Architectures and Applications

Deep Learning for Climate Scientists

10th-12th March 2026

Paul Keil

paul.keil@hereon.de

Helmholtz AI Consultants

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:

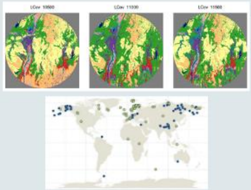
ML-GCM Coupling, AI for remote sensing, Uncertainty Quantification, LLMs for Science, …

Tobias Weigel, Danu Caus, Paul Keil, Caroline Arnold, Harsh Grover

Email: consultant-helmholtz.ai@hereon.de

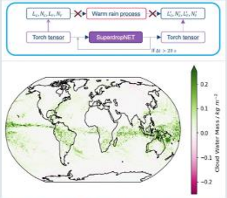
Helmholtz AI: Some of our Projects

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



Danu, Tobias

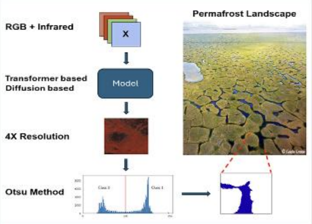
Coupling ICON with a ML Emulator for Cloud Microphysics



ICON coupled with SuperdopNET produces more cloud water mass

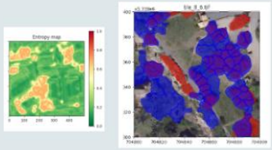
Caroline, Paul

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



Danu, Harsh

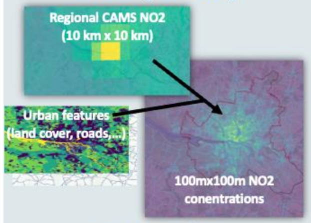
DeepTrees: Individual Tree Crown Delineation from Digital Orthophotos (UFZ)



Active Learning based on entropy

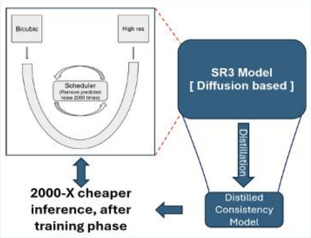
Caroline, Harsh

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



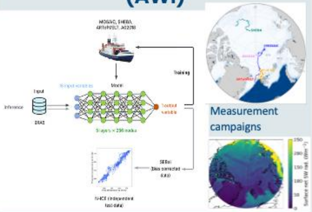
Paul, Tobias

Model distillation via consistency models (GFZ)



Danu

Correcting Arctic surface energy budget using ANNs (AWI)

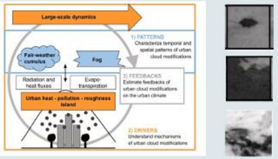


Workflow: NN trained on measurements (point locations)

Application of NN to whole Arctic

Paul, Harsh

Satellite-based detection of urban cloud holes (KIT)



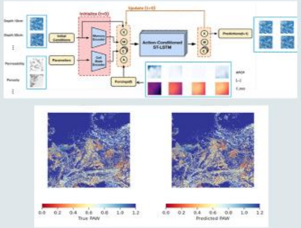
CNN for binary classification: 85% F1 Score

Caroline

Sea2Land navigator turned chatbot (GERICS)

Harsh, Tobias

Drought Analytics (FZJ)



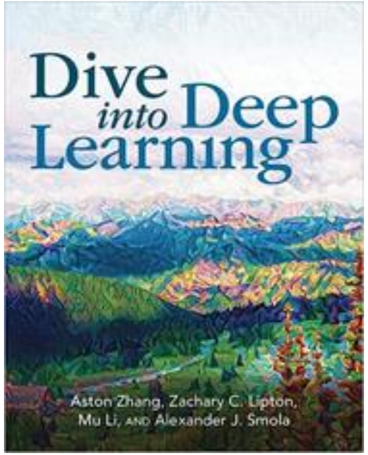
Caroline, Paul

What to expect

- A look into the toolbox
- Some math, but can‘t go too deep
- 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



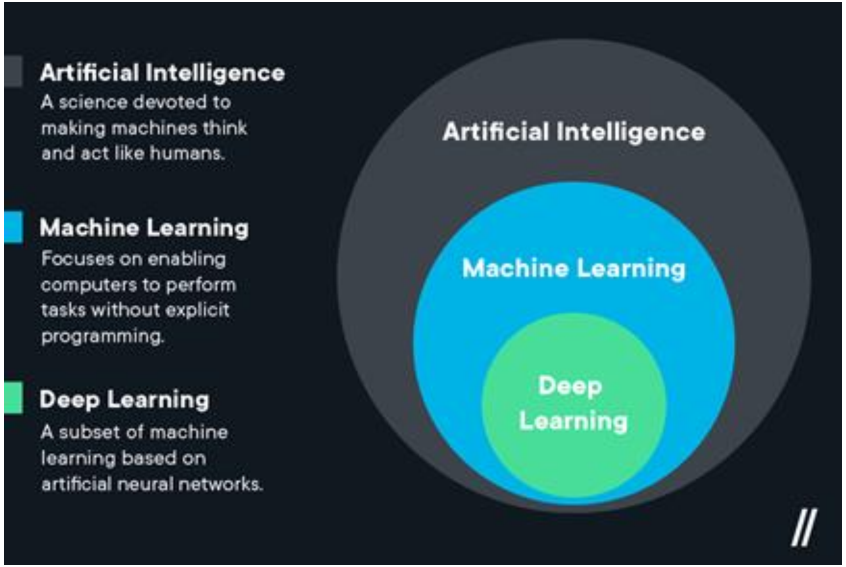
Outline

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

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 adequat and powerful.

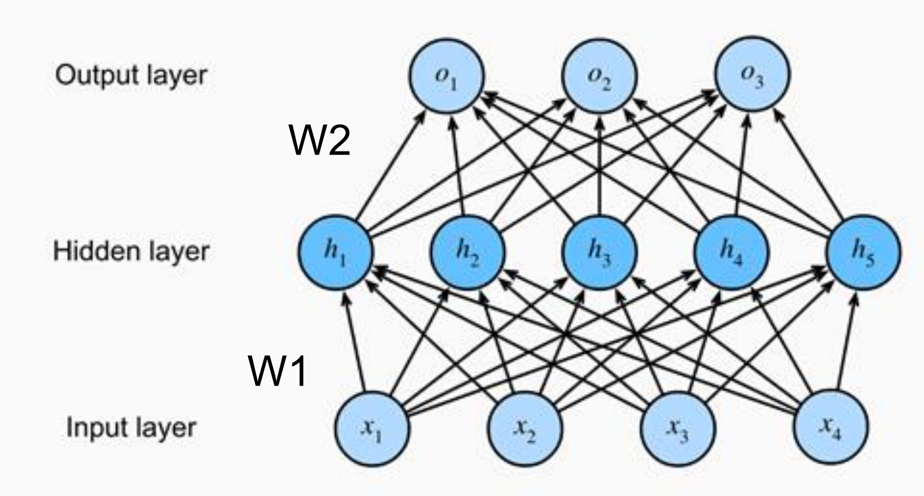


https://flatironschool.com/blog/deep-learning-vs-machine-learning/

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



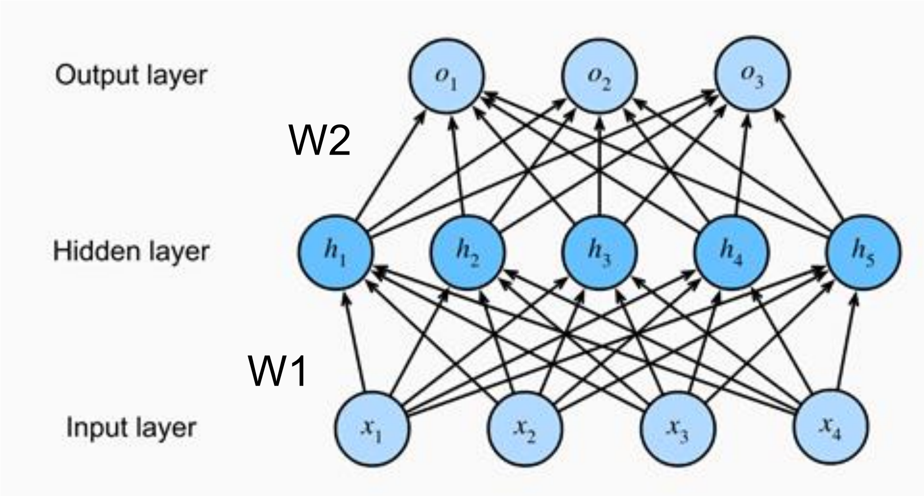
W: weight matrix
b: biases
sigma: nonlinear function

X: input vector
H: hidden or “latent” state
O: output

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

Multilayer Perceptrons

- Weights and biases are the parameters that are “learned”
- input x has “4 features”
- The number of hidden layers and the amount of neurons can be chosen
- Deep Learning is mostly just linear Algebra and some Calculus.



$$
\begin{aligned}
\mathbf{H} &= \sigma(\mathbf{X}\mathbf{W}^{(1)} + \mathbf{b}^{(1)}), \\
\mathbf{O} &= \mathbf{H}\mathbf{W}^{(2)} + \mathbf{b}^{(2)}.
\end{aligned}
$$

Activation Functions

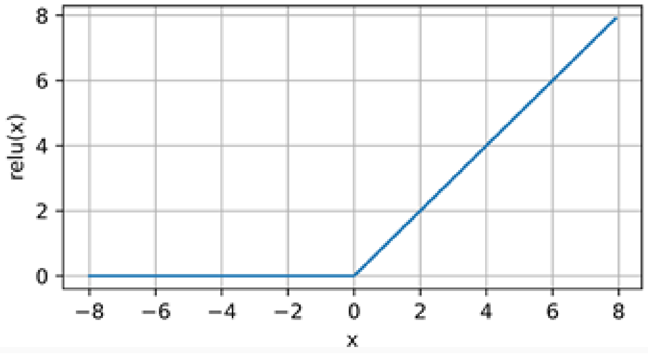
"Add non-linearity". Neurons are activated ("they fire") like brain neurons

