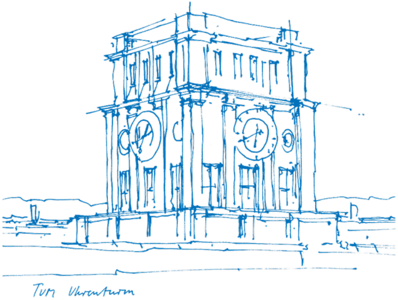
Remote Sensing Technology
TUM Department of Aerospace and Geodesy
Technical University of Munich

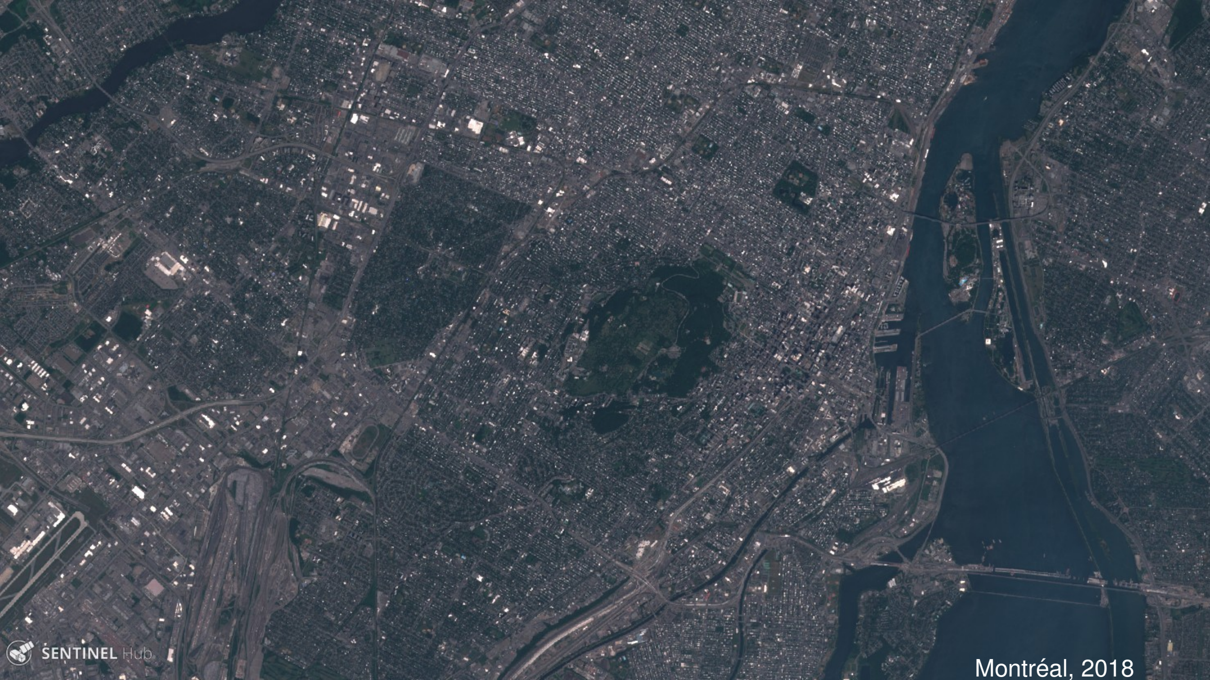
Cloud-Robust Classification of Remote Sensing Time Series

