ECMWF – DESTINATION EARTH

NEURAL NETWORKS AND DEEP LEARNING

Mario Santa Cruz Lopez

Funded by the European Union Destination Earth implemented by ECMWF esa EUMETSAT

OUTLINE

How to train you neural network?

- Neuron, layer and MLP
- Backpropagation
- Activation functions

Scaling Laws

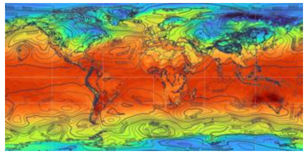
Earth System Modelling

WHAT ARE NEURAL NETWORKS?

Text

Audio

Image/Video



Weather

Inputs → **Neural Network** → **Output**

. . .

. . .

. . .

. . .

WHAT ARE NEURAL NETWORKS?

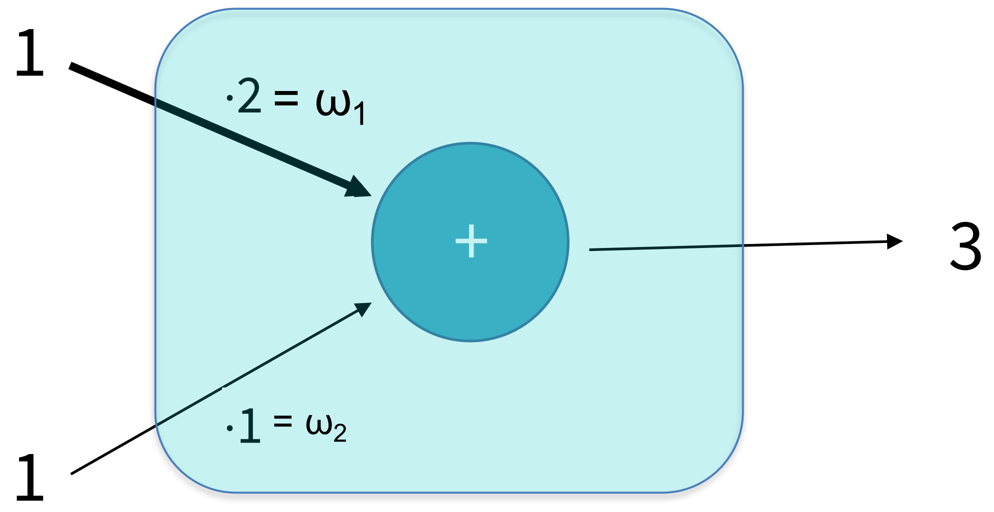
Neural networks approximate complex functions

- Input --> Output mapping with *trainable parameteres*
- Learn from (high dimensional) data
- Capture nonlinear relationships

