1 Model development

This section will cover the implementation of the U-Net architecture, as well as related processes such as data preparation and a custom dataloader. Furthermore, this section will present intermediate results obtained during development to highlight technical decisions made as well as their consequence for model performance. Decisions made will be highlighted from both a MachineLearning paradigm point of view, although when relevant they will also be explored in a context of the underlying physics.

1.1 Data preparations

The deep learning system can be disassembled into two parts working in tangent. The deep learning architecture which propagates fields containing information through its weights, and the dataloader which structures the dataset into trainable samples. This subsection will describe the process from raw data to ready sample.

The data pipeline is made such that it constitutes models of three different lead times (one, two and three day lead time). A quick overview of the pipeline is as such. The raw data used are Sea Ice Charts, OSI-SAF and AA. For the Sea Ice Charts, ice charts from the bulletin date and valid date are selected. From AA, relevant meteorological fields are selected and daily means are computed (more details in following sections). Finally, from OSI-SAF a sea ice trend is computed. For a given bulletin date, the data fetched above is stored in a .hdf5 file, such that each sample (bulletin date) is represented by its own .hdf5 file. Furthermore, a dataloader object is initialized with a list of .hdf5 files, with the list containing filenames of the samples constituting a data subset such as train, validation or test data. This processes is visualized in Figure (1).

1.2 Data sources

Data sources used are Sea Ice charts from Nick initiated at 15:00 as well as Arome Arctic initiated at 18:00 Dinessen et al. (2020); Müller et al. (2017). For a given date, the current Ice Chart is used as a predictor for the model, while the Ice Chart drawn two days later is supplied as the model target.

1.2.1 Sea Ice Charts

The Sea Ice Charts used are a derived dataset of the Sea Ice Charts presented in a previous section . The present Ice Chart dataset has been postprocessed by Nick Hughes

