Applying Teacher Forcing to LSTMs for Improved Cloud Cover Forecasting

G079 (s1846118, s1827995, s1803949)

Abstract

In this project, we look to improve the performance of the convolutional Long Short-Term Memory (ConvLSTM) network for the purpose of predicting cloud cover in future satellite images. This model produces increasingly blurred predictions as the number of future timesteps increases, a problem we aim to mitigate with the application of teacher forcing - a novel training strategy for this purpose, in which the model's input is swapped with the ground truth label at given probability. Our results found that applying teacher forcing, with the right probability annealing method, resulted in an improved mean squared error validation loss over the base model, as well as improved visual predictions.

1. Introduction

The UK National Grid aims to run at "zero-carbon" by 2025. The biggest problem currently delaying this is the unpredictably of renewable energy sources - solar PV energy in particular - and the resulting reliance on fossil fuel reserves to maintain balance in the energy grid (Ahmed et al., 2020), (Yang et al., 2018). Solar PV energy is the largest source of uncertainty for the UK National Grid forecasts. Therefore, better solar PV energy output forecasts are essential in achieving this "zero-carbon" goal. With improved PV forecasting, it has been estimated that the National Grid can reduce emissions by up to 100,000 tonnes per year along side predicted savings of £1-10 million (Jamie Taylor, 2016).

The output generated by a solar PV installation relies most heavily on cloud cover (Nespoli et al., 2022) - a highly volatile parameter in any weather forecasting model. The ability to produce detailed near-term solar PV energy output forecasts therefore requires a highly accurate cloud cover forecasting model. The World Meteorological Organisation defines this type of modelling as 'nowcasting' - the near-term weather forecasting of a period up to of 6 hours ahead.

Current state-of-the-art deep learning approaches to cloud cover nowcasting formalise the problem as a spatio-temporal sequence prediction problem, generating future cloud cover estimations based on past satellite images. This has led to the dominance of sequence transduction architectures in the area, such as the Long Short-Term Memory (LSTM) architecture (Tian et al., 2019) (with a recent rise in transformer-based architectures).

The work presented in this paper uses data and resources acquired from Open Climate Fix (OCF), a not-for-profit product lab with the goal of developing the "industry standard" near-term climatic forecasting system for predicting solar PV energy output from satellite imagery via deep learning.

One architecture currently in use by OCF is that of the convolutional LSTM (ConvLSTM) model, an adaptation on that initially proposed in 2015 (Shi et al., 2015a). This model was previously the state-of-the-art in spatiotemporal precipitation nowcasting, a prediction task similar to that of cloud cover forecasting.

A significant problem with LSTM-based models, and other recurrent networks in general, is that the model predictions become increasingly 'blurred' over larger time scales (Shi et al., 2015a). Consider a model input of a single context image x, with the goal of predicting the resulting image at the next 5 timesteps $x+1, \ldots, x+5$. As the timestep increases, the quality of the model predictions degrades, since each prediction is based solely on the last.

In this paper, we attempt to improve on this issue with the ConvLSTM architecture through the implementation of the "teacher forcing" training technique. This technique has been applied to similar tasks, such as object tracking, with some success (Roros & Kak, 2022).

The technique involves replacing the model input with that of the ground truth label at a given probability, which decreases with each epoch. Since the model bases each future prediction beyond timestep $x+1$ solely on the its prediction (i.e. prediction $x+2$ is generated solely from $x+1$) then if its previous prediction was poor, all subsequent predictions will likely be poor also. Occasionally swapping these predictions with the ground truth label is a means of mitigating this problem by 'correcting' the model, effectively "forcing" it to learn how the desired output is represented. The reasoning as to why this approach may help in improving blurred spatiotemporal predictions is further explained in more detail in 3.4.1.

2. Data set and task

The dataset for this task consists of satellite imagery obtained from the EUMETSAT Spinning Enhanved Visible and InfraRed Imager (SEVIRI) RSS satellite and is publicly available. These images are obtained from the satellite at an elevation of 36,000 km above Earth, with each pixel within the image therefore representing a 3 km x 3 km region. *Figure 1* below displays an example image from this dataset.