Φ-week 2019

Marc Rußwurm, Marco Körner

10th September 2019, ESA ESRIN, Frascati, Italy

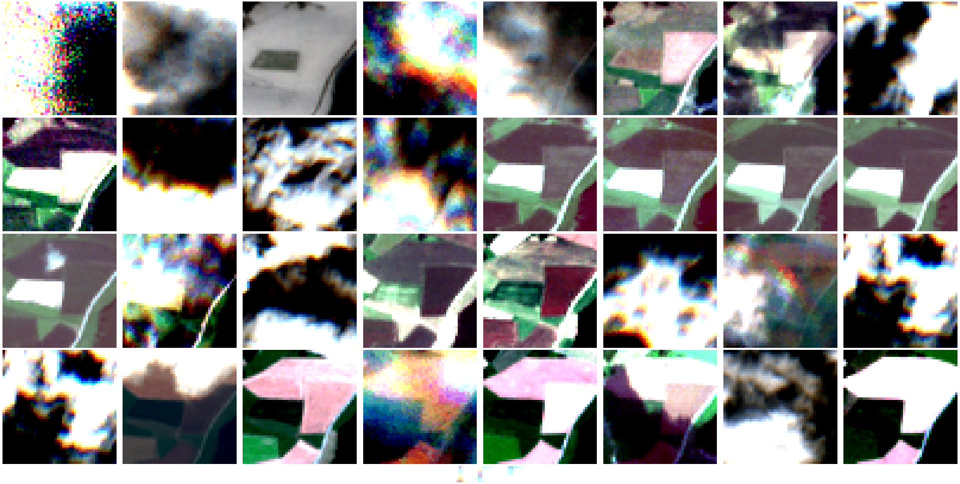




Montréal, 2018

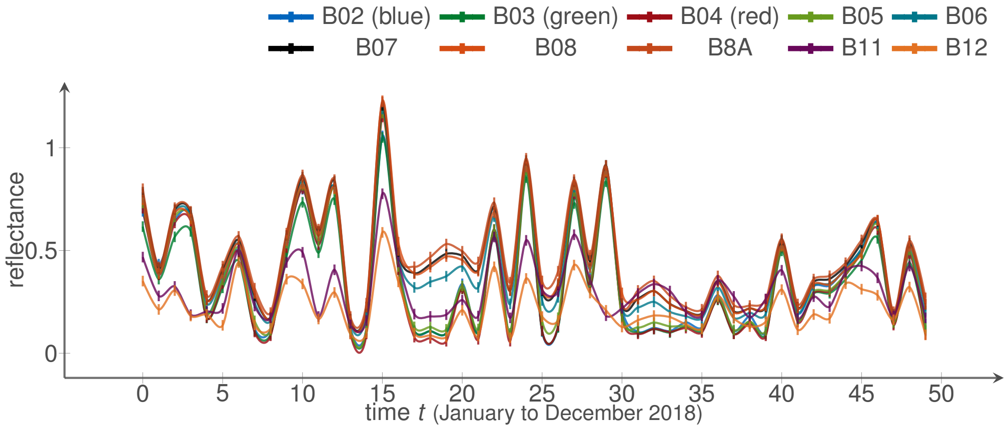
Montréal, 2018

Cloud coverage



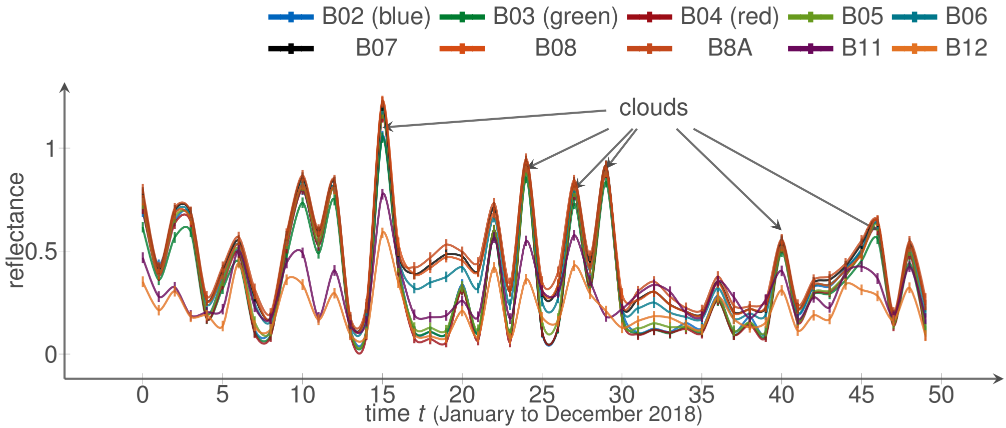
Satellite Time Series

Sentinel 2 (raw), mean-aggregated pixels of a meadow field parcel



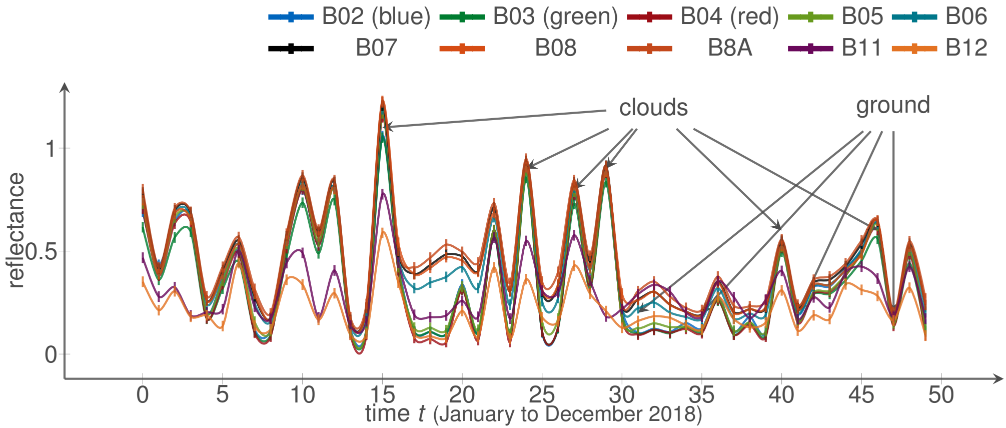
Satellite Time Series

Sentinel 2 (raw), mean-aggregated pixels of a meadow field parcel

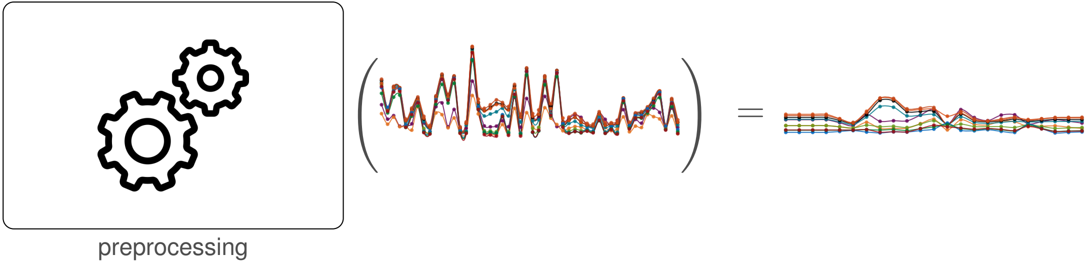


Satellite Time Series

Sentinel 2 (raw), mean-aggregated pixels of a meadow field parcel



Data Preprocessing



$f_{\theta_{\text{sel}}}(X)$ temporal selection (not considering winter period) where $\theta_{\text{sel}} = \{t_{\text{start}}, t_{\text{end}}\}$

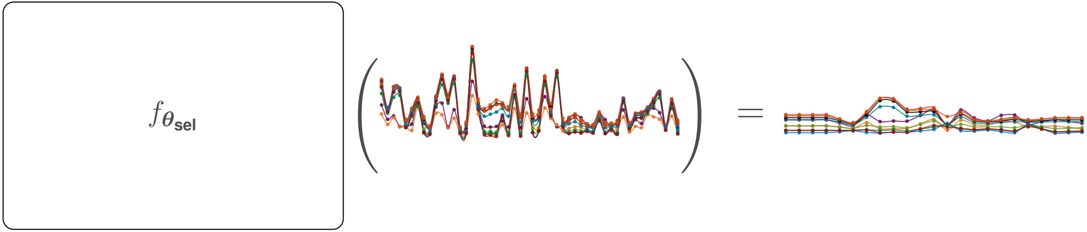
$f_{\theta_{\text{atm}}}(X)$ atmospheric correction ($X_{\text{top-of-atmosphere}} \rightarrow X_{\text{bottom-of-atmosphere}}$)

$f_{\theta_{\text{cl}}}(X)$ cloud/cloud- shadow classification (F-Mask, MAJA, CNNs, Cloud Clustering (go FDL!))

$f_{\text{int}}(X)$ temporal interpolation to generate equal sample times

$f_{\theta_{\dots}}$ **many more problem-specific chained building blocks**

Data Preprocessing



preprocessing

