# Neural networks and deep learning

## Training course: Machine learning for weather prediction

Mario Santa Cruz Lopez
ECMWF
mario.santacruz@ecmwf.int

# Back to 2012

# Back to 2012

Machine learning was already being used for a variety of tasks:

- Spam recognition
- Fraud detection
- Recommendations

- **Feature engineering** is manual, domain-specific, and time-consuming.
- It doesn't scale with data (performance & cost)

Wikipedia
The Free Encyclopedia

Search Wikipedia | Search

Donate Create account Log in

# European Centre for Medium-Range Weather Forecasts

18 languages

Article Talk | Read Edit View history Tools

From Wikipedia, the free encyclopedia

Coordinates: 51°25′11″N 0°57′03″W

Contents hide

(Top)
History
Objectives
Copernicus
Work and projects
Forecasting
Monthly and seasonal forecasts
Early warning of severe weather events
Satellite data
Reanalysis
Operational forecast model
Member and co-operating states
See also
References
Further reading
External links

The **European Centre for Medium-Range Weather Forecasts** (**ECMWF**) is an independent intergovernmental organisation supported by most of the nations of Europe. It is based at three sites: Shinfield Park, Reading, United Kingdom; Bologna, Italy; and Bonn, Germany. It operates one of the largest supercomputer complexes in Europe and the world's largest archive of numerical weather prediction data.[1]

## History [ edit ]

ECMWF was established in 1975, in recognition of the need to pool the scientific and technical resources of Europe's meteorological services and institutions for the production of weather forecasts for medium-range timescales (up to approximately two weeks) and of the economic and social benefits expected from it. The Centre employs about 350 staff, mostly appointed from across the member states and co-operating states.[1]

In 2017, the centre's member states accepted an offer from the Italian Government to move ECMWF's data centre to Bologna, Italy. The new site, a former tobacco factory, would be redesigned by the architecture firm gmp.[2]

During 2020, the Centre arranged to move its Copernicus operations away from Reading and into European Union territory.[3] Following bids from Toulouse,[3] Italy,[4] Austria,[5] Germany,[6] Spain[7] and Ireland,[8] eventually Bonn (Germany) was chosen.[9] The move has been directly attributed to Brexit.[9][10]

**European Centre for Medium-Range Weather Forecasts**

ECMWF

| | |
|---|---|
| Established | 1975 (50 years ago) |
| Headquarters | Reading |
| Country | United Kingdom |
| Coordinates | 51°25′11″N 0°57′03″W |
| Chief Executives | Florence Rabier |
| Directors | Florence Rabier |

Appearance hide

Text
Small
Standard
Large

Width
Standard
Wide

Color (beta)
Automatic
Light
Dark

Text

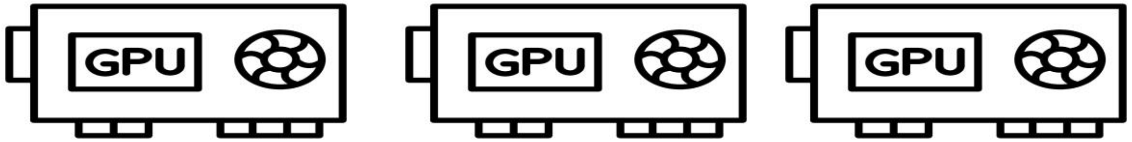
Audio

Image/Videos

## Inconvenients

- Performance plateaus as complexity grows.
- Hard to generalize across raw data types
- Decision Trees / Random Forests perform well on tabular data but poorly on high-dimensional data

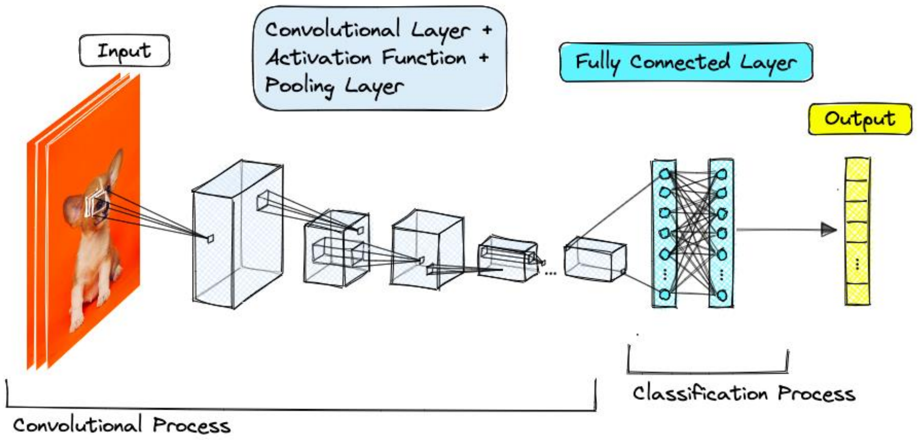
# At the same time




**GPUs** were originally developed to render 3D graphics efficiently in video games.

Handling millions of pixels and complex visual effects in parallel.

# ImageNet

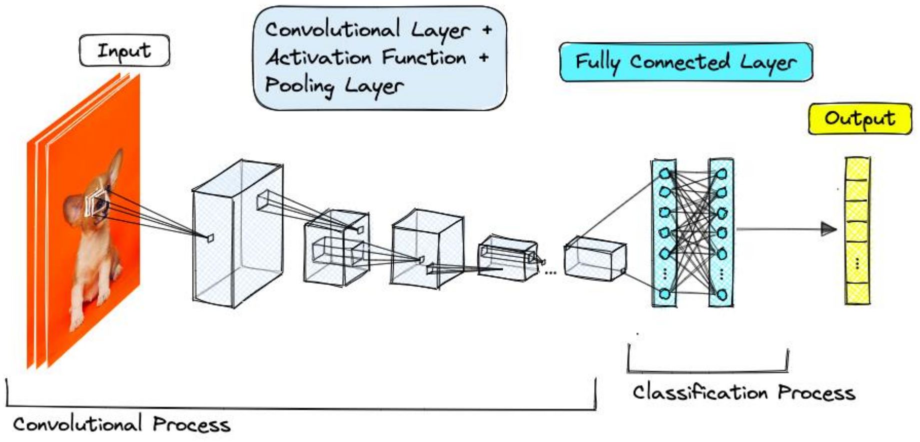



- 1.2 million images dataset
- 1000 categories

**Before 2012 …**

- The error rate hovered around **26%**

# ImageNet




- 1.2 million images dataset
- 1000 categories

**Before 2012 …**

- The error rate hovered around **26%**

**In 2012 …**

- Developed by **Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton** (University of Toronto).
- Introduced AlexNet, **neural network** trained with **GPUs.**
- It dropped the **error rate** from **~26 % to ~16%.**

# History of major breaktroughs

***2012*** - AlexNet wins the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) using a deep convolutional neural network.

***2016*** - AlphaGo (by DeepMind) defeats a top human player in Go (2016)

***2017*** - Introduction of the Transformer architecture model (by Google Brain) for natural language processing.

***2021*** - AlphaFold (DeepMind) demonstrates highly accurate protein folding predictions.

***2022*** - ChatGPT (based on GPT-3.5) publicly launched and quickly becomes viral.

# History of major breaktroughs

***2012*** - AlexNet wins the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) using a deep convolutional neural network.

***2016*** - AlphaGo (by DeepMind) defeats a top human player in Go (2016)

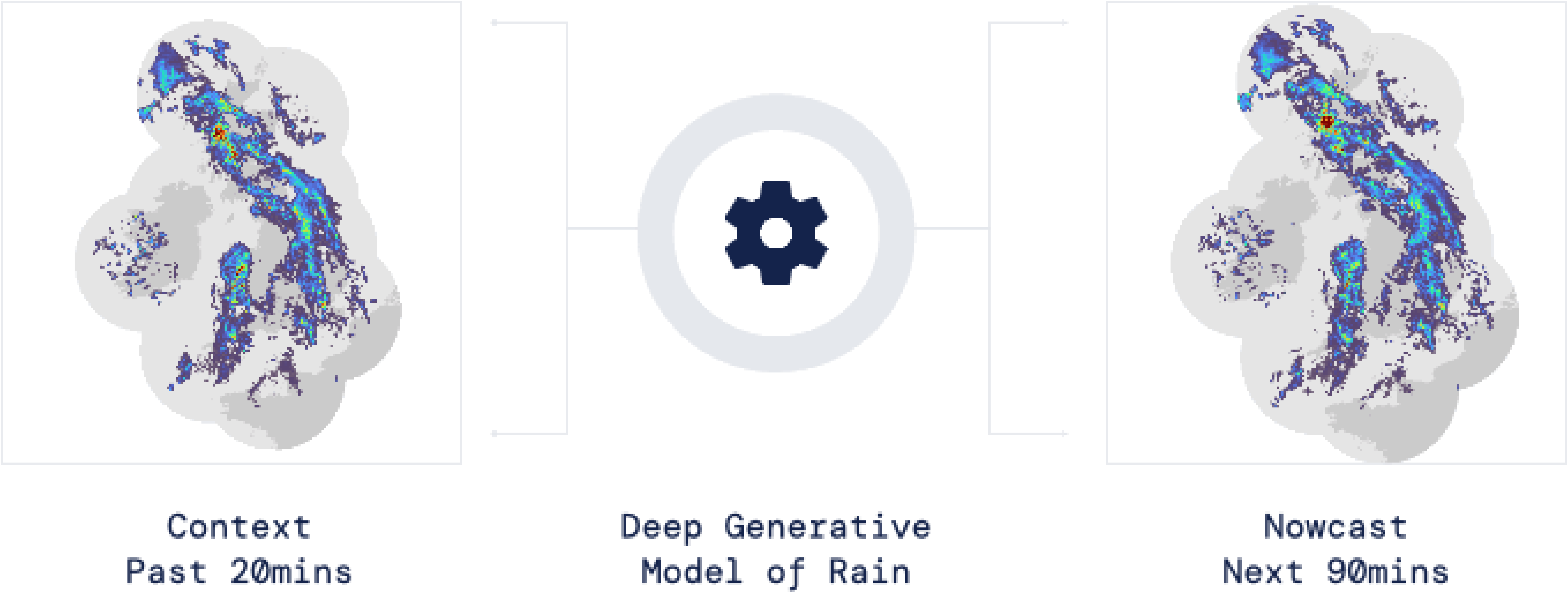
***2017*** - Introduction of the Transformer architecture model (by Google Brain) for natural language processing.

***2021*** - AlphaFold (DeepMind) demonstrates highly accurate protein folding predictions.

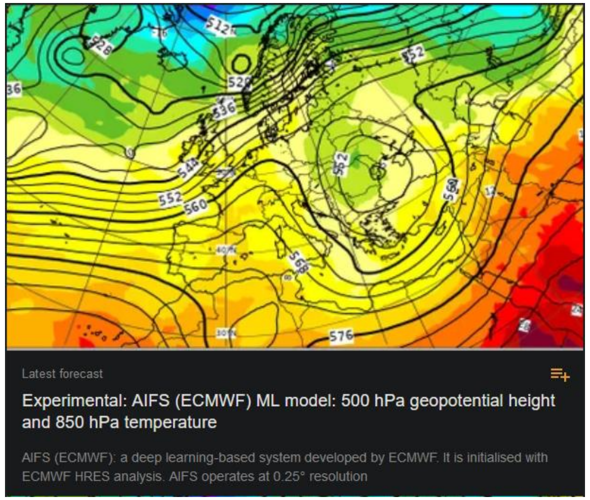
***2022*** - ChatGPT (based on GPT-3.5) publicly launched and quickly becomes viral.

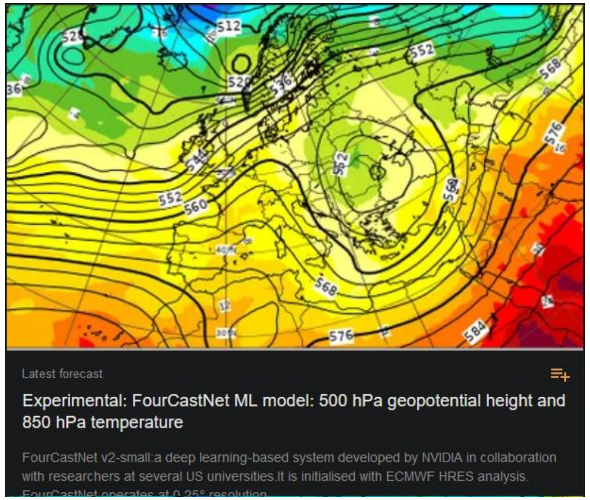
***Feb 2025*** - AIFS Single becomes an operational model.

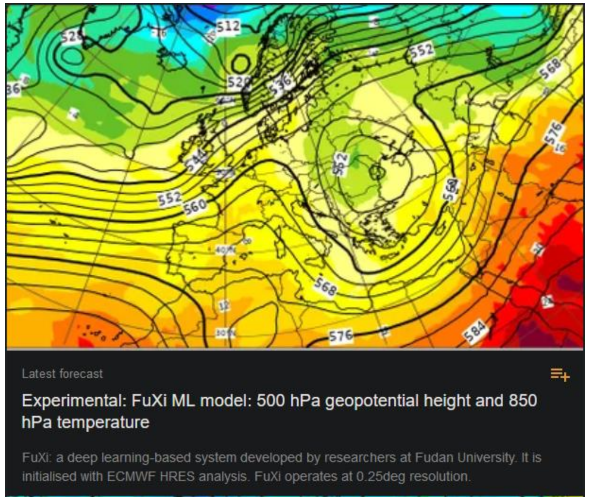
# Deepmind Nowcasting predicting the future

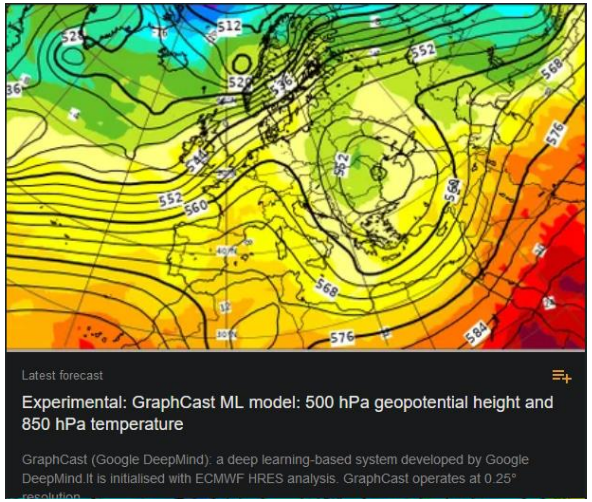


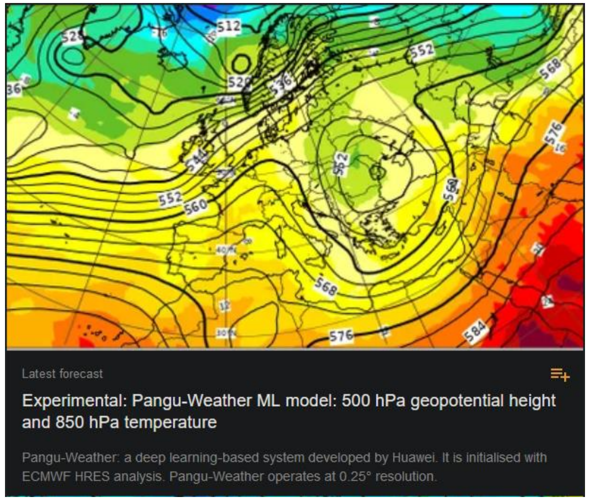
# The Rise of Data-Driven Weather Forecasting