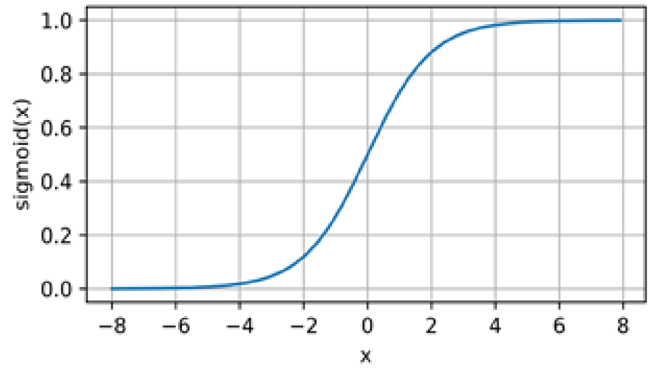
- ReLU $\mathrm{ReLU}(x) = \max(x, 0).$

  simple and good performance
  behaves "well" during backpropagation

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

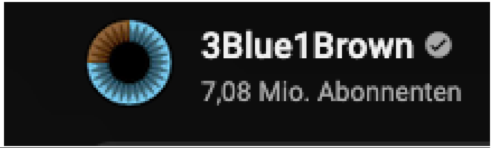
  important for some applications

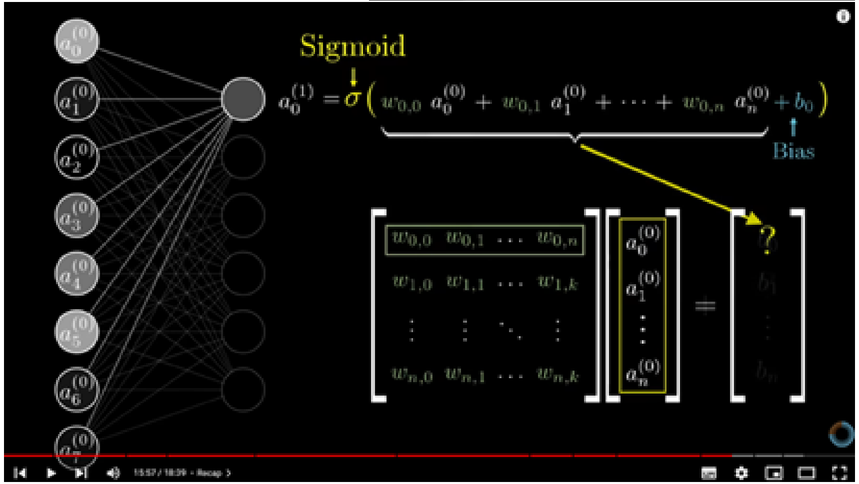




Multilayer Perceptrons

A great visual introduction:





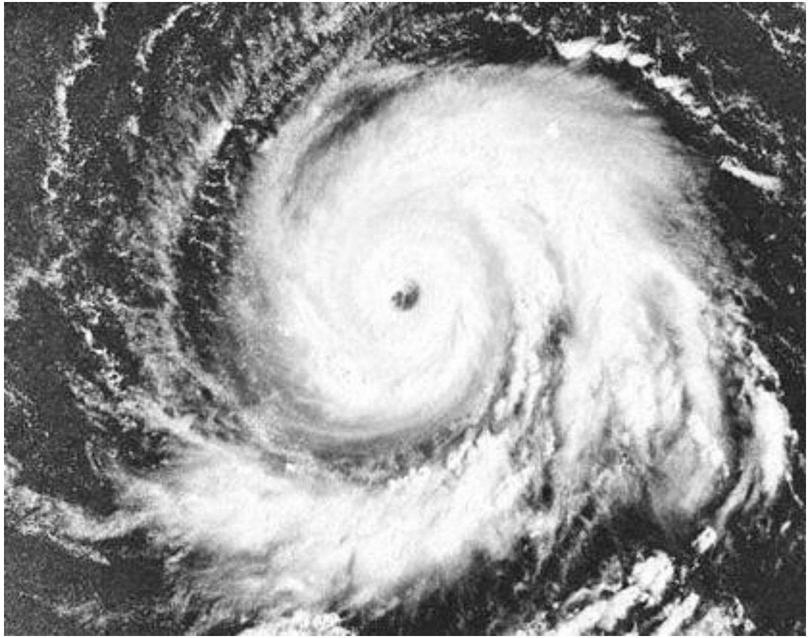
https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi

Illustration: Hurricane Classification

General

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?



Thanks to C. Kadow for the following slides

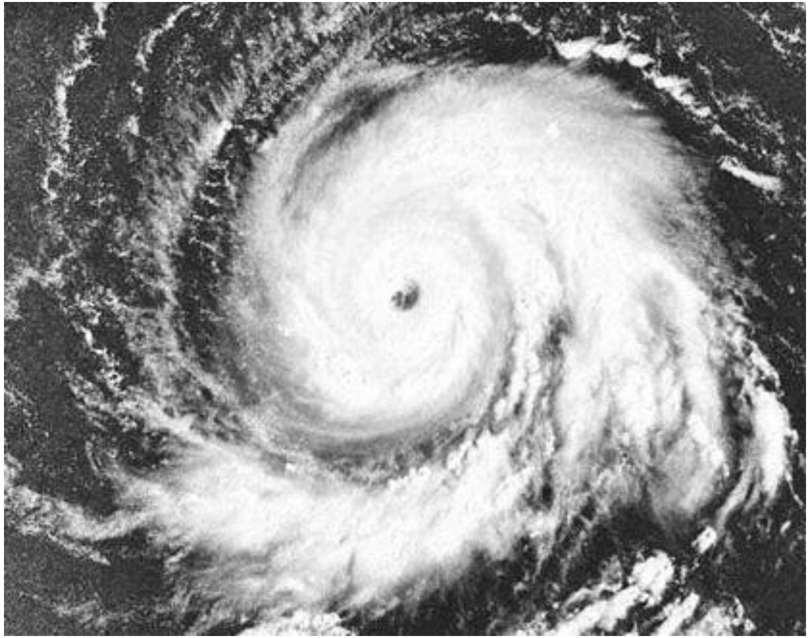
General

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.

