Part 4: Deep Neural Networks

•••

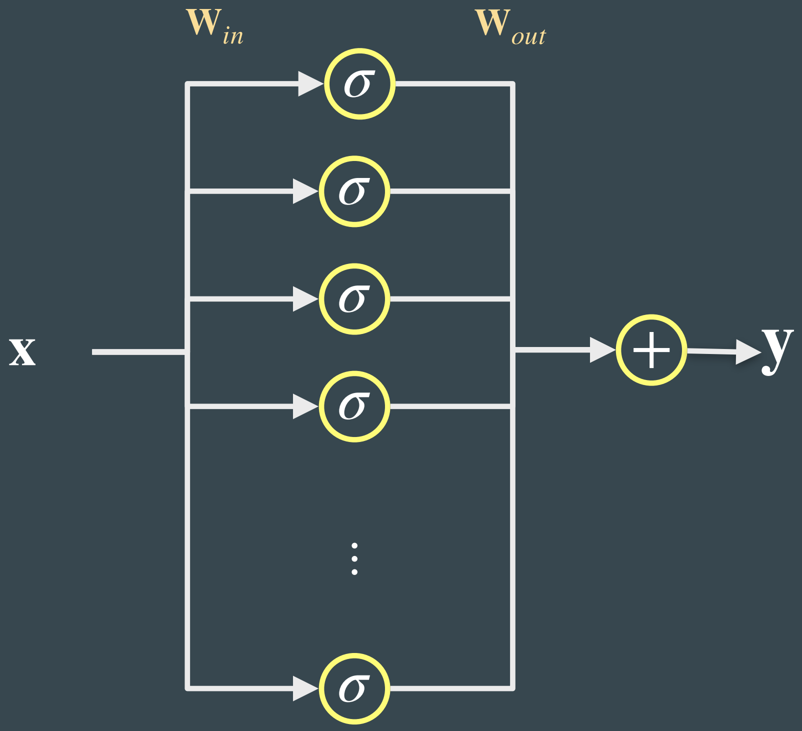
Mikhail Romanov

Gradient Vanishing

Two Layer Neural Net

$$y = w_1'\sigma(w_1 x + b_1) + b_1'$$
$$+ w_2'\sigma(w_2 x + b_2) + b_2'$$
$$+ w_3'\sigma(w_3 x + b_3) + b_3'$$

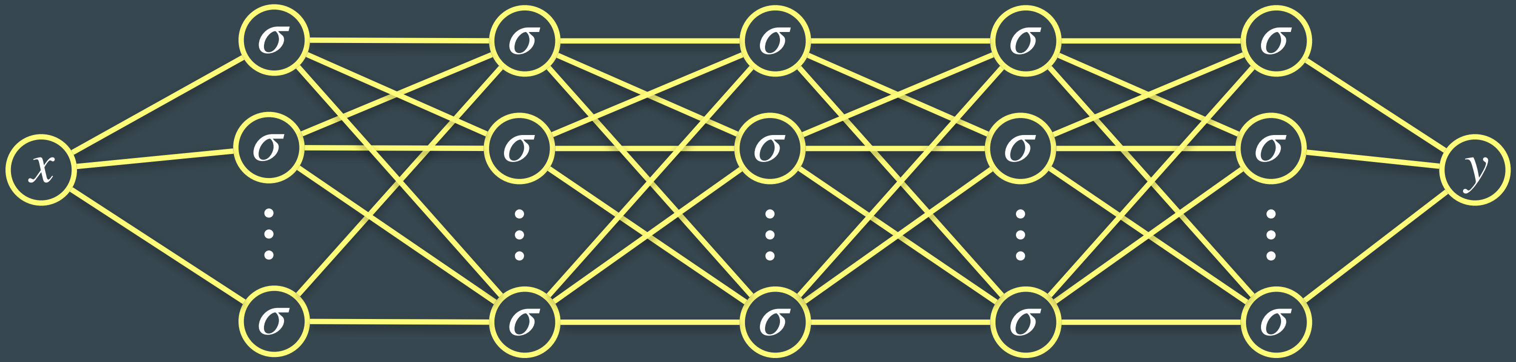
$$\hat{y} = b^{out} + \sum_{i=1}^{N} w_i^{out}\sigma(w^i x + b^i)$$



$$\hat{\mathbf{y}} = \mathbf{b}_{out} + \sum_{i=1}^{N} \mathbf{W}_{out}\sigma(\mathbf{W}_{\mathbf{in}}\mathbf{X} + \mathbf{b}_{in})$$

Linear Combination of Sigmoids is Full System!

Multi-Layer NNs



x

$\text{Linear } 1 \rightarrow N_1$

Sigmoid

$\text{Linear } N_1 \rightarrow N_2$

Sigmoid

$\text{Linear } N_2 \rightarrow N_3$

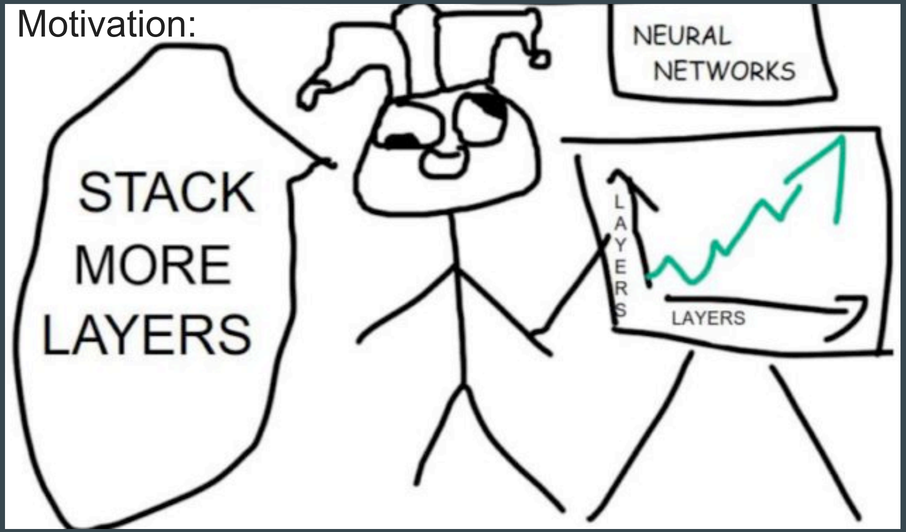
Sigmoid

$\vdots$

Sigmoid

$\text{Linear } N_{M-1} \rightarrow N_M$

y



Gradient Vanishing: Sigmoid

x

$\text{Linear } 1 \rightarrow \text{N}_1$

Sigmoid

$\text{Linear N}_1 \rightarrow \text{N}_2$

Sigmoid

$\text{Linear N}_2 \rightarrow \text{N}_3$

Sigmoid

$\vdots$

Sigmoid

$\text{Linear N}_{\text{M}-1} \rightarrow \text{N}_\text{M}$

y

$$y = f_1(f_2(f_3(\ldots)))$$

$$\frac{\partial y}{\partial x} = \frac{\partial f_N}{\partial f_{N-1}} \frac{\partial f_{N-1}}{\partial f_{N-2}} \frac{\partial f_{N-2}}{\partial f_{N-3}} \ldots \frac{\partial f_2}{\partial f_1} \frac{\partial f_1}{\partial x}$$

σ

$$\frac{\partial y}{\partial x} = \prod_{i=1}^{N} \frac{\partial \text{Linear}_\text{i}}{\partial z_i} \prod_{j=1}^{M} \frac{\partial \sigma_j}{\partial z_j} = V \prod_{j=1}^{M} \sigma(z_j)(1 - \sigma(z_j)) \leq V \frac{1}{4^M}$$

