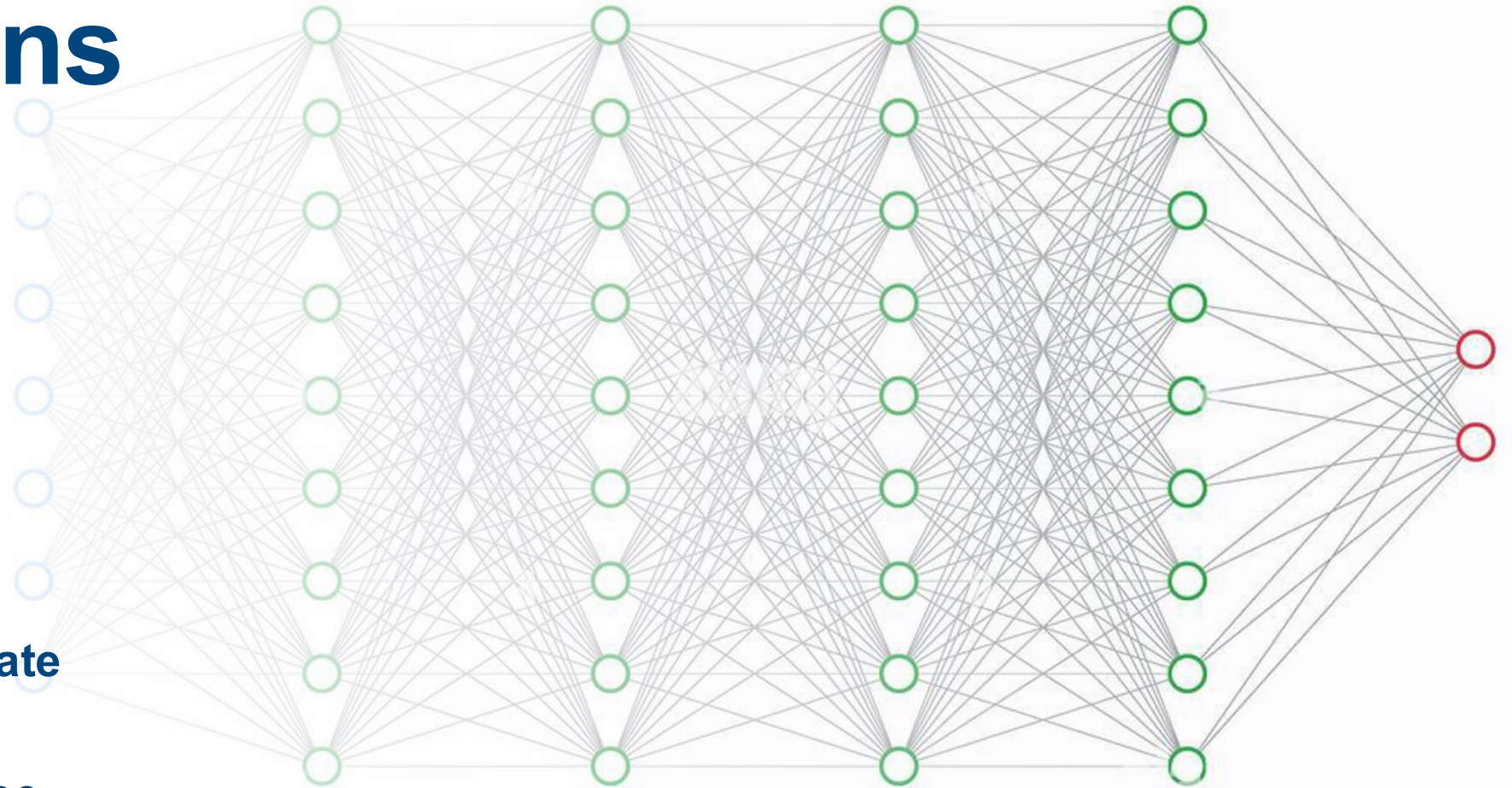


Architectures and Applications



Deep Learning for Climate Scientists

22nd-24th September 2026

Paul Keil paul.keil@hereon.de

Who we are: Helmholtz AI Consultants



Tobias Weigel



Caroline Arnold



Danu Caus



Harsh Grover



Paul Keil



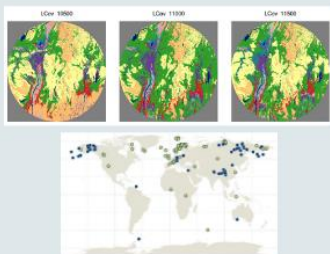
Andreas Beckert

- **AI consulting for Earth and Environment @ Helmholtz-Zentrum Hereon.**
- **Helmholtz AI offers AI implementation and method support for all researchers in Helmholtz, free at the point of use.**
- **The Voucher system: <https://voucher-system.helmholtz.ai/>**
- **Our Expertise and Interest:**
 - ML-GCM Coupling
 - AI for remote sensing,
 - LLMs for Science
 - Downscaling
 - ...

Email: consultant-helmholtz.ai@hereon.de

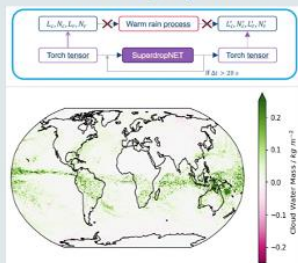
Helmholtz AI: Some of our Projects

FOUNA: Towards a deep learning/foundation model for biodiversity and nature conservation (AWI)



Danu, Tobias

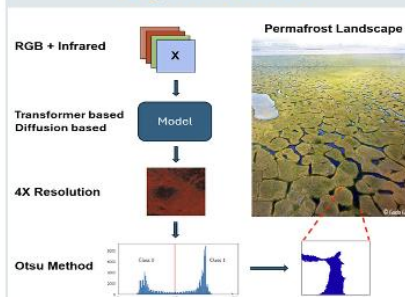
Coupling ICON with a ML Emulator for Cloud Microphysics



ICON coupled with SuperdopNET produces more cloud water mass

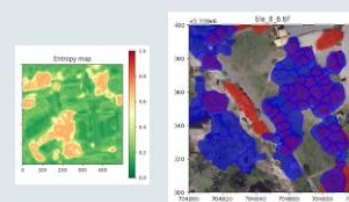
Caroline, Paul

Super-resolution of aerial images from permafrost regions (AWI)



Danu, Harsh

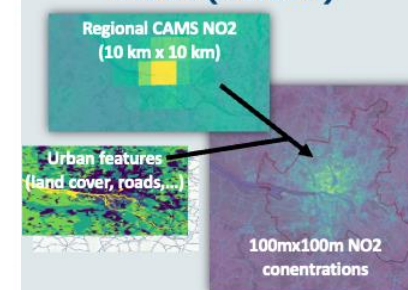
DeepTrees: Individual Tree Crown Delineation from Digital Orthophotos (UFZ)



Active Learning based on entropy

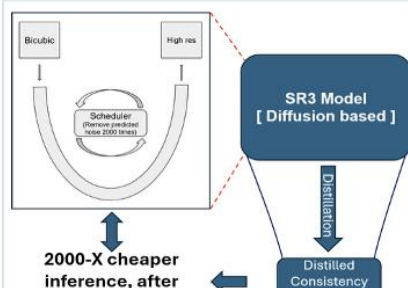
Caroline, Harsh

UrbanXACT: Downscale regional air quality concentrations for urban areas (Hereon)



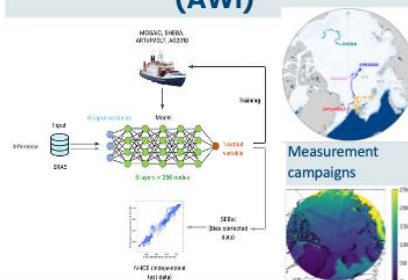
Paul, Tobias

Model distillation via consistency models (GFZ)



Danu

Correcting Arctic surface energy budget using ANNs (AWI)

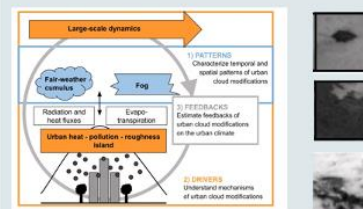


Workflow: NN trained on measurements (point locations)

Application of NN to whole Arctic

Paul, Harsh

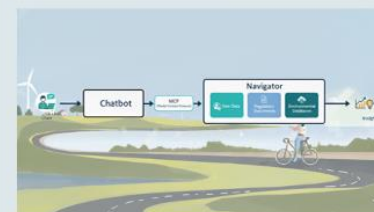
Satellite-based detection of urban cloud holes (KIT)



CNN for binary classification: 85% F1 Score

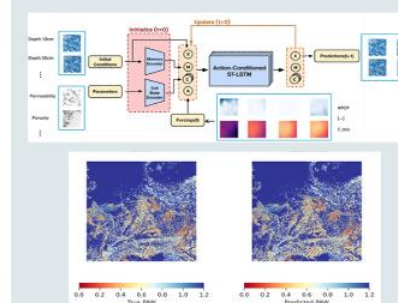
Caroline

Sea2Land navigator turned chatbot (GERICS)



Harsh, Tobias

Drought Analytics (FZJ)



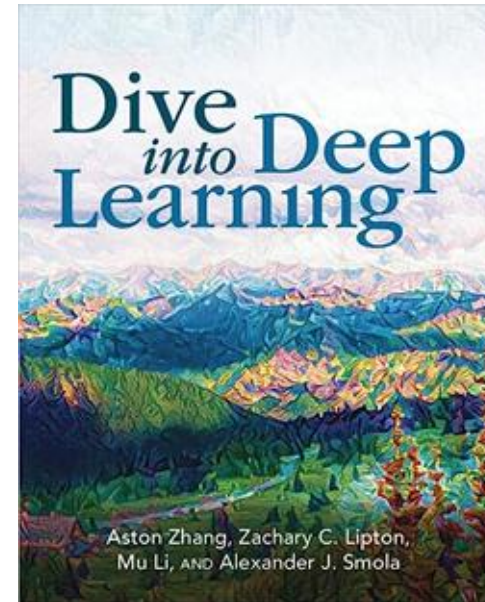
Caroline, Paul

What to expect from this lecture

- A look into the toolbox
- Some math
- It might be overwhelming
- Examples from weather and climate science

Resources

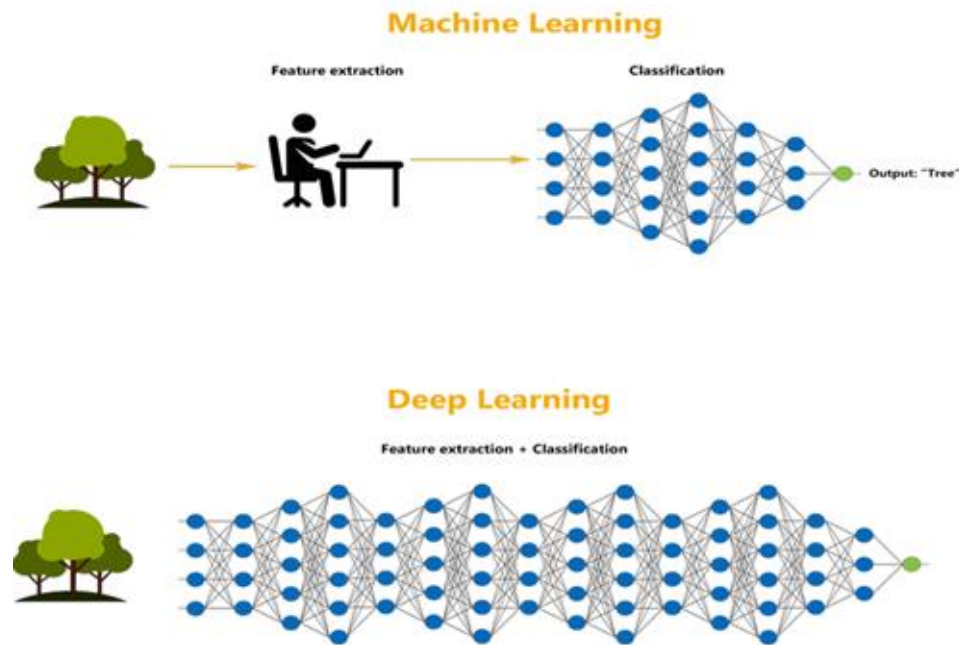
- Dive into Deep Learning:
<https://d2l.ai/index.html>
- youtube
- <https://towardsdatascience.com/>
- medium.com
- papers



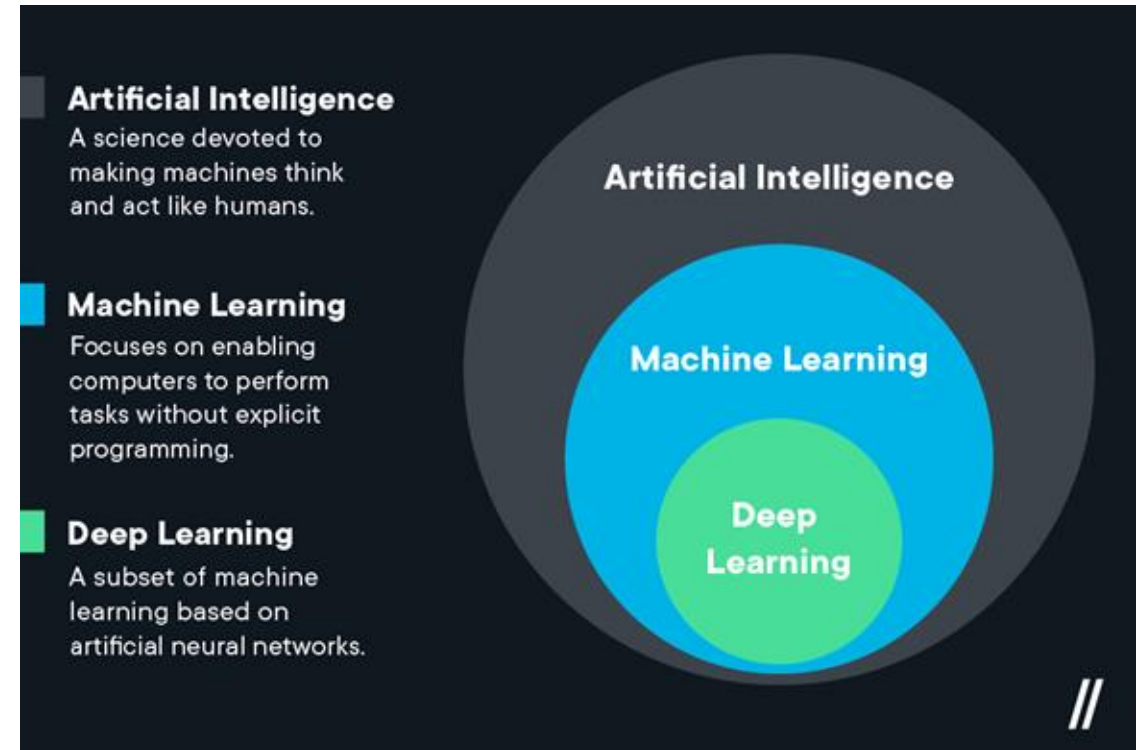
Outline

- **Deep Learning Architectures:**
 - Multilayer Perceptron
 - Convolutional Neural Networks
 - Recurrent Neural Networks
 - Transformers
 - Graph Neural Networks
 - Autoencoders
- **Probabilistic Deep Learning**
 - Bayesian Neural Networks
 - Deep Ensembles
- **Physics-Informed Deep Learning/ Hybrid Approaches**
- **Explainable AI**

Machine Learning (but not deep learning)



<https://www.baeldung.com/cs/machine-learning-vs-deep-learning>

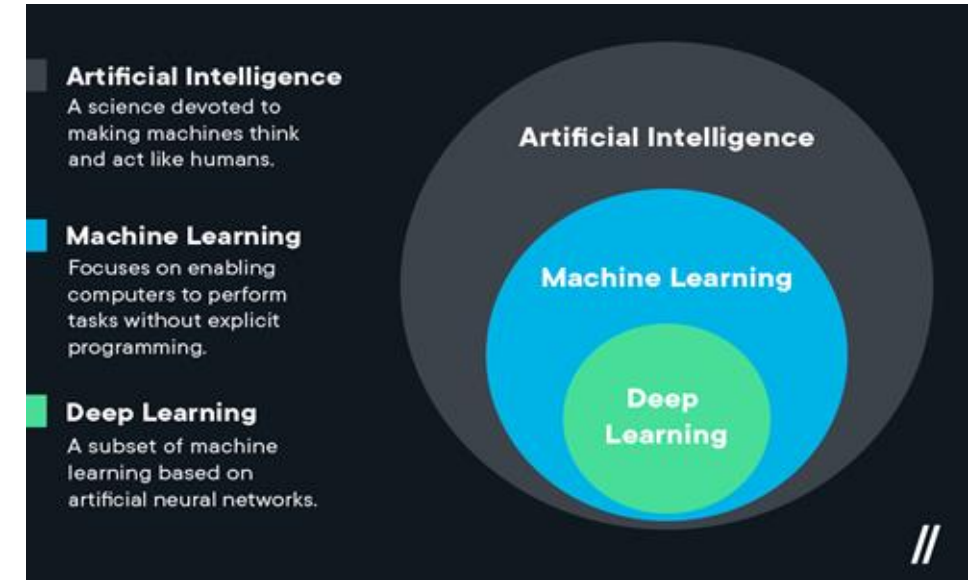


<https://flatironschool.com/blog/deep-learning-vs-machine-learning/>

Machine Learning (but not deep learning)

- **Linear Regression**
- **Support Vector Machines (SVM)**
- **Random Forests**
- **Principal Components Analysis (PCA)**
- **Xgboost**
- ...

For many tasks, these algorithms are adequate and powerful.



<https://flatironschool.com/blog/deep-learning-vs-machine-learning/>

Multilayer Perceptrons

Multilayer Perceptrons

The “standard” neural net

- “fully connected layers”, or “dense layers”
- calculate hidden state H using weights (W) and biases (b) and activation function

W : weight matrix

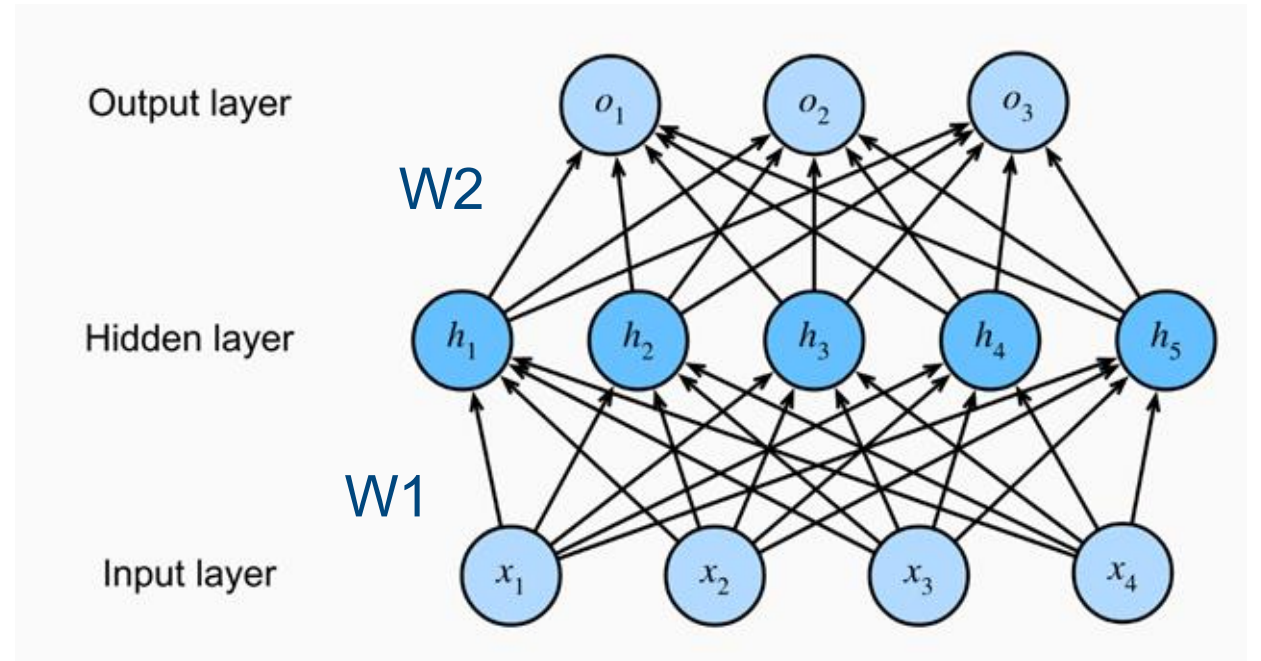
b : biases

σ : nonlinear function

X : input vector

H : hidden or “latent” state

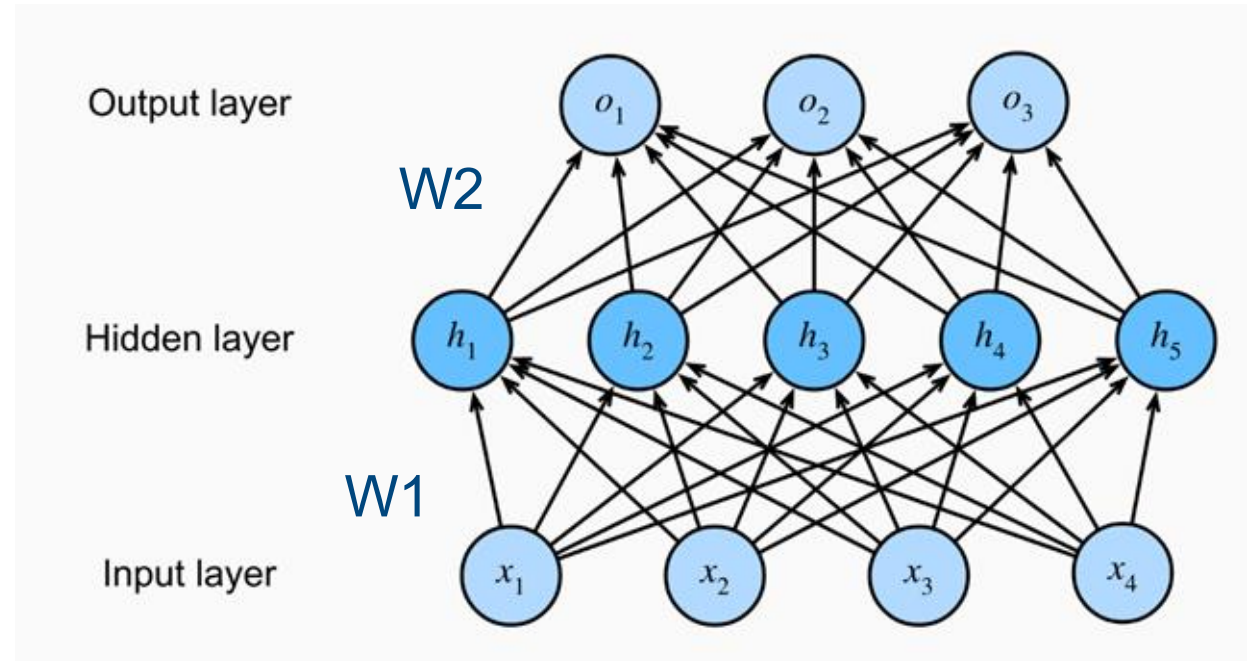
O : output



$$\mathbf{H} = \sigma(\mathbf{XW}^{(1)} + \mathbf{b}^{(1)}),$$
$$\mathbf{O} = \mathbf{HW}^{(2)} + \mathbf{b}^{(2)}.$$