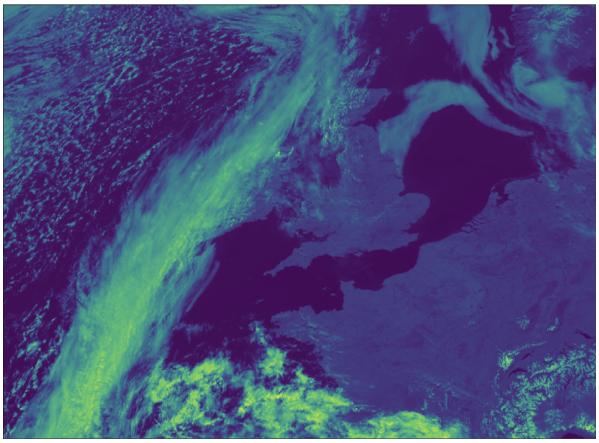

Figure 1. Example satellite image, obtained on 2021-06-01.

This satellite imagery consists of RGB-composite based images, combined with temperature information. This RGB-type image focuses on high cloud monitoring, since high clouds provide clearer colour contrasts with each other than lower-level clouds, differentiating themselves from lower-level clouds and cloud-free regions.

The images are captured daily at 5-minute intervals (i.e., at 00:00, 00:5, . . . , 23:55) over the date range from January 2020 to November 2021 inclusively, and reduced to a total 27,944 images. The dataset is split 90/10 into training and validation sets, resulting in 25,149 images for training, and 2,795 for validation. Each image is obtained from a small subset of the Rapid Scanning Service (RSS) area covering the United Kingdom and North Western Europe with an overall size of 891 x 1,843 pixels (including space pixels and minor cropping to facilitate a square image). Due to GPU memory constraints (with the extremely large size of the ConvLSTM architecture), the images are reduced to size 256 x 256 pixels and passed to the model in small batches of size 3. Each image is also normalised to the range [0-1].

The major deficiency with this satellite imagery dataset is that some cloud top features, such as overshooting tops, are not as identifiable to midday when compared to at low solar elevation. The other problem with this RGB composite type image is that snow-covered land, fog, low-level clouds, and mid-level clouds can appear fairly indistinguishable. These deficiencies should not, however, affect our investigation since we seek to examine the impact of a new training technique, teacher forcing, on our model, rather than simply maximise performance.

3. Methodology

3.1. Reccurent Neural Networks

Recurrent Neural Networks (RNNs) are a simple generalisation of the standard feedforward neural network to allow for sequential input and output (i.e. sequence transduction).

The typical RNN architecture consists of an input layer x, output layer y, and recurrently connected hidden layers h (Elman, 1990). This recurrent connection between the previous and current hidden layers enables the model to predict the current output conditioned on previous information.

However, the range of information from which the standard RNN can feasibly access is, in practice, limited to 5 - 10 time lags. This is due to the vanishing (or exploding) gradients problem - with the architecture described above, the gradient reduces/increases exponentially during the recurrent error backpropagation. Various variants on the RNN architecture have been proposed to better model these long-term dependencies (Cho et al., 2014).

3.2. Long-Short Term Memory

One of these variants is the encoder-decoder Long Short-Term Memory (LSTM) network, introduced in Google's 2014 paper "Sequence to Sequence Learning with Neural Networks" as an improved general end-to-end approach to sequence transduction.

The standard LSTM architecture addresses the above-mentioned gradients problem when modelling long-range temporal dependencies by replacing the hidden layer connections with 'memory cells' which essentially act as accumulators of information.

Understanding the purpose of each computation performed by the cell is largely unimportant in the context of this paper. It essentially enforces constant error flow, presenting gradients from vanishing (or exploding) too quickly by trapping the gradient in the cell (in what's known as a "constant error carousel"). This architecture has been proven by as capable of bridging time lags in excess of 1,000 time steps, a significant improvement over the basic RNN.

Google's application of the standard LSTM architecture stacked two LSTM networks, each consisting of the above-described memory cells, together in a multilayered approach (Sutskever et al., 2014a). The first network performs the role of an encoder, mapping the input sequence one timestep at a time to a large fixed-dimensional vector representation. The second, deeper, decoder network then extracts the target sequence from the encoded vector representation. The initial states and cell outputs of the decoder network are directly copied from the final state of the encoder network. This approach proved highly successful in modelling more complex sequences with longer range

temporal dependencies.

3.3. Convolutional Long Short Term Memory

Despite the aforementioned success of the LSTM for modelling temporal dependencies, the architecture is not well suited for the spatiotemporal modelling between image sequences due to its lack of spatial information encoding. The convolutional LSTM (ConvLSTM) network, specifically for the purpose of precipitation nowcasting, in order to better model spatiotemporal dependencies, by extending the encoder-decoder LSTM architecture to include convolutional structures (Shi et al., 2015b). A slight adaptation of this model architecture is the one used for the experiments presented in this paper.