Examples: forecasting, spatial/temporal downscaling, emulation, …

NEURON

x

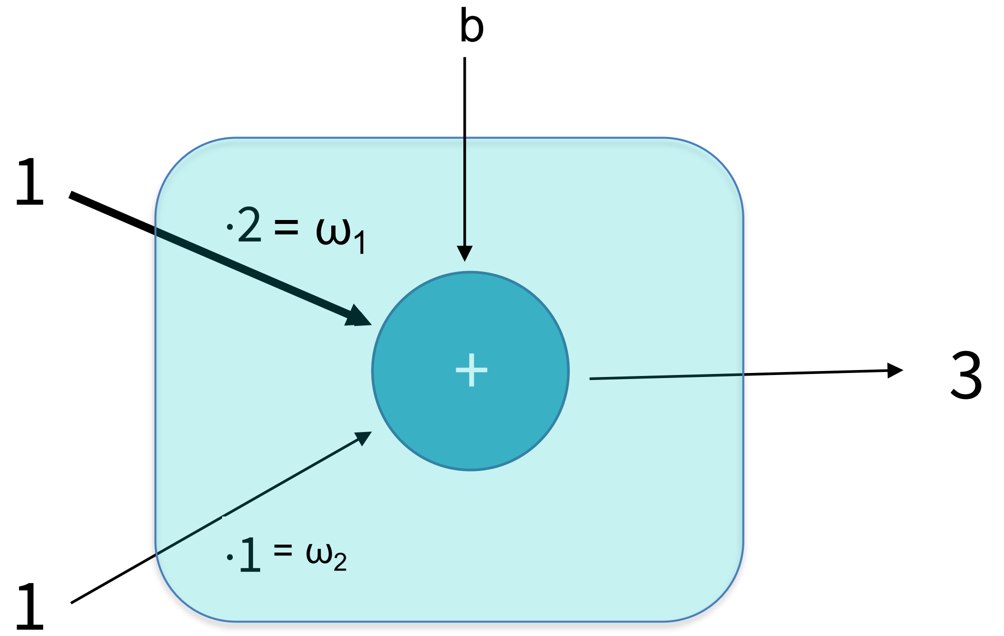


$$y = \omega \bullet x$$

$$\omega = (\omega_{1,} \omega_2)$$

NEURON

x



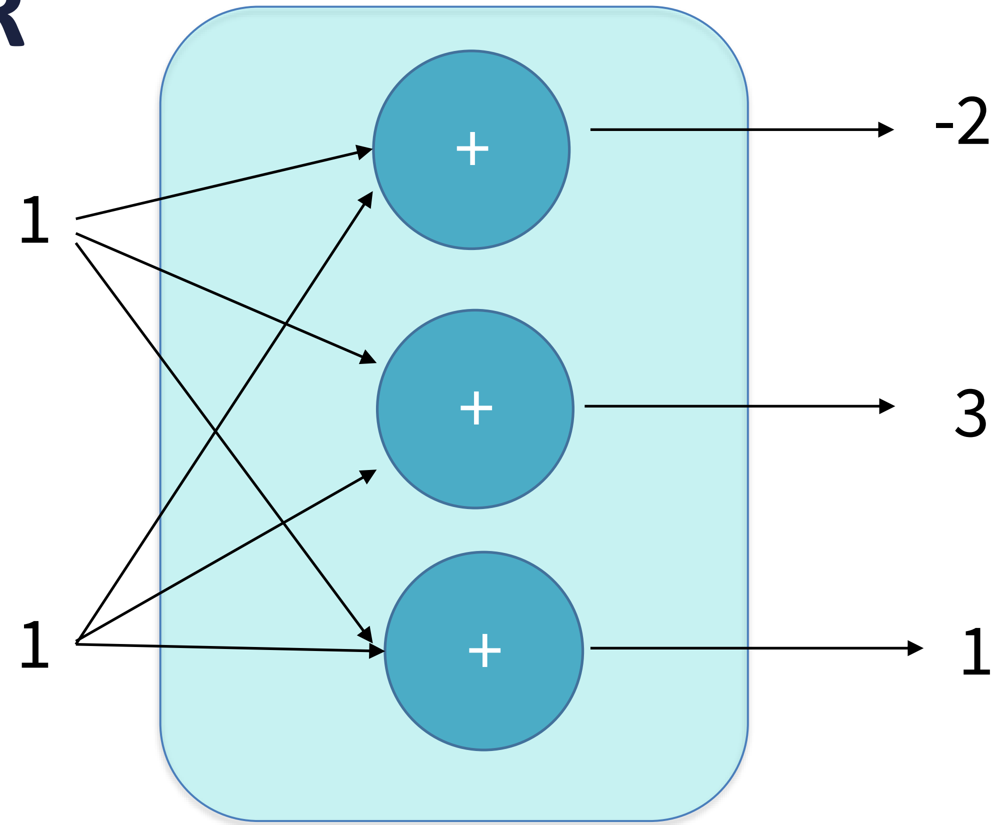
$$y = \omega \bullet x + b$$

$$\omega = (\omega_{1}, \omega_2)$$

$$b = (b_{1}, b_2)$$

DENSE LAYER

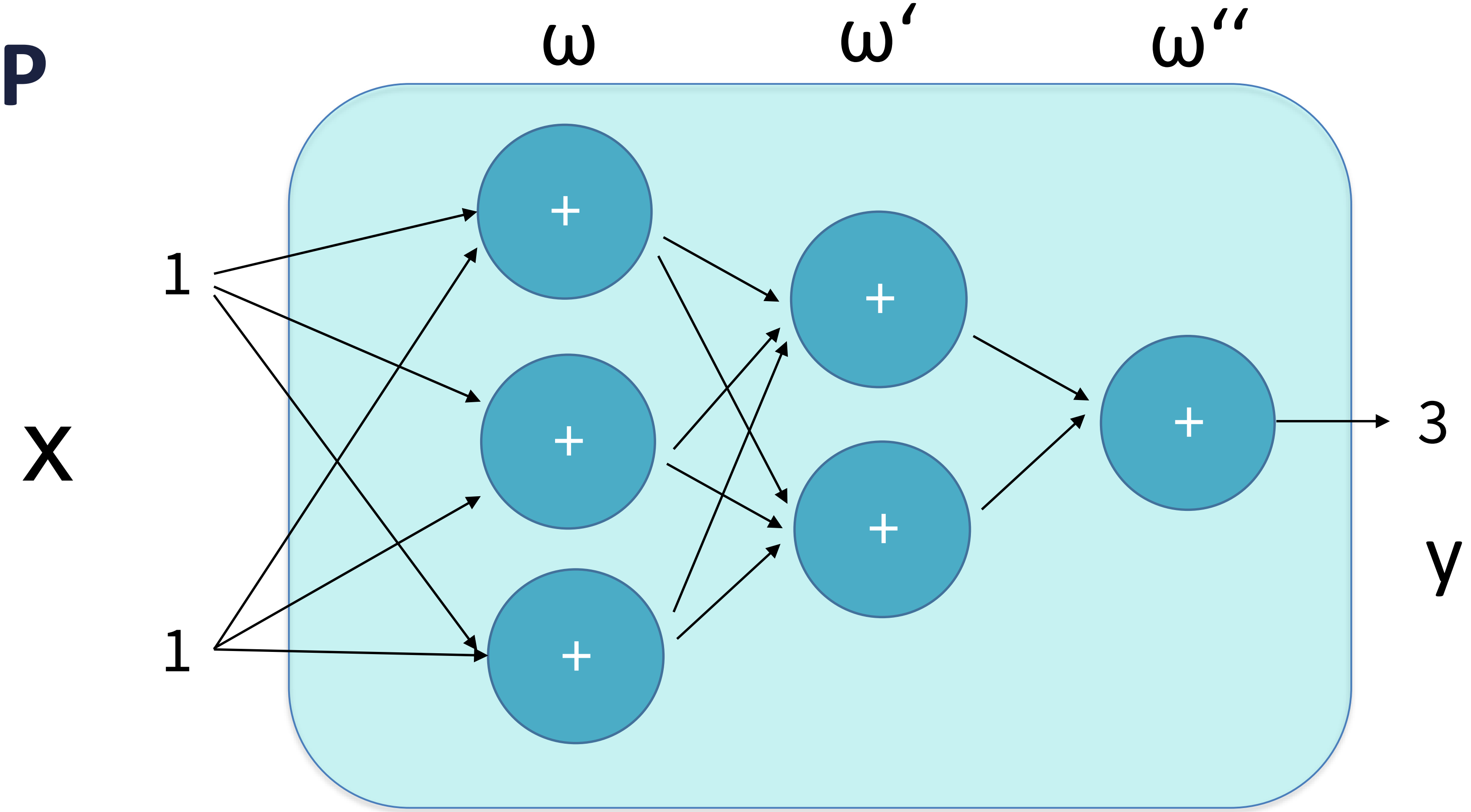
x



$$y = \omega \bullet x$$

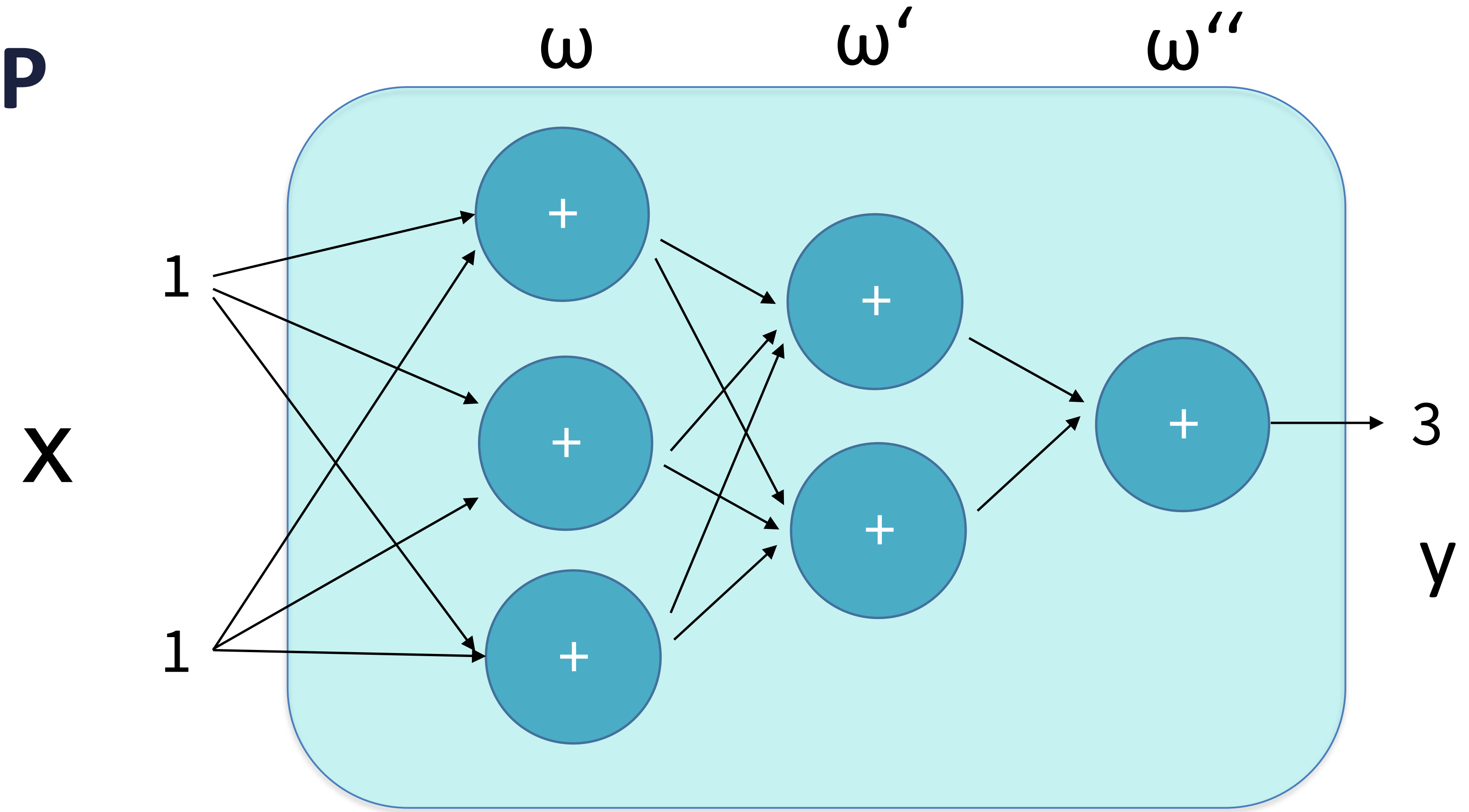
ω is a (2, 3) matrix

MLP



$$y = \omega'' (\omega' (\omega \bullet x))$$