label sections

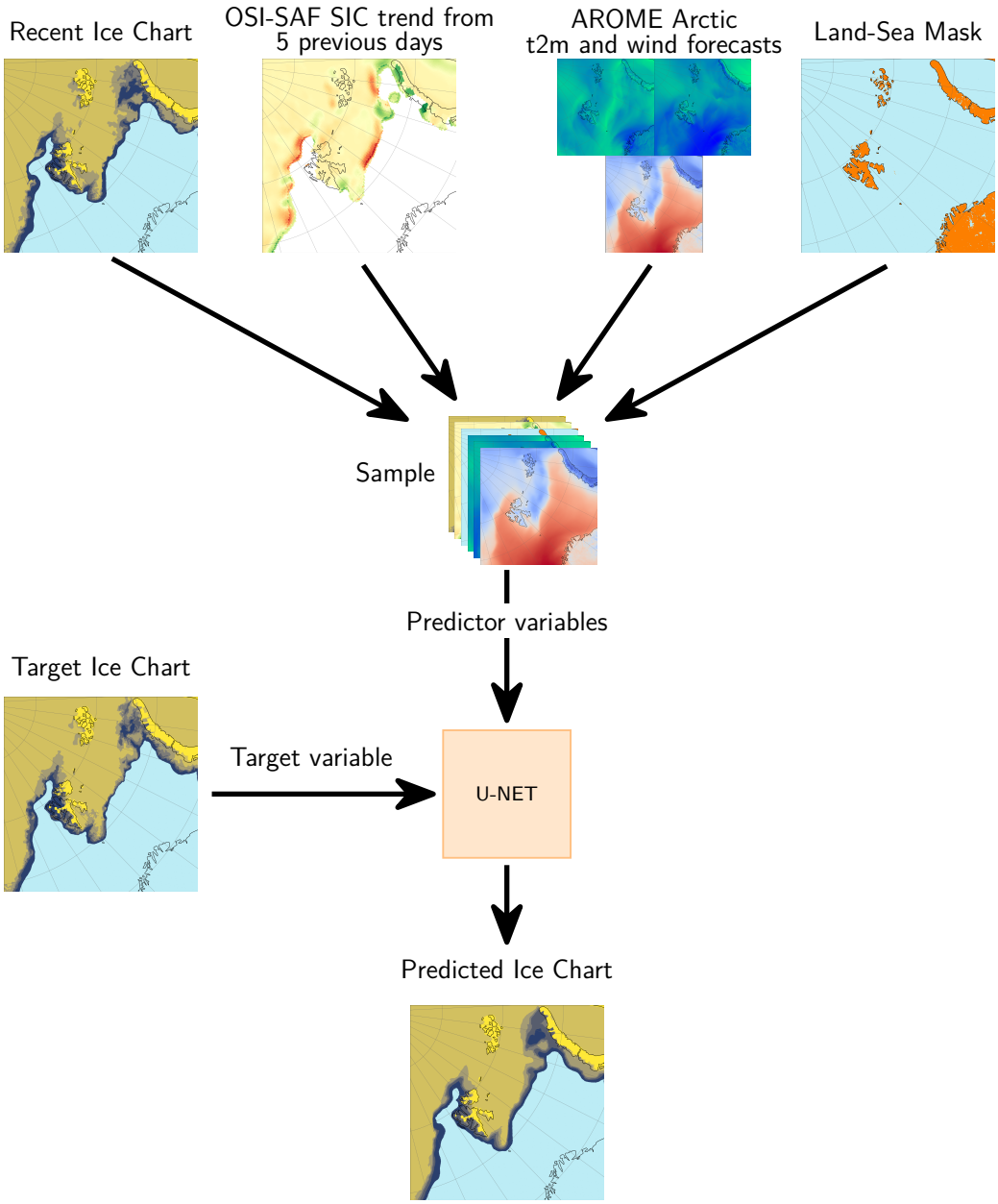


Figure 1: Workflow figure providing a overview of the data pipeline. Data is sampled from four sources (Recent Sea Ice Chart, OSI SAF, AROME Arctic and a Land Sea Mask), preprocessed and merged into a single sample. The sample is fed into the network together with an associated ground truth target sea ice chart. The predicted sea ice chart is compared against the ground truth sea ice chart, and their binary cross entropy error is propagated backwards throughout the network, which constitutes a step in the training loop.

of the National Ice Service , such that they are presented on a 1km Arome Arctic grid. Furthermore, the Ice Charts does not feature a land-mask, which has been replaced with interpolated values resulting in a spatially consistent dataset where all values present are according to the WMO Sea Ice Concentration intervals JCOMM Expert Team on Sea Ice (2014).

Say thanks in acknowledgements

1.3 AROME-Arctic

The Arome Arctic data is structured such that the period between forecast initialization and machine learning forecast lead time is stored as a mean product in the temporal dimension at intervals [0 - 18, 42, 66]. This ensures that temporal AA information is encoded into a single field up until 12:00 UTC of the publishing date of the target ice chart. The 4d variables used from AA are T2M, uwind and vwind. Finally, the land sea mask present in AA is fetched and used as a predictor, though this land sea mask is also used for validation purposes given the case where no other SIC-product is considered.

1.4 OSI-SAF

A linear SIC trend of variable temporal length is computed from 12.5km OSI-SAF data . In the case of OSI-SAF, the product is scheduled to be published daily at 15:00 UTC . However, given operational concerns of the developed forecasting system, where the availability of data is essential for the model to run, the previous day OSI-SAF trend is utilized. .

Mention how when using Osi Saf trend as predictor, the trend up to but not including the forecast start date is used. This is to make the model ready for operational

OSI-SAF SSMIS is a continously developed operational product, where changes are not required to act retroactively on the data. As such, the Sea Ice Concentration used for few samples with t2m runs and many data samples no t2m differ due to the introduction of a filtered ice concentration variable 10/05-2017 Tonboe et al. (2017). Thus, the filtered ice concentration will be used when the training data spans 2019-2020, and the unfiltered ice concentration will be used when the training data spans 2011 - 2018 to assert that there is no sudden shift in the ice concentration trend which can negatively impact the training period.

1.5 Deviations from the U-Net

The model developed for the two day prediction is based on the SimpleUNET architecture, though with a different sized Input layer to accommodate for the changed dataloader. The dataloader has subsequently been changed to appropriately select the correct fields from

the .hdf5 samples and appoint them as input or target variables. As a result of using three variables of two days mean AA forecast, as well as sst, land-sea-mask and current time-step ice chart, the total number of predictors fed into the model is 9. Moreover, the resolution of all fields are kept at 1km, though their spatial extent is limited to (1920 x 1840). This resolution and spatial size conserves (almost) the entirety of the west-east axis of the AA domain. However, the southern border is raised by 450km compared to the AA domain. There are two main motivations behind readjusting the spatial extent of the predictors and targets.

