Using Convolutional LSTMs for Cloud-Robust Segmentation of Remote Sensing Imagery

Anonymous Author(s)
Affiliation
Address
email

Abstract

Dynamic spatiotemporal processes on the Earth can be observed by an increasing number of optical Earth observation satellites that measure spectral reflectance at multiple spectral bands in regular intervals. Clouds partially covering the surface is an omnipresent challenge for the majority of remote sensing approaches that are not robust regarding cloud coverage. In these approaches, clouds are typically handled by cherry-picking cloud-free observations or by pre-classification of cloudy pixels and subsequent masking. In this work, we demonstrate the robustness of a straightforward *convolutional long short-term memory* network for vegetation classification using all available cloudy and non-cloudy satellite observations. We visualize the internal gate activations within the recurrent cells and find that, in some cells, modulation and input gates close on cloudy pixels. This indicates that the network has internalized a cloud-filtering mechanism without being specifically trained on cloud labels. The robustness regarding clouds is further demonstrated by experiments on sequences with varying degrees of cloud coverage where our network achieved similar accuracies on all cloudy and non-cloudy datasets. Overall, our results question the necessity of sophisticated pre-processing pipelines if robust classification methods are utilized.

Supplementary material can be accessed via `https://tinyurl.com/NIPS18ST-supplement`

1 Introduction

A wide range of dynamic spatiotemporal processes of the Earth can be observed with remote sensing satellites that revisit the same position on Earth at discrete time intervals. Seasonal vegetation life-cycles and other land cover dynamics are typically monitored at weekly intervals at spatial resolutions of several meters that allow distinguishing large single objects. Imagery acquired by these optical satellites is, however, regularly covered by clouds. These coverages are typically addressed by either selecting exclusively cloud-free observations or masking and removing clouds by computationally sophisticated pre-processing pipelines. We investigate the robustness of convolutional long short-term memory networks [8] with regard to temporal noise induced by cloud coverage for remote sensing imagery.

2 Related Work

Clouds distinguish themselves from ground pixels by their the high reflectance compared to ground pixels. Decision-tree based models [4, 10, 2] applied on expert-designed features are used for many remote sensing applications. The `fmask`algorithm [10] and improved versions [9, 1] additionally implement a projection of the detected cloud on the surface as initialization to additionally predict the shadow casted by the cloud. Other approaches extract features from a time series and utilize the

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

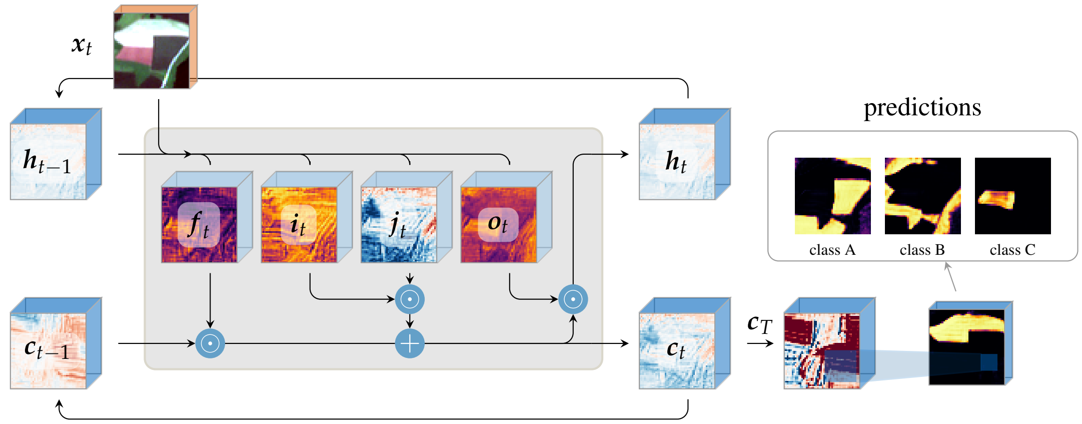

Figure 1: Illustration of the two-layer convolutional long short-term memory network (LSTM) topology. Each input image $\boldsymbol{x}_t$ of T images is passes sequentially to the LSTM encoder that extracts classification relevant features to the internal cell state tensor $\boldsymbol{c}_T$. A second convolutional layer compresses the dimensionality to the number of classes yielding activations per class.

sudden increase in reflectance to identify cloudy pixels [2]. *Convolutional neural networks (CNNs)* have also shown to compare well [11] indicating that these features can be learned by deep methods. These methods have proven beneficial in the last years and are widely implemented in remote sensing approaches. However, masking single pixels by a pre-classification introduces an additional layer of procedural complexity and raises the question of how to treat these pixels accordingly in the designed framework. Overall, cloud-filtering remains a pre-processing necessity for most remote sensing approaches that are prone to fail in the presence of data noise.

