

A Variational U-Net for Weather Forecasting

Pak Hay Kwok¹, Qi Qi²

¹pak_hay_kwok@hotmail.com

²qiq208@gmail.com

Abstract

Not only can discovering patterns and insights from atmospheric data enable more accurate weather predictions, but it may also provide valuable information to help tackle climate change. Weather4cast is an open competition that aims to evaluate machine learning algorithms' capability to predict future atmospheric states. Here, we describe our third-place solution to Weather4cast. We present a novel Variational U-Net that combines a Variational Autoencoder's ability to consider the probabilistic nature of data with a U-Net's ability to recover fine-grained details. This solution is an evolution from our fourth-place solution to Traffic4cast 2020 with many commonalities, suggesting its applicability to vastly different domains, such as weather and traffic.

The code for this solution is available at https://github.com/qiq208/weather4cast2021_Stage1

Keywords

IARAI, Traffic4cast, Weather4cast, U-Net, Variational Autoencoder

1. Introduction

Meteorological satellites around the globe are constantly gathering a trove of data about the atmosphere. However, the high-dimensionality nature of atmospheric data makes it challenging to analyse, hindering the discovery of valuable insights. With the advent of machine learning methods, it is believed these methods can help better understand atmospheric data. To evaluate the applicability of such techniques to atmospheric data, Weather4cast [1] by the Institute of Advanced Research in Artificial Intelligence is an open competition that challenges its participants to develop algorithms to predict the future states of the atmosphere over specific regions.

The Weather4cast dataset [2] is obtained from Meteosat geostationary meteorological satellites operated by EUMETSAT for the period from February 2019 to February 2021. The Meteosat images are processed by NWC SAF software into weather products. The weather products of interest are: Cloud Top Temperature and Height (CTTH), Convective Rainfall Rate (CRR), Automatic Satellite Image Interpretation - Tropopause Folding detection (ASII-TF), Cloud Mask (CMA), and Cloud Type (CT). Each of these weather products is recorded in 15-minute intervals and consists of multiple channels. Each channel is in the format of an image of shape 256x256 pixels, with each pixel covering an area of about 4x4 km. The regions of interest are illustrated in Figure 1; regions

R1-3 correspond to the core challenge in which training, validation and test data are provided, while regions R4-6 correspond to the transfer learning challenge in which only the test data are provided. In addition, static information, such as altitude, latitude and longitude, are also given for all regions.

Weather4cast demands an algorithm that can return the atmospheric states over the defined regions for the next 8 hours (32-off 15-minute intervals) given an hour (4-off 15-minute intervals) worth of data. While only 4 target variables are required, namely *temperature* (a channel of CTTH), *crr_intensity* (a channel of CRR), *asii_turb_trop_prob* (a channel of ASII-TF) and *cma* (a channel of CMA), any channels of the weather products or static information of the regions can be used as input variables.

This work describes a novel Variational U-Net solution which achieved third place in both the core and transfer learning challenges of Weather4cast. This Variational U-Net can be viewed as a U-Net with a Variational Autoencoder (VAE) style bottleneck, or as a VAE with U-Net style skip connections. The intuition behind this architecture is to combined VAE's ability to consider the probabilistic nature of data with U-Net's ability to recover fine-grained details.

2. Related work

Weather4cast can be viewed as a video frame prediction problem, in which the inputs are the first 4 frames of a video, and the outputs are the subsequent 32 frames. This format of the problem is identical to that of Traffic4cast [3, 4]. Overlooking the difference in domains between Weather4cast and Traffic4cast, the two competitions can be considered the same, hence solutions for Traffic4cast


CDCEO 2021: 1st workshop on Complex Data Challenges in Earth Observation, November 1, 2021, Virtual

✉ pak_hay_kwok@hotmail.com (P. H. Kwok); qiq208@gmail.com (Q. Qi)

🌐 <https://github.com/ivans-github/> (P. H. Kwok);

<https://github.com/qiq208/> (Q. Qi)

© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

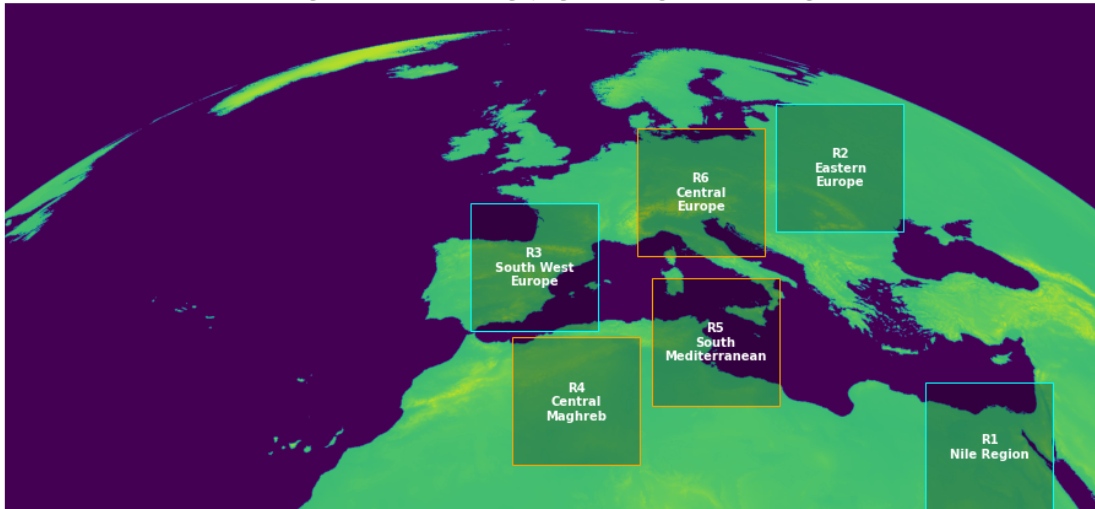


Figure 1: Weather4cast regions

should be somewhat transferable to Weather4cast. A range of algorithms, including U-Net, LSTM and Graph Neural Network were proposed for Traffic4cast [5, 6], yet various flavours of U-Net dominated the competition in both 2019 and 2020, with all winning teams adopting U-Nets in their final solutions [5, 7]. Thus, it is sensible to consider U-Net-based solutions for Weather4cast.

While the formats of Weather4cast and Traffic4cast are equivalent, the differences in the underlying domains cannot be ignored. Specifically, weather is considered more random than traffic. Multiple scenarios are possible given a set of observations, and this inherent randomness needs particular attention, as it is not compatible with the deterministic nature of a typical U-Net. Segmentation of medical images also suffers from intrinsic ambiguities. To handle these ambiguities, Kohl et al. [8] proposed a Probabilistic U-Net, a combination of a U-Net with a conditional VAE, capable of producing an unlimited number of hypotheses from a set of inputs. Myronenko [9] also proposed a different way to combine a U-Net with a VAE, which a VAE was applied to regularise a shared encoder. His solution was proven successful and won first place in the Multimodal Brain Tumour Segmentation Challenge (BraTS) in 2018.

3. Methods

3.1. Model architecture

Given the similarities between Weather4cast and Traffic4cast, the main structure of the proposed Variational U-Net largely resembles the authors' fourth-place solu-

tion to last year's Traffic4cast [5]. The encoder is made up of Dense Blocks connected by 2D Max Pooling. Each Dense Block consists of 4 repeats of 2D Convolution, ELU [10], Group Normalisation [11] and 2D Dropout [12], followed by another 2D Convolution and ELU. Different to the encoder, the decoder consists of repeats of 2D Transposed Convolution, ELU, 2D Convolution, ELU, Group Normalisation and 2D Dropout. The encoder and the decoder are joined by skip connections.

Inspired by the works of Kohl et al. [8] and Myronenko [9], the bottleneck of the Variational U-Net, the part which connects the end of the encoder to the start of the decoder, is replaced with one that is typically found in VAE. At the end of the encoder, the input is reduced to 2 vectors of size 512, representing the means and standard deviations of the latent variables. With the assumption that the latent variables are Gaussian, a sample is drawn, and the drawn vector is reconstructed into an image which is then passed through the decoder.