Serious risk of plateau

Tanh

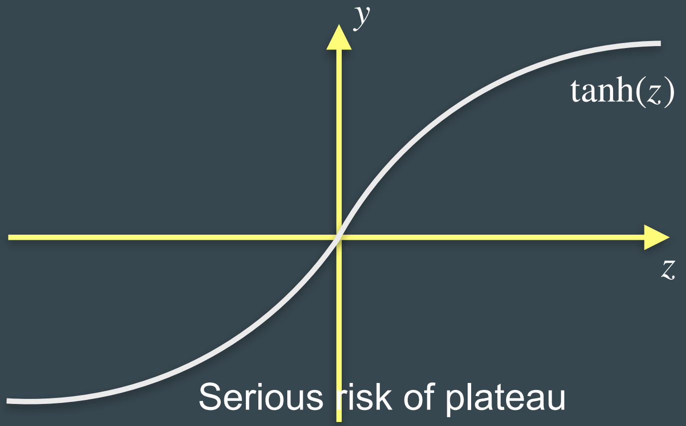
x
Linear $1 \to \mathrm{N}_1$
Tanh
Linear $\mathrm{N}_1 \to \mathrm{N}_2$
Tanh
Linear $\mathrm{N}_2 \to \mathrm{N}_3$
Tanh
$\vdots$
Tanh
Linear $\mathrm{N}_{\mathrm{M}-1} \to \mathrm{N}_\mathrm{M}$
y

$$y = f_1(f_2(f_3(\ldots)))$$

$$\frac{\partial y}{\partial x} = \frac{\partial f_N}{\partial f_{N-1}} \frac{\partial f_{N-1}}{\partial f_{N-2}} \frac{\partial f_{N-2}}{\partial f_{N-3}} \cdots \frac{\partial f_2}{\partial f_1} \frac{\partial f_1}{\partial x}$$

$$\tilde{\sigma} = \tanh$$

$$\frac{\partial y}{\partial x} = \prod_{i=1}^{N} \frac{\partial \mathrm{Linear_i}}{\partial z_i} \prod_{j=1}^{M} \frac{\partial \tilde{\sigma}_j}{\partial z_j} = V \prod_{j=1}^{M} (1 + \tilde{\sigma}(z_j))(1 - \tilde{\sigma}(z_j)) \leq V 1^M$$



Leaky ReLU

$$y = f_1(f_2(f_3(\ldots)))$$

$$\frac{\partial y}{\partial x} = \frac{\partial f_N}{\partial f_{N-1}} \frac{\partial f_{N-1}}{\partial f_{N-2}} \frac{\partial f_{N-2}}{\partial f_{N-3}} \cdots \frac{\partial f_2}{\partial f_1} \frac{\partial f_1}{\partial x}$$

x

$\mathrm{Linear}\ 1 \to \mathrm{N}_1$

$\mathrm{L-ReLU}$

$\mathrm{Linear}\ \mathrm{N}_1 \to \mathrm{N}_2$

$\mathrm{L-ReLU}$

$\mathrm{Linear}\ \mathrm{N}_2 \to \mathrm{N}_3$

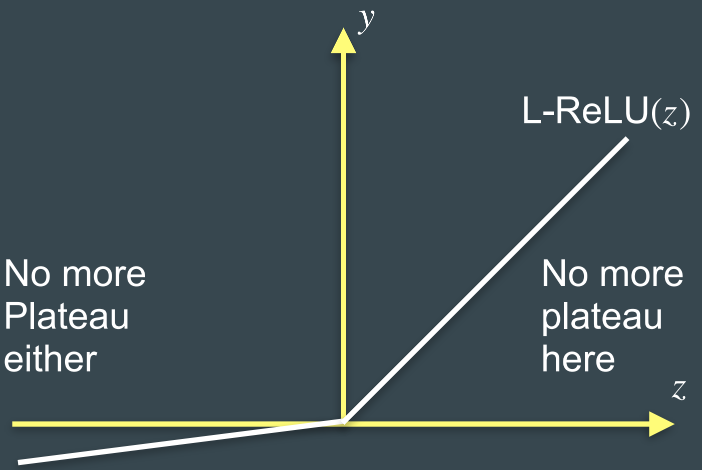
$\mathrm{L-ReLU}$

$\vdots$

$\mathrm{L-ReLU}$

$\mathrm{Linear}\ \mathrm{N}_{\mathrm{M}-1} \to \mathrm{N}_\mathrm{M}$

y

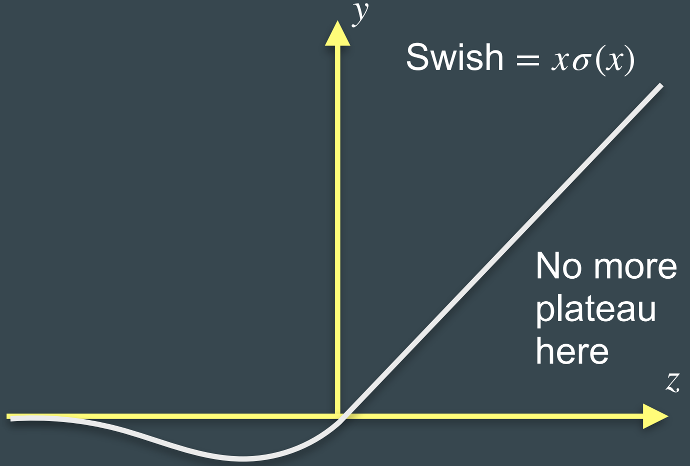


Leaky ReLU

$$y = f_1(f_2(f_3(\ldots)))$$

$$\frac{\partial y}{\partial x} = \frac{\partial f_N}{\partial f_{N-1}} \frac{\partial f_{N-1}}{\partial f_{N-2}} \frac{\partial f_{N-2}}{\partial f_{N-3}} \cdots \frac{\partial f_2}{\partial f_1} \frac{\partial f_1}{\partial x}$$

x
Linear $1 \rightarrow \mathrm{N}_1$
$\mathrm{L-ReLU}$
Linear $\mathrm{N}_1 \rightarrow \mathrm{N}_2$
$\mathrm{L-ReLU}$
Linear $\mathrm{N}_2 \rightarrow \mathrm{N}_3$
$\mathrm{L-ReLU}$
$\vdots$
$\mathrm{L-ReLU}$
Linear $\mathrm{N}_{\mathrm{M}-1} \rightarrow \mathrm{N}_{\mathrm{M}}$
y



Linear Layer

$$y = f_1(f_2(f_3(\ldots)))$$

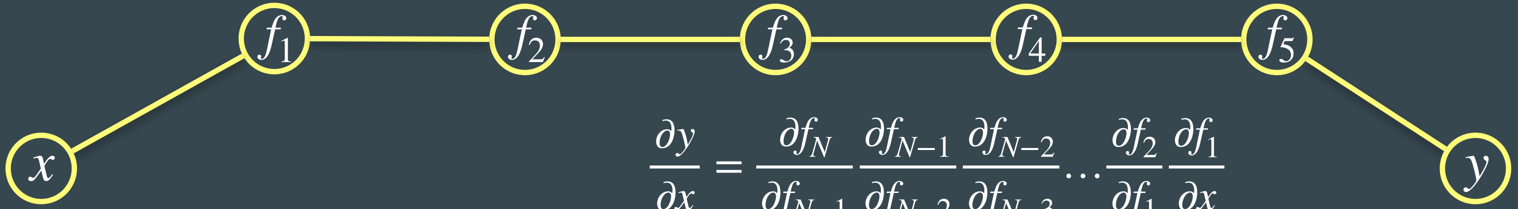
$$\frac{\partial y}{\partial x} = \frac{\partial f_N}{\partial f_{N-1}} \frac{\partial f_{N-1}}{\partial f_{N-2}} \frac{\partial f_{N-2}}{\partial f_{N-3}} \ldots \frac{\partial f_2}{\partial f_1} \frac{\partial f_1}{\partial x}$$

$$z = ax + b$$

$$\partial_x z = a$$

x
$\text{Linear } 1 \to \text{N}_1$
$\text{L} - \text{ReLU}$
$\text{Linear } \text{N}_1 \to \text{N}_2$
$\text{L} - \text{ReLU}$
$\text{Linear } \text{N}_2 \to \text{N}_3$
$\text{L} - \text{ReLU}$
$\vdots$
$\text{L} - \text{ReLU}$
$\text{Linear } \text{N}_{\text{M}-1} \to \text{N}_\text{M}$
y

What about linear operations?



