Intro to Machine Learning for Earth Scientists

Lauren Hoffman
SIOC 221B, Winter 2023

Overview

- My research: An application of ML in the Geosciences

- Introduction to ML
  - ML model types & architectures
  - Data (satellite, maps, time series, etc.) (train, validate, test)
  - Evaluating ML (skill, correlation)
  - Coding environments (Python, Tensorflow Keras)
  - ML resources (UCSD, MOOC w/ ECMWF, Deep Learning AI Coursera)

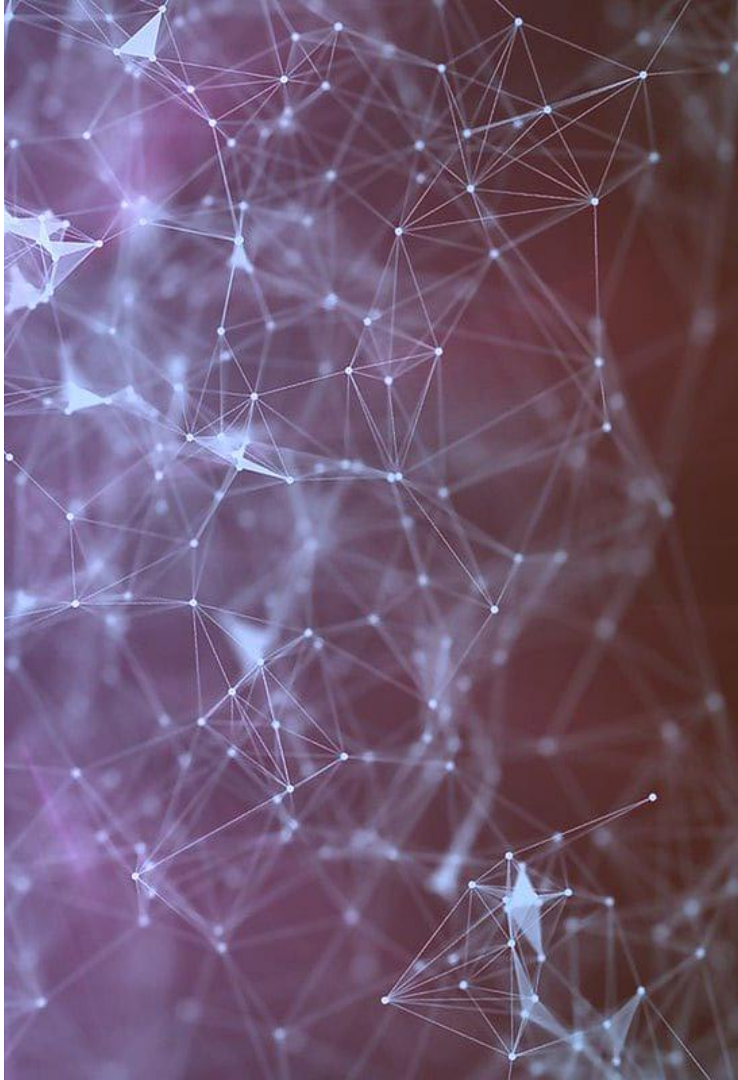
- Exercise in ML
  - https://github.com/lahoffman/ml_tutorial

A Brief Overview

What is Machine Learning?

"Field of study that gives computers the ability to learn without being explicitly programmed."
-coined by Arthur Samuel in 1959

- Fitting a model to data
- An application of AI



Where have we encountered machine learning?

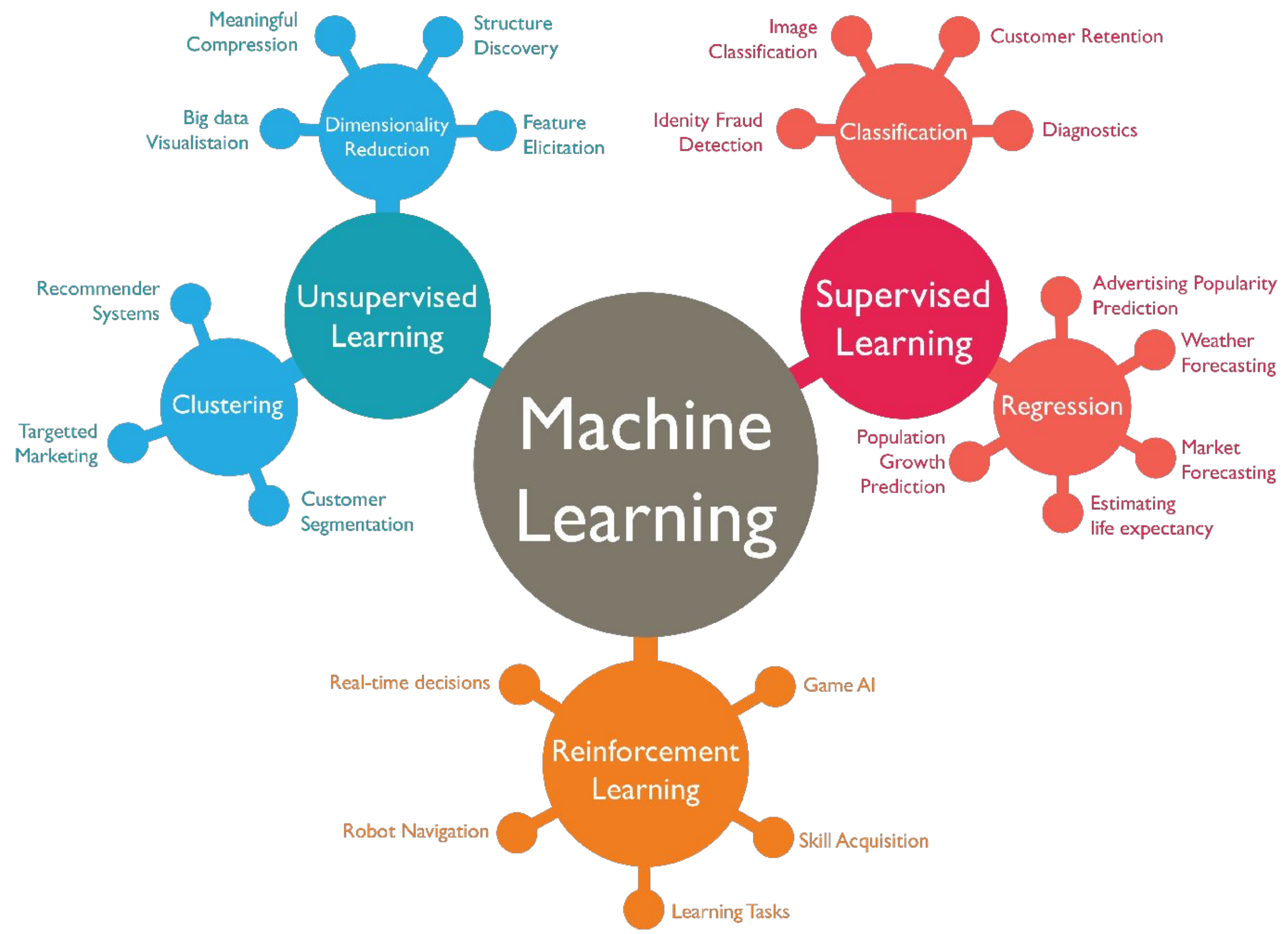


Image Classification: Chihuahua or Blueberry Muffin?

A Case Study:

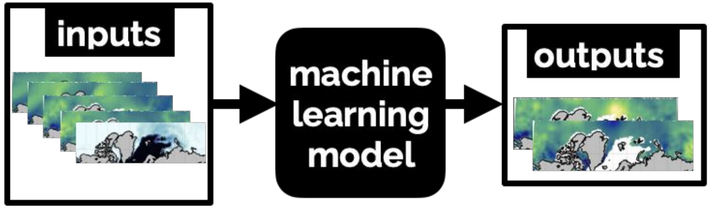
Machine learning is a useful tool to predict and understand sea-ice motion in the Arctic.

Machine learning is a useful tool to predict and understand sea-ice motion in the Arctic.

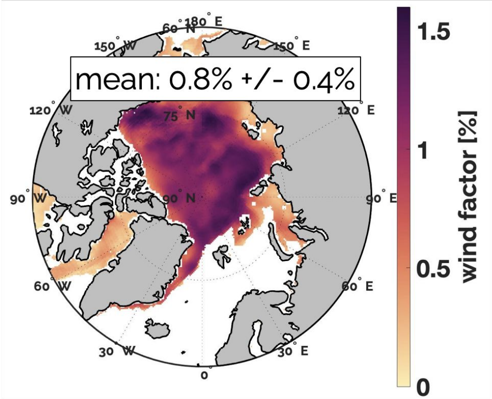
Lauren Hoffman[1], Matthew Mazloff[1], Sarah Gille[1], Donata Giglio[2], Cecilia Bitz[3], Patrick Heimbach[4]
[1] Scripps Institution of Oceanography, [2] University of Colorado Boulder, [3] University of Washington, [4] University of Texas at Austin

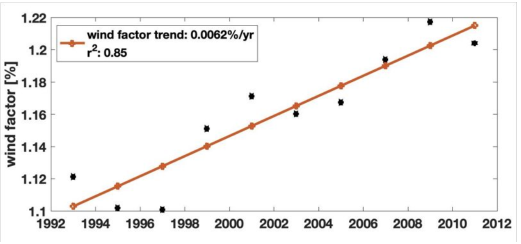
Predictability

Machine learning models are used to make one-day predictions of sea-ice velocity in the Arctic.



Understanding sea-ice motion





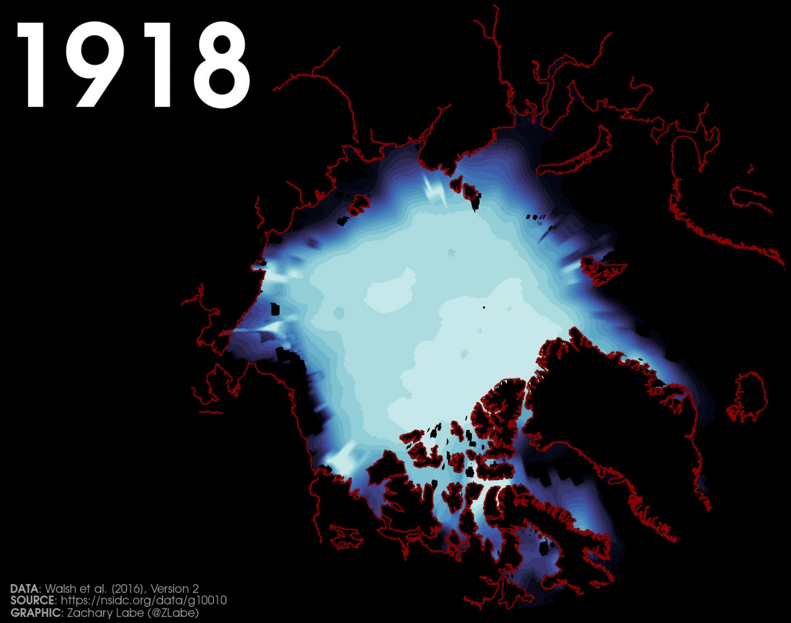
wind factor: $\frac{\text{sea-ice speed}}{\text{wind speed}}$

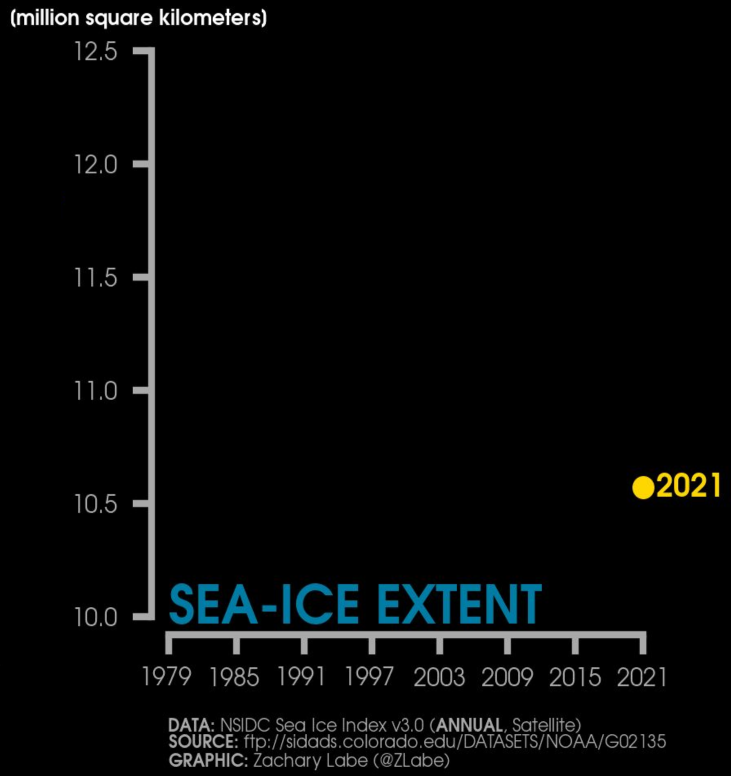
The **wind factor is increasing!**

As the ice melts it is becoming more responsive to wind forcing.