The architecture of the Variational U-Net is shown in Figure 2.

3.2. Inputs and target variables

Similar to the authors' Traffic4cast solution [5], the temporal dimension of the input tensor is combined with the channel dimension, resulting in the number of input channels of 4×8 . Furthermore, since it seems intuitive that weather patterns are dependent on geographical location, the static features of altitude, latitude and longitude are appended, resulting in an additional 3 input channels. As such, the final number of input channels to the Varia-

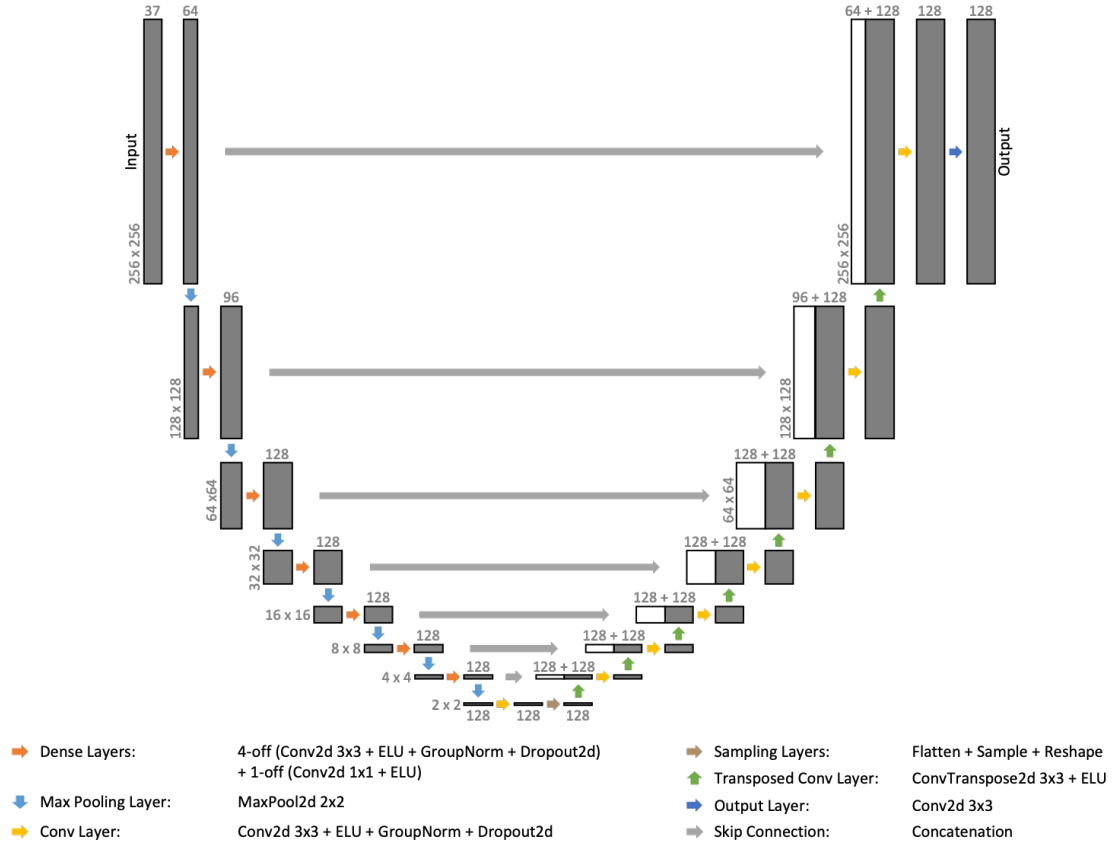


Figure 2: Variational U-Net architecture

Table 1
Summary of input features and target variables

Feature	Target Variable	Description
<i>temperature</i>	Yes	Combined cloud top and ground temperature
<i>ctth_pres</i>	No	Cloud top pressure
<i>crr_intensity</i>	Yes	Convective rainfall rate intensity in mm/h
<i>crr_accum</i>	No	Convective rainfall rate hourly accumulations
<i>asii_turb_trop_prob</i>	Yes	Probability of occurrence of tropopause folding
<i>cma</i>	Yes	Cloud mask
<i>ct</i>	No	Cloud type
<i>ctth_tempe</i> mask	No	A mask showing pixel locations containing cloud top temperature measurements

tional U-Net is $4 \times 8 + 3 = 35$. Finally, the model is designed to predict all 32 output frames in one go, resulting in the number of output channels being $32 \times 4 = 128$. Furthermore, any missing data has been zero-filled.

