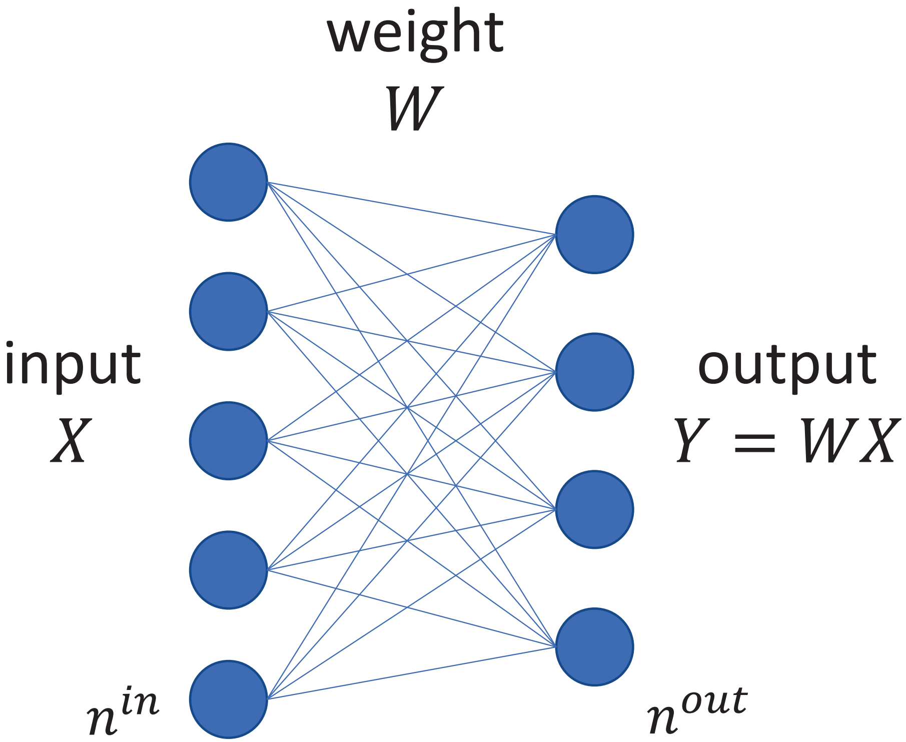
Initialization



If:

- Linear activation
- x, y, w: independent

Then:

1-layer:

$$Var[y] = (n^{in} Var[w]) Var[x]$$

Multi-layer:

$$Var[y] = \left(\prod_{d} n_d^{in} Var[w_d]\right) Var[x]$$

LeCun et al 1998 "Efficient Backprop"
Glorot & Bengio 2010 "Understanding the difficulty of training deep feedforward neural networks"

Initialization

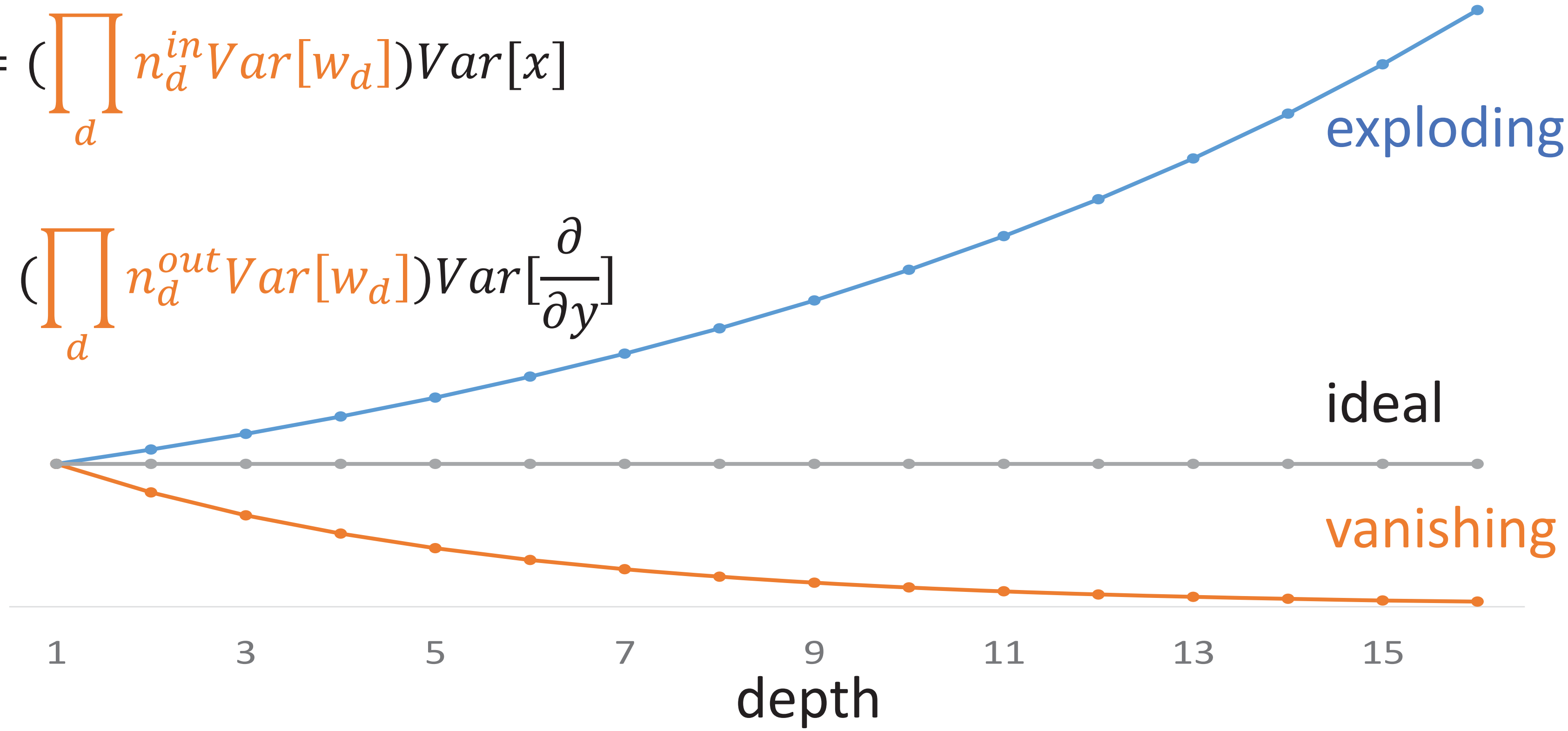
Both forward (response) and backward (gradient) signal can vanish/explode