Multilayer Perceptrons

- **Weights and biases** are the parameters that are “learned”
- **Dimensionality:**
 - input x has “4 features”
 - One sample has dimension 1×4
 - Typically, multiple samples are grouped together to a “batch” for improved efficiency
 - For batch size 32 input x has dimensions 32×4
- **The number of hidden layers and the amount of neurons can be chosen**
- **Deep Learning is mostly just linear Algebra and some Calculus.**



$$\mathbf{H} = \sigma(\mathbf{XW}^{(1)} + \mathbf{b}^{(1)}),$$
$$\mathbf{O} = \mathbf{HW}^{(2)} + \mathbf{b}^{(2)}.$$

Activation Functions

“Add non-linearity”. Neurons are activated (“they fire”) like brain neurons

- **ReLU**

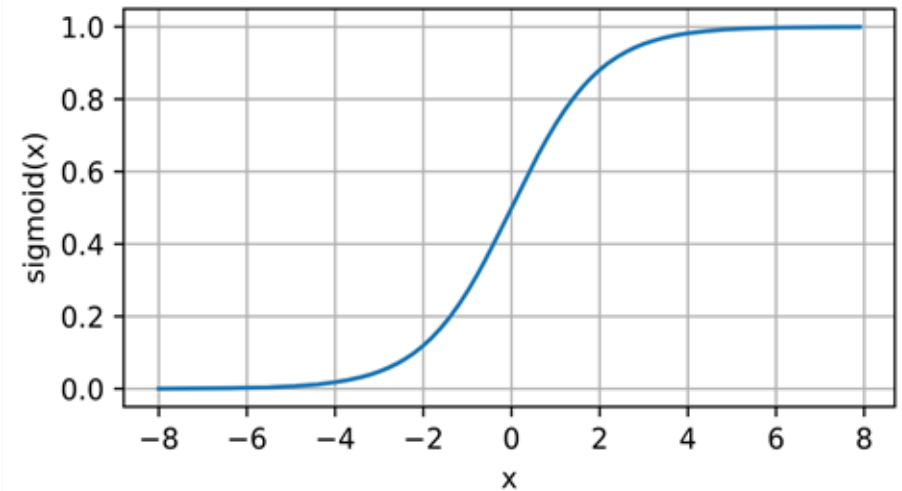
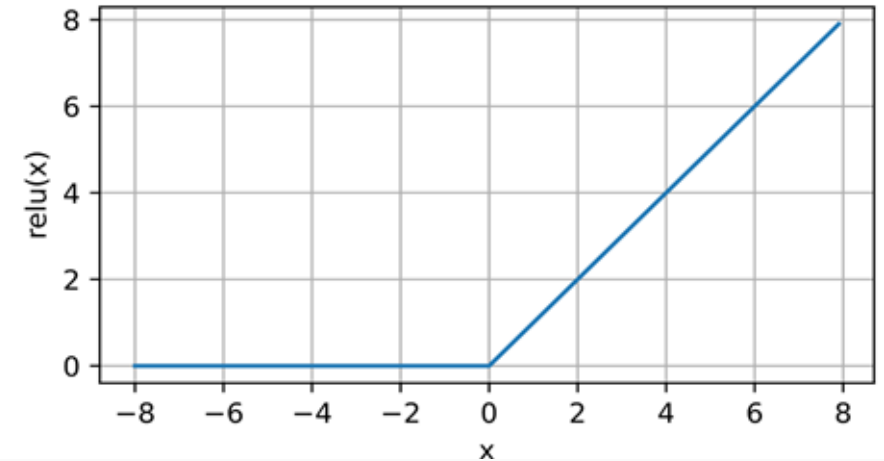
$$\text{ReLU}(x) = \max(x, 0).$$

simple and good performance
behaves “well” during backpropagation

- **Sigmoid**

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)}.$$

important for some applications



Multilayer Perceptrons: A great visual introduction

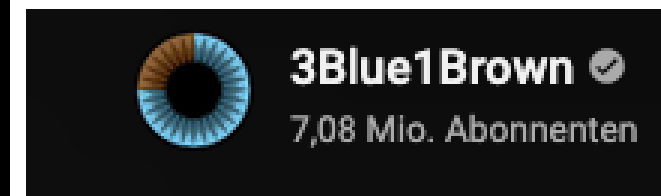
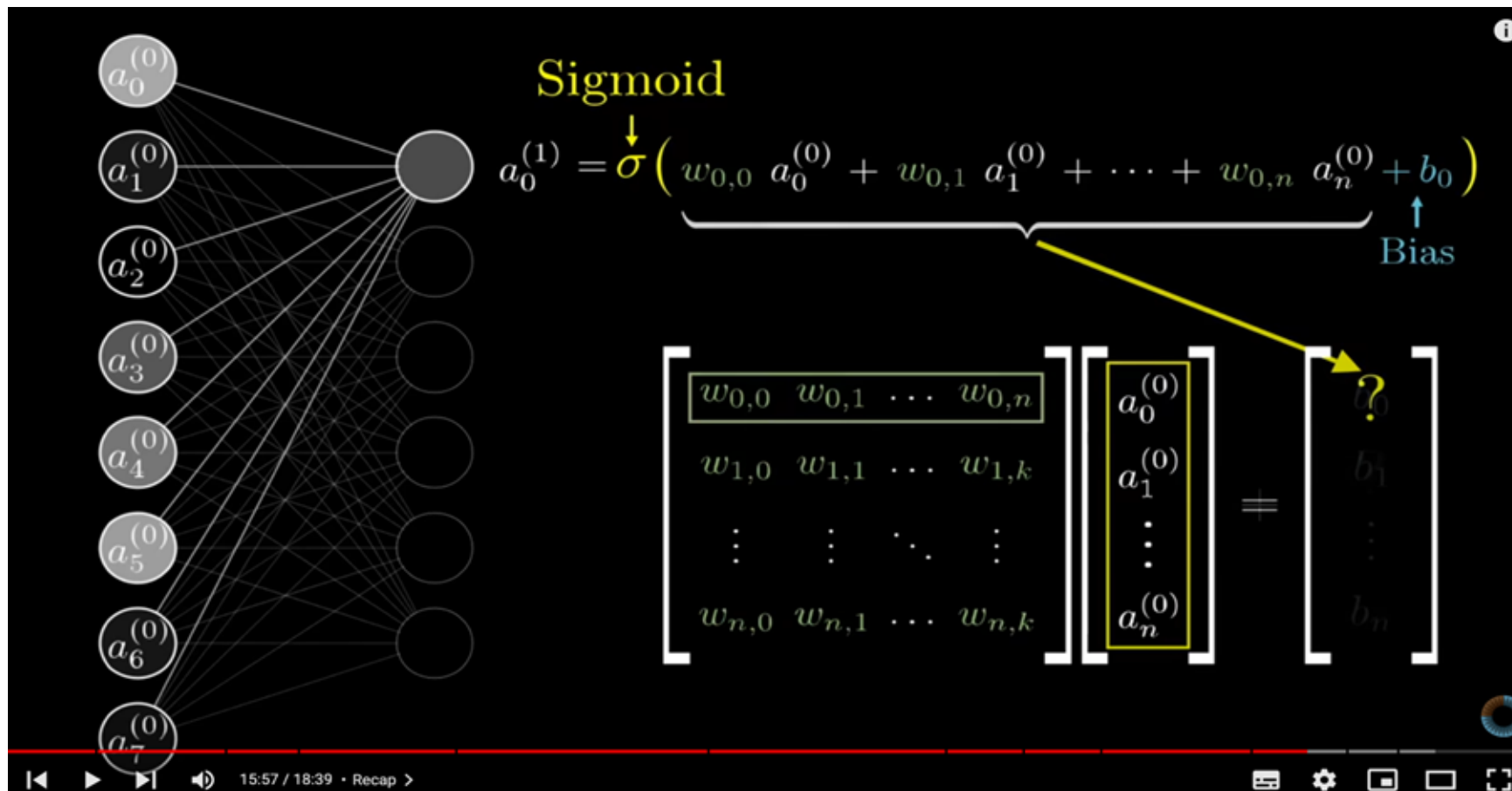
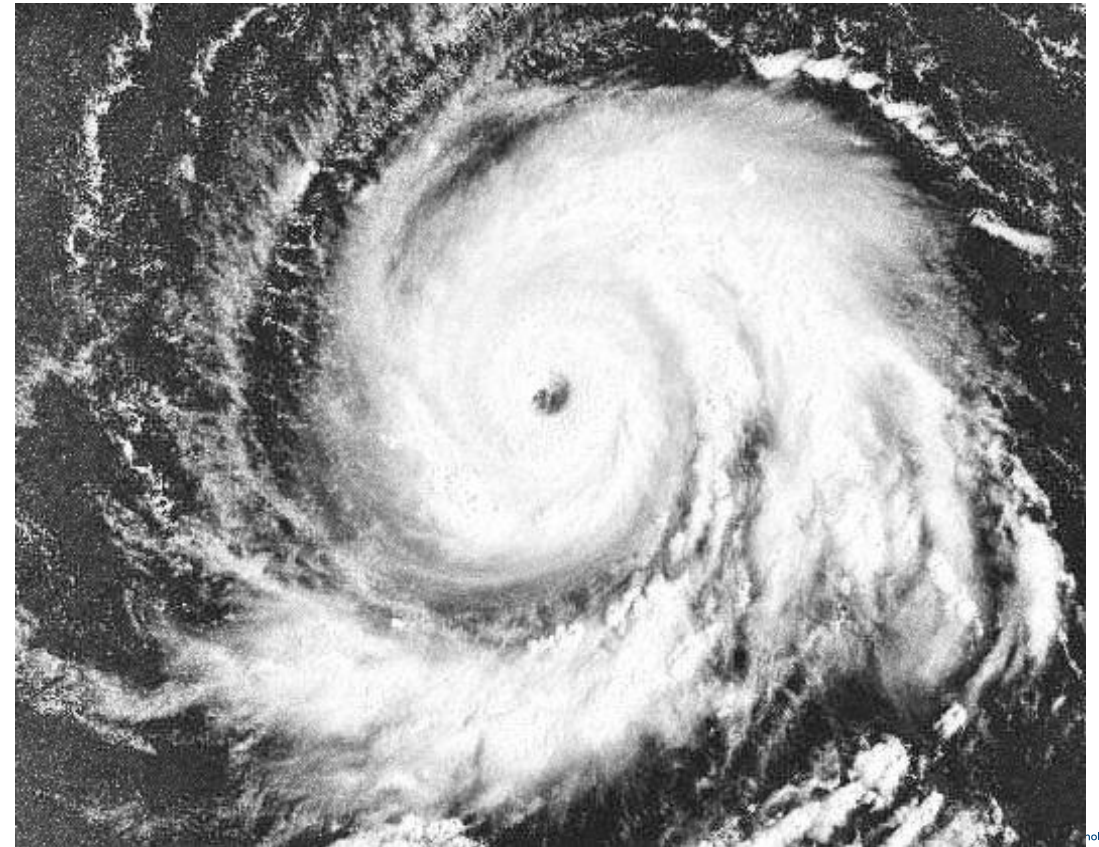


Illustration: Hurricane Classification



Inputs

Source data fed into the neural network, with the goal of making a decision or prediction about the data. Example: Is this a level 5 Category Hurricane?



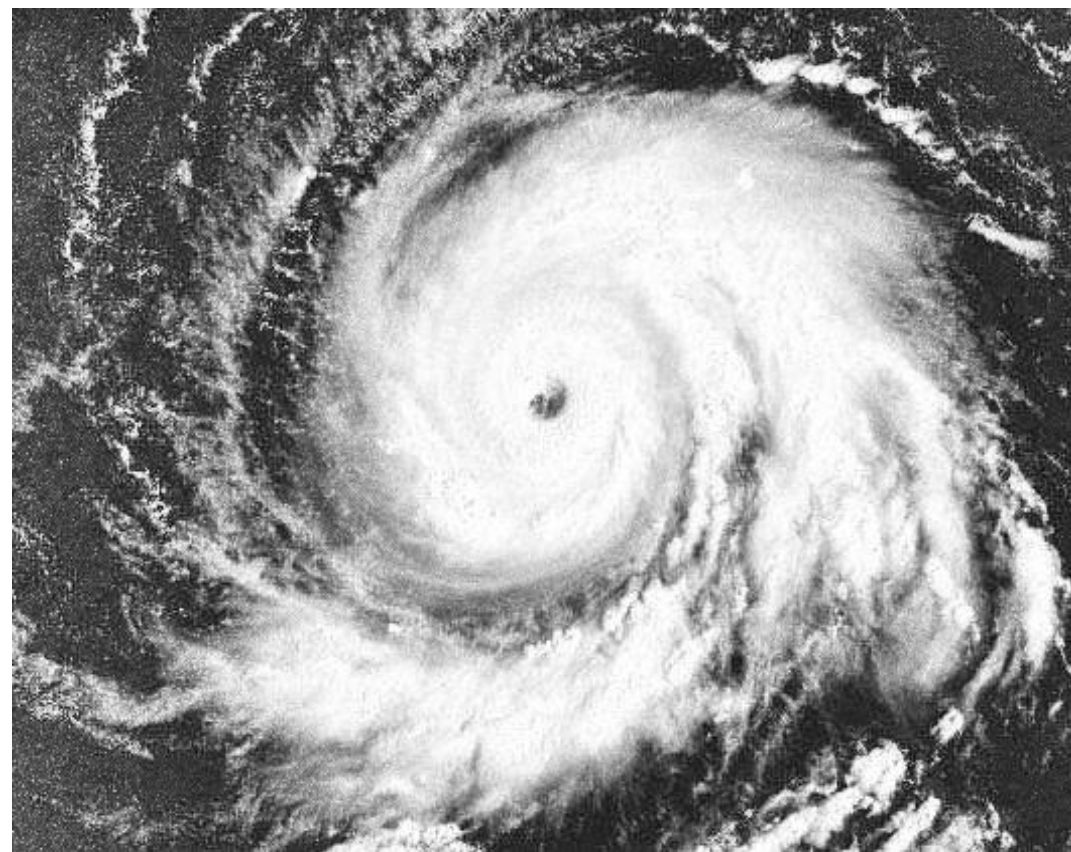


Inputs

Source data fed into the neural network, with the goal of making a decision or prediction about the data. Example: Is this a level 5 Category Hurricane?

Training, Validation, Test Set

A set of outputs for which the correct outputs are known, which can be used to train the neural networks. For example Pre-Classified Images of Hurricanes labelled by hand.

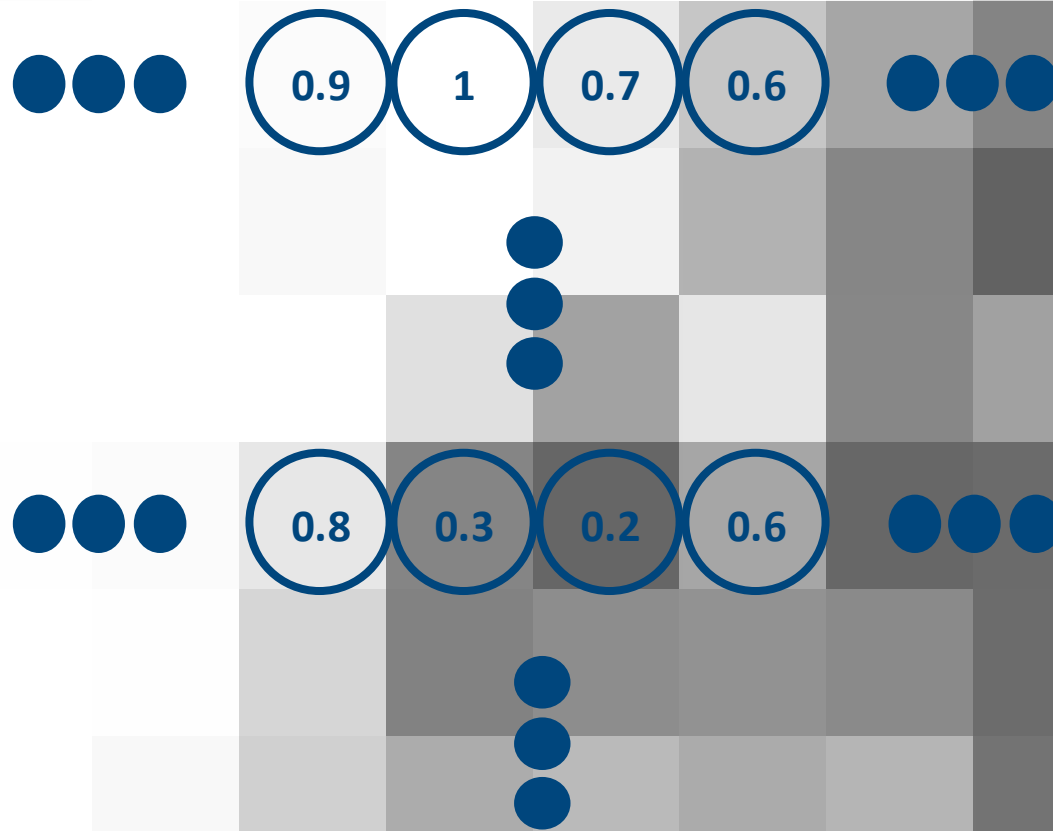




Inputs



Training (and Validation) Dataset





Inputs



Training, Validation, Test Set



Nodes, Weights, Biases

A network with one layer that contains one node



Activation Function

input a

0.6

0.7

1



0.3

0.2

0.6

weights

bias



activation
function σ

$$\sigma(w_1 a_1 + w_2 a_2 + \dots + w_n a_n + b)$$



Inputs



Training, Validation, Test Set



Nodes, Weights, Biases

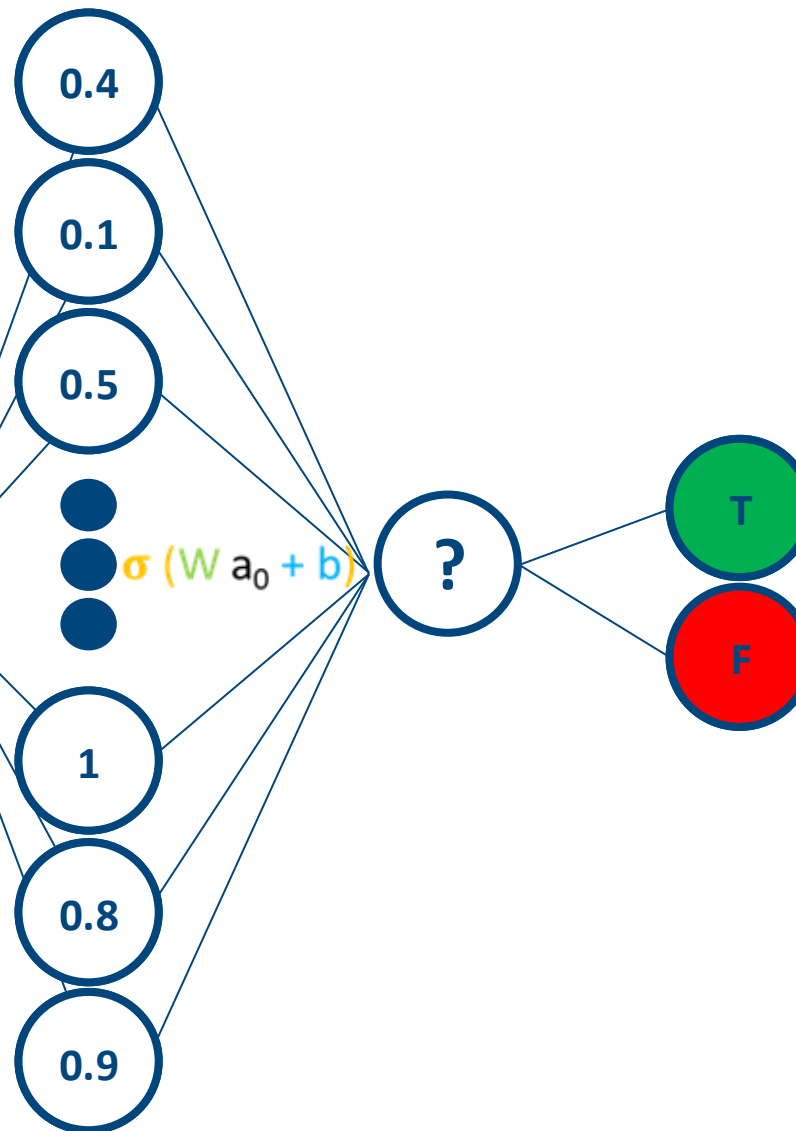


Activation Function



Outputs

The output of the neural network can be a bounded to a real value between 0 and 1 (classification) or any value (regression).