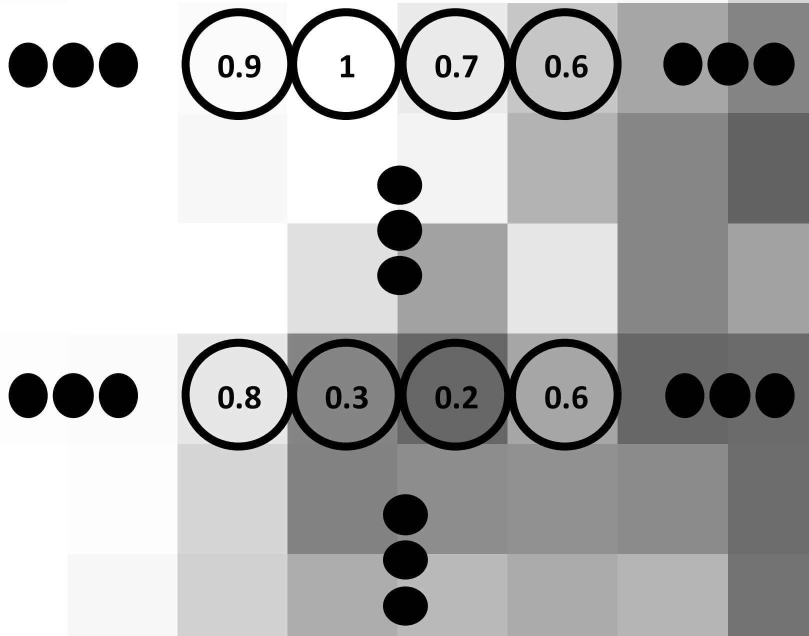


Thanks to C. Kadow for the following slides

General

Inputs

Training (and Validation) Dataset



General

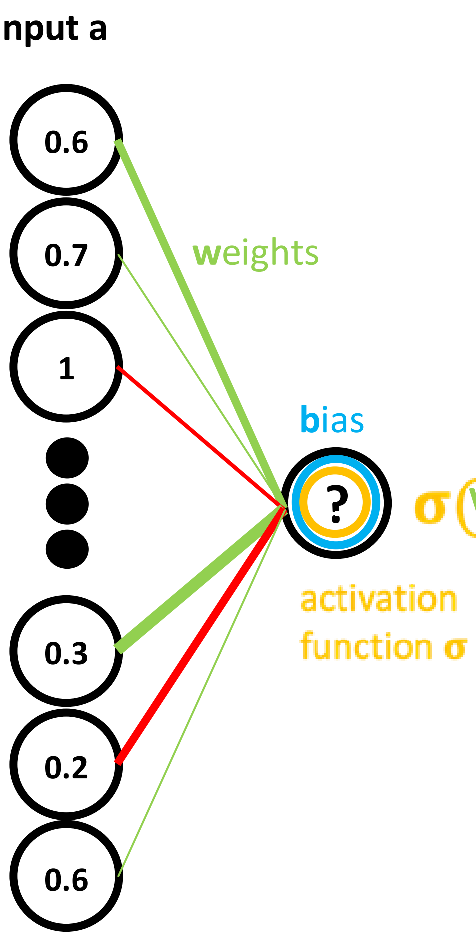
Inputs

Training, Validation, Test Set

Nodes, Weights, Biases

A network with one layer that contains one node

Activation Function



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

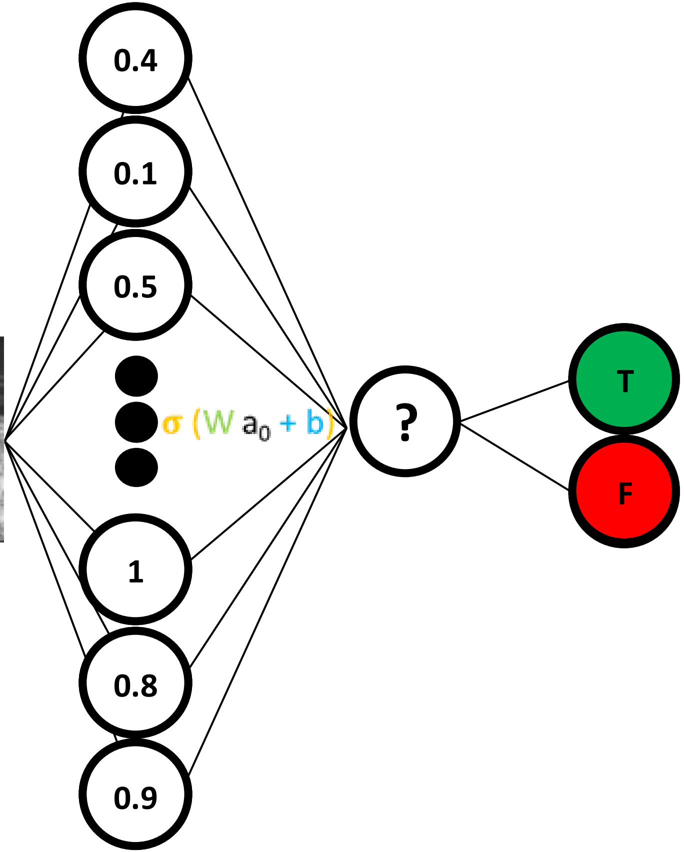
General

Inputs

Training, Validation, Test Set

Nodes, Weights, Biases

Activation Function



Outputs

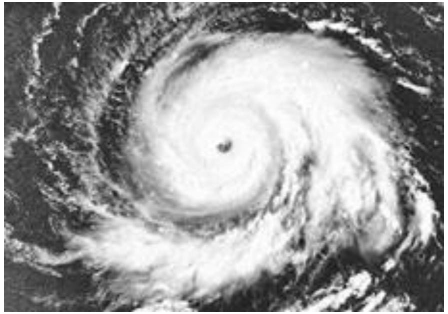
The output of the neural network can be a bounded to a real value between 0 and 1 (classification) or any value (regression).

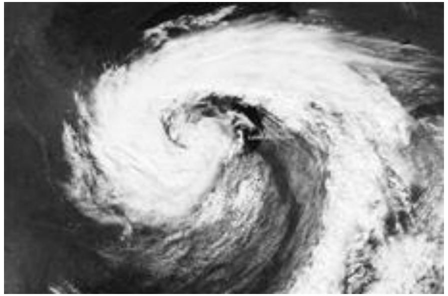
Hurricane Classification with an MLP

Would you expect this to work?

No, the MLP does not understand the spatial structure of the Hurricane.

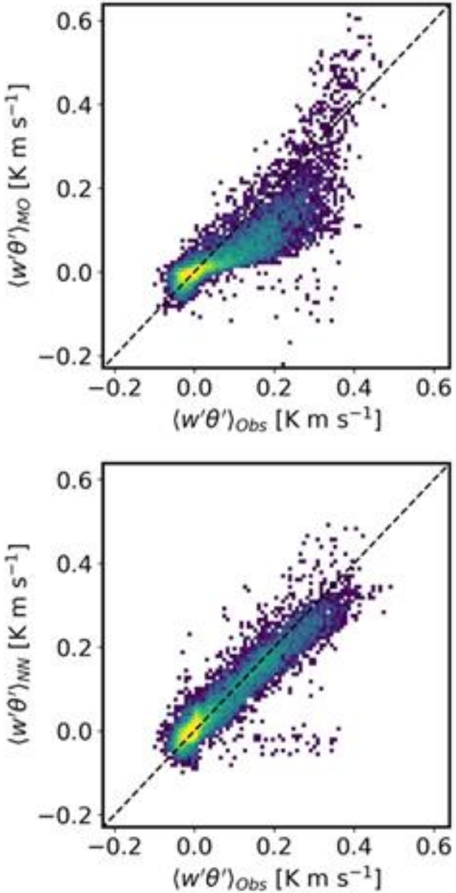
Any more questions?





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

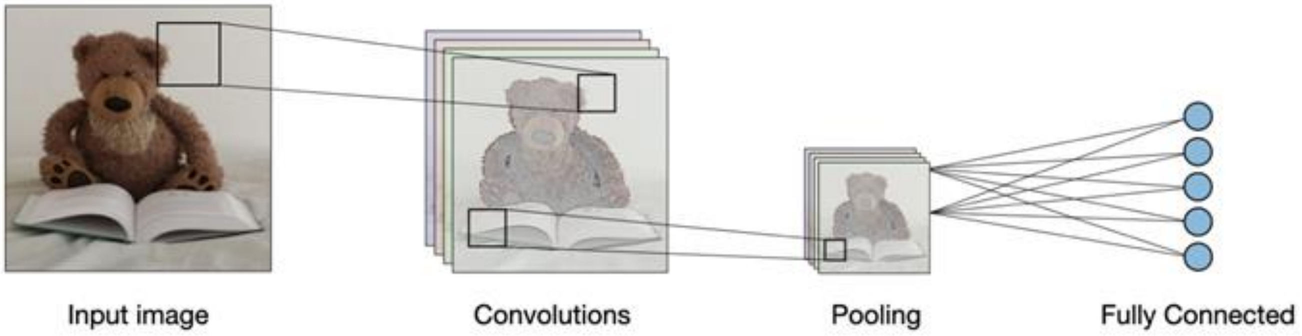


Muñoz-Esparza, Domingo, et al. "On the Application of an Observations-Based Machine Learning Parameterization of Surface Layer Fluxes Within an Atmospheric Large-Eddy Simulation Model." *Journal of Geophysical Research: Atmospheres* 127.16 (2022): e2021JD036214.

Convolutional Neural Networks

Convolutional Neural Networks

- Multilayer Perceptrons do not account for data structure
- Convolutional Neural Networks solve this problem
- Popular for any 2-dimensional data, especially image classification tasks
- You will be programming one later



1. Convolution

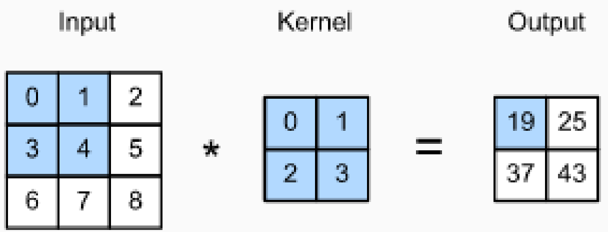
$$[\mathbf{H}]_{i,j} = u + \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} [\mathbf{V}]_{a,b} [\mathbf{X}]_{i+a,j+b}.$$

i,j : pixel location

V: weight matrix of kernel
u: bias
Δ: kernel size

a,b: kernel indices
X: input

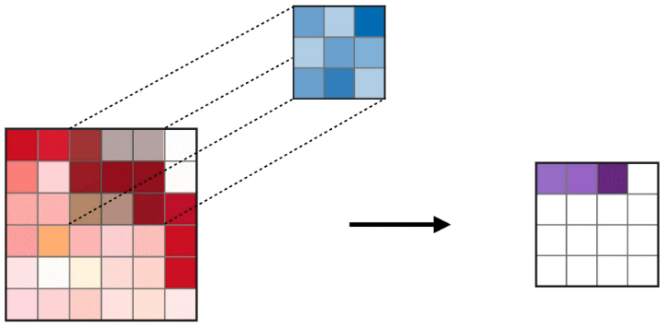
H: hidden or “latent” state



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

1. Convolution

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

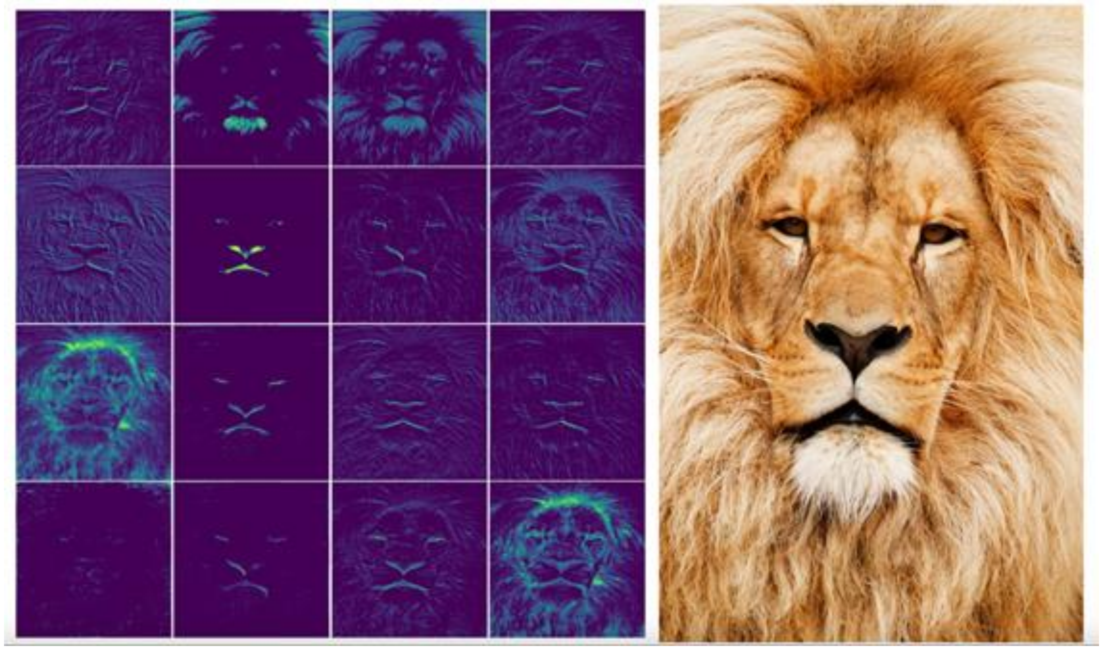


https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

- 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?

