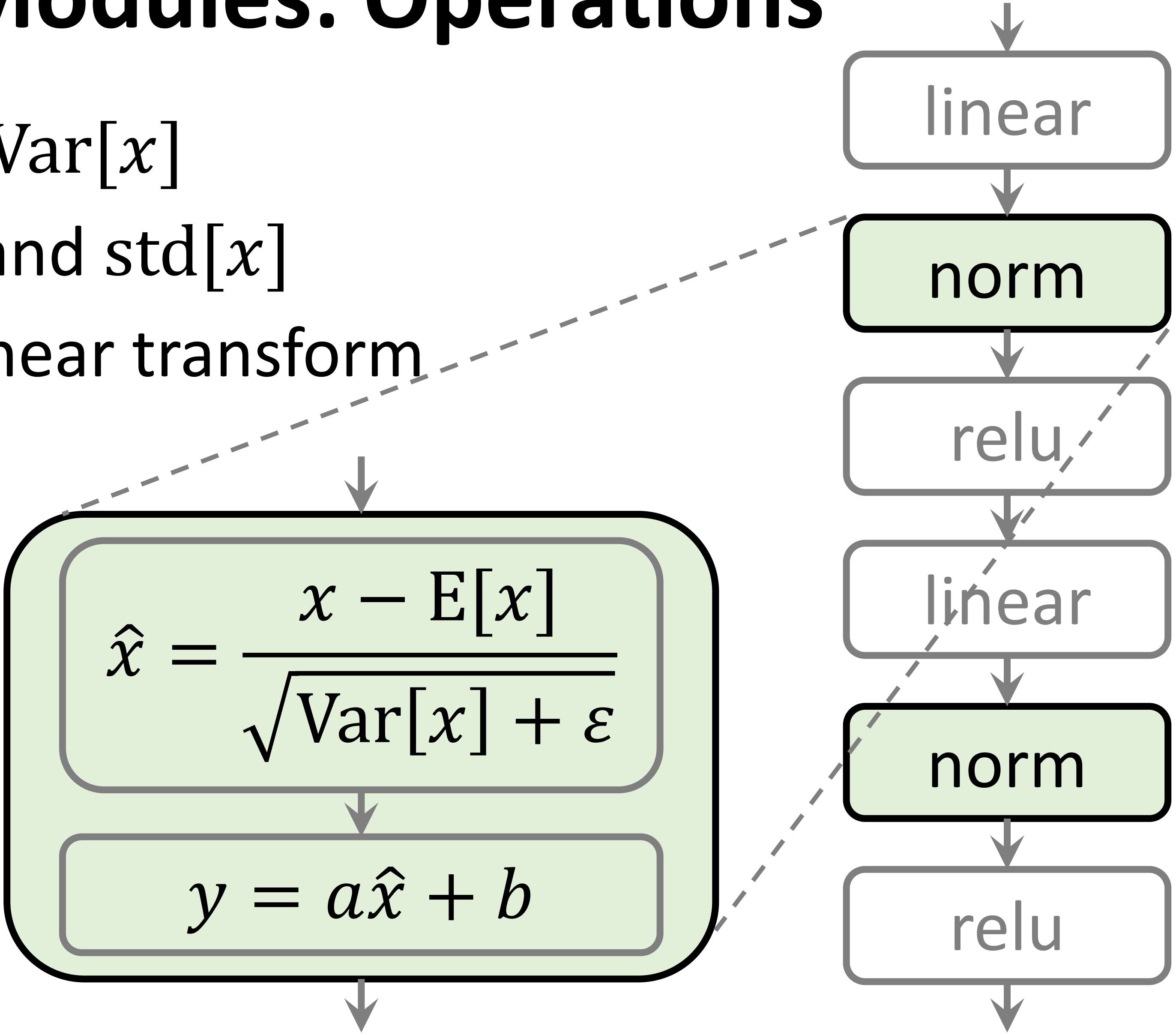
Normalization Modules

- We want to maintain variance for all layers
- normalize features in the network
- train end-to-end by BackProp



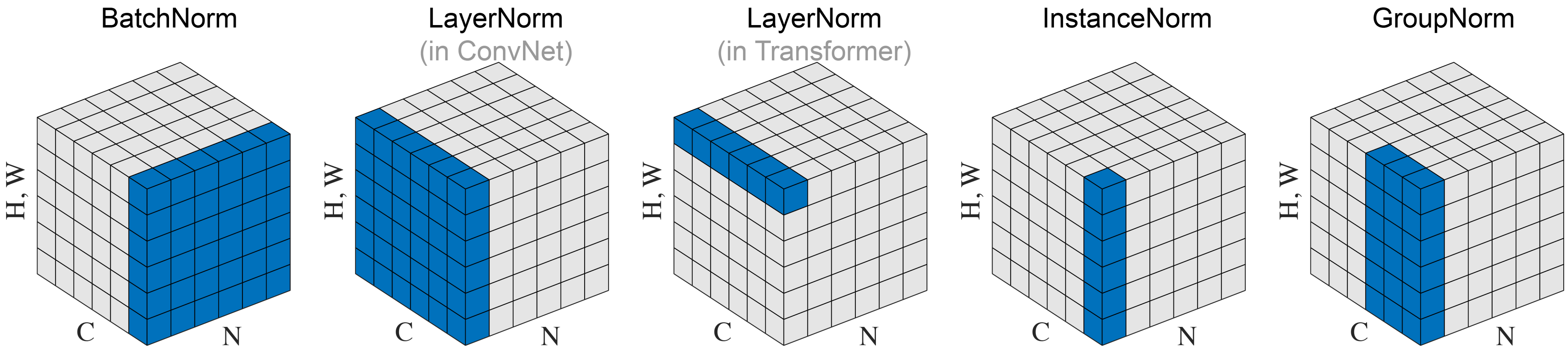
Normalization Modules: Operations

1. compute $\mathrm{E}[x]$ and $\mathrm{Var}[x]$
2. normalize by $\mathrm{E}[x]$ and $\mathrm{std}[x]$
3. compensate by a linear transform



Normalization Modules: Variants

differ in support sets of $\mathrm{E}[x], \mathrm{Var}[x]$



Ioffe & Szegedy, 2015; Ba, Kiros, Hinton, 2016; Ulyanov, Vedaldi, Lempitsky, 2016; Wu & He, 2018