$$\frac{\partial y}{\partial x} = \frac{\partial f_N}{\partial f_{N-1}} \frac{\partial f_{N-1}}{\partial f_{N-2}} \frac{\partial f_{N-2}}{\partial f_{N-3}} \dots \frac{\partial f_2}{\partial f_1} \frac{\partial f_1}{\partial x}$$

x
Linear $1 \rightarrow \mathrm{N}_1$
Sigmoid
Linear $\mathrm{N}_1 \rightarrow \mathrm{N}_2$
Sigmoid
Linear $\mathrm{N}_2 \rightarrow \mathrm{N}_3$
Sigmoid
$\vdots$
Sigmoid
Linear $\mathrm{N}_{\mathrm{M}-1} \rightarrow \mathrm{N}_\mathrm{M}$
y

$$y = f_1(f_2(f_3(\dots)))$$

$$z = wx + b$$

$$\frac{\partial L}{\partial x} = w \frac{\partial L}{\partial z}$$

$$\left|\frac{\partial L}{\partial x}\right| = w \left|\frac{\partial L}{\partial z}\right|$$

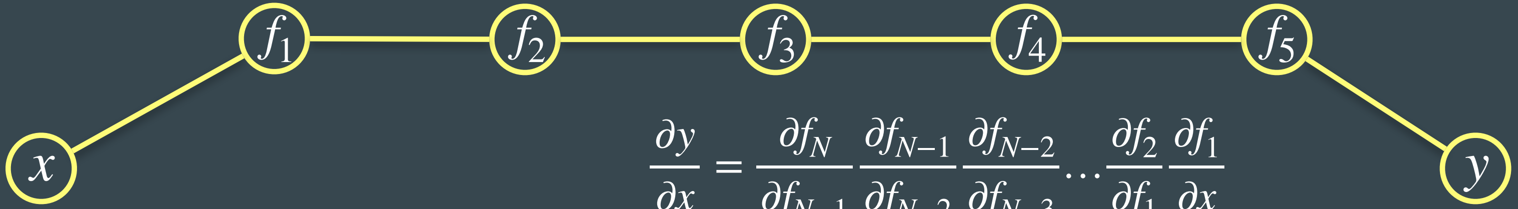
$$\mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}$$

$$\frac{\partial L}{\partial \mathbf{x}} = \mathbf{W}^T \frac{\partial L}{\partial \mathbf{z}}$$

$$\sigma_{min} \left|\frac{\partial L}{\partial z}\right| \leq \left|\frac{\partial L}{\partial x}\right| \leq \sigma_{max} \left|\frac{\partial L}{\partial z}\right|$$

One cannot avoid gradient Vanishing!
AT ALL!

Can we avoid gradient vanishing?



$$\frac{\partial y}{\partial x} = \frac{\partial f_N}{\partial f_{N-1}} \frac{\partial f_{N-1}}{\partial f_{N-2}} \frac{\partial f_{N-2}}{\partial f_{N-3}} \dots \frac{\partial f_2}{\partial f_1} \frac{\partial f_1}{\partial x}$$

$$y = f_1(f_2(f_3(\dots)))$$

$$\left|\frac{\partial f}{\partial z}\right|$$

Non-linear activation

$$\left|\frac{\partial f}{\partial z}\right| \neq const$$

$$\left|\frac{\partial f}{\partial z}\right| < 1 \qquad \left|\frac{\partial f}{\partial z}\right| = 1 \qquad \left|\frac{\partial f}{\partial z}\right| > 1$$

Linear activation

$$\left|\frac{\partial f}{\partial z}\right| = const$$

One cannot avoid gradient Vanishing!

x

Linear $1 \to \mathrm{N}_1$

Sigmoid

Linear $\mathrm{N}_1 \to \mathrm{N}_2$

Sigmoid

Linear $\mathrm{N}_2 \to \mathrm{N}_3$

Sigmoid

$\vdots$

Sigmoid

Linear $\mathrm{N}_{\mathrm{M}-1} \to \mathrm{N}_\mathrm{M}$

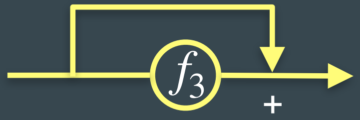
y

Residual Connections

Residual Connection

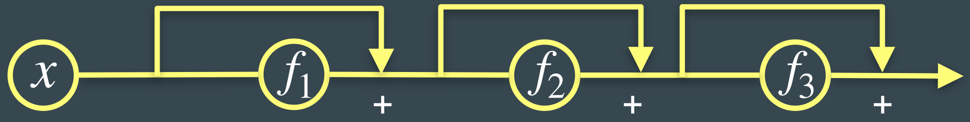
$$y = f_1(f_2(f_3(\ldots)))$$

$$\partial_x y = \frac{\partial f_N}{\partial z_{N-1}} \frac{\partial f_{N-1}}{\partial z_{N-2}} \frac{\partial f_{N-2}}{\partial z_{N-3}} \ldots \frac{\partial f_2}{\partial z_1} \frac{\partial f_1}{\partial x}$$