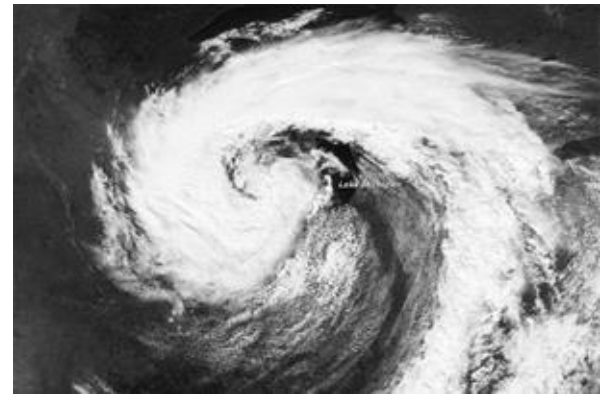


Hurricane Classification with an MLP

Would you expect this to work?

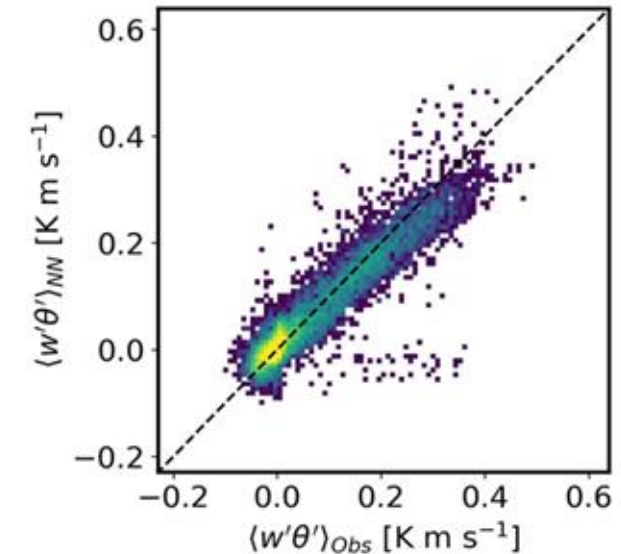
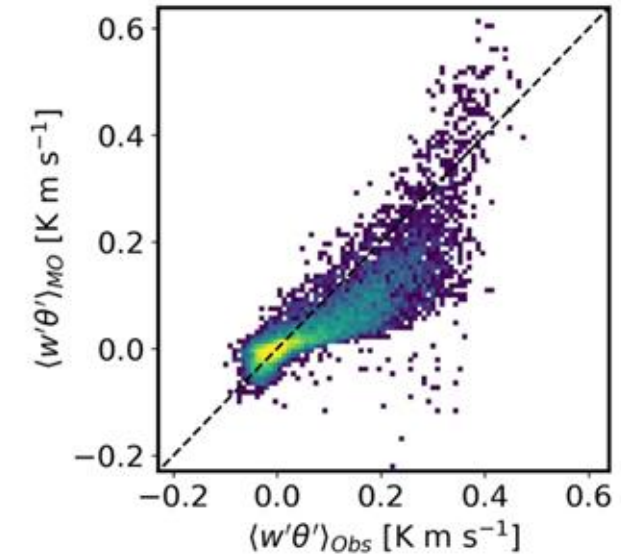
No, the MLP does not have translation equivariance.

Any more questions?



Predicting surface heat fluxes

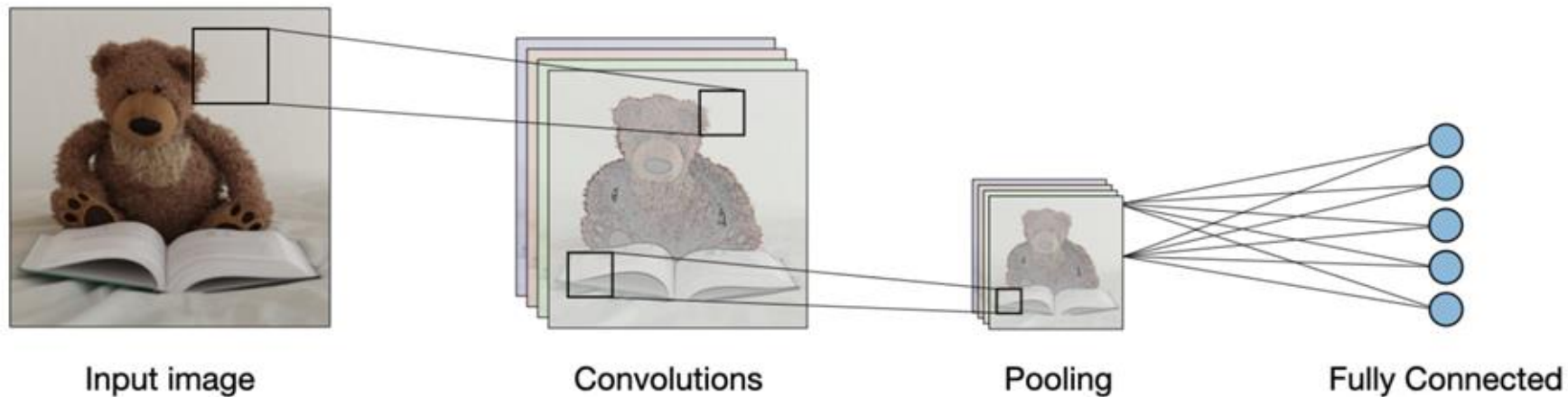
- **Predicting surface heat fluxes using a neural network trained on 1D observations**
 - input: wind speed, temperature, Richardson number
 - two hidden layers with 64 neurons each
 - Top: Traditional Approach
 - Bottom: Neural Network



Convolutional Neural Networks

Convolutional Neural Networks

- **Multilayer Perceptrons are not translation equivariant**
- **Convolutional Neural Networks solve this problem**
- **Popular for any 2-dimensional data, especially image classification tasks**
- **You will be programming one later**



1. Convolution

$$[\mathbf{H}]_{i,j} = u + \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} [\mathbf{V}]_{a,b} [\mathbf{X}]_{i+a,j+b}.$$

i,j : pixel location

$\mathbf{V}$: weight matrix of kernel

u : bias

Δ : kernel size

a,b : kernel indices

$\mathbf{X}$: input

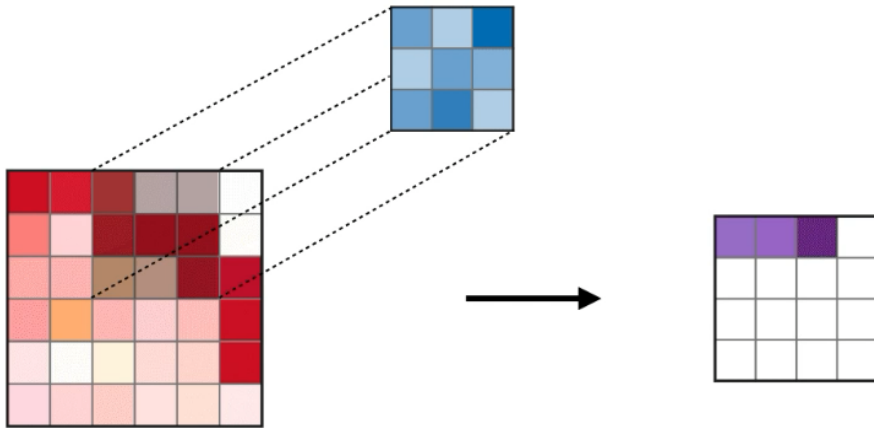
$\mathbf{H}$: hidden or “latent” state

(not strictly a convolution, but a cross-correlation)

Input				Kernel			Output	
0	1	2	*	0	1	=	19	25
3	4	5		2	3		37	43
6	7	8						

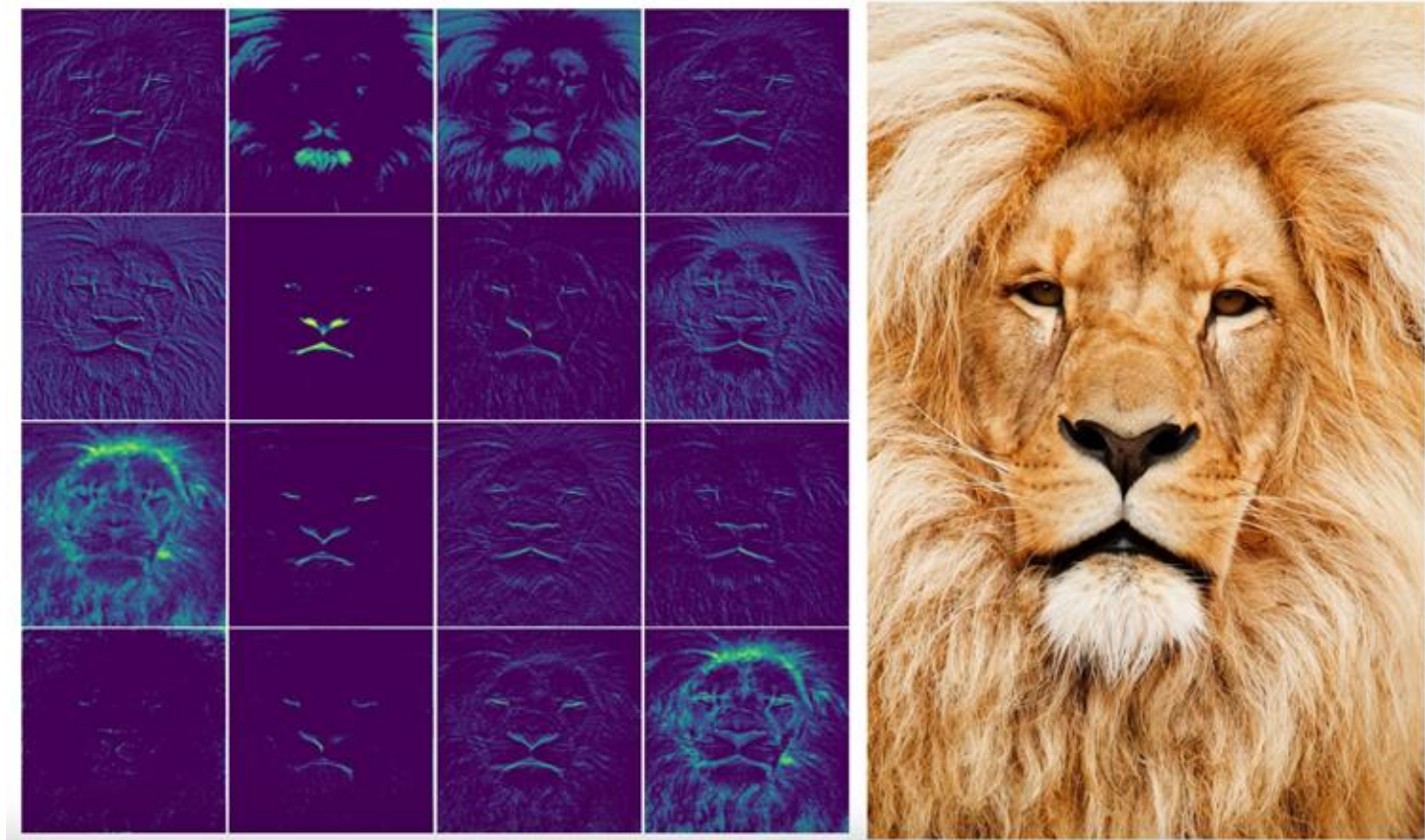
1. Convolution

$$[\mathbf{H}]_{i,j} = u + \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} [\mathbf{V}]_{a,b} [\mathbf{X}]_{i+a,j+b}.$$



- Typically, there are multiple kernels → output "channels"
- The amount of input channels is determined by the input image or the output channels of the previous layer
- Input channels = kernel depth (in z-direction)
- Kernels are trained

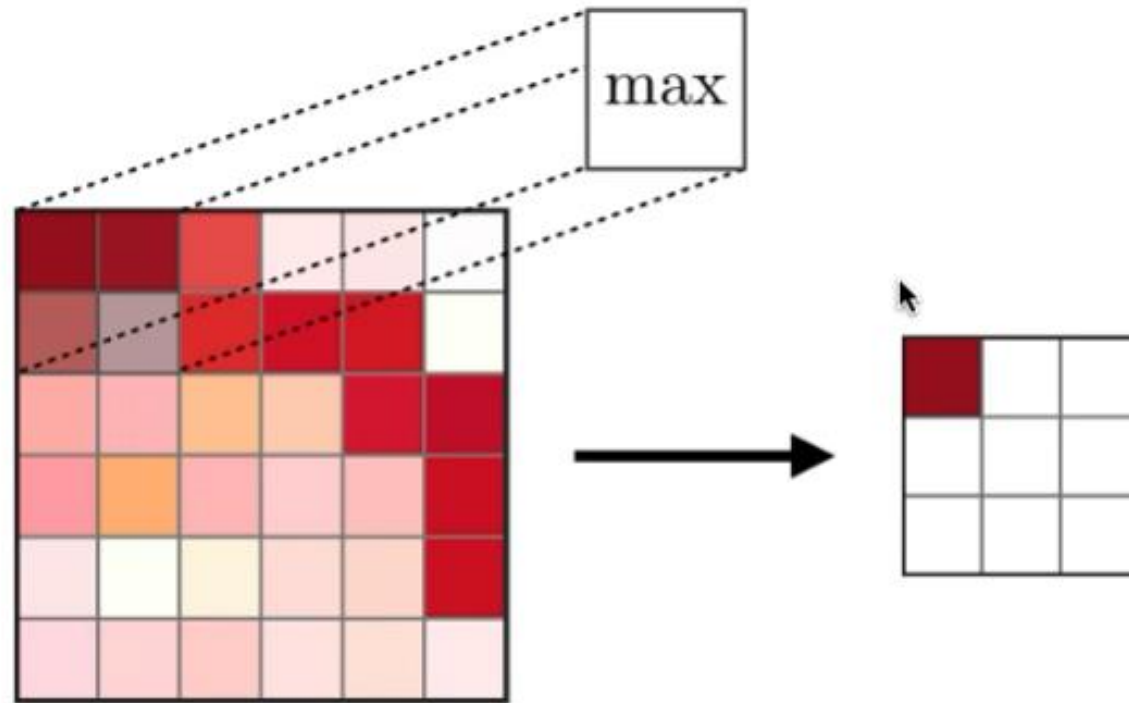
How does a CNN learn?



<https://towardsdatascience.com/exploring-feature-extraction-with-cnns-345125cefc9a>

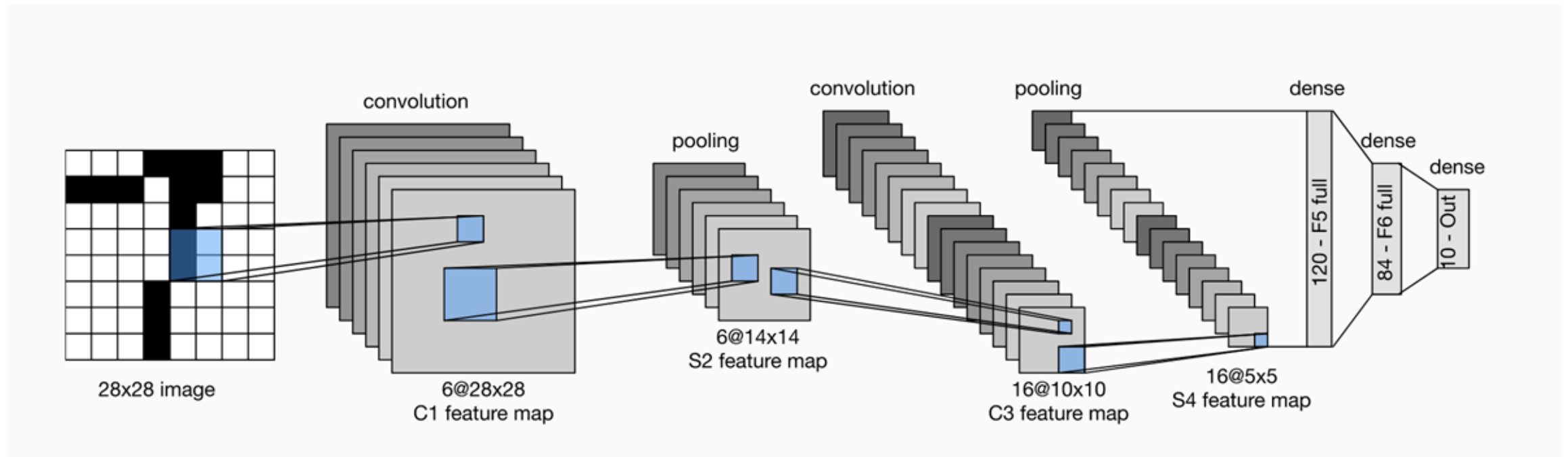
2. Pooling

A downsampling operation typically after the convolution layer

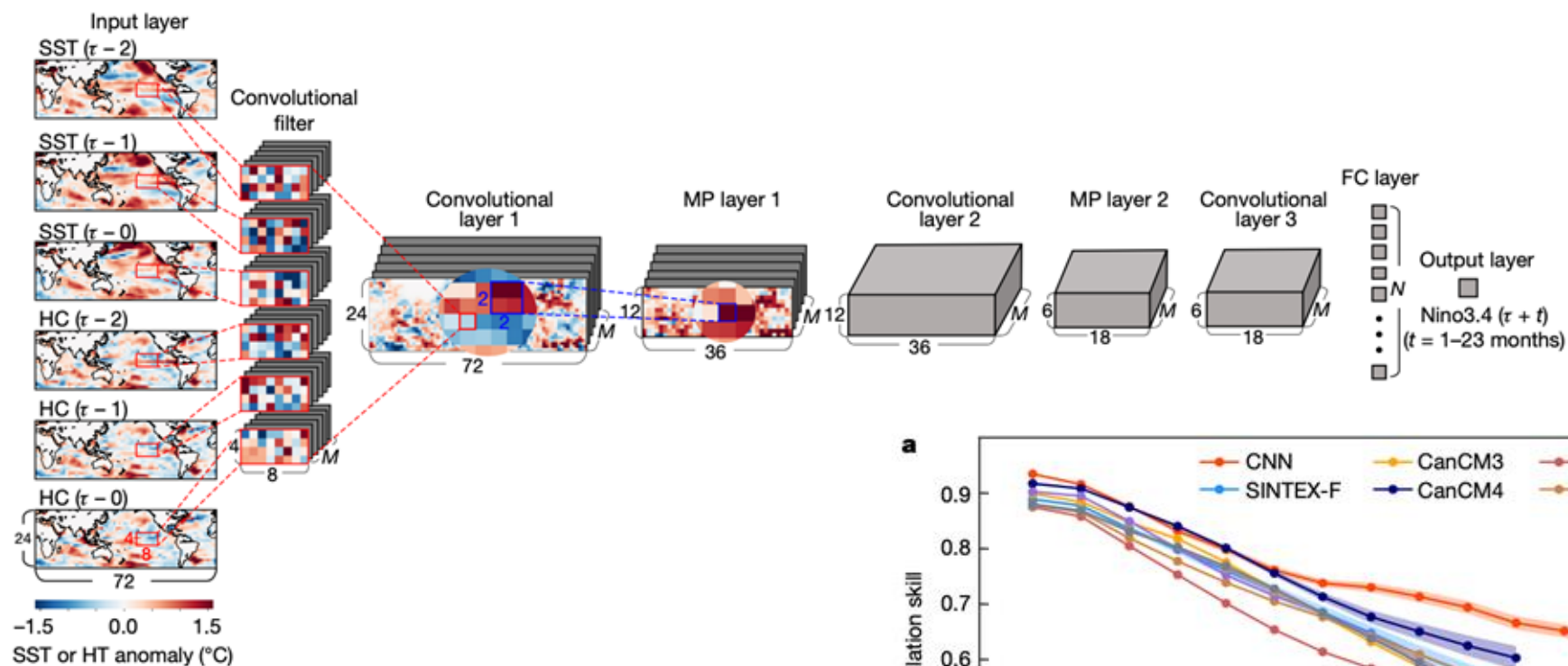


- Max pooling
- Average pooling
- ...