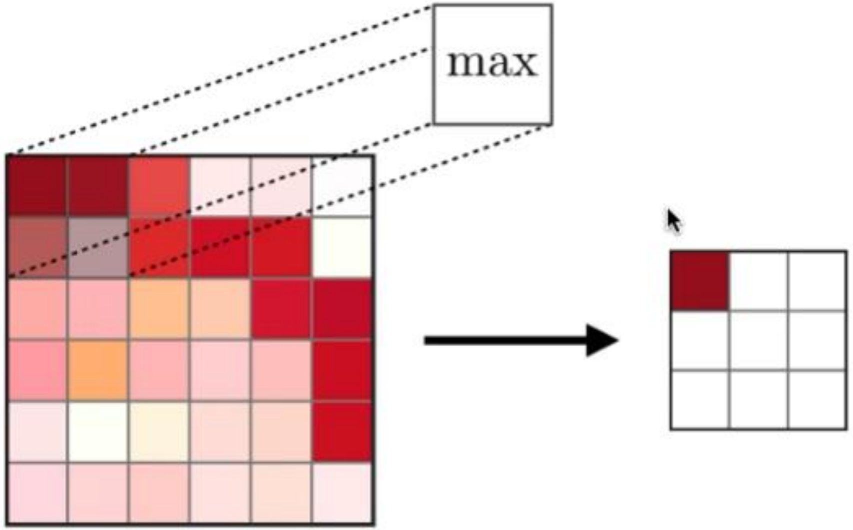
convolutions input



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

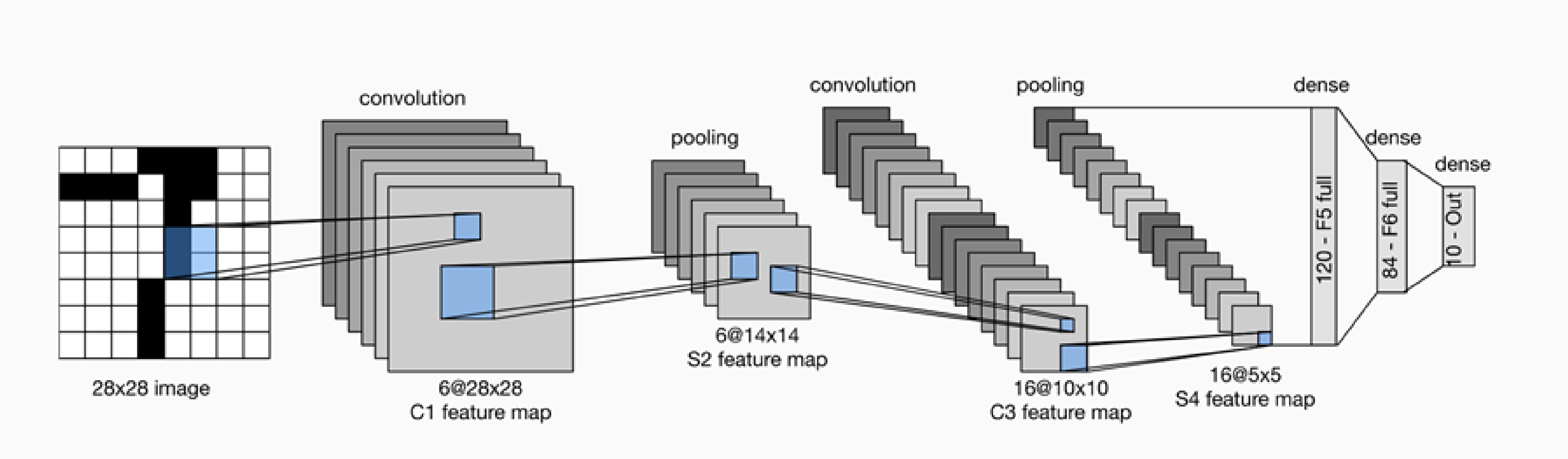
2. Pooling

A downsampling operation typically after the convolution layer

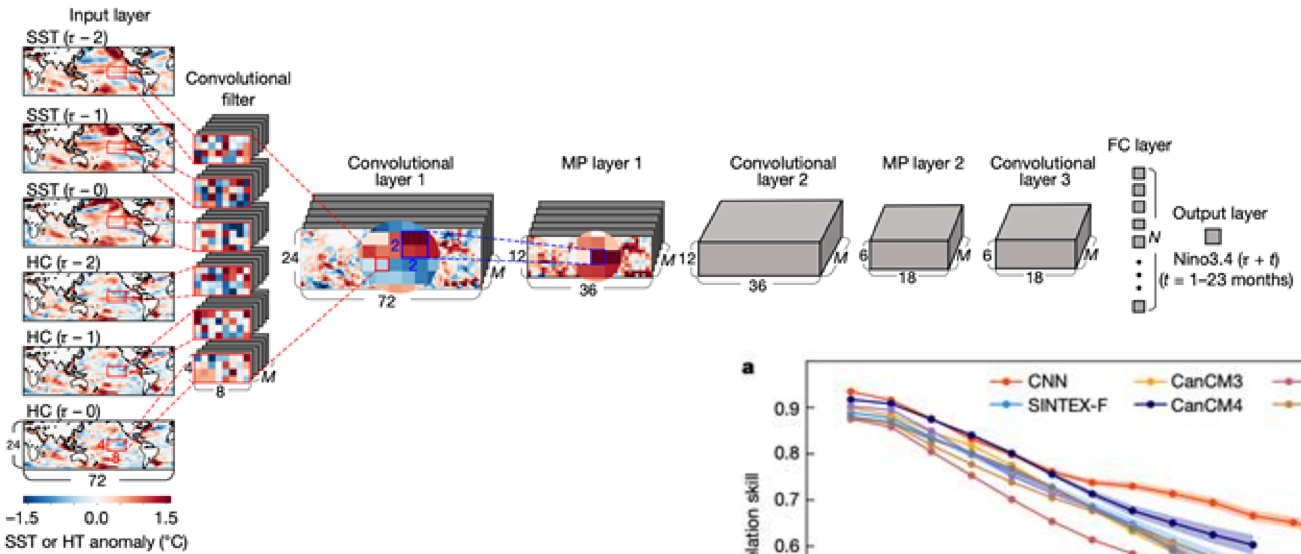


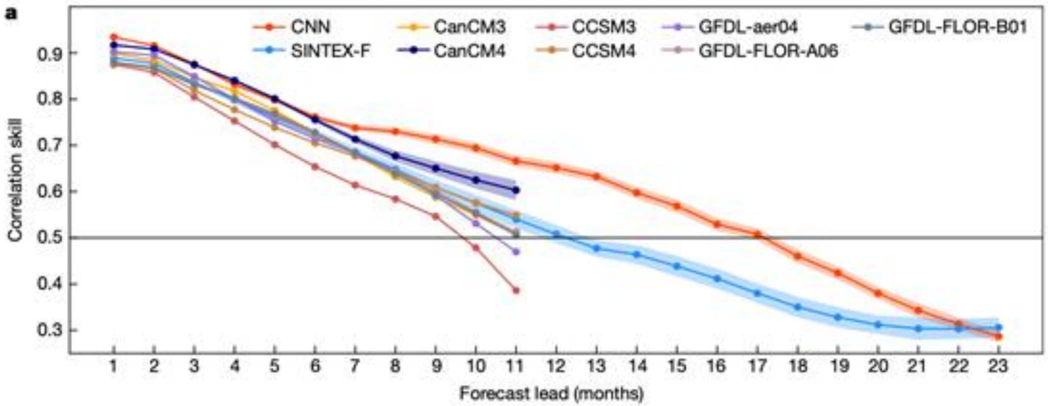
- Max pooling
- Average pooling
- …

Putting it all together



CNNs for multi-year ENSO forecasts



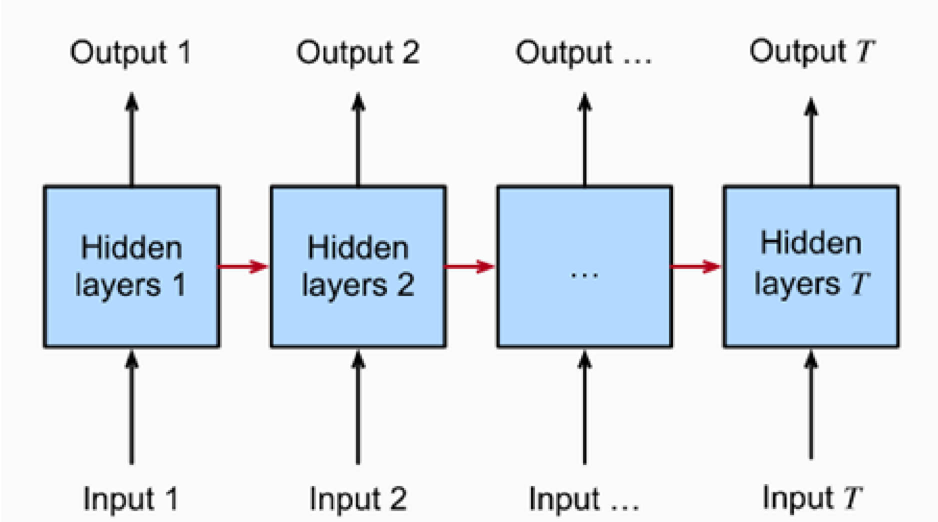


Ham, Yoo-Geun, Jeong-Hwan Kim, and Jing-Jia Luo. "Deep learning for multi-year ENSO forecasts." *Nature* 573.7775 (2019): 568-572.