Similar to our work, only a few approaches have tried to design robust methods that do not require this additional pre-classification step. One approach added pre-classified cloud labels as additional prediction targets that allowed the implemented network to distinguish cloud from ground classes [7]. Also, ensemble-based methods of supervised classifiers have shown robustness regarding the appearance of clouds [6].

3 Method

In this section, we outline the theoretical basis of *convolutional long short-term memory* (convLSTM) networks utilized in this work and detail the employed network topology.

3.1 Convolutional Long Short-term Neural Networks

Long short-term memory networks (LSTM) [3] implement internal gates to control the gradient-flow through time and an additional container for long-term memory $\boldsymbol{c}_t$. This yields the LSTM update $(\boldsymbol{h}_t, \boldsymbol{c}_{t-1}) \leftarrow (\boldsymbol{x}_t, \boldsymbol{h}_{t-1}, \boldsymbol{c}_{t-1})$ that map an input $\boldsymbol{x}_t$ and short-term context $\boldsymbol{h}_{t-1}$ to a hidden representation $\boldsymbol{h}_t$. Additionally, a long-term cell state $\boldsymbol{c}_{t-1}$ is updated to $\boldsymbol{c}_t$ at each iteration and can store information for a theoretically unlimited number of iteration. Three gates control the update of the cell state

$$\boldsymbol{c}_t \leftarrow \boldsymbol{c}_{t-1} \odot \boldsymbol{f}_t + \boldsymbol{i}_t \odot \boldsymbol{j}_t \tag{1}$$

by element-wise multiplication denoted by the *Hadamard* operator $\odot$. The forget gate $\boldsymbol{f}_t = \sigma\left(\boldsymbol{x}_t * \boldsymbol{\theta}_{\text{fx}} + \boldsymbol{h}_{t-1} * \boldsymbol{\theta}_{\text{fh}} + \boldsymbol{1}\right)$ evaluates the influence of the previous cell state $\boldsymbol{c}_{t-1}$ with a sigmoidal $\sigma(\cdot) \in]0,1[$ activation function. The input and modulation gates

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{x}_t * \boldsymbol{\theta}_{\text{ix}} + \boldsymbol{h}_{t-1} * \boldsymbol{\theta}_{\text{ih}}\right), \text{and} \quad \boldsymbol{j}_t = \tanh\left(\boldsymbol{x}_t * \boldsymbol{\theta}_{\text{jx}} + \boldsymbol{h}_{t-1} * \boldsymbol{\theta}_{\text{jh}}\right) \tag{2}$$

are element-wise multiplied for the cell state update. The output gate $\boldsymbol{o}_t = \tanh\left(\boldsymbol{x}_t * \boldsymbol{\theta}_{\text{ox}} + \boldsymbol{h}_{t-1} * \boldsymbol{\theta}_{\text{oh}}\right)$ determines with the cell state the current cell output $\boldsymbol{h}_t \leftarrow \boldsymbol{o}_t \odot \boldsymbol{c}_t$. Convolutional recurrent networks implement a convolution, denoted by $*$, instead of a matrix multiplication of the formulation of recurrent networks. Each respective gate activation, referred by subscripts $\text{f}, \text{i}, \text{j}, \text{o}$, is controlled by trainable weights for input $\boldsymbol{\theta}_{\text{fx}}, \boldsymbol{\theta}_{\text{ix}}, \boldsymbol{\theta}_{\text{jx}}, \boldsymbol{\theta}_{\text{ox}} \in \mathbb{R}^{k \times k \times d \times r}$ and

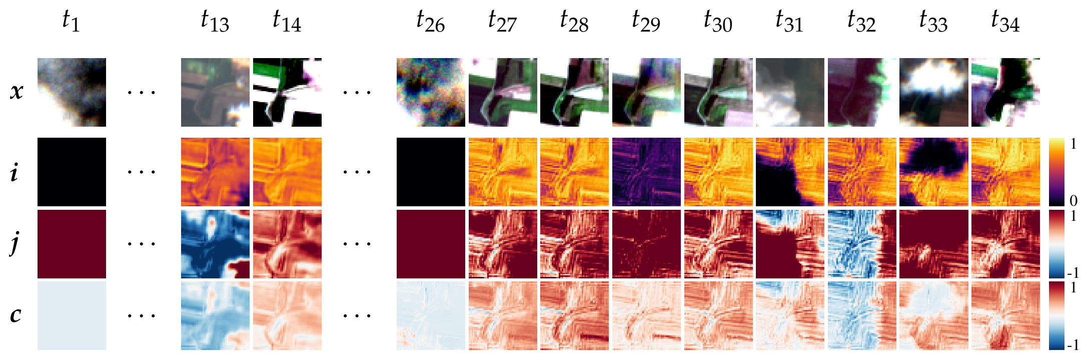

Figure 2: Activations of the cell state and selected gates of one convolutional LSTM cell that indicate that the cell has internalized a cloud-fiiltering scheme. The input gate $\boldsymbol{i}$ in this specific cell seems to be assigned values of zero on cloudy pixels as seen at steps $t = 13, 26, 31, 33$.