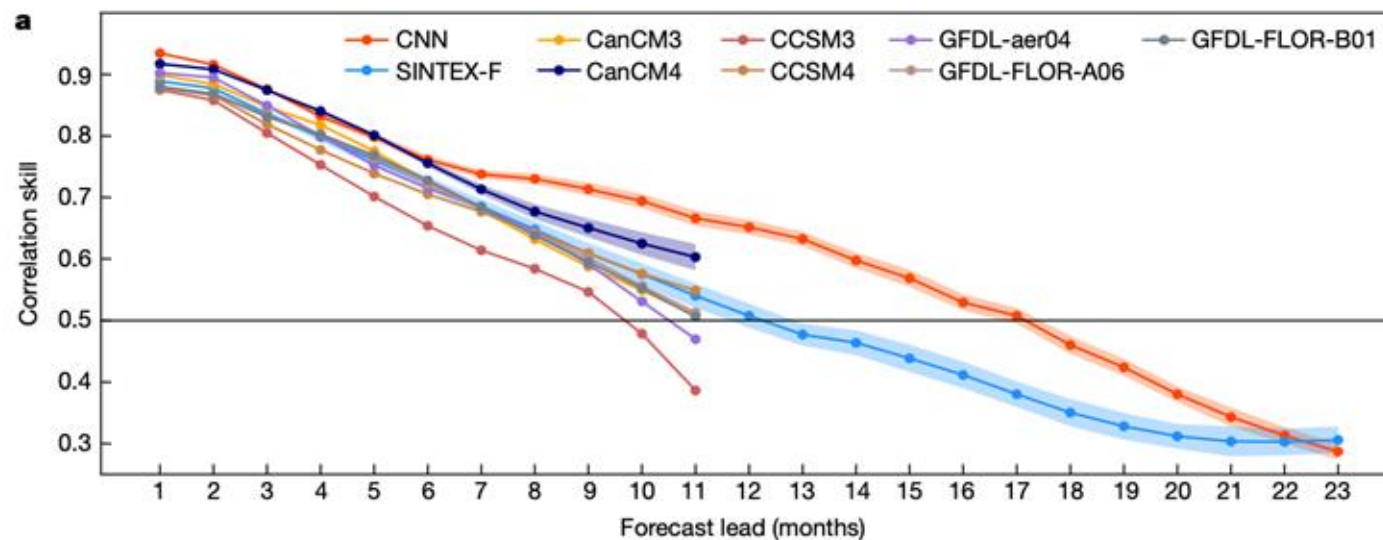
Putting it all together



CNNs for multi-year ENSO forecasts



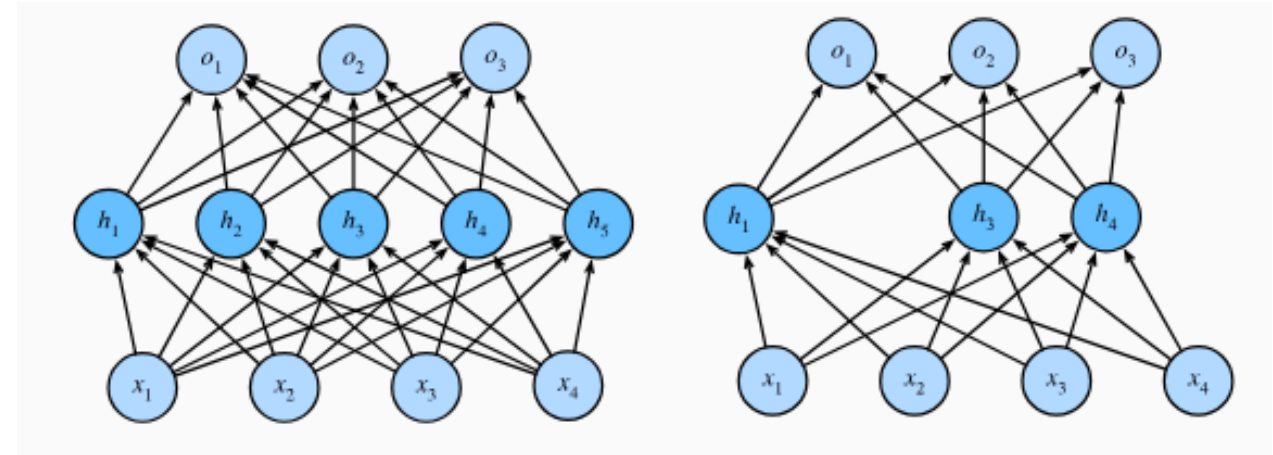
Ham, Yoo-Geun, Jeong-Hwan Kim, and Jing-Jia Luo. "Deep learning for multi-year ENSO forecasts." *Nature* 573.7775 (2019): 568-572.



Important Building Blocks

Important Building Blocks: Dropout Layers

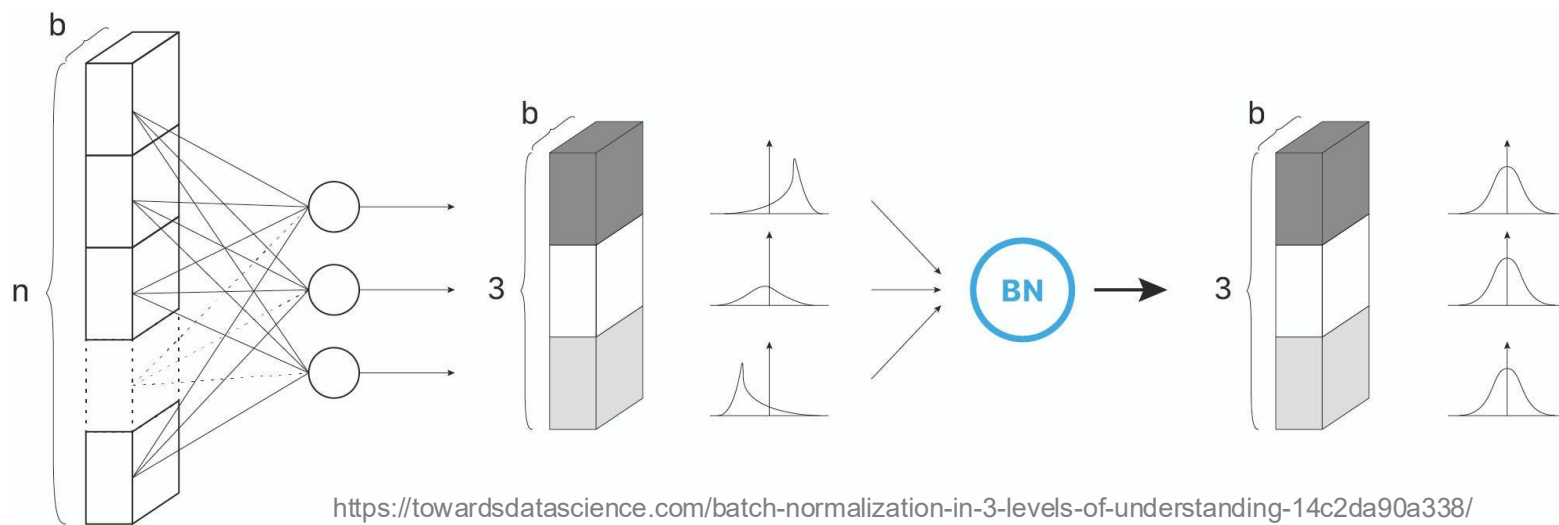
- **Neural Networks are prone to overfitting and sometimes lack robustness**
- **One cause: Over-relying on certain neurons**
- **Solution: At every epoch, randomly remove a fraction p of neurons**
 - Typically $p=0.1$ to $p=0.5$
- **At test-time ("inference") all neurons are typically active**
- **Often used for fully connected layers**



Important Building Blocks: Batch Normalisation

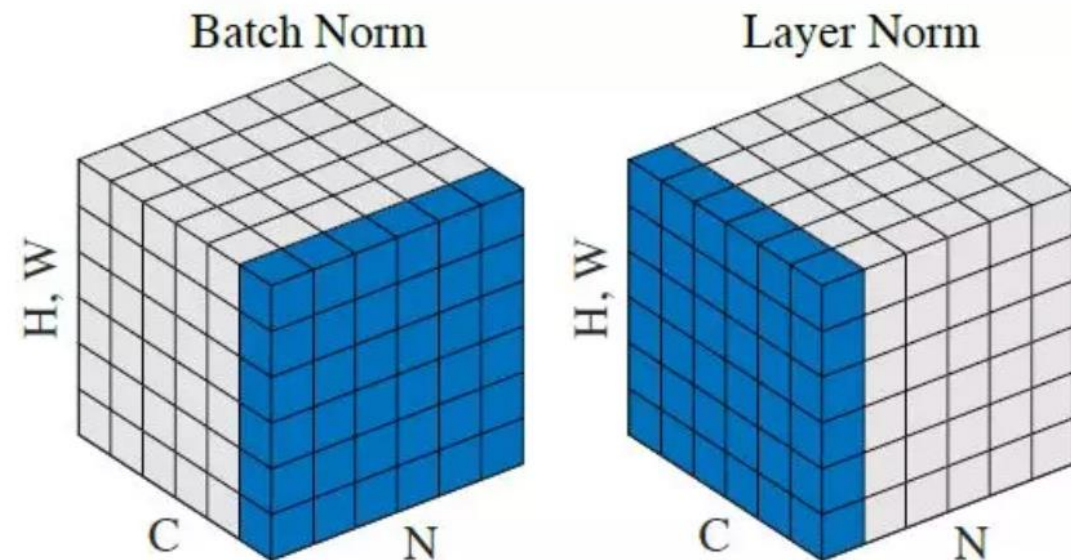
- Problem in deep networks: Internal distribution shift
- Can Impact training speed, robustness and overfitting
- Solution: Normalise the current batch and the rescale with learnable parameters γ and β
- In CNNs: Calculated over Batch and Image dimension, but not channel dimension

$$\text{BN}(\mathbf{x}) = \gamma \odot \frac{\mathbf{x} - \hat{\boldsymbol{\mu}}_B}{\hat{\boldsymbol{\sigma}}_B} + \beta.$$



Important Building Blocks: Layer Normalisation

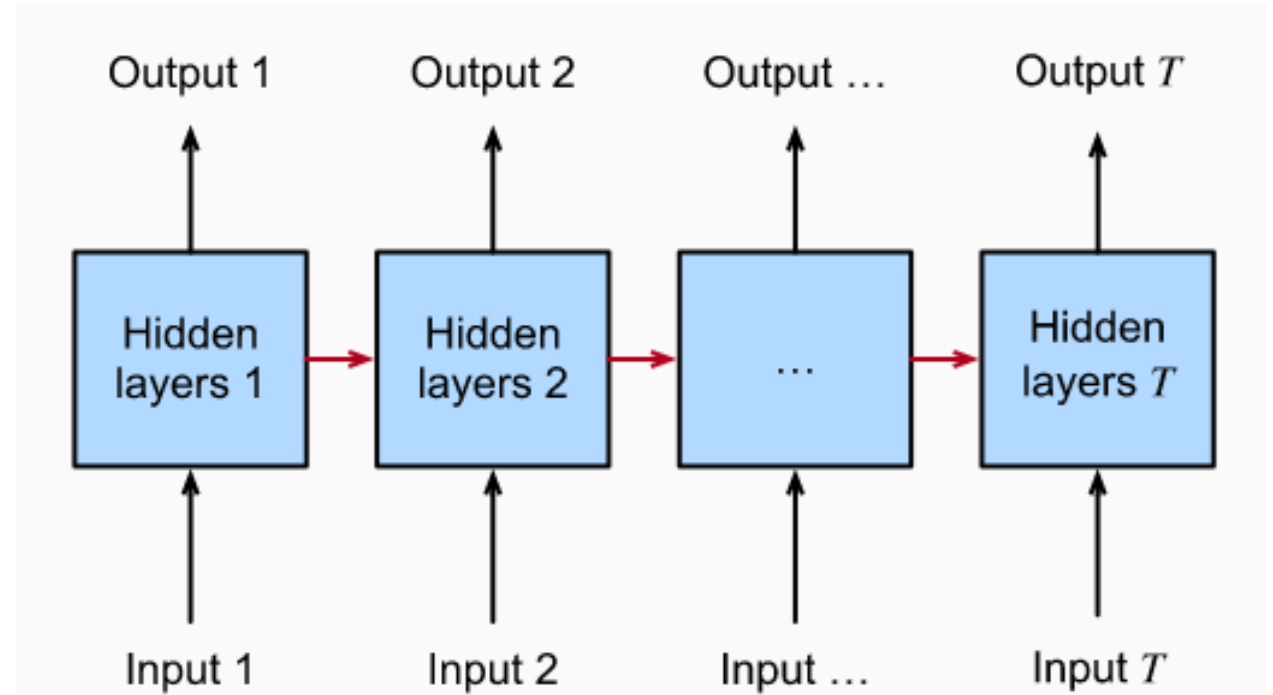
- For consecutive layer activations, gradients can become very small or large (exploding/vanishing gradients)
 - This slows down or completely breaks training
- **Solution: Normalise over channels and image dimensions, but not over samples in a batch**
- Often used in deep sequential models like transformers or recurrent neural networks



Recurrent Neural Networks

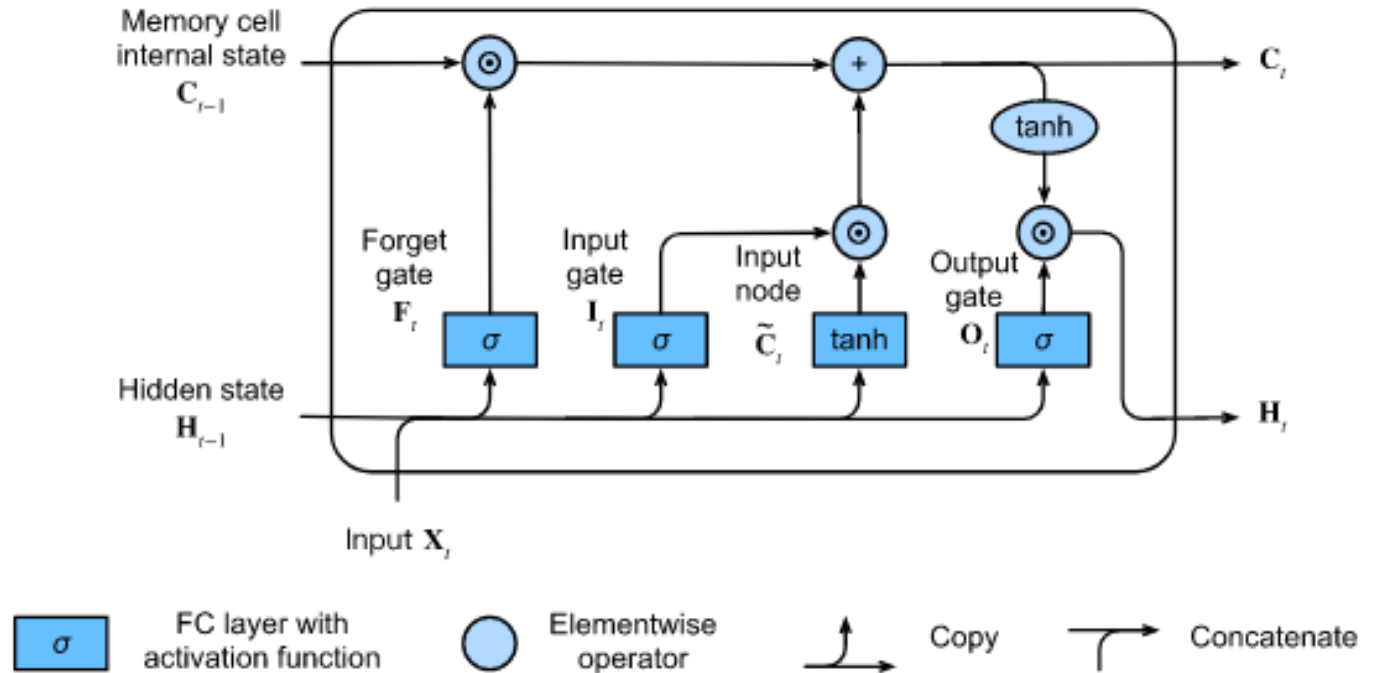
Recurrent Neural Networks

- Used for sequential data: time series prediction, language processing
- memory mechanism
- problem: exploding or vanishing gradients during learning
- (not necessarily state-of-the-art anymore)



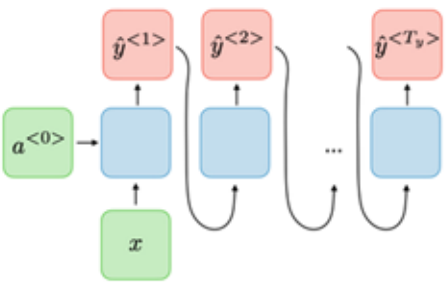
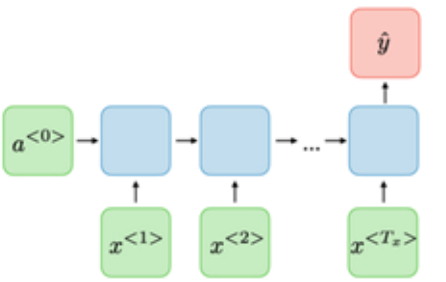
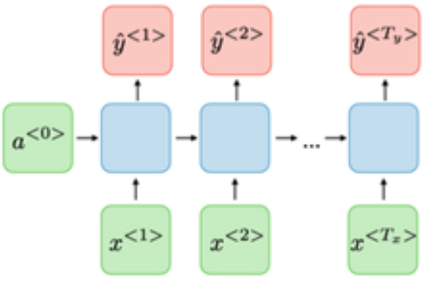
Long short-term memory (LSTM)

- Can learn what important data is, and remembers this data for a long time
- Hidden State: Short term memory
- Memory Cell: Long term memory
- Alternative: „Gated Recurrent Unit“ Networks