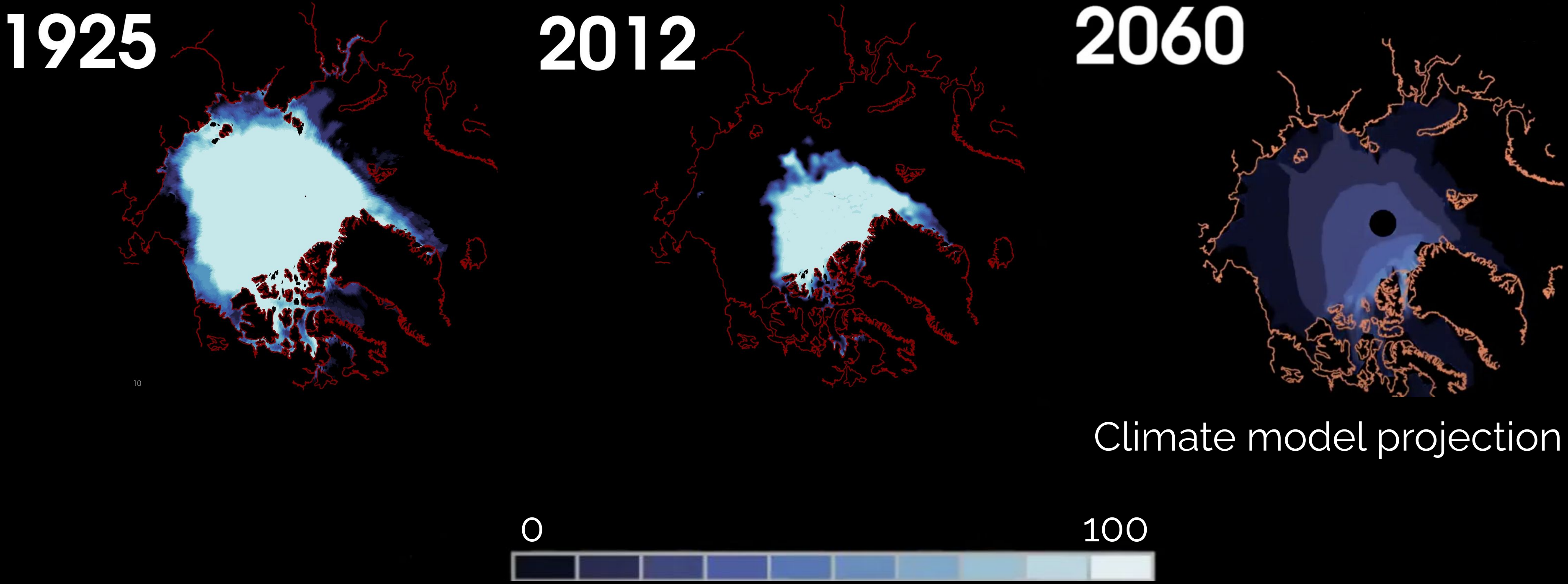
lahoffma@eng.ucsd.edu

Sea Ice in the Arctic is melting



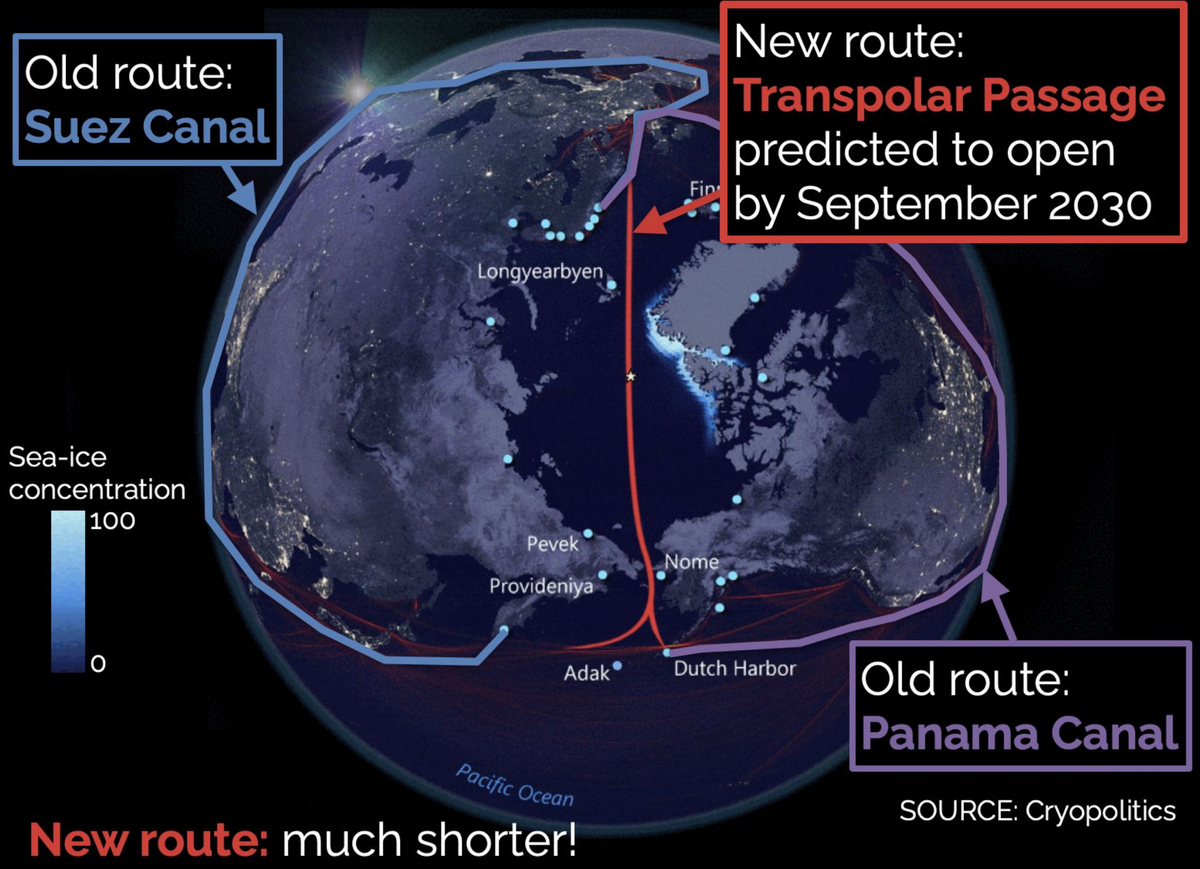


Sea Ice in the Arctic is melting



September sea-ice concentration (%)

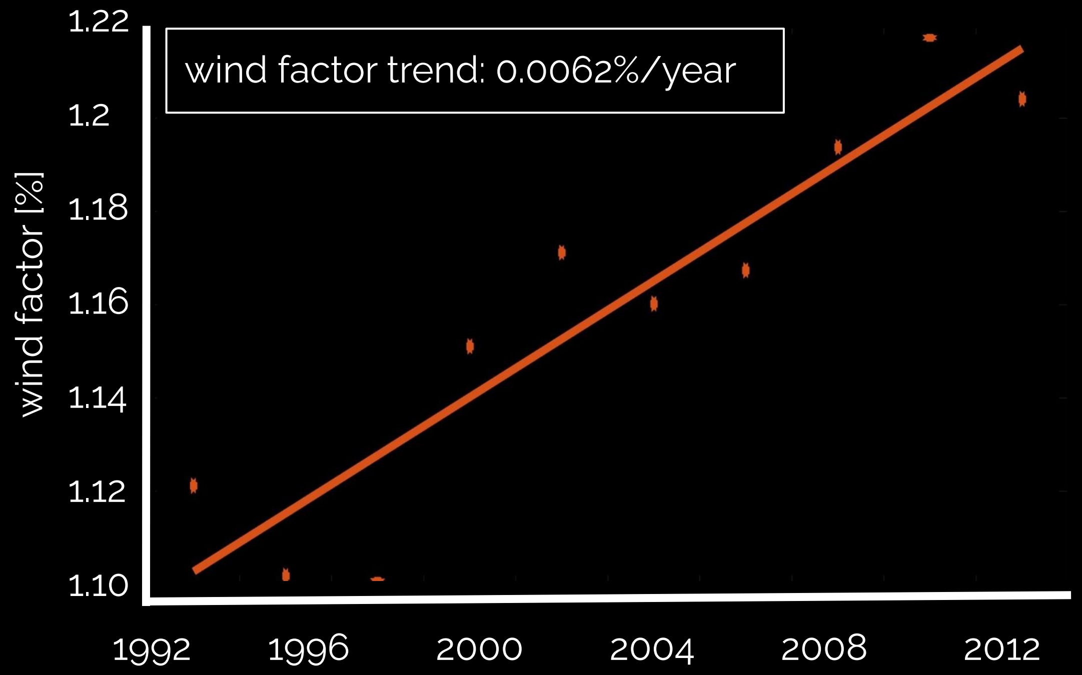
DATA: Walsh et al. (2016), Version 2
SOURCE: https://nsidc.org/data/g10010
GRAPHIC: Zachary Labe (@ZLabe)



This opens new opportunities for trade & transportation.

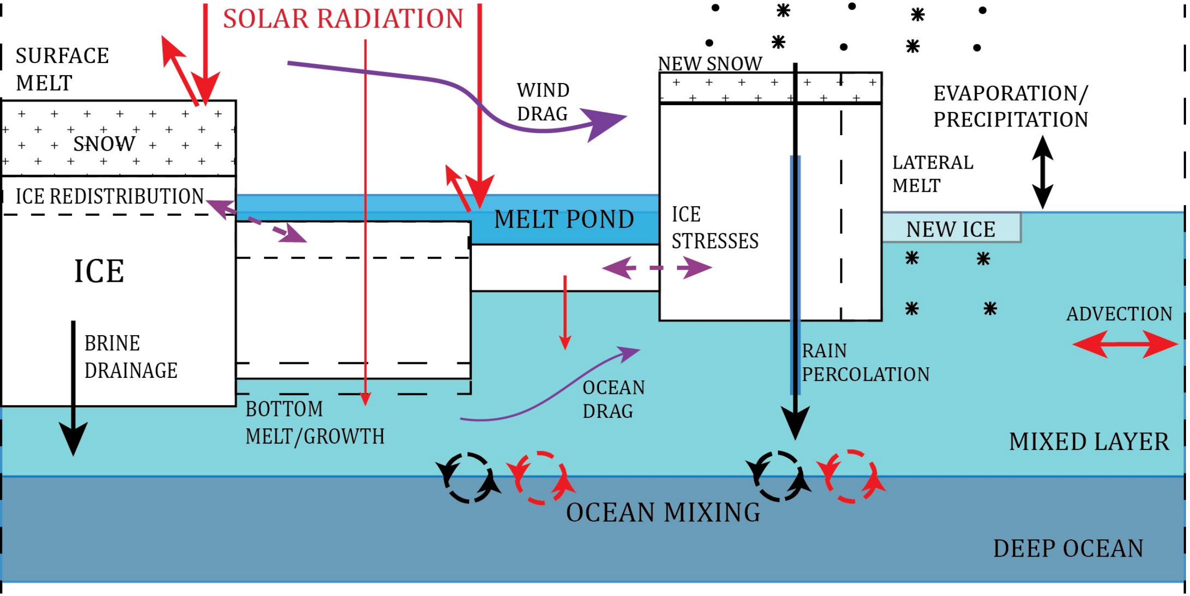
Which brings with it a need to (i) know where sea-ice is & (ii) for an ability predict where it will be.

The relationship between sea-ice motion and wind is changing in time.



wind factor (ratio of sea-ice speed to wind speed) is increasing - why?

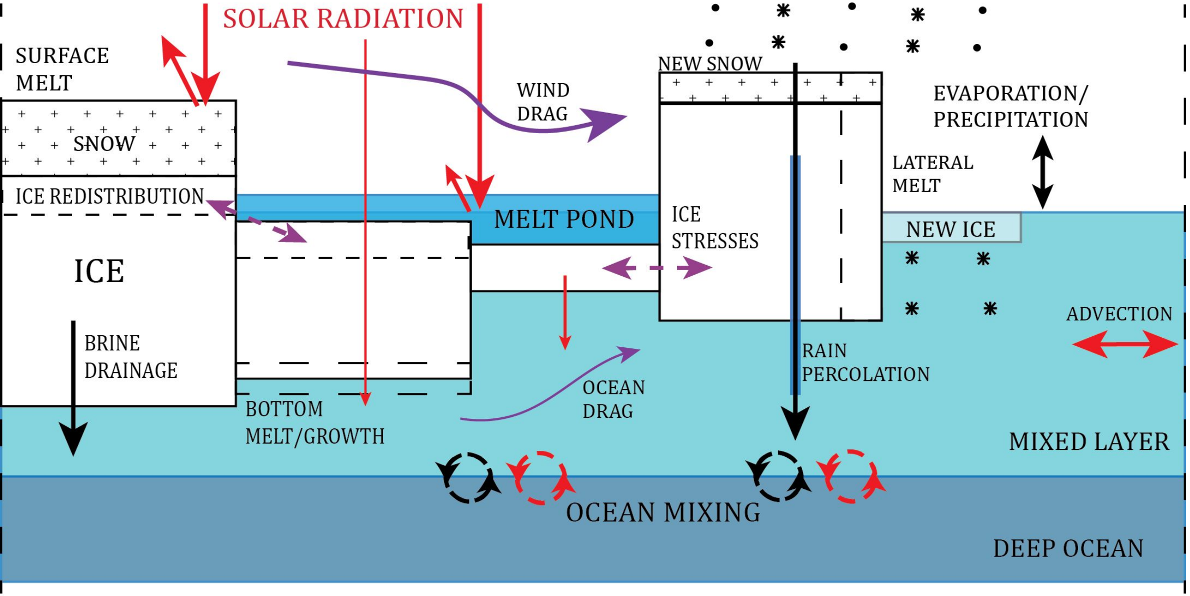
Machine learning models for sea-ice drift have fewer complexities and a lower computational cost than traditional physics-based models.



Physical processes now included in state-of-the-art sea ice models such as CICE (Ed Hawkins, 2015).

Hoffman et al. 2023, *submitted*

Machine learning models for sea-ice drift have fewer complexities and a lower computational cost than traditional physics-based models.



Physical processes now included in state-of-the-art sea ice models such as CICE (Ed Hawkins, 2015).

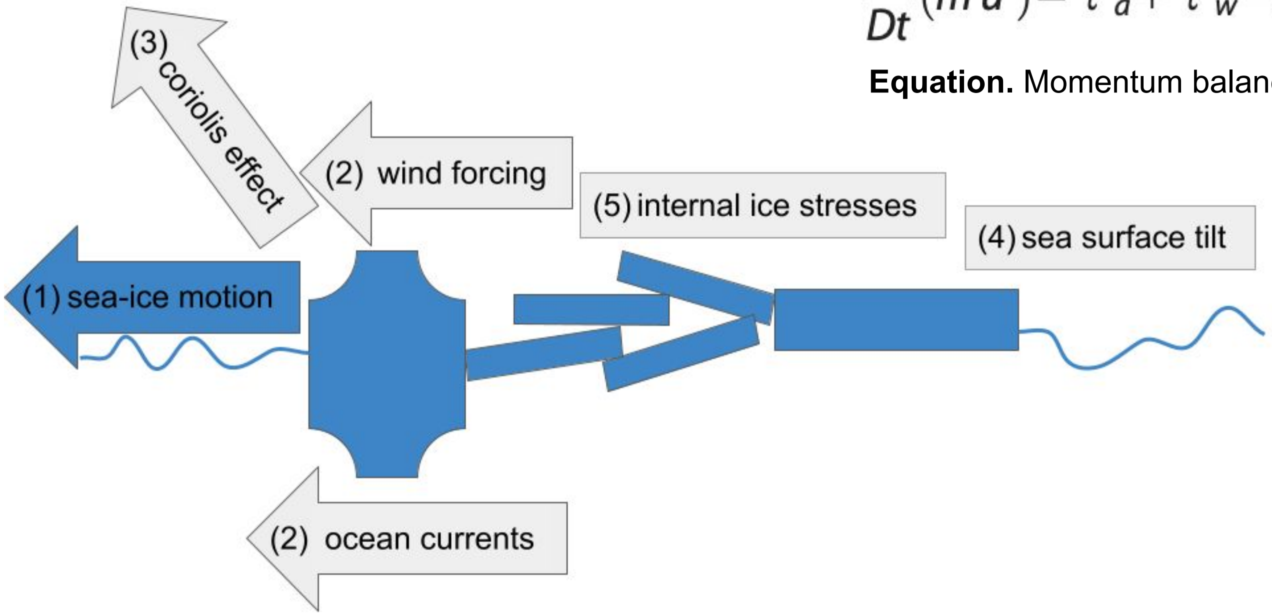
Machine Learning models can be used to understand sea-ice motion because they are drawing information from the data.

Hoffman et al. 2023, *submitted*

Sea-ice motion is largely determined by oceanic & atmospheric forces and internal ice stresses.

$$\underbrace{\frac{D}{Dt}(m\vec{u})}^{(1)}=\overbrace{\vec{\tau}_a+\vec{\tau}_w}^{(2)}\overbrace{-mf\vec{k}\times\vec{u}}^{(3)}\overbrace{-mg\vec{\nabla}H}^{(4)}\overbrace{-\nabla\cdot\sigma}^{(5)}.$$

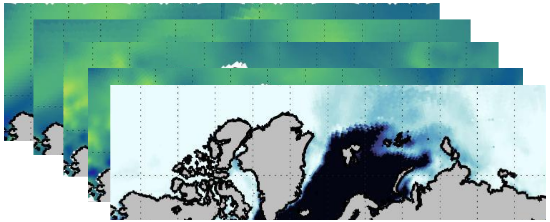
Equation. Momentum balance for sea ice.



Hibler. *JPO.* (1979).
Thorndike & Colony. *JGR.* (1982).