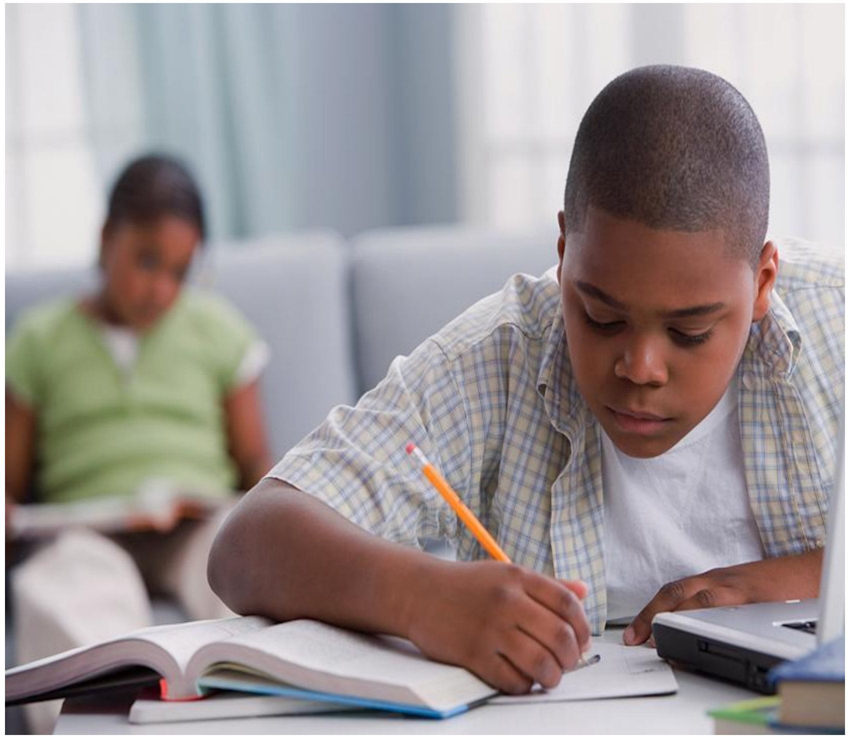

Charts.ECMWF.int

# Outline

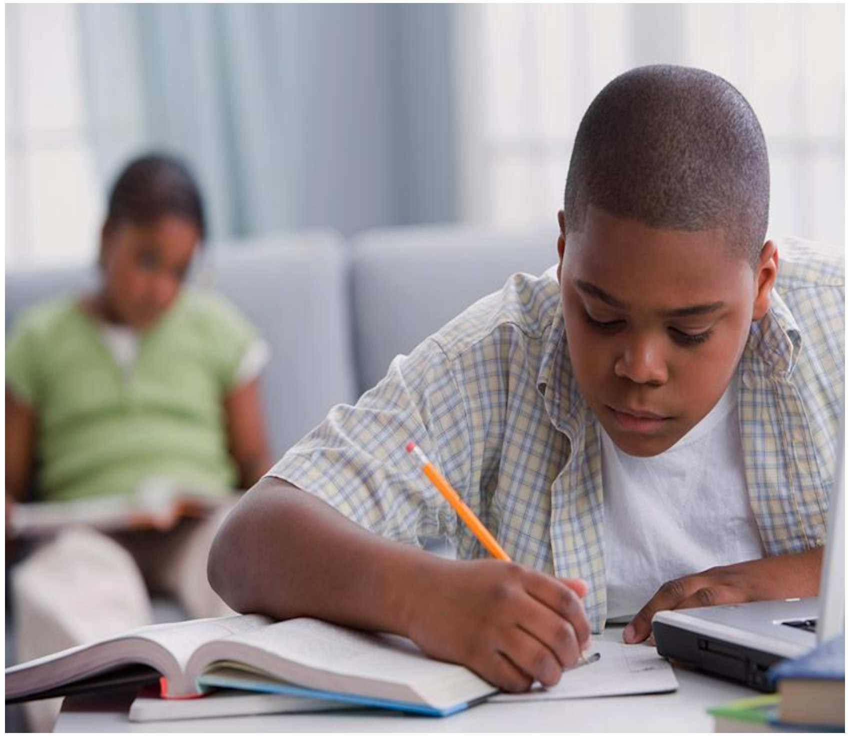
- Learning process
- Dense networks
- Neural network training and GPUs
- Convolutional neural networks
- Recurrent networks
- Transformers

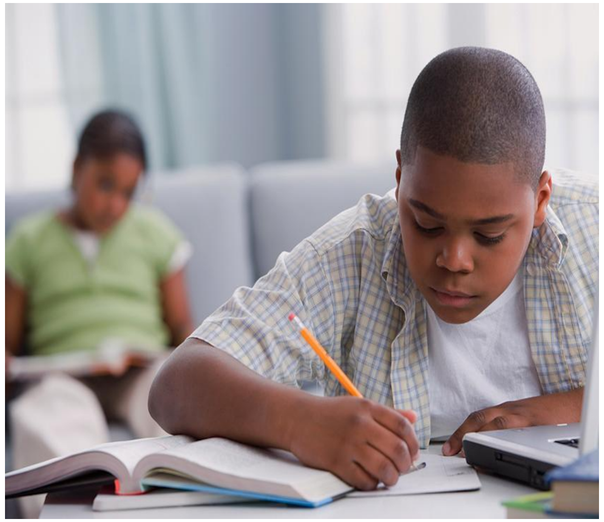
# How do we (humans) learn?