The actual architecture of the ConvLSTM network itself is identical to that of the LSTM as described above. The main difference lies in the use of the convolutional operator in the layer connections. The other differing feature of the ConvLSTM is that all cell outputs, hidden states, and other data structures throughout the network, are held in three-dimensional tensor form in order to preserve the spatial information from the input. This ConvLSTM network has been proven to significantly better map spatiotemporal correlations between image sequences than the standard LSTM, as well as actually outperforming the state-of-the-art ROVER algorithm for the task of precipitation nowcasting, a highly similar task to that of cloud cover nowcasting.

Recent research has proposed numerous further improvements on the standard ConvLSTM mode for the purpose of cloud-cover nowcasting (and segmentation) (Tan et al., 2018), (Rußwurm & Körner, 2018), (Bo et al., 2020). However, each of these proposed methods suffer from the same problem of increasingly blurred future predictions due to the inaccuracy of recurrent architectures in predicting more than a single timestep into the future (one-step-ahead prediction).

3.4. Teacher Forcing

Teacher forcing is a training strategy designed to improve the efficiency and speed at which recurrent neural networks (RNNs, LSTMs, etc.) can be trained. With a given probability, the model's recurrent prediction at a time step is swapped with the ground truth label (i.e. in this case, the real satellite image) for that time step. This is then fed back into the model as the input for the prediction of the next time step, effectively 'forcing' the model to learn how the desired output is represented. The probability of a 'swap' occurring should anneal per epoch as the model trains; for example, linearly as defined by (1), or exponentially as defined by (2).

As described in the previous subsections, the ConvLSTM model is a sequence transduction, or 'seq2seq' model (Sutskever et al., 2014b), meaning it takes a sequential input and produces a sequential output, using an encoder-decoder architecture. The input sequence is first mapped into some fixed-dimensional vector representation by the encoder, which is then passed to the decoder to generate an output vector as the prediction. This output vector is then passed back to the decoder to generate the next prediction, and so forth. This decoding process goes on for as many output predictions that one wants the model to generate. *Figure 2* shows how the ConvLSTM would pass a vector sequence (x_1, x_2, x_3) into the encoder phase of the model. These are iteratively transformed to one encoded vector which is then passed to the decoder phase and recurrently transformed into three outputs (y_1,y_2,y_3). Looking at the decoder phase, we should note that the model will create an output at each iteration of the decoding, based solely on the previous decoded output.

The problem we are addressing in this paper occurs in the decoder phase of the model. As discussed above, the model bases future predictions solely off of previous predictions. Consider *Figure 2* once more.At the decoder phase of the training iteration, the model first generates a prediction y_1. If this prediction is poor, then the subsequent y_2 prediction will also be poor, since its prediction is based solely on y_1. This trend continues for the y_3 prediction. Essentially, an initially poor prediction can ruin all subsequent predictions and result in a high calculated loss for the iteration; this can cause the model to overcompensate during the error back propagation, leading to slower and potentially worse loss convergence over a given number of epochs.

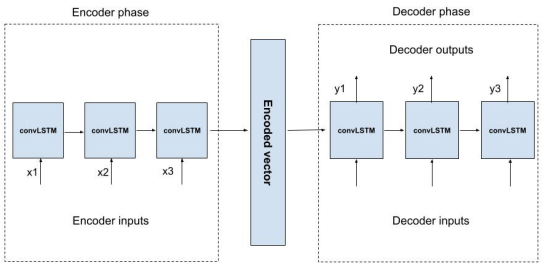

Figure 2. Original diagram showing the encoding and decoding phase of the ConvLSTM. We can see the inputs which are transformed to encoded vectors in the 'encoding phase' are then passed to the 'decoding phase' and recurrently decoded, where each decoded vector (y_1, y_2, y_3) is passed back into the subsequent ConvLSTM cell. Note that this diagram shows encoding and decoding states of length 3, however these phases can be of arbitrary length.

$$P_{\text{Ground Truth}} = 1 - \frac{epoch}{total\ number\ of\ epochs} \tag{1}$$

$$P_{\text{Ground Truth}} = \exp(-0.5 * epoch) \tag{2}$$

$$P_{\text{Ground Truth}} = \exp(-0.5 * timestep) \tag{3}$$

3.4.1. Why Teacher Forcing?

Teacher forcing can be applied in an attempt to mitigate this problem. At the beginning of the models training, the probability of an output having a large loss and ruining the subsequent training iterations is high. This probability will slowly decrease as the model learns. Therefore, applying teacher forcing to our model with a high probability during the early epochs can reduce the impact of poor initial predictions by swapping these predictions with the ground truth labels for generating the next prediction instead, to continue the sequence. The swap probability is then decreased at some rate with the increasing epochs, as discussed above, slowly phasing out teacher forcing and allowing the model to learn.