The machine learning models make one-day predictions of sea-ice velocity given input data from satellite & reanalysis sources (1992-2017).

inputs



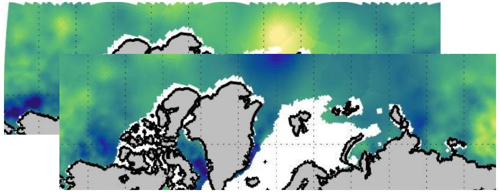
Present-day wind velocity[1]: $u_{a,\,t=1}$ & $v_{a,\,t=1}$

Previous-day sea-ice velocity[2]: $u_{i,\,t=0}$ & $v_{i,\,t=0}$

Previous-day sea-ice concentration[3]: $c_{i,\,t=0}$

machine learning model

outputs



Present-day sea-ice velocity: $u_{i,\,t=1}$ & $v_{i,\,t=1}$

Model Inputs: [1]JRA55; [2]Polar Pathfinder; [3]Nimbus-7

Hoffman et al. 2023, *submitted*

Three different models are applied to make one-day predictions of sea-ice velocity.

Persistence:

PS

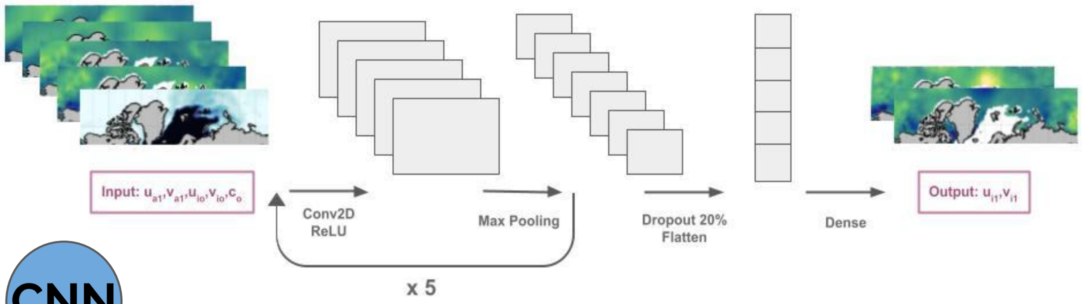
$$\bar{u}_{i,\,t=1} = \bar{u}_{i,\,t=0}$$

Linear Regression:

LR

$$\bar{u}_{i,\,t=1} = A^{*}\bar{u}_{a,\,t=1} + B^{*}\bar{u}_{i,\,t=0} + C^{*}c_{t=0}$$

Convolutional Neural Network [CNN]:



CNN

A convolutional neural network (CNN) outperforms linear regression (LR) and persistence (PS) models.

Model Evaluation Metric

$$Corr_{x,y} = \frac{\sum_i^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i^n (x_i - \bar{x})^2}\sqrt{\sum_i^n (y_i - \bar{y})^2}},$$

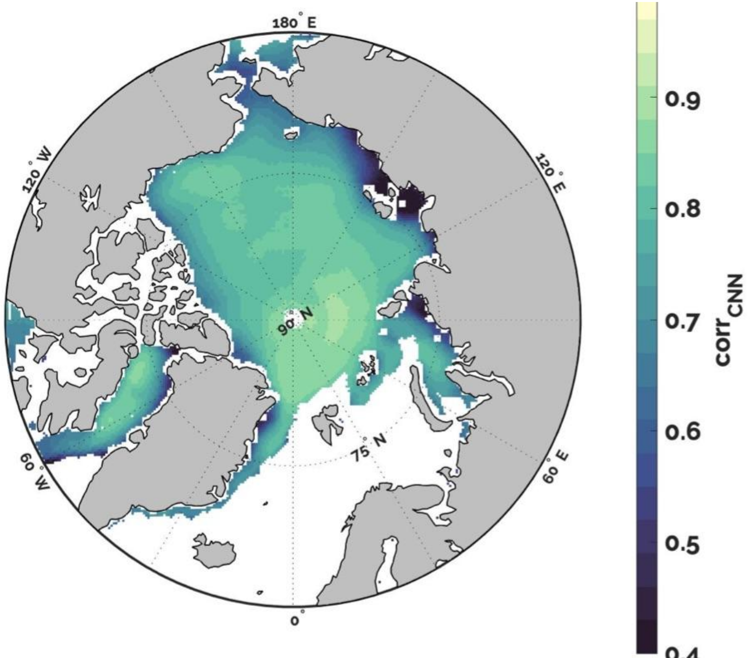
x: data; y: predicted

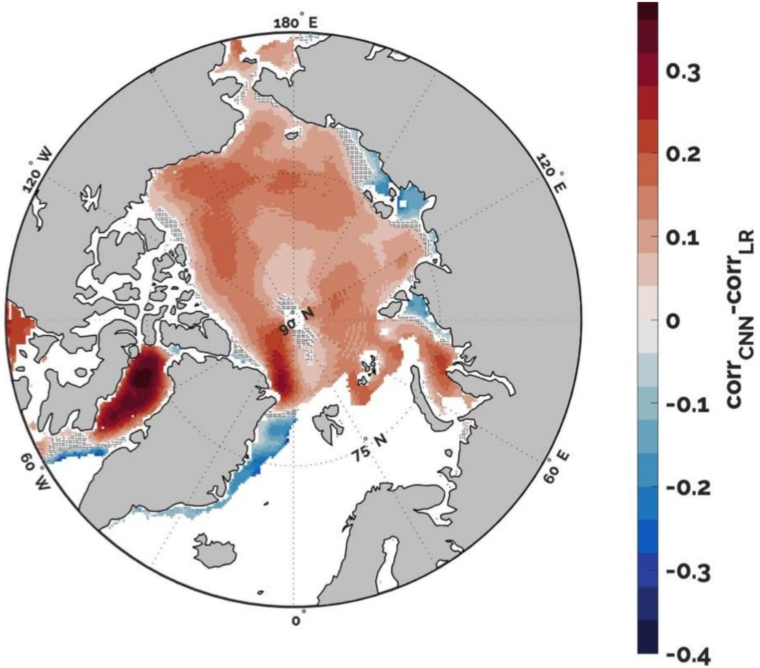
increasing skill

model	correlation
Persistence	0.69 +/- 0.02
Linear Regression	0.77 +/- 0.02
CNN	0.80 +/- 0.01

Hoffman et al. 2023, *submitted*

Models that incorporate non-linear relationships between inputs capture important information (i.e. $corr_{CNN} > corr_{LR}$).

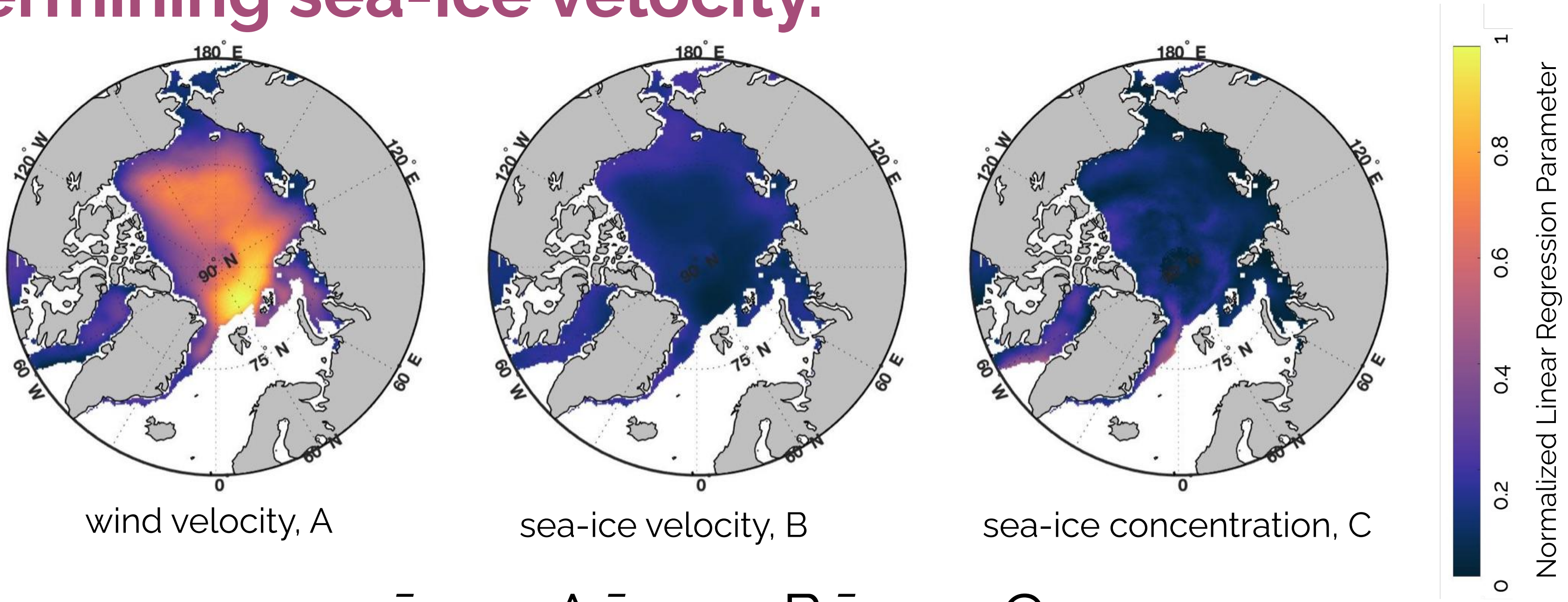




Predictability

Hoffman et al. 2023, *submitted*

Machine learning methods confirm historical results that wind velocity has the largest relevance in determining sea-ice velocity.

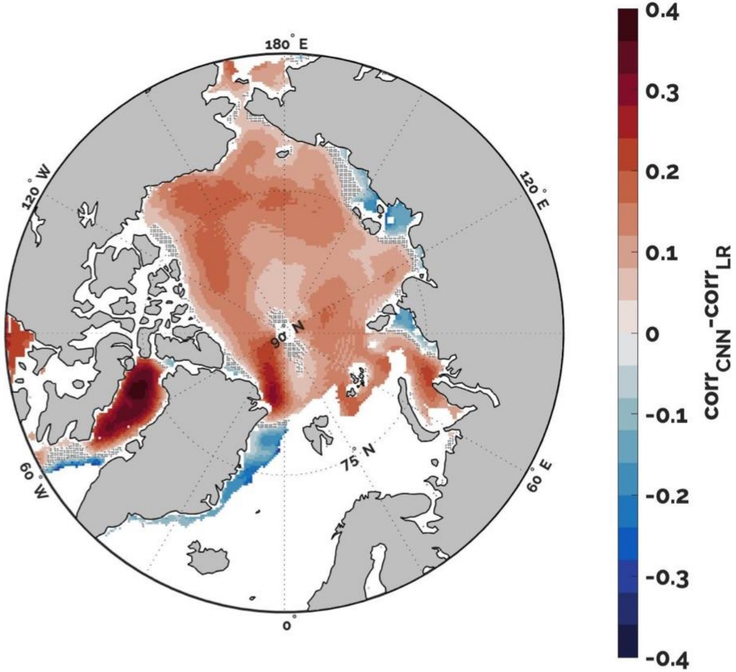


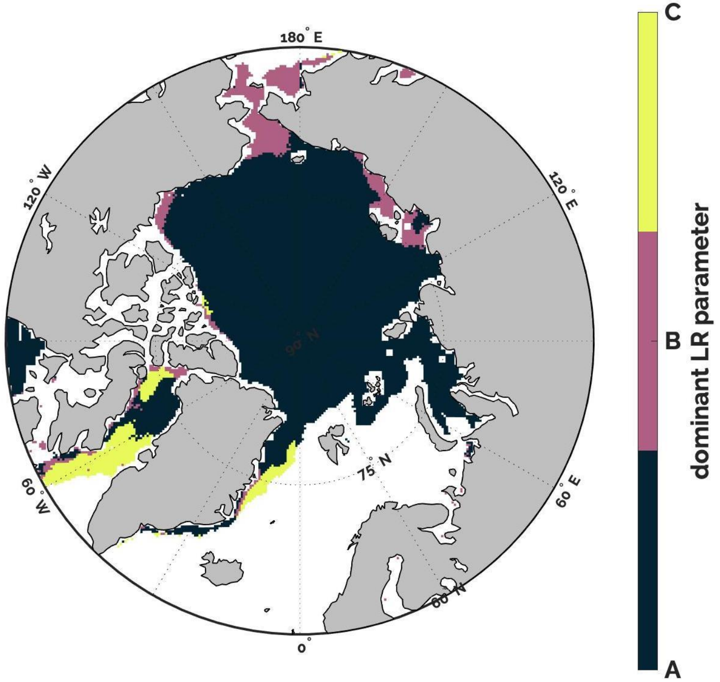
$$\bar{u}_{i,\ t=1} = A\bar{u}_{w,\ t=1} + B\bar{u}_{i,\ t=0} + Cc_{t=0}$$

Understanding sea-ice motion: LR

Hoffman et al. 2023, *submitted*

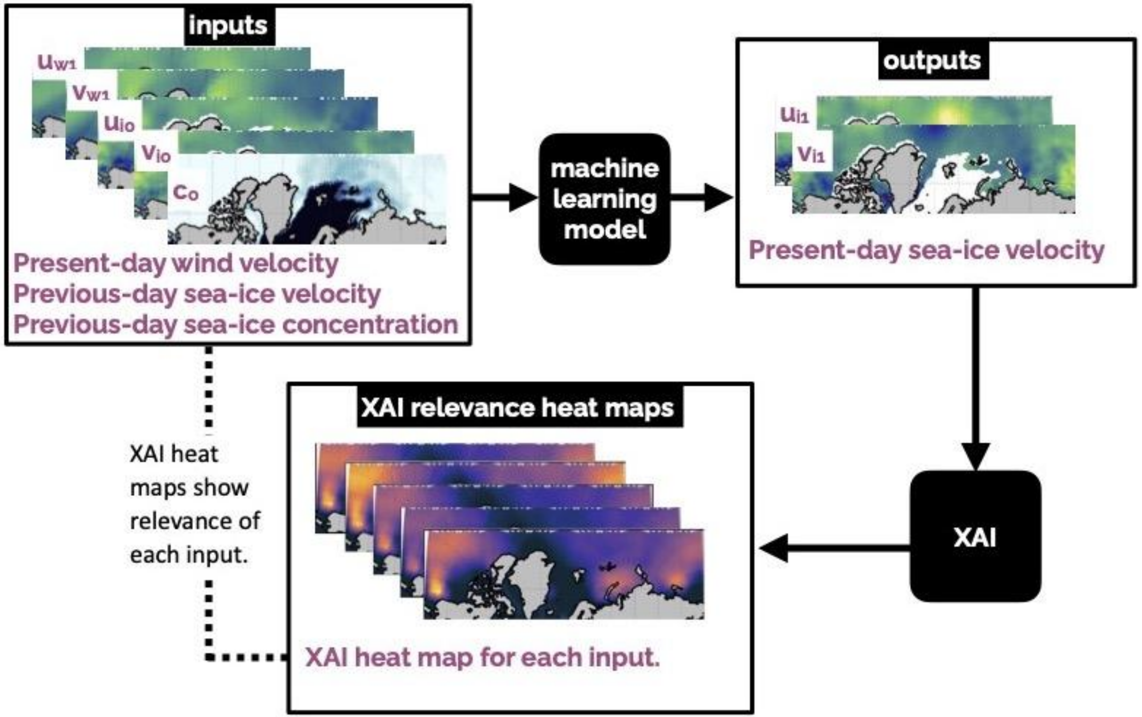
The CNN outperforms the LR primarily in the central Arctic where wind speed (A) is the dominant predictor of ice motion.