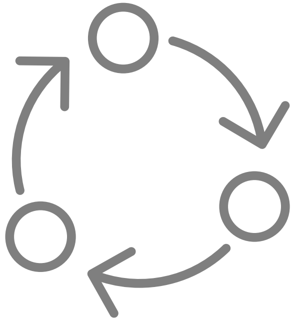
# Homework

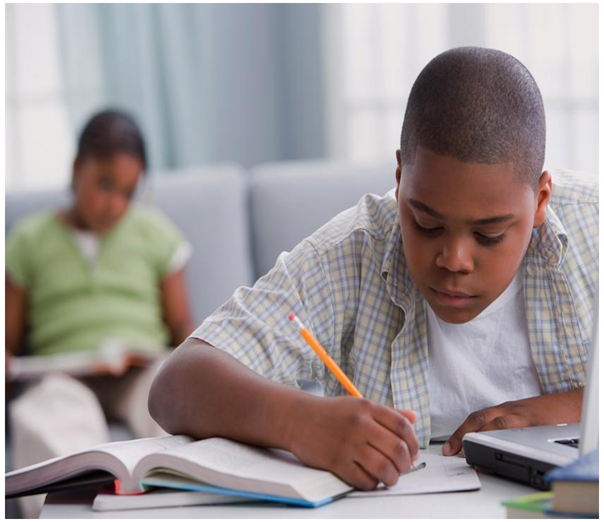



# Review your homework

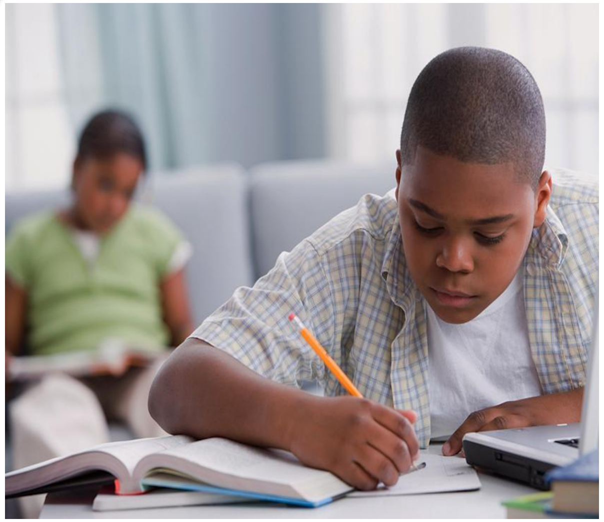



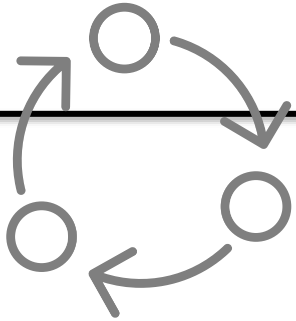


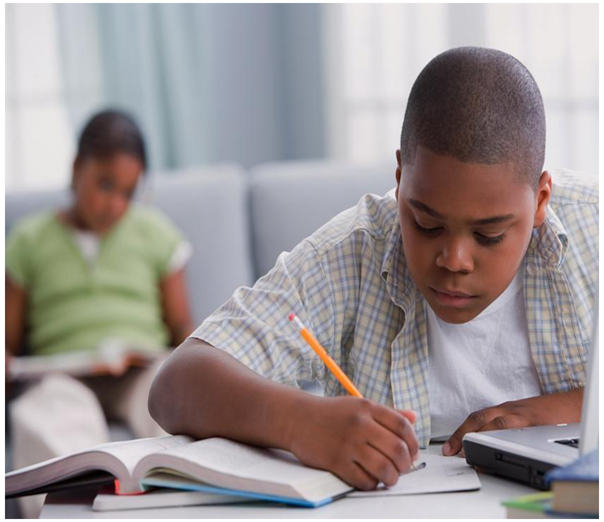


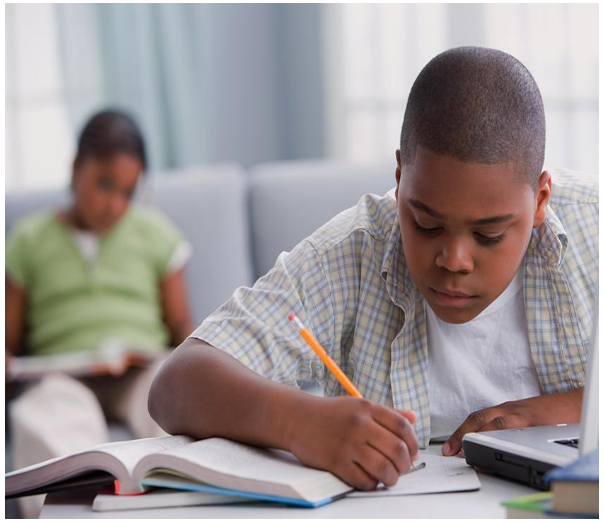


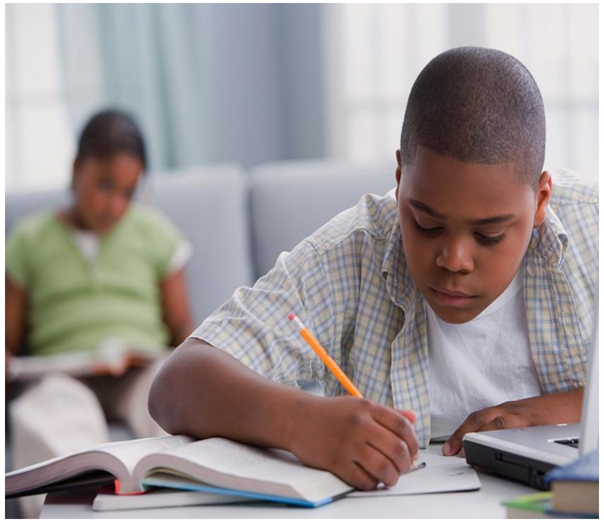
Forward pass



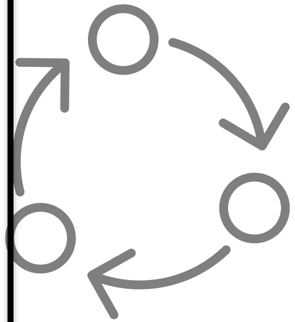


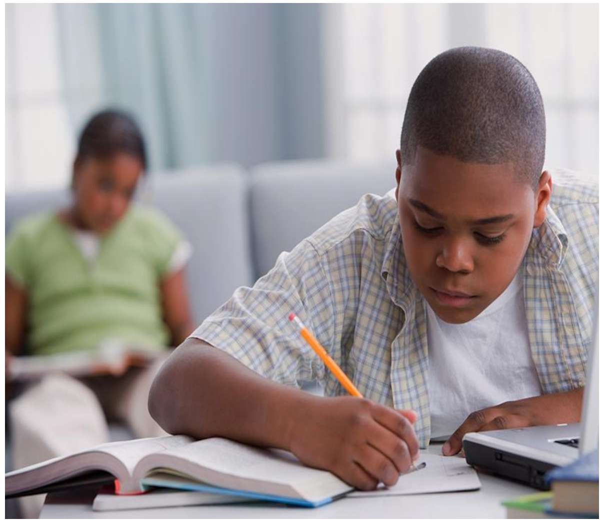


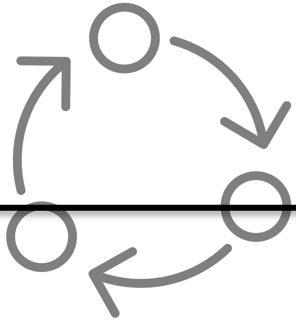




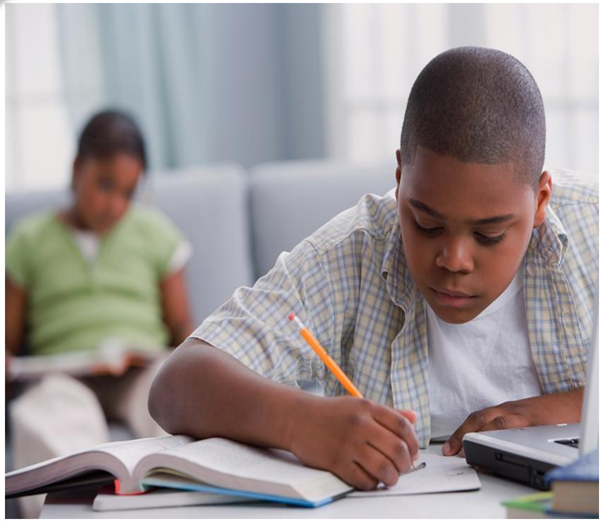
Loss function



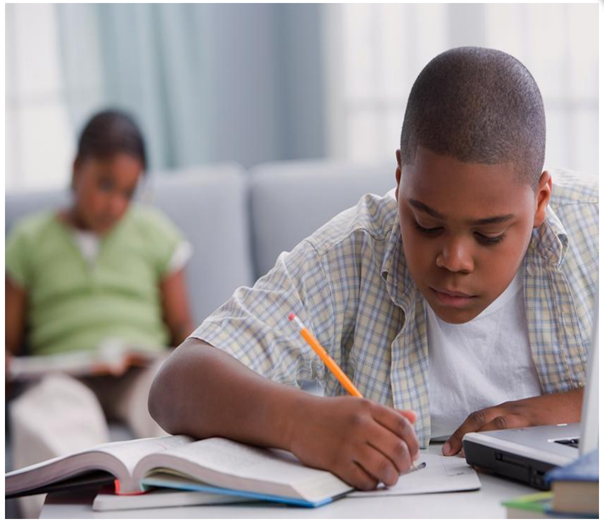


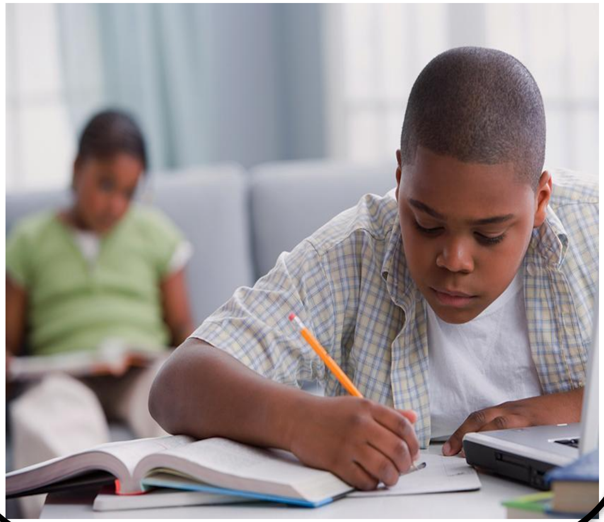


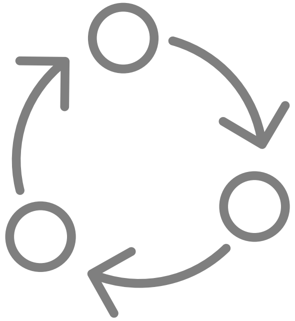
Backward pass

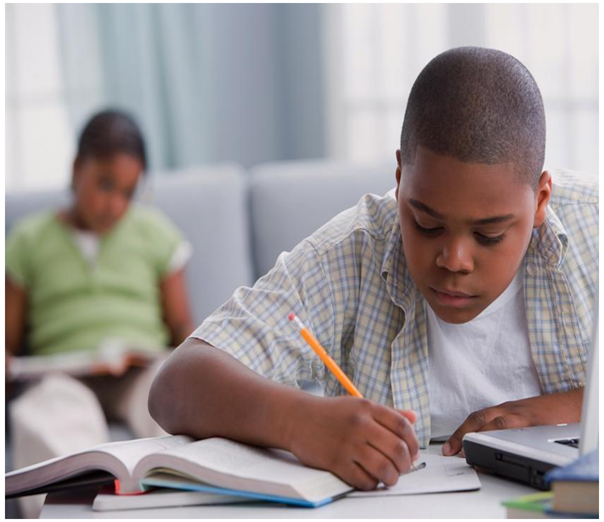


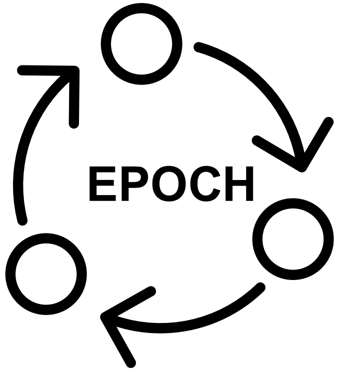
Parameters update

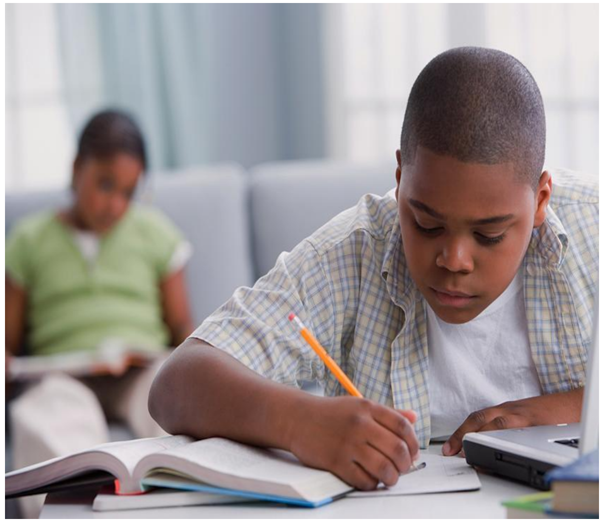









EPOCH

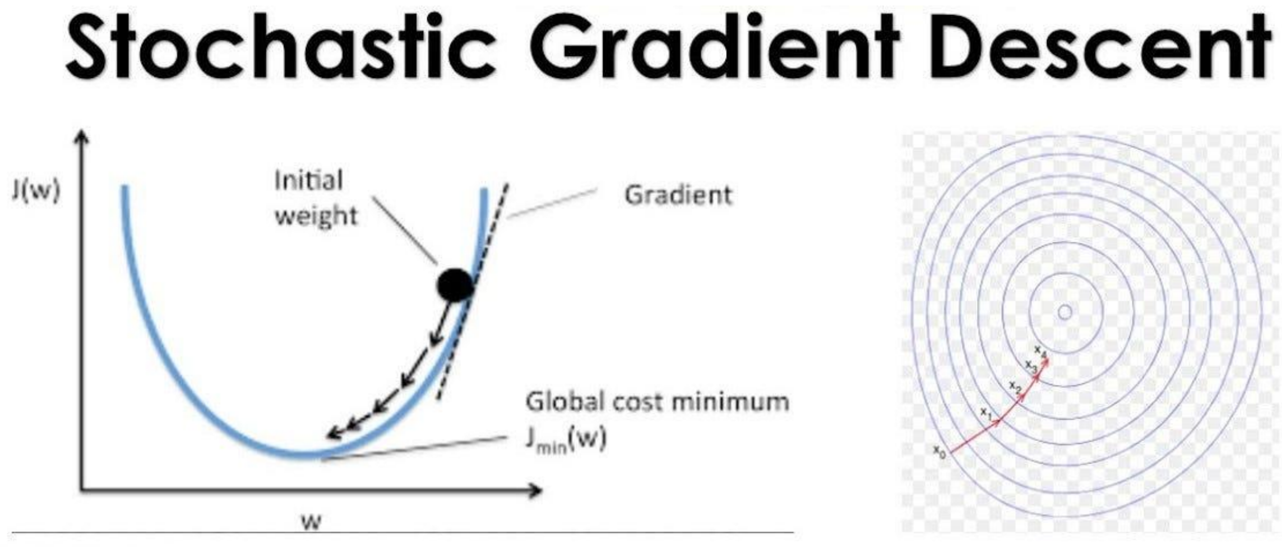


# The training loop

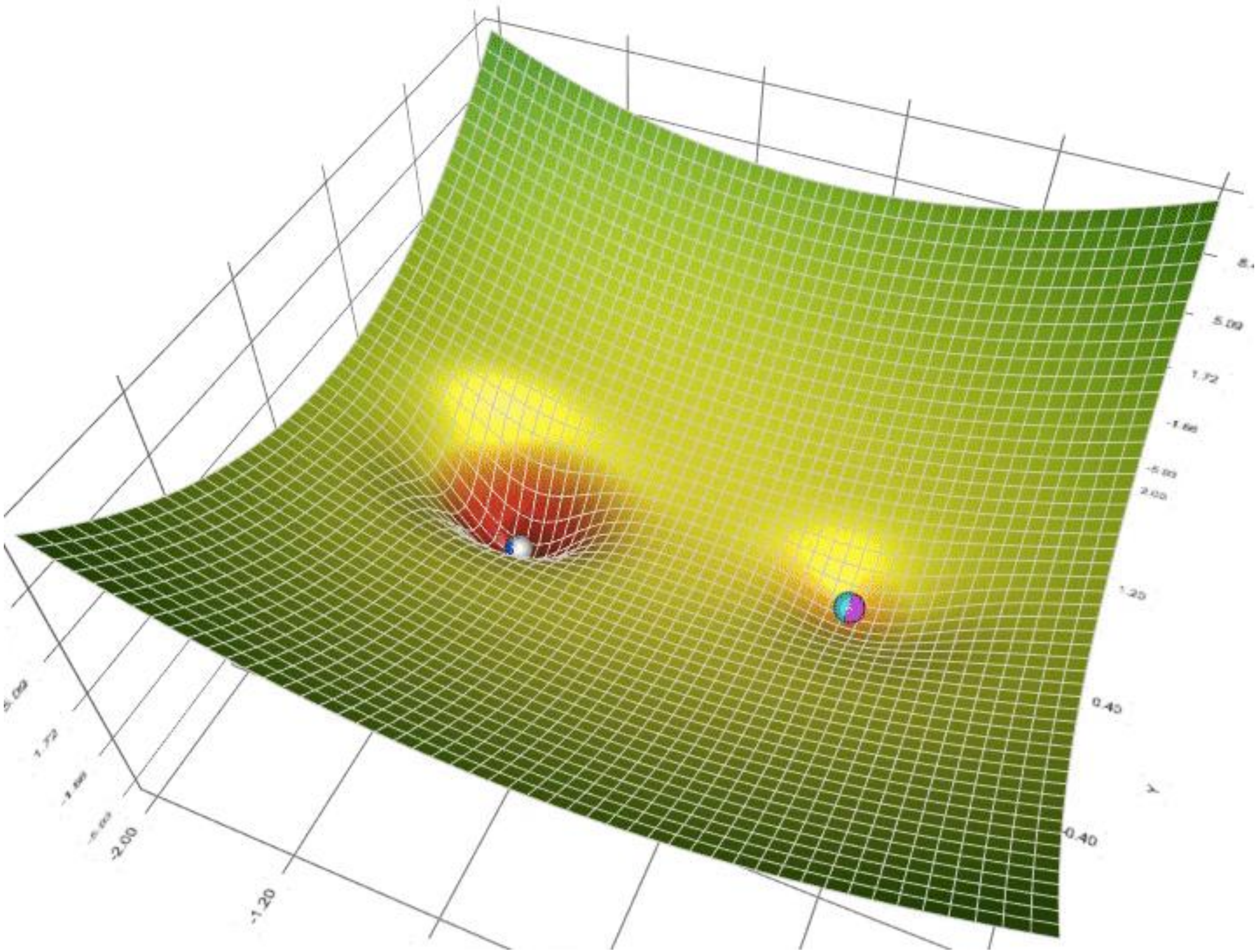
```
for epoch in 0,1, …, 100 :
    # Forward pass
    pred = model(input)

    # Loss function
    loss = loss(pred, target)

    # Backward pass
    gradients = model.backward(loss)

    # Update model weights
    model.update_weights(gradients)
```

# The training loop

```
for epoch in 0,1, …, 100 :
    # Forward pass
    pred = model(input)

    # Loss function
    loss = loss(pred, target)

    # Backward pass
    gradients = model.backward(loss)

    # Update model weights
    model.update_weights(gradients)
```



