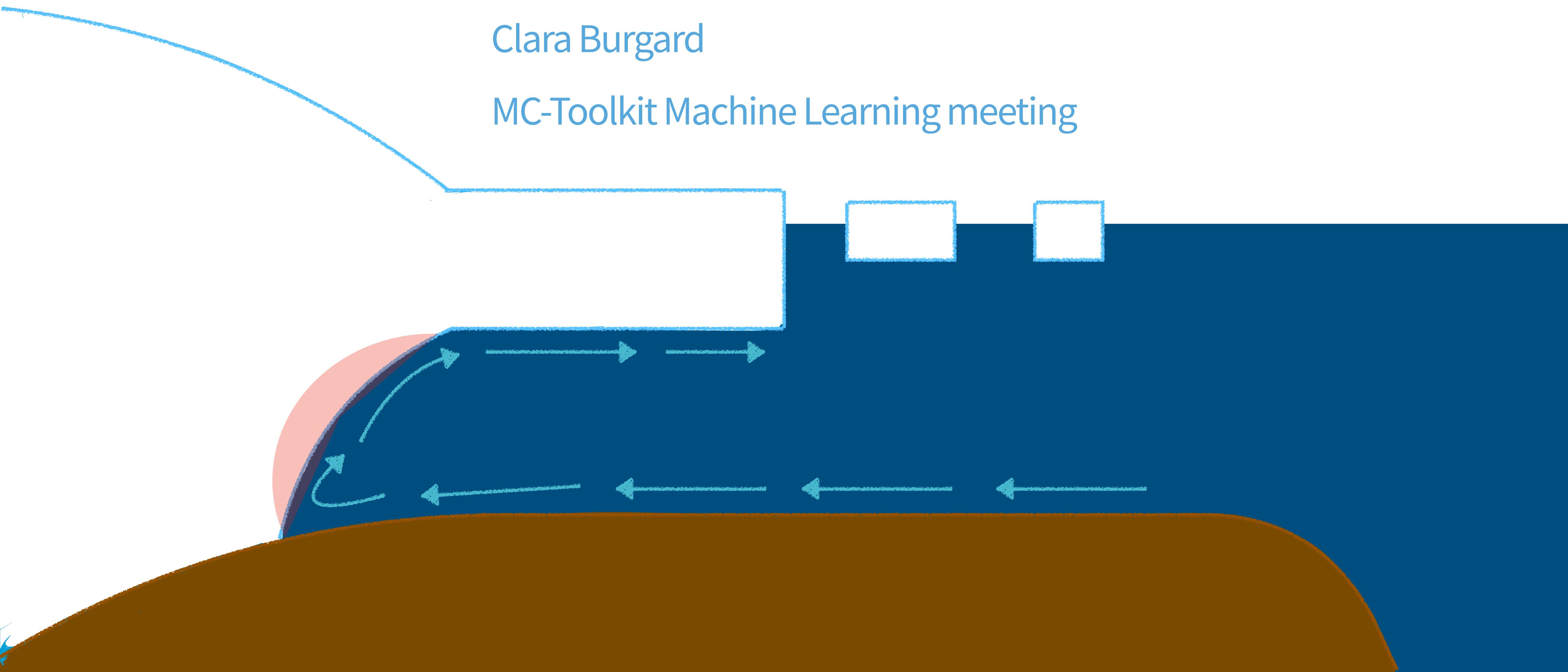


Representing the ocean-induced sub-shelf melt in Antarctica with a neural network?

Clara Burgard

MC-Toolkit Machine Learning meeting

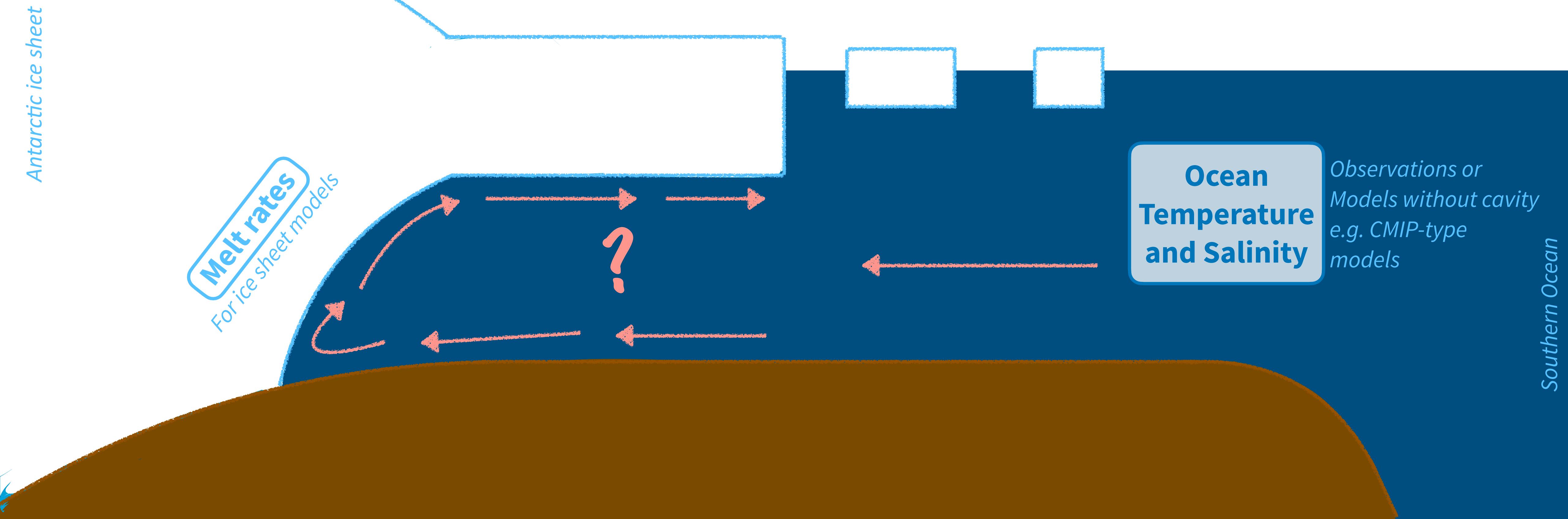


The problem: Representing ocean basal melting in (uncoupled) ice sheet models

Ice sheet models need information about the ocean water interacting with the ice at the lower boundary of the ice shelves...