Recurrent Neural Networks

Recurrent Neural Networks

- Used for sequential data: time series prediction, language processing
- memory mechanism
- problem: exploding or vanishing gradients during learning

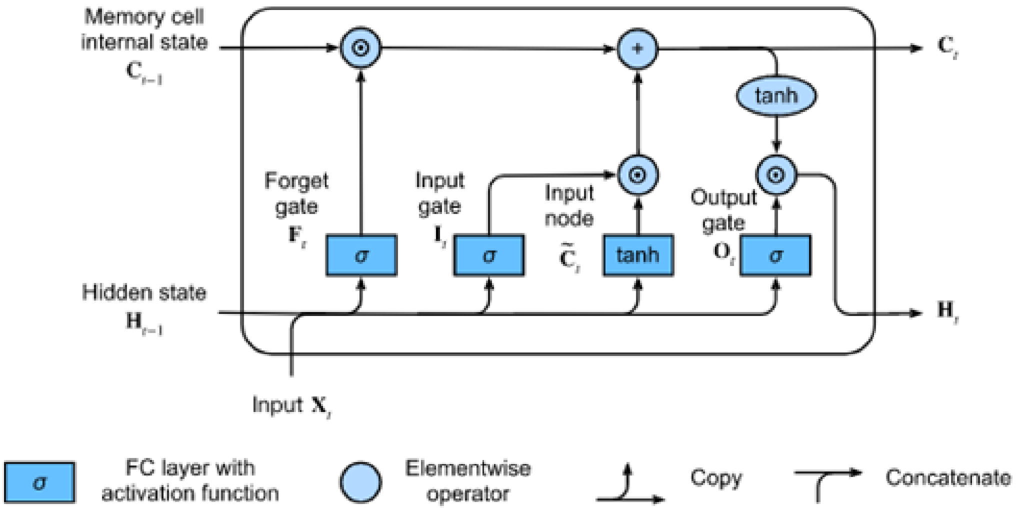


Long short-term memory (LSTM)

Can learn what important data is, and remembers this data for a long time

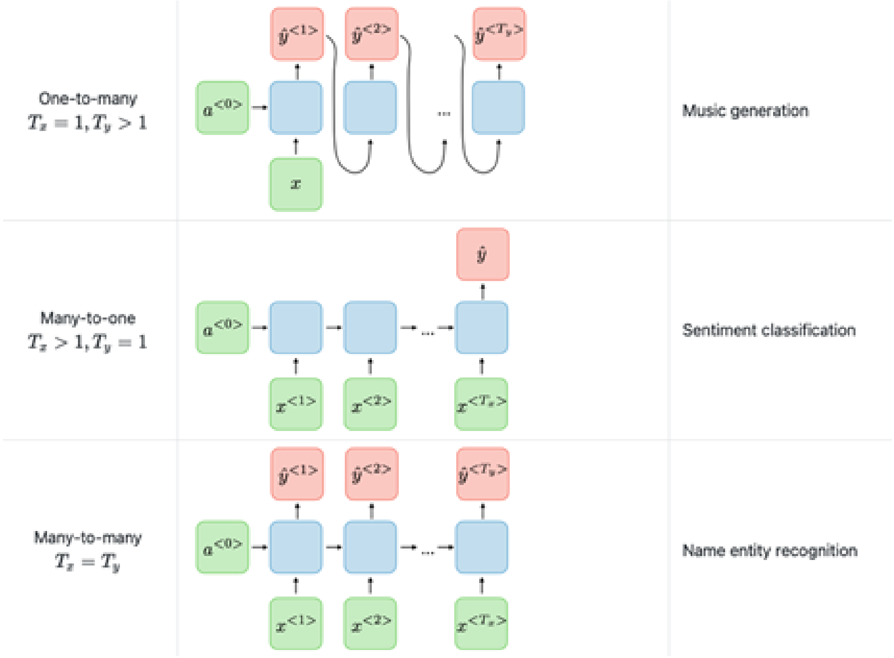
Hidden State: Short term memory
Memory Cell: Long term memory

Alternative: Gated Recurrent Unit Networks



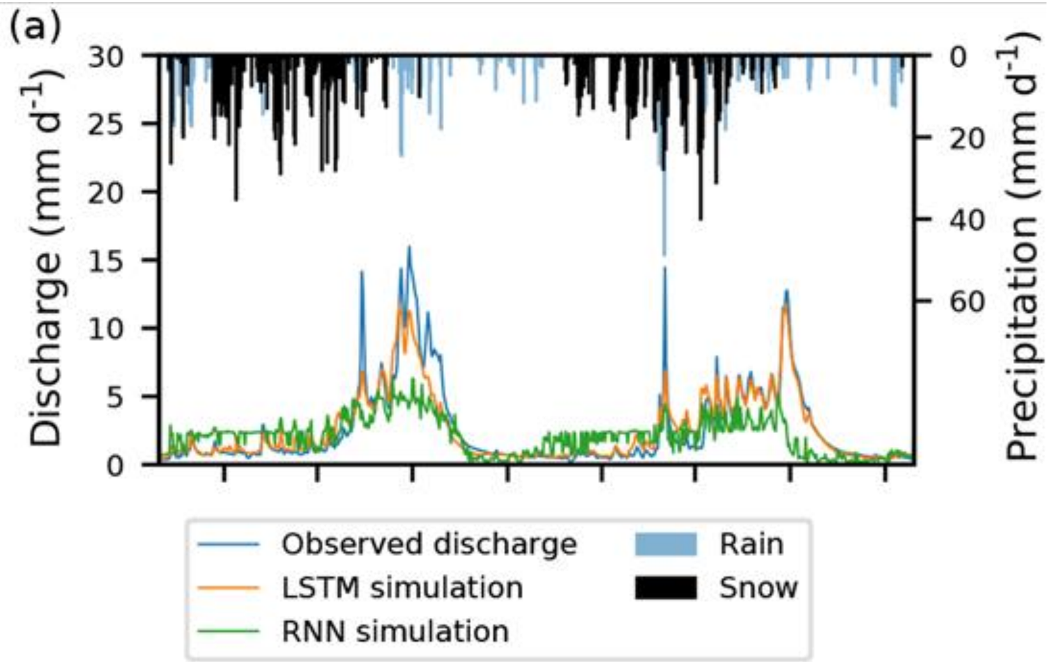
$$
\begin{aligned}
\mathbf{I}_t &= \sigma(\mathbf{X}_t\mathbf{W}_{\text{xi}} + \mathbf{H}_{t-1}\mathbf{W}_{\text{hi}} + \mathbf{b}_\text{i}),\\
\mathbf{F}_t &= \sigma(\mathbf{X}_t\mathbf{W}_{\text{xf}} + \mathbf{H}_{t-1}\mathbf{W}_{\text{hf}} + \mathbf{b}_\text{f}),\\
\mathbf{O}_t &= \sigma(\mathbf{X}_t\mathbf{W}_{\text{xo}}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

Different ways to employ them



https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

LSTM for rainfall-runoff modelling



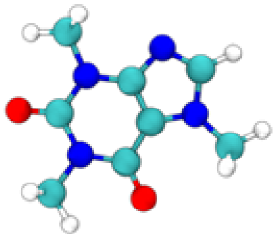
Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M.: Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks, Hydrol. Earth Syst. Sci., 22, 6005–6022, https://doi.org/10.5194/hess-22-6005-2018, 201

Graph Neural Networks

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

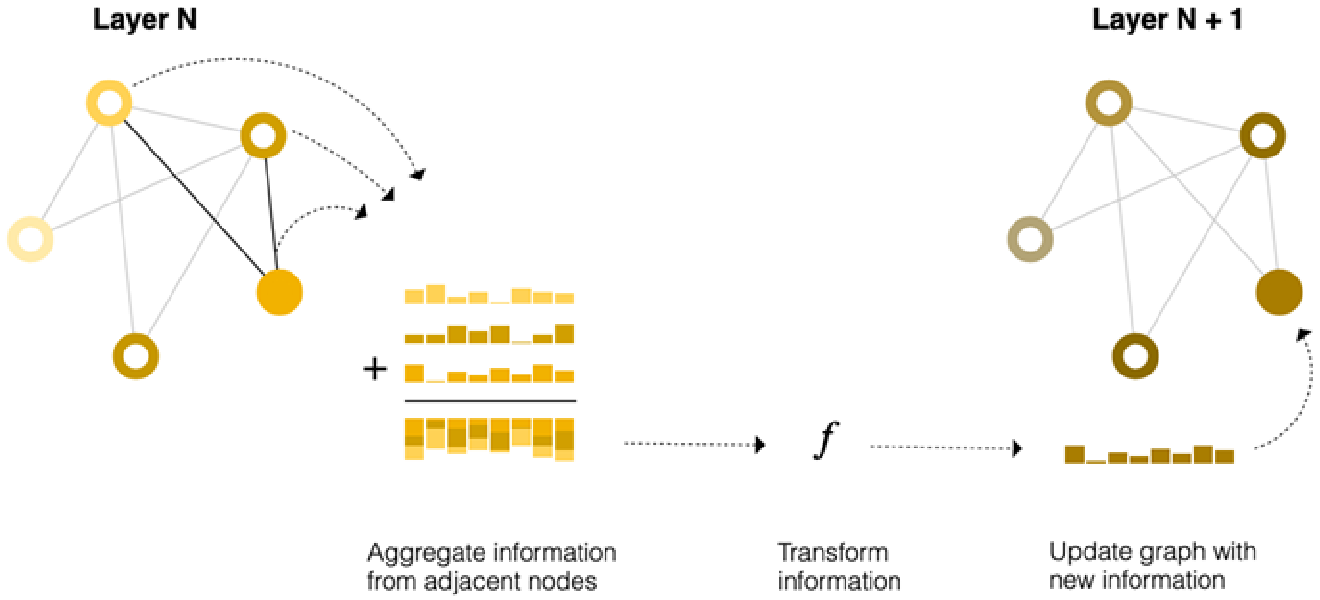
Graph Neural Networks

- 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, …



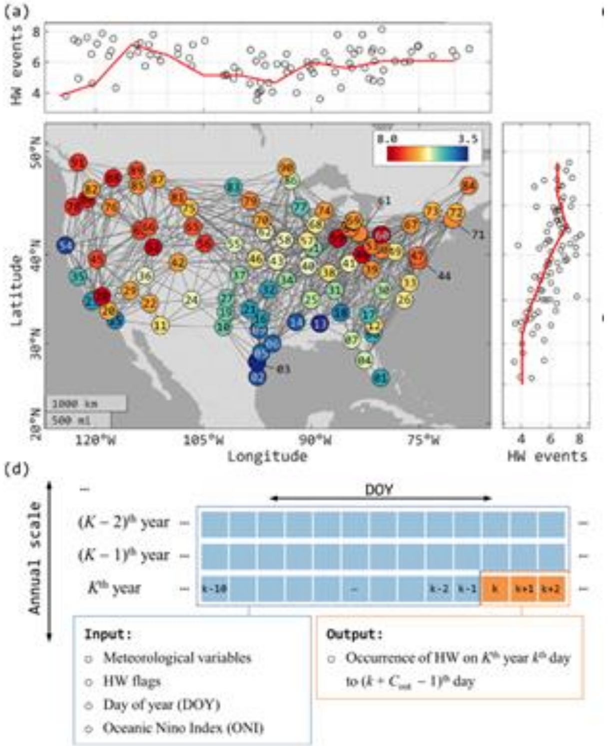
https://mpimet.mpg.de/forschung/modellierung

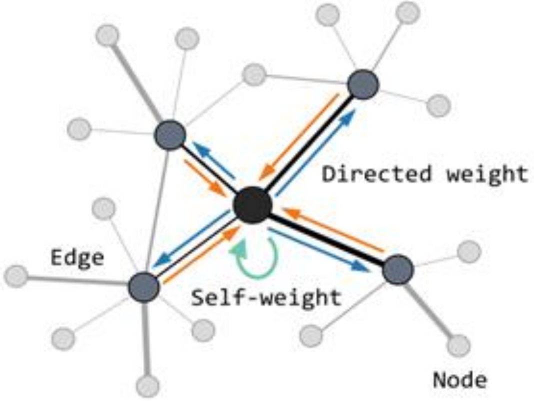
Graph Neural Networks: Message Passing



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.