If we refer again back to *Figure 2* and consider the event that output y_1 is selected to have teacher forcing applied to it. The model would still store y_1 as an output (which will contribute to the overall loss calculation for the given training iteration). However, the ground truth value for y_1, $\hat{y_1}$, would instead be used as the recurrent input to the next ConvLSTM to predict y_2. This could improve the rest of the predictions of the iteration as a result, and potentially allow the model to learn better representations.

3.4.2. How To Apply Teacher Forcing to ConvLSTM

Given this description of teacher forcing, and how it can help the model train, we now look at it can be applied to the ConvLSTM model. *Figure 3* and *Figure 4* show how we would apply teacher forcing to the ConvLSTM model. *Figure 3* shows the model encoding the input vectors x_1, x_2, and x_3 normally, as described above. When the model outputs y_1 however, this output is selected (with some given probability) to have teacher forcing applied. The value for y_1 is then stored as an output. *Figure 4* shows how we then return to the encoding phase of the model, passing only $\hat{y1}$ as an input (since $\hat{y1}$ needs to be mapped to the fixed-dimensional vector space from which it can be decoded). From here, the model will generate the encoder vector the same as previously described, and begin recurrently decoding outputs until the next output is selected for teacher forcing. It should be noted, for clarity, that when the probability of teacher forcing being applied is at 0.5, we would expect this process to occur for about roughly half of the output vectors.

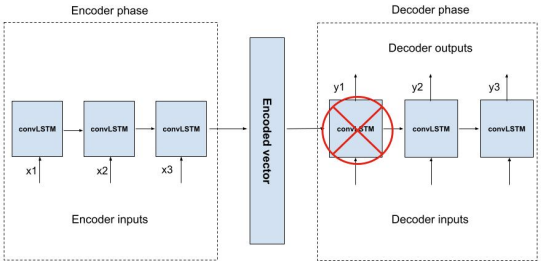

Figure 3. Original diagram showing a representation of teacher forcing being applied after prediction y_1 is output by the ConvLSTM. We can see how y_1 is stored as an output but is not applied to the next cell for predicting y_2.

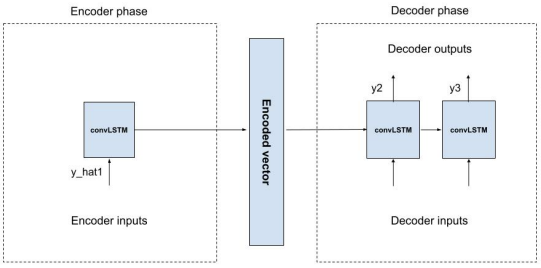

Figure 4. Original diagram showing how we would restart the encoding phase of the model after teacher forcing has been applied. We simply encode $\hat{y_1}$, before resuming the decoding phase for the rest of the forecast steps.

4. Experiments and Results

4.1. Ensuring Comparable Results

The standard model architecture employed across all experiments consists of two LSTM networks, as described in 3.2, with a total of 128 layers in each. This is the same architecture as employed by (Shi et al., 2015b). Our implementation differs from this in three ways:

- Instead of RMSProp, we instead use the Adam optimiser with a learning rate of 1e-3 and beta parameters of 0.9 and 0.999 respectively.
- A smaller kernel of size 1 is employed in order to detect small image differences due to the slow nature of cloud movements.
- Mean Squared Error (MSE) is used to measure the model's loss.

The source code for the unmodified baseline model was obtained from OCF, and can be viewed at:
https://github.com/openclimatefix/satflow

4.2. Experiments

We set out a number of experiments highlighting different probability annealing methods combined with different implementations of the teacher forcing strategy to determine which, if any, is most effective.

Each model, for each experiment, is run for a set 100 epochs. The models are fed a single input image *x*, with the task of predicting the next 5 timesteps (i.e. *x+1*, ... , *x+5*). The input is passed to the models in batches of size 3 (due to GPU memory constraints).