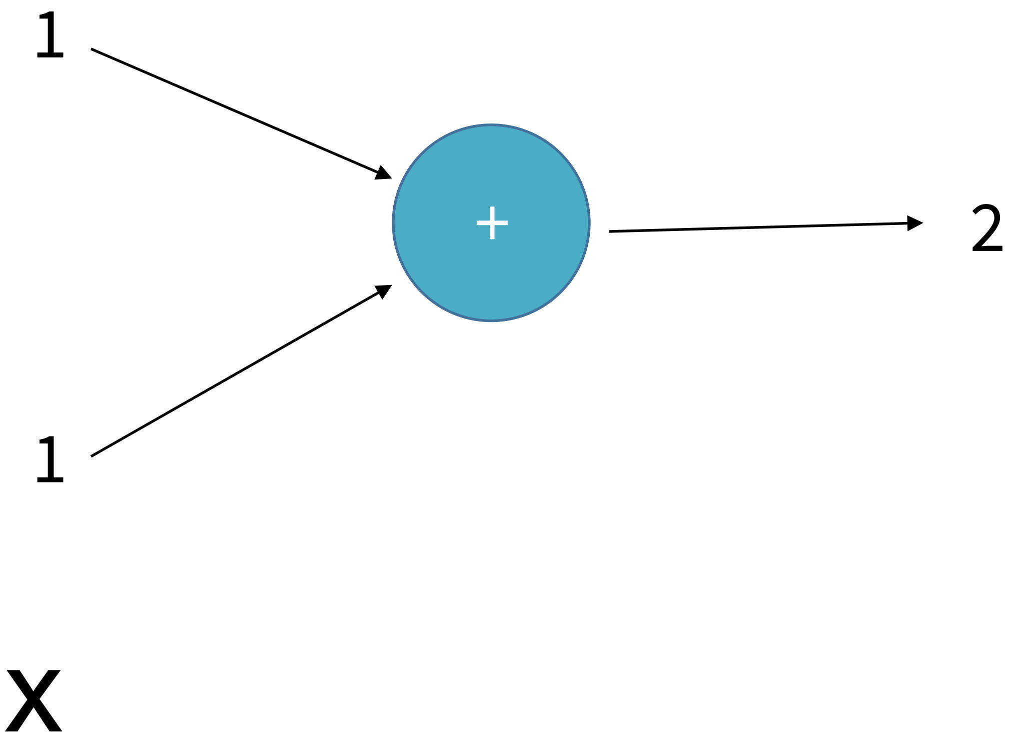
# Stochastic Gradient Descent



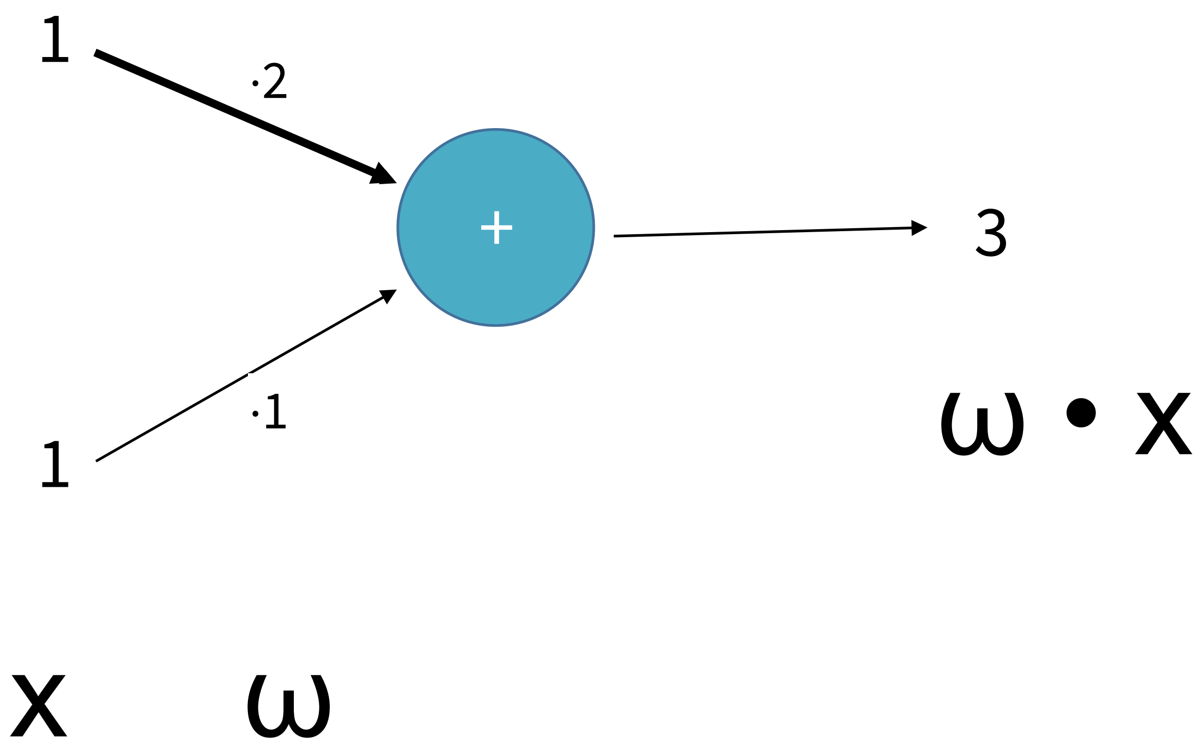
ECMWF

# Dense networks

# A Simple Neuron for Addition

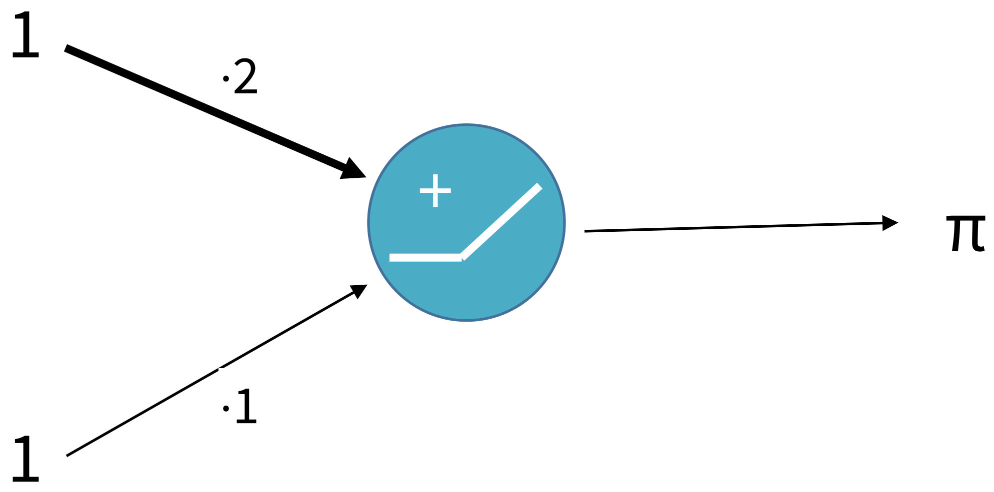


# A Simple Neuron – Changing Weights



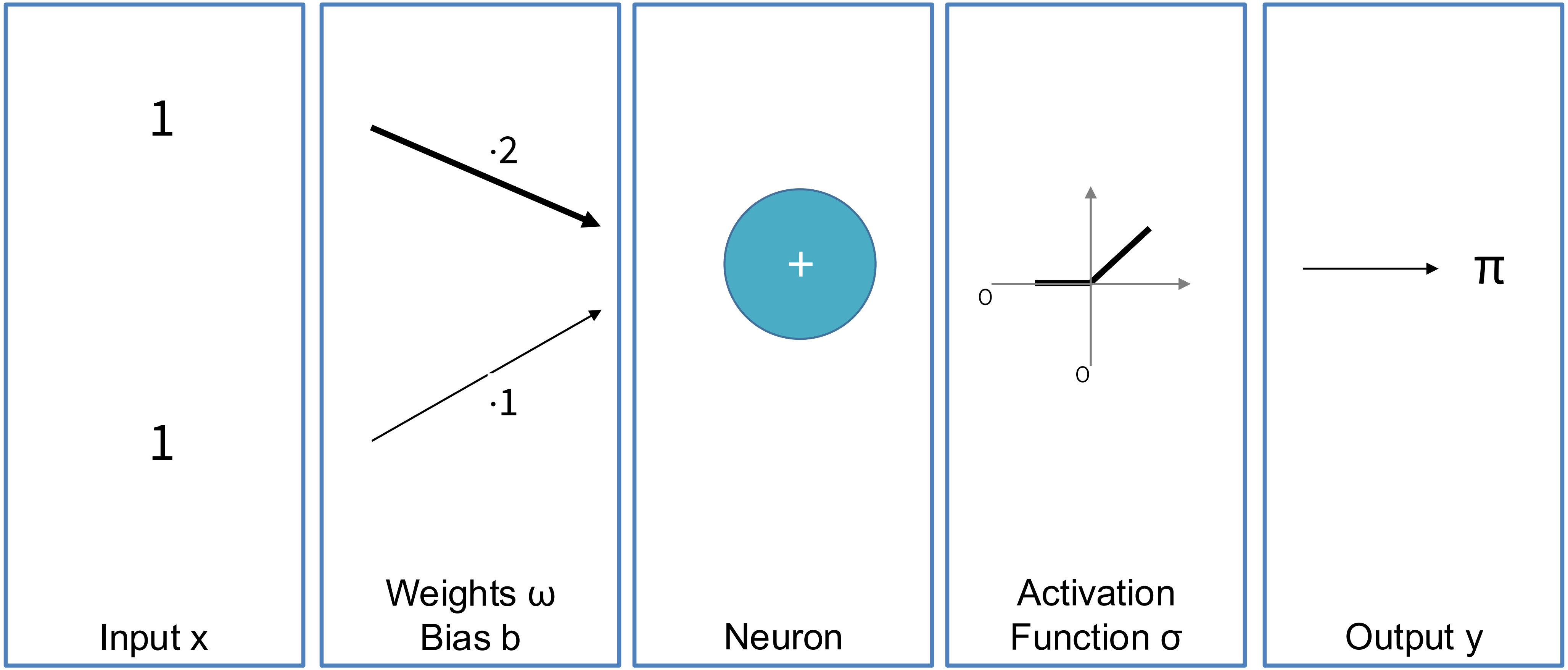

ECMWF

# A Simple Neuron – Activation Function



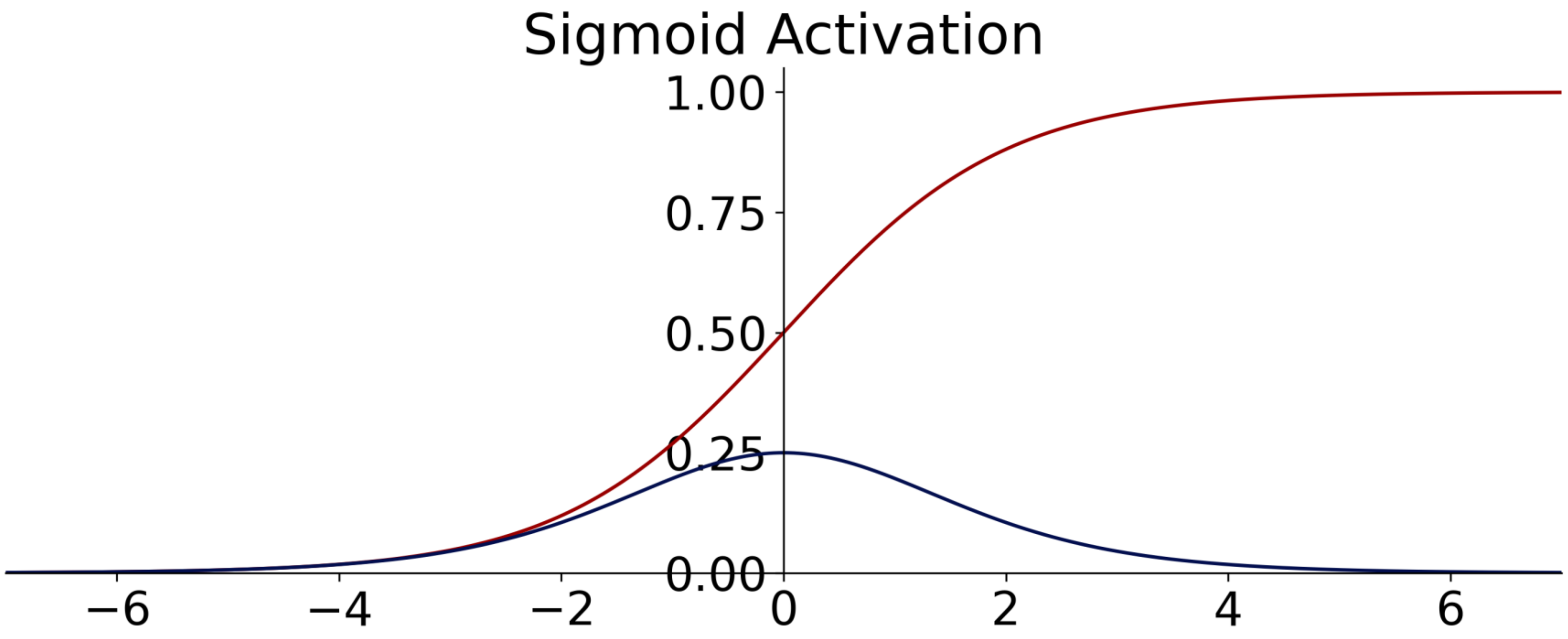

non-linear function

# A Simple Neuron – Deconstructed

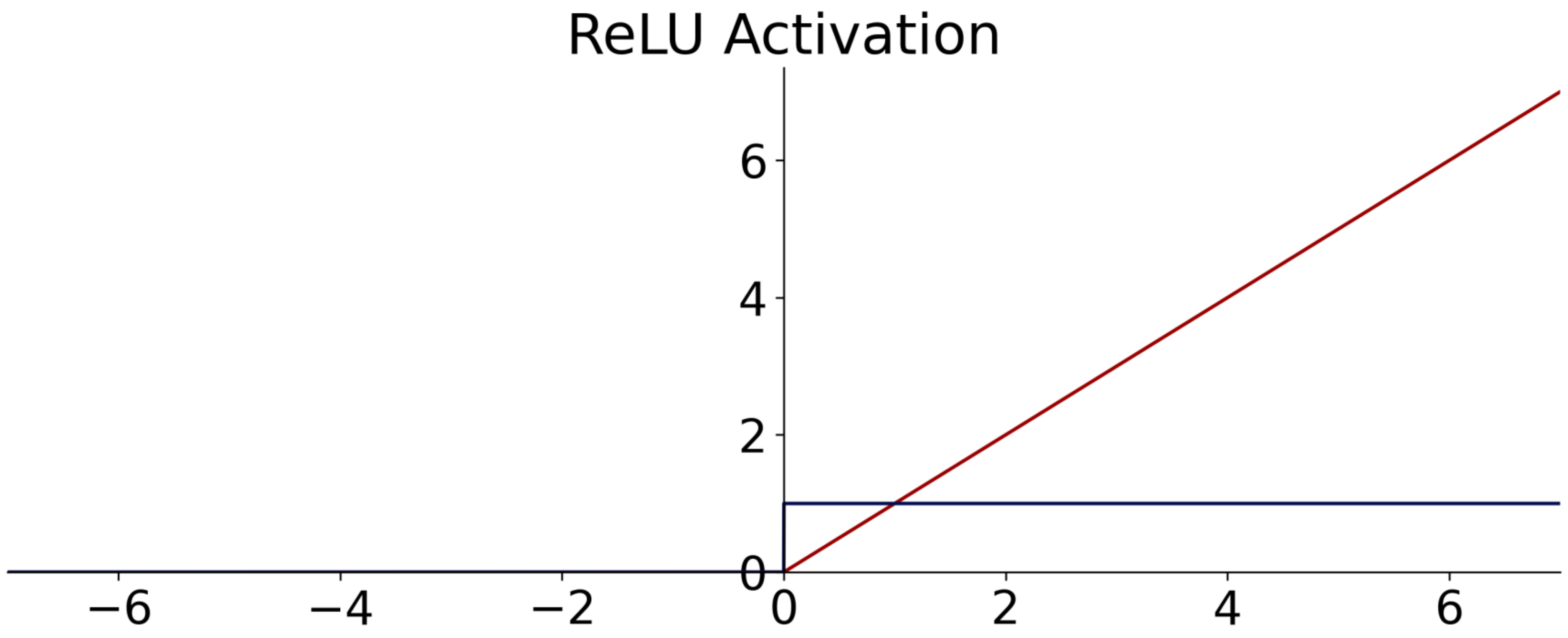



$$y = \sigma\,(\,\omega \cdot x + b)$$

# Classic Activation Function

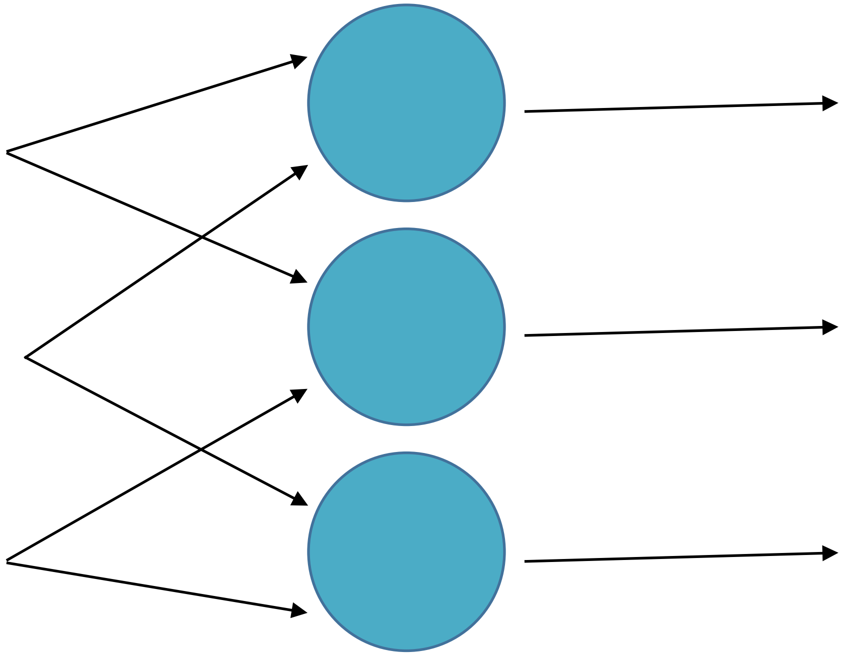


# Modern Activation Functions



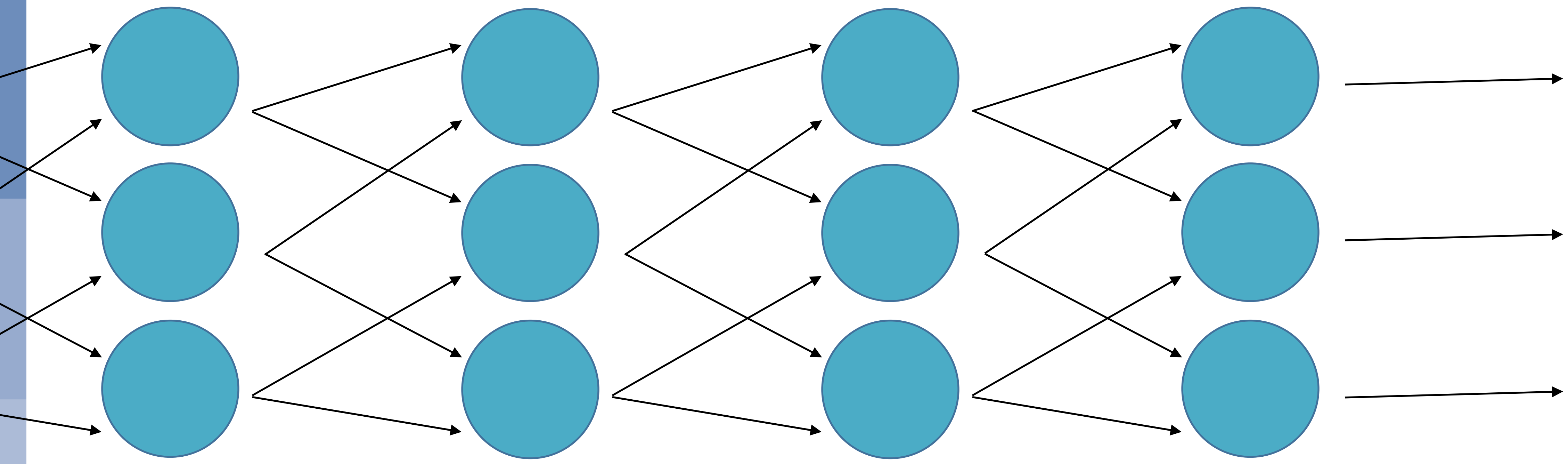

and its variations: LeakyReLU, PReLU, ELU, GELU, SELU, ...