$f_{\boldsymbol{\theta}_{\text{sel}}}(\boldsymbol{X})$ temporal selection (not considering winter period) where $\boldsymbol{\theta}_{\text{sel}} = \{t_{\text{start}}, t_{\text{end}}\}$

$f_{\boldsymbol{\theta}_{\text{atm}}}(\boldsymbol{X})$ atmospheric correction ($\boldsymbol{X}_{\text{top-of-atmosphere}} \rightarrow \boldsymbol{X}_{\text{bottom-of-atmosphere}}$)

$f_{\boldsymbol{\theta}_{\text{cl}}}(\boldsymbol{X})$ cloud/cloud- shadow classification (F-Mask, MAJA, CNNs, Cloud Clustering (go FDL!))

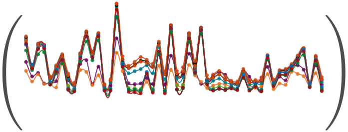
$f_{\text{int}}(\boldsymbol{X})$ temporal interpolation to generate equal sample times

$f_{\theta_{\dots}}$ **many more problem-specific chained building blocks**

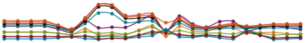
Data Preprocessing

$f_{\boldsymbol{\theta}_{\text{sel}}}\left(f_{\boldsymbol{\theta}_{\text{atm}}}\right)$

preprocessing



$=$



$f_{\boldsymbol{\theta}_{\text{sel}}}(\boldsymbol{X})$ temporal selection (not considering winter period) where $\boldsymbol{\theta}_{\text{sel}} = \{t_{\text{start}}, t_{\text{end}}\}$

$f_{\boldsymbol{\theta}_{\text{atm}}}(\boldsymbol{X})$ atmospheric correction ($\boldsymbol{X}_{\text{top-of-atmosphere}} \rightarrow \boldsymbol{X}_{\text{bottom-of-atmosphere}}$)

$f_{\theta_{\text{cl}}}(\boldsymbol{X})$ cloud/cloud- shadow classification (F-Mask, MAJA, CNNs, Cloud Clustering (go FDL!))