MLP

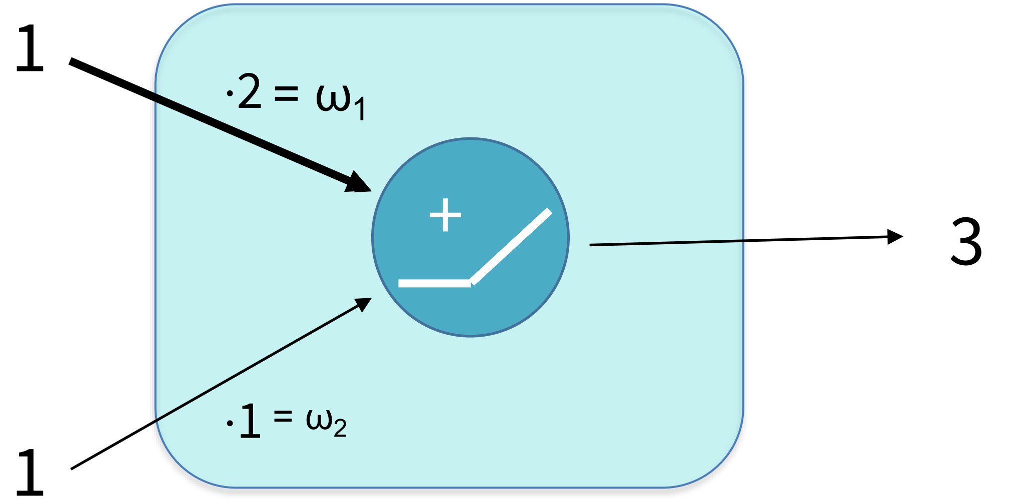


$$y = \omega'' (\omega' (\omega \bullet x))$$

Stacked linear layers = 1 linear layer

ACTIVATION FUNCTION

x



$$y = f(\ \omega \bullet x\ )$$

x ω

f - non-linear function

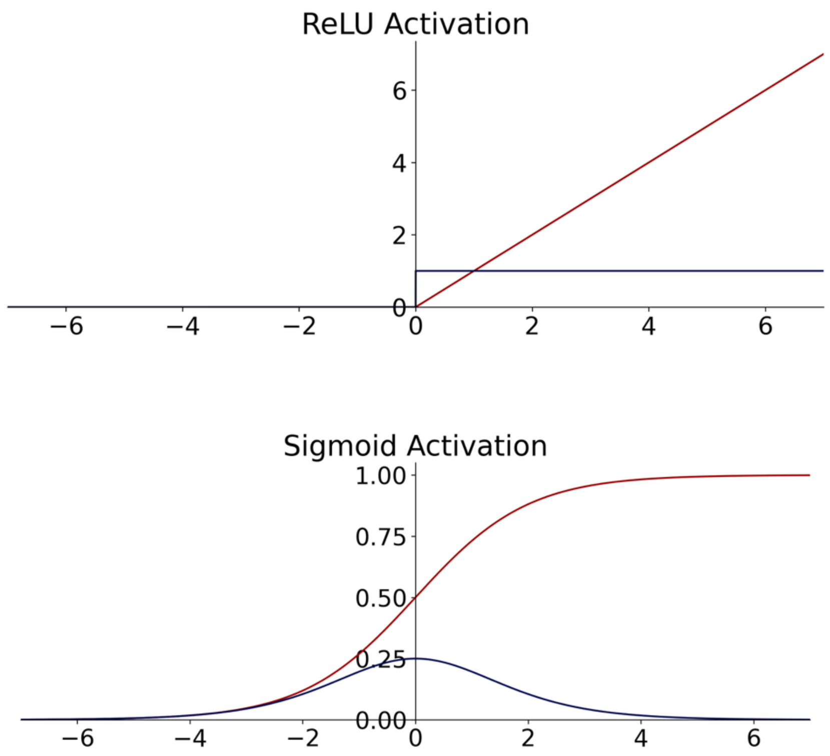
ACTIVATION FUNCTION

Allow nonlinearity

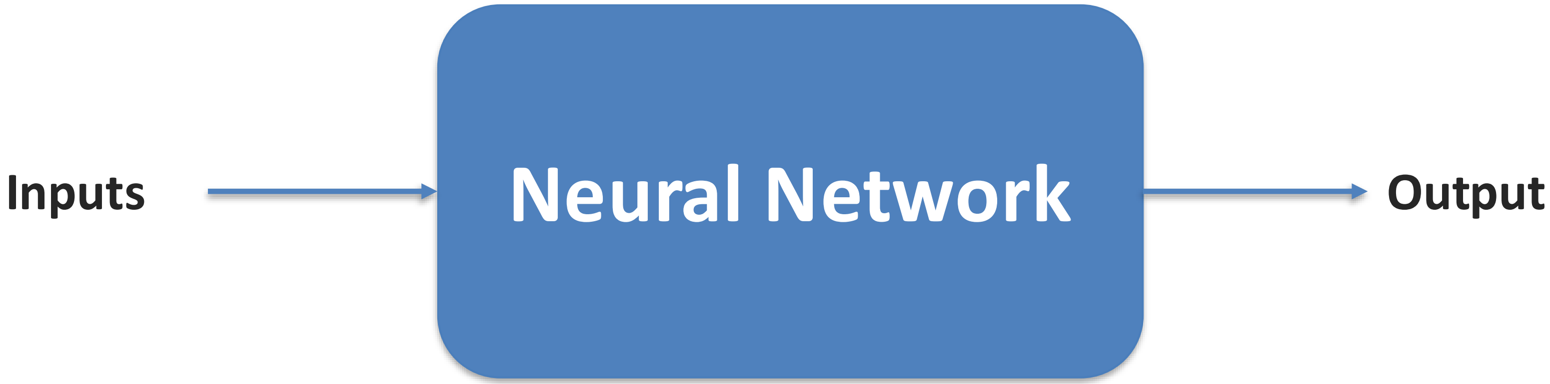
Stabilise training

Allow customized behaviour:

- Probabilities
- Physical constraints

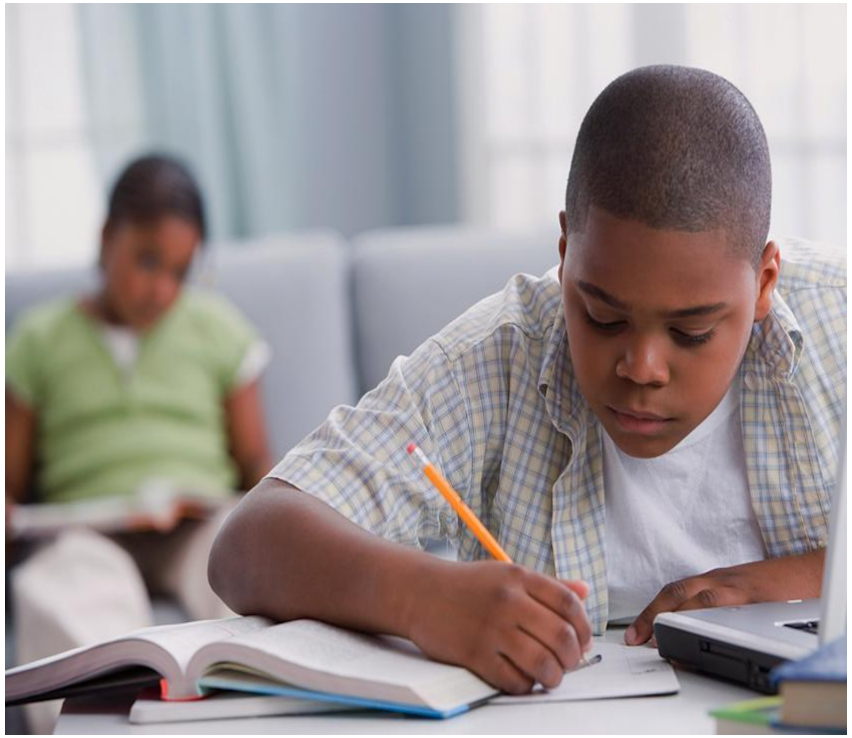


HOW DO WE OPTIMISE THE PARAMETERS?