Forward:

$$Var[y] = \left(\prod_d n_d^{in} Var[w_d]\right) Var[x]$$

Backward:

$$Var\left[\frac{\partial}{\partial x}\right] = \left(\prod_d n_d^{out} Var[w_d]\right) Var\left[\frac{\partial}{\partial y}\right]$$



LeCun et al 1998 “Efficient Backprop”

Glorot & Bengio 2010 “Understanding the difficulty of training deep feedforward neural networks”

Initialization: "Xavier"

- Initialization under **linear** assumption

$$\prod_d n_d^{in} Var[w_d] = const_{\mathrm{fw}} \text{ (healthy forward)}$$

and

$$\prod_d n_d^{out} Var[w_d] = const_{\mathrm{bw}} \text{(healthy backward)}$$

$$\Rightarrow \boxed{\begin{array}{c} n_d^{in} Var[w_d] = 1 \\ \text{or} \\ n_d^{out} Var[w_d] = 1 \end{array}}$$

LeCun et al 1998 "Efficient Backprop"
Glorot & Bengio 2010 "Understanding the difficulty of training deep feedforward neural networks"

Initialization: "MSRA"

- Initialization under **ReLU**

$$\prod_d \frac{1}{2} n_d^{in} Var[w_d] = const_{\text{fw}} \text{ (healthy forward)}$$

and

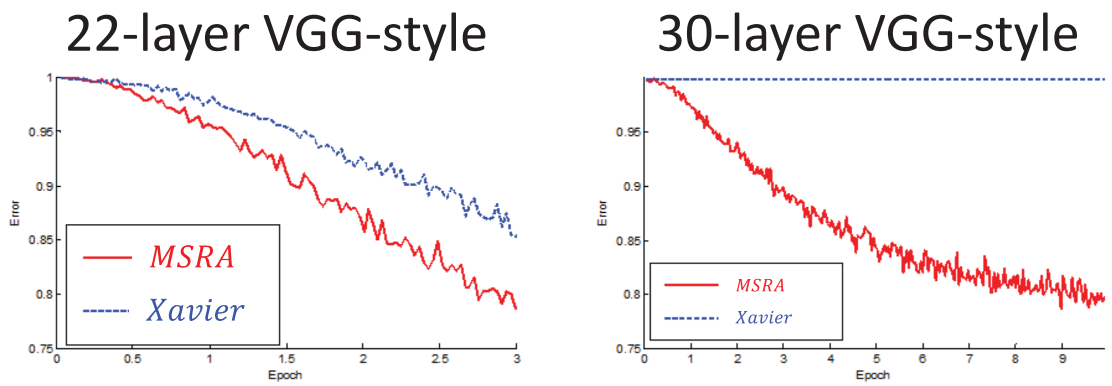
$$\prod_d \frac{1}{2} n_d^{out} Var[w_d] = const_{\text{bw}} \text{(healthy backward)}$$

➡ $$\boxed{\begin{gathered} \frac{1}{2} n_d^{in} Var[w_d] = 1 \\ \text{or} \\ \frac{1}{2} n_d^{out} Var[w_d] = 1 \end{gathered}}$$

With D layers, a factor of 2 per layer has exponential impact of 2^D

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". ICCV 2015.

Initialization



*Figures show the beginning of training

Xavier/MSRA init

- Required for training VGG-16/19 from scratch
- Deeper (>20) VGG-style nets can be trained w/ MSRA init
  - but deeper plain nets are not better (see ResNets)
- Recommended for newly initialized layers in fine-tuning
  - e.g., Fast/er RCNN, FCN, etc.

- $\sqrt{\frac{1}{n}}$ or $\sqrt{\frac{2}{n}}$ doesn't directly apply to multi-branch nets (e.g., GoogleNet)
  - but the same derivation methodology is applicable
  - does not matter, if BN is applicable…

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". ICCV 2015.