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

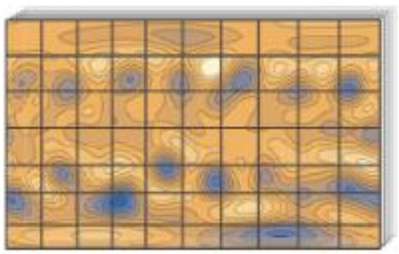
Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†
*equal technical contribution, †equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@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.[1]

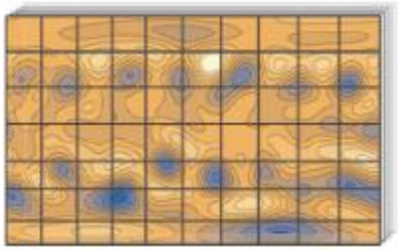
Transformers

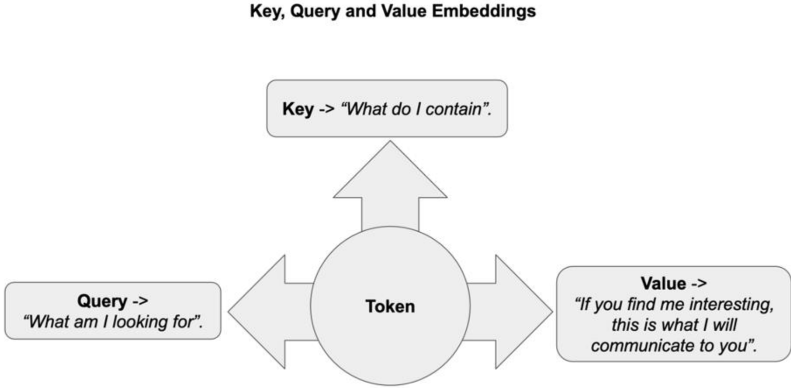
- Divide data into "tokens"

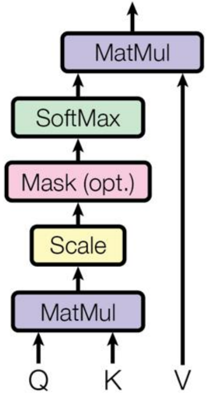


Transformers

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



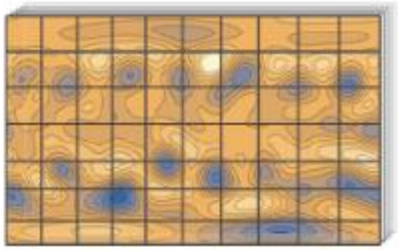


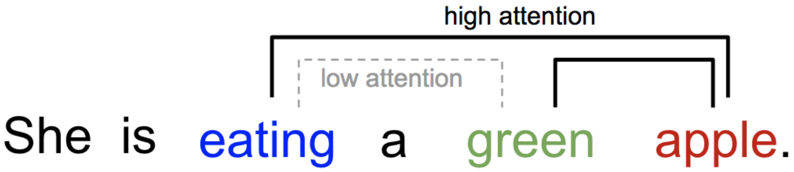


Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Transformers

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

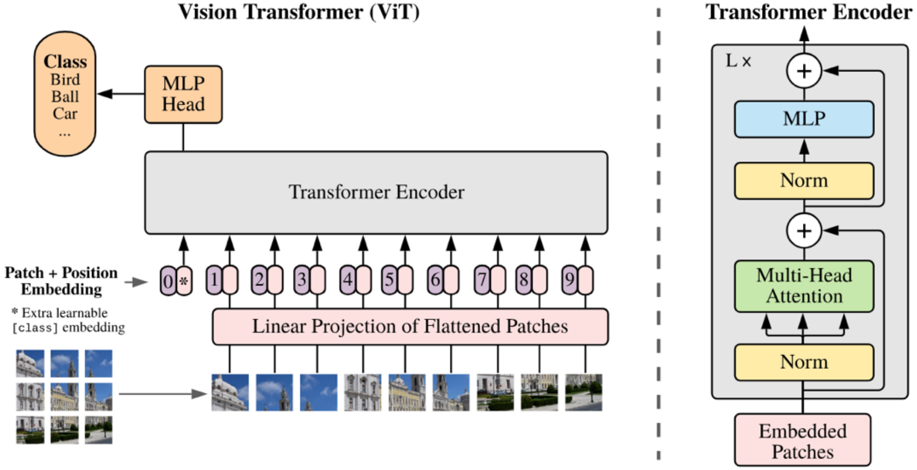




https://lilianweng.github.io/posts/2018-06-24-attention/

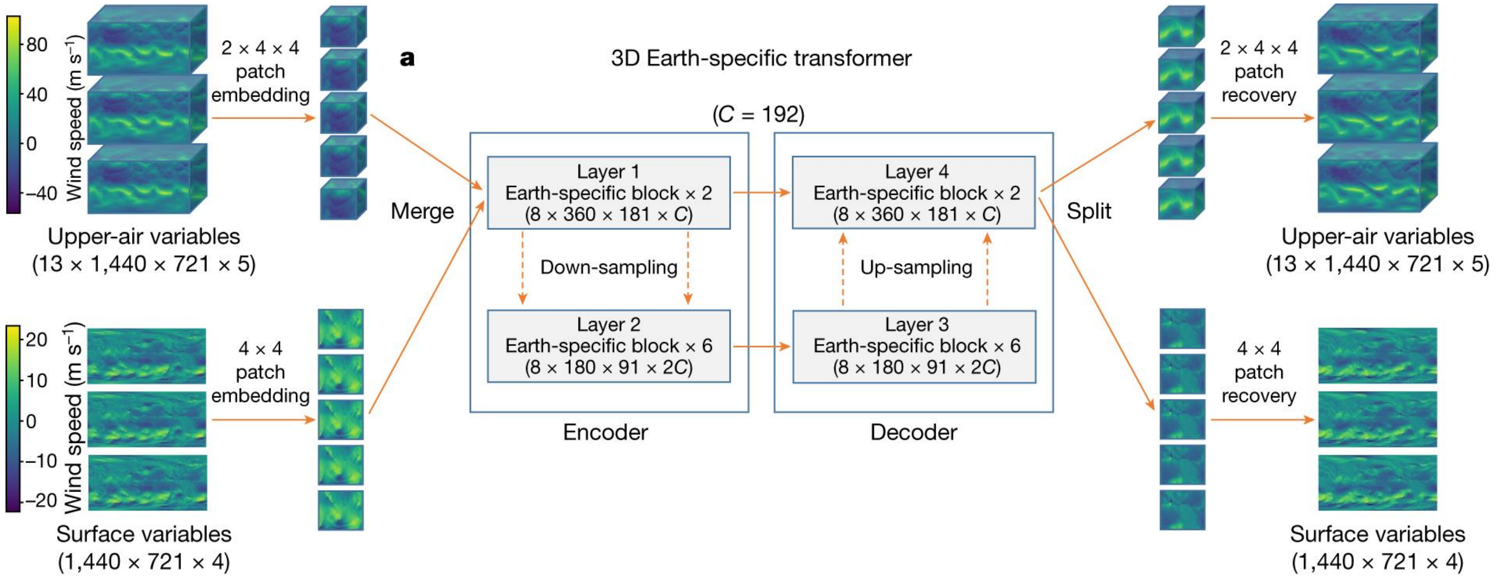
Transformers

- Divide data into “tokens”
- Embed the tokens into:
  - Key
  - Query
  - Value
- Apply Attention Mechanism
- A vision transformer combines (Multi-Head) Attention with MLPs and some other clever tricks



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

Transformers: PanguWeather



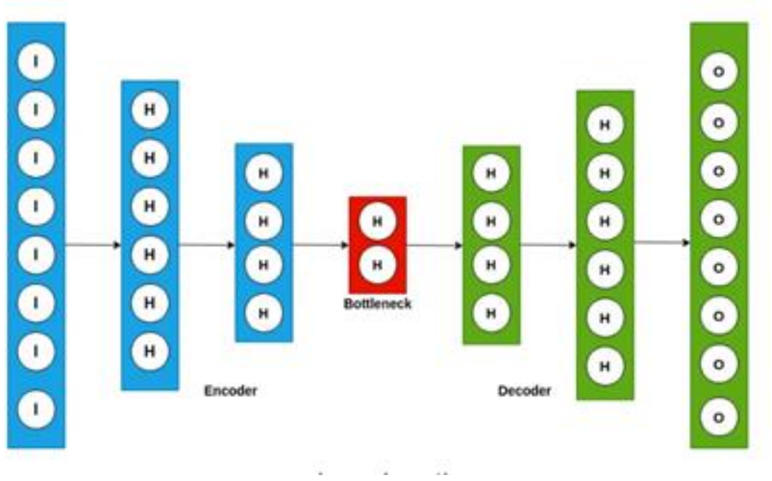
Bi, K., Xie, L., Zhang, H. *et al.* Accurate medium-range global weather forecasting with 3D neural networks. *Nature* **619**, 533–538 (2023). https://doi.org/10.1038/s41586-023-06185-3