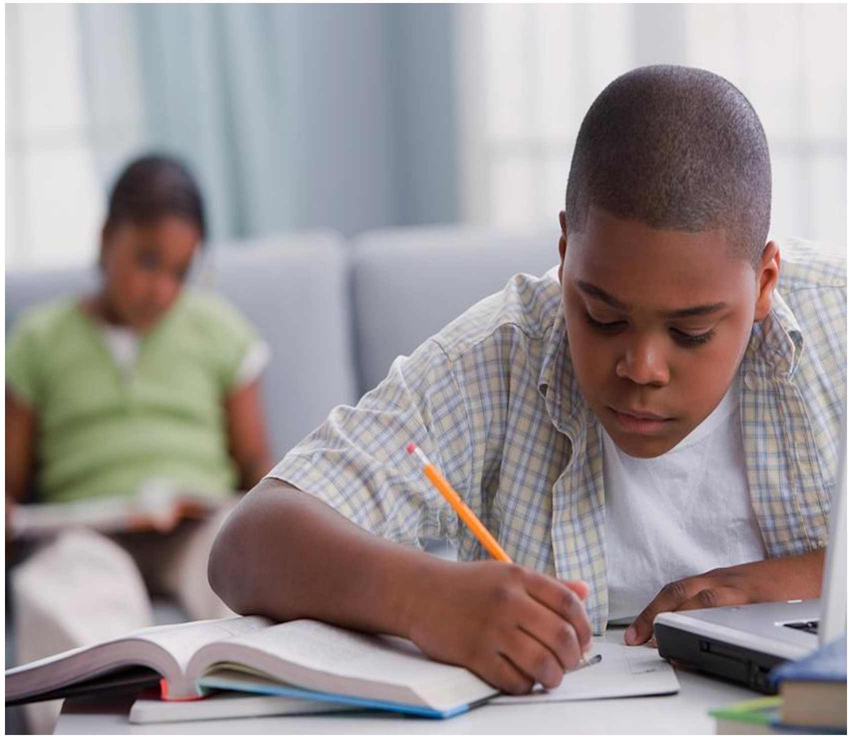


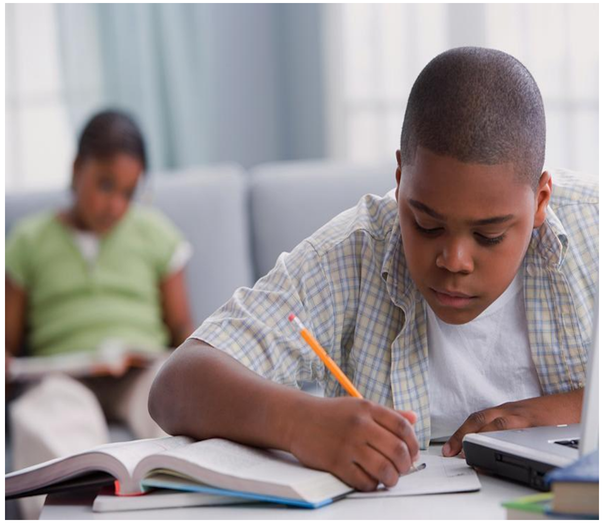
HOW DO WE (HUMANS) LEARN?

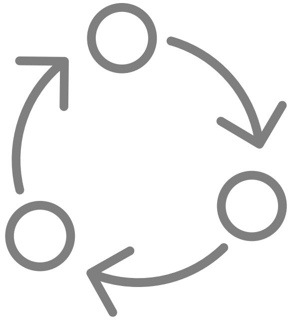
GET YOUR HOMEWORK DONE!

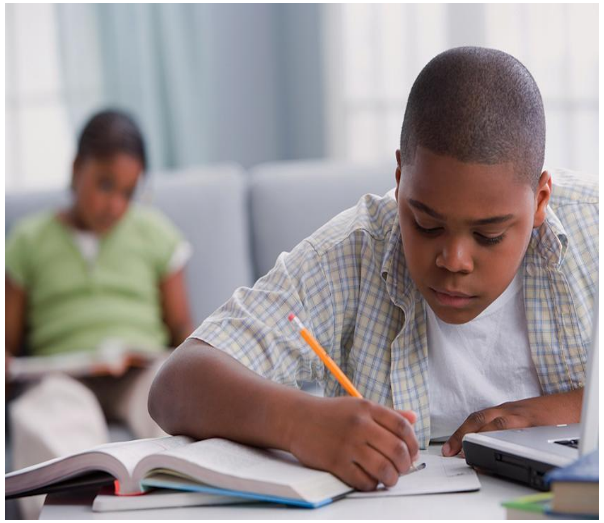


REVIEW YOUR HOMEWORK!