4.2.1. Baseline Model

The baseline model in our investigation is the standard ConvLSTM model, with default parameters, and without teacher forcing applied.

4.2.2. Linearly Decaying Probability

The first experiment we implemented was applying a basic linear per epoch annealing method for the teacher forcing probability, defined by (1). This method will mean that for early epochs, the probability that a given output has teacher forcing applied to it will be close to 1. This probability will then decrease slowly (and linearly), not reaching close to 0 until the final epochs.

4.2.3. Exponentially Decaying Probability

The next experiment we conduct was applying an inverse exponential decay per epoch annealing method for the teacher forcing probability. The idea behind this experiment is that, intuitively, it would make sense that teacher forcing would have most benefit during the early epochs, in which the model has a much higher probability of outputting a poor prediction, but should be phased out more rapidly in order to give the model the chance to learn on its own during the later epochs. We define this annealing method in (2). This method will result in most of the teacher forcing occurring in the first 5 epochs, after which the probability will have dropped close to 0.

4.2.4. Decaying Probability Within The Decoder

The first two applications of teacher forcing imply that the probability of teacher forcing being applied to each of the decoder outputs (i.e. timesteps) for an iteration remains constant. However, applying teacher forcing to the final decoder outputs is likely less effective than applying it to the early decoder outputs. The reason for this is that, as discussed above, if our model outputs a poor prediction for the early outputs then all of the subsequent outputs will be useless. On the other hand, if the later predictions are poor then there should not be such a significant affect, due to the limited number of subsequent predictions. For this experiment we propose a new per timestep probability annealing method for applying teacher forcing within the decoder. We decided to apply the same exponential annealing as defined by (2), but with the subtle difference highlighted in (3). This means that the probability of teacher forcing being applied will be close to 0 at timestep *x+5*.

4.2.5. Combining Experiments

We will run a combination of these experiments in order to find the best combination, if any, of the aforementioned teacher forcing applications. We will run the model for each implementation on both the epoch and timestep level.

4.3. Results

The full results for each experiment can be viewed in the table in 8. Throughout this section, including in 8, loss is defined as the average Mean Squared Error (MSE) loss for timestep *x+5* in each experiment.

4.3.1. No Epoch-Level Teacher Forcing

Here, no teacher forcing is applied at the epoch level, combined with both no teacher forcing and exponential decay teacher forcing within the decoder at the timestep level (i,e, as we make predictions at each iteration). Applying no teacher forcing at the epoch level with exponential decay teacher forcing at the timestep level resulted in slightly better training but significantly worse validation performance than our baseline model (i.e. without teacher forcing applied at any level). This indicates potential overfitting on the training data.

Figure 5 shows an example prediction for timestep *x+5* from this baseline model, alongside the ground truth image for comparison.

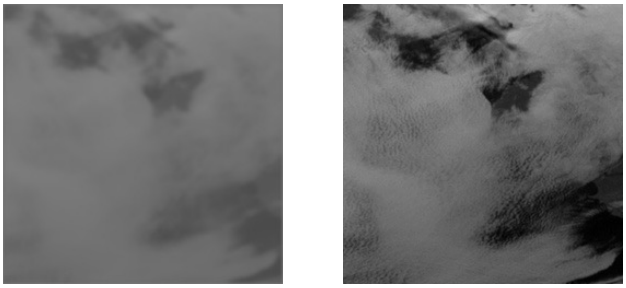

Figure 5. Original diagram showing an example timestep x+5 predicted image from our baseline model with no teacher forcing applied at either the epoch level or timestep level, alongside the ground truth image for comparison.

4.3.2. Linearly Decaying Probability

Here, linear teacher forcing is applied at the epoch level, combined with both no teacher forcing and exponential decay teacher forcing applied within the decoder at the timestep level.

Applying linear decay teacher forcing at the epoch level with no teacher forcing at the timestep level resulted in worse performance for both training and validation than the baseline model, suggesting this method of probability annealing is a poor approach.

Applying linear teacher forcing at the epoch level with exponential decay teacher forcing at the timestep level resulted in even worse performance for both training and validation, in line with the results seen for its application to the baseline model.

The results obtained from this method are also significantly worse than the baseline model on examination of the model predictions. Figure 6 below shows an example timestep x+5 predicted image from the best performing model of these experiments; linear decay teacher forcing with no teacher forcing applied at the timestep level. As you can see, the predicted image is significantly worse than that of the base model, and looks almost "inverted" in comparison to the ground truth.