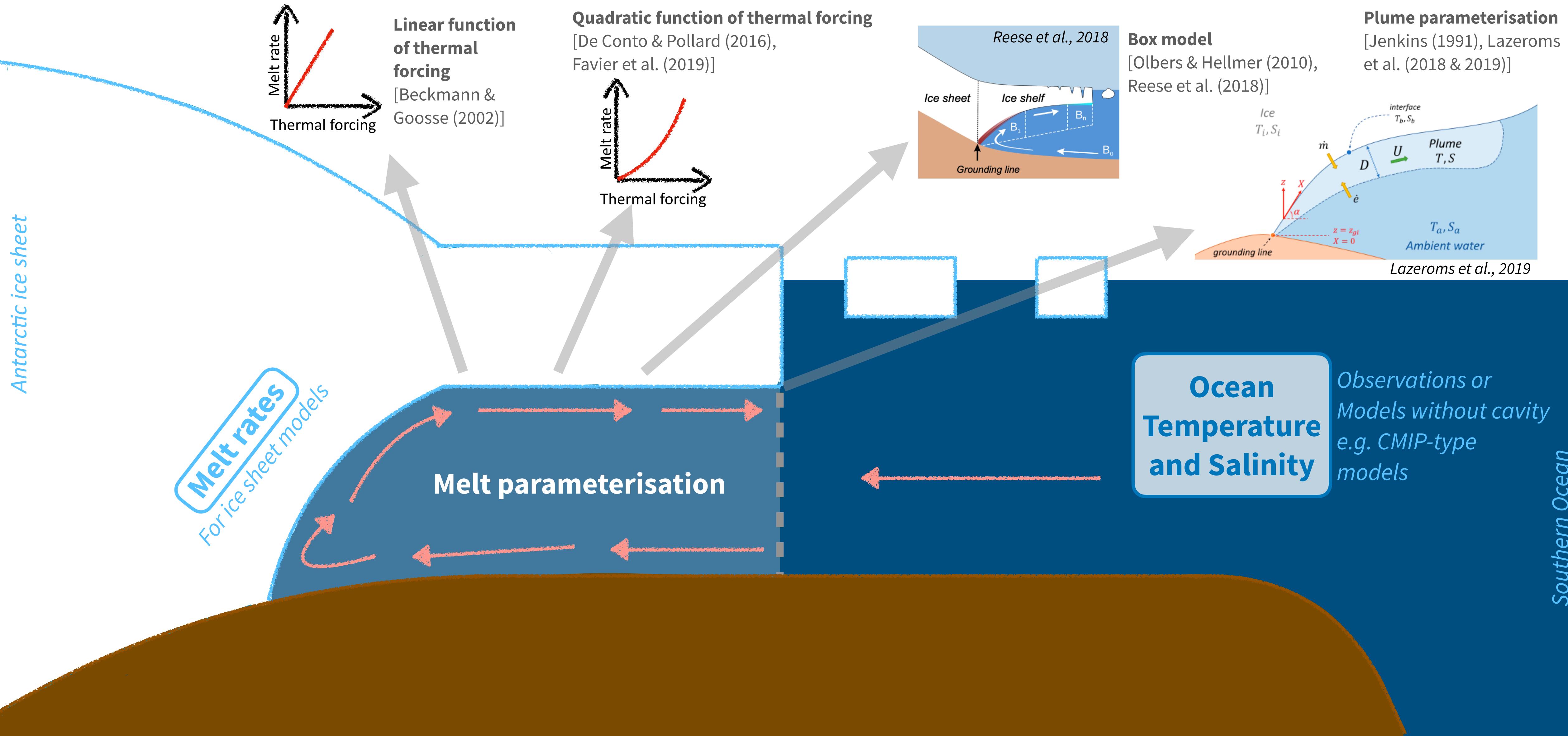
However: Sub-shelf cavities are typically not resolved in ocean or coupled-climate models...

The link between the open ocean and the ocean-ice sheet interface is missing!



Until now: Assess and improve existing ocean basal melting parameterisations for ice sheet models

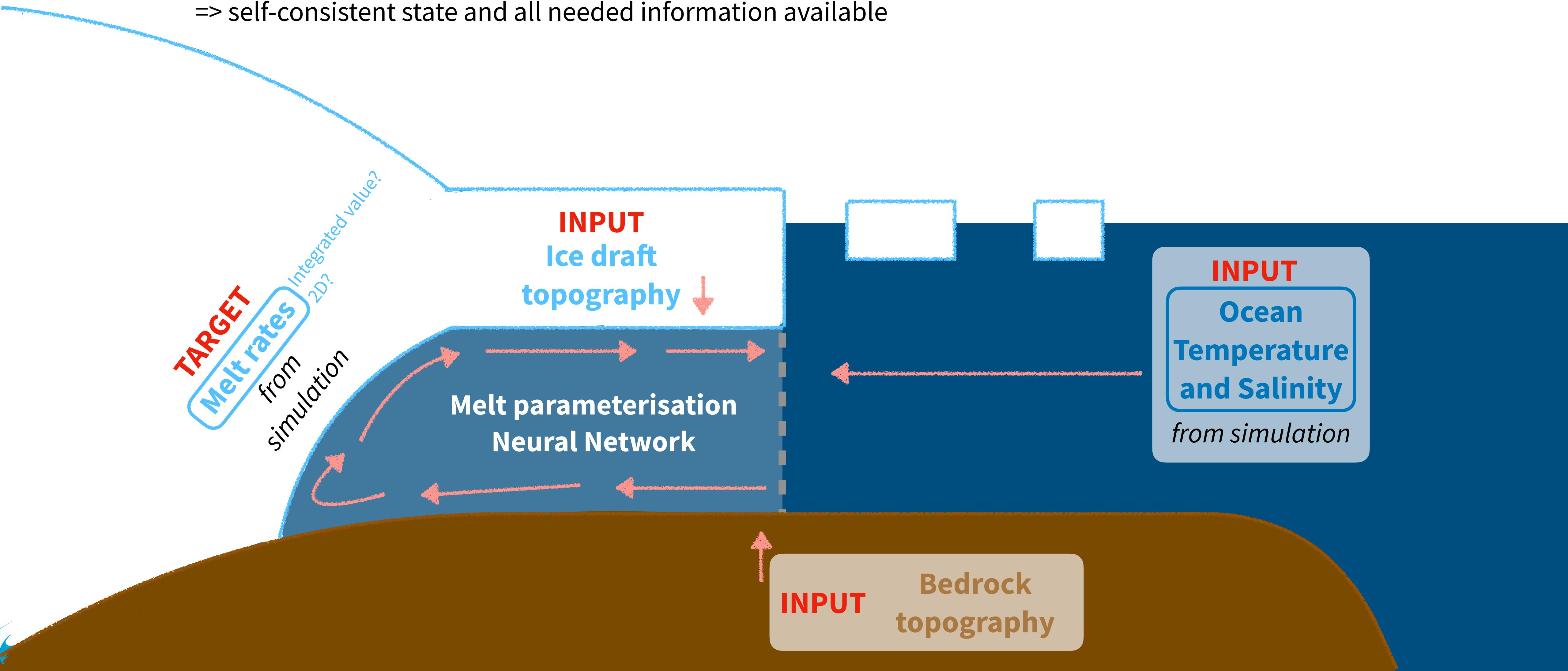
We assess and tune existing sub-shelf melt parameterisations in a circum-Antarctic approach.