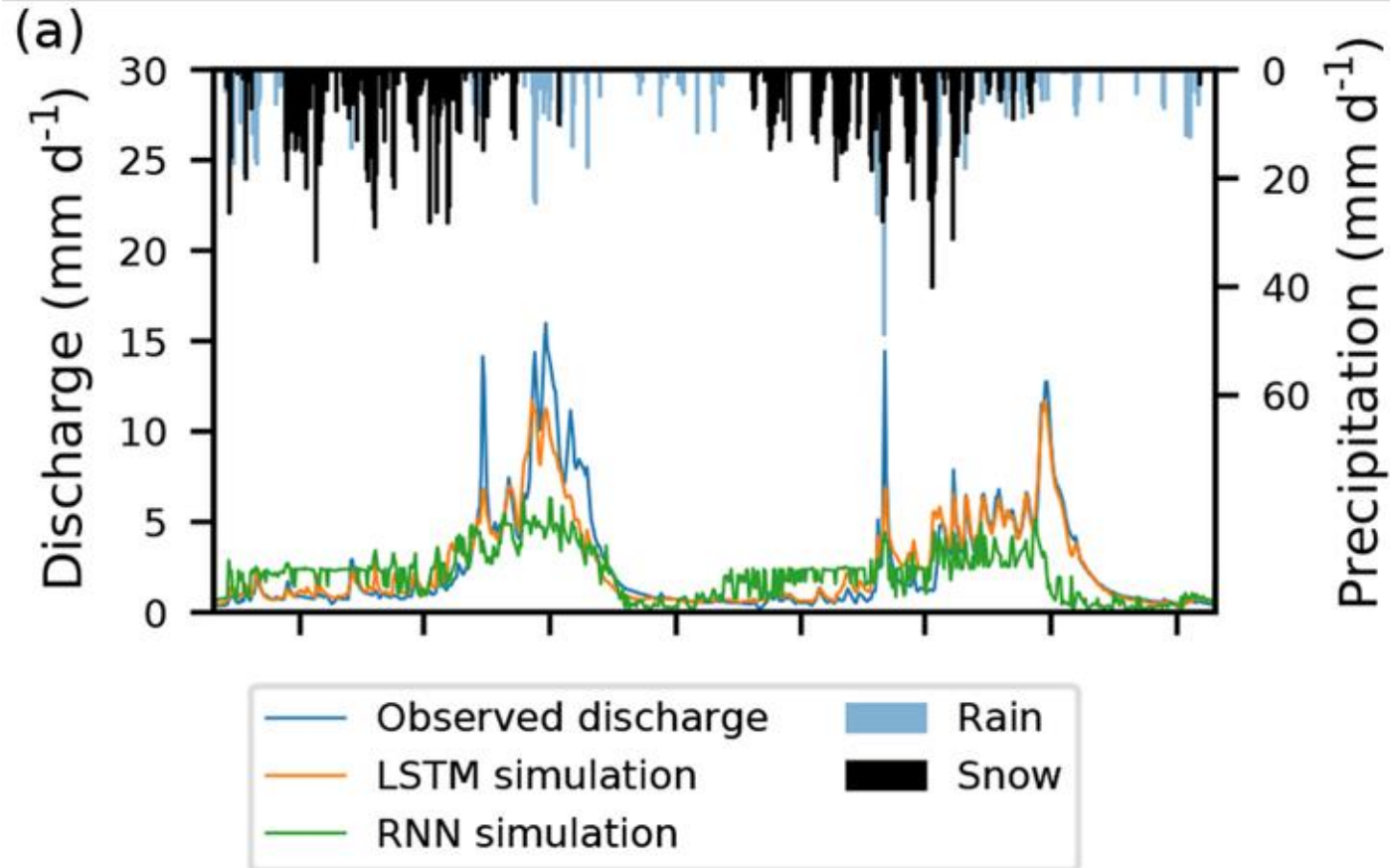


$$\begin{aligned} \mathbf{I}_t &= \sigma(\mathbf{X}_t \mathbf{W}_{xi} + \mathbf{H}_{t-1} \mathbf{W}_{hi} + \mathbf{b}_i), \\ \mathbf{F}_t &= \sigma(\mathbf{X}_t \mathbf{W}_{xf} + \mathbf{H}_{t-1} \mathbf{W}_{hf} + \mathbf{b}_f), \\ \mathbf{O}_t &= \sigma(\mathbf{X}_t \mathbf{W}_{xo} + \mathbf{H}_{t-1} \mathbf{W}_{ho} + \mathbf{b}_o), \end{aligned}$$

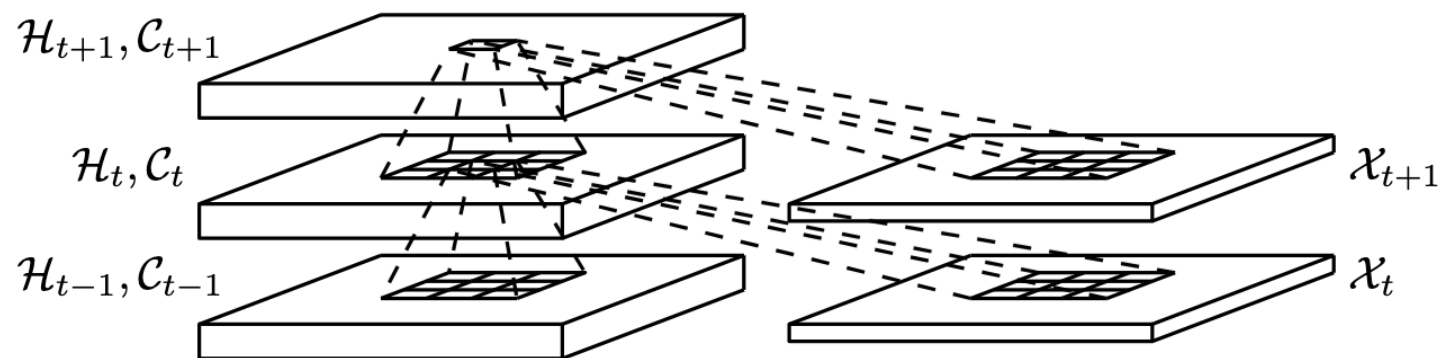
Different ways to employ them

One-to-many $T_x = 1, T_y > 1$		Music generation
Many-to-one $T_x > 1, T_y = 1$		Sentiment classification
Many-to-many $T_x = T_y$		Name entity recognition

LSTM for rainfall-runoff modelling

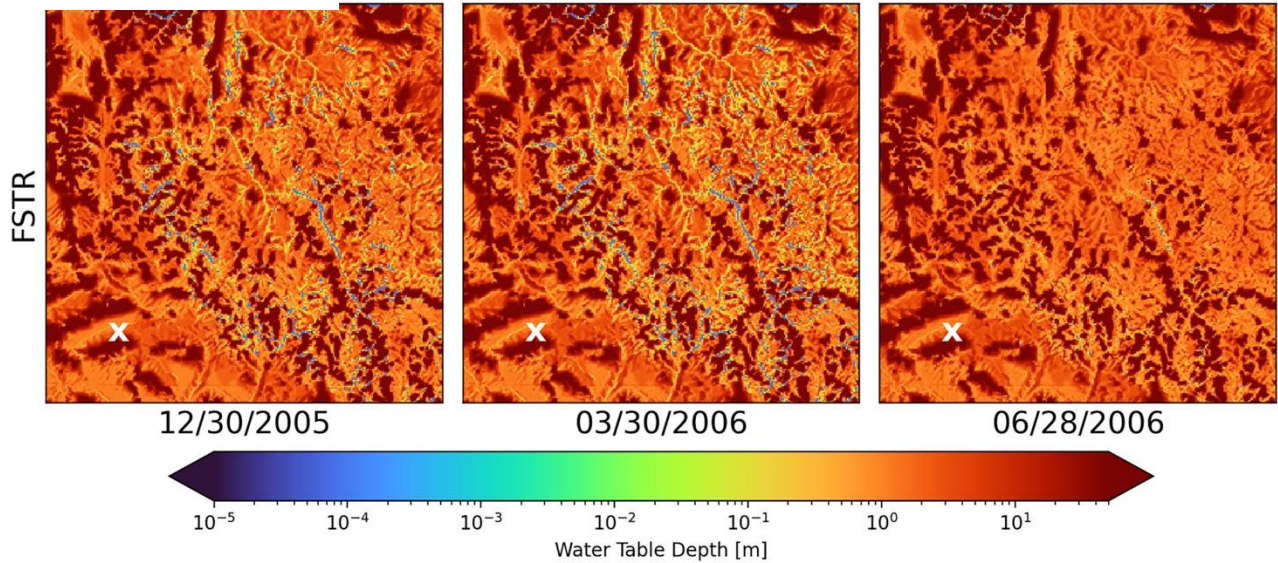


Convolutional LSTM: ConvLSTM



Used to be state-of-the-art for “video” tasks

Shi, Xingjian, et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." *Advances in neural information processing systems* 28 (2015).



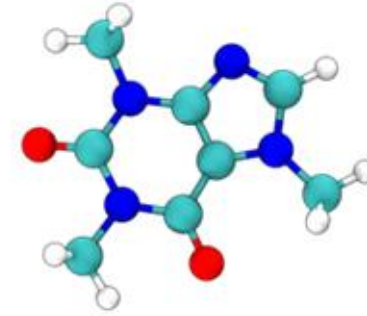
Bennett, A., Tran, H., De la Fuente, L., Triplett, A., Ma, Y., Melchior, P., et al. (2024). *Spatio-temporal machine learning for regional to continental scale terrestrial hydrology*. *Journal of Advances in Modeling Earth Systems*, 16, e2023MS004095. <https://doi.org/10.1029/2023MS004095>

Graph Neural Networks

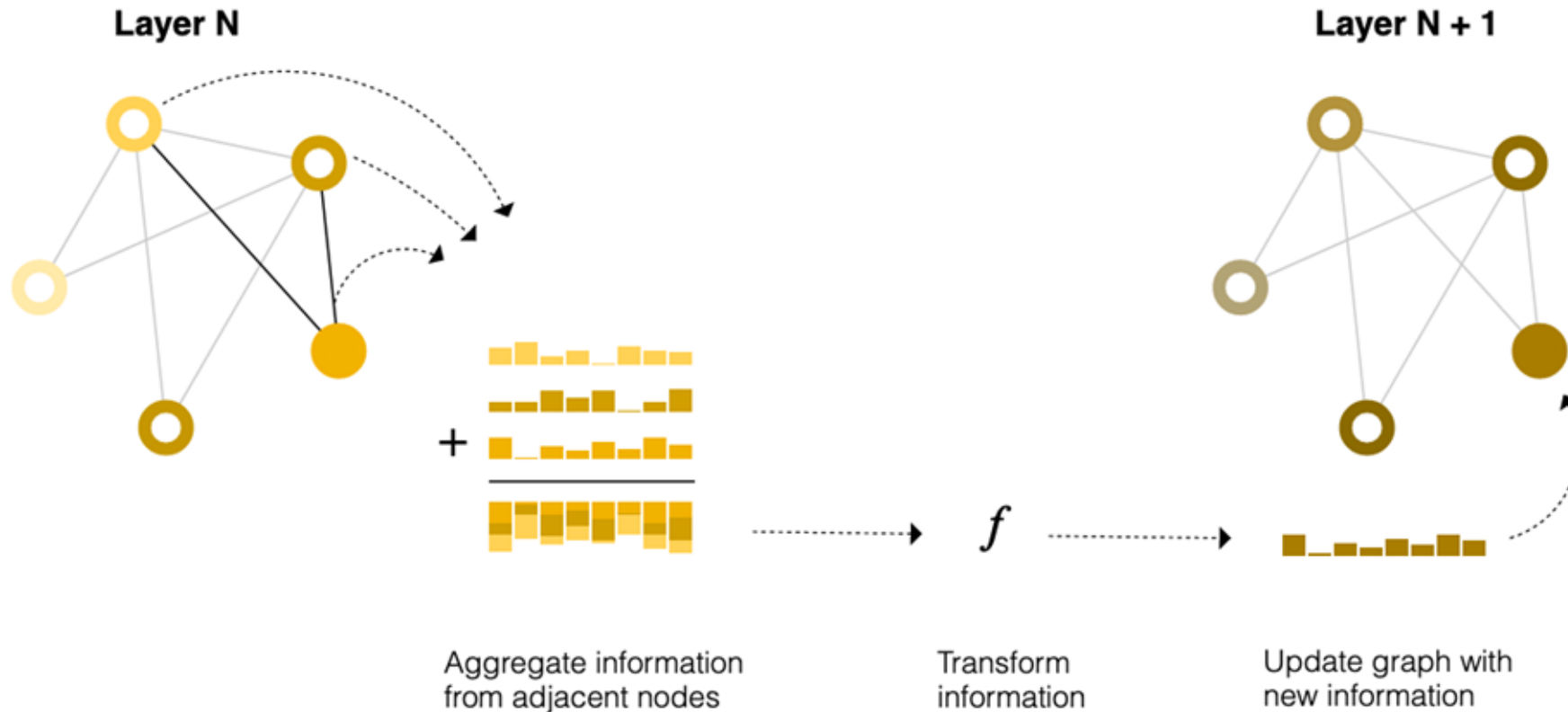
<https://mpimet.mpg.de/forschung/modellierung>

Graph Neural Networks

- Similar to CNNs, but for arbitrary graph structures
- Molecules, social networks, climate model grid, weather stations
- nodes, edges
- Used to predict node characteristics, edge characteristics, ...



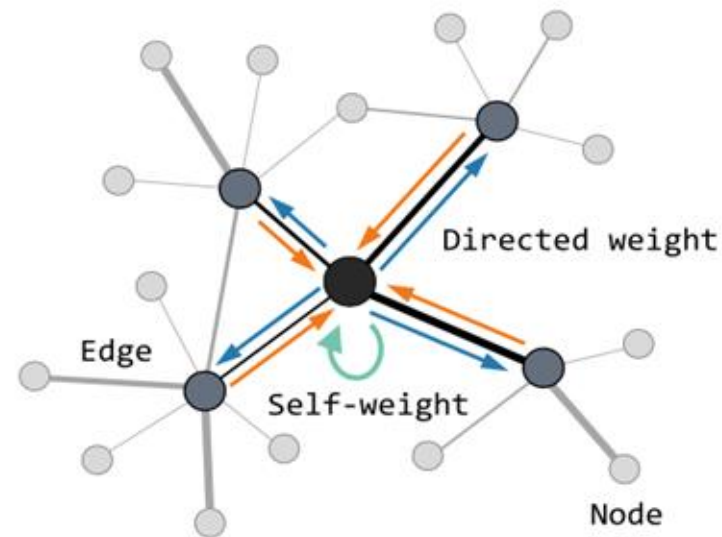
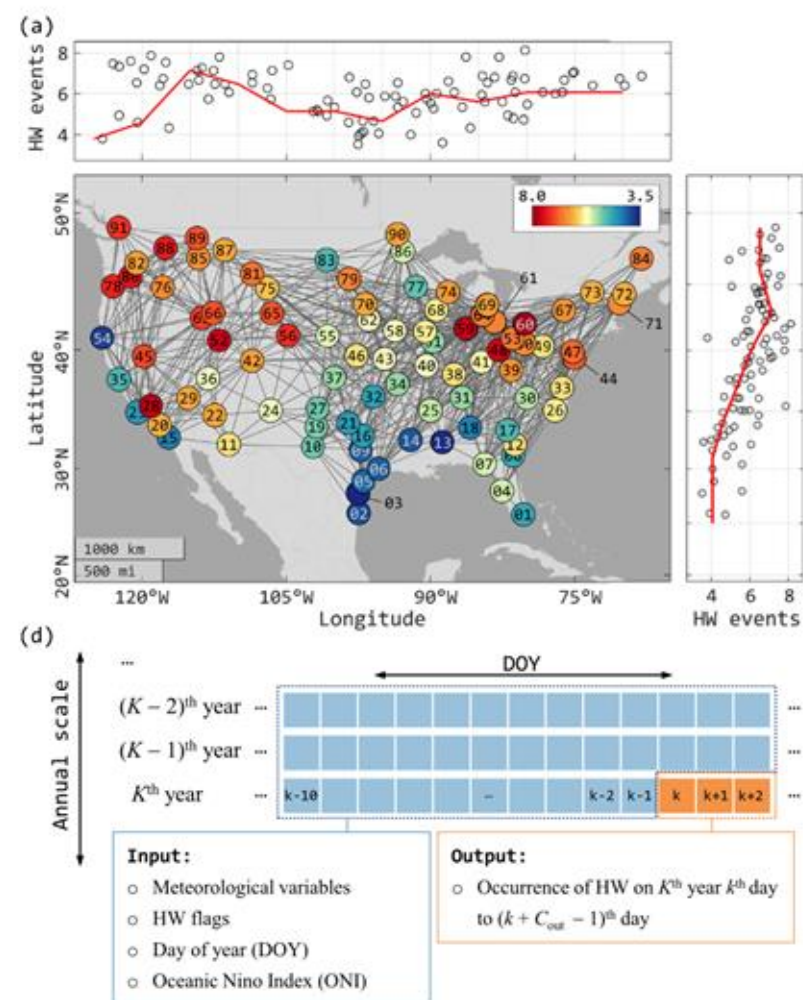
Graph Neural Networks: Message Passing



- f :
- Convolution
 - Attention
 - ...

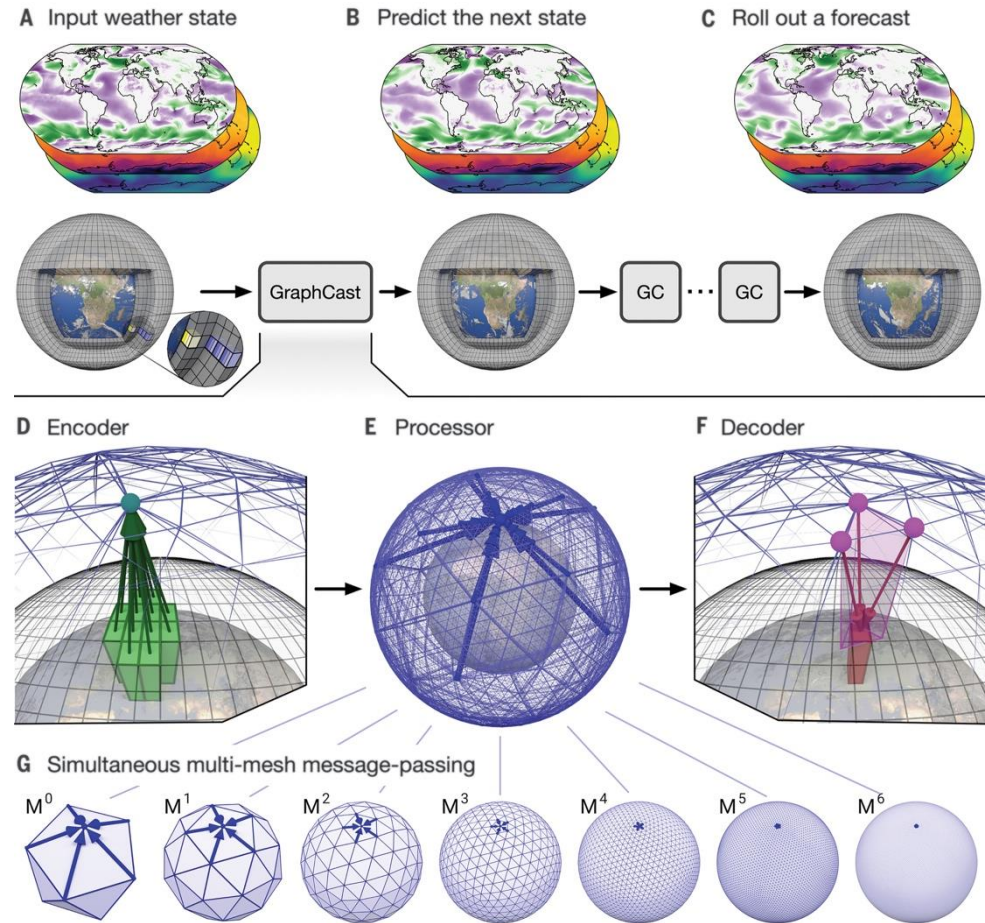
For a great introduction to GNNs: <https://distill.pub/2021/gnn-intro/>

GNNs for predicting heat waves

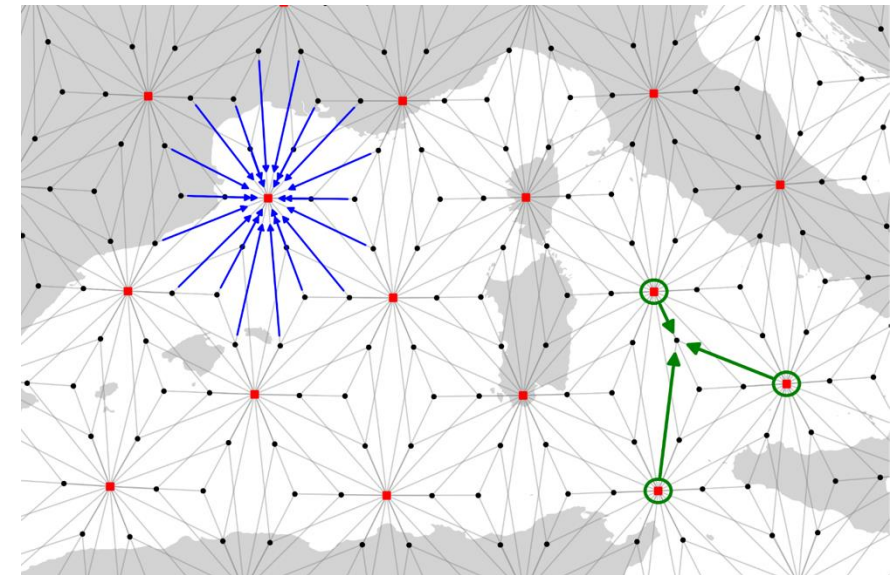
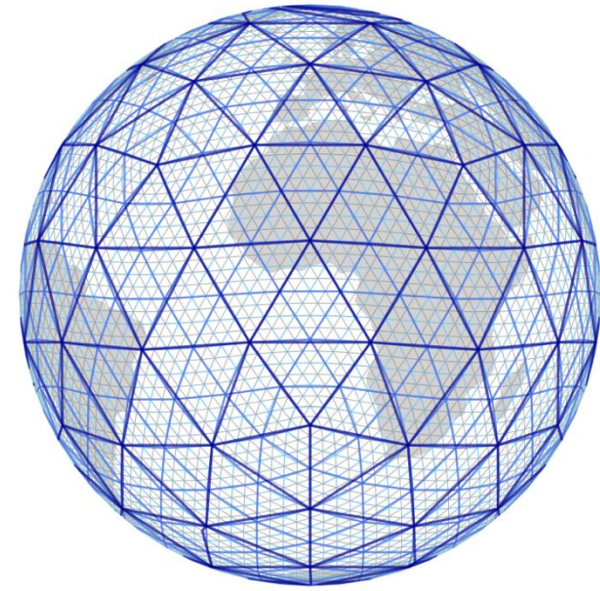


Li, Peiyuan, et al. "Regional heatwave prediction using Graph Neural Network and weather station data." *Geophysical Research Letters* 50.7 (2023): e2023GL103405.