hidden representation $\boldsymbol{\theta}_{\text{fh}}, \boldsymbol{\theta}_{\text{ih}}, \boldsymbol{\theta}_{\text{jh}}, \boldsymbol{\theta}_{\text{oh}} \in \mathbb{R}^{k \times k \times r \times r}$ where d represents the dimensional depth of the input image, k the convolutional kernel size, and r a hyper-parameter determening the number of recurrent cells by setting dimensionality of the hidden states. With this change, image data of certain width, height and depth can be processed where convolutions partially connect the local pixel neighborhoods between layers.

3.2 Network architecture

We utilize this single-layer convolutional LSTM neural network to encode a sequence of T satellite images to the fixed length representation $\boldsymbol{c}_T$, as illustrated in Fig. 1. To balance the influence of the sequence order, we also encode the reversed sequence and append the final cell states. In initial published experiments, we found 256 recurrent cells to be optimal and used this hyper-parameter of dimensionality for the hidden tensors within the LSTM network.

After sequential encoding, the combined cell state is passed to a second convolution layer that compresses the dimensionality from 2×256 hidden dimensions to the number of classes. Applying softmax normalization produces activations that can be interpreted as network-confidences per class and are illustrated in the figure. We used convolutional kernels of $3 \times 3px$ in size throughout the network. To train, we evaluate the cross-entropy between the last layer and a one-hot representation of the ground truth labels. The influence of each weight on the evaluated loss is determined by back-propagated gradients and iterative adjustments are determined by the Adam optimizer[5].

4 Results

The primary objective for this network was to identify the type of cultivated crops in an area of interest of 100 km $\times$ 40 km. Hence, we trained our network end-to-end on label data describing the crop-type on disctinct field parcels. No additional label information about cloud coverages was used. We used a sequence of 46 SENTINEL 2 satellite observations from the year 2016 for this objective. This satellite measures the reflectances of 13 spectral bands at 10 m, 20 m, and 60 m resolution. To harmonize the data sources, we bi-linearly interpolated these to 10 m resolution and rasterized the crop labels accordingly. In this section, we evaluate the robustness of the proposed network regarding cloud coverage.

4.1 Long-short term memory cell activations

We trained the network on field crop labels for thirty epochs using raw sequences of cloudy and non-cloudy observations. The top row of Fig. 2 shows an partocular example input sequence of $T = 34$ images of $48 \times 48px$ in size. The following rows illustrate activations of the internal convolutional LSTM gates $\boldsymbol{i}, \boldsymbol{j}$ and cell state $\boldsymbol{c}$ given each input element. While all of the 256 recurrent cells likely contribute to the classification decision, only few were visually interpretable similar to the shown example. Following Eq. (1), the cell state is updated with new information based on

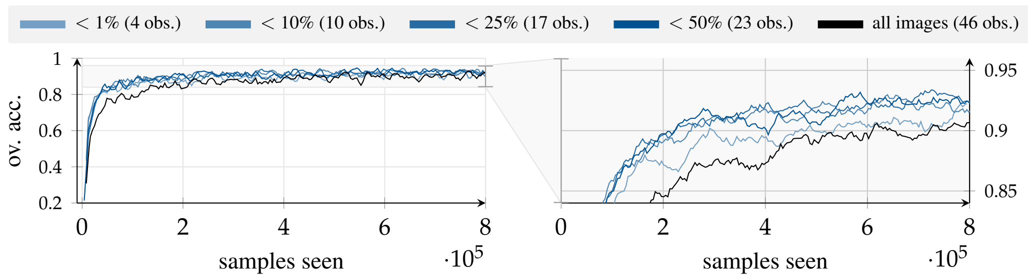

Figure 3: Overall accuracy over the training progress of the same convolutional LSTM network topology trained on datasets with different degrees of cloud coverage.

the input and modulation gates $\boldsymbol{i}, \boldsymbol{j}$. The activations in Fig. 2 of these gates in the second and third row show that the input gate $\boldsymbol{i}$ approaches zero at pixels that are covered by clouds. This effect can be observed at time steps $t = \{13, 26, 31, 33\}$. At time $t = 32$ the input gate seems unchanged, however, the modulation gate $\boldsymbol{j}$ changes sign. Overall, these results indicate that the convolutional recurrent network has internalized a mechanism for cloud-filtering. More activation examples can be obtained from the supplementary material.