1. The spatial extent of the input domain has to be divisible by the reducing factor enforced by the MaxPooling operation performed in the encoding component of the UNET.
2. The southern latitudes covered by AA has a proportionally skewed Sea Ice / Ice Free open water ratio, as exemplified in Figure (2). Increasing the southern bounding latitude of the subdomain thus decreases the number of guaranteed ice free pixels, which in turn decreases the skewness towards the ice free open water class for the UNET.

2 Model Architecture

The model architecture follows an encoder - decoder structure, commonly referred to as a U-NET Ronneberger et al. (2015) due to its shape funnelling the spatial data to coarser resolution, which resembles the letter "U". The current U-NET implementation follows that of Ronneberger et.al, though it has been modified with batch normalization after each convolution operation to ensure a more stable gradient flow. The weights of the model are Kaiming-He initialized He et al. (2015), as the activation function used throughout the network is the ReLU function Nair and Hinton (2010). The final output of the model is a (1920, 1840, 7) tensor containing softmaxed probabilities along its final axis.

2.1 CategoricalCrossEntropy-Loss

As the title suggests, these runs of the model involved using CategoricalCrossEntropy as the loss function for multi-class image segmentation. Categorical Cross Entropy loss is defined as

$$CE = -\sum_{i}^{C} y_i \log\left(\widehat{y}_i\right) \tag{1}$$

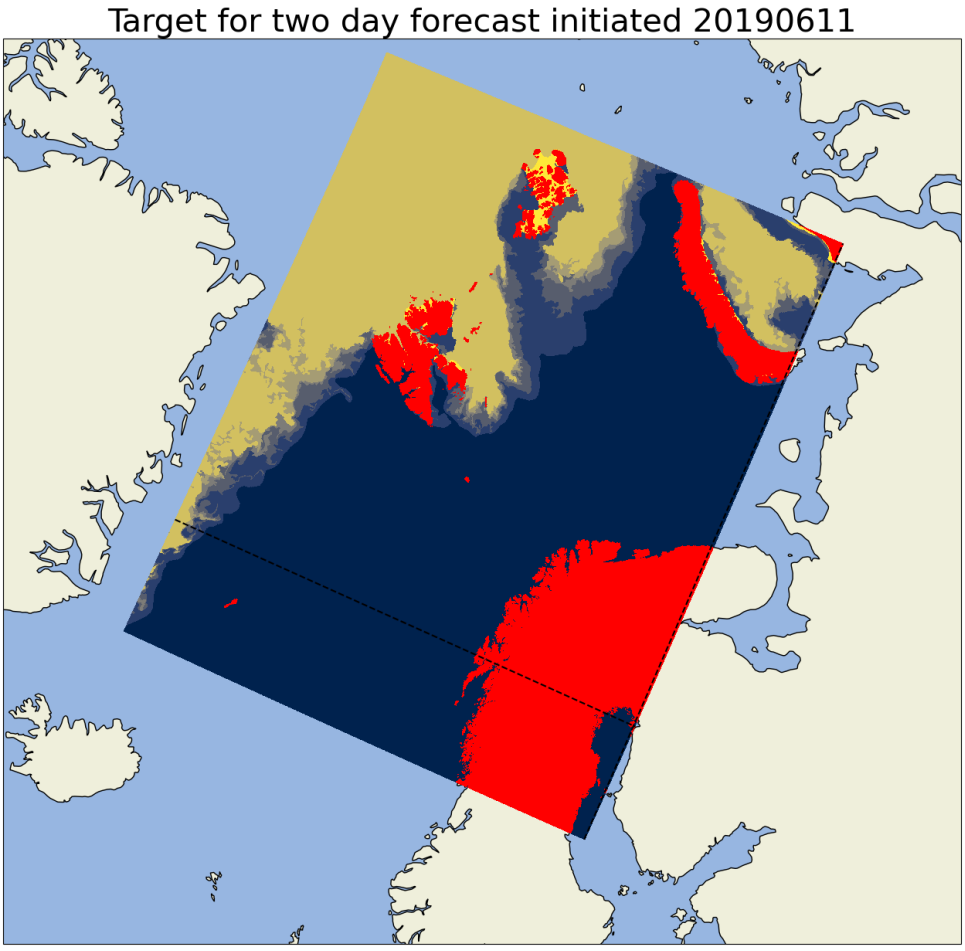


Figure 2: Example sample displaying an Ice Chart on a 1km Arome Arctic projection. Note the horizontal and vertical dashed black line which indicate the domain subsection used by the UNET

where C denotes the number of available classes, y the ground truth and $\hat{y}$ a prediction of y. Note that as y is onehot-encoded, the formulated function only contributes to the overall loss with the log of the predicted probability of the correct class according to the ground truth.