GNNs for Weather Prediction



Remi Lam *et al.*, Learning skillful medium-range global weather forecasting. *Science* **382**, 1416-1421 (2023). DOI: [10.1126/science.adi2336](https://doi.org/10.1126/science.adi2336)



Goecke, Tobias, et al. "AICON: An operational global machine learning weather forecasting model." *arXiv preprint arXiv:2608.24651* (2026).

Break

Transformers

Transformers

- **Most large language models are based on the transformer architecture**
- **Vision Transformers for diverse vision tasks**
- **Core idea: Attention Mechanism**
- **Can learn long-range dependencies**

AN IMAGE IS WORTH 16x16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy^{*,†}, Lucas Beyer^{*}, Alexander Kolesnikov^{*}, Dirk Weissenborn^{*},
Xiaohua Zhai^{*}, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby^{*,†}

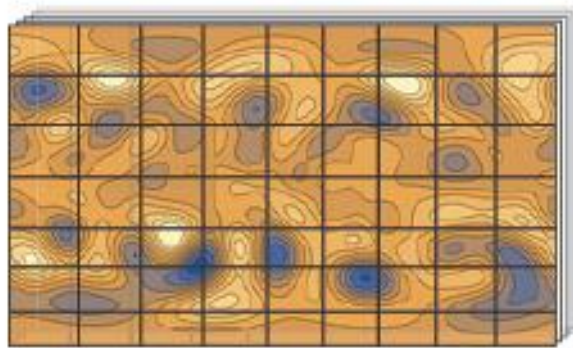
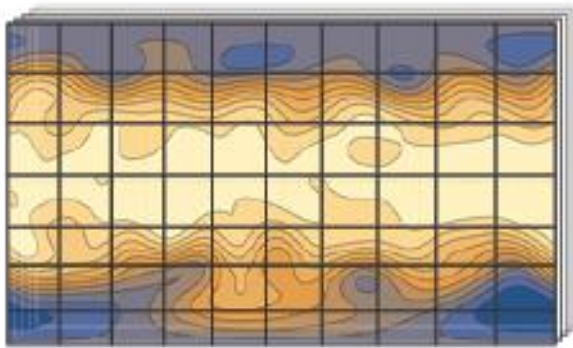
^{*}equal technical contribution, [†]equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train. 

Transformers

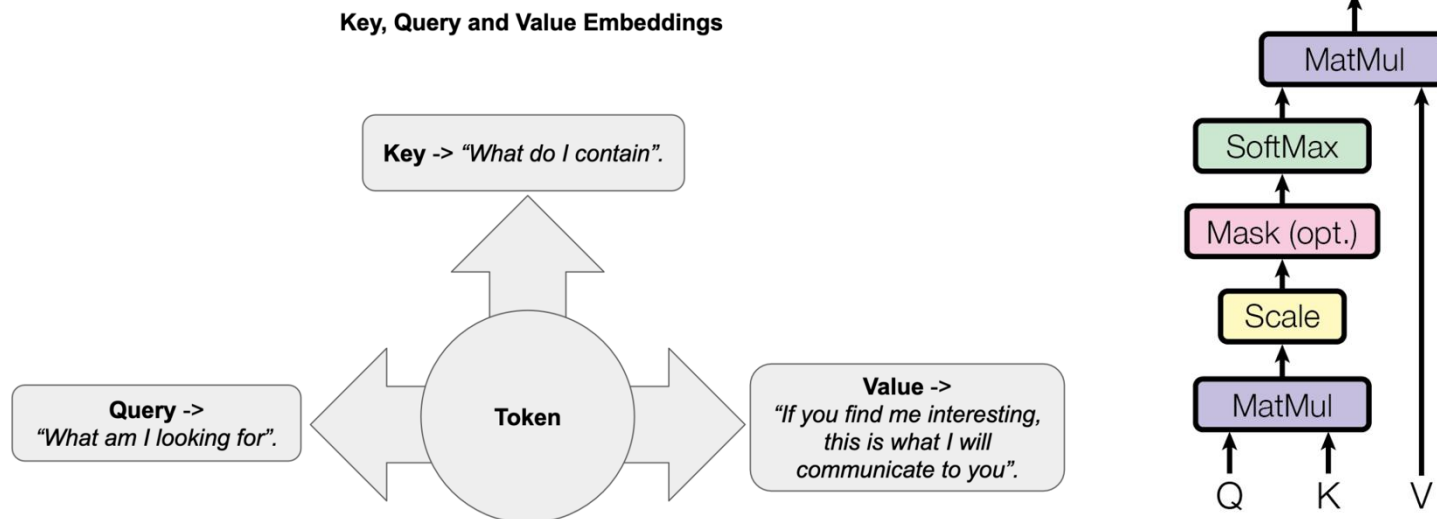
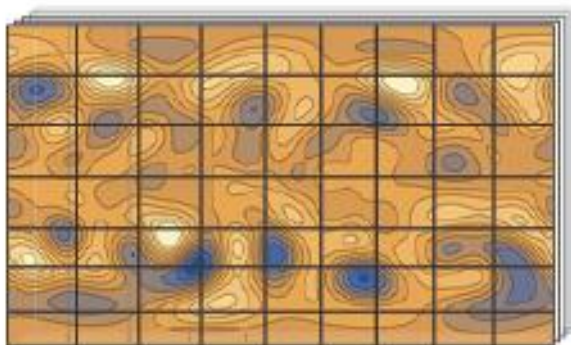
- Divide data into “tokens”



<https://www.atmorep.org/>

Transformers

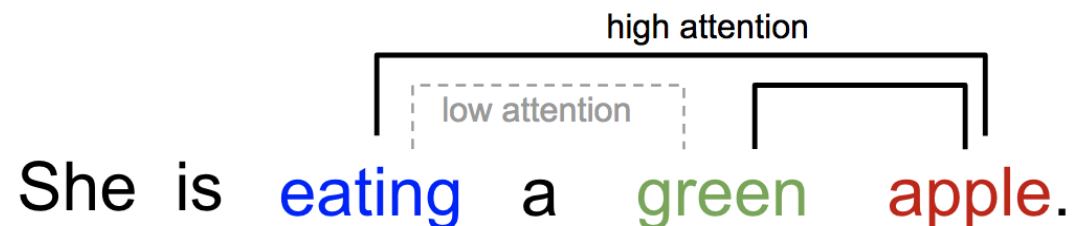
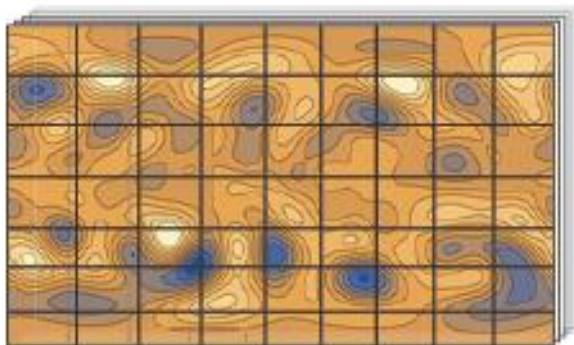
- Embed data into “tokens”
- Transform the tokens into:
 - Key
 - Query
 - Value
- Apply Attention Mechanism



$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

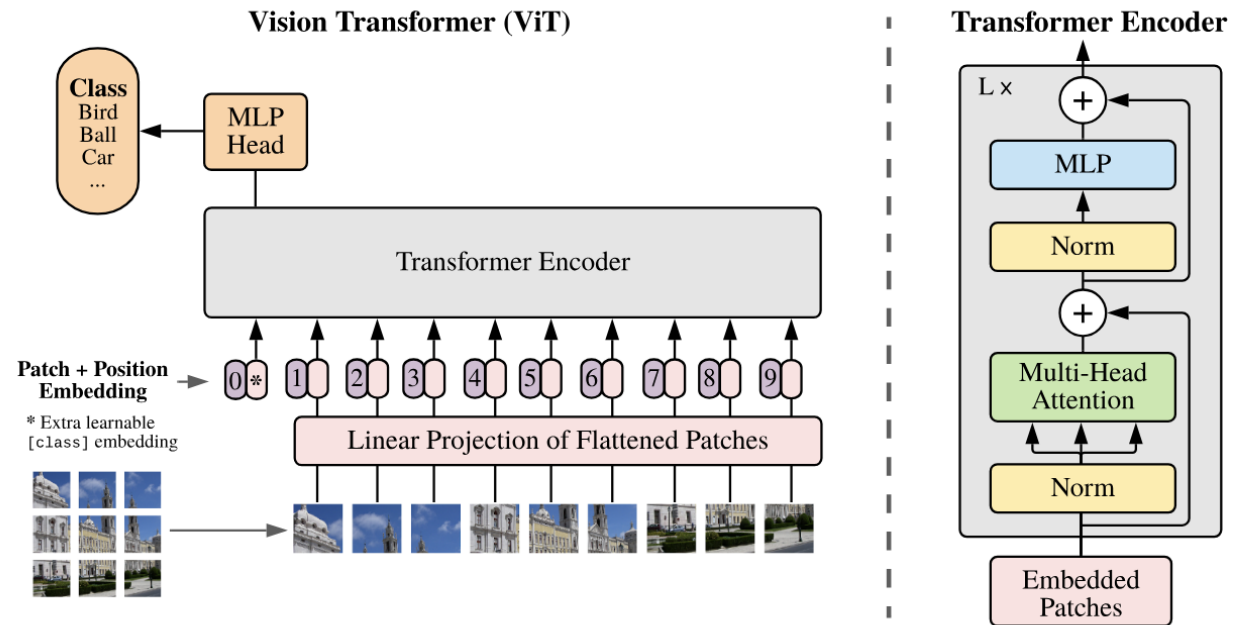
Transformers

- Embed data into “tokens”
- Transform the tokens into:
 - Key
 - Query
 - Value
- Apply Attention Mechanism



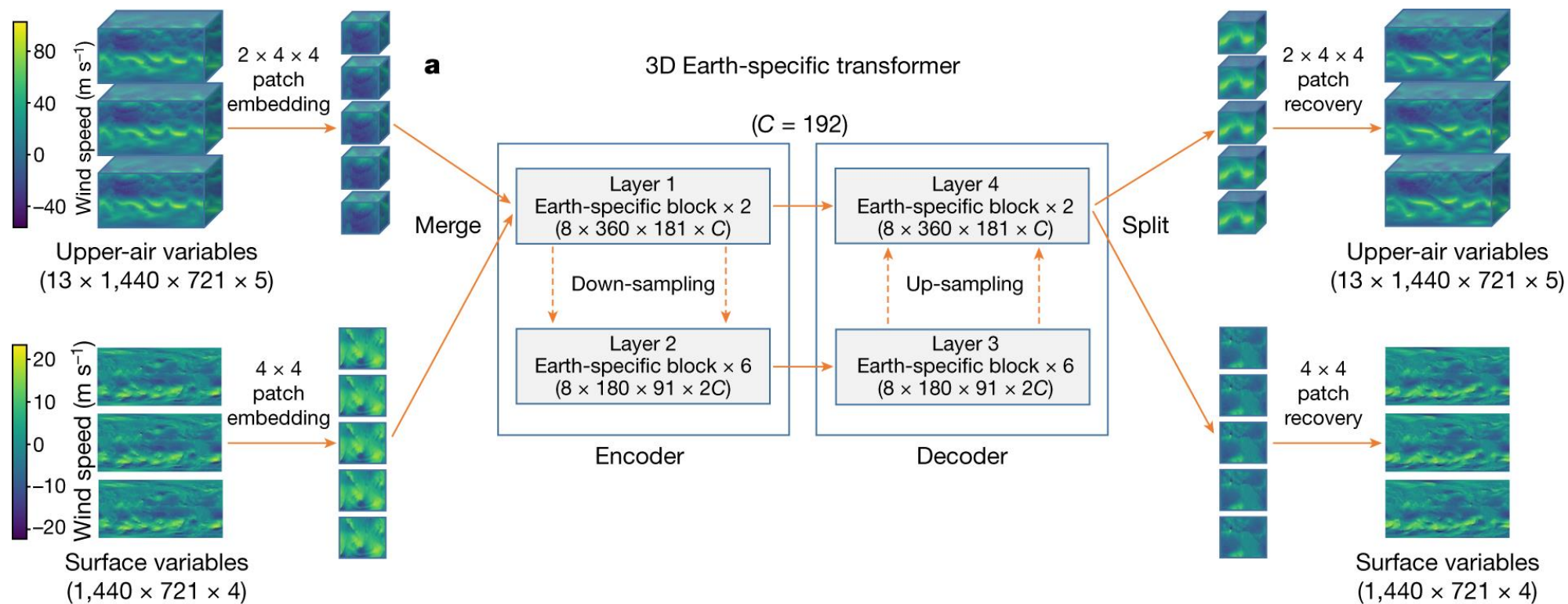
Transformers

- Embed data into “tokens”
- Transform the tokens into:
 - Key
 - Query
 - Value
- Apply Attention Mechanism
- A vision transformer combines (Multi-Head) Attention with MLPs and some other clever tricks
- Attention does not assume locality (CNN) or a sequential order (RNN). This makes it flexible, but data hungry



Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).

Transformers: PanguWeather

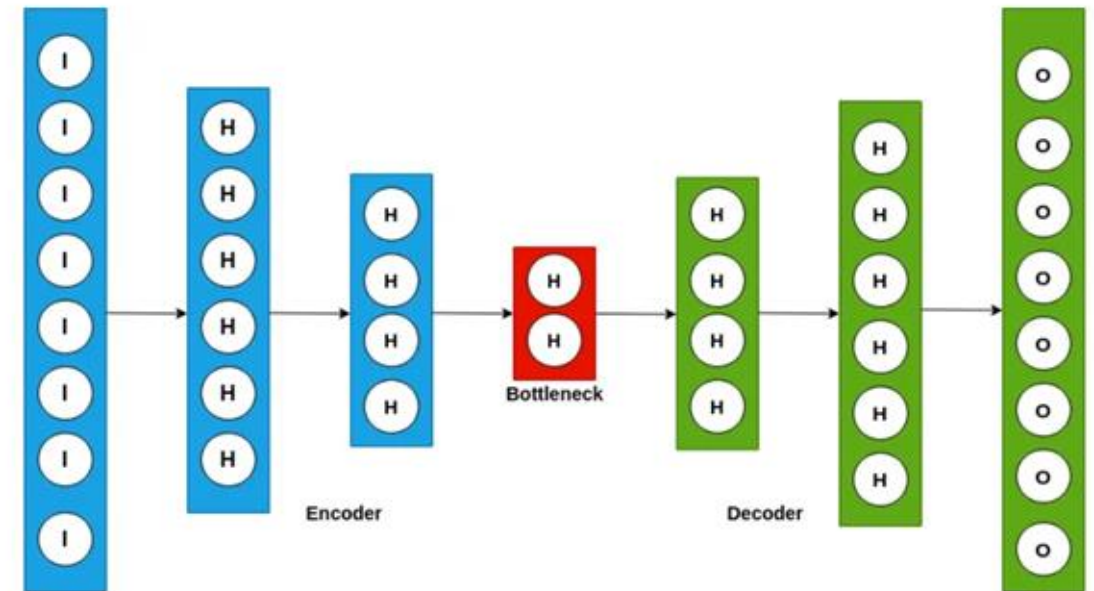


Bi, K., Xie, L., Zhang, H. *et al.* Accurate medium-range global weather forecasting with 3D neural networks. *Nature* **619**, 533–538 (2023). <https://doi.org/10.1038/s41586-023-06185-3>

Autoencoders

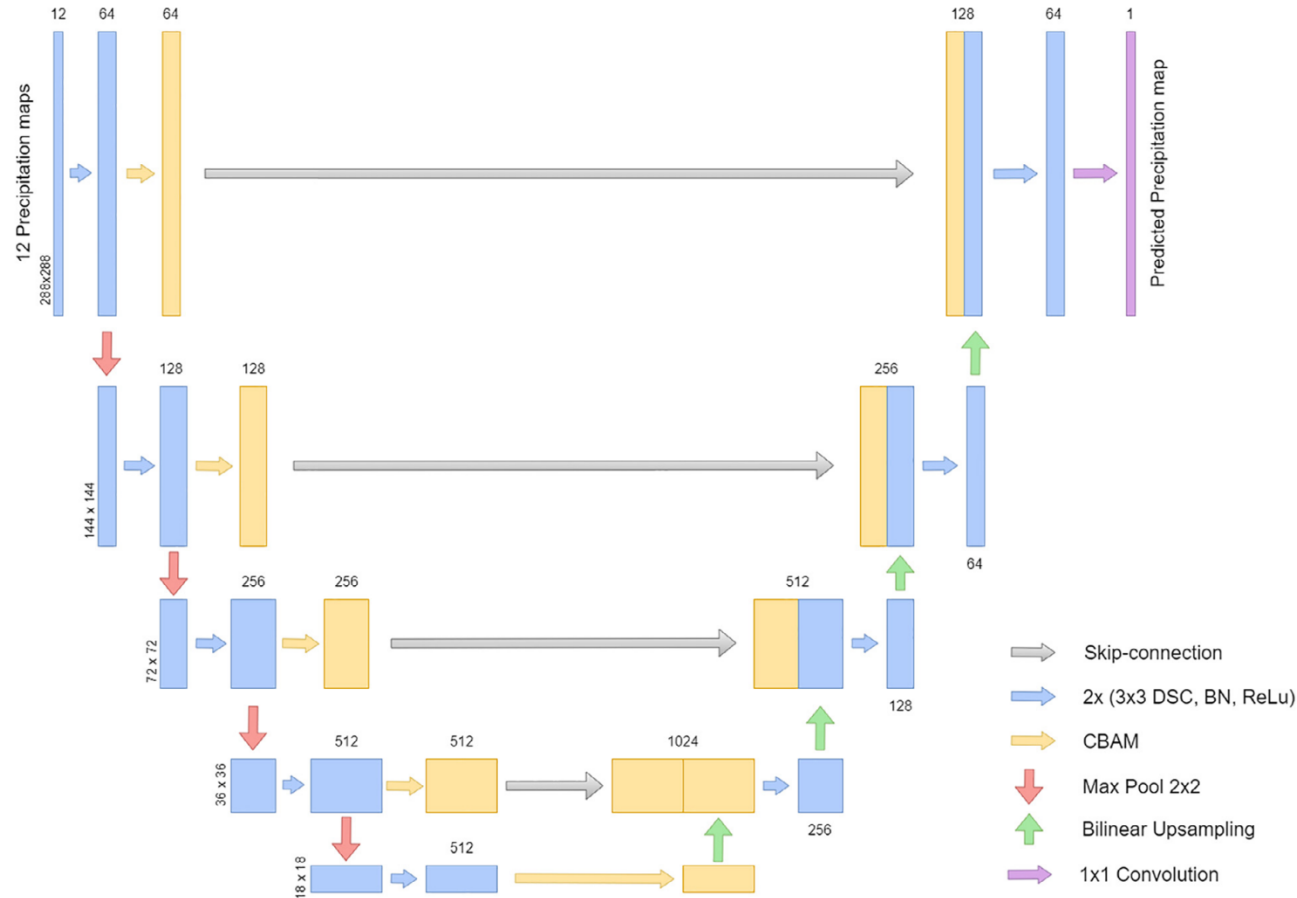
Autoencoders

- **The input dimensionality is reduced by an encoder**
 - Unsupervised Learning: The decoder tries to reconstruct the input
 - Supervised Learning: The decoder produces an altered version of the input
 - exploits underlying correlations among data
- **Can consist of elements from different architectures (MLPs, CNNs, GNNs,...)**



<https://towardsdatascience.com/introduction-to-autoencoders-7a47cf4ef14b>

Autoencoders: UNET



The U-NET has proven to be a very successful and versatile architecture for image-to-image tasks.

