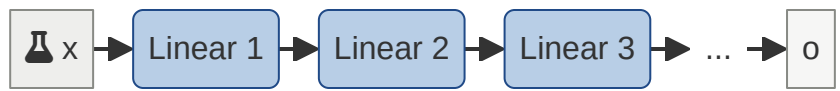
Vanishing and Exploding Gradients

A Simple Example

n-layer linear network

- No non-linearities
- Scalar weight $w_i \in \mathbb{R}, w_i \geq 0$
- Bias $b_i \in \mathbb{R}$



Activations

Assume $w_i \approx w \quad \forall_i$

Case 1: $w < 1$

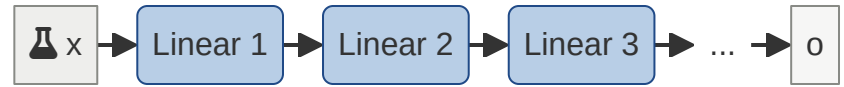
- $w^n x \to 0$ for large n
- Vanishing inputs: $a_n \approx \frac{b}{1-w}$

Case 2: $w = 1$

- $a_n = x + nb$

Case 3: $w > 1$

- Activation n: $w^n \to \infty$
- Exploding activations



Layer	Activation
$l = 1$	$a_1 \approx wx + b$
$l = 2$	$a_2 \approx w^2 x + (w+1)b$
$l = 3$	$a_3 \approx w^3 x + (w^2 + w + 1)b$
$l = n$	$a_n \approx w^n x + b \underbrace{\sum_{k=0}^{n-1} w^k}_{\frac{1-w^n}{1-w}}$

Gradients

Assume $w_i \approx w \quad \forall_i$

Case 1: $w < 1$

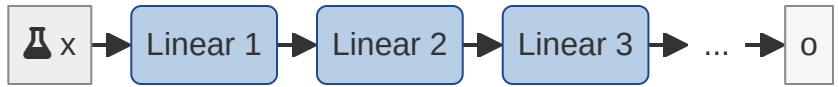
- Gradient vanishes: $w^{n-i} \to 0$ for large n

Case 2: $w = 1$

- Gradient stable: $\nabla_{w_i} y = a_{i-1}$

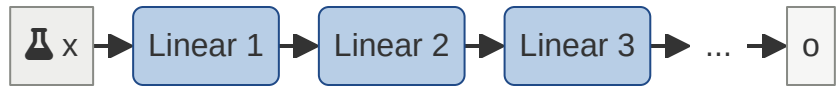
Case 3: $w > 1$

- Gradient explodes: $w^{n-i-1} \to \infty$ for large n
- In practice: exploding activations $\to$ `NaN`



Layer	Gradient $\nabla_{w_i} y = a_{i-1} \underbrace{\prod_{k=i+1}^{n} w^k}_{\approx w^{n-i}}$
$l = 1$	$\nabla_{w_1} y \approx x w^{n-1}$
$l = 2$	$\nabla_{w_2} y \approx a_1 w^{n-2}$
$l = k$	$\nabla_{w_k} y \approx a_{k-1} w^{n-k}$
$l = n$	$\nabla_{w_n} y \approx a_{n-1}$

Simple Example - Summary



$w < 1$

- Training stable
- Vanishing activations
- Vanishing gradients
- Network does not train

$w = 1$

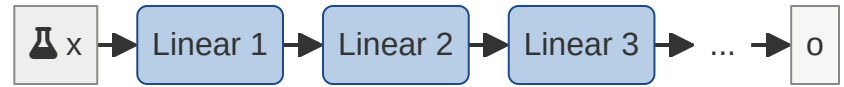
- Training stable
- Network does train
- Nearly impossible to maintain

$w > 1$

- Training explodes
- Exploding activations
- Exploding gradients
- Network does not train

General Linear Networks - Exploding Gradients

Weights: $W_i \in \mathbb{R}^{n \times n}$



Exploding activations

$$\|a_k\| = \|\prod_{i=1}^{k} W_k x\| \approx \|\prod_{i=1}^{k} W_k\| \|x\| \to \infty$$