Hoffman et al. 2023, *submitted*

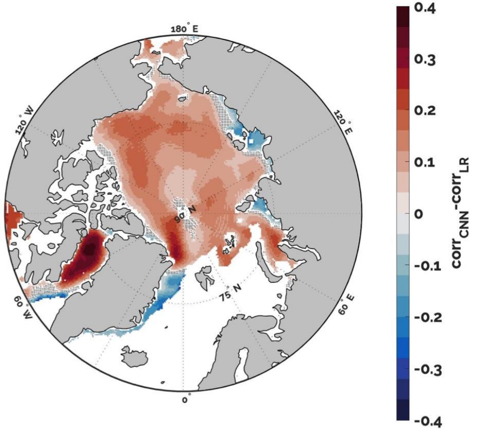
Explainable AI methods interpret when and where each input is relevant for the machine learning model in predicting the output.

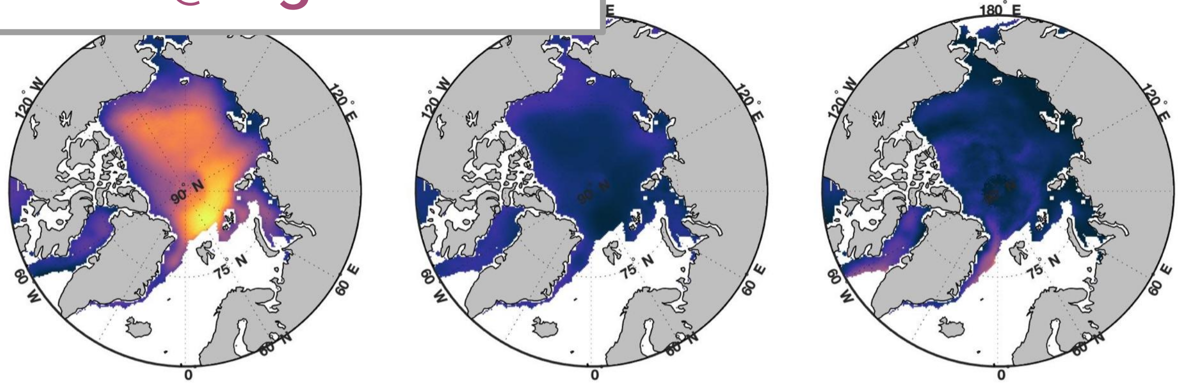


CNN

Machine learning is a useful tool to predict and understand sea-ice motion in the Arctic.

lahoffma@eng.ucsd.edu





wind velocity, A

sea-ice velocity, B

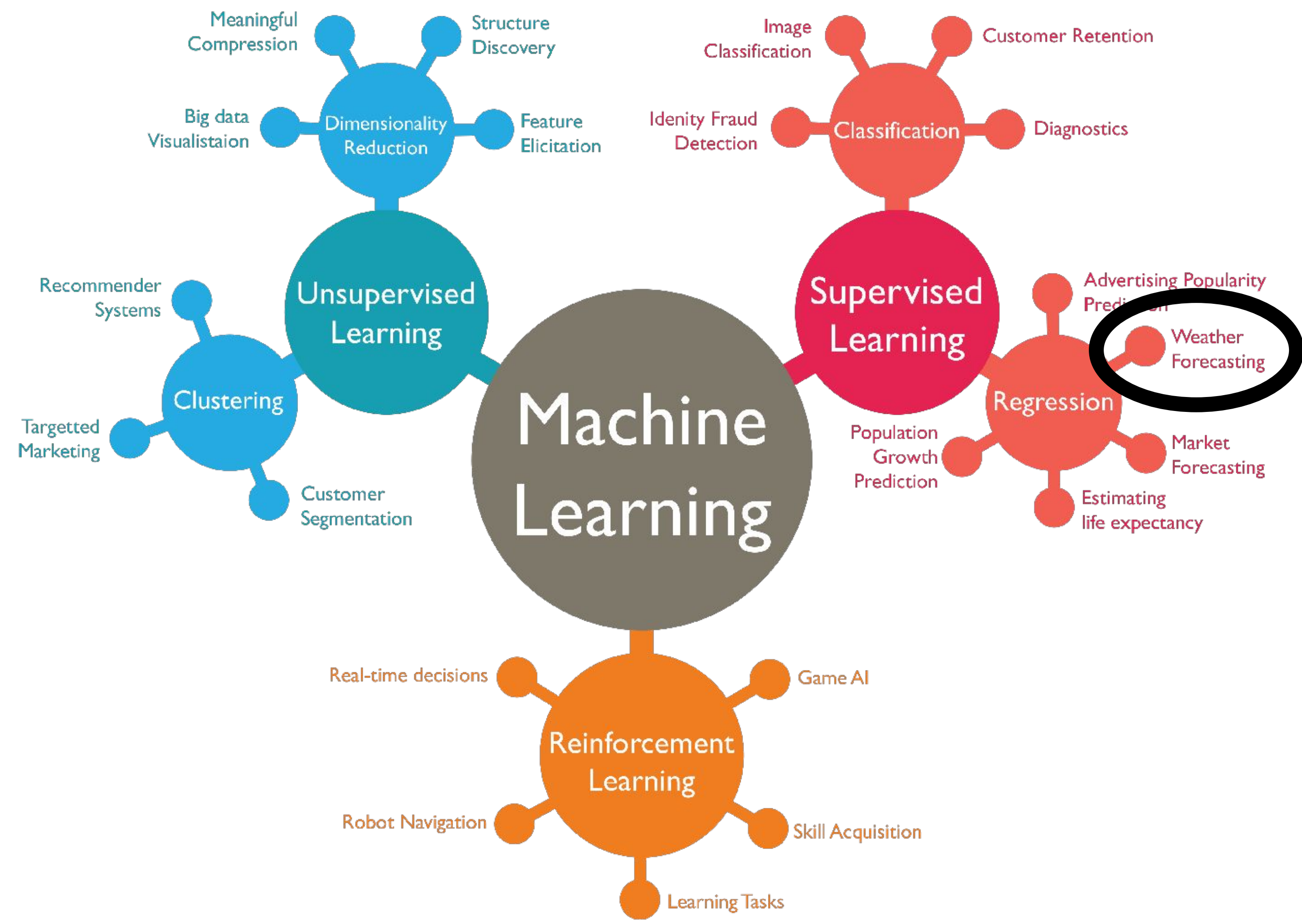
sea-ice concentration, C

Machine learning models that incorporate non-linearities between inputs capture important information.

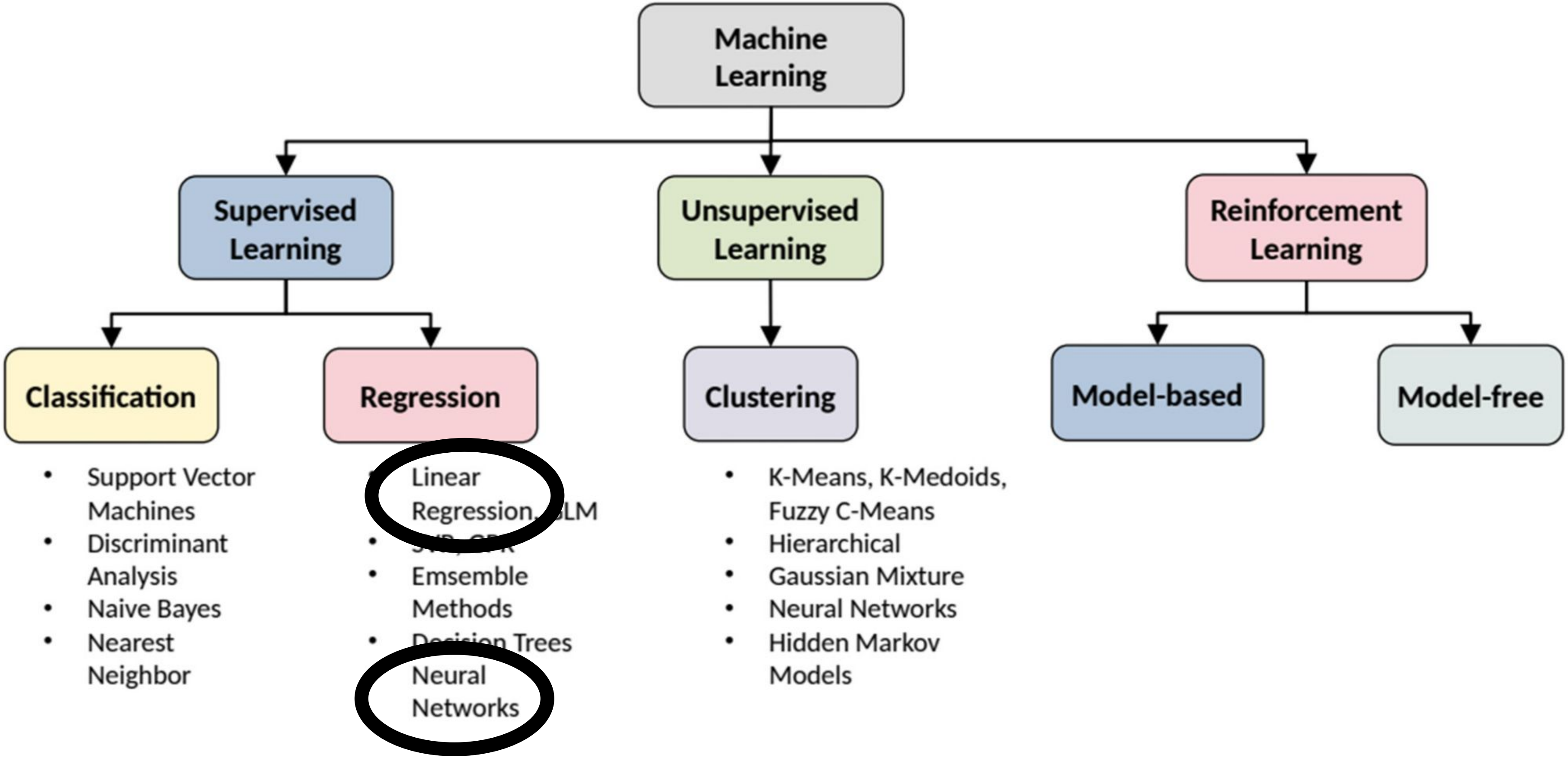
Machine learning confirms historical results that wind velocity has the largest relevance in determining sea-ice velocity.

Intro to ML

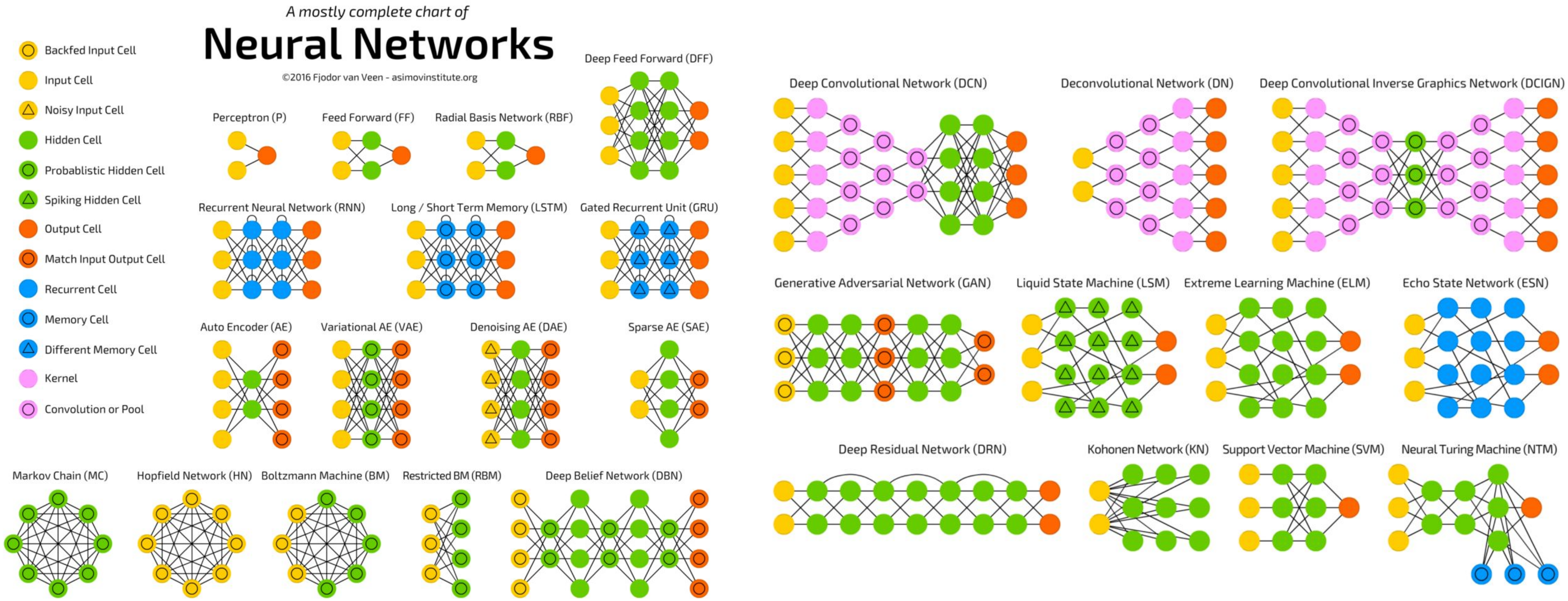
There are several different kinds of machine learning.



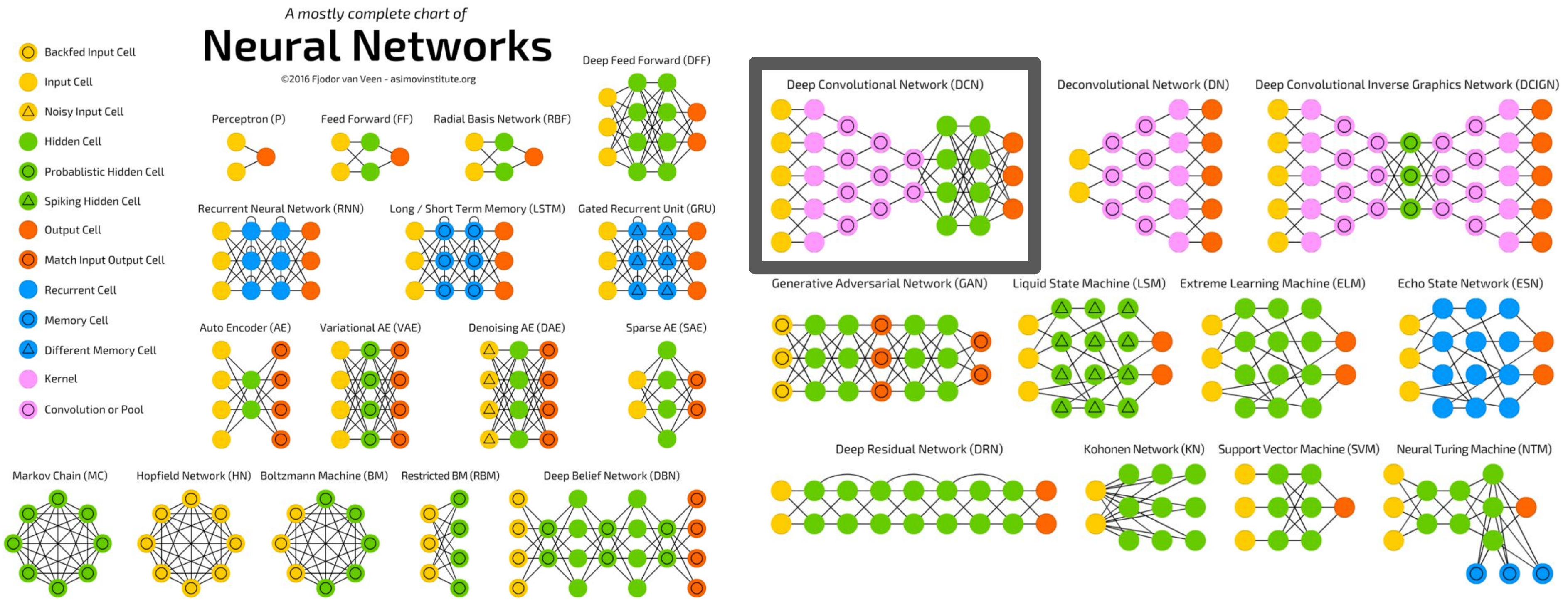
There are several different kinds of machine learning.