$$z = f(x) + x$$

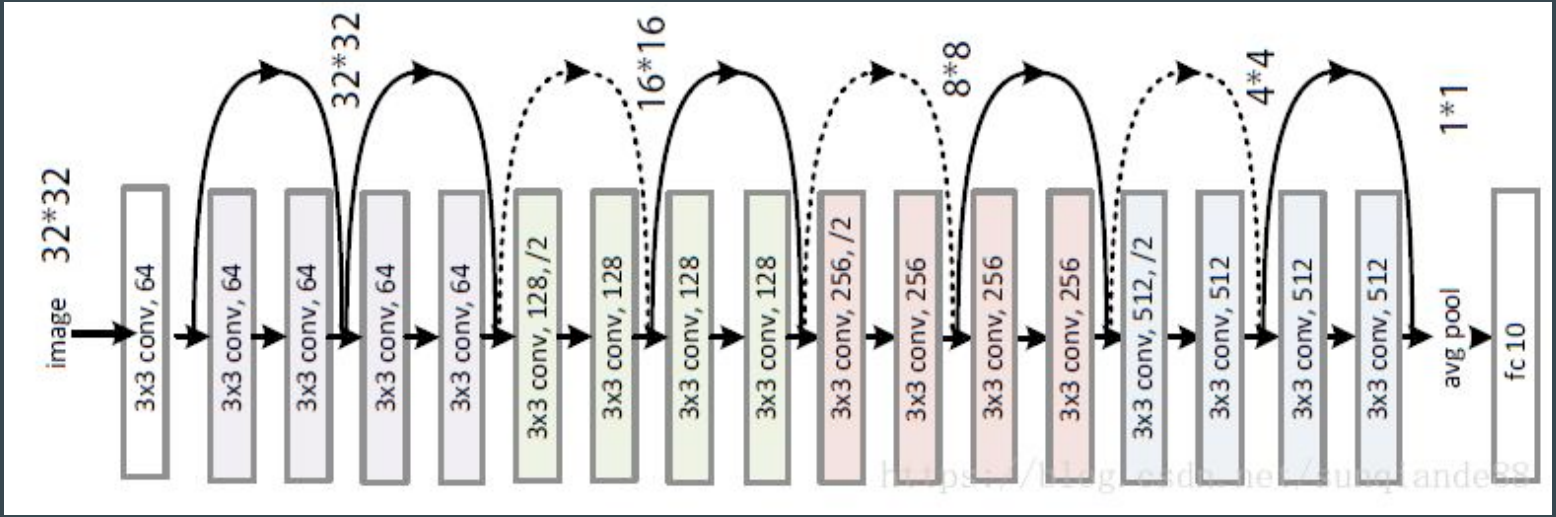
$$\partial_x z = \partial_x f(x) + 1$$



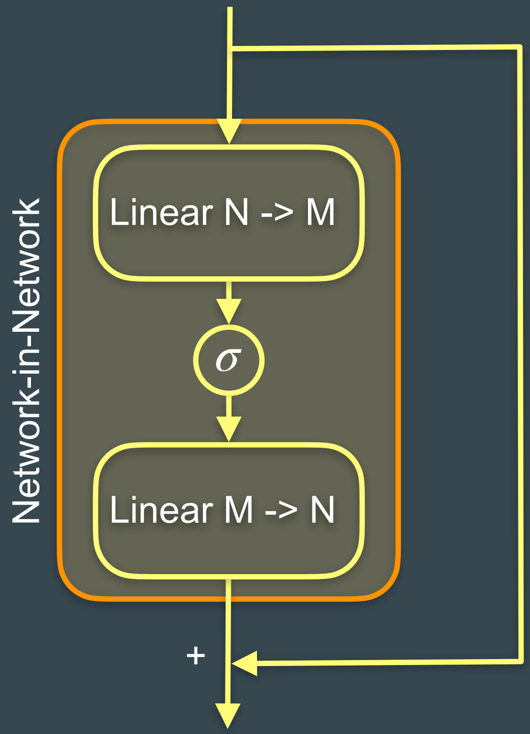
$$y = f_1(f_2(f_3(\ldots)))$$

$$\frac{\partial y}{\partial x} = \left(1 + \frac{\partial f_N}{\partial z_{N-1}}\right) \ldots \left(1 + \frac{\partial f_2}{\partial z_1}\right) \left(1 + \frac{\partial f_1}{\partial x}\right)$$

ResNet Allows: Extremely Deep Networks

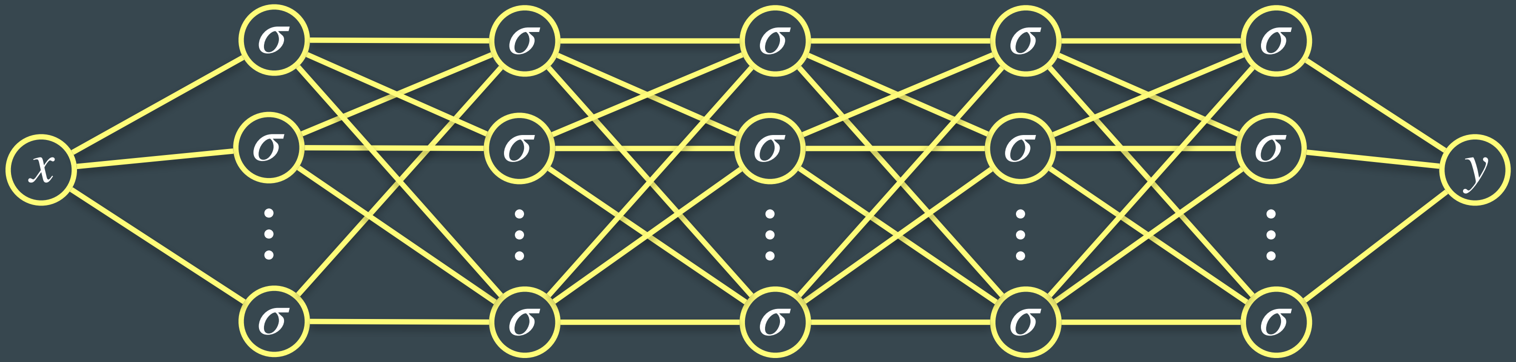


ResNet Allows: “Boosted” Neural Networks

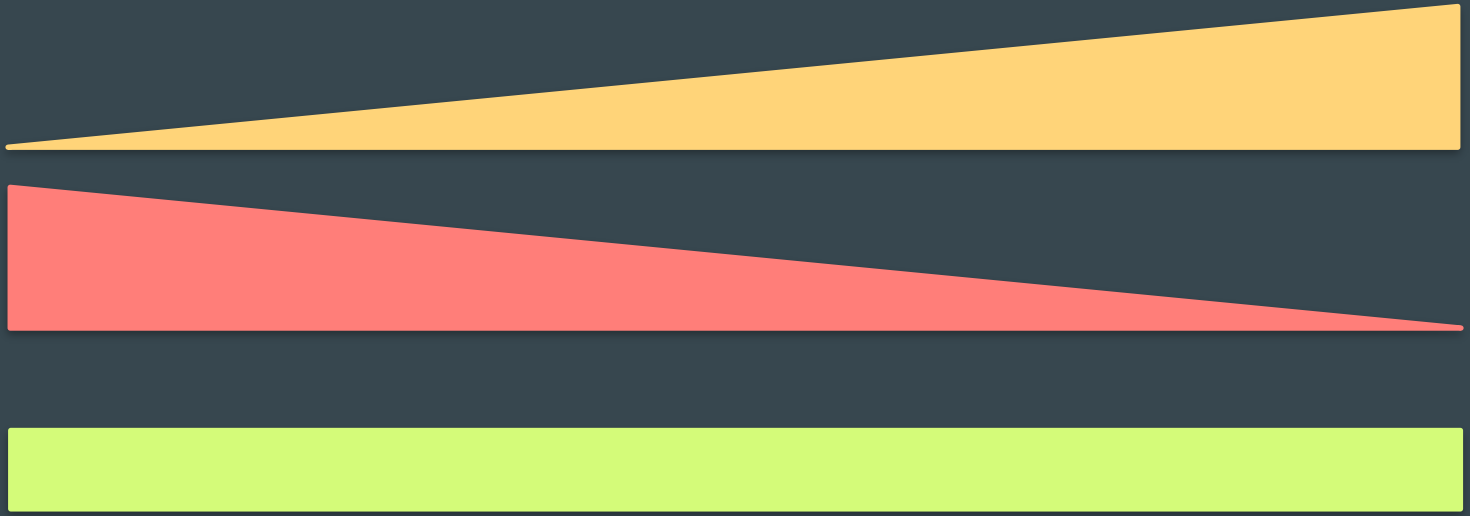


Normal signals

What if the signals are normal



Gradient amplitudes:

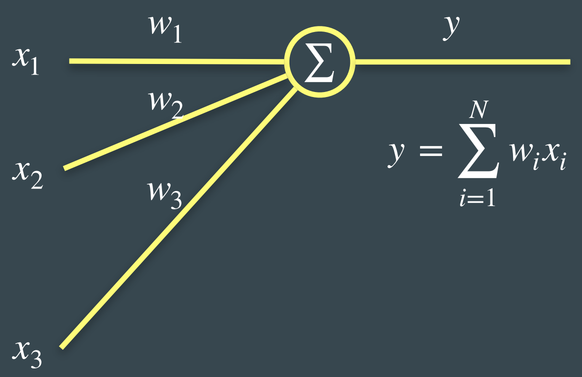


How to preserve normality

$x:$

$\mathbb{E}x = 0$

$\mathbb{V}x = 1$



$y \propto ?$

$$\mathbb{E}y = \mathbb{E}\left(\sum_{i=1}^{N} w_i x_i\right) = \sum_{i=1}^{N} w_i \mathbb{E}x_i = 0$$

$$\mathbb{V}y = \mathbb{V}\left(\sum_{i=1}^{N} w_i x_i\right) = \sum_{i=1}^{N} \mathbb{V}\left(w_i x_i\right) = \sum_{i=1}^{N} w_i^2 \mathbb{V}x = \sum_{i=1}^{N} w_i^2$$

Zero mean is preserved

How to keep unit variance?

$$\sum_{i=1}^{N} w_i^2 \approx 1$$

$w_i \propto \mathcal{N}\left(0, \frac{1}{\sqrt{N}}\right)$ He initialisation

$w_i \propto \mathcal{U}\left(-\frac{C}{\sqrt{N}}, \frac{C}{\sqrt{N}}\right)$ Xavier initialisation

How to enforce normality on an input

$$\mathbf{x}^* = \frac{\mathbf{x} - \mu}{\sigma}$$

$$\mu = \frac{1}{N}\sum_{s=1}^{S} \mathbf{x}_s$$

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{s=1}^{S} (\mathbf{x}_s - \mu)^2}$$

1) We set mean to zero

2) We set standard deviation to one

5-Sigma rule:

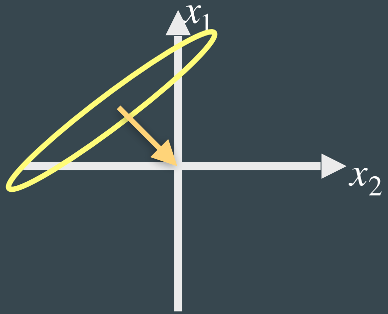
- Now the input signals are bound
- To the interval [-5, 5]
- And only 1 out of 1 million
- Leaves this interval

Now the inputs are perfectly OK!