Idea: What I wanted to try with neural networks

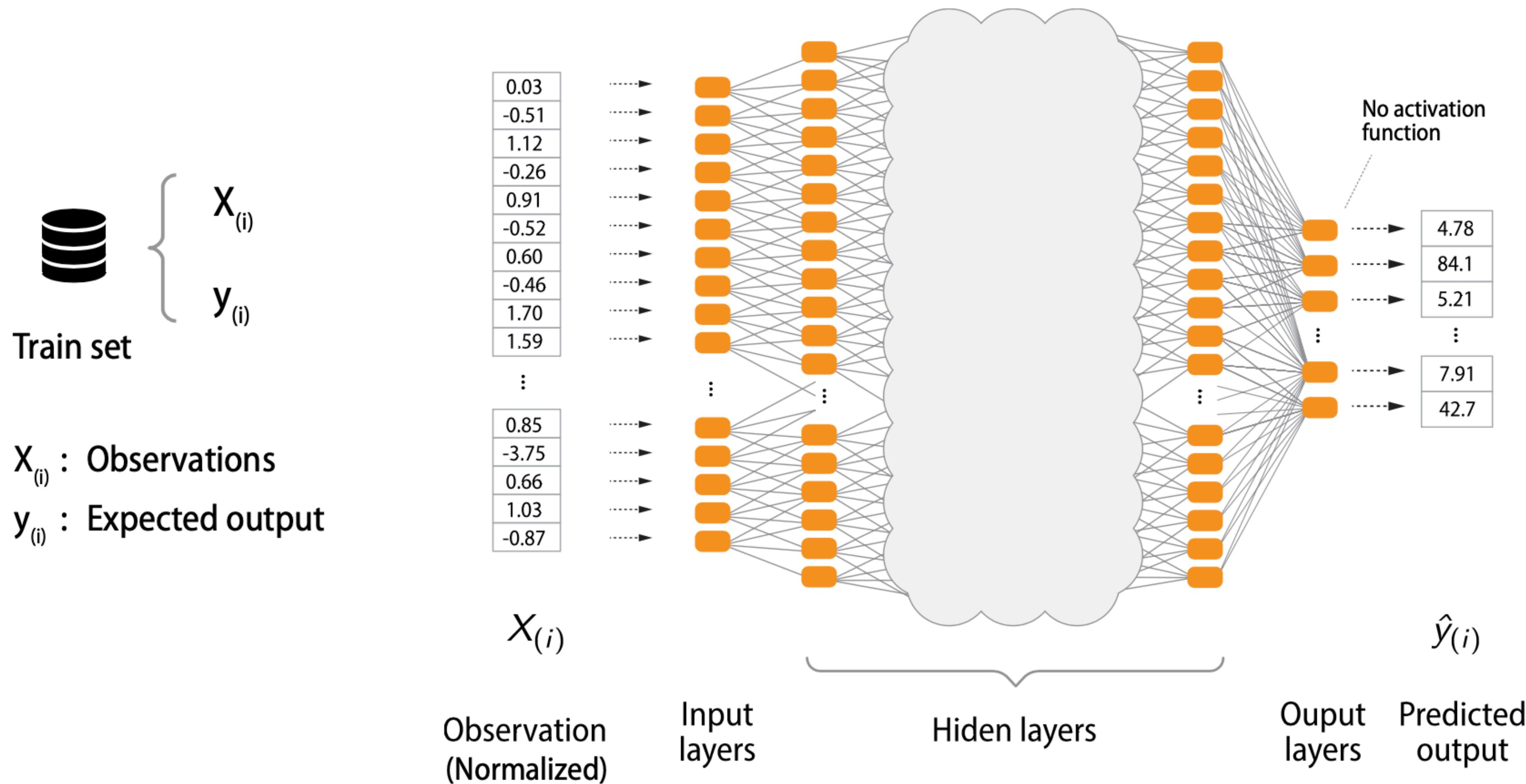
“Perfect model” approach

We could use circum-Antarctic ocean simulation (resolving cavities) or ideal experiments (MISOMIP) as a virtual reality to train a neural network
=> self-consistent state and all needed information available



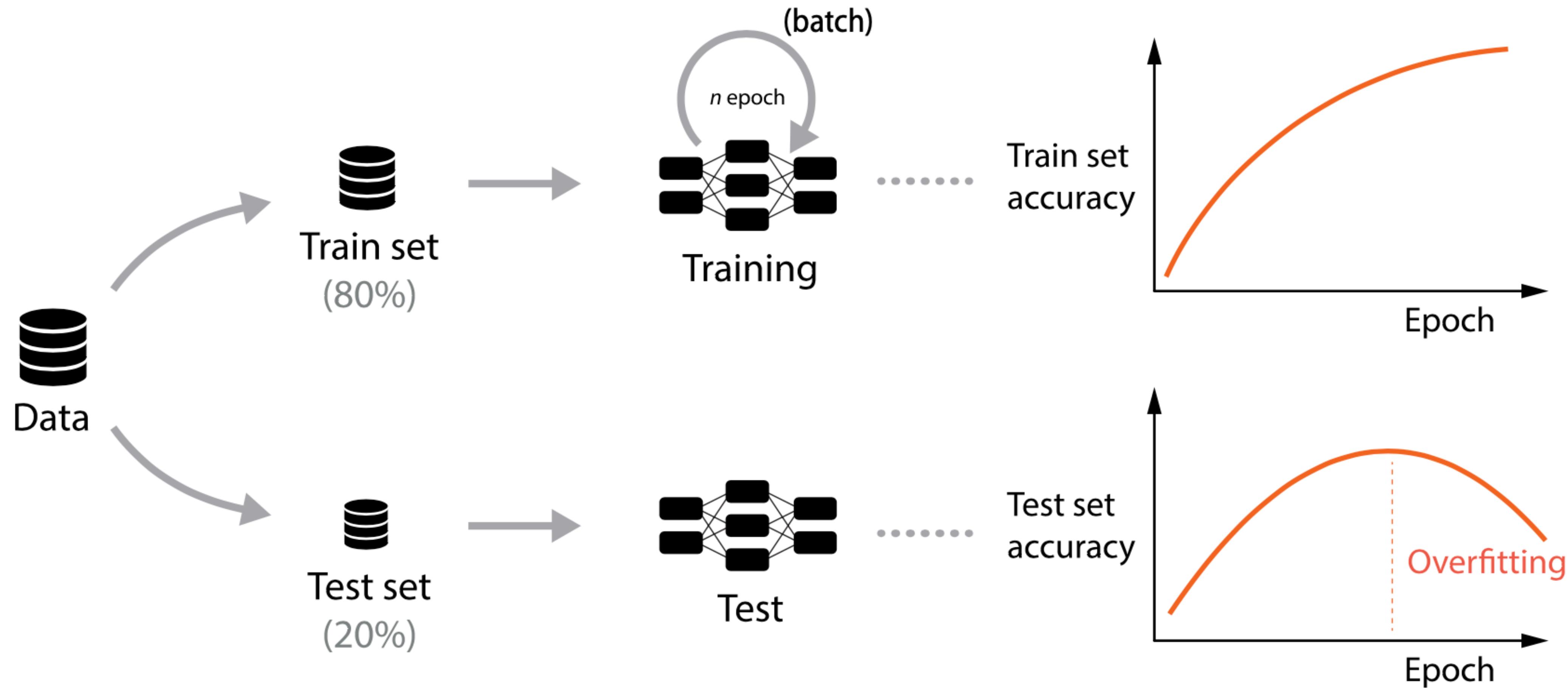
How to approach the problem?