A series of experiments were performed to find the most effective set of input features, and the validation set was used to evaluate the performance of each feature set. The resulting input feature set is listed in Table 1, and

those rejected are summarised in Table 2.

3.3. Loss function

The loss function consists of 2 terms:

$$L = L_{L2} + 80 * L_{KL} \quad (1)$$

L_{L2} is a modified mean squared error, it takes into account missing values and the difference in scale of the

Table 2
Summary of input features not used in the final model

Feature	Description
<i>ctth_alt</i>	Cloud top altitude
Linear interpolation of <i>temperature</i>	Using linear interpolation to fill in missing <i>temperature</i>
Linear interpolation of <i>ctth_pres</i>	Using linear interpolation to fill in missing <i>ctth_pres</i>

4 target variables:

$$L_{L2} = \frac{1}{32 \times 4} \sum_{t=1}^{32} \sum_{v \in V} \frac{w_v}{P_{t,v}} \sum_{p=1}^{P_{t,v}} (y_{t,v,p} - \hat{y}_{t,v,p})^2 \quad (2)$$

where $V = \{temperature, err_intensity, cma, asii_turb_trop_prob\}$, $P_{t,v}$ is the total number of non-missing pixels for a given target variable v at a given time t and w_v is the target variable weighting:

$$w_v = \begin{cases} 31.610, & v = temperature \\ 4139.4, & v = err_intensity \\ 5.2191, & v = cma \\ 142.17, & v = asii_turb_trop_prob \end{cases}$$

L_{KL} is the KL divergence between the estimated Gaussian distribution $N(\mu, \sigma^2)$ and a prior distribution $N(0, 1)$:

$$L_{KL} = \frac{1}{2} \sum_{i=1}^{512} \mu_i^2 + \sigma_i^2 - \log \sigma_i - 1 \quad (3)$$

The L_{KL} factor of 80 in Equation 1 was determined empirically to balance the relative importance of the two terms in the loss function.

3.4. Optimisation

The Variational U-Net is trained using the Adam optimiser with Cyclic Cosine Annealing described by Loshchilov et al. [13]. The training process is split into cycles, with each cycle consisting of 2 epochs. At each cycle, the learning rate is first set to a maximum of $2e-4$, then is reduced following a cosine annealing schedule. Resetting the learning rate at the beginning of each cycle perturbs the models and encourages them to explore different basins of attraction. The training is continued until an additional cycle failed to return a better validation score.

Using a batch size of 12, the final model was first trained for 6 cycles (12 epochs) on the training data, then it was further trained for an additional cycle (2 epochs) on both the training and validation data.

3.5. Regularisation

From initial experiments, it became apparent that controlling overfitting of the model to the training data was a key to success in both the core and transfer learning challenges. Hence, several regularisation strategies were employed. Within the model itself, the move to the Variational U-Net from a traditional U-Net, combined with the introduction of dropout layers throughout the encoder and decoder, both aimed to improve the generalisation of the model. To expose the model to as much variation in input as possible, a single model was used for all regions in the competition and trained on all available training data. Furthermore, for the final leaderboard submission, the model was further trained for another cycle on all the validation data available.

4. Results

The majority of experimentation on the design of features and model architecture was conducted on single regions to allow for quicker feedback and learning. However, the final model was trained on data from all regions, so there is a risk that some of the decisions made might not be optimum for a model trained on data from all regions. Results from the main experiments can be found in Appendix A.

Final experiments on all three regions were conducted, and models were evaluated based on either the test leaderboard or the final leaderboard. It is worth noting that the test leaderboard allowed multiple submissions and was open up to the final week of the competition. In the final week, the final leaderboard was opened and competitors were only allowed three submissions. The results of the submissions can be found in Table 3.