Two variants of the previously described model have been trained with the Categorical-CrossEntropy described in equation (1). The first model was trained with an encoder consisting of 4 convolutional blocks with channel dimensions (64, 128, 256, 512). The second model consisted of 5 convolutional blocks, with an identical architecture except for the last convolutional block increasing the channel dimension to (1024). Example outputs as well as target can be seen in Figure (3).

By inspecting Figure (3), two observations can be made. The first observation is regarding how the model complexity affects how it fit to the data. By comparing Figure (3a) with (3b), it can be seen that the latter is resolving the finer structures of the ice edge to larger extent than the prior. Though the overall correctness is left to be discovered, this shows that increasing the depth of the encoder (increasing the trainable parameter count from 7 million to 31 million) is reflected by the model preserving the details of the ice edge structure. Though it is non-trivial to say why the 1024-model preserves the details to a larger extent than the 512-model, it does follow from the U-Net architecture that a deeper encoder (higher channel count and more convolutional blocks) is better at describing "WHAT" is in the image compared to the shallow-layers, which include a larger amount of spatial information and tells the model to a larger extent "WHERE" things are in the model.

The second observation made from inspecting both forecasts is their inability to represent classes 2 and 3. This likely arises from the general movement-pattern of the sea ice, where the intermediate classes are much less likely to appear than the edge-most classes. Furthermore, the sea ice is much more likely to represent a wider range of concentration classes in the intermediate ice edge region over time, making it more difficult for the network to confidently predict those classes compared to the more probable classes. As can be seen by the network immediately predicting class 4 after class 1, creating an artificial cut-off region. However, to what extent the intermediate classes are predicted has not been inspected directly , though it is likely to assume that they are predicted though with a lower confidence than that of class 4 (which is consequently why it is visualized, as the most probable class is chosen regardless).

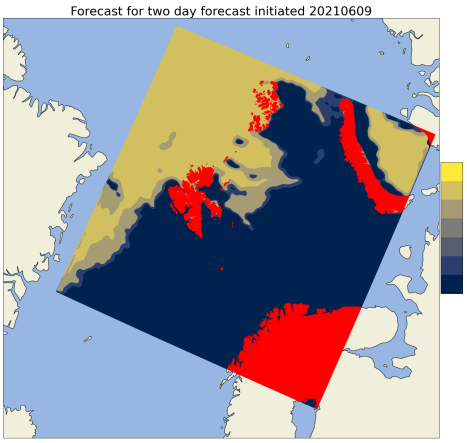
This may have a source

They should be

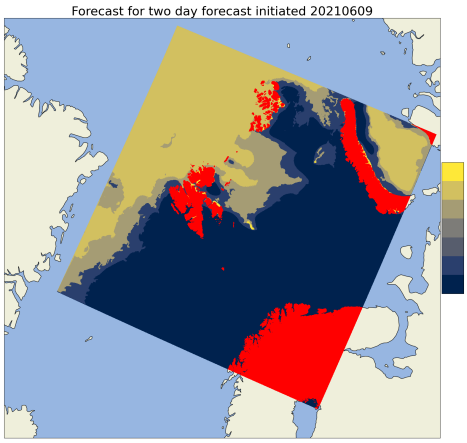
2.2 FocalLoss

The focal loss is derived as a generalization of the Cross Entropy Loss listed in Equation (1). The intent of the loss function is to downweight the easy to predict samples, while