# From the neuron to the layer

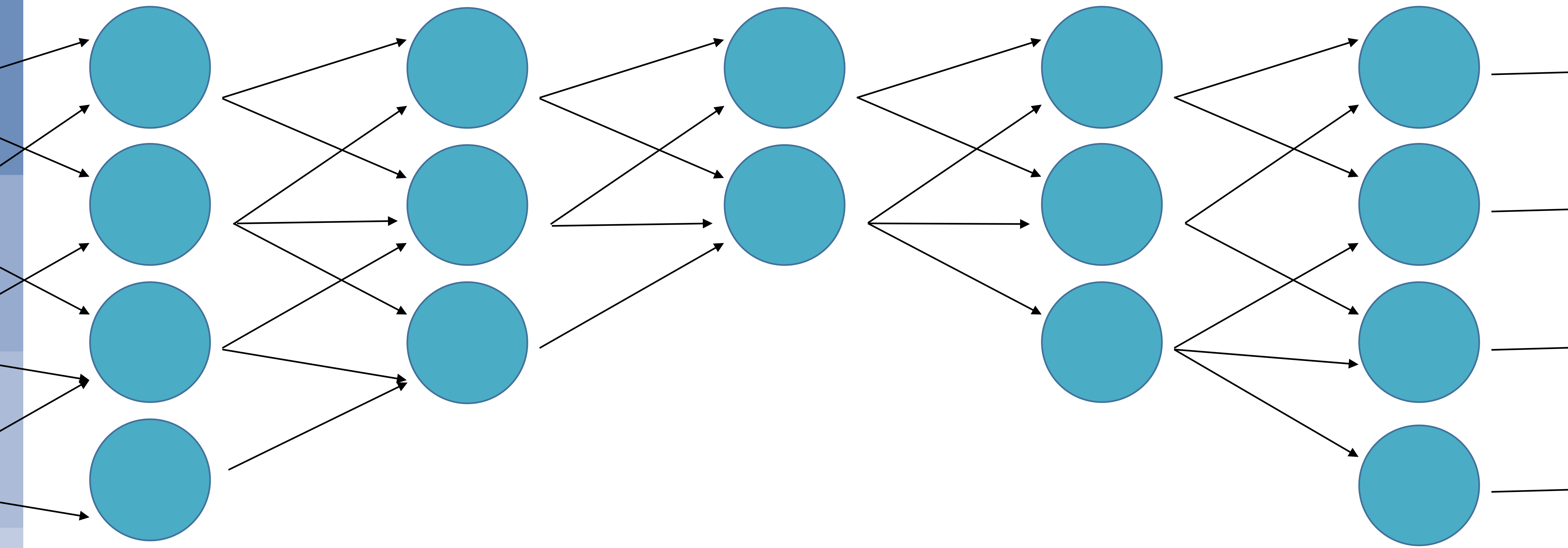


ECMWF

# A Deep Neural Network

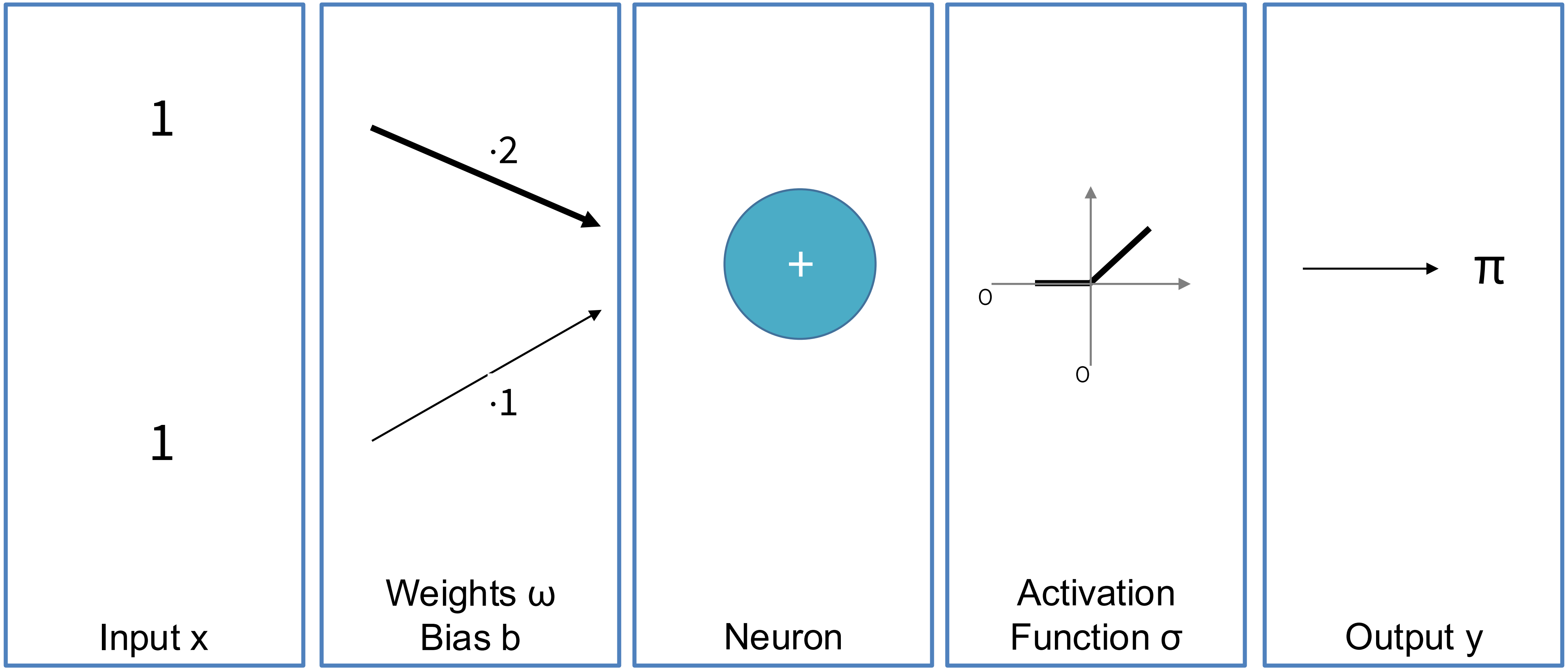


# Different Combinations In Neural Networks



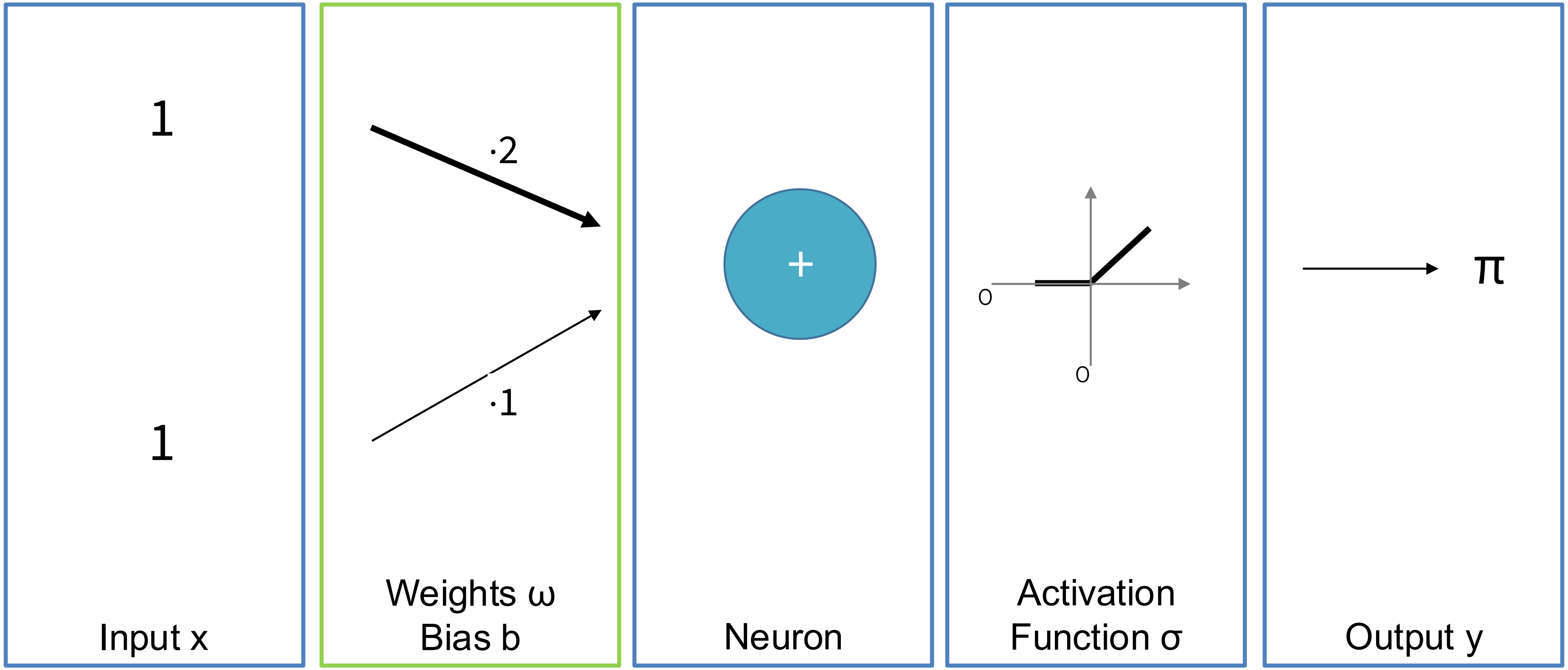
# Neural network training

# Learnable Parameters



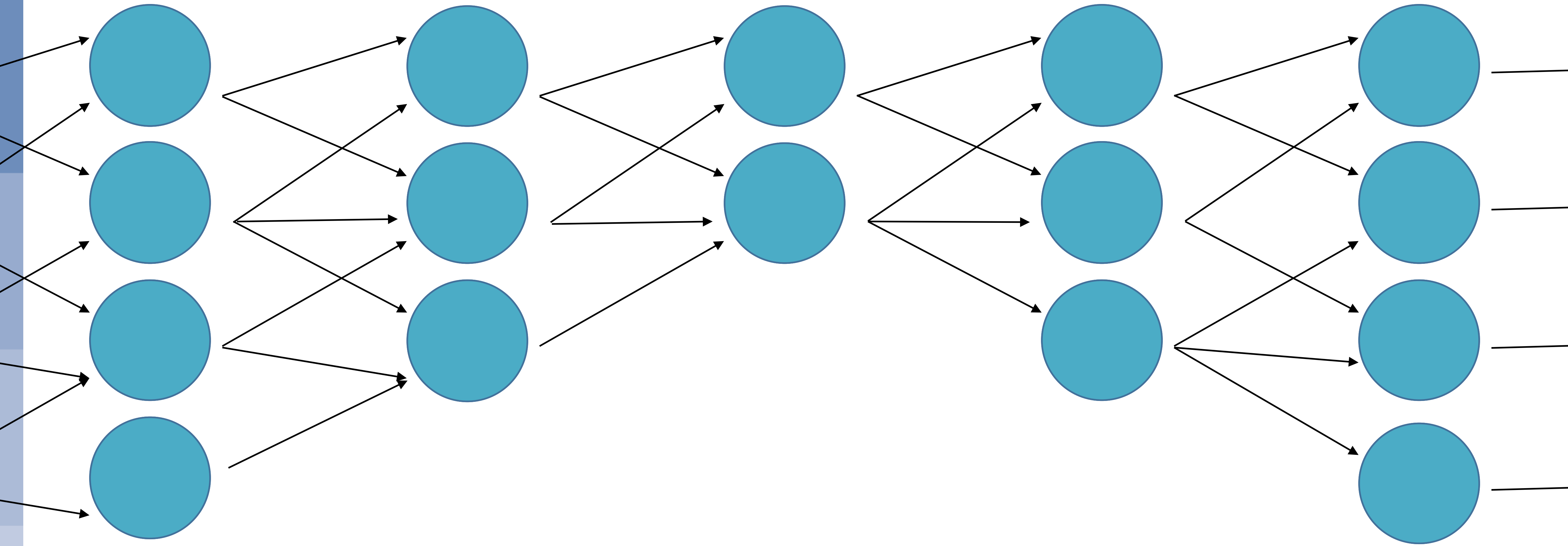

$$y = \sigma ( \omega \cdot x + b)$$

# Learnable Parameters



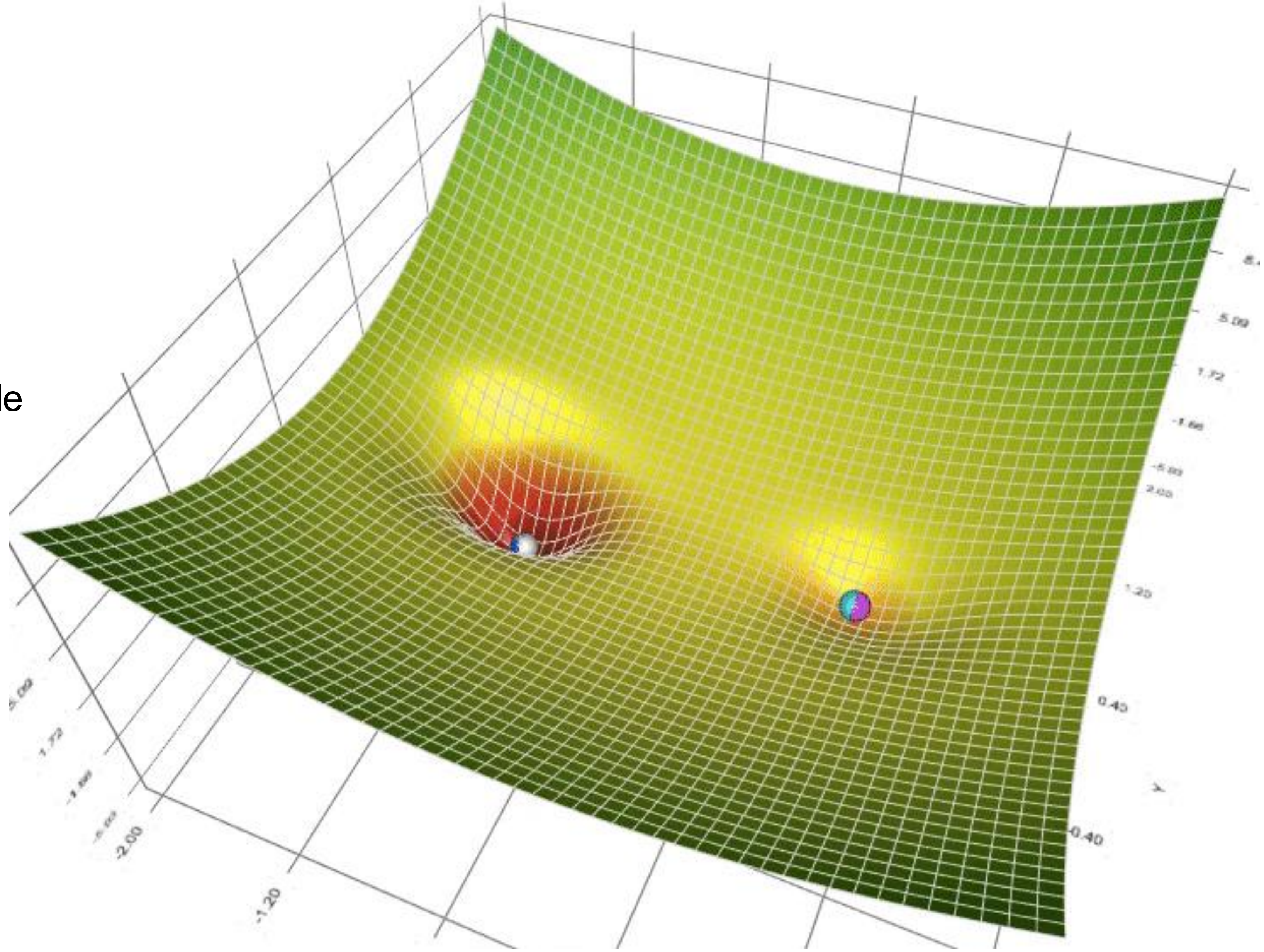

$$y = \sigma\,(\,\underline{\mathbf{W}} \bullet x + \underline{\mathbf{B}})$$

# Forward Pass In Neural Networks



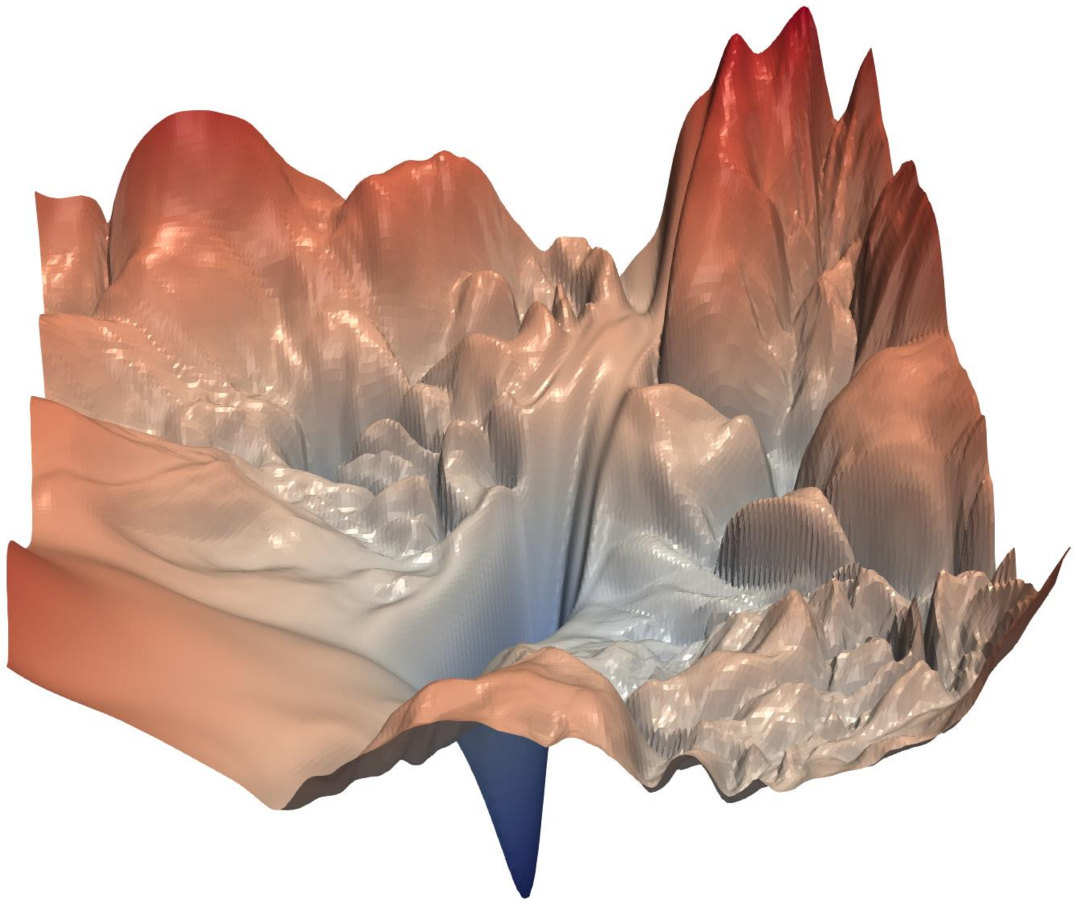
# Backward Pass with Numerical Optimization

- Calculate Error
- Stochastic Gradient Descent
- Go towards minimum
- Correct network with chain rule
- Hopefully the global minimum



# Realistic choices during training

- Loss surface usually highly irregular
- Different architectures choices change surface
- Take small steps toward minimum
- Use averaging and momentum
  - Adam optimiser