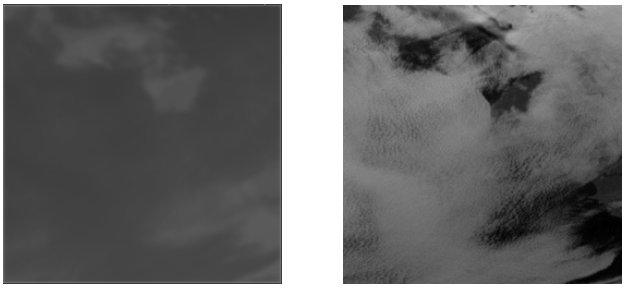

Figure 6. This figure shows an example timestep x+5 predicted image from the linear decay teacher forcing model with no teacher forcing applied at the timestep level, alongside the ground truth image for comparison.

4.3.3. Exponentially Decaying Probability

Here, exponential decay teacher forcing is applied at the epoch level combined with no teacher forcing and exponential decay teacher forcing applied within the decoder at the timestep level.

With exponential decay teacher forcing at the epoch level combined with no teacher forcing at the decoder level, the model outperformed the baseline, and our other teacher forcing implementations, in terms of both training and validation performance. This improvement is ALSO clear to see from the actual predicted images produced by the model.

Applying exponential decay teacher forcing at the epoch level now combined with exponential decay teacher forcing at the timestep level also again resulted in worse performance than without its application. This trend has continued across our experiments, indicating that applying teacher forcing at the timestep level is a poor approach.

The side by side images in *Figure 8* shows a comparison between example images produced for the best performing model from each of the above described experiments.

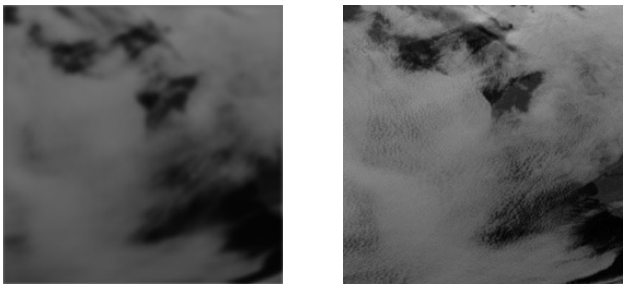

Figure 7. This figure shows an example timestep x+5 predicted image from the exponential decay teacher forcing model with no teacher forcing applied at the timestep level, alongside the ground truth image for comparison.

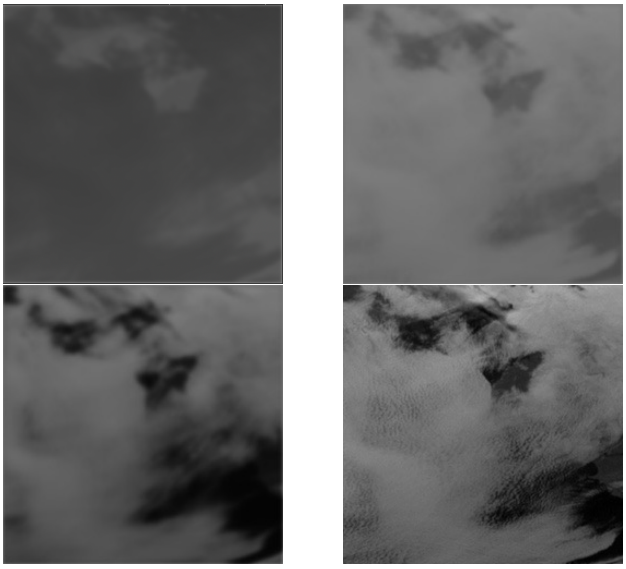

Figure 8. This figure shows the baseline ConvLSTM prediction (top right) compared to linear teacher forcing with no teacher forcing applied at the timestep level (top left), exponential teacher forcing with no teacher forcing applied at the timestep level (bottom left), and the ground truth (bottom right).

5. Discussion

Analysing the results obtained in the previous section we can see that applying teacher forcing within the decoder at the timestep level alone did not improve its prediction when compared to the baseline model without teacher forcing applied. This is intuitive; applying such teacher forcing alone may disrupt training as it is not phased out in later epochs, instead remaining during the entirety of the model's training. With exponential decay teacher forcing, the first few timestep predictions will always have a large probability of being swapped for the ground truth which remains consistent at every epoch, thereby potentially not giving the model a chance to really learn.