Regression with a DNN



How to train the model?

Training process - general



Implementation steps

Step 1 - Import and init



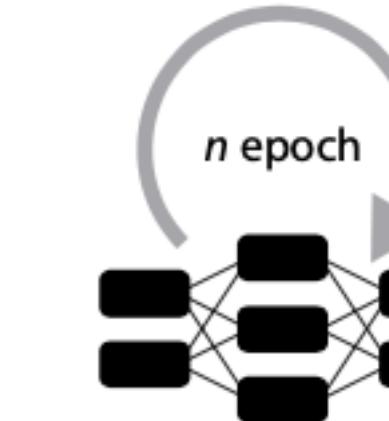
Step 4 - Build a model



Step 2 - Retrieve data



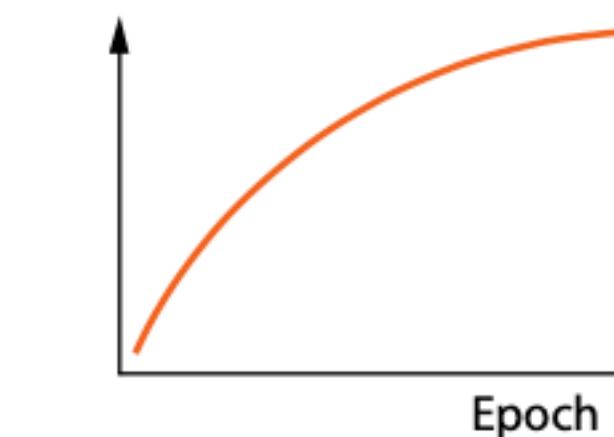
Step 5 - Train the model



Step 3 - Preparing the data



Step 6 - Evaluate



Implementation steps

Step 1 - Import and init



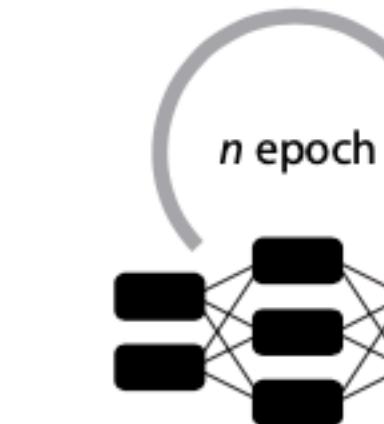
Step 4 - Build a model



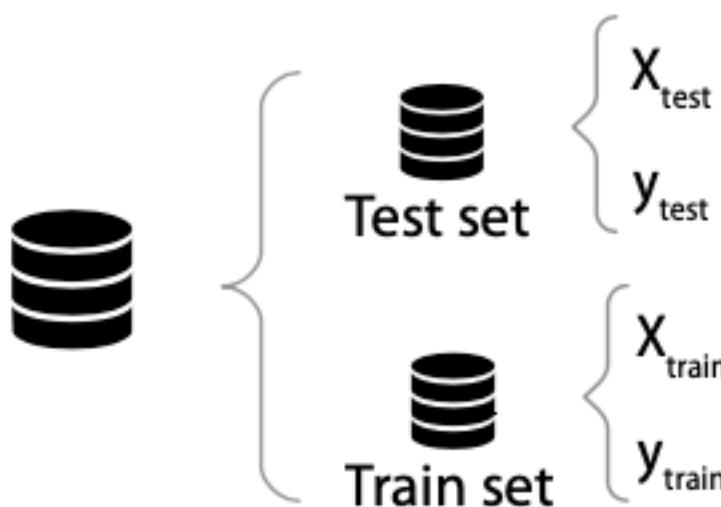
Step 2 - Retrieve data