The competition is based on the final leaderboard scores and the final model resulted in a third-place finish for both the core and transfer learning challenges. The training history of the final model is shown in Figure 3, highlighting the loss progression during both the normal training phase, as well as the additional cycle training on the validation data.

Table 3
Summary of leaderboard scores for final models

Model	Core Challenge			Transfer Learning Challenge	
	Validation	Test Leaderboard	Final Leaderboard	Test Leaderboard	Final Leaderboard
Mean baseline	-	0.8822	-	-	-
IARAI U-Net baseline [2]	-	0.6689	-	0.6111	-
One model per region	-	0.5095	-	-	-
Single model	0.3912	0.4977	0.5140	0.4878	0.4711
Single model + linear interpolation of <i>temperature</i>	0.3887	-	0.5218	-	-
Single model + training on validation data	-	-	0.5102	-	0.4670

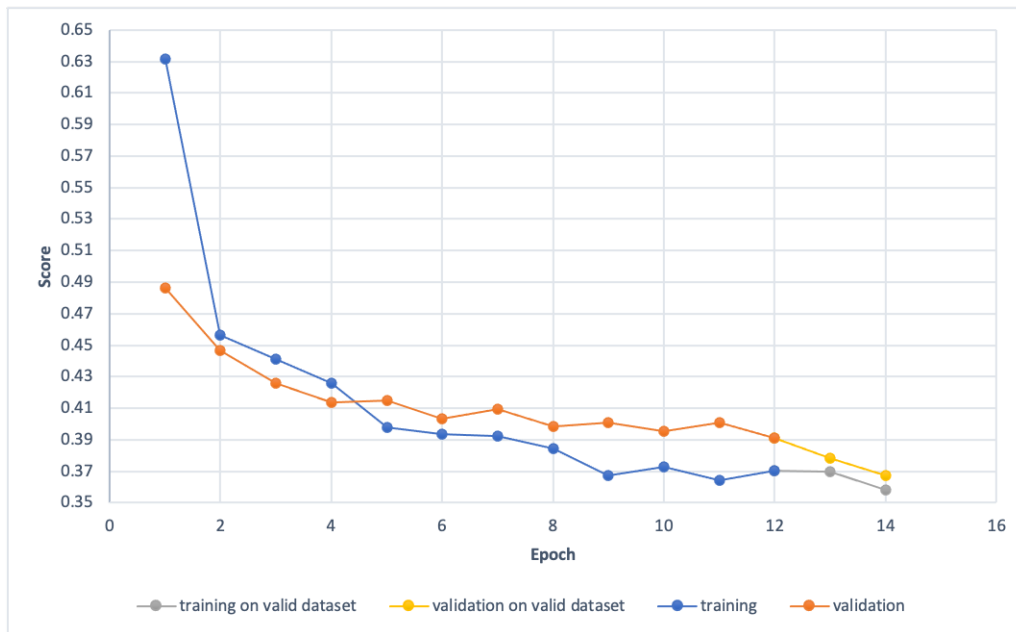


Figure 3: Training history of the final model

5. Discussion

Although various U-Net architectures were explored, it was interesting to observe that the final architecture was very similar to the architecture used for Traffic4cast [5]. The only changes were moving to max pooling from average pooling, the addition of dropout layers and the adoption of the VAE style bottleneck. The authors would be interested in exploring whether these improvements would also read back across to the traffic prediction task.

In terms of feature engineering, the experiments showed that the inclusion of some extra features (e.g. cloud top pressure) improved predictive capability,

whereas others (e.g. cloud top altitude) did not. It was found that linearly interpolating temperature provided an improvement to the validation score, however, this did not read across to the final leaderboard score. The authors still believe that strategies to compute missing data is an interesting area for further work.

Perhaps most surprisingly was the benefit gained from training a single model on data from all regions instead of individual models for each region. The model trained on all regions displayed a significant improvement in the test leaderboard score (~2.3%) over individually trained models. This finding suggests that that the model may continue to improve its general predictive ability for any

region with the addition of more training data. This hypothesis was further supported as training on the validation data further improved the final leaderboard score for both core and transfer learning challenges.