Linear teacher forcing at the epoch level produced worse results than those of the baseline model. An explanation for this could be that linearly decreasing the probability of teacher forcing being applied means it does not decrease at a fast enough rate. Similarly to as discussed

above, applying teacher forcing with a linearly decaying probability may disrupt training, as the swap probability decays at such a slow rate that swaps will occur fairly regularly throughout the entire training phase, leaving the model with little chance to really learn. The small training loss with significantly higher validation loss reinforces this.

However, exponential decay when applied at epoch level, with no teacher forcing occurring in the decoder, produced better results than both the baseline model and the other methods models applied. This is an interesting result and can potentially be explained by the fact that, as mentioned in 3.4, the majority of the poorly predicted images produced by the model appear in the first few epochs, whilst the probability remains high. The application of teacher forcing is then phased out rapidly, dropping to close to 0 by epoch 5. Then, once the teacher forcing is phased out, the model can train on its own with the benefit of the training gain from the early epochs.

Figure 8 shows a great example of how applying teacher forcing this way can result in less blurry image predictions for later time steps.

6. Future Research

An initial suggestion for future research is to look at other, more complex, means of annealing or varying the teacher forcing probability throughout the training phase.

There are also many ways in which teacher forcing can be applied to this complex model architecture. Future work should look to find potentially more optimal ways of applying teacher forcing to the ConvLSTM, or other recurrent networks in use today.

For example, one way which future researchers could attempt to improve on our implementation of teacher forcing on the ConvLSTM architecture is by trying a different method of encoding when teacher forcing is applied to a given cell. If we refer back to *Figure 3* and *Figure 4*, we described how teacher forcing would be applied in this paper by re-encoding the ground truth only before carrying on with decoding. The issue with this method is that when teacher forcing is applied, we solely pass the ground truth back to the encoder before resuming our decoding process; therefore we lose all long-term memory from the previous inputs. Although we are still reducing the chance of a poorly formed image ruining a whole training iteration, we are also losing a lot of information by applying teacher forcing this way. *Figure 9 and 10* shows a proposed improved method for applying teacher forcing the the ConvLSTM model. Consider applying teacher forcing to output y_2 in *Figure 9*. In this paper, to apply teacher forcing we would only encode the ground truth of y_2 before continuing the decoding phase. However, if we now look at *Figure 10*, we can see that given teacher forcing is applied to output y_2, then to optimise the encoding phase with the ground truth of y_2, we could 're-encode' from the beginning each input *(*x_1, x_2, x_3*)* and all the subsequent y outputs all the way up to the ground truth of y_2. This would preserve long-term memory and potentially further improve performance after teacher forcing is applied.

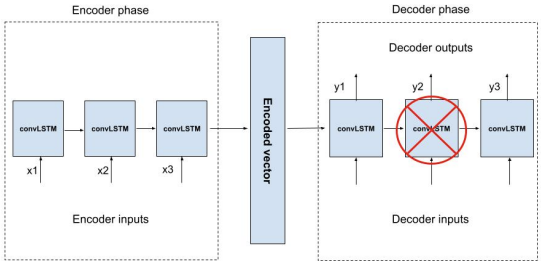

Figure 9. Original diagram showing teacher forcing applied at output y_2 in the ConvLSTM

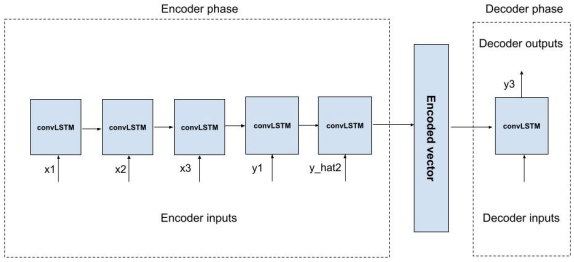

Figure 10. Original diagram showing an alternative proposal for applying teacher forcing by 're-encoding' from the input sequence all the way to the ground truth.

7. Conclusions

To conclude, we have found that applying teacher forcing can in fact improve the performance of the baseline ConvLSTM for predicting future cloud cover satellite images, when applied properly. Our improvement over the baseline ConvLSTM model was achieved by teacher forcing applied with 'exponential decay' at the epoch level, meaning we decrease the probability of teacher forcing being applied for every epoch by equation (2). We also experimented with applying teacher forcing at the epoch level with a linearly decaying probability, which produced significantly worse results than the baseline. Applying teacher forcing with varying probability at the 'timestep level', that is decaying at some rate per timestep or as we make each prediction, also worsened the results for each model to which it was applied.