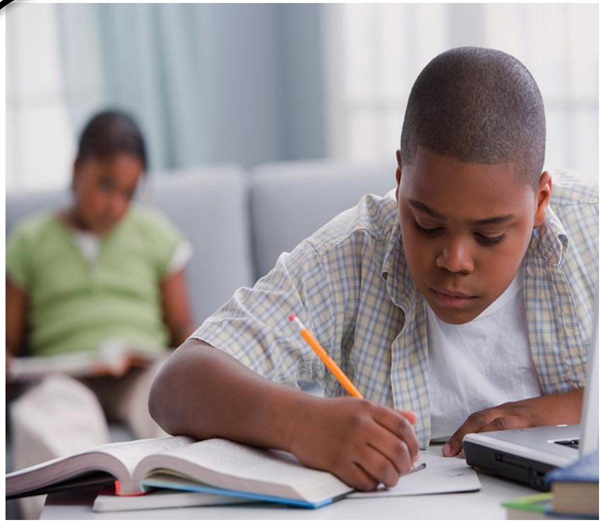




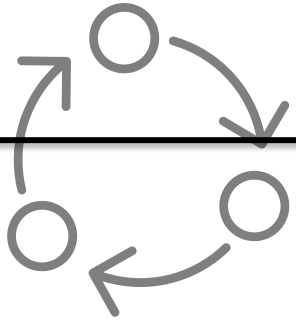


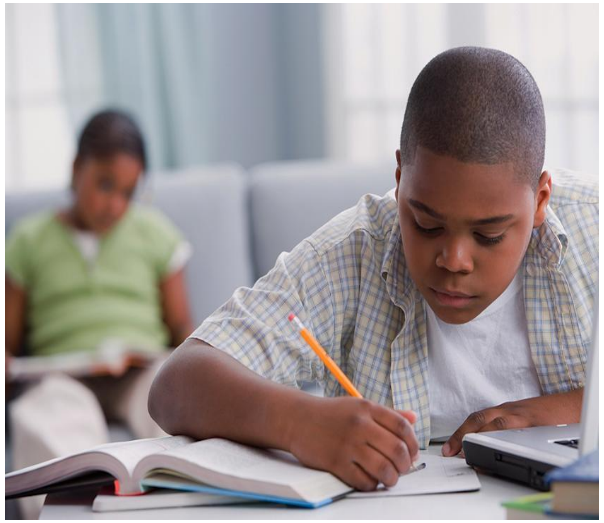


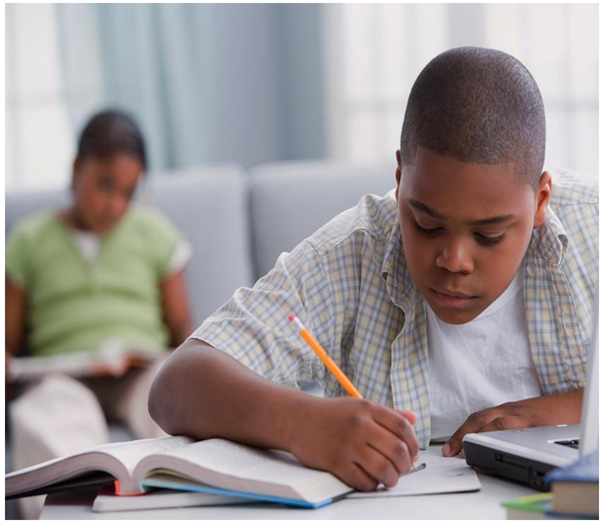
Model
forward pass

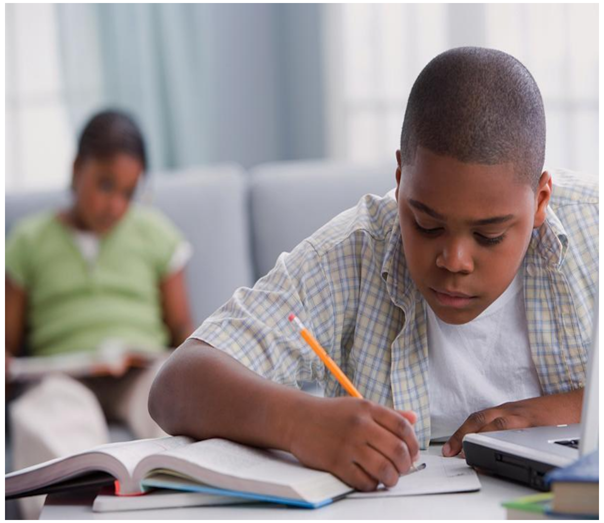


prediction

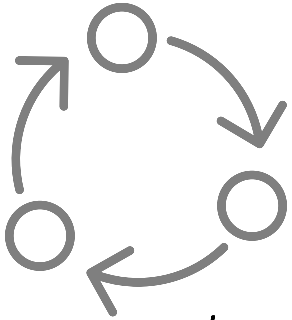






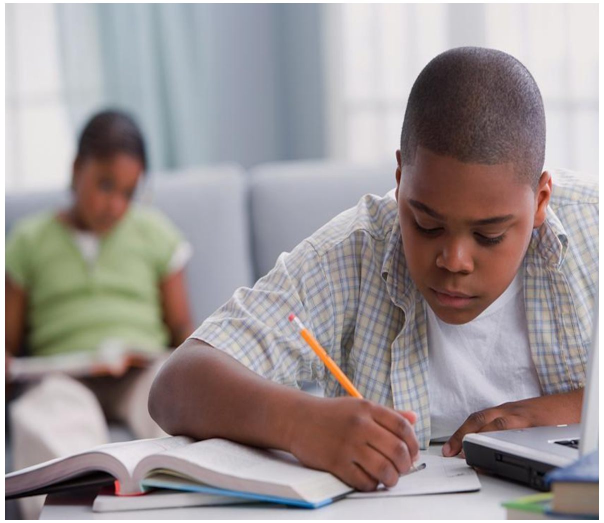


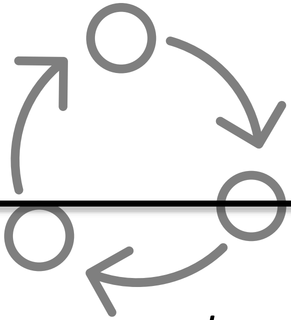
prediction



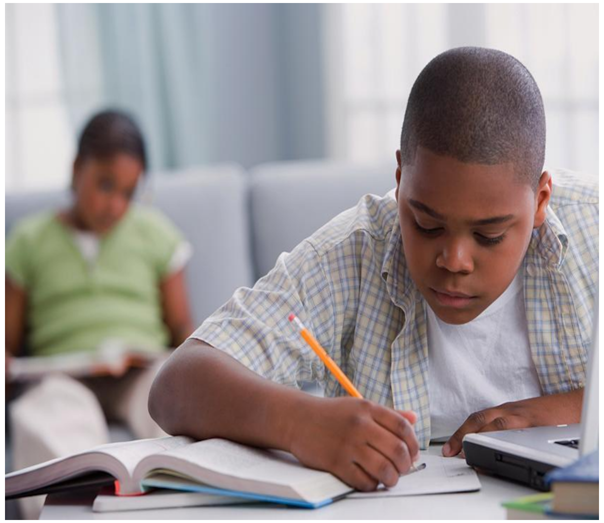
target

Loss *computation*

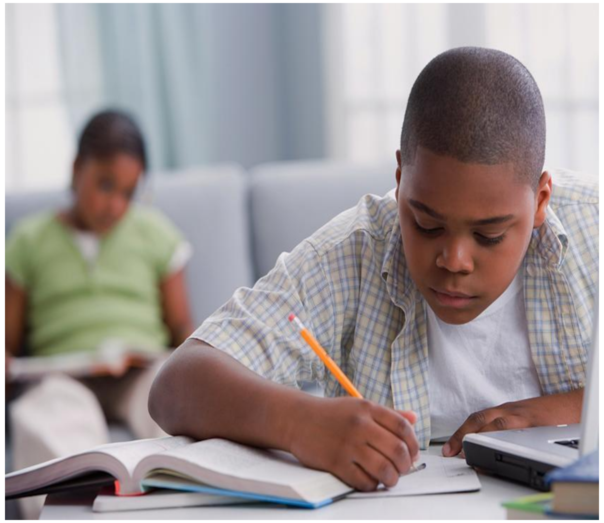


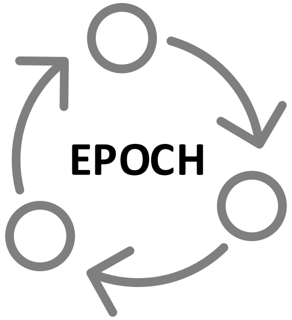


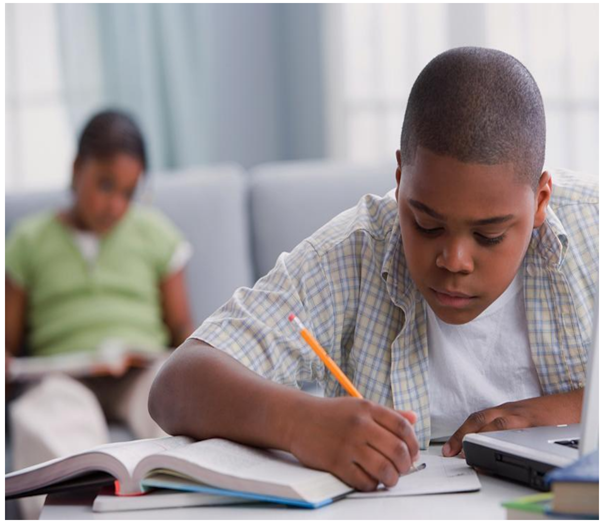
Model
backward pass



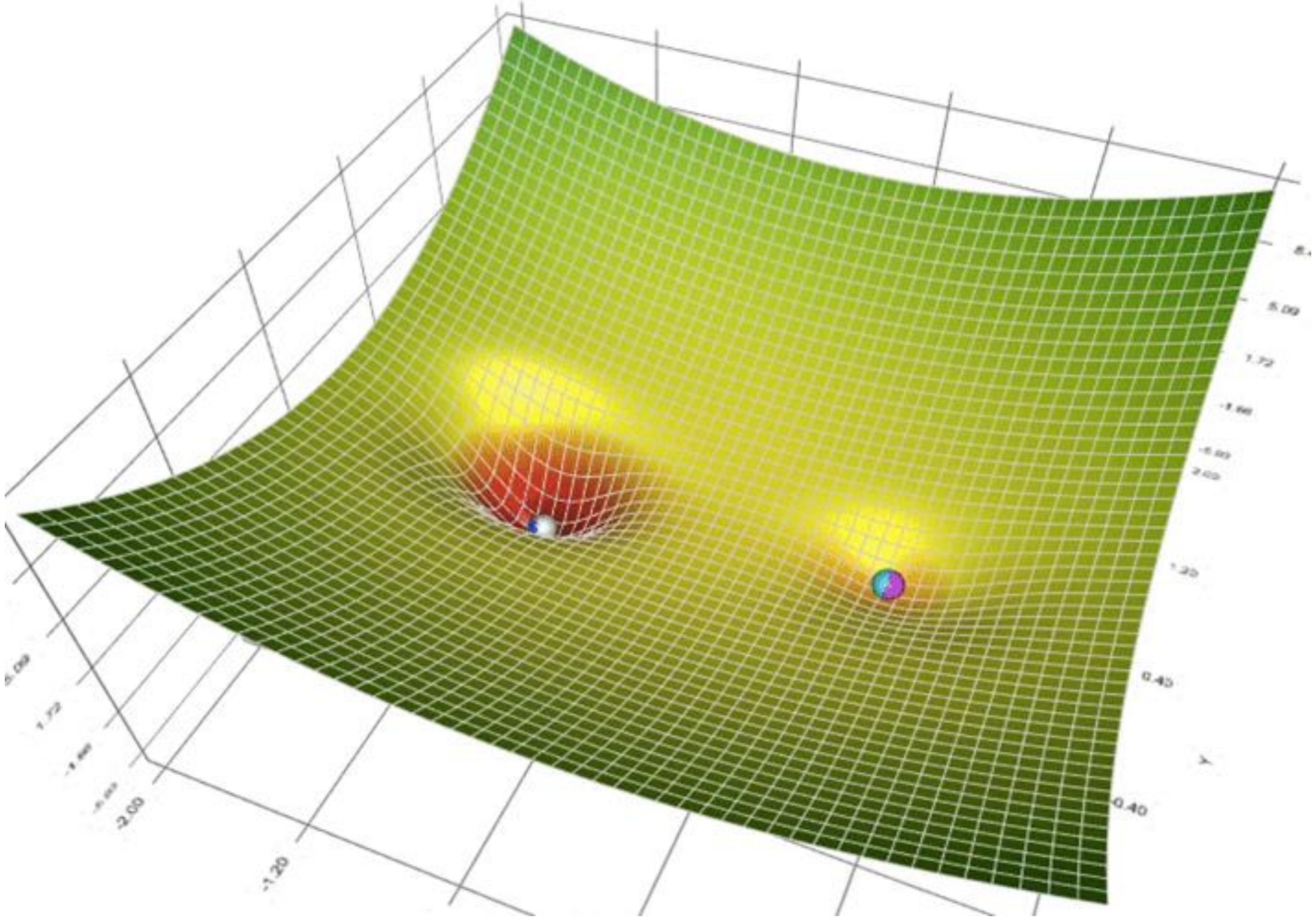
target




EPOCH



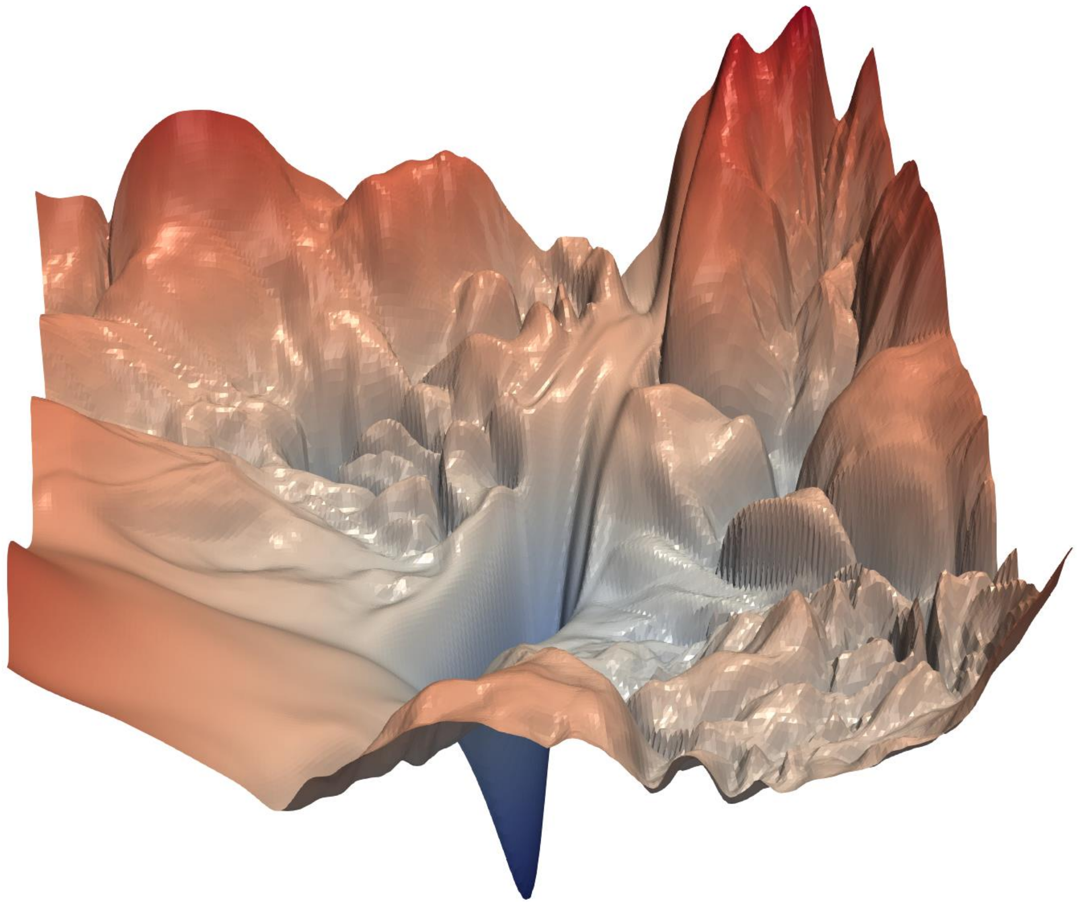
OPTIMIZATION PROBLEM



1. Compute loss
2. Gradient Descent
3. Update model parameters

IN REALITY …

- Loss surface usually highly irregular
- Different architectures choices change surface
- Take small steps toward minimum
- Use state of the art optimiser
  - Adam (or other variants)
- Regularisation for better optimum