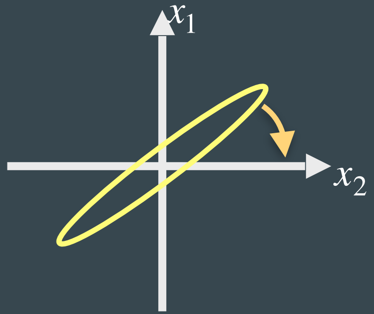
What the normality is

$$\mathbf{x}^* = \Sigma^{-1}(\mathbf{x} - \mu) \qquad \mu = \frac{1}{N}\sum_{s=1}^{S} \mathbf{x}_s \qquad \Sigma^T\Sigma = \frac{1}{N-1}\sum_{s=1}^{S} (\mathbf{x}_s - \mu)(\mathbf{x}_s - \mu)^T$$

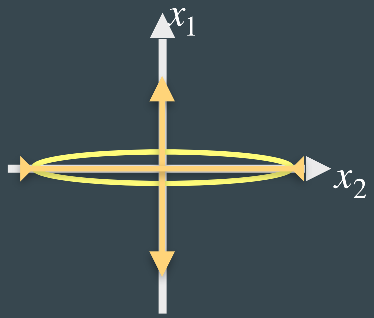
Centering



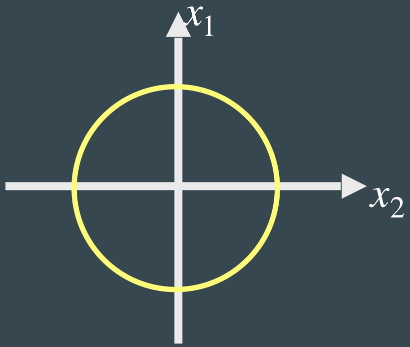
Decorrelation



Standardisation



Result



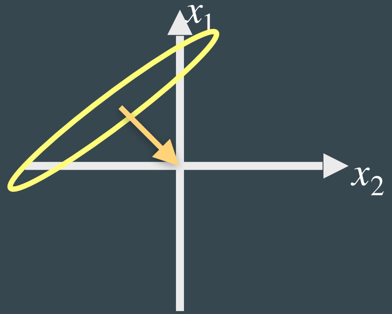
What each of the steps imply

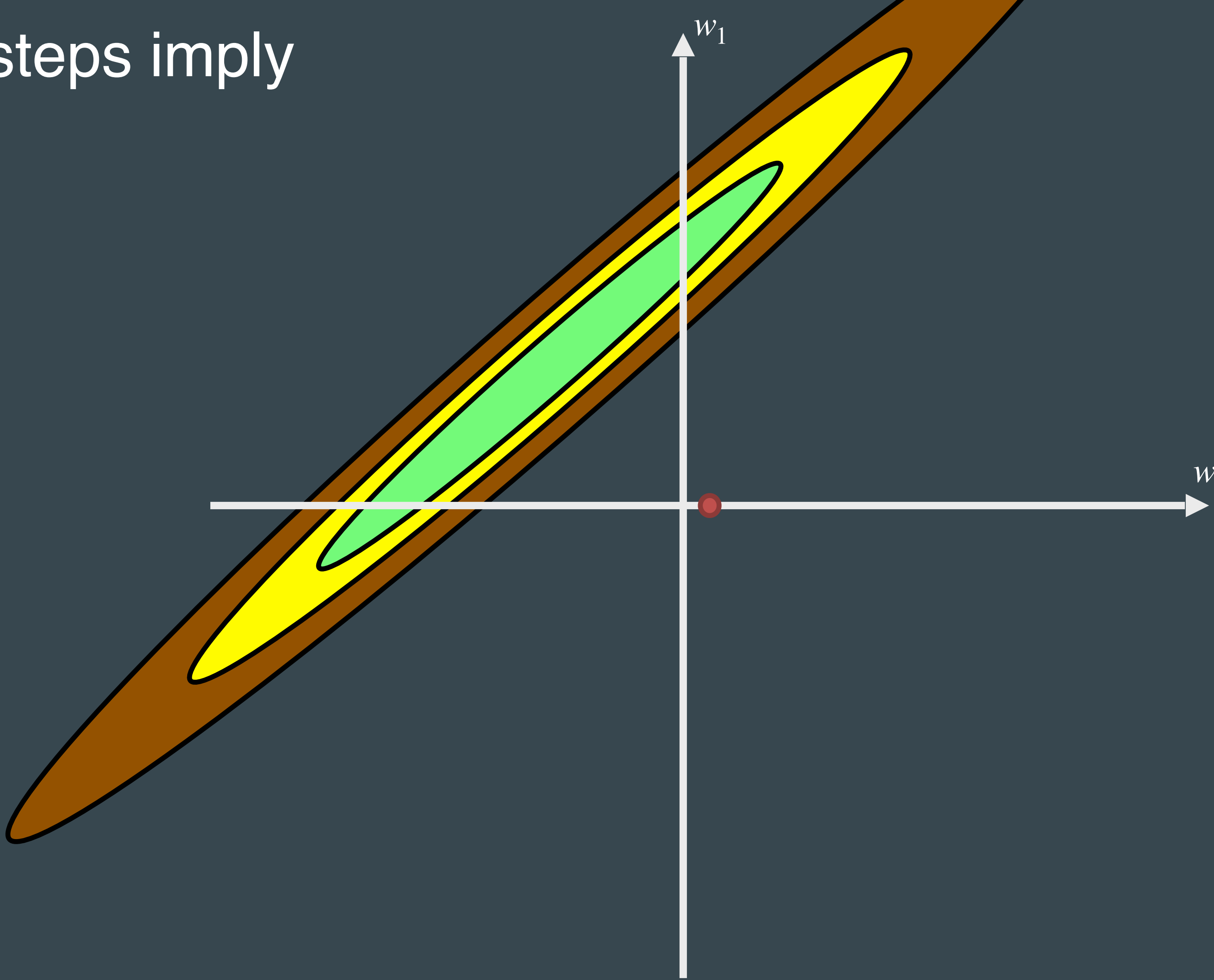
$$\mu = \frac{1}{N}\sum_{s=1}^{S}\mathbf{x}_s$$

$$\Sigma^T\Sigma = \frac{1}{N-1}\sum_{s=1}^{S}(\mathbf{x}_s-\mu)(\mathbf{x}_s-\mu)^T$$