- Regularisation for better optimum



Li, et al. 2017: Visualizing the Loss Landscape of Neural Nets

# 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.

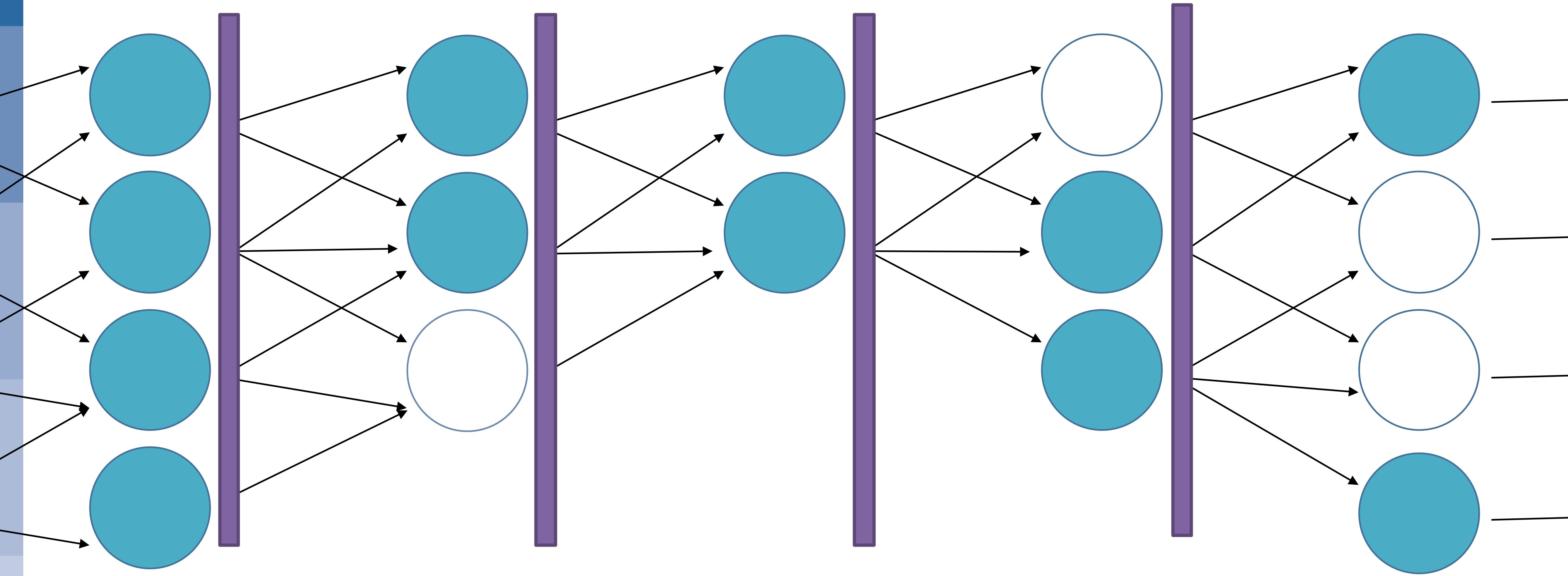
- They can learn not just patterns (but also *noise* ) if left unchecked.
- This happens when the network fits training data **too perfectly**, losing the ability to generalize to unseen data.

**Overfitting !!!**

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

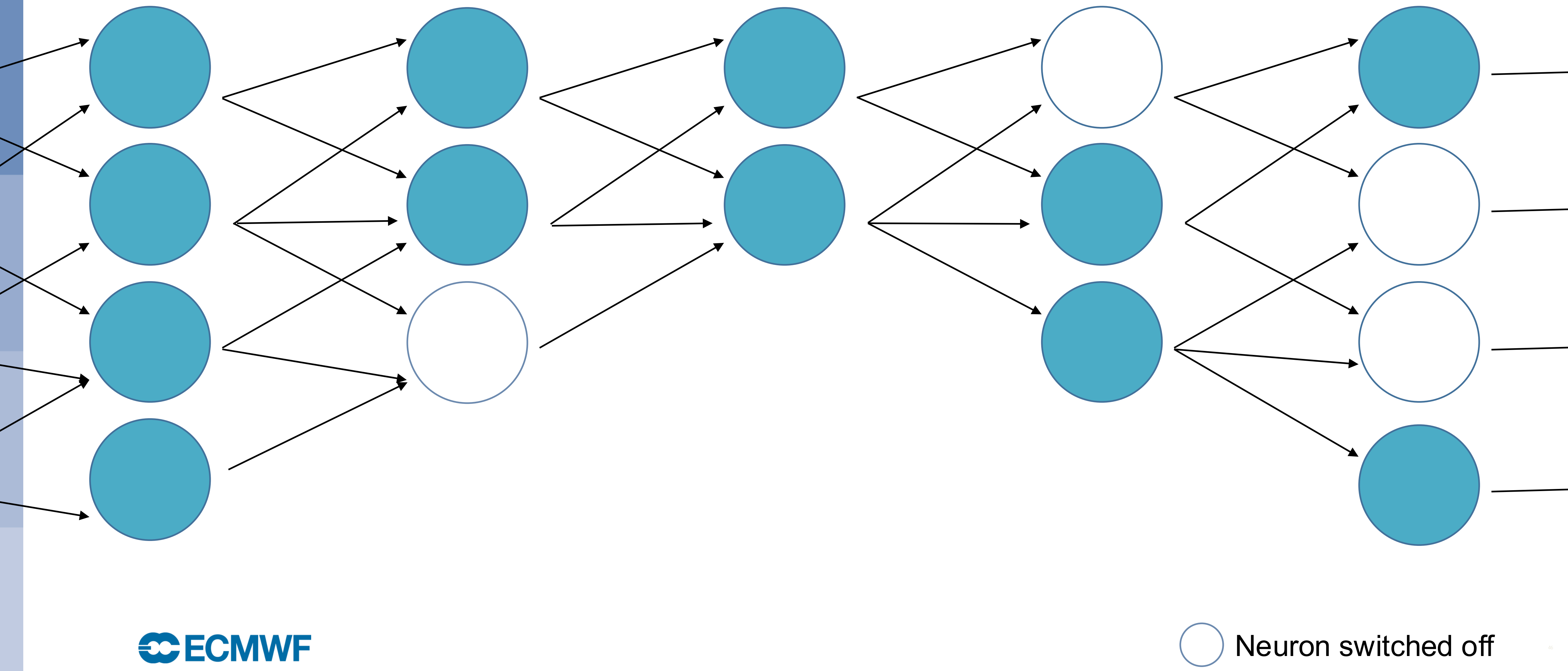
They guide the model to learn simpler, smoother, or more robust representations.

# Standardisation using BatchNormalisation



Normalise each batch of data

# Regularisation using Dropout




ECMWF

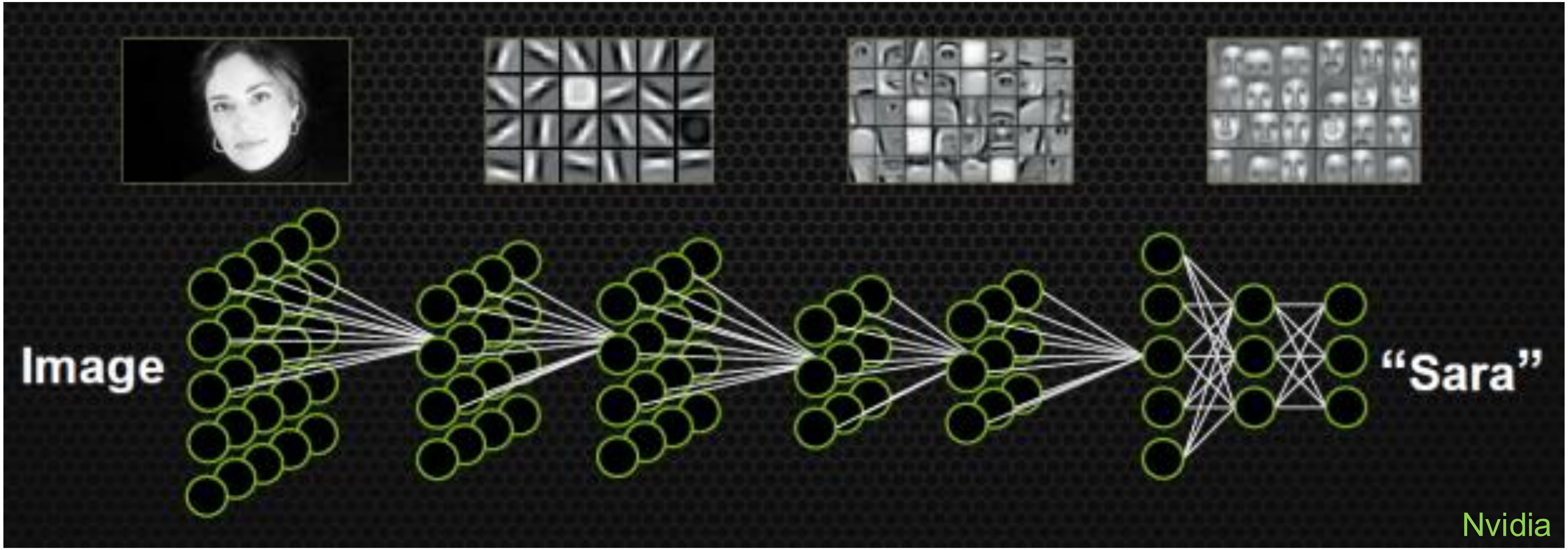
# Other

- Adapting the loss function
  - L1 regularization --> It keeps weights sparse
  - L2 regularization --> It makes weights smaller
- Training strategy
  - Early stopping
  - Data augmentation
  - Noise injection