SCALE YOUR MODEL

Compute

Data

Model

REGULARISATION

Neural networks are extremely powerful function approximators.

- They can learn not just patterns (but also *noise*) if left unchecked.
- This happens when the network fits training data **too perfectly**, losing the ability to generalize to unseen data.

REGULARISATION

Neural networks are extremely powerful function approximators.

- They can learn not just patterns (but also *noise*) if left unchecked.
- This happens when the network fits training data **too perfectly**, losing the ability to generalize to unseen data.

= **Overfitting !!!**

REGULARISATION

Neural networks are extremely powerful function approximators.

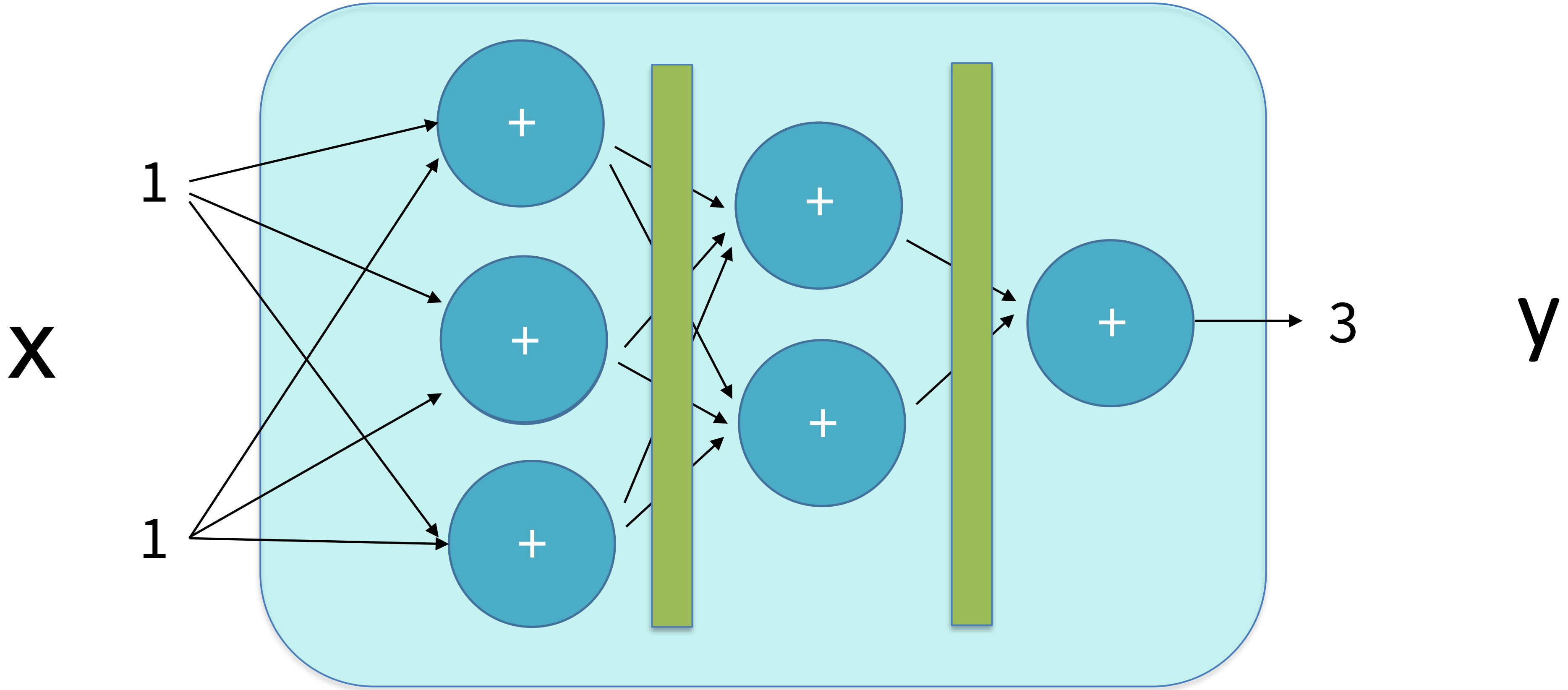
- They can learn not just patterns (but also *noise*) if left unchecked.
- This happens when the network fits training data **too perfectly**, losing the ability to generalize to unseen data.

= **Overfitting !!!**

Regularisation methods **add constraints or penalties** that discourage overfitting.