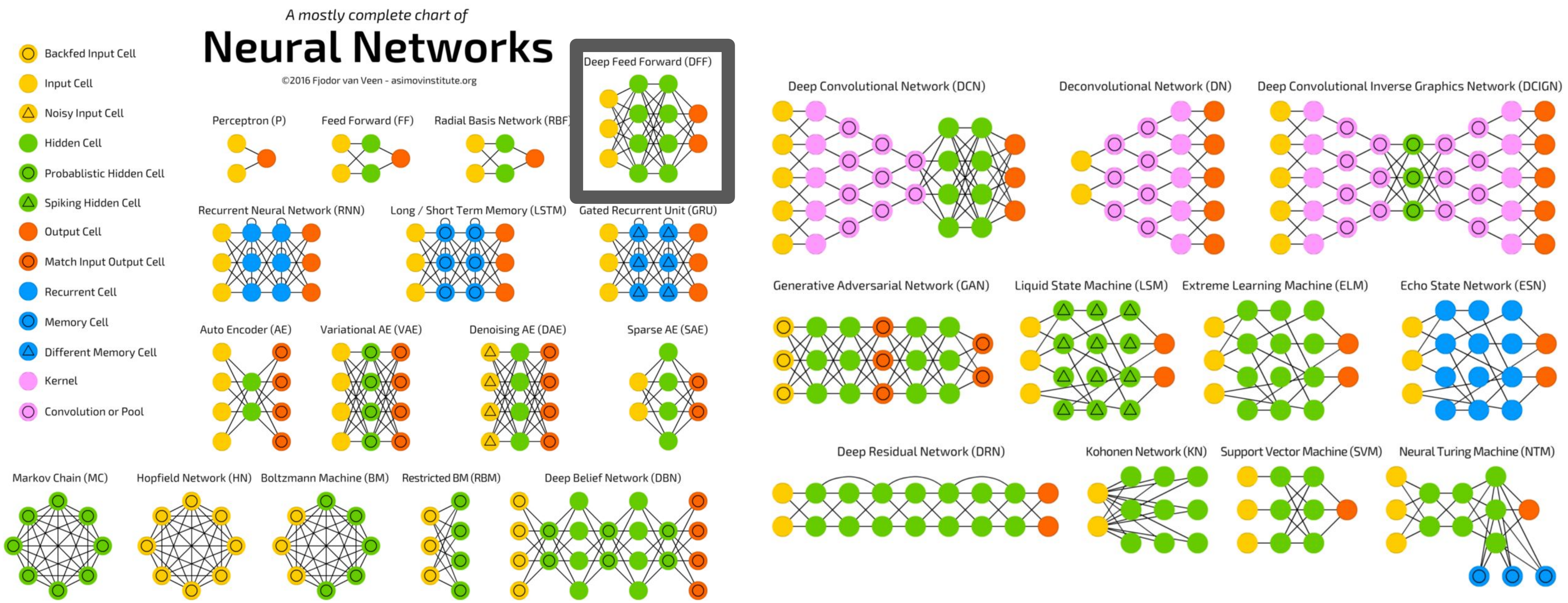
There are many different types of neural networks. Particular NNs can be beneficial depending on your application.



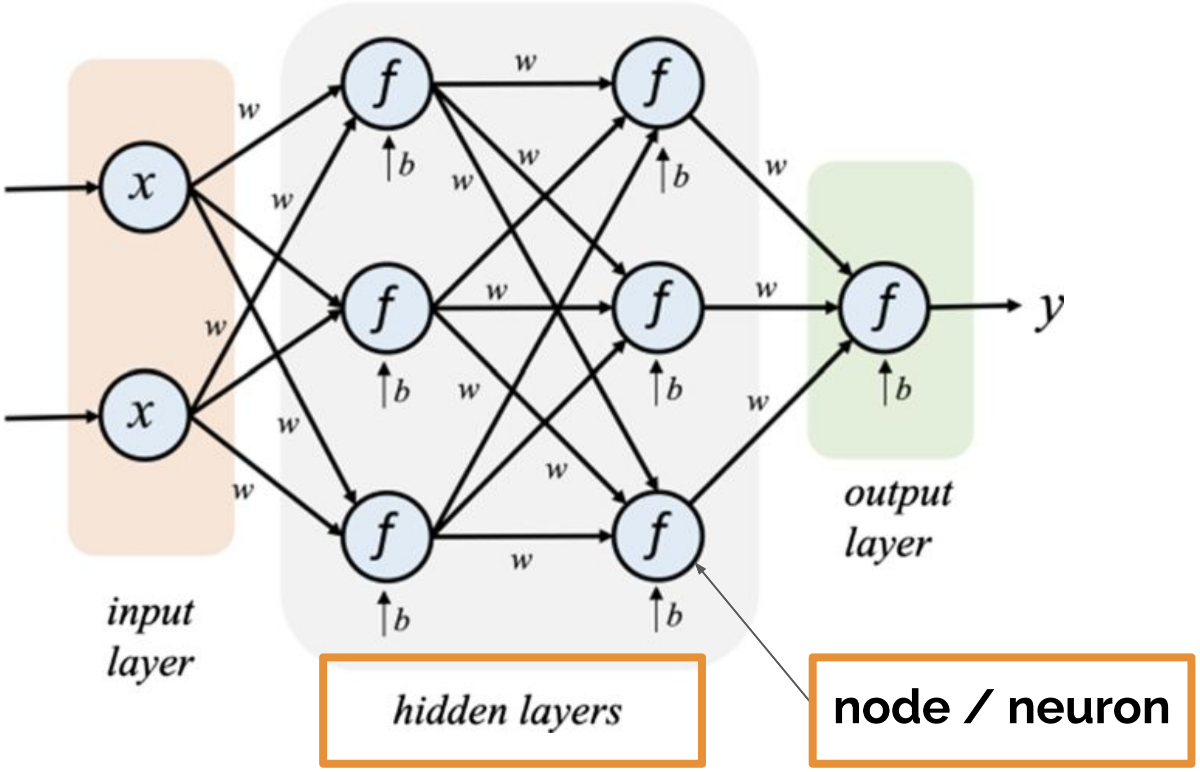
For example, my research on sea-ice motion used a *Convolutional Neural Network (CNN)* (here labeled DCN), which is particularly useful for images or image-like datasets (i.e. mapped products).



Today we will play with a tutorial that uses a *deep neural network* (here called DFF) with several hidden layers.

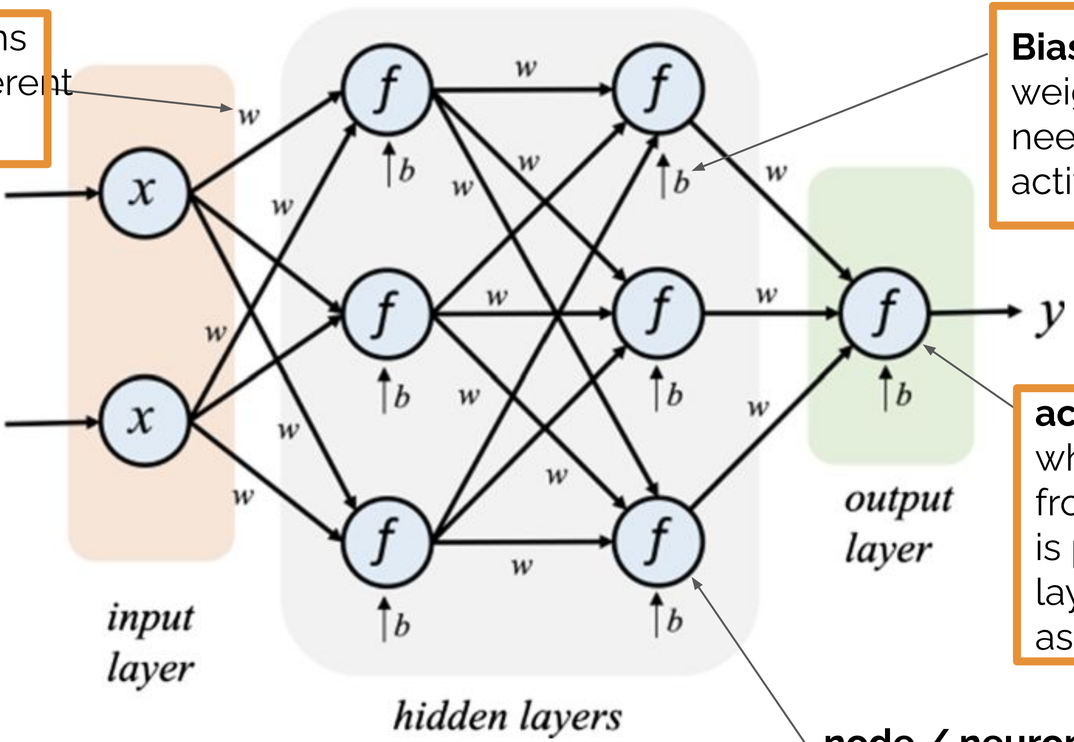


Today we will focus on neural networks (NNs).
This is an example of the architecture of a NN, which consists of several *nodes* embedded within multiple *hidden layers*.



Information is passed through a NN based on the *weights*, *biases*, and *activations*. These are the NN *parameters*.

weights, w: connections between nodes of different layers



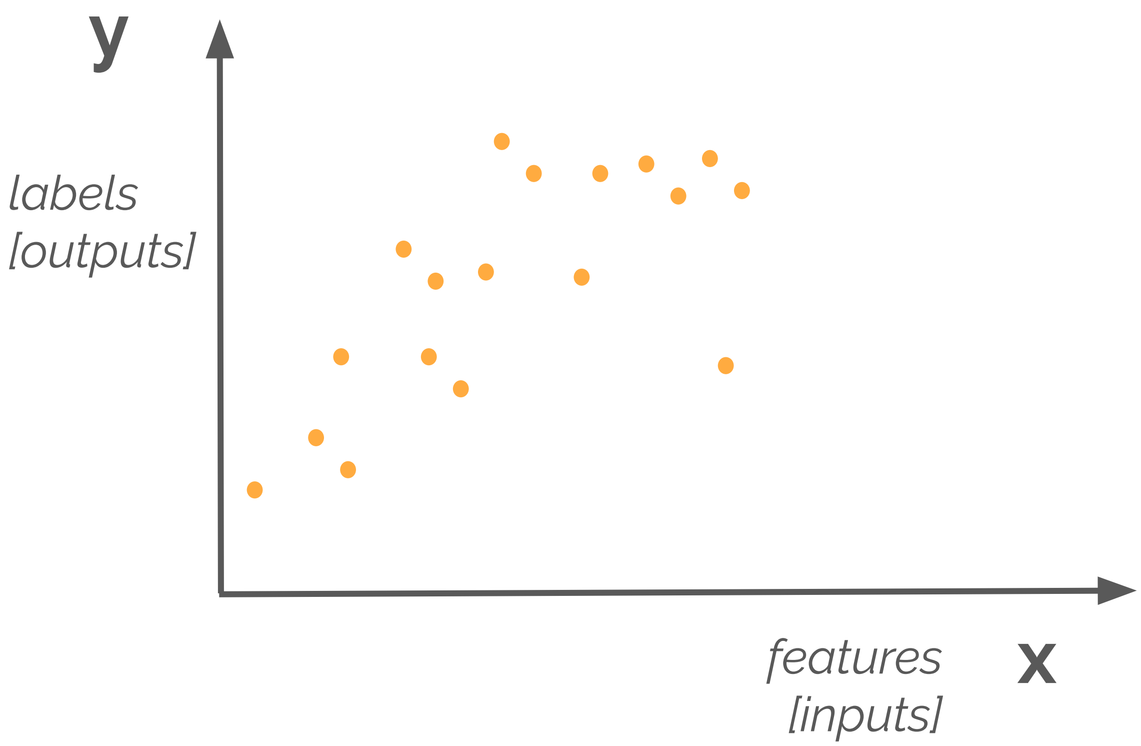
Biases, b: how high the weighted sum of $x_i w_i$ needs to be for activation of a neuron

activations, f: determine whether the information from the particular node is passed on to the next layer; sometimes labeled as ***a*** instead of ***f***

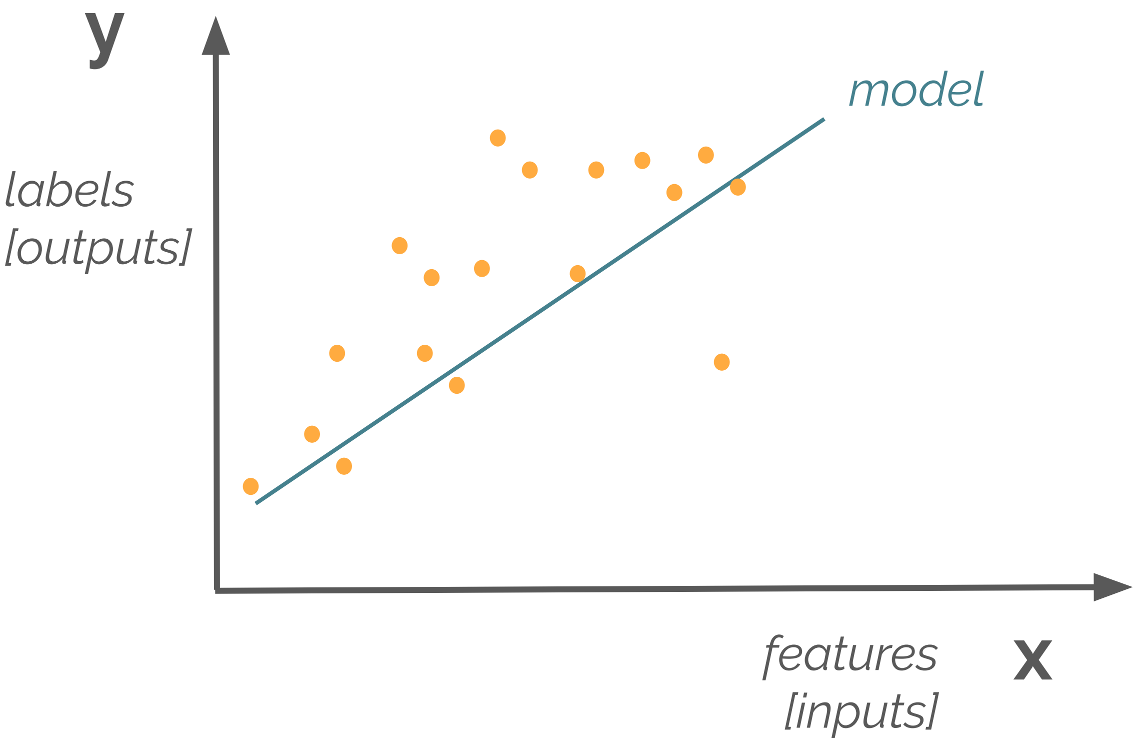
node / neuron

What is the machine “learning”?