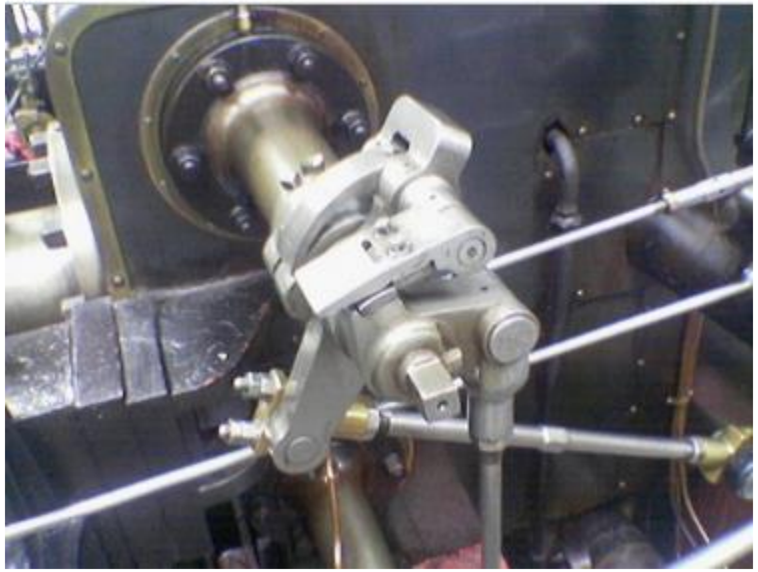
# Working with Spatial Data

Tuesday 12:00

# Networks on Images



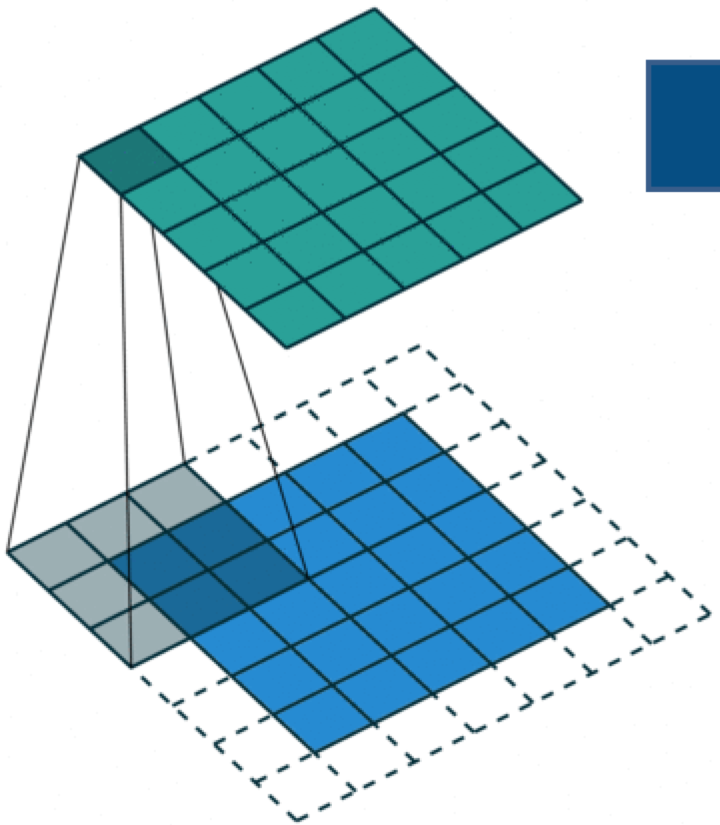
# Sobel operator



$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A}$$

# 2D Convolutions

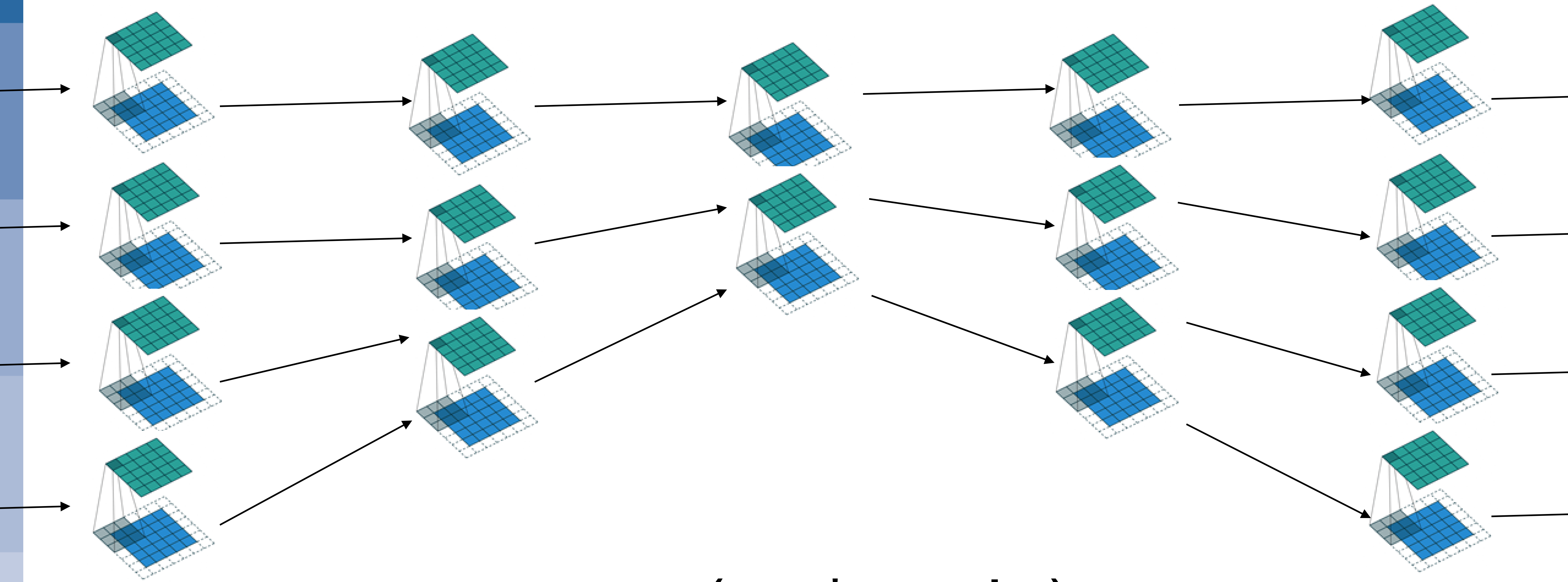


Output

Learned Convolution Filter / Kernel

Input

# Convolutional Neural Networks – Overly Simplified



$$y = \sigma ( \omega * x + b )$$

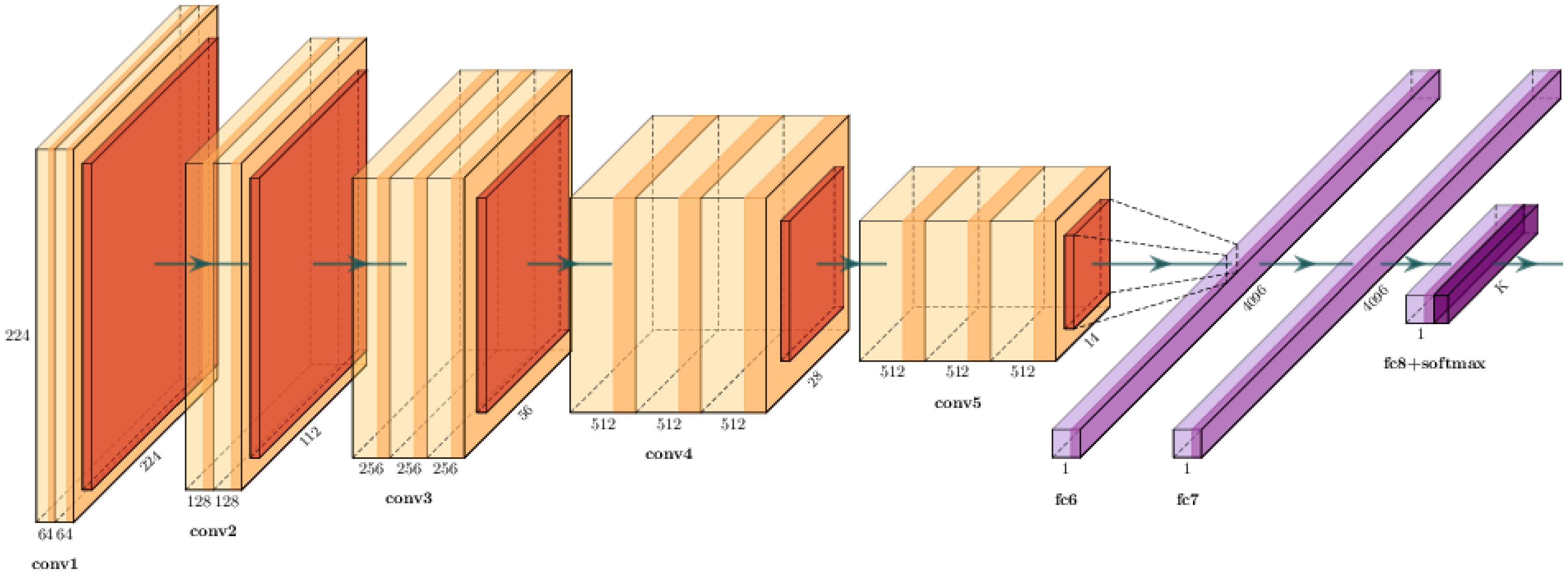
ECMWF

# Convolutional Neural Networks

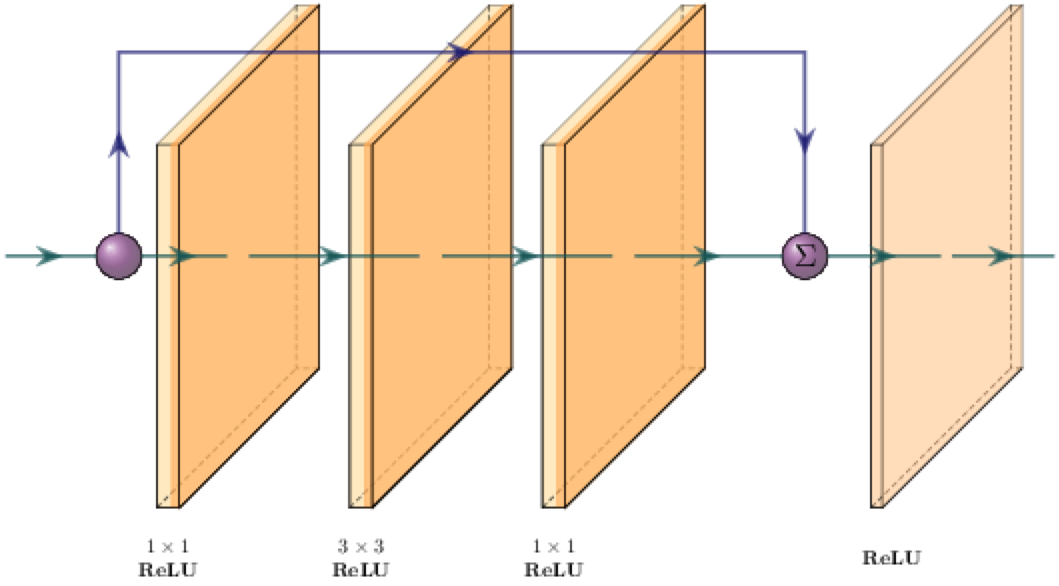
- Works with Locally Connected Data, e.g.
  - Photos
  - Satellite data
  - Weather fields
- Convolutional filters are learnt from data
- Compression changes focus of different layers
- Convolutions share weights and reduce computation

# Combining Concepts into Architectures

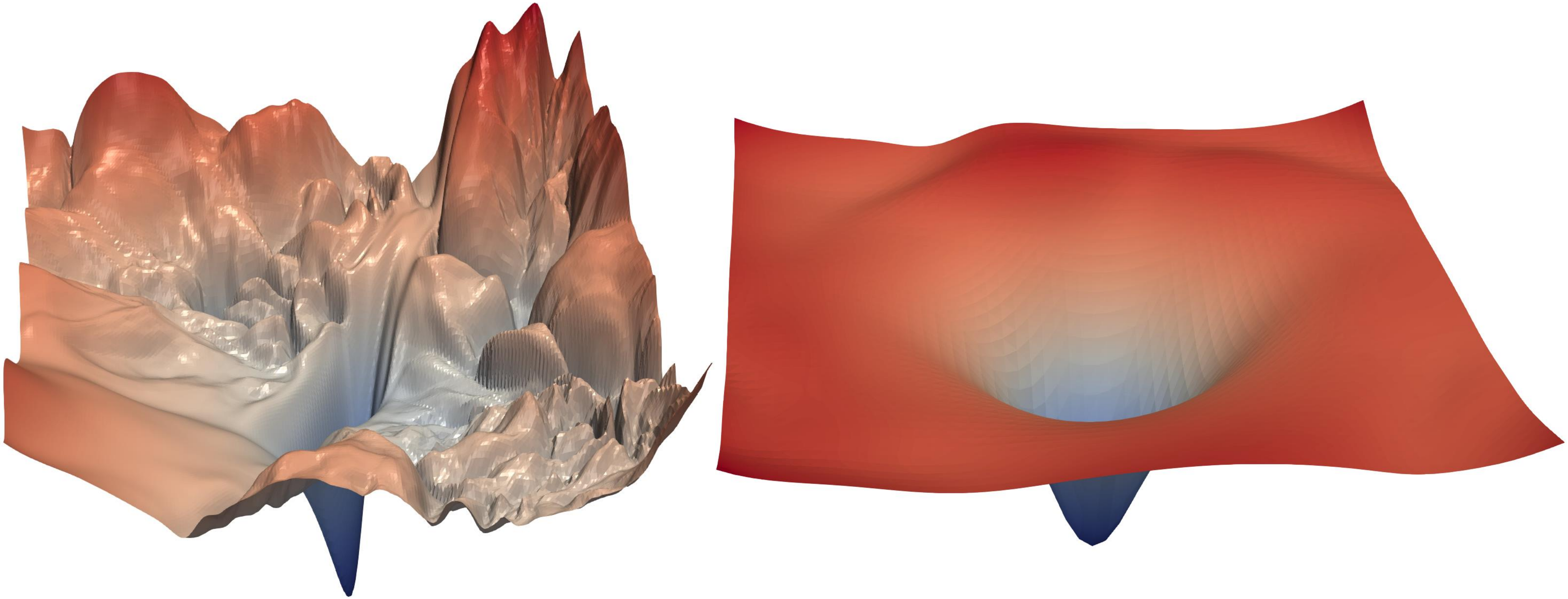
# CNN + Dense: Classification Architecture (VGGNet-16)