4.2 Experiments with varying degrees of cloud coverage

In this experiment, we trained the network on datasets with different degrees of cloud coverage. To determine the cloud coverage per observation, all satellite observations have been pre-processed using the `fmask`algorithm implemented in the `Sen2Cor` software, as being common practice in remote sensing. This yields a per-pixel cloud classification label. With this, a cloud coverage pixel ratio per observation can be calculated. Based on this, several sub-datasets have been created with either all 46 observations, the 26 images covered with less than 50%, 17 images with less than 25%, 10 with less than 10%, and 4 completely cloud-free images.

We trained the network on these pre-filtered datasets. In Fig. 3 one can observe that the overall accuracy over the training process remains remarkably similar for all of the sub-sampled datasets. The right graph shows a zoomed view and reveals some differences between the dataset performances. Datasets containing observations and the four completely cloud-free observations have been slightly worse classified than the intermediate ones of 10%, 25%, and 40% coverage. It seems that the rejection of completely covered observations was beneficial as indicated by the slightly worse accuracy on the dataset of all observations. Similarly, the four cloud-free observations may have missed some characteristic vegetation-related events. Intuitively, these results show a trade-off between restrictions on cloud coverage and sequence length and demonstrate that cherry-picking single cloud-free observations may lead to inferior classification accuracy. Overall, these results demonstrate the robustness of the convolutional long short-term memory network to handle data containing temporal noise induced by cloud coverage.

5 Conclusion

Noise in temporal data is a common challenge for a variety of disciplines. In this work, we focused on noise induced by cloud coverage in multi-temporal remote sensing imagery. Most Earth observation approaches either select few completely cloud-free observations or use a pre-classification to mask cloudy pixels. The experiments of this work showed that this cloud-induced temporal noise can be learned purely from the data in an end-to-end fashion with an appropriate model design. We utilized long short-term memory cells that are popularly used in natural language processing tasks, such as translation or text generation in a straightforward two-layer network. Our results demonstrate this model design is able to consistently extract the classification-relevant features from observations between cloudy observations.

Our work questions the necessity of sophisticated, partly hand-crafted pre-processing pipelines for remote sensing imagery. These results show that methods perform well in the seemingly unrelated field of remote sensing and Earth observation. To encourage further research with spatiotemporal data in remote sensing and related fields, we will publish source code and data upon acceptance.

References

[1] David Frantz, Achim Röder, Thomas Udelhoven, and Michael Schmidt. Enhancing the detectability of clouds and their shadows in multitemporal dryland landsat imagery: extending fmask. *IEEE Geoscience and Remote Sensing Letters*, 12(6):1242–1246, 2015.

[2] Olivier Hagolle, Mireille Huc, D Villa Pascual, and Gérard Dedieu. A multi-temporal method for cloud detection, applied to formosat-2, venμs, landsat and sentinel-2 images. *Remote Sensing of Environment*, 114(8):1747–1755, 2010.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[4] André Hollstein, Karl Segl, Luis Guanter, Maximilian Brell, and Marta Enesco. Ready-to-use methods for the detection of clouds, cirrus, snow, shadow, water and clear sky pixels in sentinel-2 msi images. *Remote Sensing*, 8(8):666, 2016.

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[6] Chuc Duc Man, Thuy Thanh Nguyen, Hung Quang Bui, Kristofer Lasko, and Thanh Nhat Thi Nguyen. Improvement of land-cover classification over frequently cloud-covered areas using landsat 8 time-series composites and an ensemble of supervised classifiers. *International Journal of Remote Sensing*, 39(4):1243–1255, 2018.

[7] Marc Rußwurm and Marco Körner. Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images. In *CVPR Workshops*, pages 1496–1504, 2017.

[8] Shi Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[9] Zhe Zhu, Shixiong Wang, and Curtis E Woodcock. Improvement and expansion of the fmask algorithm: Cloud, cloud shadow, and snow detection for landsats 4–7, 8, and sentinel 2 images. *Remote Sensing of Environment*, 159:269–277, 2015.

[10] Zhe Zhu and Curtis E Woodcock. Object-based cloud and cloud shadow detection in landsat imagery. *Remote sensing of environment*, 118:83–94, 2012.

[11] Anze Zupanc. Improving cloud detection with machine learning. `https://medium.com/sentinel-hub/improving-cloud-detection-with-machine-learning-c09dc5d7cf13`, 2017.