How do you predict y from x?



Linear regression is an example of least squares regression.



Linear Regression

Model Architecture:

y = mx + b

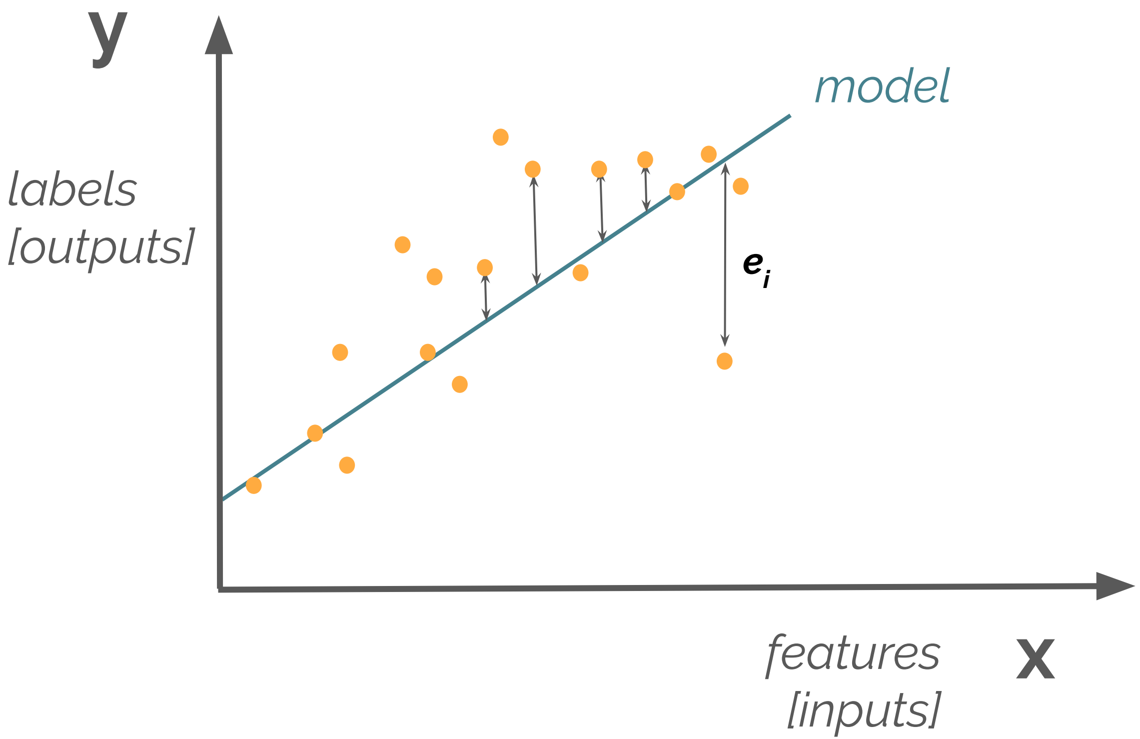
Model:

(m, b)

Model optimization: Minimize the **cost function**

- L_2 norm: $$||e||_2 = \left(\sum_{i=1}^{N} e_i^2\right)^{1/2}$$

A linear regression model is built based on minimization of the cost function (in this case the L2 norm or MSE).



Linear Regression

Model Architecture:

y = mx + b

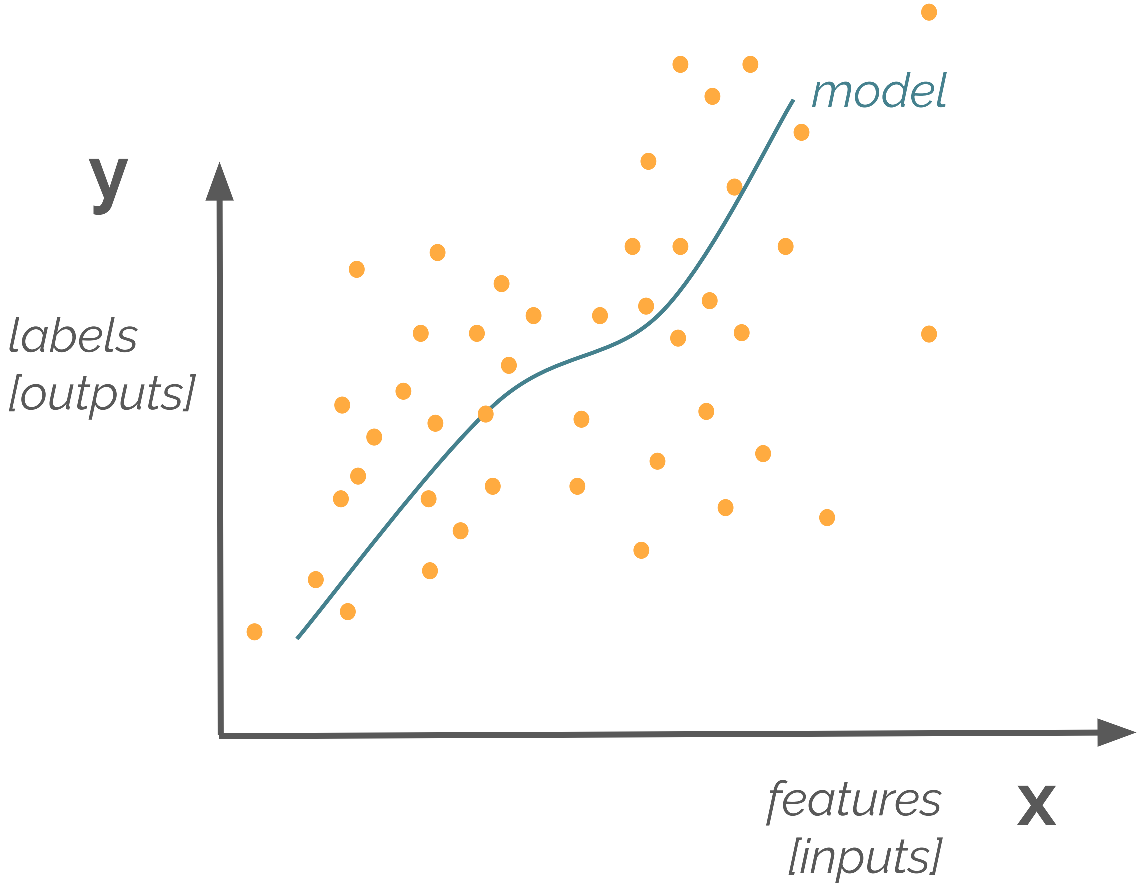
Model:

(m, b)

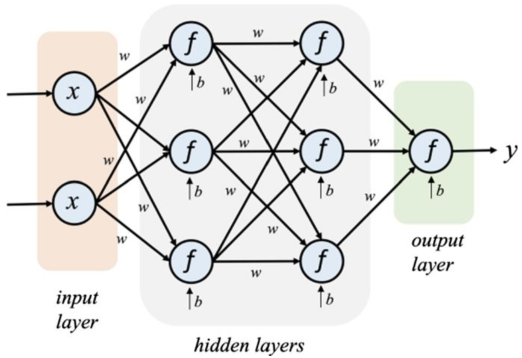
Model optimization: Minimize the **cost function**

- L_2 norm: $||e||_2 = \left(\sum_{i=1}^{N} e_i^2\right)^{1/2}$

Neural Networks are also trained to minimize a cost function.



Model Architecture:

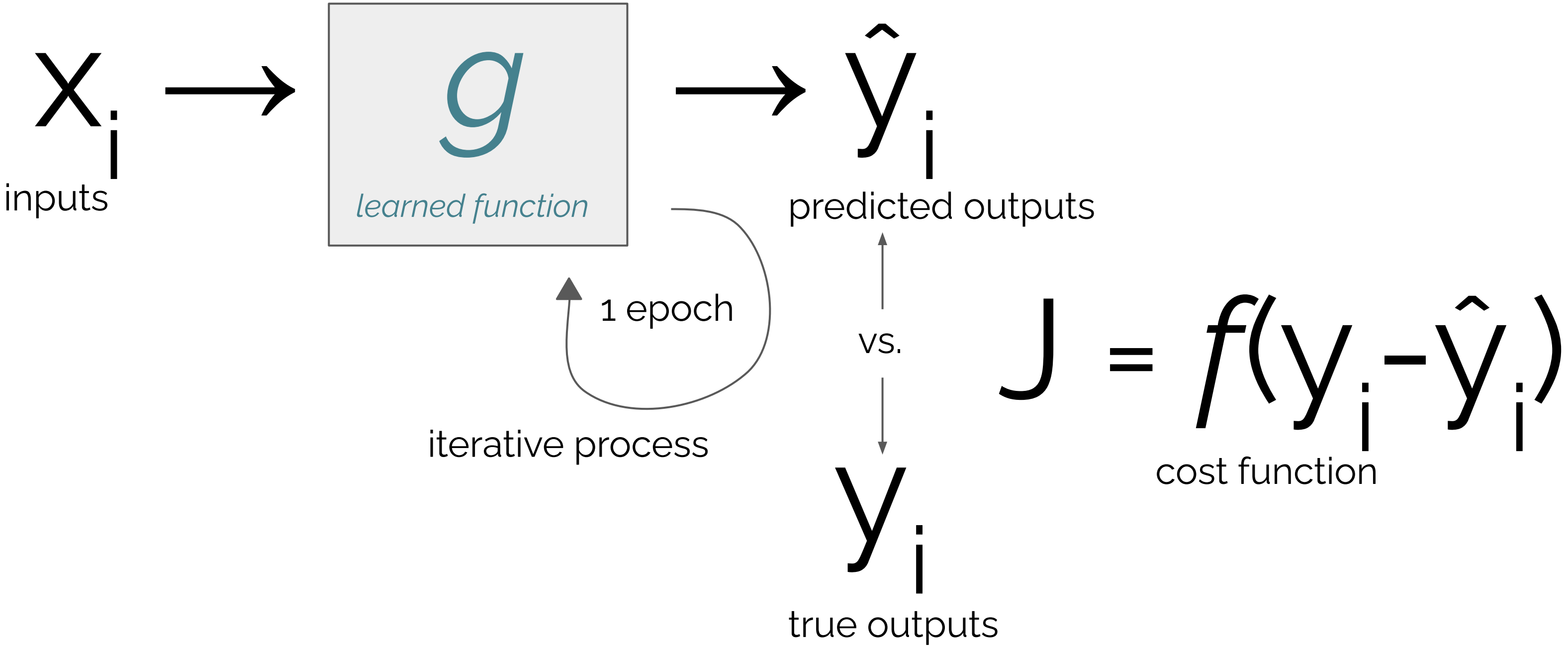


Model:

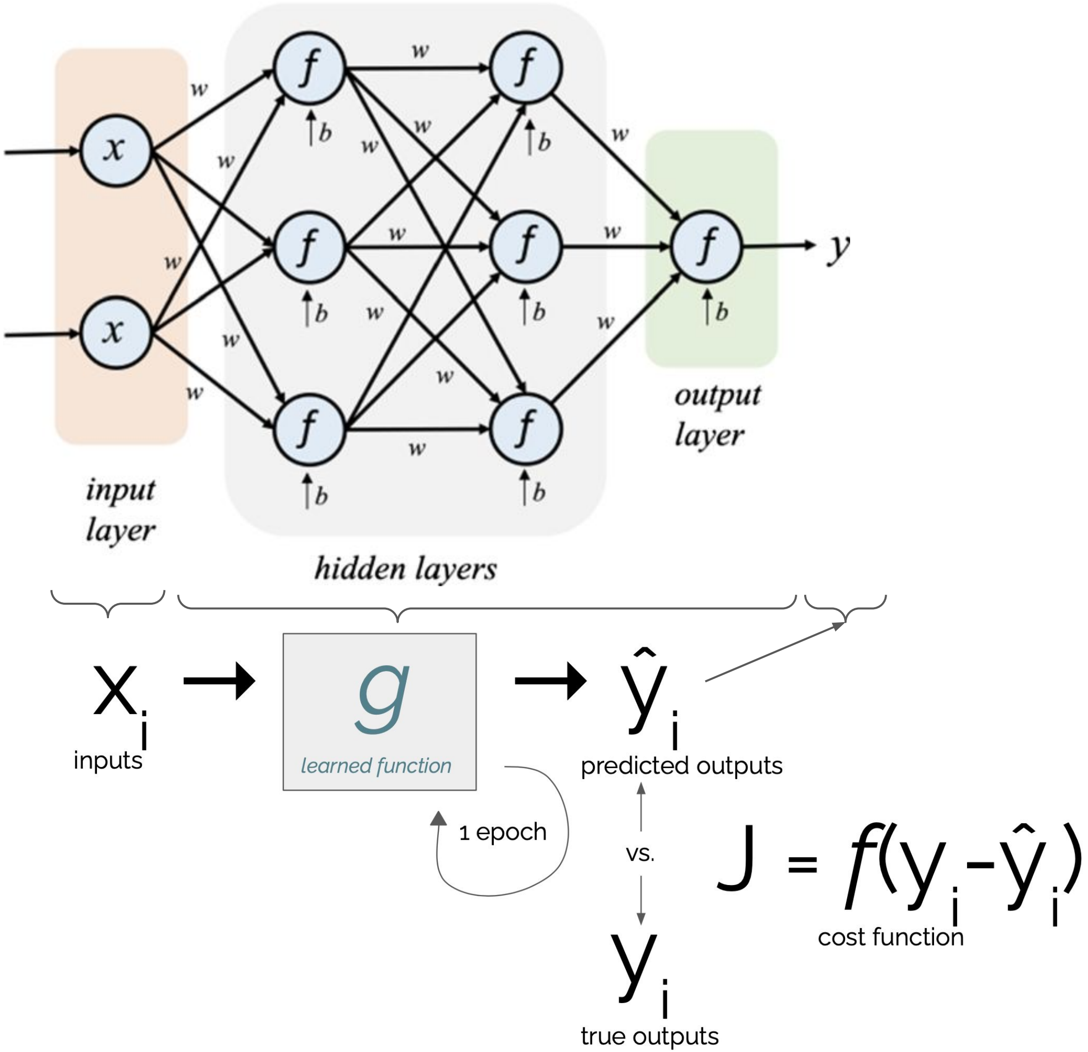
$$x \longrightarrow \boxed{g} \longrightarrow \hat{y}$$

Model optimization: Minimize the **cost function** through the *iterative* process of *gradient descent*.