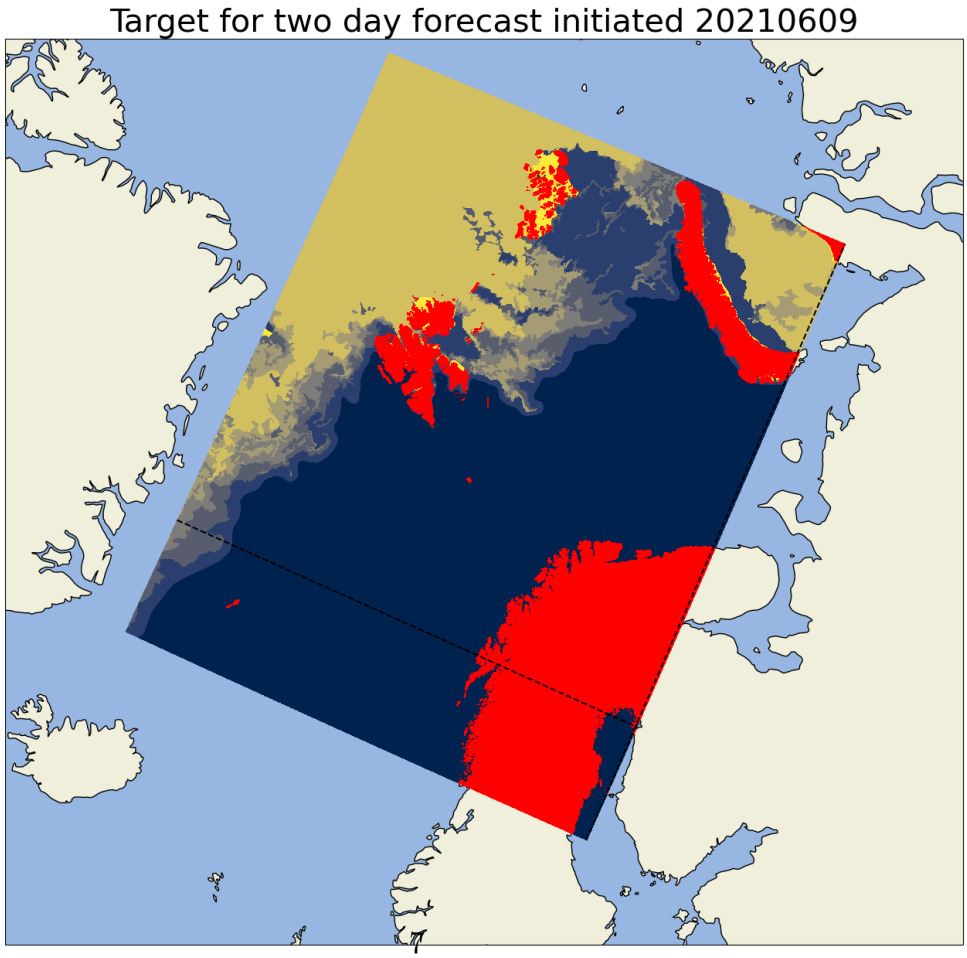
Include figure showing focal loss output, discuss implications



(a) Forecast with two day lead time with model_512 architecture



(b) Forecast with two day lead time with model_1024 architecture



(c) Target for forecast with two day lead time

Figure 3: Example forecast attempt made by model_512 and model_1024 09-06-2021

focusing on the hard to predict samples by allowing their gradient to have a higher impact on the network Lin et al. (2017). Mathematically, focal loss is defined as

$$\mathrm{FL} = -\sum_{i}^{C} \alpha_i (1 - \hat{y}_i)^{\gamma} y_i \log(\hat{y}_i) \tag{2}$$

where α is a balancing parameter, γ is the focusing parameter ($\gamma = 0 \rightarrow CE$), with the rest similar as Equation (1).

By inspecting Equation (2), it can be seen that predictions that the model is quite confident in making, i.e. $\hat{y}_i \rightarrow 1$ send the Focal Loss towards zero. For the current application, the assumptive motivation is that this affects (by reducing) the contribution made by the Ice Free Open Water pixels as well as the Very Close Drift Ice (class 6), which are the most represented classes in the CE loss model seen in Figure (3). Consequently, as the loss contributions of the most likely (and most represented classes) is reduced, the harder to predict (both due to being less represented and due to sea ice movement) have a larger impact on the overall loss propagating backwards throughout the model. As a result, these intermediate classes should be predicted as the most likely class, resulting in a less sharp ice edge which closer represent the Ice Charts.

2.3 Cumulative probability distribution model

2.3.1 Separate convolutionial layers as output

2.4 Model Selection

During the training of a deep learning system, there exists several different ways to save a state of the model during training. A naive approach would be to let the model train all predetermined epochs, and save the weights of the model at the end of the final epoch. However, this approach would be indifferent to whether the model has converged, generalized or overfitted and is thus an inadequate way to save the weights. The Tensorflow Keras API supplies functions which can be used customize the training loop in the form of callbacks, with the EarlyStopping and ModelCheckpoint callbacks relevant for model selection Abadi et al. (2015). EarlyStopping is a technique which ends the training loop

Discuss difference in dataloader, same dataset is used differently

Data exists, start writing

when it detects that a monitored values has stopped decreasing. On the other hand, ModelCheckpoint continuously saves the model if a certain condition is met, without terminating the training loop. Both callbacks support monitoring the validation loss as the metric in which to optimize the model. However, a custom metric such as yearly mean IIEE Goessling et al. (2016) could be monitored instead.