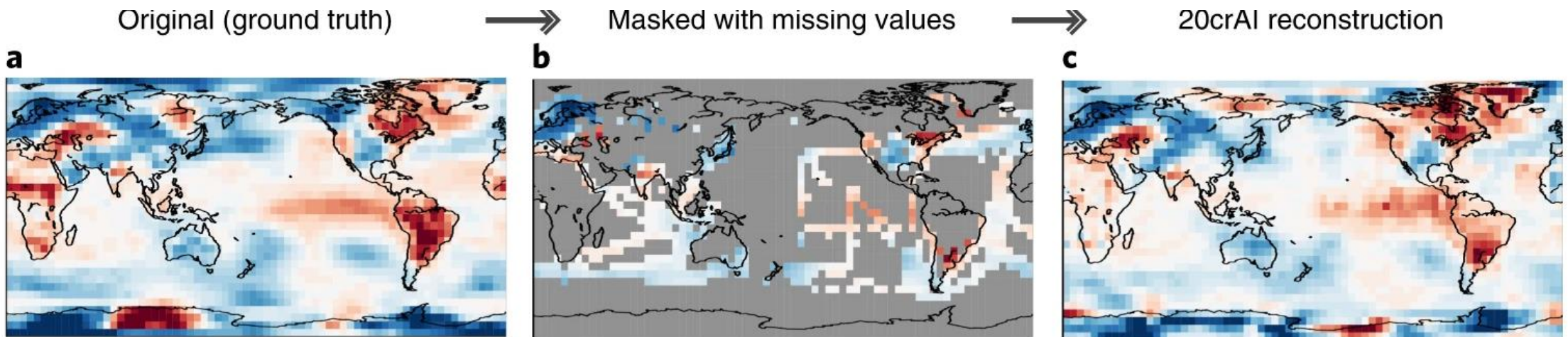
Applications:

- Compression
- Denoising
- Infilling
- Prediction
- Downscaling

Trebing, Kevin, Tomasz Stańczyk, and Siamak Mehrkanoon. "SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture." *Pattern Recognition Letters* 145 (2021): 178-186.

Autoencoders for Reconstruction

- **Reconstructing ("Infilling") on missing sea surface temperature measurements**
- **Outperforms existing statistical methods**
- **Exercise Tomorrow**

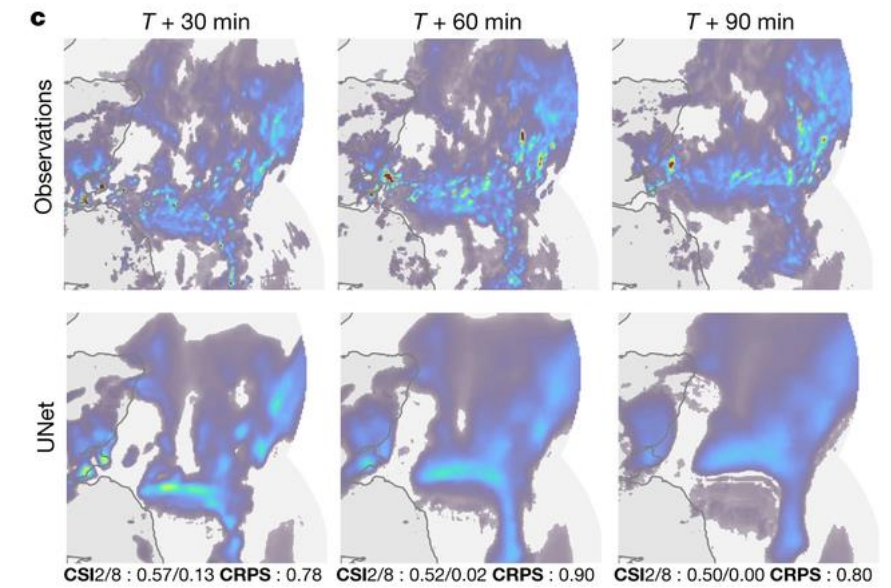


Kadow, C., Hall, D.M. & Ulbrich, U. Artificial intelligence reconstructs missing climate information. *Nat. Geosci.* **13**, 408–413 (2020).
<https://doi.org/10.1038/s41561-020-0582-5>

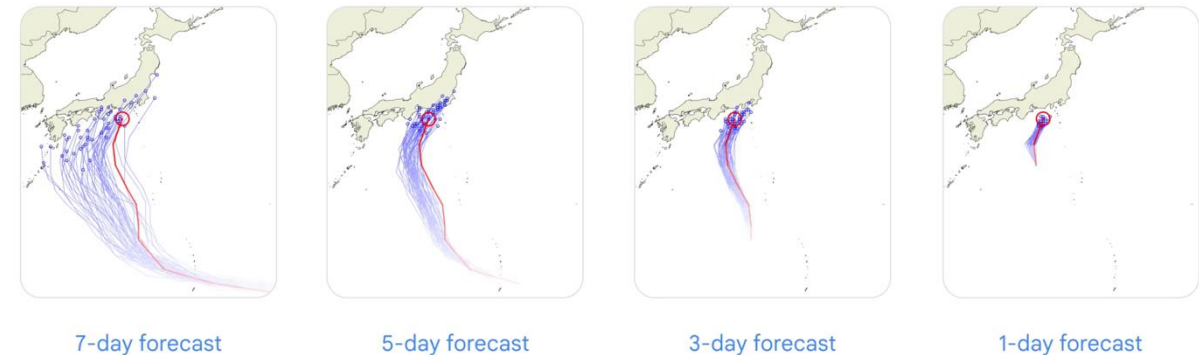
Probabilistic Deep Learning

Probabilistic Deep Learning

- **Standard NNs are overconfident**
- **Many standard architectures trained on RMSE suffer from oversmoothing**
 - In probabilistic deep learning the probability distribution is optimised
- **Probabilistic Methods are used for classical modelling (ensembles)**
 - Risk communication to stakeholders
 - Deep Learning enables extremely large ensembles



Ravuri, S., Lenc, K., Willson, M. *et al.* Skilful precipitation nowcasting using deep generative models of radar. *Nature* **597**, 672–677 (2021).
<https://doi.org/10.1038/s41586-021-03854-z>

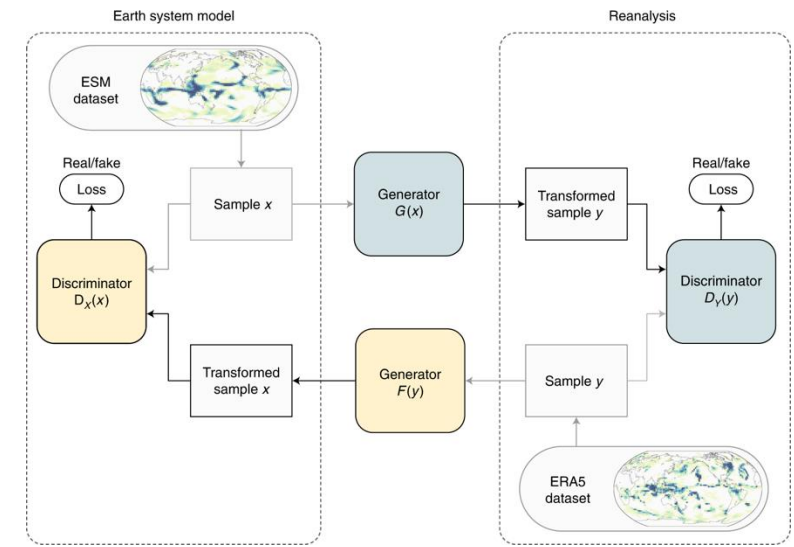


—○— Cyclone track up to October 12, 2019, at 0600 UTC —○— Possible paths predicted by GenCast

Common Methods

- (Cycle-) Generative Adversarial Models
- Variational Autoencoder
- Dropout and Batchnorm Uncertainty
- Bayesian Neural Networks
 - Now
- Deep Ensembles
 - Now
- Diffusion Models
 - Thursday

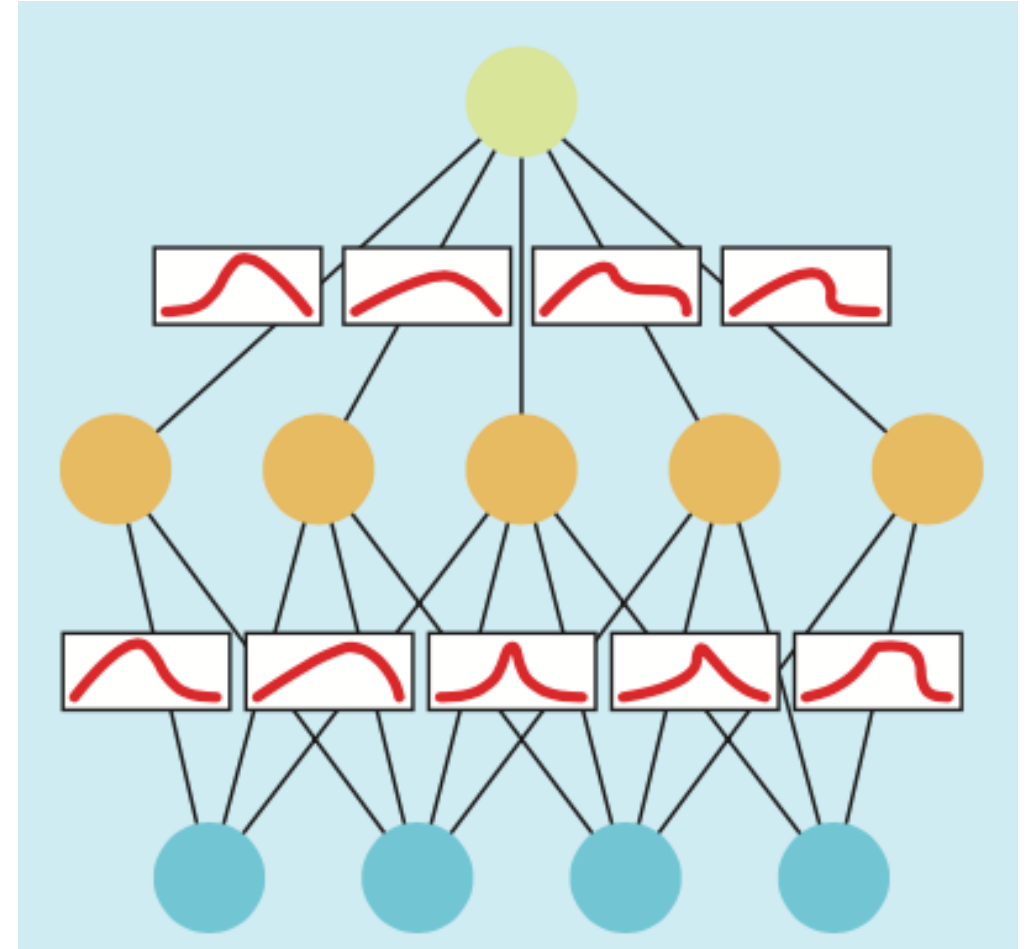
All these methods are (more or less) agnostic to the chosen architecture!



Hess, P., Drüke, M., Petri, S. *et al.* Physically constrained generative adversarial networks for improving precipitation fields from Earth system models. *Nat Mach Intell* 4, 828–839 (2022). <https://doi.org/10.1038/s42256-022-00540-1>

Bayesian Neural Networks

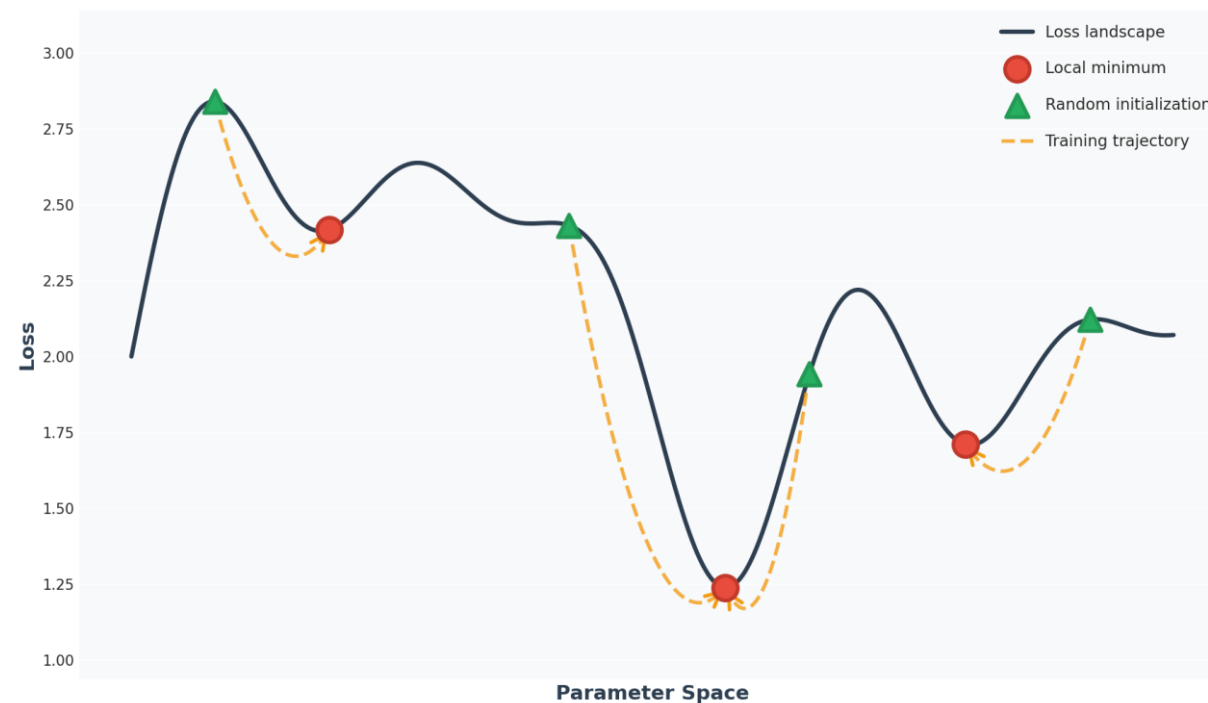
- **Weights are stochastic**
- **The output is also stochastic and therefore allows an uncertainty estimate for the prediction**
- **Training Routine is built on Bayes theorem and is fundamentally different than standard deep learning architectures**
 - Weights sampled with e.g. Markov-Chain Monte Carlo Methods
 - Weights optimised with Variational Inference Methods
- **Tricky and expensive to train, have not seen mainstream adoption**
- **Python Packages: Pyro, Bayesian-Torch, TorchUncertainty**



Jospin, Laurent Valentin, et al. "Hands-on Bayesian neural networks—A tutorial for deep learning users." *IEEE Computational Intelligence Magazine* 17.2 (2022): 29-48.

Deep Ensembles

- An ensemble of deep learning models
- Trained on the same data with the same architecture but initialised with different random weights
- Sample different minima of the loss landscape
- Compete or outperform Bayesian Neural Network approaches for many cases (Wilson and Izmailov, 2021)
- Recommended reading:
<https://cims.nyu.edu/~andrewgw/deepestmbles/>

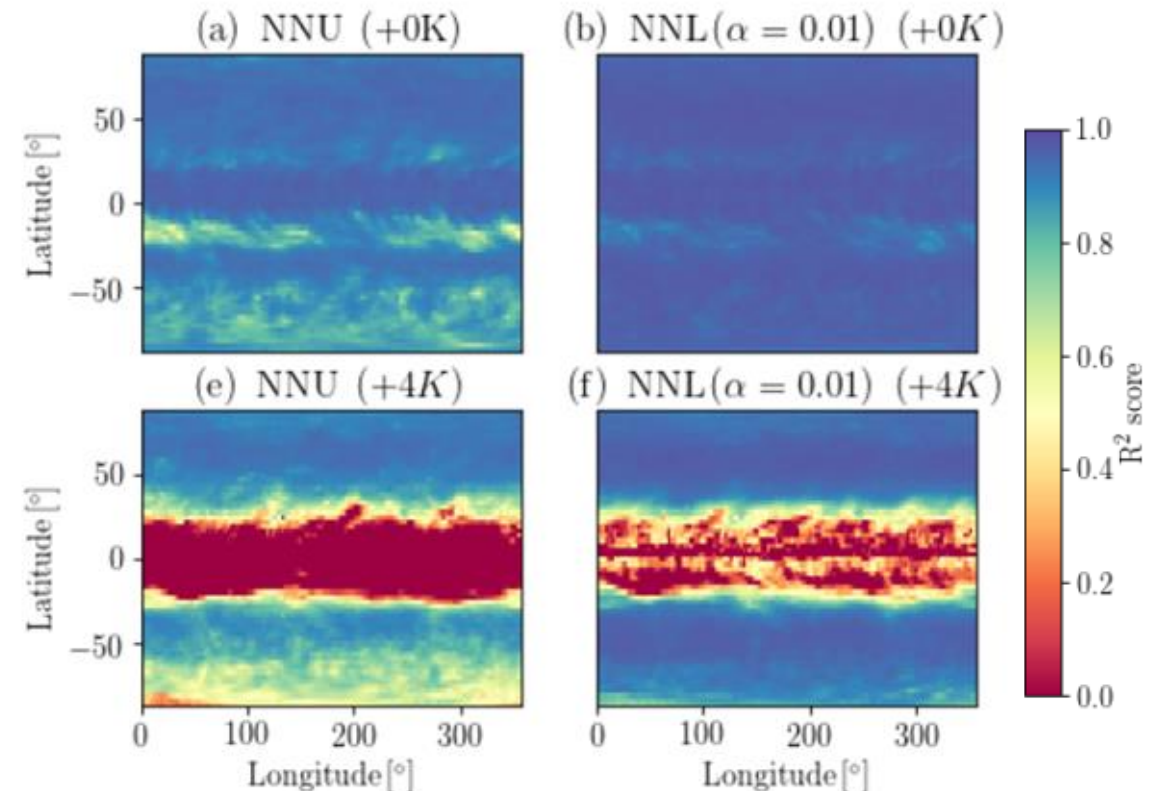


Physics-Informed Deep Learning

Physics-Informed Deep Learning

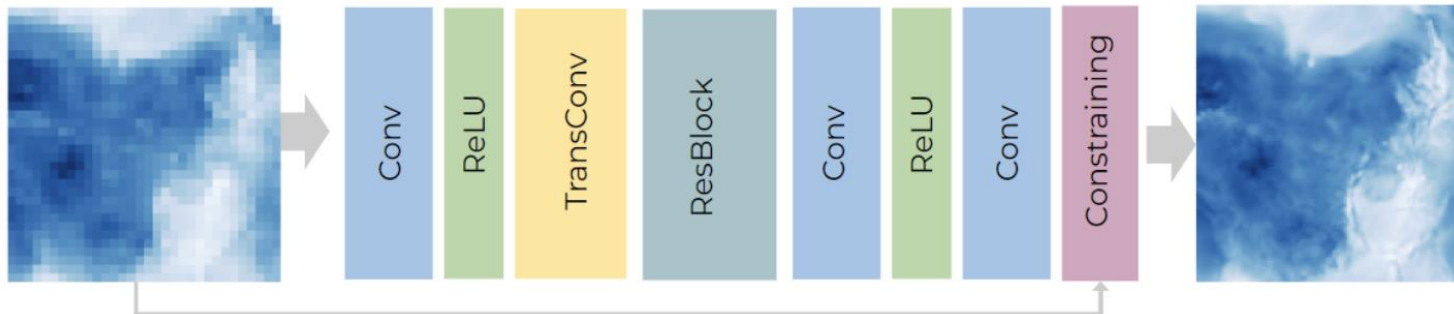
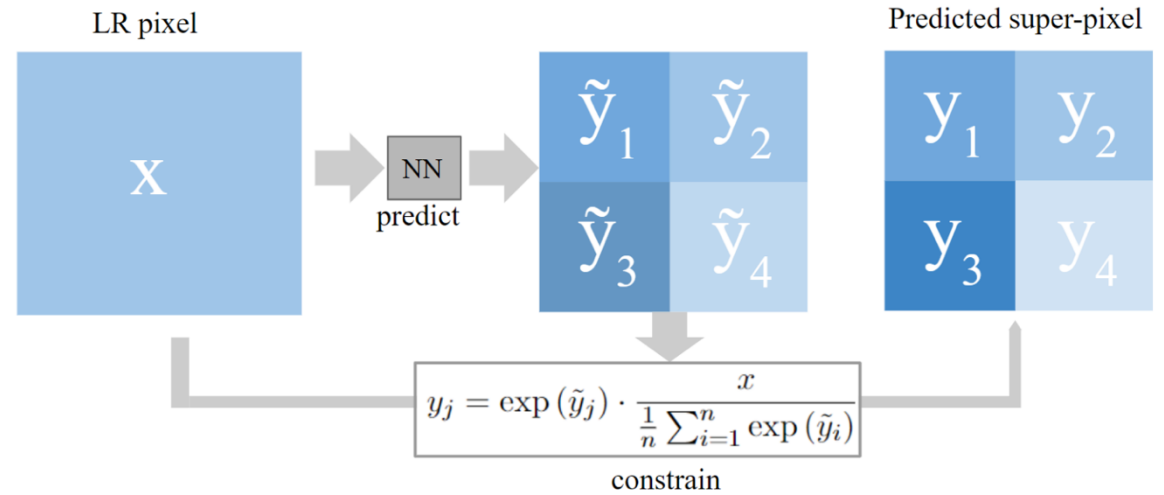
- **Idea: Force your Deep Learning model to conserve energy/mass or apply another physical constraint**
- **Typically achieved by**
 - constraining the model architecture
 - modifying the loss function
- **Can help with applications beyond of the training sample**

$$\mathcal{L}(\alpha) = \alpha \mathcal{P}(x, y_{\text{NN}}) + (1 - \alpha) \text{MSE}(y, y_{\text{NN}})$$



Physics-Informed Deep Learning

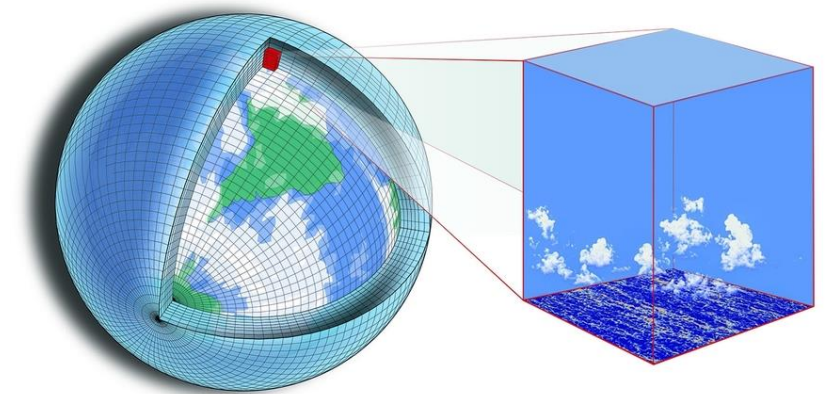
- **Hard Constraint: Enforcing conservation laws in the final layer of the network**
- **Improves overall performance.**



Harder, Paula, et al. "Hard-constrained deep learning for climate downscaling." *Journal of Machine Learning Research* 24.365 (2023): 1-40.

