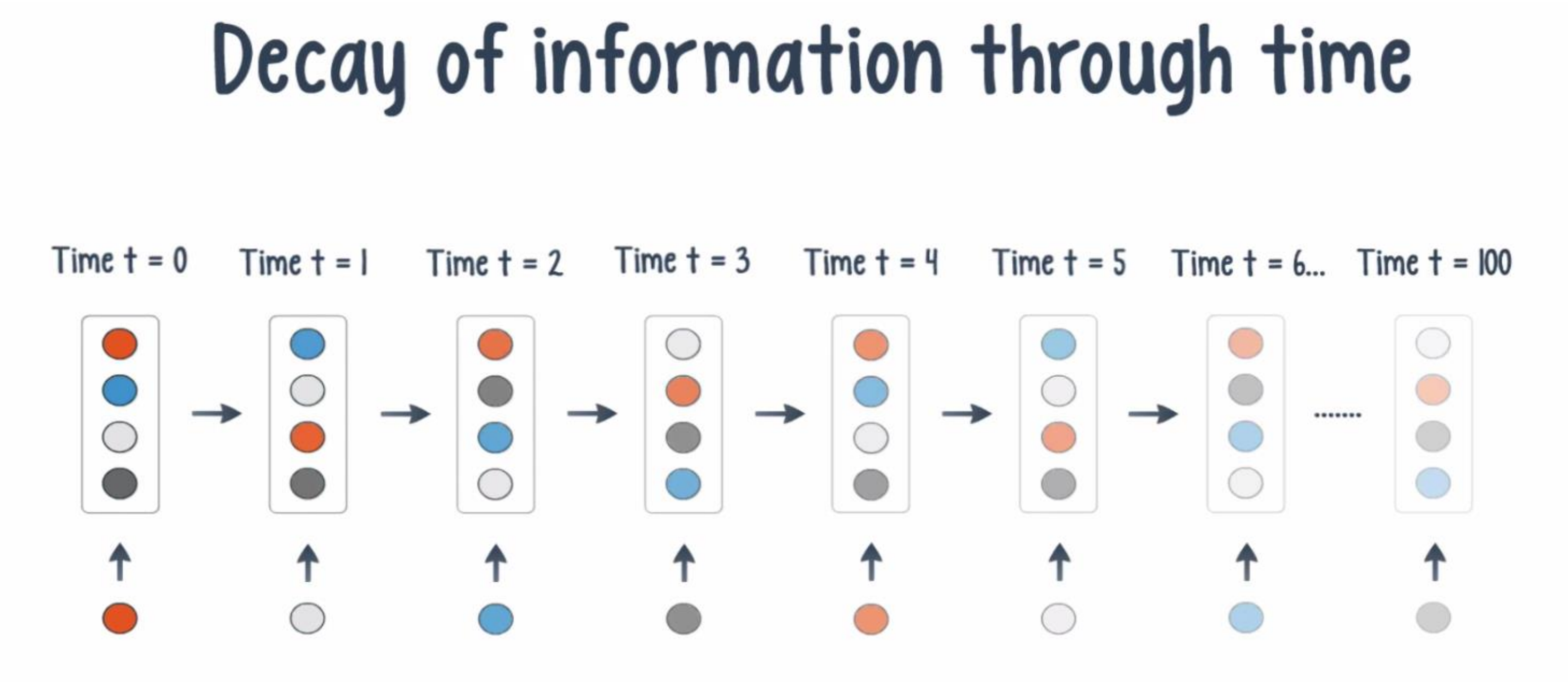
Vanishing & exploding gradients



Going deeper

- Very deep networks stop learning after a bit
  - An accuracy is reached, then the network saturates and starts unlearning
- Signal gets lost through so many layers
- Models start failing

Gradients behavior

- Modular learning → consistent behavior per module
- Let's check the backpropagation gradients

$$\frac{\partial \mathcal{L}}{\partial w_l} = \frac{\partial \mathcal{L}}{\partial a_L} \cdot \frac{\partial a_L}{\partial a_{L-1}} \cdot \frac{\partial a_{L-1}}{\partial a_{L-2}} \cdot \ \dots \cdot \frac{\partial a_l}{\partial w_l}$$