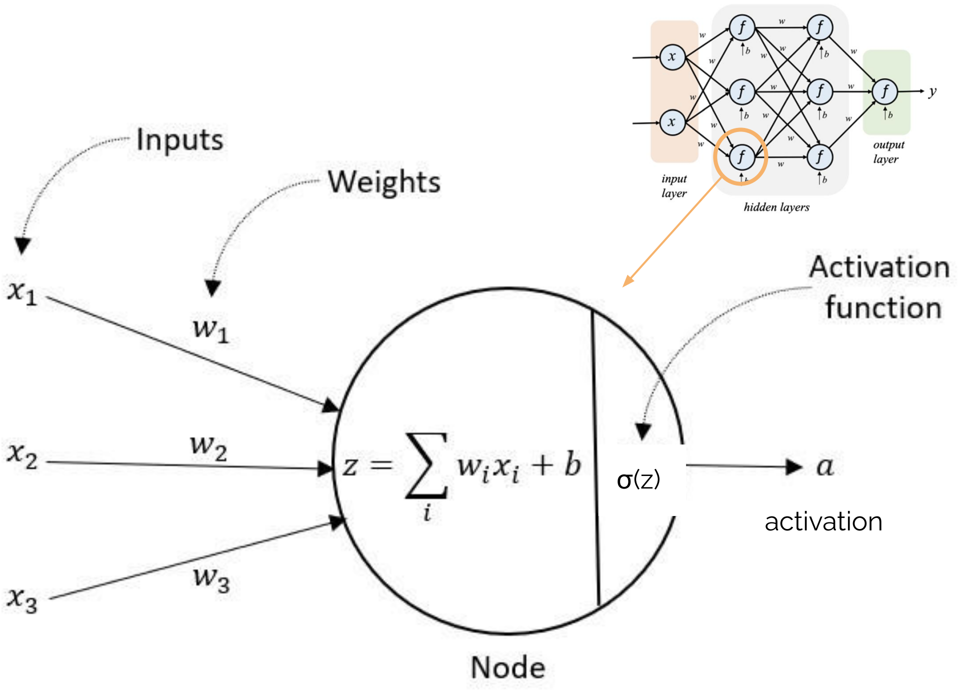
Neural Networks are trained through an *iterative process* that works to minimizing the cost function.



The learned function, g, is representative of the *learned parameters* within the layers of the model (i.e. weights, biases, activations).



What happens within each node?



$$a = \sigma\left(\Sigma w_i x_i + b\right)$$

Activation of each neuron = activation function (sum (weights * inputs) + biases)

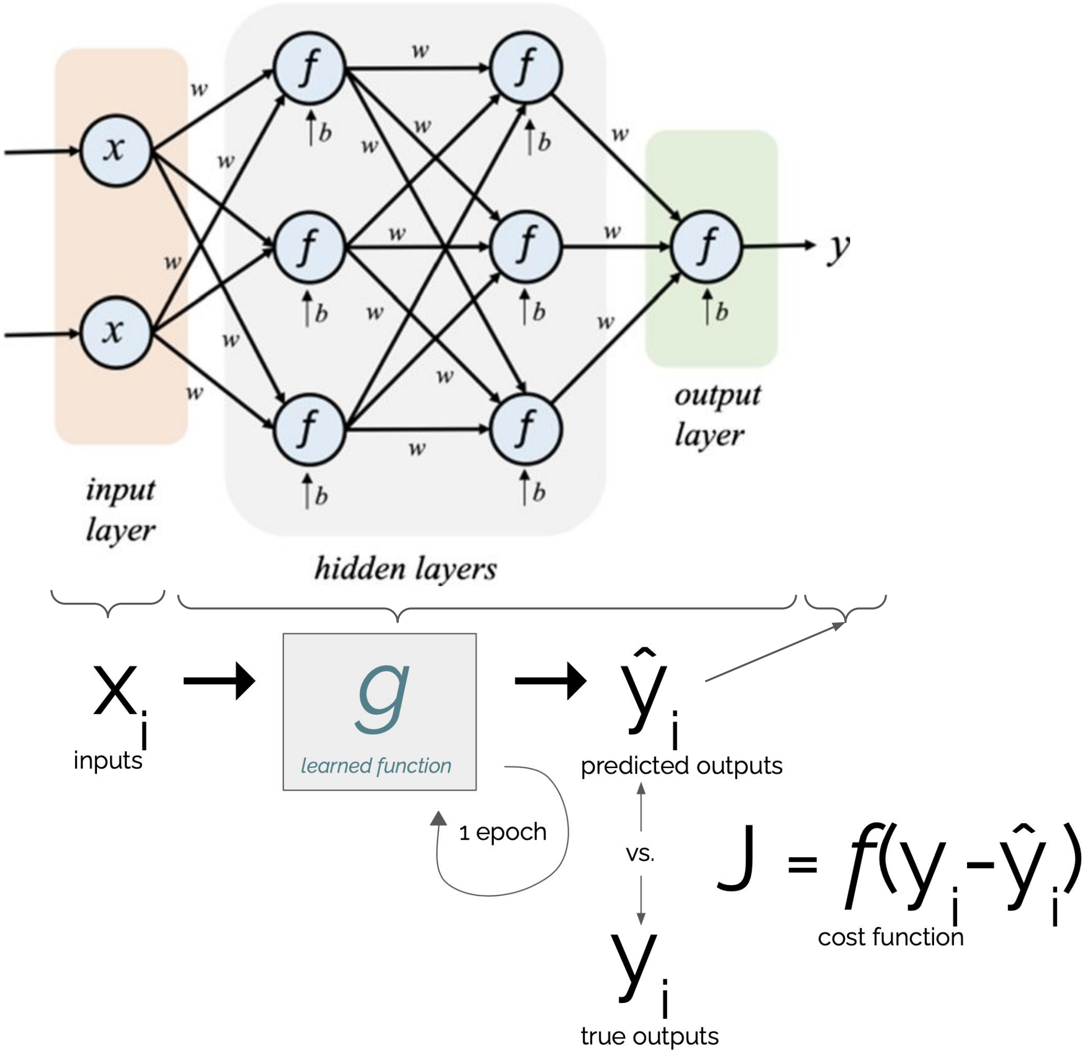
Parameters:

- **Inputs, x_i**
- **Weights, w_i**: connections between neurons
- **Biases, b:** how high the weighted sum of $x_i w_i$ needs to be for activation of a neuron
- **Activations, a**: number inside each neuron; determines whether or not neuron is "lit up"

Hyperparameters:

- **Activation function, σ** (sigmoid, ReLU, tanh, etc.): determines if information from specific node is propagated forward in the NN
- **Number of neurons / nodes**
- **Number of layers**
- **Number of epochs / iterations**

The learned function, g, is representative of the *learned parameters* within the layers of the model (i.e. weights, biases, activations).



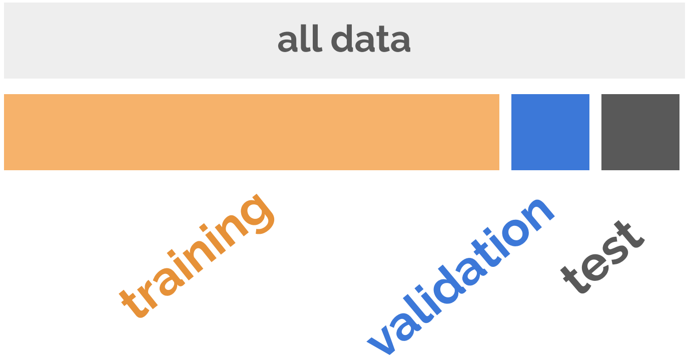
Before training your model it is important to apply *feature scaling* and *data splitting* to your dataset.

Feature Scaling makes it so inputs are on similar scales.

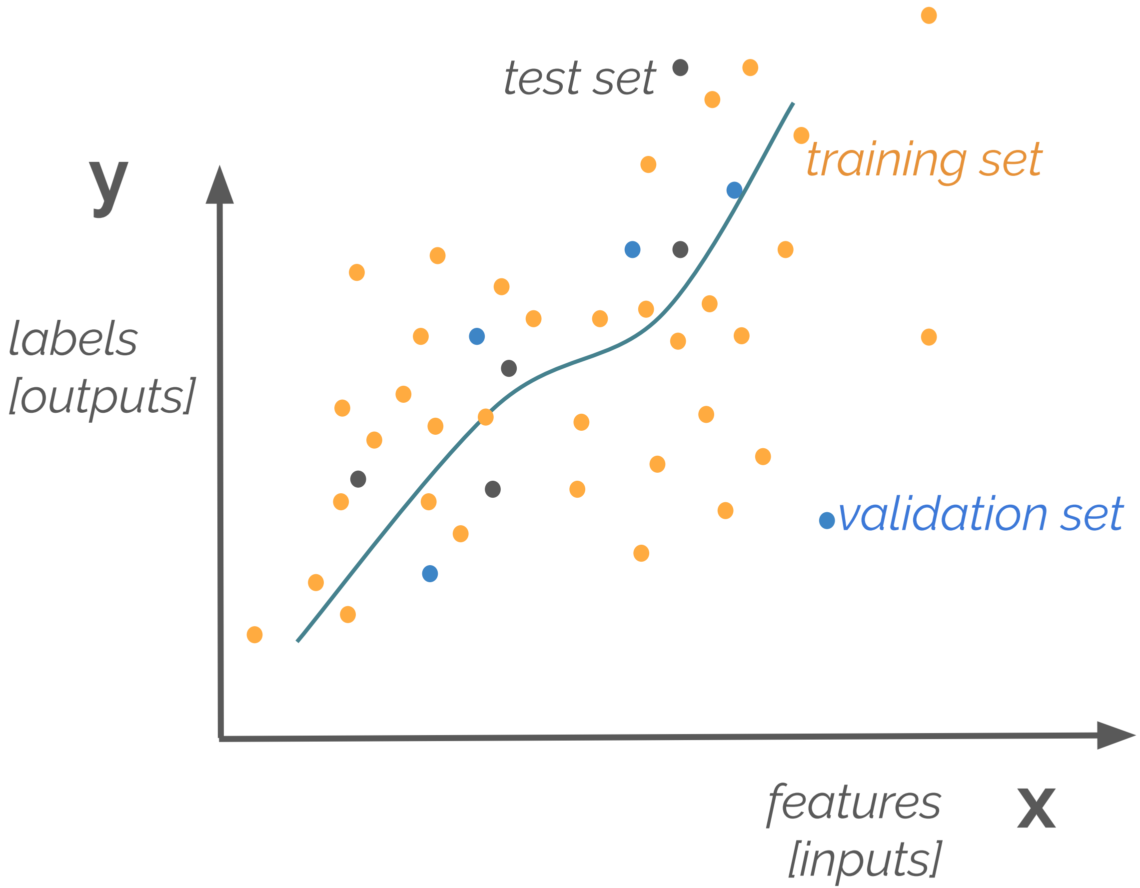
Apply either:

- **Standardization:** zero mean, one standard deviation
  $data_S = (data - \mu)/\sigma$
- **Normalization:** from 0 to 1
  $data_N = [data - max] / [min - max]$

Data Splitting



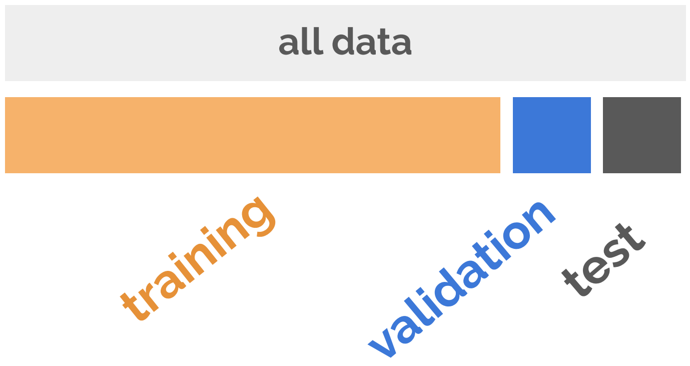
Data are split into training, validation and testing sets.



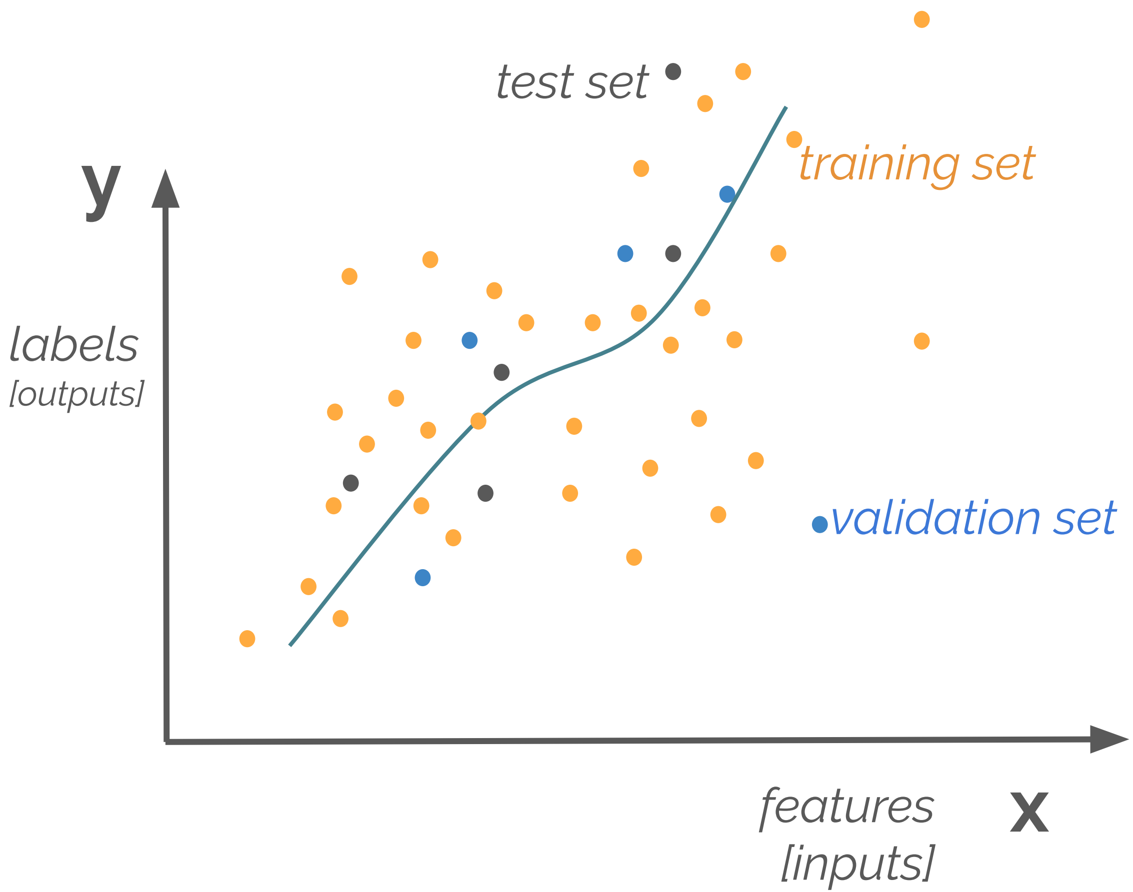
Data Splitting

Why do we split data?

- Reduce overfitting
- Optimize *hyperparameters*



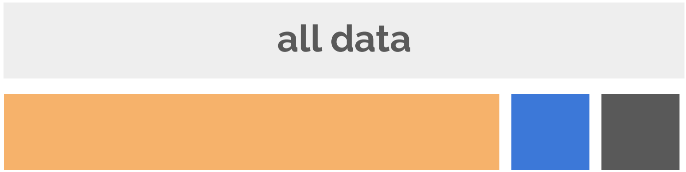
Data are split into training, validation and testing sets.



training: Data used to fit the model. The subset of data that the model uses to *learn* (i.e. optimize the cost function)

validation: Data subset used to optimize model *hyperparameters* during training.

test: Data subset used to evaluate the model on data it has not seen before.