To aid in model selection, I developed a custom callback which computed the Normalized IIEE with respect to a climatological Ice Edge length derived from ten years of OsiSaf data , following the observation in 2 that IIEE is correlated across spatial resolutions. The callback computes said metric for all samples and reduces them to a yearly mean of the validation set. Similar to the aforementioned callbacks, the developed callback is executed at the end of an epoch where it computes the mean Normalized IIEE for all predicted samples from the validation set, which it appends to the *logs* dictionary used by Tensorflow to keep track of other computed metrics, such as loss and validation_loss for the current case. Thus, the newly developed callback would allow for model selection based on Normalized IIEE, as well as the already computed validation loss.

When comparing different models to asses their performance, this project will frequently compare their Normalized IIEE as the metric is Normalized by the ice edge, thus reducing the seasonal variability of the Metric Palerme et al. (2019) . As such, it would be beneficial to select a model based on its Normalized IIEE validation performance. With the above callback, such a selection is possible. However, including the IIEE verification metric as is done in the above callback increases training-time of ten epochs from ≈ two hours without the IIEE callback to ≈ 24 hours with the IIEE callback. As 20 epochs is currently an adequate number of epochs at the time of writing , it would be too computationally costly to select a model based in its validation Normalized IIEE performance.

On the other hand, it can be seen by inspecting Figure (4) that the Normalized IIEE tend to evolve conjunctionally with the validation loss, in the current case defined as the mean cross entropy of all validation samples. Furthermore, the validation loss and Normalized IIEE in Figure (4) have a correlation of 0.82 with regards to epoch. Note that this has been calculated only using the numbers present in Figure (4). As such, there is reason to believe that selecting a model based on its validation loss, which is quick to compute, would result in a generalized model which may also excel at lowering its Normalized IIEE.

When selecting the best model, this project will apply the ModelCheckpoint callback with regards to validation loss as outlined above. ModelCheckpoint is preferred compared to EarlyStopping, as interrupting the training loop early may result in an ”undercooked” model. E.g. the weights in earlier model layers are adjusting slower than later weights, giving the impression that the model training has reached a plateau which causes the model to stop. Whereas if the model where to continue training, the later adjustment of

Write about the climatological Ice Edge dataset, ref section from here

This citation is actually for SPS_{length} but SPS is reduced to IIEE for a deterministic ice edge Goessling and Jung (2018)