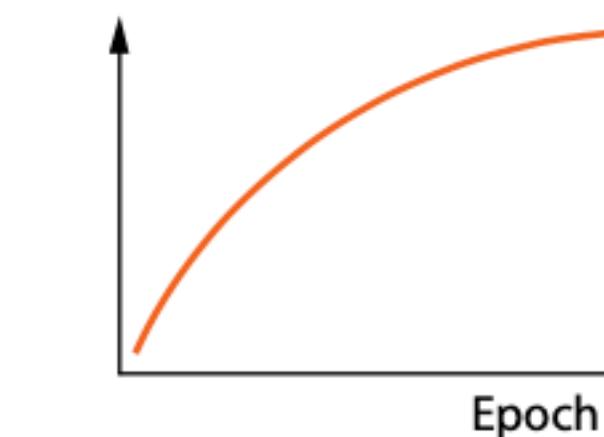
Step 5 - Train the model



Step 3 - Preparing the data



Step 6 - Evaluate



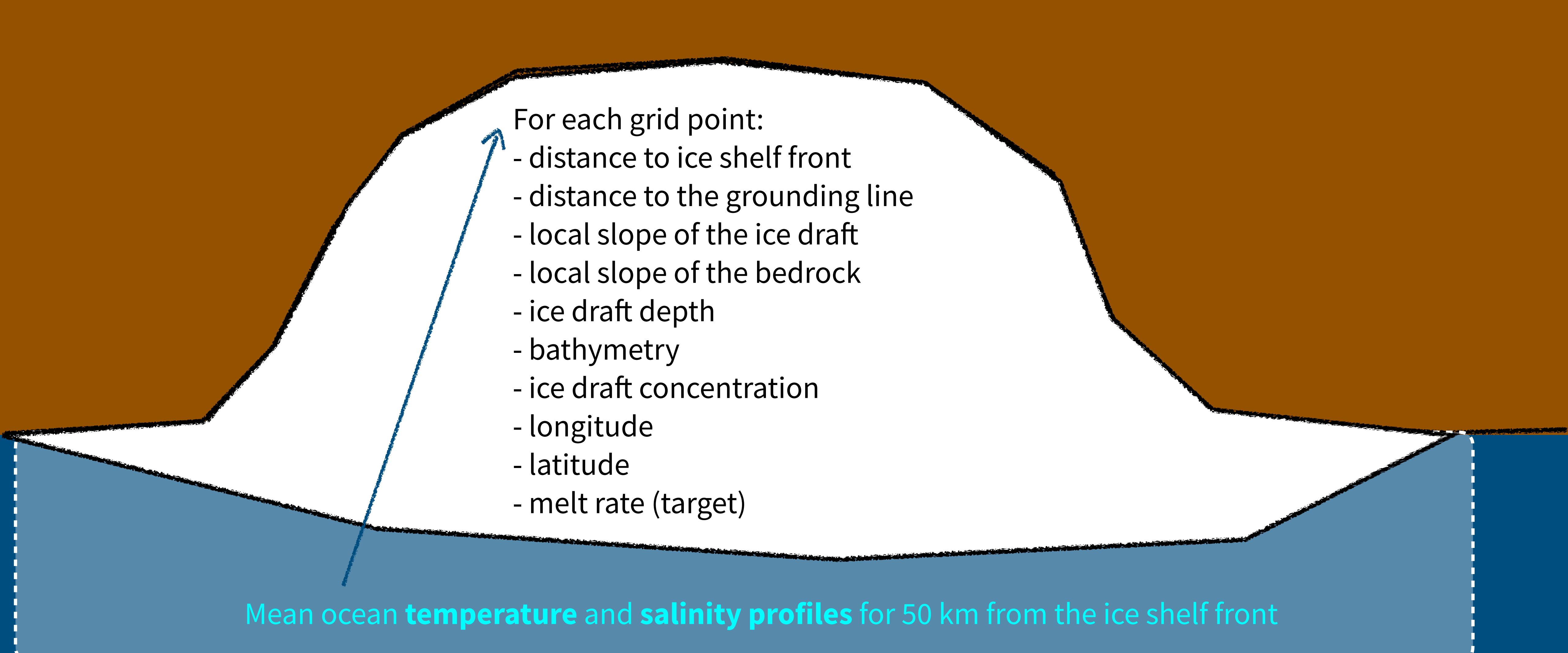
Preparing the data

For each grid point:

- distance to ice shelf front
- distance to the grounding line
- local slope of the ice draft
- local slope of the bedrock
- ice draft depth
- bathymetry
- ice draft concentration
- longitude
- latitude
- melt rate (target)

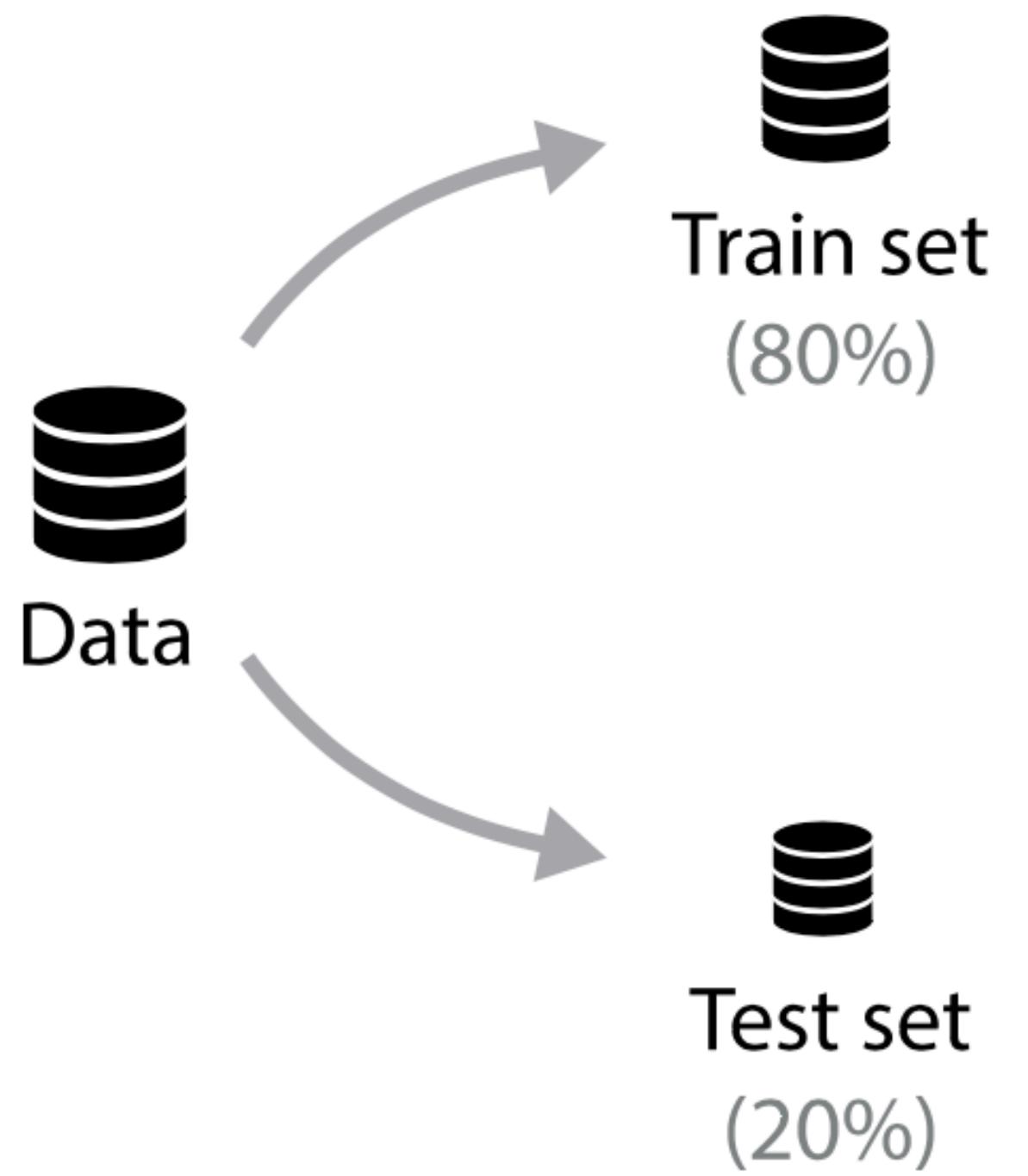
Mean ocean **temperature** and **salinity profiles** for 50 km from the ice shelf front