$$\mathbf{x}^* = \Sigma^{-1}(\mathbf{x}-\mu)$$

Centering

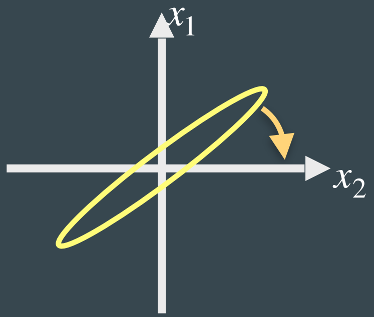


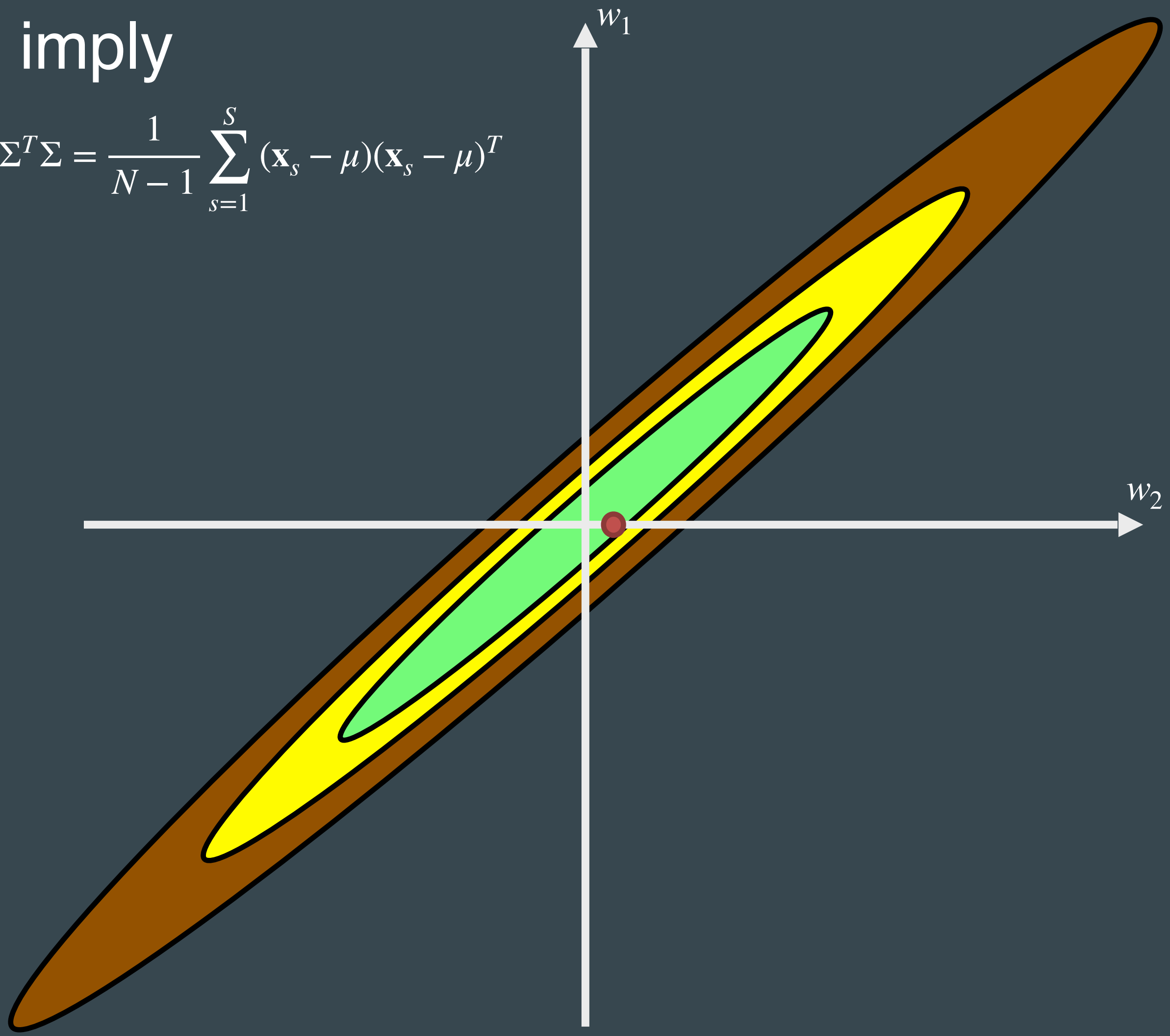


What each of the steps imply

$$\mathbf{x}^* = \Sigma^{-1}(\mathbf{x} - \mu) \qquad \mu = \frac{1}{N}\sum_{s=1}^{S} \mathbf{x}_s \qquad \Sigma^T\Sigma = \frac{1}{N-1}\sum_{s=1}^{S} (\mathbf{x}_s - \mu)(\mathbf{x}_s - \mu)^T$$

Decorrelation

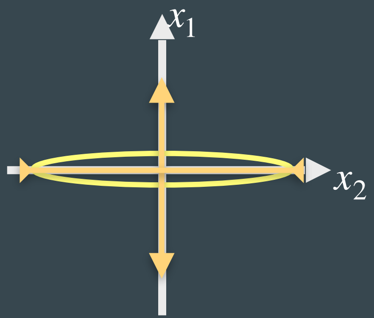


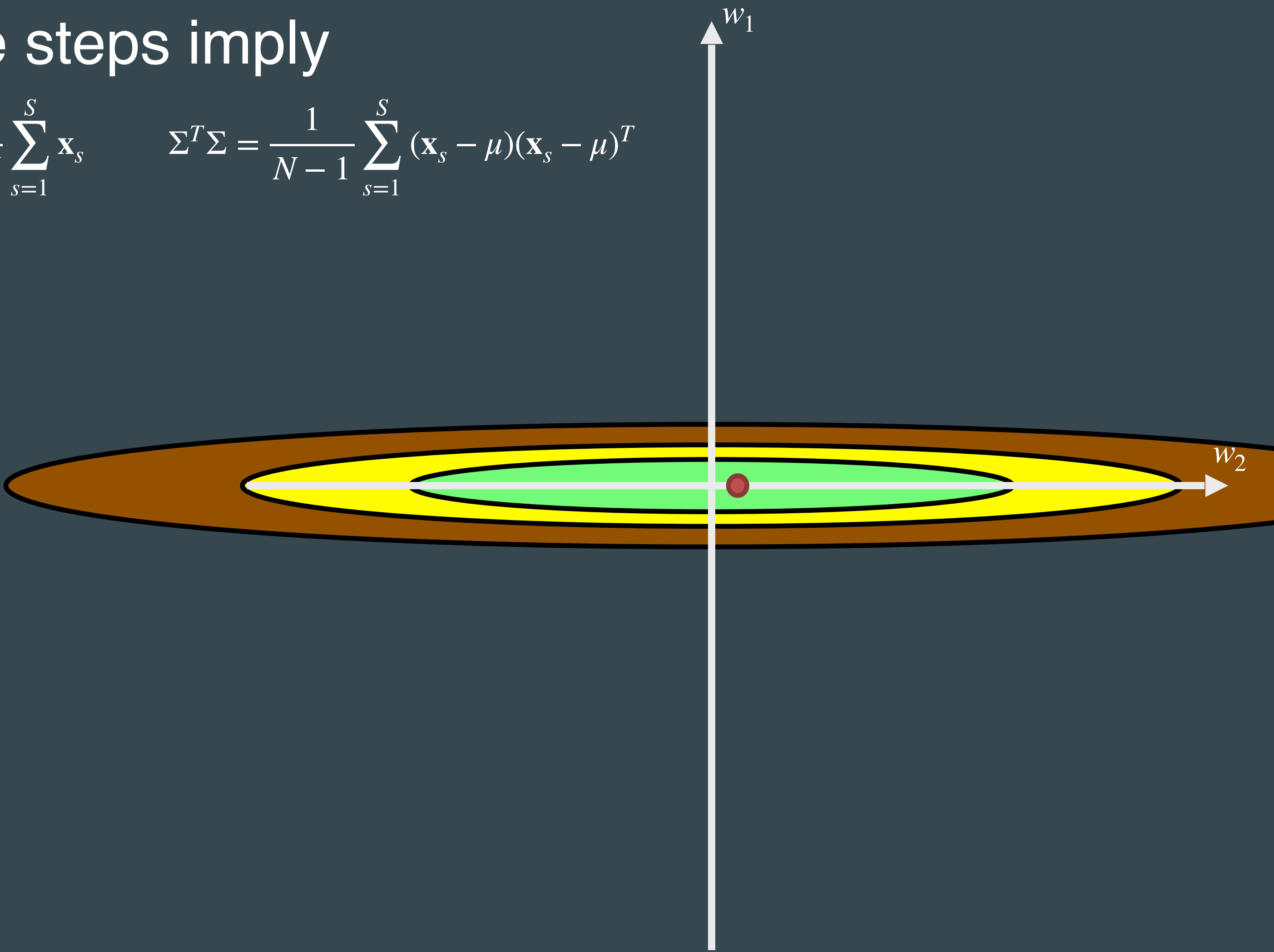


What each of the steps imply

$$\mathbf{x}^* = \Sigma^{-1}(\mathbf{x} - \mu) \qquad \mu = \frac{1}{N}\sum_{s=1}^{S}\mathbf{x}_s \qquad \Sigma^T\Sigma = \frac{1}{N-1}\sum_{s=1}^{S}(\mathbf{x}_s - \mu)(\mathbf{x}_s - \mu)^T$$