Exploding gradients

$$\|\nabla_{W_k} y\| \approx \|a_{k-1}\| \| \prod_{i=k+1}^{n} W_i \| \|y\| \to \infty$$

- Network poorly initialized $\|W_i\| > 1$
- Learning rate too large
- Historically in recurrent networks

Handling Exploding Gradients

Symptoms

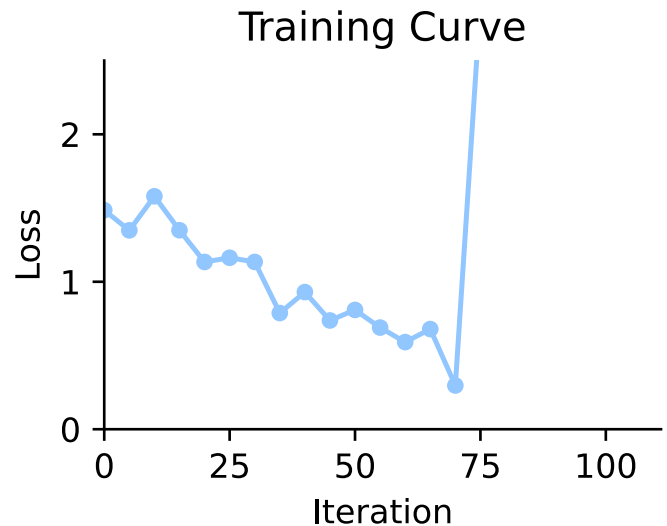
- Weights or loss are ∞ or `NaN`

Diagnosis

- Plot weight norms per layer
- Plot gradient norms per layer

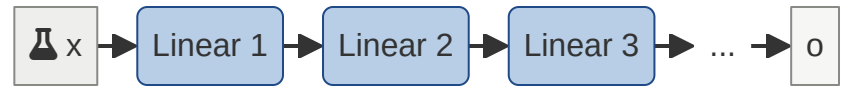
Remedy

- Reduce learning rate
- *Rarely:* change initialization



General Linear Networks - Vanishing Gradients

Weights: $W_i \in \mathbb{R}^{n \times n}$



Vanishing inputs

$$\|a_k\| = \|\prod_{i=1}^{k} W_k x\| \leq \prod_{i=1}^{k} \|W_k\| \|x\| \to 0$$

Vanishing gradients

$$\|\nabla_{W_k} y\| \leq \|a_{k-1}\| \prod_{i=k+1}^{n} \|W_i\| \|y\| \to \infty$$

- Occurs in almost all deep networks

Handling Vanishing Gradients

Symptoms

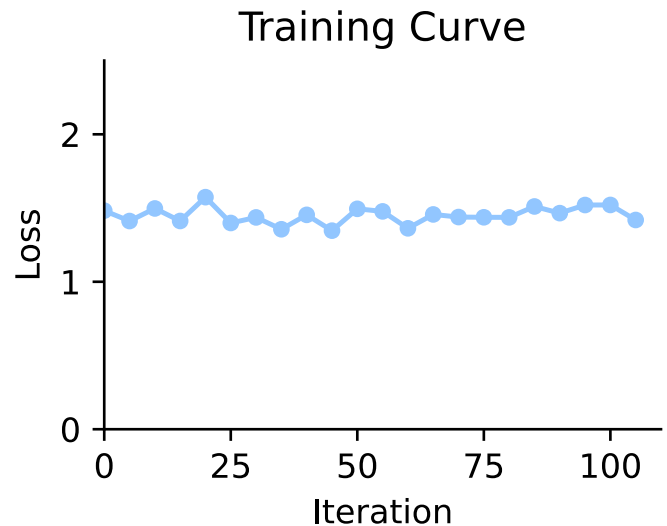
- Network does not train

Diagnosis

- Train with 0 learning rate and compare
- Plot weight norms per layer
- Plot gradient norms per layer

Remedy

- Happens to all but the shallowest networks
- Tune learning rate
- Change network architecture



Vanishing and Exploding Gradients - TL;DR

Vanishing gradients occur in most deep networks

Exploding gradients lead to `NaN`; fixed by lower learning rate