This may change

tmp figure, redo with

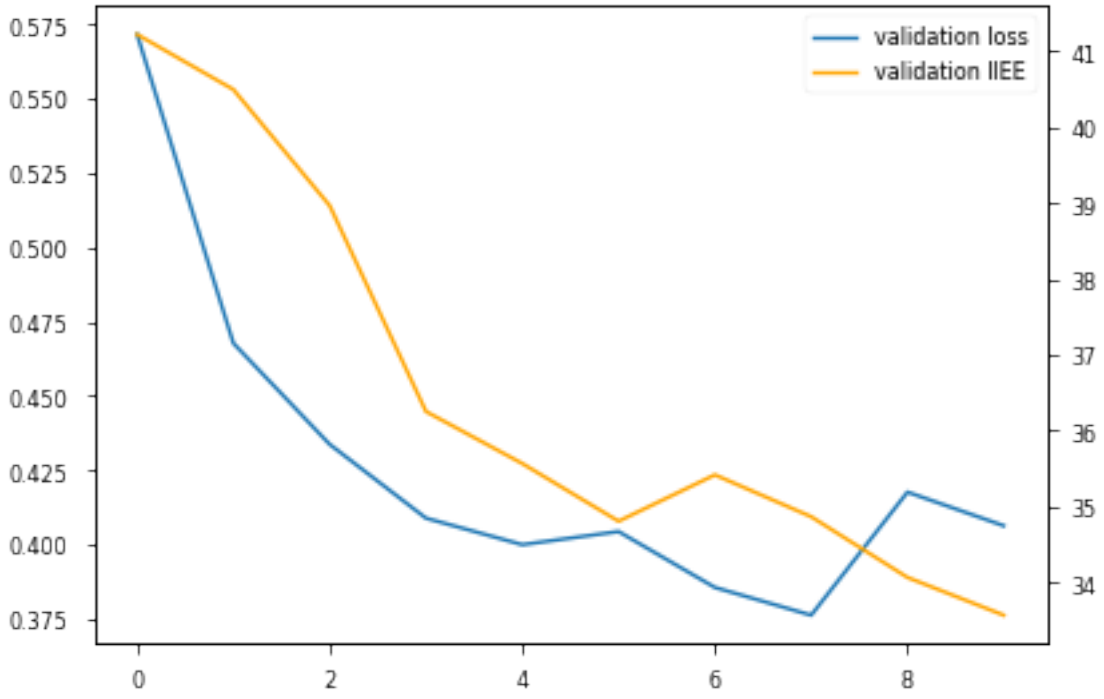


Figure 4: validation loss and Normalized IIEE computed as mean of validation set for each epoch during training

earlier weights would cause a later spur in increased model performance. ModelCheckpoint was chosen since behavior such as what was just exemplified is possible with the callback.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, URL https://www.tensorflow.org/, software available from tensorflow.org, 2015.

Dinessen, F., Hackett, B., and Kreiner, M. B.: Product User Manual For Regional High Resolution Sea Ice Charts Svalbard and Greenland Region, Tech. rep., Norwegian Meteorological Institute, 2020.

Goessling, H. F. and Jung, T.: A probabilistic verification score for contours: Methodology and application to Arctic ice-edge forecasts, Quarterly Journal of the Royal Meteorological Society, 144, 735–743, https://doi.org/10.1002/qj.3242, 2018.

Goessling, H. F., Tietsche, S., Day, J. J., Hawkins, E., and Jung, T.: Predictability of the Arctic sea ice edge, Geophysical Research Letters, 43, 1642–1650, https://doi.org/10.1002/2015gl067232, 2016.

He, K., Zhang, X., Ren, S., and Sun, J.: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, https://doi.org/10.48550/ARXIV.1502.01852, 2015.

JCOMM Expert Team on Sea Ice: Sea-Ice Nomenclature: snapshot of the WMO Sea Ice Nomenclature WMO No. 259, volume 1 – Terminology and Codes; Volume II – Illustrated Glossary and III – International System of Sea-Ice Symbols) ., https://doi.org/10.25607/OBP-1515, 2014.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P.: Focal Loss for Dense Object Detection, 2017.

Müller, M., Batrak, Y., Kristiansen, J., Køltzow, M. A. Ø., Noer, G., and Korosov, A.: Characteristics of a Convective-Scale Weather Forecasting System for the European Arctic, Monthly Weather Review, 145, 4771–4787, https://doi.org/10.1175/mwr-d-17-0194.1, 2017.

Nair, V. and Hinton, G.: Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair, vol. 27, pp. 807–814, 2010.

Palerme, C., Müller, M., and Melsom, A.: An Intercomparison of Verification Scores