Standardisation





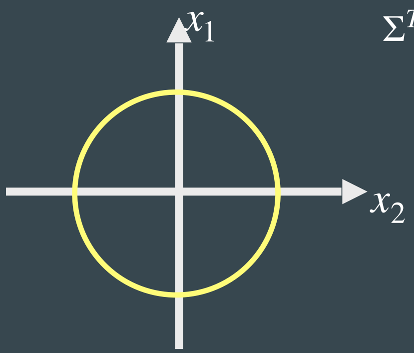
What each of the steps imply

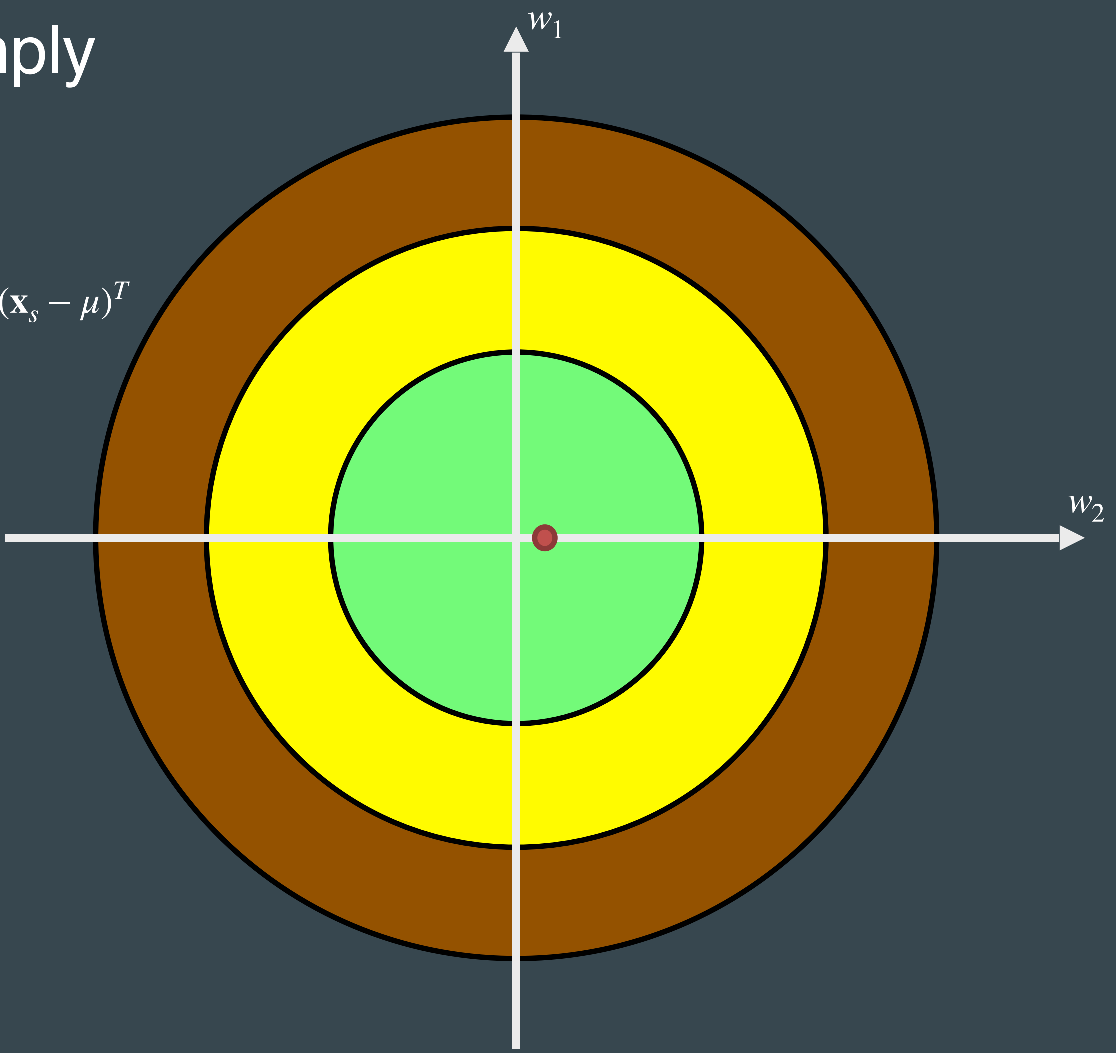
$$\mathbf{x}^* = \Sigma^{-1}(\mathbf{x} - \mu)$$

$$\mu = \frac{1}{N}\sum_{s=1}^{S} \mathbf{x}_s$$

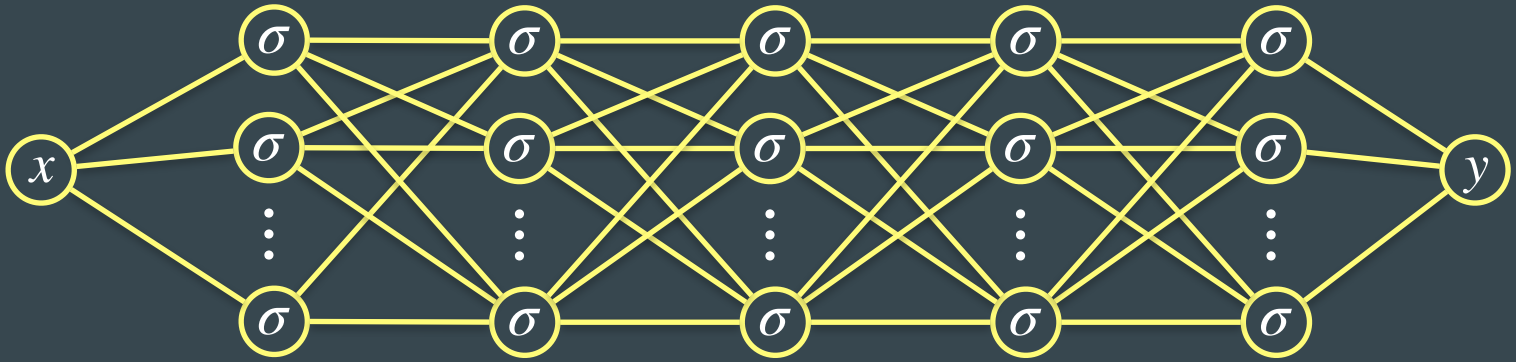
$$\Sigma^T\Sigma = \frac{1}{N-1}\sum_{s=1}^{S} (\mathbf{x}_s - \mu)(\mathbf{x}_s - \mu)^T$$

Result





How to enforce normality on signals



$$\mathbf{z}^* = \frac{\mathbf{z} - \mu}{\sigma}\mathbf{a} + \mathbf{b}$$

μ, σ	Statistical parameters	Computed on one batch
a, b	Trainable parameters	Optimised

Validation:

$$\hat{\mu} = EMA(\mu)$$

$$\hat{\sigma}^2 = EMA(\sigma^2)$$

z

Other reasons why that is useful

Regularisation

What the regularisation is

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

$$\|\mathbf{Ax} - \mathbf{b}\|_2^2 = 0$$

$$\mathbf{x} = \underset{\mathbf{x}}{\operatorname{argmax}} \|\mathbf{Ax} - \mathbf{b}\|_2^2$$

One solution if

$$\dim(\mathbf{x}) \leq \dim(\mathbf{b})$$

No issues here

Infinitely many solutions if

$$\dim(\mathbf{x}) > \dim(\mathbf{b})$$

We are interested only in the simpliest solution

How to measure the simplicity of a solution?