Break

Autoencoders

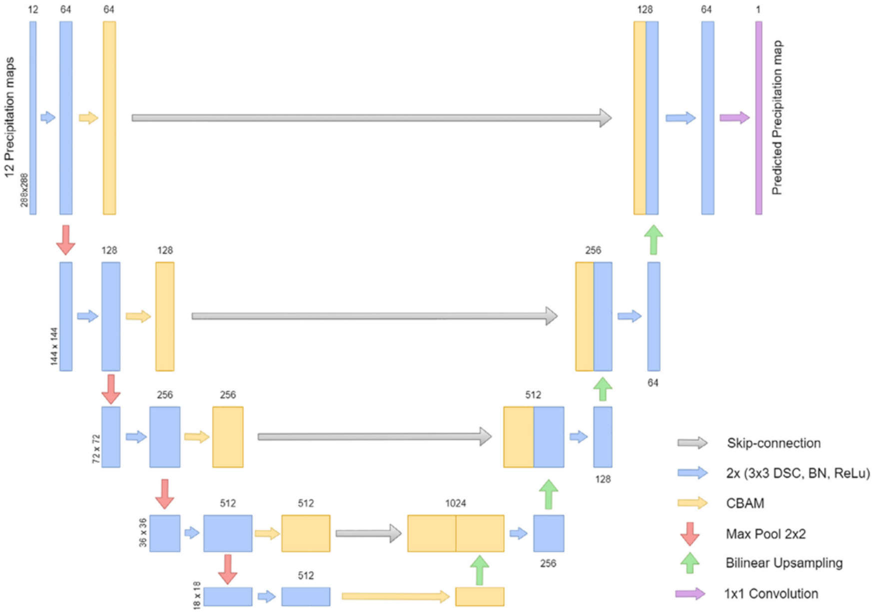
Autoencoders

- 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
- Applications include compression, denoising, infilling, prediction,…
- Can consist of elements from different architectures (MLPs, CNNs, GNNs,...)



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

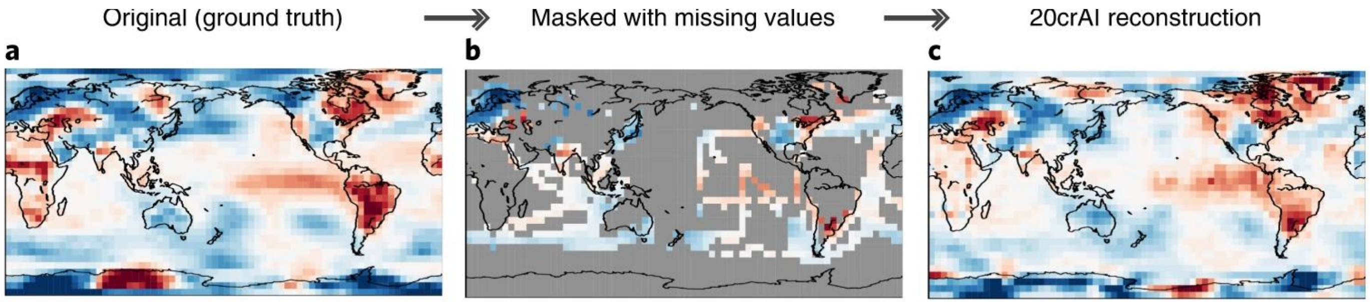
Autoencoders: UNET



Trebing, Kevin, Tomasz Staǹczyk, 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

Common problems in deep learning:

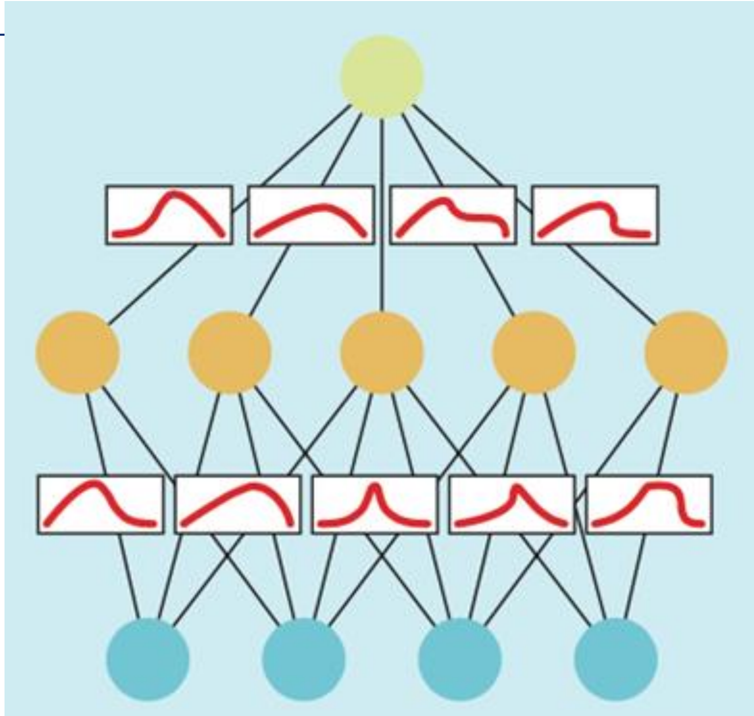
- overfitting
- overconfidence

Probabilistic Approaches to deep learning can help.

The main goal is not to be better than point-estimate methods, although this might be the case, but to provide an uncertainty estimate.

Bayesian Neural Networks

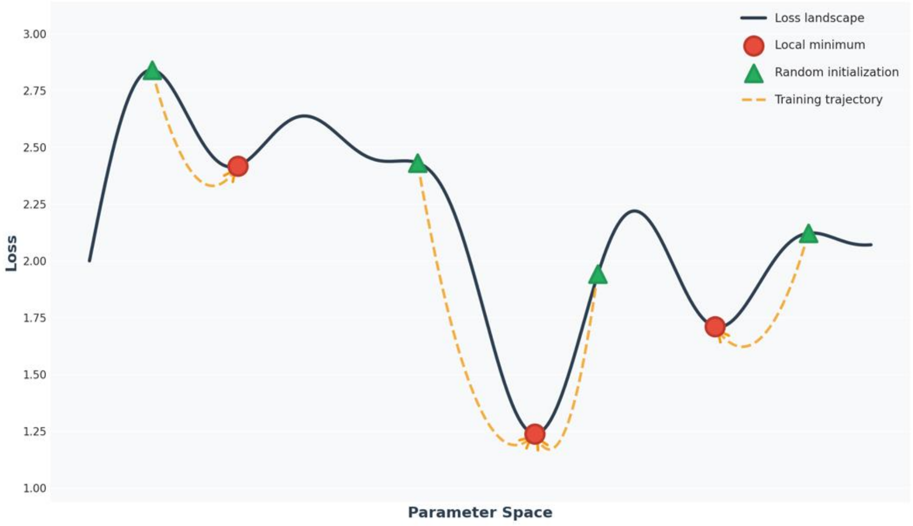
- weights are stochastic
- the output is also stochastic and therefore allows an uncertainty estimate for the prediction
- weights and biases are sampled based on Bayes theorem using e.g. Markov-Chain Monte Carlo methods or Variational Inference
- Training Routine is fundamentally different than standard deep learning architectures
- Python Packages: `Pyro`, `Bayesian-Torch`, `TorchUncertainty`, `TensorFlow Probability`



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 but intitialised 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/deepensembles/

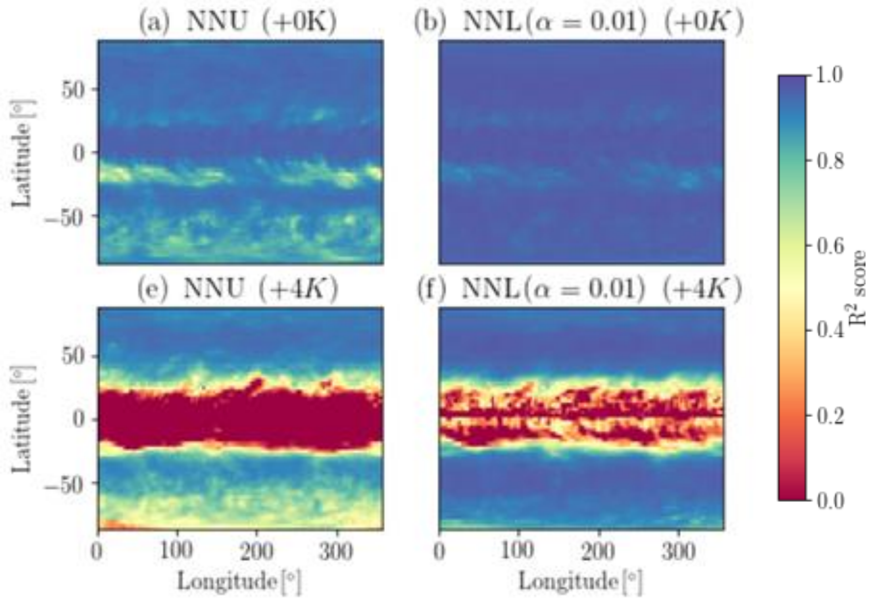


Physics-Informed Deep Learning

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

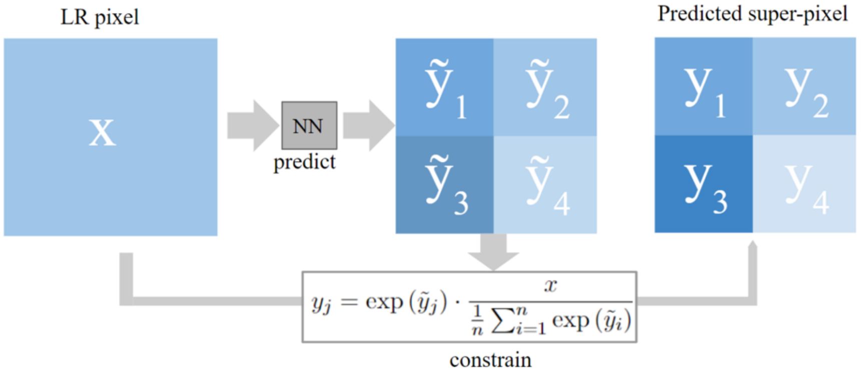
$$\mathcal{L}(\alpha) = \alpha \mathcal{P}(x, y_{\mathrm{NN}}) + (1-\alpha)\,\mathrm{MSE}(y, y_{\mathrm{NN}})$$