Preparing the data

- 
- For each grid point:
- distance to ice shelf front
 - distance to the grounding line
 - local slope of the ice draft
 - local slope of the bedrock
 - ice draft depth
 - bathymetry
 - ice draft concentration
 - longitude
 - latitude
 - melt rate (target)

Mean ocean **temperature** and **salinity profiles** for 50 km from the ice shelf front

Preparing the data



Implementation steps

Step 1 - Import and init



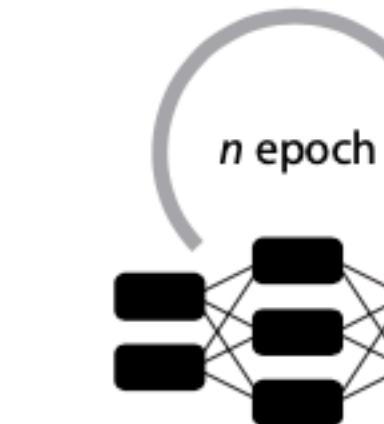
Step 4 - Build a model



Step 2 - Retrieve data



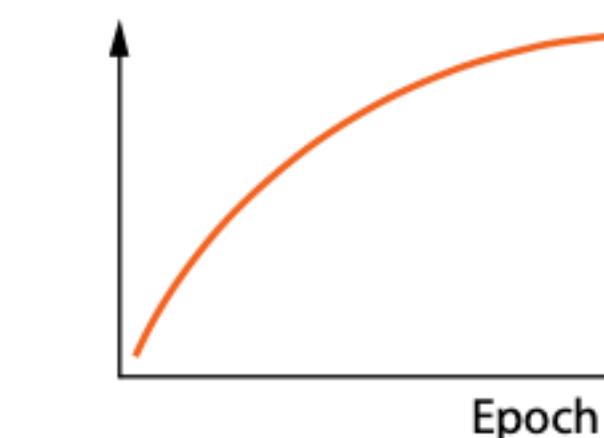
Step 5 - Train the model



Step 3 - Preparing the data



Step 6 - Evaluate



Build the model

```
def get_model_v1(shape):

    model = keras.models.Sequential()
    model.add(keras.layers.Input(shape, name="InputLayer"))
    model.add(keras.layers.Dense(32, activation='relu', name='Dense_n1'))
    model.add(keras.layers.Dense(64, activation='relu', name='Dense_n2'))
    model.add(keras.layers.Dense(32, activation='relu', name='Dense_n3'))
    model.add(keras.layers.Dense(1, name='Output'))

    model.compile(optimizer = 'adam',
                  loss      = 'mse',
                  metrics   = ['mae', 'mse'])

    return model
```

Model: "sequential"

Layer (type)	Output Shape	Param #
<hr/>		
Dense_n1 (Dense)	(None, 32)	5216
Dense_n2 (Dense)	(None, 64)	2112
Dense_n3 (Dense)	(None, 32)	2080
Output (Dense)	(None, 1)	33
<hr/>		
Total params: 9,441		
Trainable params: 9,441		
Non-trainable params: 0		

Implementation steps

Step 1 - Import and init



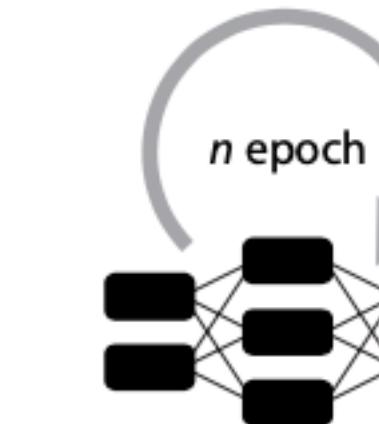
Step 4 - Build a model



Step 2 - Retrieve data



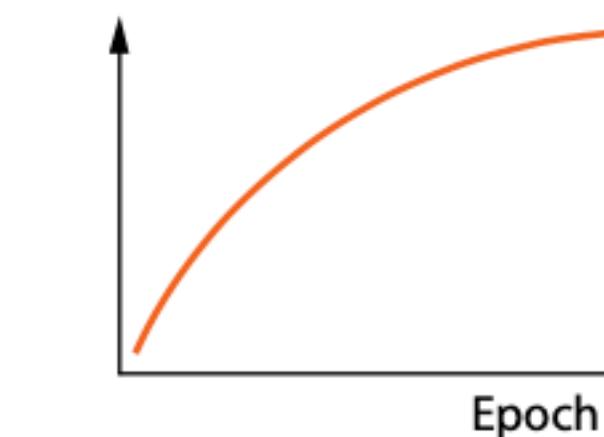
Step 5 - Train the model



Step 3 - Preparing the data



Step 6 - Evaluate



Train the model

```
history = model.fit(x_train_arr,  
                     y_train_arr,  
                     epochs          = 60,  
                     batch_size       = 10,  
                     verbose         = 1,  
                     validation_data = (x_test_arr, y_test_arr))
```