$\|\mathbf{x}\|$ — complexity measure

$$L = \|\mathbf{Ax} - \mathbf{b}\|_2^2$$

Does not take into account solution complexity

Many solutions

$$L^* = \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_2^2$$

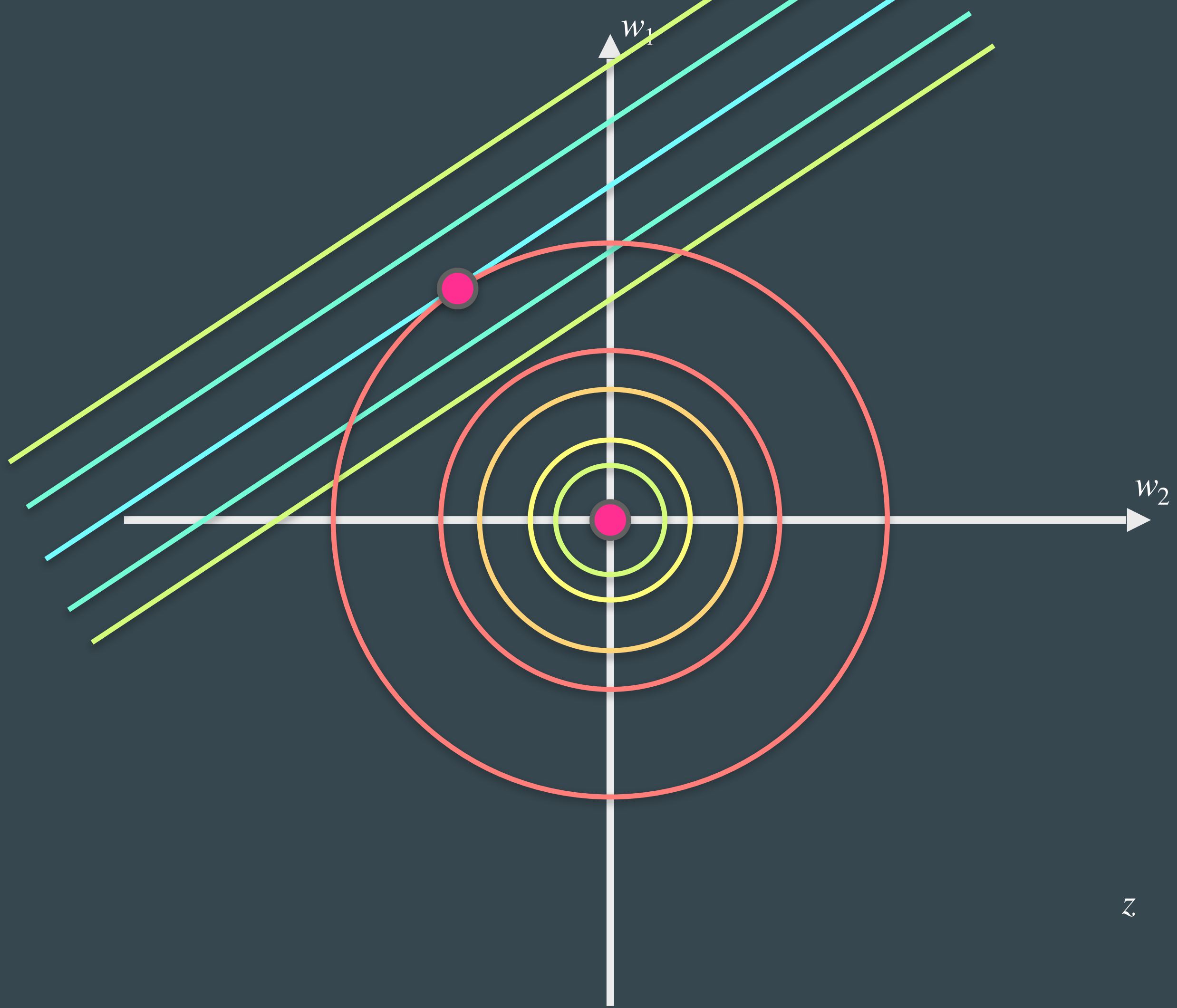
Takes into account solution complexity

But displaces the solution

Only one solution

L2 regularisation

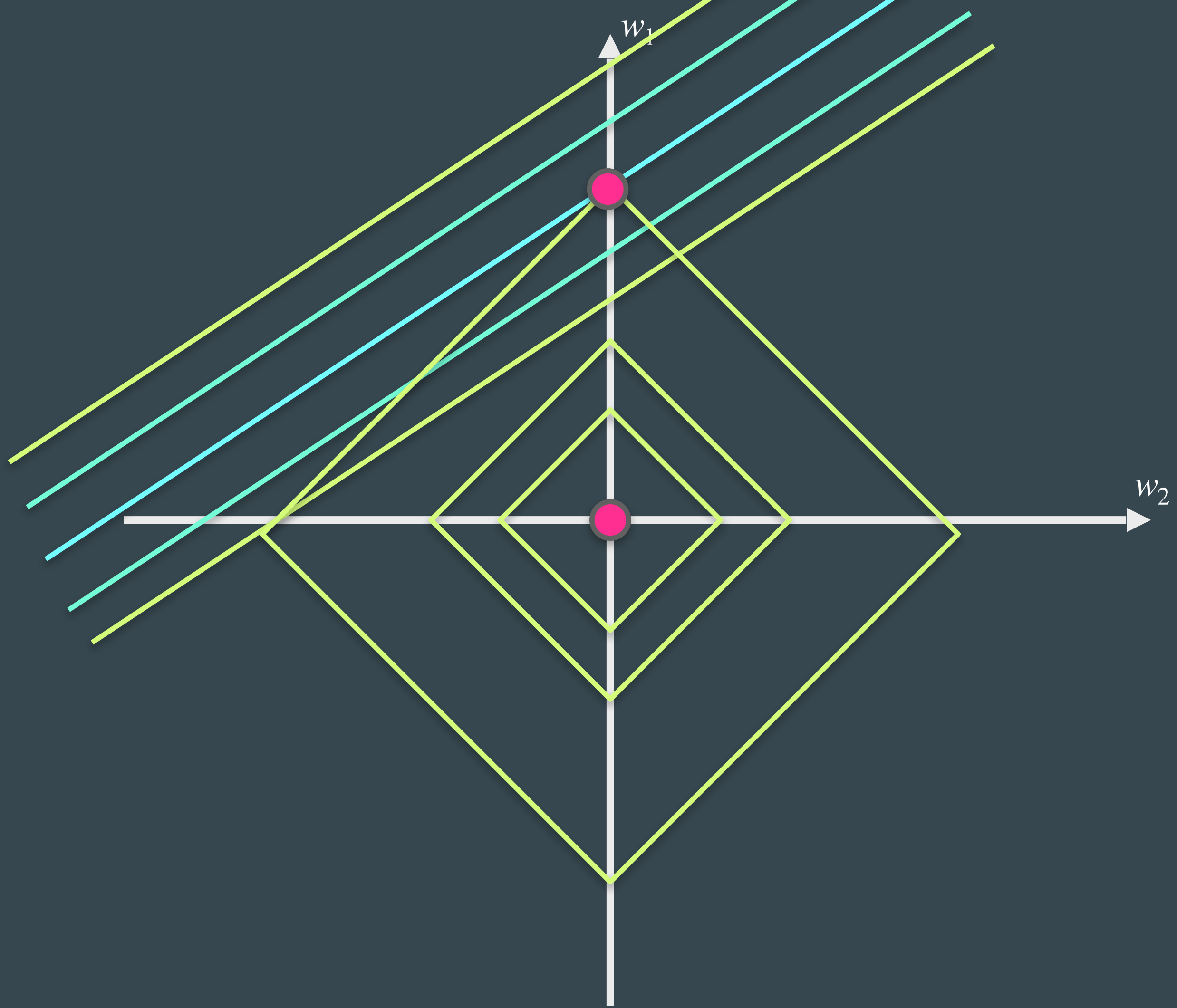
$$L^* = \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_2^2$$



L1 regularisation

$$L^* = \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_M$$

$$\|\mathbf{x}\|_M = \ x_1 \ + \ x_2 \ + \ldots + \ x_N$$



Summary

- Gradient Vanishing problem and its source
- Choice: Gradient Vanishing or Gradient Explosion
- Methods of mitigating Gradient Vanishing
- Batch Normalisation: enforcing the Normality on Neural Network’s signals
- Other positive sides of BatchNorms
- Regularisation and why is that good