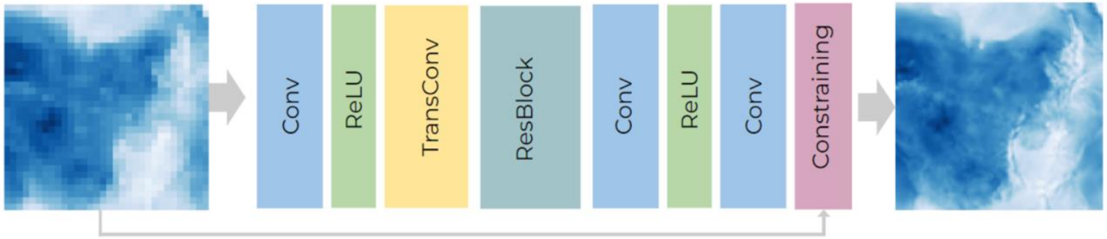


“Achieving Conservation of Energy in Neural Network Emulators for Climate Modeling”,Beucler et al 2019, https://arxiv.org/abs/1906.06622

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“)

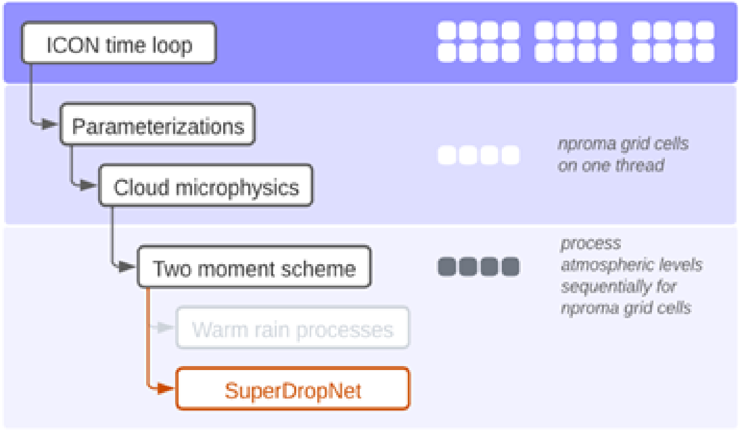
Potential:

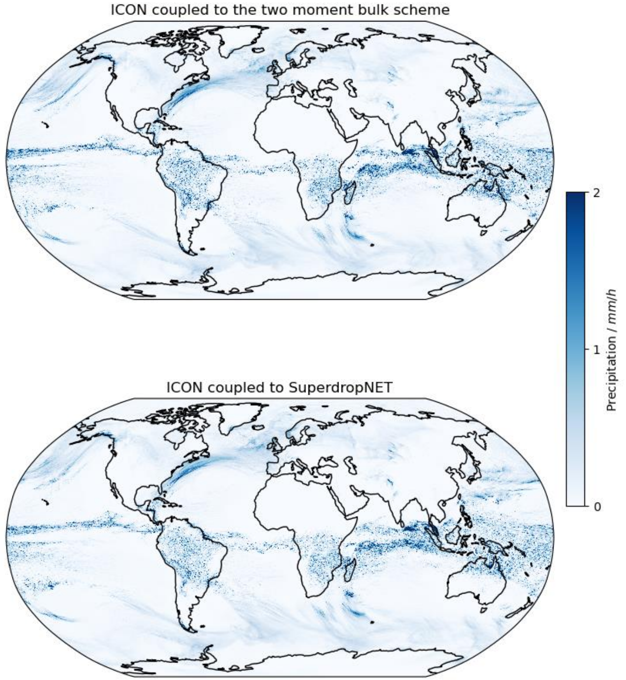
- Replace parameterisations with more accurate ML-based relationships that are learned from observations.
- Replace computationally expensive components with ML-based model

Challanges:

- Efficient Technical implementation of python code into FORTRAN or C++
- Offline vs online behaviour of ML model
- Application to problems outside of training data

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.

FTorch

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

Fully Differentiable General Circulation Models

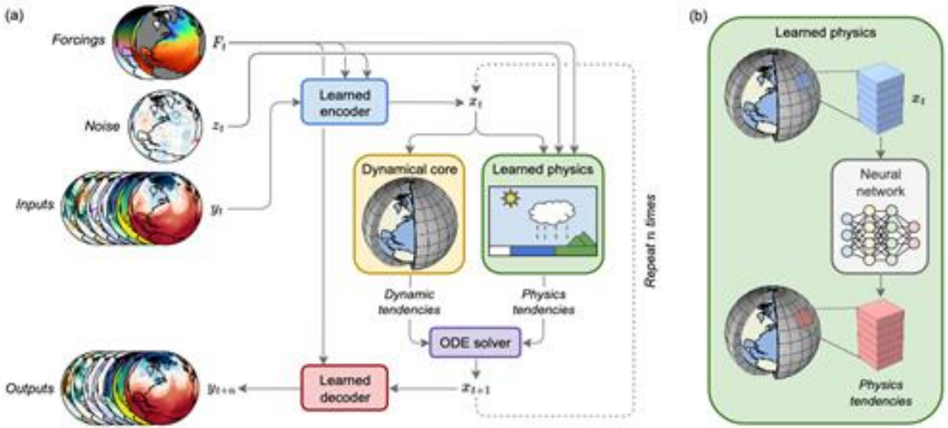


Fig. 1 Structure of the NeuralGCM model. (a) Overall model structure, showing how forcings F_t, noise z_t (for stochastic models), and inputs y_t are encoded into the model state x_t. Model state is fed into the dynamical core, and alongside forcings and noise into the learned physics module. This produces 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 the context of 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!

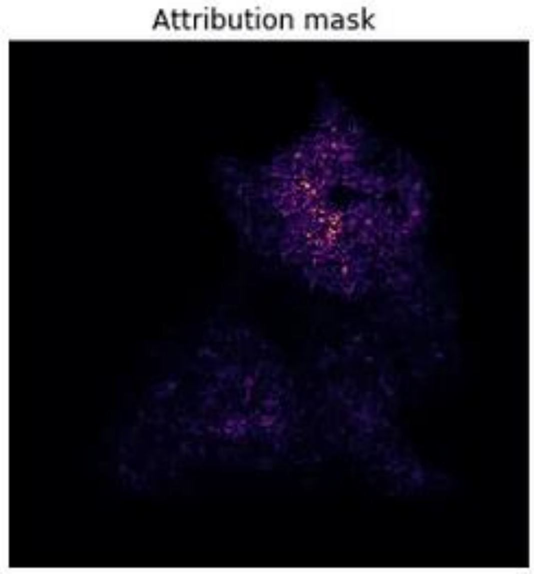
Explainable AI: Integrated Gradients

- 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

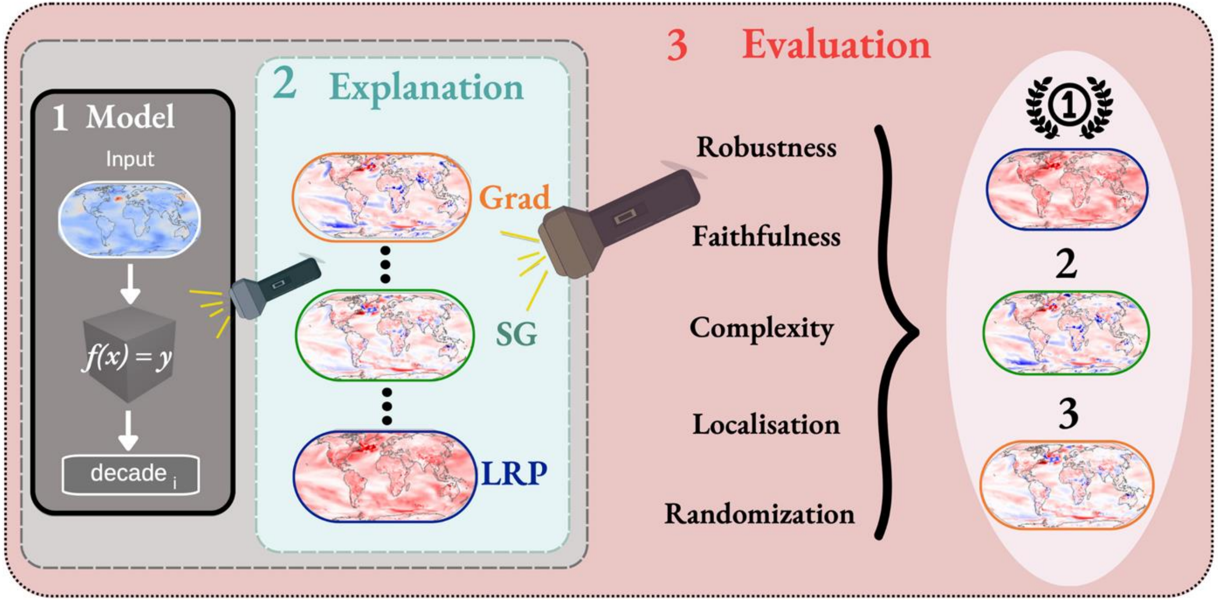
https://www.xaifoundation.org/copy-of-lrp

Explainable AI: Other methods

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

https://www.xaifoundation.org/xai-for-neural-networks

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
- 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
- High quality, meaningful datasets and a use-case that makes sense are the most important ingredients

Thank you!