Implementation steps

Step 1 - Import and init



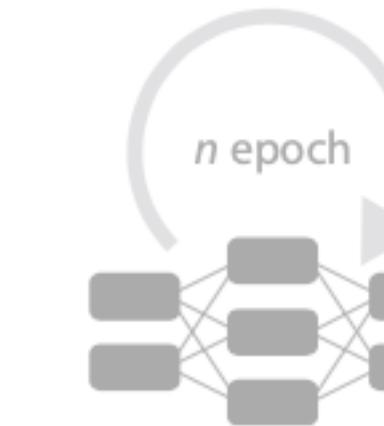
Step 4 - Build a model



Step 2 - Retrieve data



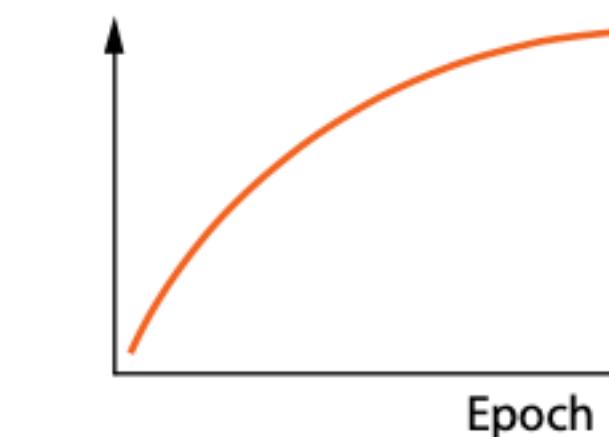
Step 5 - Train the model



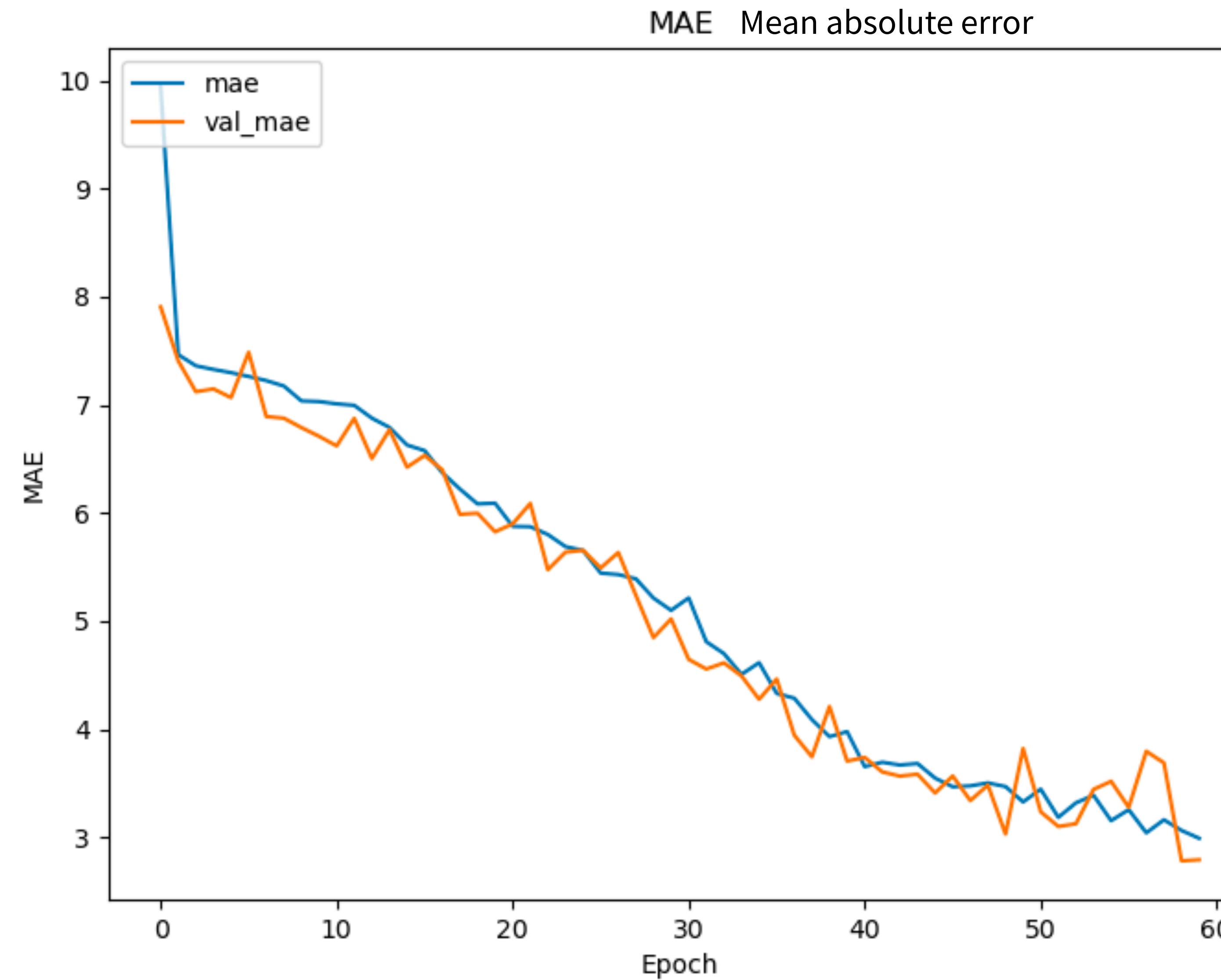
Step 3 - Preparing the data



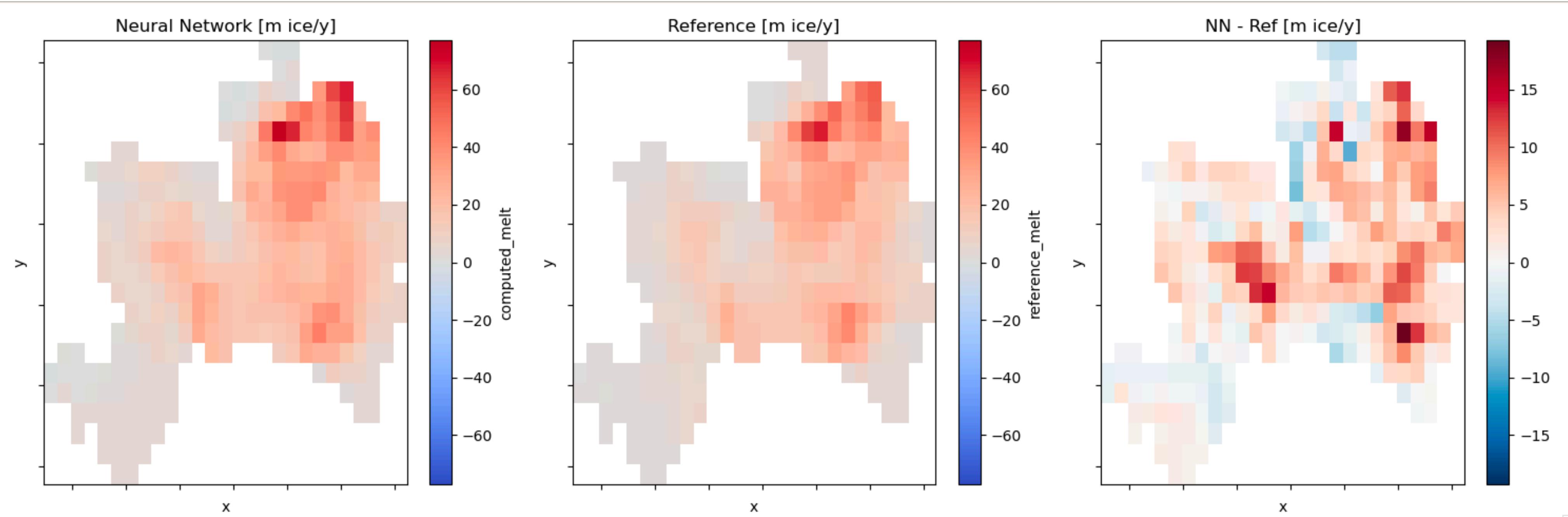