6. Conclusion

Weather4cast provided the opportunity to explore the use of machine learning techniques to the age-old problem of weather forecasting. Furthermore, the similarity of format to Traffic4cast also provided the chance to investigate how transferable machine learning models can be across vastly different domains. After experimenting with various U-Net architectures, the final model was very similar to the authors' Traffic4cast model. The main differences being changes to suppress overfitting, i.e. moving to the Variational U-Net model and inclusion of dropout layers throughout. The authors also found that training on data from all regions in one model outperformed training individual models on each region for both the core and transfer learning challenges. This suggests that the model prediction for all regions can be improved by training on more data.

References

- [1] IARAI, Weather4cast: Multi-sensor weather forecast competition, 2021. URL: <https://www.iarai.ac.at/weather4cast>.
- [2] IARAI, Weather4cast: Multi-sensor weather forecasting competition & benchmark dataset, 2021. URL: <https://github.com/iarai/weather4cast>.
- [3] D. Kreil, M. Kopp, D. Jonietz, M. Neun, A. Gruca, P. Herruzo, H. Martin, A. Soleymani, S. Hochreiter, The surprising efficiency of framing geo-spatial time series forecasting as a video prediction task—insights from the iarai traffic4cast competition at neurips 2019, in: NeurIPS 2019 Competition and Demonstration Track, PMLR, 2020, pp. 232–241.
- [4] M. Kopp, D. Kreil, M. Neun, D. Jonietz, H. Martin, P. Herruzo, A. Gruca, A. Soleymani, F. Wu, Y. Liu, et al., Traffic4cast at neurips 2020 - yet more on the unreasonable effectiveness of gridded geo-spatial processes, in: NeurIPS 2020 Competition and Demonstration Track, PMLR, 2021, pp. 325–343.
- [5] Q. Qi, P. H. Kwok, Traffic4cast 2020—graph ensemble net and the importance of feature and loss function design for traffic prediction, arXiv preprint arXiv:2012.02115 (2020).
- [6] H. Martin, Y. Hong, D. Bucher, C. Rupprecht, R. Bufat, Traffic4cast-traffic map movie forecasting—team mie-lab, arXiv preprint arXiv:1910.13824 (2019).

- [7] S. Choi, Utilizing unet for the future traffic map prediction task traffic4cast challenge 2020, arXiv preprint arXiv:2012.00125 (2020).
- [8] S. A. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J. R. Ledsam, K. H. Maier-Hein, S. Eslami, D. J. Rezende, O. Ronneberger, A probabilistic u-net for segmentation of ambiguous images, arXiv preprint arXiv:1806.05034 (2018).
- [9] A. Myronenko, 3d mri brain tumor segmentation using autoencoder regularization, in: International MICCAI Brainlesion Workshop, Springer, 2018, pp. 311–320.
- [10] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (elus), arXiv preprint arXiv:1511.07289 (2015).
- [11] Y. Wu, K. He, Group normalization, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.
- [12] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, C. Bregler, Efficient object localization using convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 648–656.
- [13] I. Loshchilov, F. Hutter, Sgdr: Stochastic gradient descent with warm restarts, arXiv preprint arXiv:1608.03983 (2016).

A. Experiments on R1

Table A1 details some of the experiments done on R1 to explore which input features should be included in the final model. All these experiments were done using the training and validation data provided. The underlying assumption was that the results from these experiments would read across to the final leaderboard.

Table A1
Summary of experimental results on R1

Experiment	Base	1	2	3	4
<i>ctth_pres</i>	-	-	Yes	Yes	Yes
<i>crr_accum</i>	-	Yes	Yes	Yes	Yes
<i>ct</i>	-	-	Yes	Yes	Yes
<i>ctth_tempe</i> mask	-	-	-	Yes	Yes
<i>ctth_alt</i>	-	-	Yes	-	-
Interpolated <i>ctth_tempe</i>	-	-	-	-	Yes
Epoch	20	27	32	24	20
Training score	0.2247	0.2155	0.2091	0.2087	0.2229
Validation score	0.1933	0.1935	0.1894	0.1879	0.1889