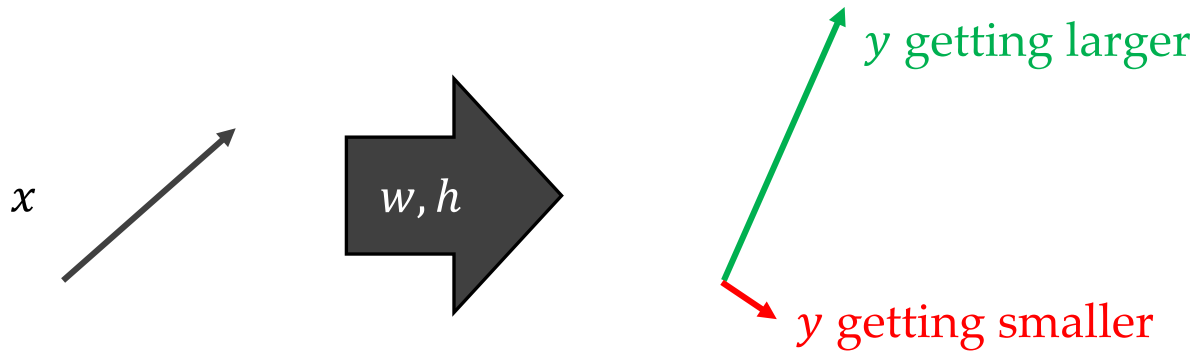
- The gradient depends on a product of L Jacobian matrices/tensors

$$\prod_{j=l+1}^{L} \frac{\partial a_j}{\partial a_{j-1}}$$

- What is the relation between gradient norm and depth L?

Spectral (matrix) norm of Jacobian

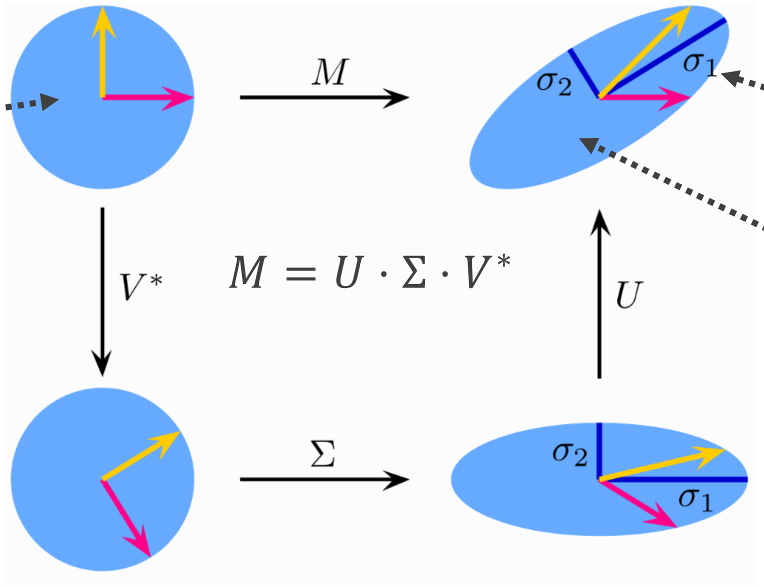
- After the module, does our input vector get larger in magnitude or smaller?
- To check this, we should check the matrix (spectral) norm
- How to compute spectral norm?



Singular values

- The spectral norm is the largest of the singular values
  - Computed with singular value decomposition (SVD)
- Singular values $\Leftrightarrow$ square roots of matrix eigenvalues
- E.g., a matrix operator M that transforms a unit sphere to an ellipsoid

Input x is a unit sphere in this example



The largest singular value σ_1 gives us an indication of how much larger the input has become

Ouput y after transforming our unit sphere by M

What is the spectral norm of our module?