Step 6 - Evaluate



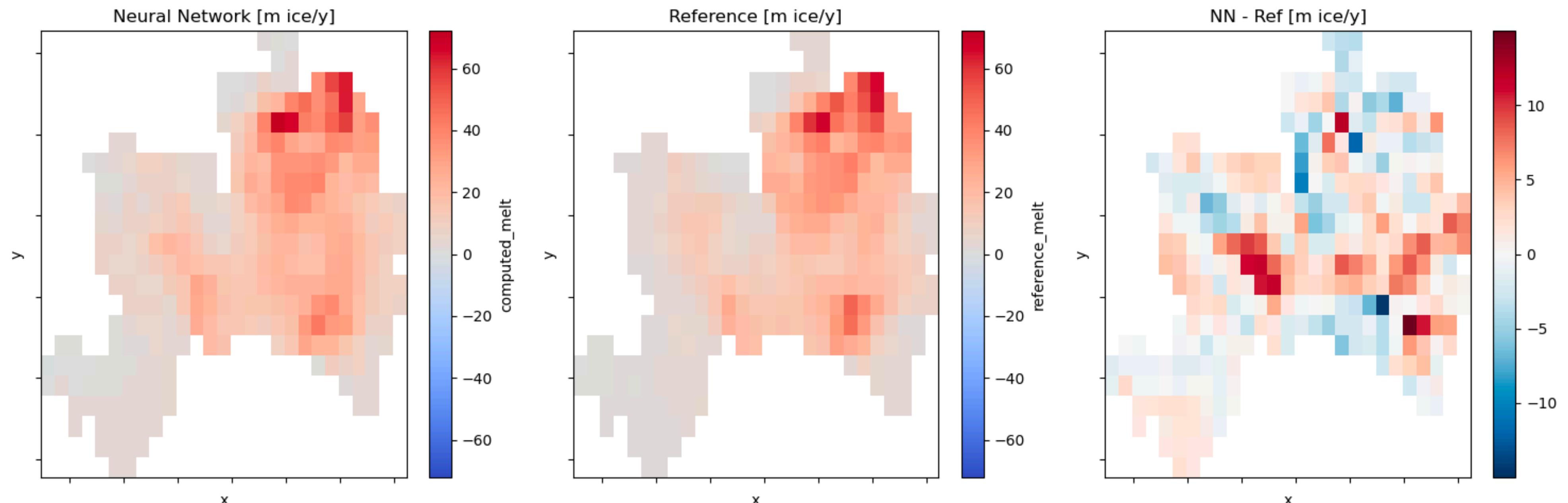
Evaluate the model



Make a prediction



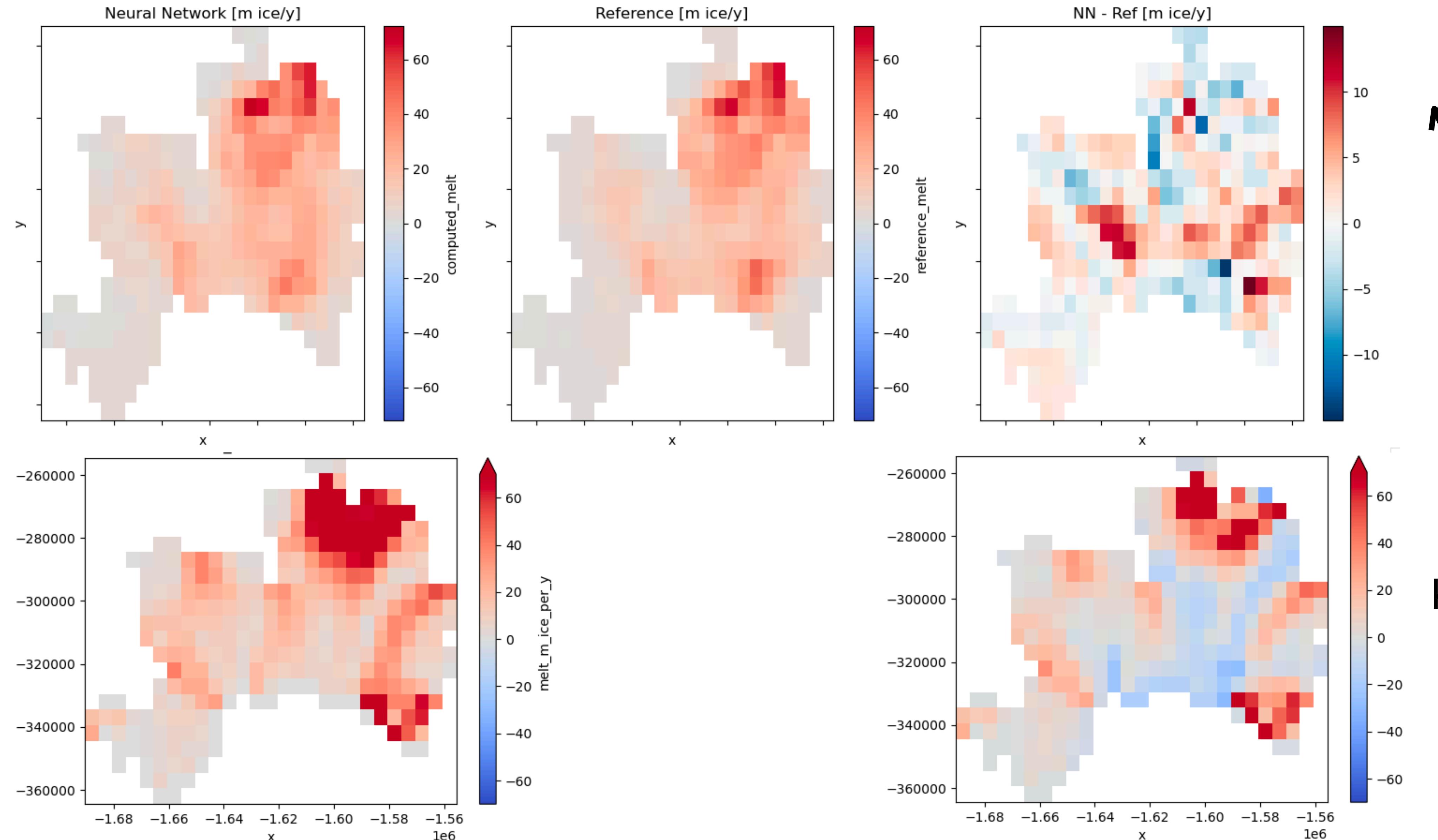
2016



2018

Comparison to existing simple parameterisation

2018



Neural
network

$K \cdot \frac{\text{thermal forcing}}{|\text{thermal forcing}|} \cdot \sin(\text{local slope})$