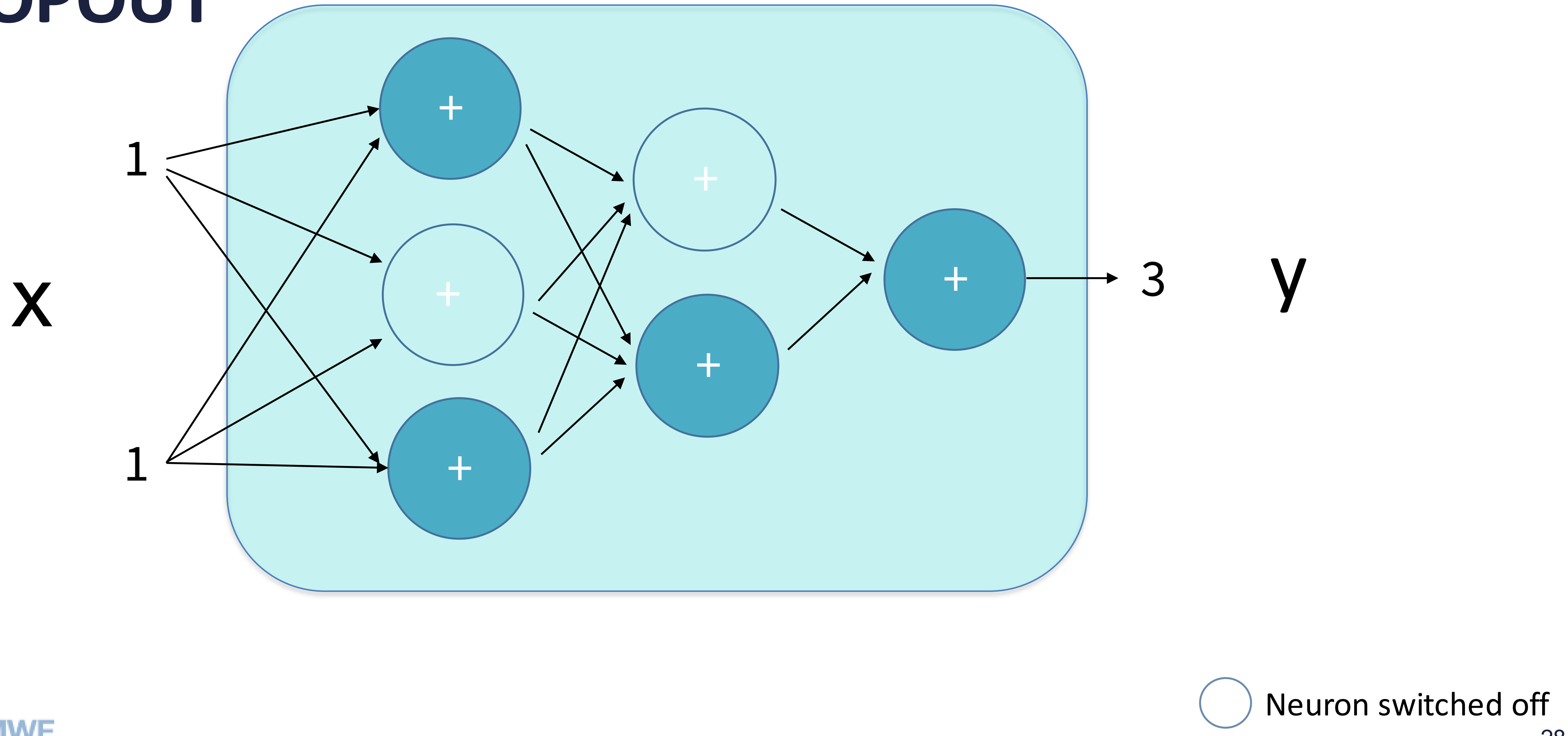
They guide the model to learn simpler, smoother, or more robust representations.