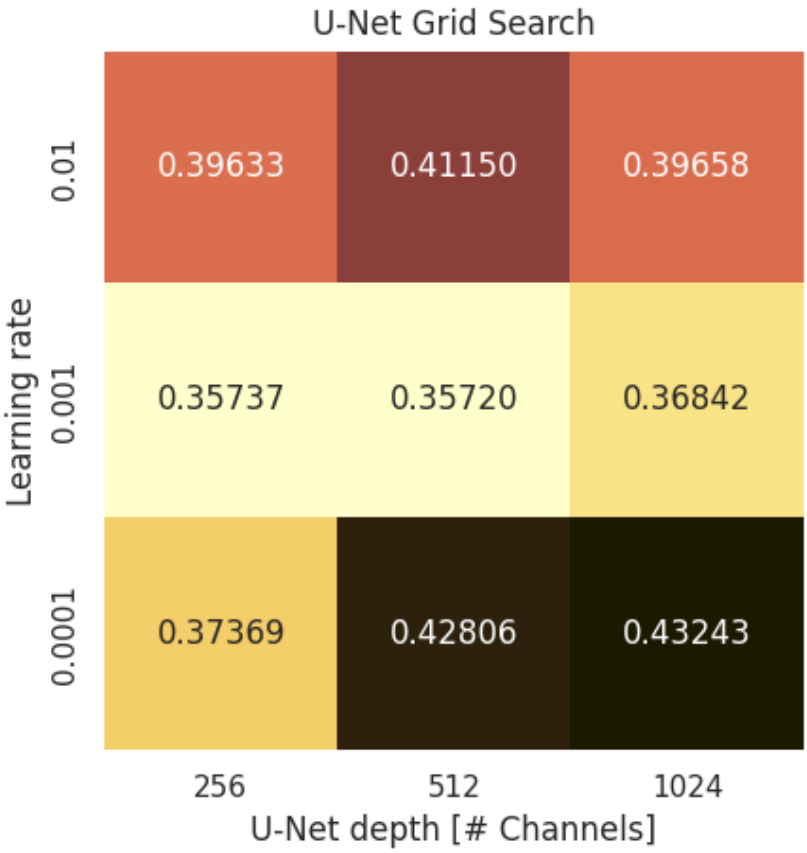


Figure 5: Grid search performed over variations of the learning rate as well as an increasing U-Net depth (represented by the number of feature maps at the at the final convolutional block). Each cell contain the minimum obtained validation loss of its respective combination which is associated with the best validation performance during training.

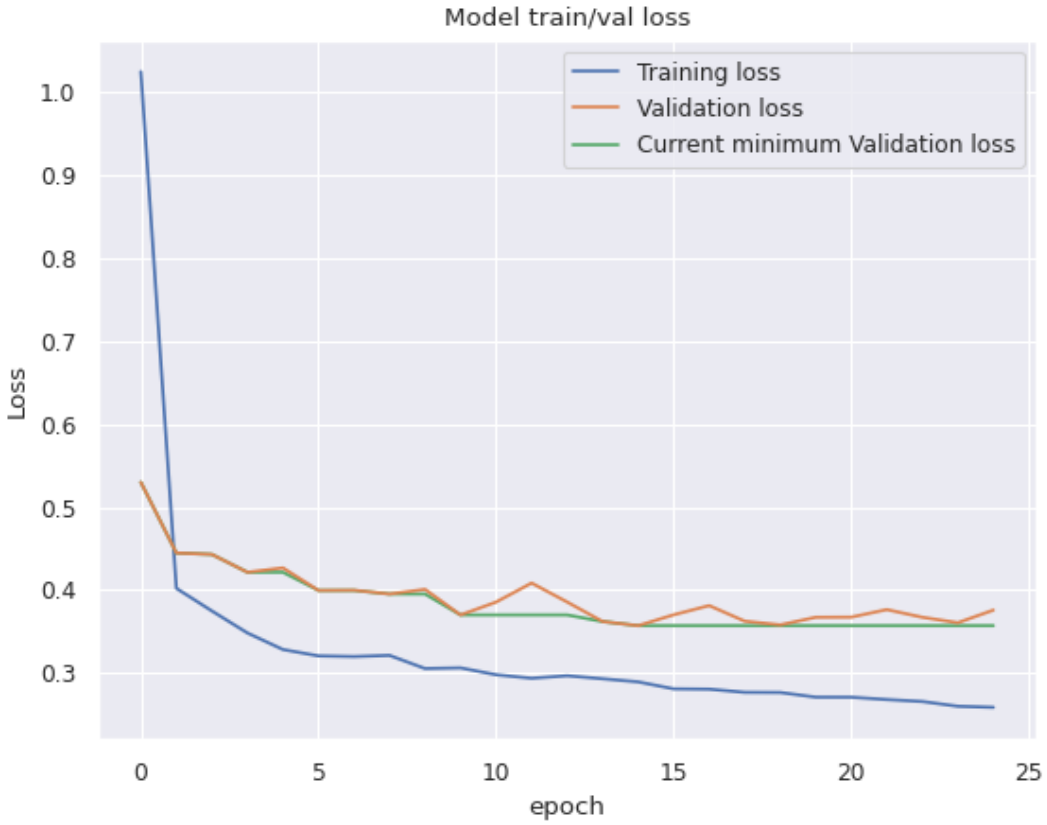


Figure 6: Training and validation loss from the model attaining lowest validation loss in Figure (5). The current minimum validation loss is also displayed.

for Evaluating the Sea Ice Edge Position in Seasonal Forecasts, Geophysical Research Letters, 46, 4757–4763, https://doi.org/10.1029/2019gl082482, 2019.

Ronneberger, O., Fischer, P., and Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation, in: Lecture Notes in Computer Science, pp. 234–241, Springer International Publishing, https://doi.org/10.1007/978-3-319-24574-4_28, 2015.

Tonboe, R., Lavelle, J., Pfeiffer, R.-H., and Howe, E.: Product User Manual for OSI SAF Global Sea Ice Concentration, Tech. Rep. 1.6, Danish Meteorological Institute, 2017.