ML Resources

Cognitive Class [free]
AI: https://cognitiveclass.ai/courses/machine-learning-with-python
Python: https://cognitiveclass.ai/courses/python-for-data-science

Coursera [free week trial]
https://www.coursera.org/learn/neural-networks-deep-learning

Beucler Lab at UNIL [data-driven atmospheric and water dynamics]
https://wp.unil.ch/dawn/getting-started-with-machine-learning/#site-header

Exercise

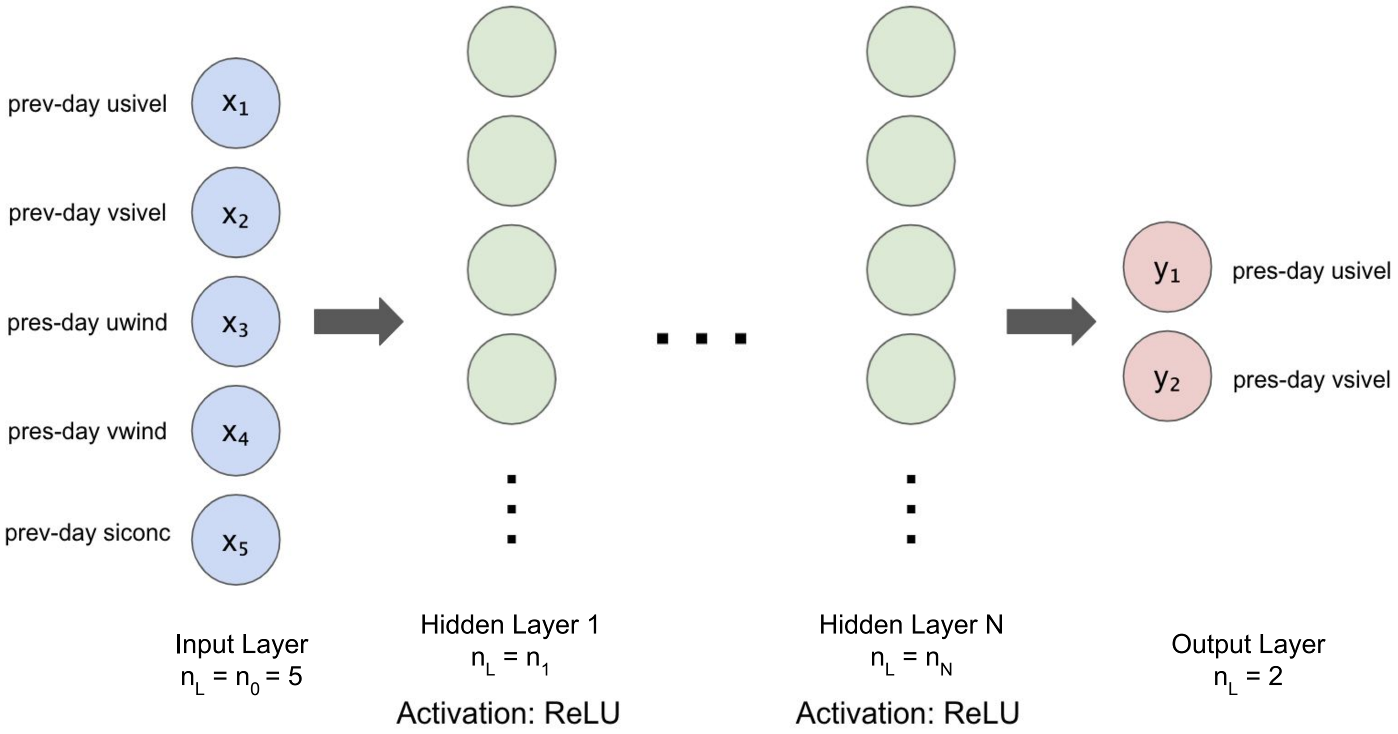
Tutorial: *predicting sea-ice velocity using a Neural Network.*

Everything required to run this tutorial can be found at:
https://github.com/lahoffman/ml_tutorial

Open the Google Colaboratory Notebook (‘nn_regression.ipynb’) and follow the instructions.

Make sure to read and follow the instructions under “0. Set up” before you begin (i.e. make a copy of the notebook that you can edit & get the data in your drive).

General NN architecture



Coding Environments

TensorFlow Keras: https://www.tensorflow.org

Choices you can to make...

Model Inputs & Outputs:

- Under "**Section 1.3.3 Create Input & Output Data**" you can choose different inputs and outputs depending on the question you want to ask.

Hyperparameters:

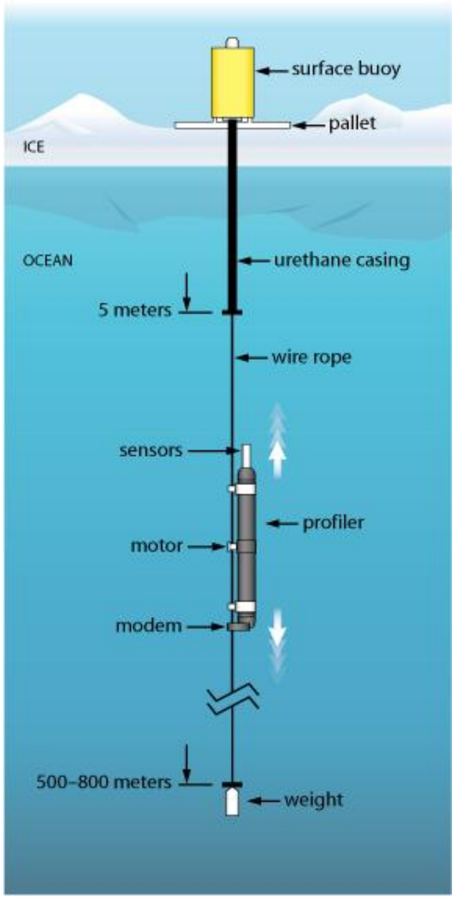
- **Activation function, σ** (sigmoid, ReLU, tanh, etc.): determines if information from specific node is propagated forward in the NN
- **Number of neurons / nodes**
- **Number of layers**
- **Number of epochs / iterations**
- **Batch size**
- **Learning rate**
- **Gradient descent algorithm**
- **Regularization**

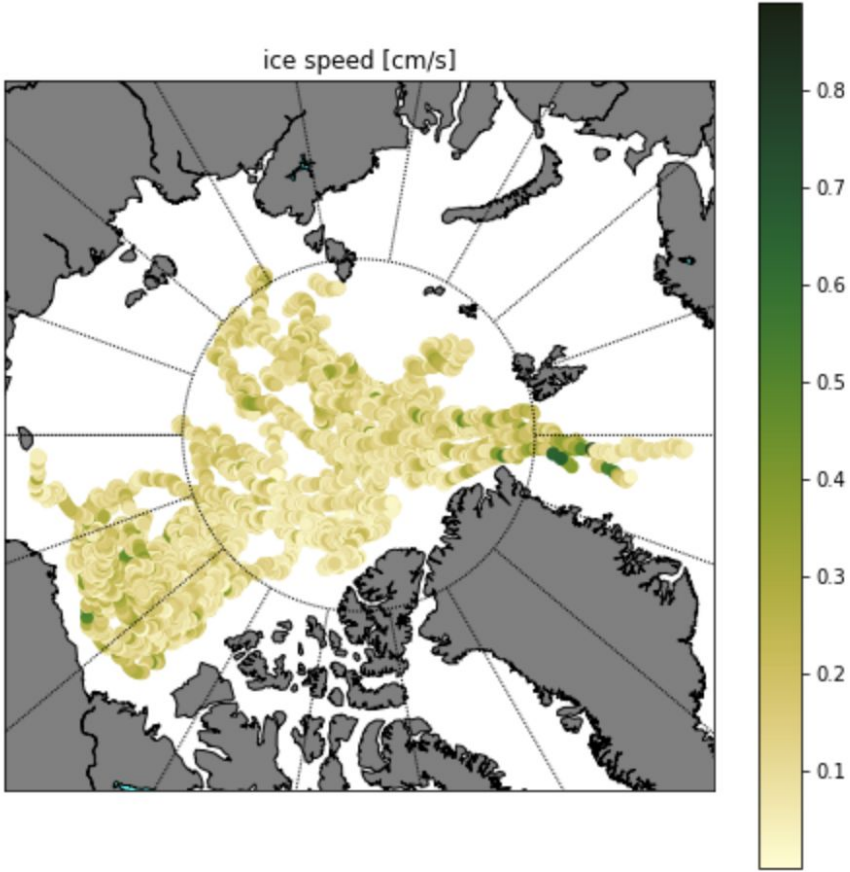
What the NN chooses for you...

Parameters:

- **Inputs, x_i**
- **Weights, w_i**: connections between neurons
- **Biases, b:** how high the weighted sum of $x_i w_i$ needs to be for activation
- **Activations, a**: number inside each neuron; determines whether or not neuron is "lit up"

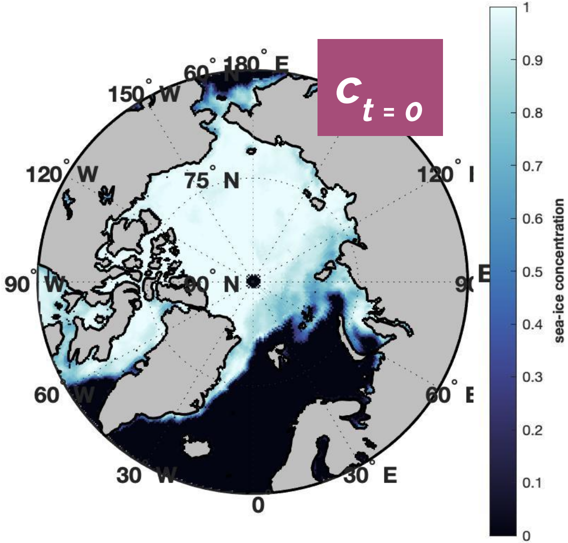
Ice-Tethered Profilers: *Sea-Ice Velocity*



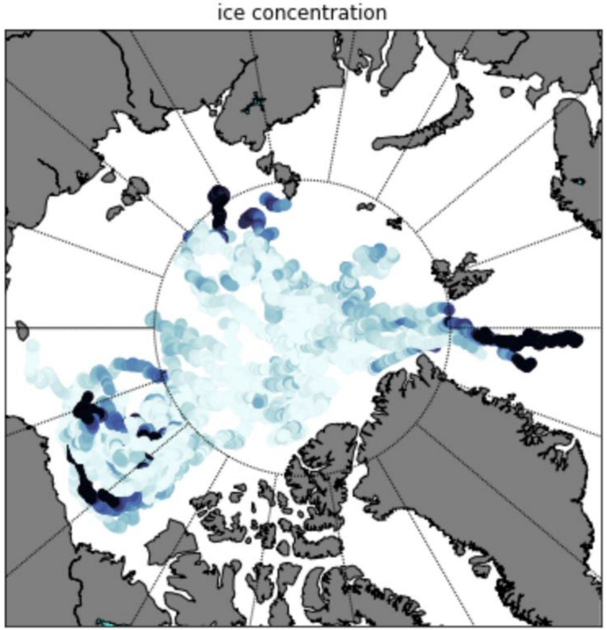


The Ice-Tethered Profiler data were collected and made available by the Ice-Tethered Profiler Program (Toole et al., 2011; Krishfield et al., 2008) based at the Woods Hole Oceanographic Institution (https://www.whoi.edu/itp).

Nimbus-7 Passive Microwave: *Sea-Ice Concentration*

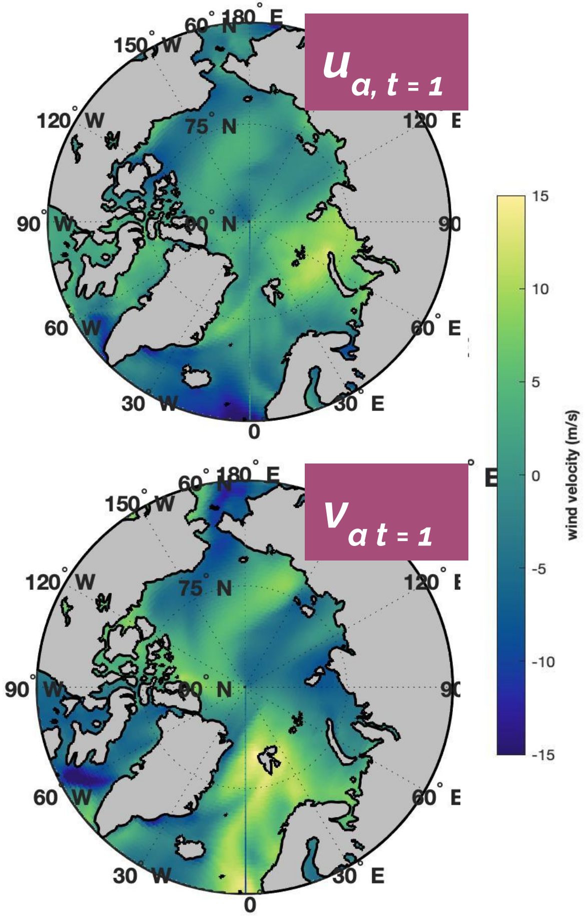


co-located to the ITP in time and space

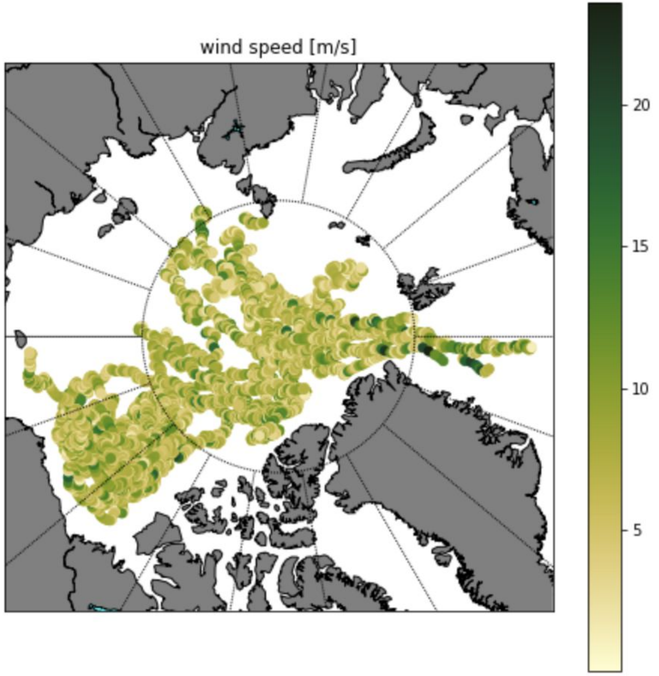


Note: the figure on the left is ci plotted for a particular day, while the figure on the right contains sea-ice concentration co-located to the ITP data in space and time. Note the difference in the orientation of the map.

Japanese 55-year Reanalysis: *Wind Velocity*



co-located to the ITP in time and space



Note: the figure on the left shows wind velocity vectors plotted for a particular day, while the figure on the right contains wind speed co-located to the ITP data in space and time. Note the difference in the colorbar, as well as the orientation of the map.

Image Sources

- Chihuahua or Blueberry Muffin: https://www.freecodecamp.org/news/chihuahua-or-muffin-my-search-for-the-best-computer-vision-api-cbda4d6b425d/
- Types of ML 1: https://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications
- Types of ML 2: https://www.mdpi.com/2071-1050/13/9/5248
- Neural Network Architectures: https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464
- NN Architecture: Ünal, H.T., Başçiftçi, F. Evolutionary design of neural network architectures: a review of three decades of research. *Artif Intell Rev* 55, 1723–1802 (2022). https://doi.org/10.1007/s10462-021-10049-5
- Inside a neuron: https://levelup.gitconnected.com/a-review-of-the-math-used-in-training-a-neural-network-9b9d5838f272
- Some slides adapted from: https://zenodo.org/record/6686879#.ZAY22y-B1DM