- For simplicity, assuming each module to be
  - a linear operator $\boldsymbol{w}: \boldsymbol{x} \rightarrow \boldsymbol{y}$ (our linear transformation module)
  - followed by a nonlinearity h

$$\boldsymbol{a}_j = h(\boldsymbol{w}_j \cdot \boldsymbol{a}_{j-1})$$

- The spectral norm of the Jacobian is bounded by
  - the spectral norm of the linear operator
  - multiplied by the spectral norm of the nonlinear operator gradient h'

$$\left\| \frac{\partial \boldsymbol{a}_j}{\partial \boldsymbol{a}_{j-1}} \right\| \leq \left\| \boldsymbol{w}_j^T \right\| \cdot \left\| \mathrm{diag}\left(h'\left(\boldsymbol{a}_{j-1}\right)\right) \right\|$$

  - assuming an element-wise nonlinearity (non-diagonal entries are 0)

Combining per module spectral norms

- Our final spectral norm is bounded by

$$\left\|\frac{\partial\mathcal{L}}{\partial w_l}\right\| \propto \left\|\prod_{j=l+1}^{L}\frac{\partial a_j}{\partial a_{j-1}}\right\| \leq \prod_{j=l+1}^{L}\left\|\boldsymbol{w}_j^T\right\| \prod_{j=l+1}^{L}\left\|\operatorname{diag}\left(h'(\boldsymbol{a}_j)\right)\right\|$$

$$= \prod_{j=l}^{L}\sigma_j^{a}\cdot\sigma_j^{h'}$$

Where σ_j is the maximum singular value for module j

Vanishing and exploding gradients

- As depth L becomes larger

$$\left\|\frac{\partial \mathcal{L}}{\partial w_l}\right\| \leq \prod_{j=l}^{L} \sigma_j^a \cdot \sigma_j^{h'}$$

- For singular values $\sigma_j < 1$ we ***could*** obtain very small, vanishing gradients
  - E.g., $\sigma_j = 0.5$ and 10 layers we would have a norm of $9.7 \cdot 10^{-5}$
  - Very small gradients, learning is slowed down significantly
- For singular values $\sigma_j > 1$ we ***could*** obtain ever-growing, exploding gradients
  - E.g., $\sigma_j = 1.5$ and 10 layers we would have a norm of $4.06 \cdot 10^{17}$
  - Unstable optimization, oscillations, divergence

Pascanu, Mikolov, Bengio, On the difficulty of training recurrent neural networks, JMLR 2013

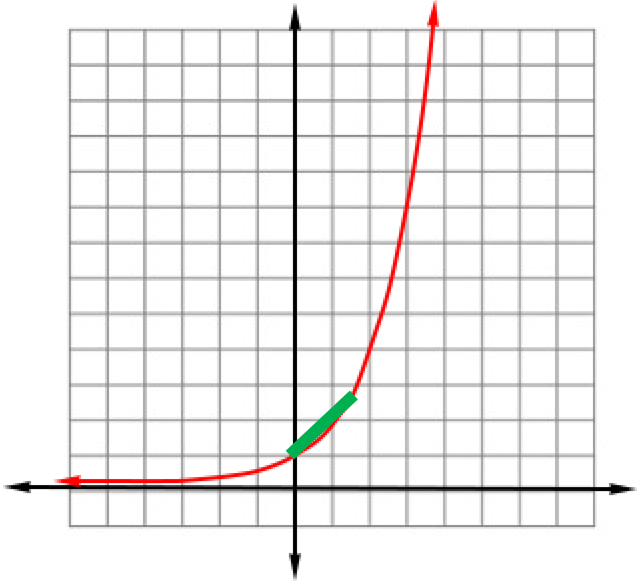
Later layers favored more

- As depth L becomes larger

$$\left\|\frac{\partial \mathcal{L}}{\partial w_l}\right\| \leq \prod_{j=l}^{L} \sigma_j^a \cdot \sigma_j^{h'}$$

- Effects exponential to layer depth
- Layers closer to the loss
  - less multiplications → less exponentiation (a bit linear) → little effect
- Layers further way from the loss
  - more multiplications → good ol' exponential growth → dramatic effects

Exponential growth



$y = ab^x \quad a > 0 \quad b > 0$