Our results motivate further research into more complex ways of applying teacher forcing to the ConvLSTM, and other recurrent architectures, including advanced probability annealing/variation methods as well as different encoding methods.

8. Results Table

Per Epoch Annealing Method	Per Cell Annealing Method	Training Loss (MSE)	Validation Loss (MSE)
Baseline (None)	Baseline (None)	0.037	0.011
	Exponential	0.025	0.167
Linear	Baseline (None)	0.056	0.622
	Exponential	0.063	0.713
Exponential	Baseline (None)	0.011	0.0093
	Exponential	0.041	0.177

References

Eumetsat: Rapid scanning service. URL https://www.eumetsat.int/rapid-scanning-service. Accessed: 2022-10-02.

Ahmed, R., Sreeram, V., Mishra, Y., and Arif, M.D. A review and evaluation of the state-of-the-art in pv solar power forecasting: Techniques and optimization. *Renewable and Sustainable Energy Reviews*, 124:109792, 2020. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2020.109792. URL https://www.sciencedirect.com/science/article/pii/S1364032120300885,.

Bo, Ma, Ning, Yang, Chenggang, Cui, Jing, Chen, and Peifeng, Xi. Cloud position forecasting based on convlstm network. In *2020 5th International Conference on Power and Renewable Energy (ICPRE)*, pp. 562–565. IEEE, 2020.

Cho, Kyunghyun, van Merrienboer, Bart, Gülçehre, Çaglar, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. URL http://arxiv.org/abs/1406.1078.

Elman, Jeffrey L. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. ISSN 0364-0213. doi: https://doi.org/10.1016/0364-0213(90)90002-E. URL https://www.sciencedirect.com/science/article/pii/036402139090002E.

Jamie Taylor, Aldous M. Everard, Julian Briggs Alastair R. Buckley Stephen Casement Hannah Jones Joseph Harwood Jeremy Caplin. Estimating real time gb aggregated solar pv generation. 2016. URL https://www.researchgate.net\T1\guilsinglrightPV-Live\T1\guilsinglrightdownload.

Nespoli, Alfredo, Niccolai, Alessandro, Ogliari, Emanuele, Perego, Giovanni, Collino, Elena, and Ronzio, Dario. Machine learning techniques for solar irradiation nowcasting: Cloud type classification forecast through satellite data and imagery. *Applied Energy*, 305:117834, 2022. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2021.117834. URL https://www.sciencedirect.com/science/article/pii/S0306261921011600.

Roros, Constantine J. and Kak, Avinash C. maskgru: Tracking small objects in the presence of large background motions. *CoRR*, abs/2201.00467, 2022. URL https://arxiv.org/abs/2201.00467.

Rußwurm, Marc and Körner, Marco. Convolutional lstms for cloud-robust segmentation of remote sensing imagery. *arXiv preprint arXiv:1811.02471*, 2018.

Shi, Xingjian, Chen, Zhourong, Wang, Hao, Yeung, Dit-Yan, Wong, Wai-Kin, and Woo, Wang-chun. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *CoRR*, abs/1506.04214, 2015a. URL http://arxiv.org/abs/1506.04214.

Shi, Xingjian, Chen, Zhourong, Wang, Hao, Yeung, Dit-Yan, Wong, Wai-Kin, and Woo, Wang-chun. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015b.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014a. URL http://arxiv.org/abs/1409.3215.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014b.

Tan, Chao, Feng, Xin, Long, Jianwu, and Geng, Li. Forecast-clstm: A new convolutional lstm network for cloudage nowcasting. In *2018 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1–4. IEEE, 2018.

Tian, Lin, Li, Xutao, Ye, Yunming, Xie, Pengfei, and Li, Yan. A generative adversarial gated recurrent unit model for precipitation nowcasting. *IEEE Geoscience and Remote Sensing Letters*, 17(4):601–605, 2019.

Yang, Dazhi, Kleissl, Jan, Gueymard, Christian A., Pedro, Hugo T.C., and Coimbra, Carlos F.M. History and trends in solar irradiance and pv power forecasting: A preliminary assessment and review using text mining. *Solar Energy*, 168:60–101, 2018. ISSN 0038-092X. doi: https://doi.org/10.1016/j.solener.2017.11.023. URL https://www.sciencedirect.com/science/article/pii/S0038092X17310022. Advances in Solar Resource Assessment and Forecasting.