# ResNet Blocks: Utilizing Shortcuts

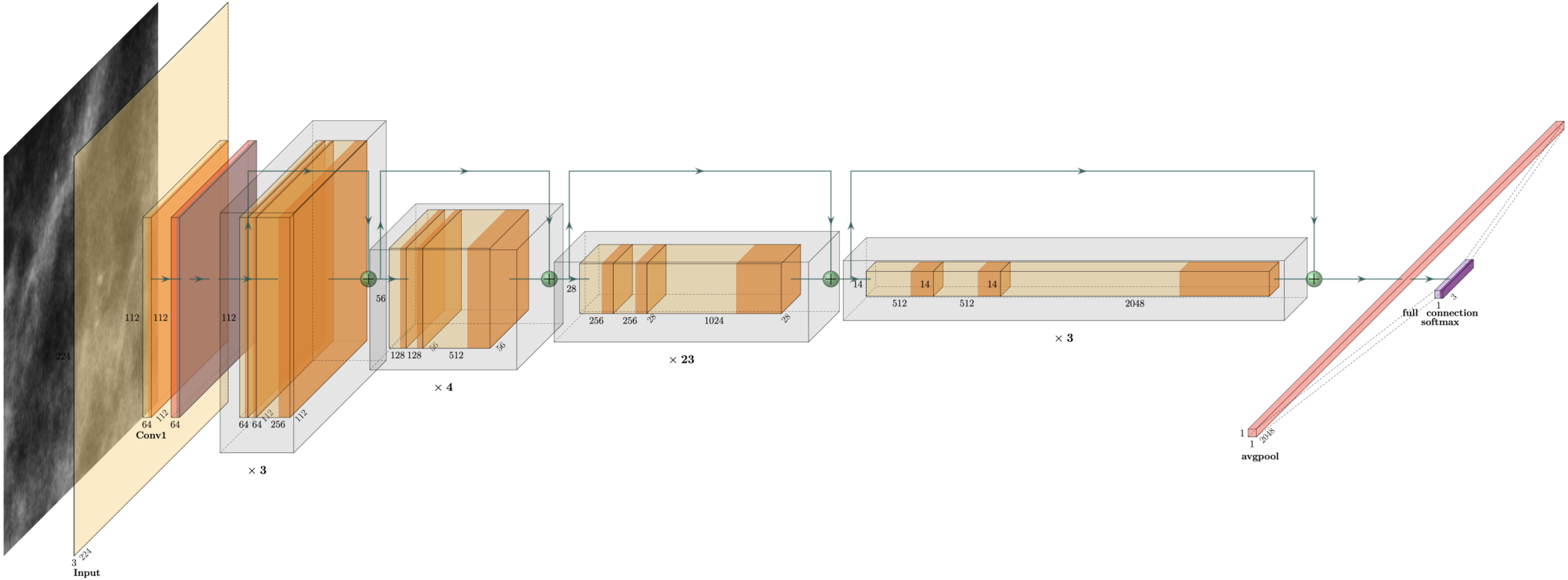


# Why we use Residual Connections

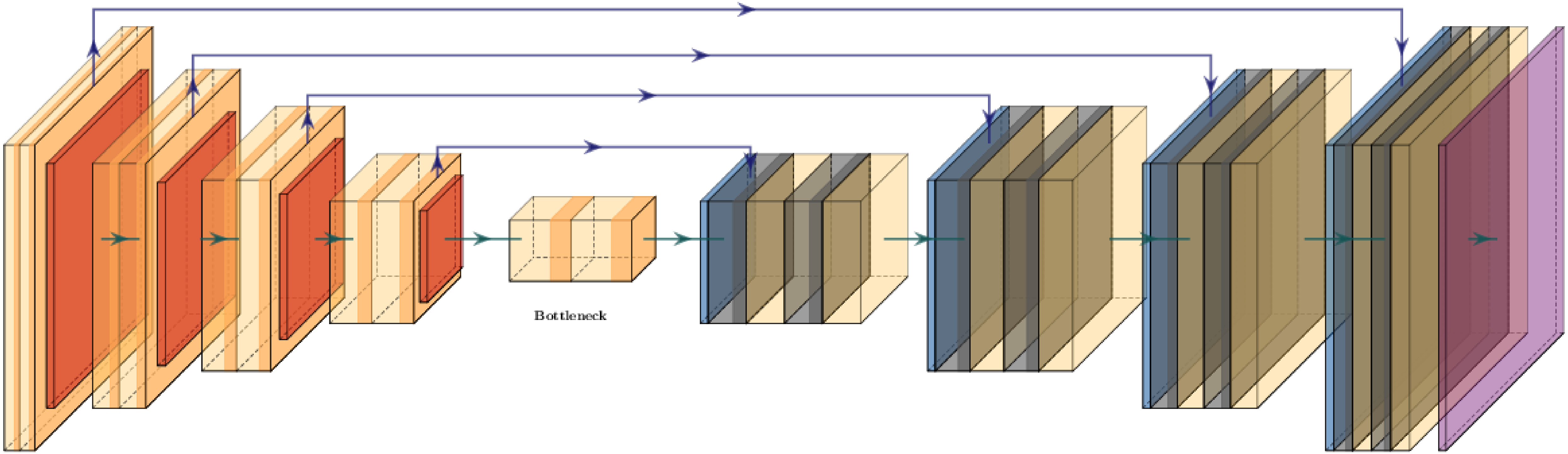


Li, et al. 2017: Visualizing the Loss Landscape of Neural Nets

# Going deep: ResNet-101



# Unet: Utilizing Compression for Encoding / Decoding




Encoder

Bottleneck

Decoder

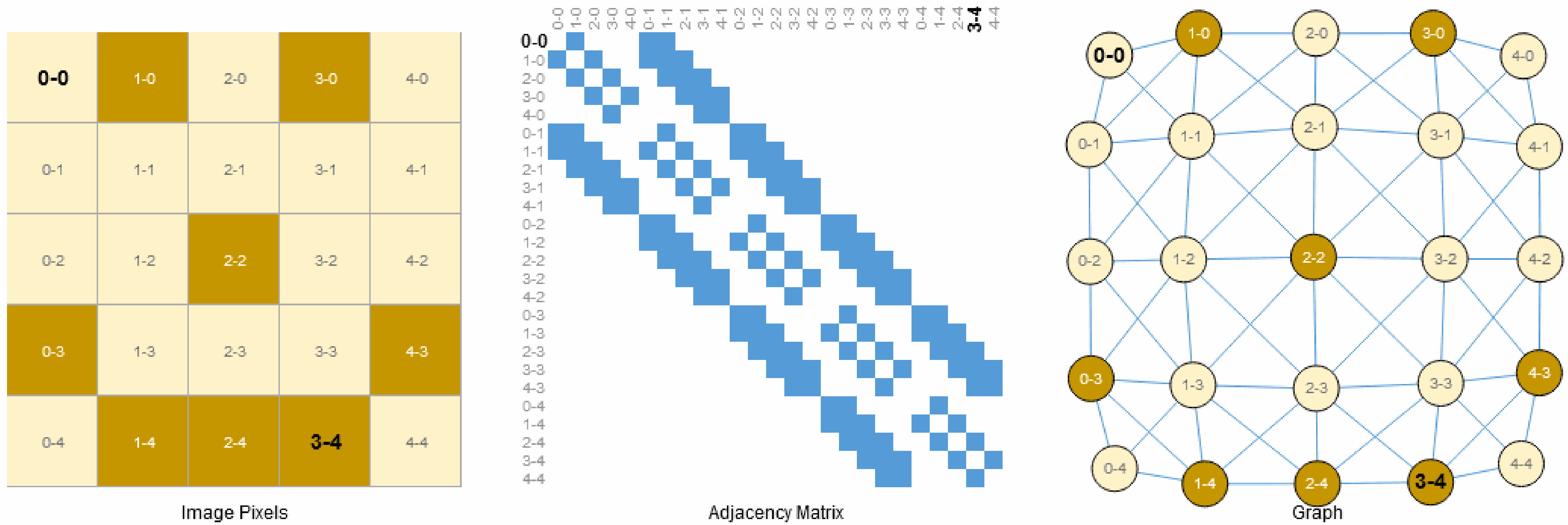
# Why we use Compression / Latent Spaces


"Compression forces understanding."

# Graph Neural Networks

Wednesday 12:00

# Defining Operations on Graphs

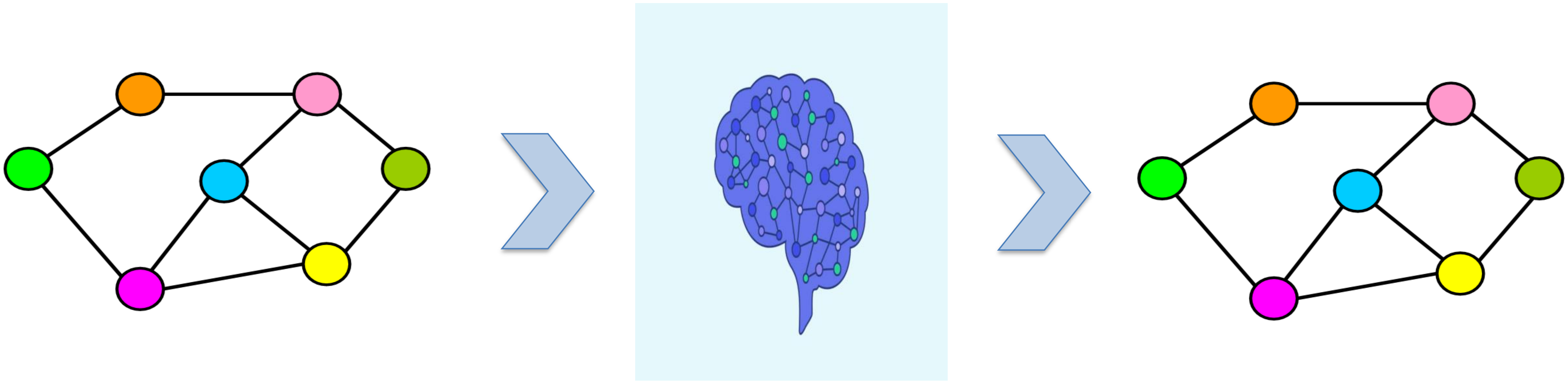



Click on an image pixel to toggle its value, and see how the graph representation changes.

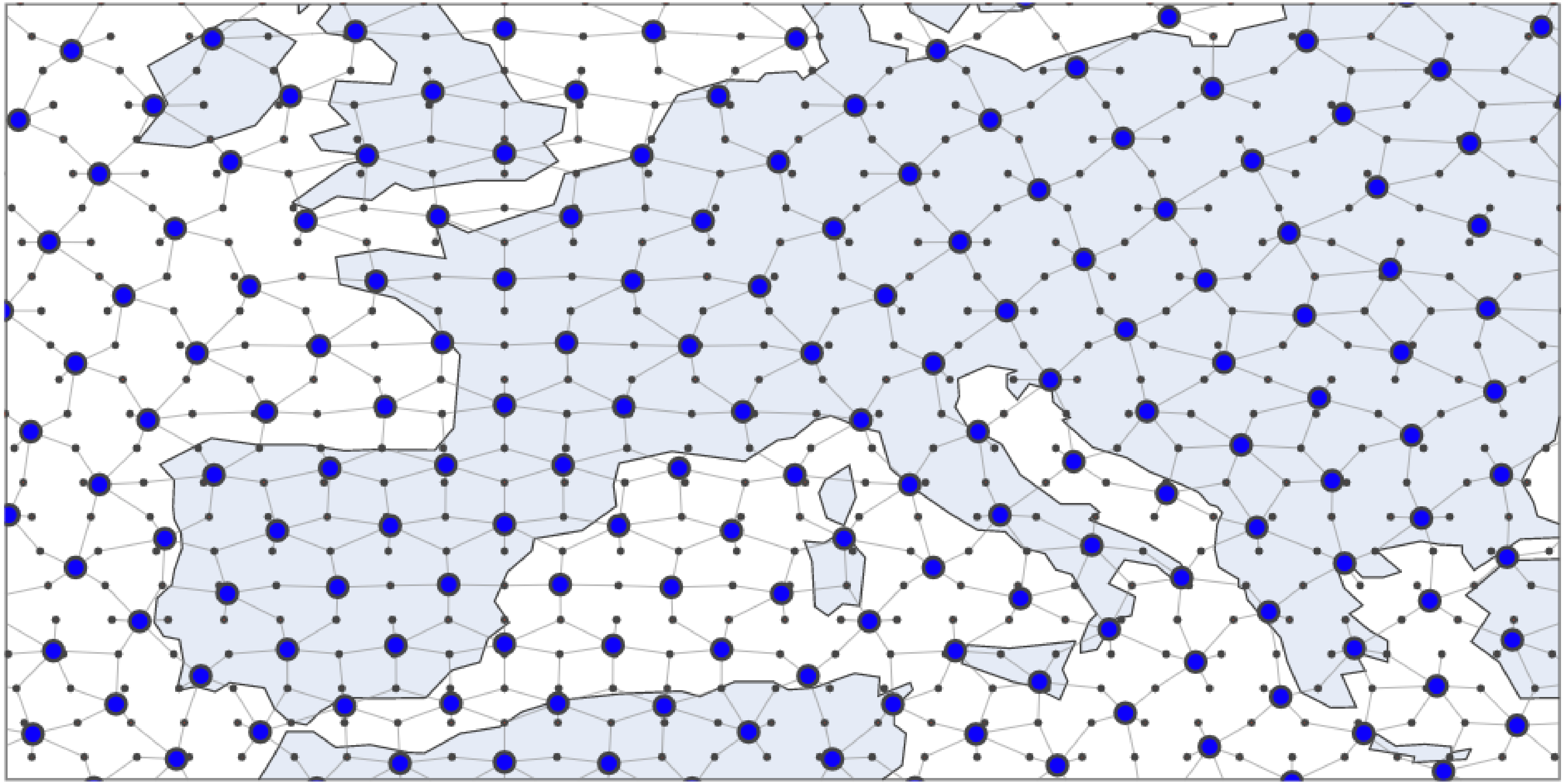


Sanchez-Lengeling, et al., "A Gentle Introduction to Graph Neural Networks", Distill, 2021.

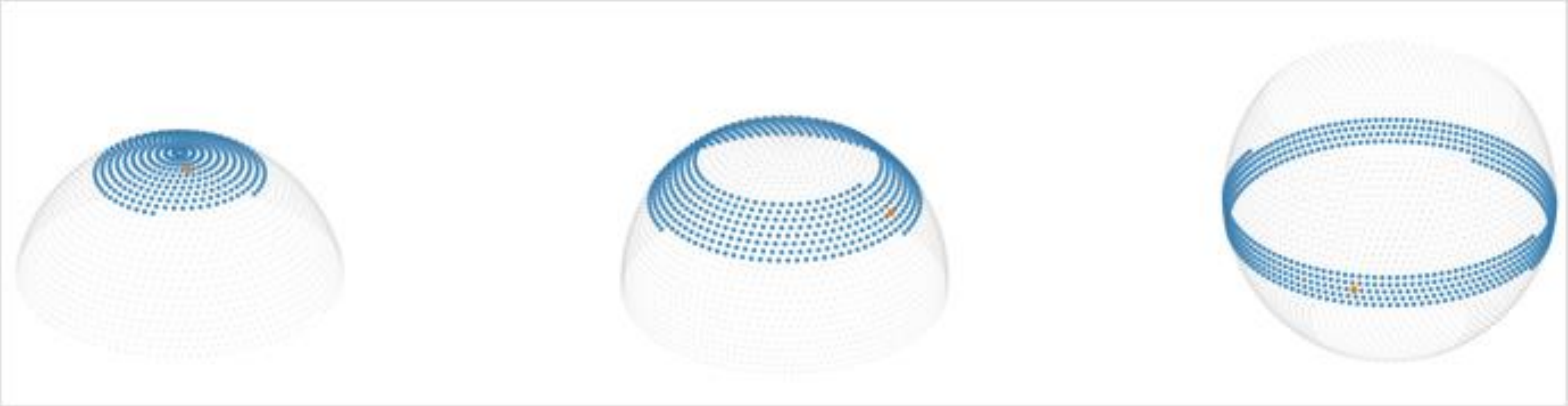
# Graph Neural Networks



# Defining Operations on Graphs: Convolutions



# Defining Operations on Graphs: Transformers



AIFS



https://www.ecmwf.int/en/about/media-centre/aifs-blog/2024/first-update-aifs

# Conclusion

# What We Learned

- Neural Network Training
- Network Types
  - Dense Neural Networks
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Transformers
  - Graph Neural Networks
- Example Architectures
- Compression
- Shortcuts / Residual Connections

# Questions