NORMALISATION

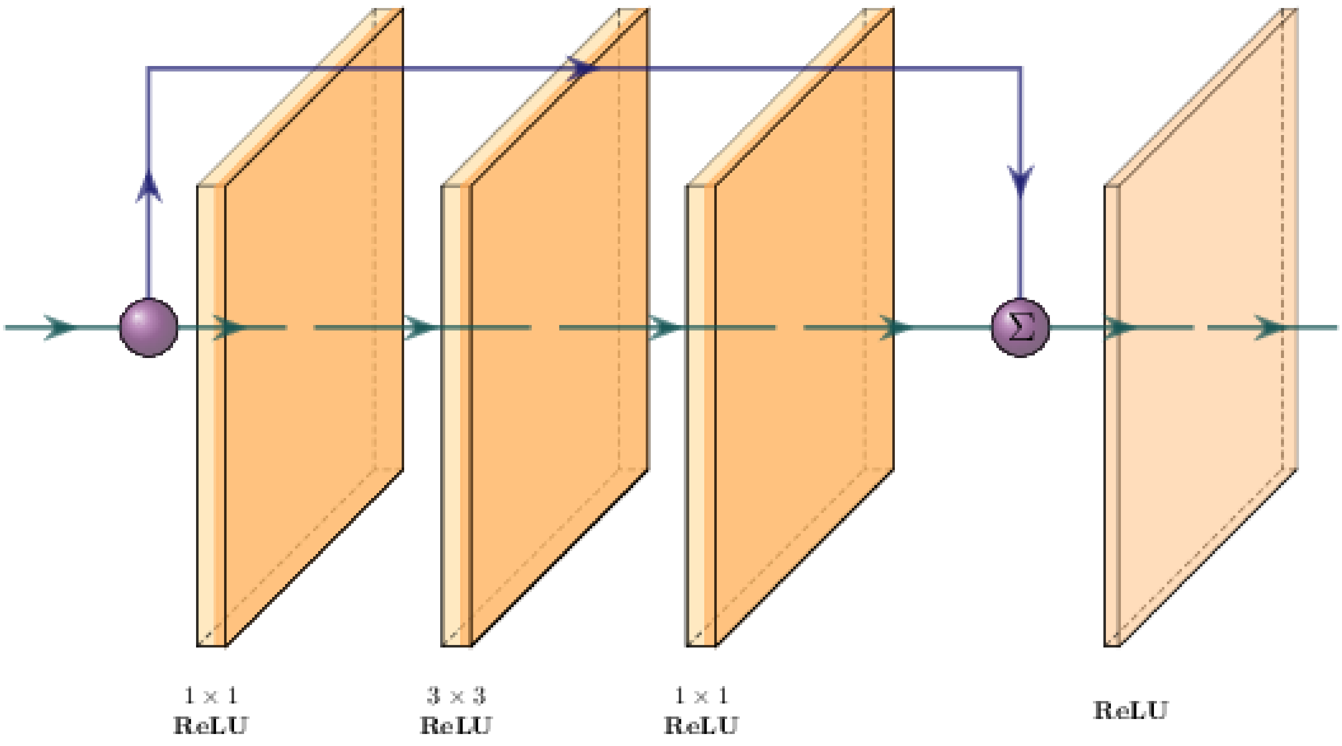


BatchNorm, **LayerNorm**, GroupNorm, …

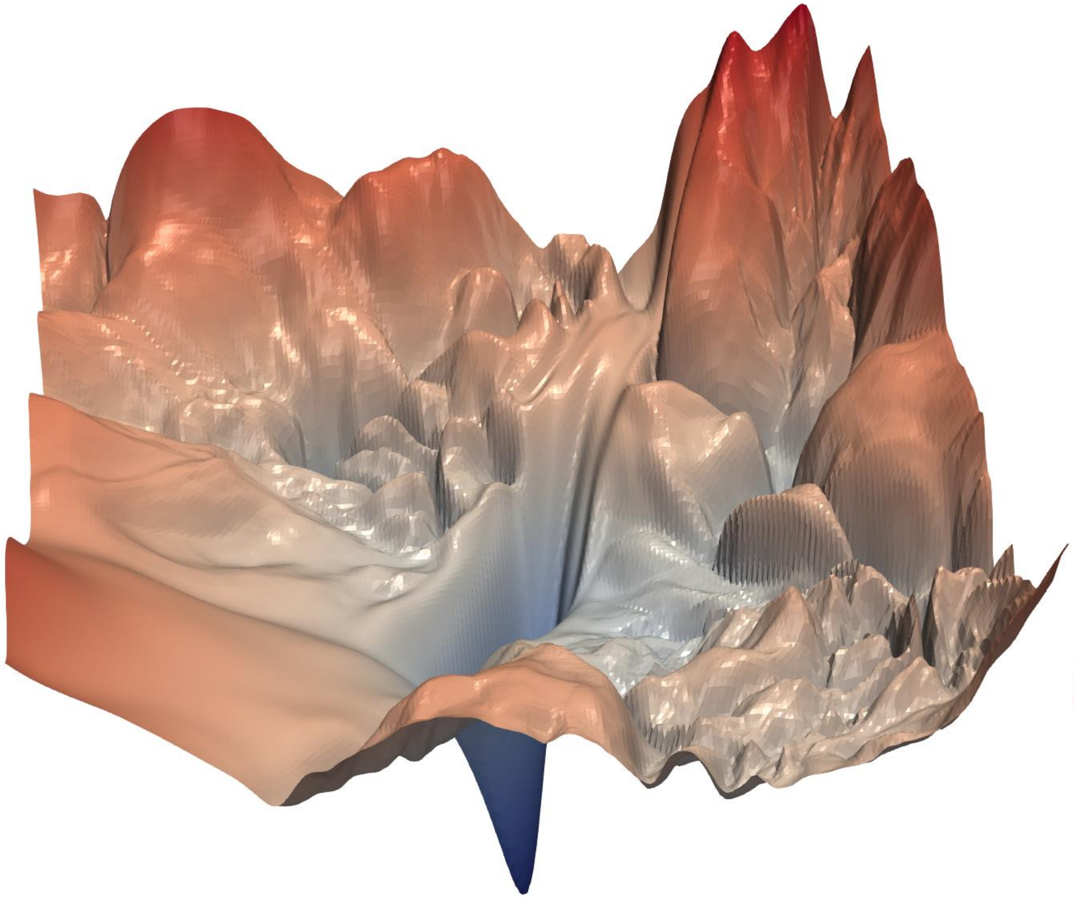
DROPOUT

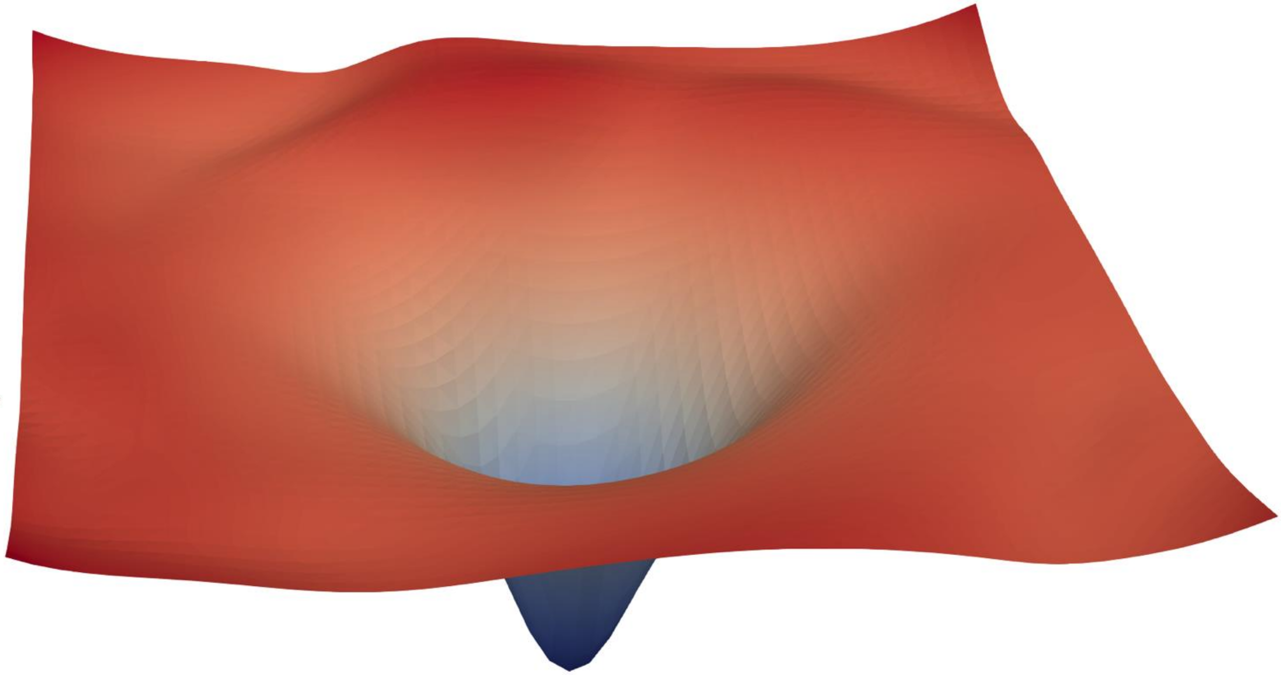


RESIDUAL CONNECTIONS

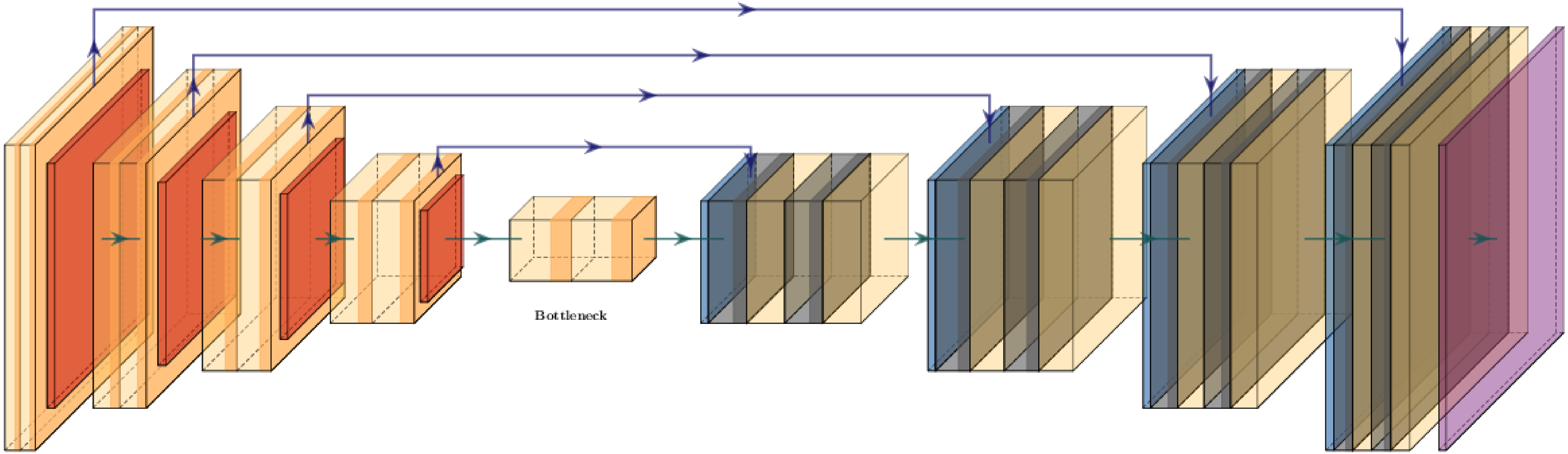


WHY DO WE USE RESIDUAL CONNECTIONS?





U-NET



Encoder

Bottleneck

Decoder

WHY DO WE COMPRESS THE INFORMATION?

“Compression forces understanding.”

EARTH SYSTEM MODELLING

Data processing

- Normalize per variable
- Use training statistics

Modelling

- Exploit spatial and temporal relationships
- Learn increments/residuals
- High memory requirements -> small batches

Validation

- Loss is computed in normalised space
- Metrics should be in physical units
- Be careful with distribution shifts:
  - Seasonal cycle
  - Daily cycles
  - Regime change

SUMMARY

Neural networks

Learn hierarchical representations

Model nonlinear physical processes

Training

Minimise a loss function

Use gradient-based optimisation

Scale your model

Stabilise training

Deep architectures

Avoid overfitting (regularisation techniques)

Earth system context

Respect physical scales

Choose normalization carefully

REFERENCES

- Getting Started with Neural Networks (trainingnns.github.io)
- A curate list of awesome Deep Learning tutorials, projects and communities.
- Stanford CS230 | Autumn 2025 | Deep Learning | YouTube Series
- 3 Blue 1 Brown – Animated math