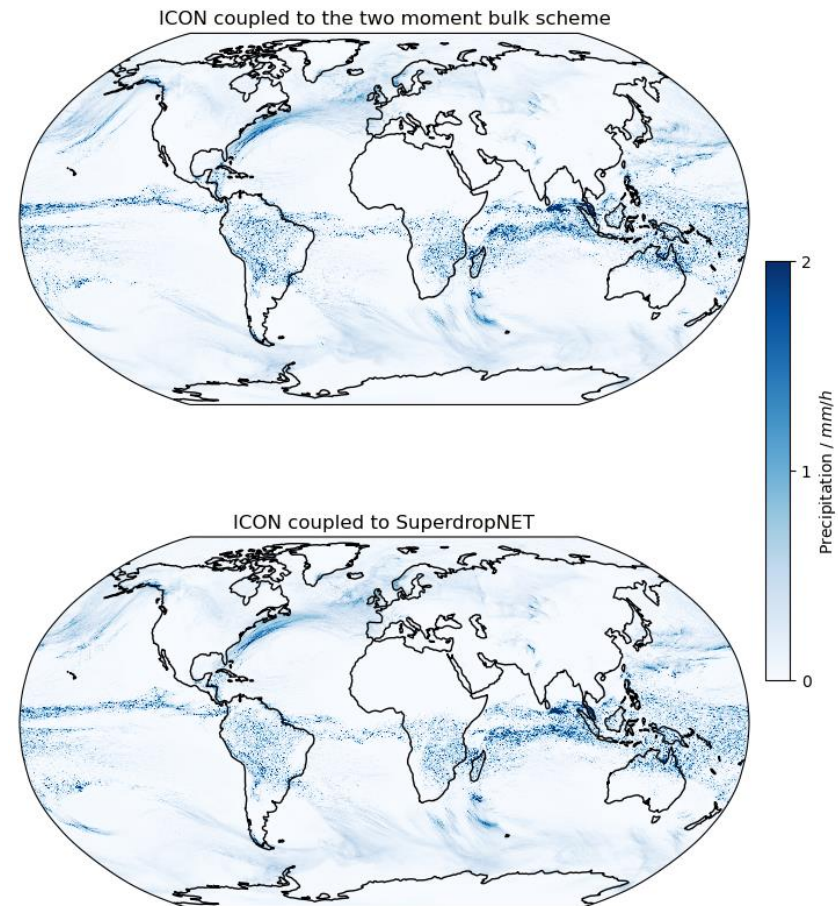
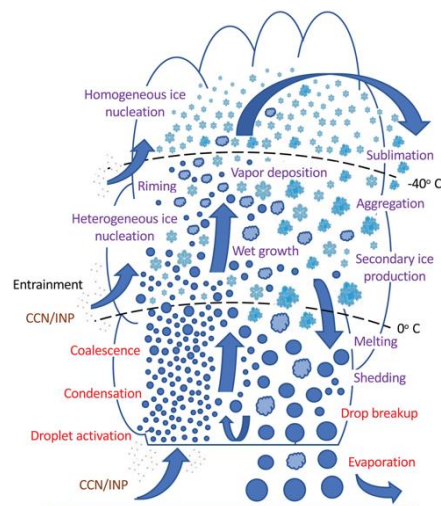
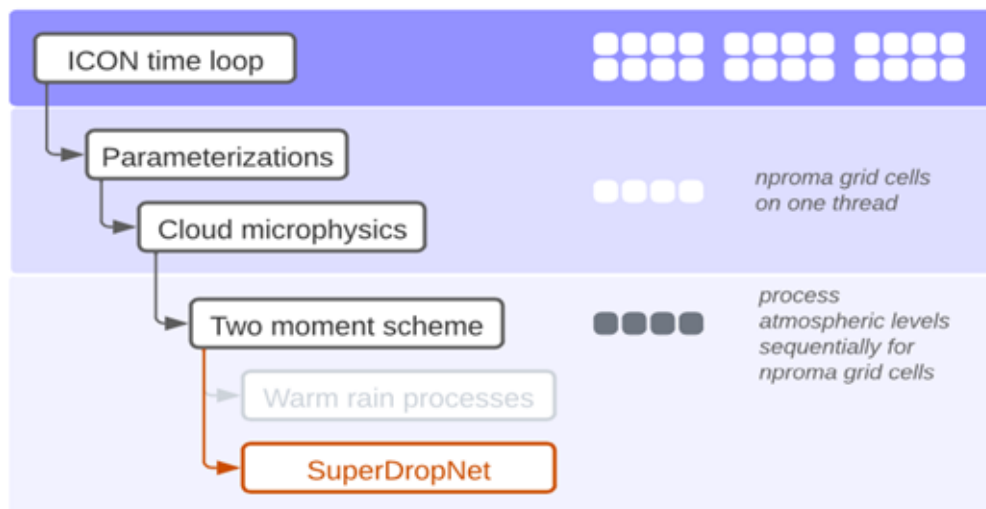
Coupling General Circulation Models with ML (“Hybrid Modelling“)

- **Parameterisations are the largest source of error in general circulation models.**
- **Potential for deep learning:**
 - Replace parameterisations with more accurate ML-based relationships that are learned from observations.
 - Replace computationally expensive components with ML
- **Challenges:**
 - Efficient Technical implementation of python code into FORTRAN or C++
 - Offline vs online behaviour of ML model
 - Application to problems outside of training data
 - Data regime
 - Resolution / time stepping
 - Long-term Stability



<https://news.mit.edu/2018/new-climate-modeling-alliance-clima-1212>

Simulating rain in ICON with a neural network trained on superdroplet simulations



Arnold, Caroline, et al. "Efficient and stable coupling of the SuperdropNet deep-learning-based cloud microphysics (v0. 1.0) with the ICON climate and weather model (v2. 6.5)." *Geoscientific Model Development* 17.9 (2024): 4017-4029.



<https://cambridge-iccs.github.io/FTorch/>



Fully Differentiable General Circulation Models

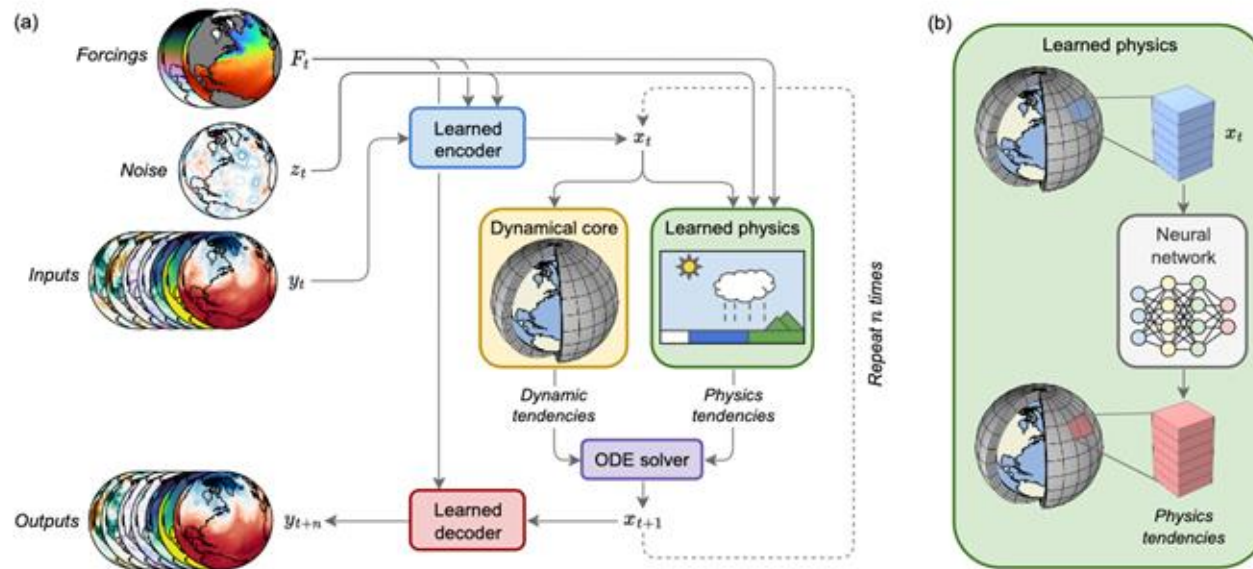


Fig. 1 Structure of the NeuralGCM model. (a) Overall model structure, showing how forcings F_t , noise z_t (for stochastic models), and inputs y_t are encoded into the model state x_t . Model state is fed into the dynamical core, and alongside forcings and noise into the learned physics module. This produces tendencies (rates of change) used by an implicit-explicit ODE solver to advance the state in time. The new model state x_{t+1} can then be fed back into another time step, or decoded into model predictions. (b) Inset of the learned physics module, which feeds data for individual columns of the atmosphere into a neural network used to produce physics tendencies in that vertical column.

- Online Learning of NN parameterisations is possible if backpropagation can be calculated through the whole model
- NEURAL GCM (Kochkov et al, 2024)
- PseudospectralNet (Gelbrecht et al, 2025)
- Improves Long term stability
- Extension to climate simulations (?)

Kochkov, Dmitrii, et al. "Neural general circulation models for weather and climate." *Nature* 632.8027 (2024): 1060-1066.

Gelbrecht, Maximilian, Milan Klöwer, and Niklas Boers. "PseudospectralNet: Toward hybrid atmospheric models for climate simulations." *Journal of Advances in Modeling Earth Systems* 17.10 (2025): e2025MS004969.

Explainable AI

Explainable AI

- **A flaw of DL models is the inability of humans to understand decisions and predictions**
- **XAI is used to improve:**
 - Transparency
 - Error detection
 - Ethical compliance
- **In climate science XAI can help with validation and provide new insights into mechanisms**

Explainable AI

- **Explanation target:**
 - Local, individual samples
 - Global, aggregated datapoints
- **Explanation output:**
 - Sensitivity
 - Feature contribution
- **Different XAI methods lead to different explanations!**

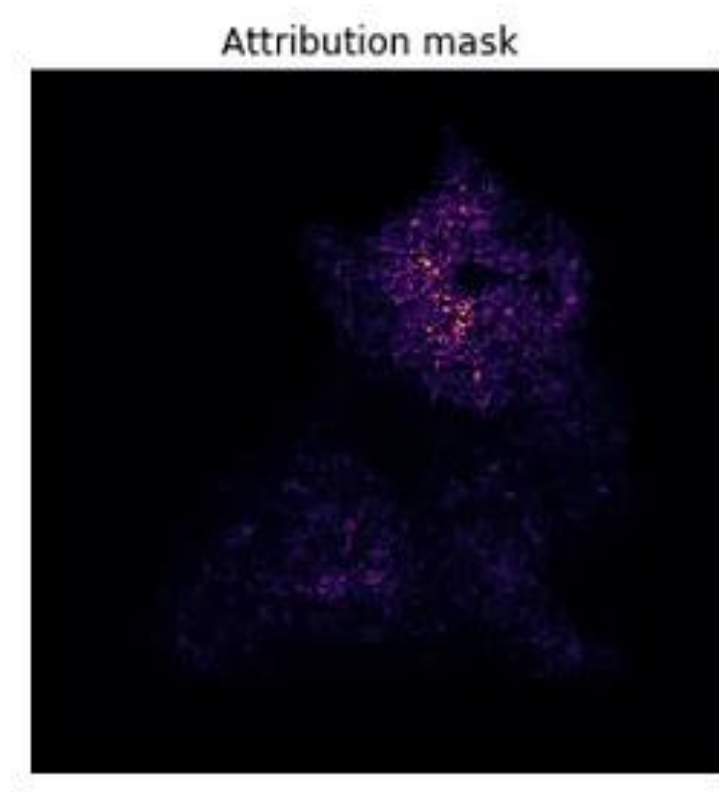
Explainable AI: Integrated Gradients

- Explanation target: local; explanation output: feature contribution
- Integrated Gradients works by computing the integral of the gradients of the output with respect to the input along a straight-line path from a baseline input to the actual input.

$$IG = (x - x_0) \int_0^1 \nabla F(\alpha x + (1 - \alpha)x_0) d\alpha$$

- The “line” is defined in the feature space
- The baseline x_0 is some reference (mean or zeros)
- The gradient of the model F is calculated for every input feature of x , which results in a gradient that has the same dimensions as your input

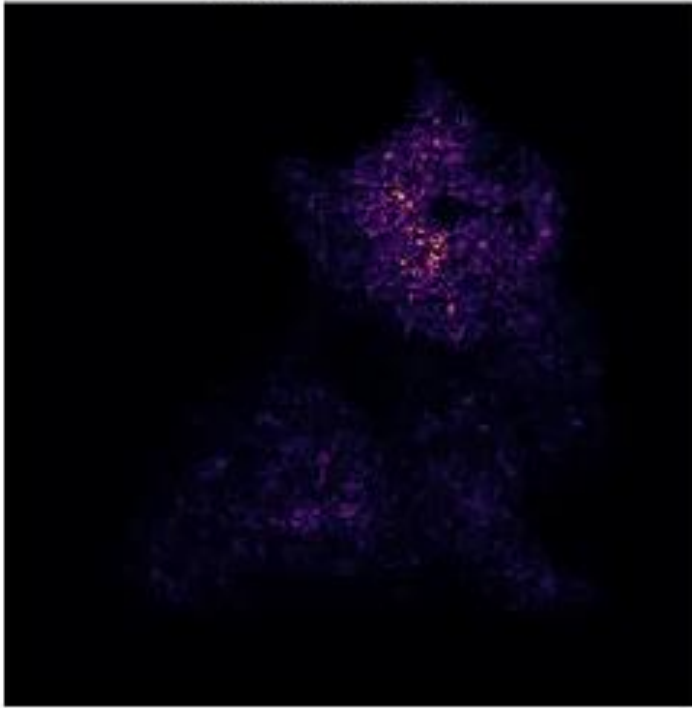
Explainable AI: Integrated Gradients



<https://www.xaifoundation.org/copy-of-lrp>

Explainable AI: Integrated Gradients

Attribution mask



Overlay IG on Input image

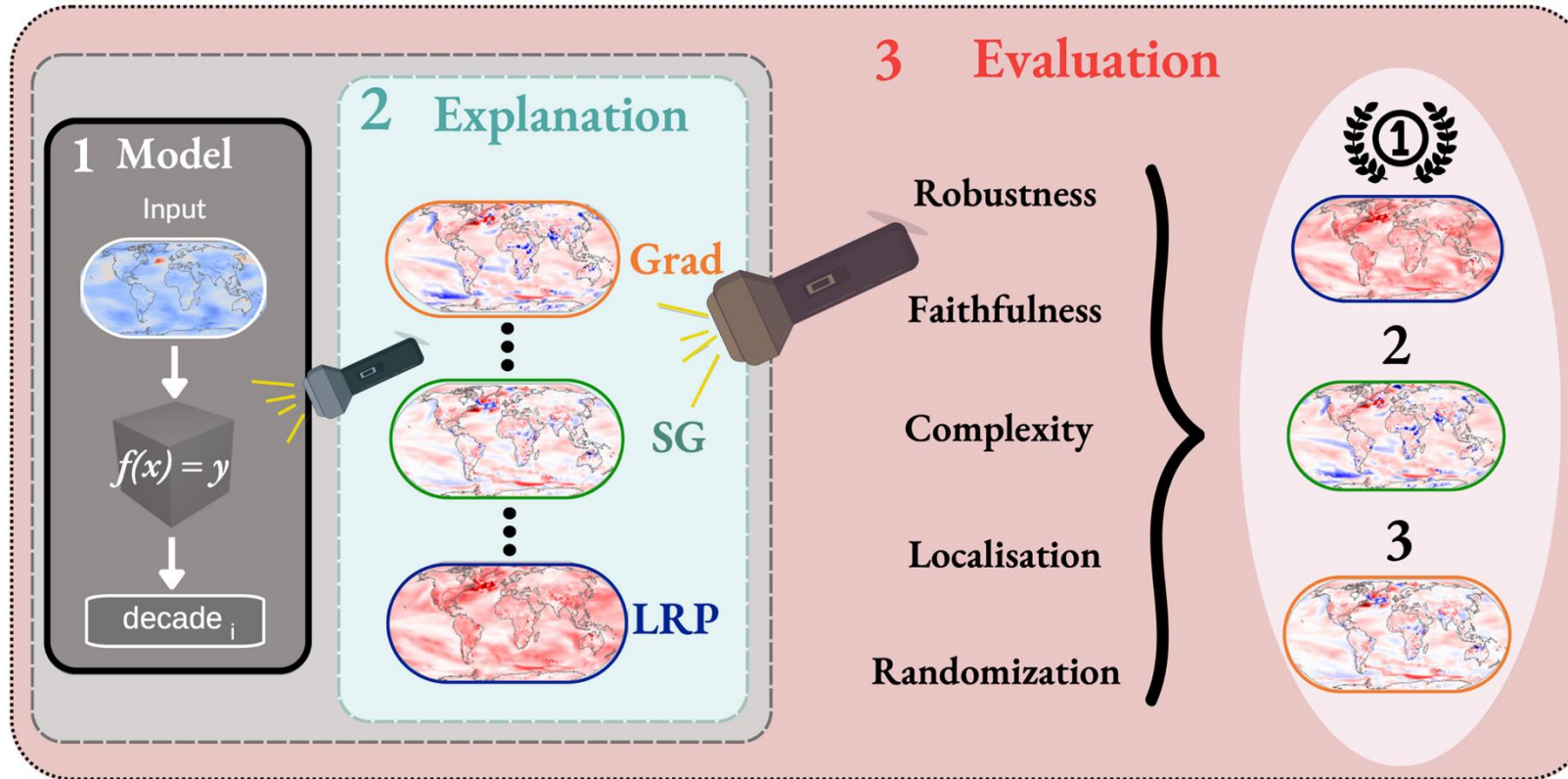


<https://www.xaifoundation.org/copy-of-lrp>

Explainable AI: Other methods

- **Gradient**
- **Input times Gradient**
- **Layerwise Relevance Propagation**
- **SmoothGrad, NoiseGrad, FusionGrad**
- **SHAP/DeepSHAP**

Explainable AI: Overview for Climate Science



For a good in-depth overview read this paper!

Bommer, Philine Lou, et al. "Finding the right XAI method—A guide for the evaluation and ranking of explainable AI methods in climate science." *Artificial Intelligence for the Earth Systems* 3.3 (2024): e230074.

Conclusion

- **Fast moving science, state of the art changes every year**
- **Find what works best for you**
- **Think about your data structure and your use case**
- **In theory a NN can learn any function, in practice you have to help the NN with design choices**
- **The right architecture is not everything**
 - High quality, meaningful datasets and a use-case that makes sense are the most important ingredients

Conclusion

- Fast moving science, state of the art changes every year
- Find what works best for you
- Think about your data structure and your use case
- In theory a NN can learn any function, in practice you have to help the NN with design choices
- The right architecture is not everything
 - High quality, meaningful datasets and a use-case that makes sense are the most important ingredients

Thank you for your attention!