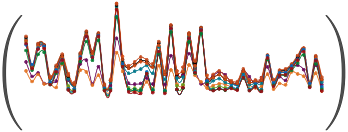
$f_{\text{int}}(\boldsymbol{X})$ temporal interpolation to generate equal sample times

$f_{\theta_{\dots}}$ **many more problem-specific chained building blocks**

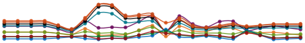
Data Preprocessing

$$f_{\theta_{\text{sel}}}\left(f_{\theta_{\text{atm}}}\left(f_{\theta_{\text{cl}}}\right)\right)$$

preprocessing



$=$



$f_{\theta_{\text{sel}}}(\boldsymbol{X})$ temporal selection (not considering winter period) where $\boldsymbol{\theta}_{\text{sel}} = \{t_{\text{start}}, t_{\text{end}}\}$

$f_{\theta_{\text{atm}}}(\boldsymbol{X})$ atmospheric correction ($\boldsymbol{X}_{\text{top-of-atmosphere}} \rightarrow \boldsymbol{X}_{\text{bottom-of-atmosphere}}$)

$f_{\theta_{\text{cl}}}(\boldsymbol{X})$ cloud/cloud- shadow classification (F-Mask, MAJA, CNNs, Cloud Clustering (go FDL!))

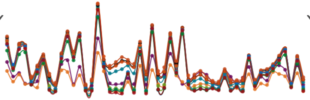
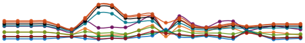
$f_{\text{int}}(\boldsymbol{X})$ temporal interpolation to generate equal sample times

$f_{\theta_{\dots}}$ **many more problem-specific chained building blocks**

Data Preprocessing

$$f_{\theta_{\text{sel}}}\left(f_{\theta_{\text{atm}}}\left(f_{\theta_{\text{cl}}}\left(f_{\text{int}}\right)\right)\right)$$

preprocessing

$\Big($  $\Big) =$ 

$f_{\theta_{\text{sel}}}(\boldsymbol{X})$ temporal selection (not considering winter period) where $\boldsymbol{\theta}_{\text{sel}} = \{t_{\text{start}}, t_{\text{end}}\}$

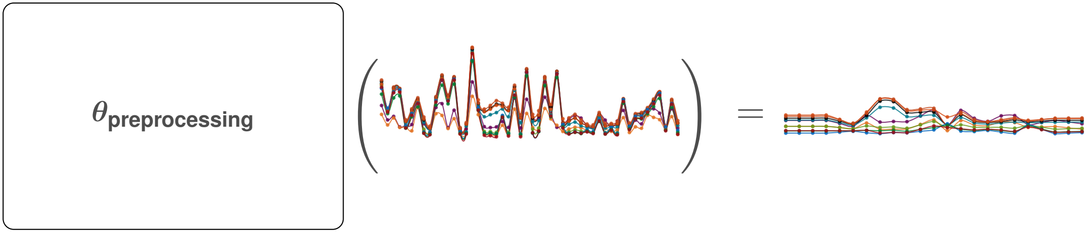
$f_{\theta_{\text{atm}}}(\boldsymbol{X})$ atmospheric correction ($\boldsymbol{X}_{\text{top-of-atmosphere}} \rightarrow \boldsymbol{X}_{\text{bottom-of-atmosphere}}$)

$f_{\theta_{\text{cl}}}(\boldsymbol{X})$ cloud/cloud- shadow classification (F-Mask, MAJA, CNNs, Cloud Clustering (go FDL!))

$f_{\text{int}}(\boldsymbol{X})$ temporal interpolation to generate equal sample times

$f_{\theta_{\ldots}}$ many more problem-specific chained building blocks

Data Preprocessing

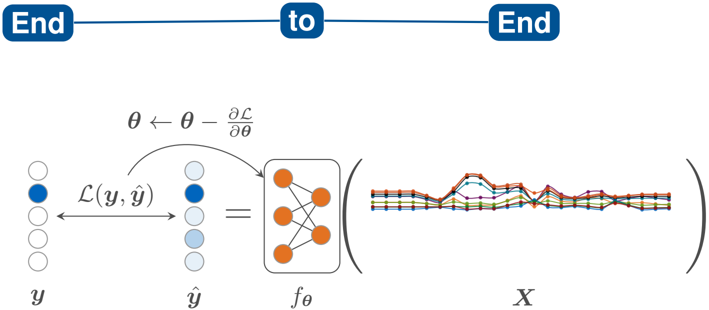


preprocessing

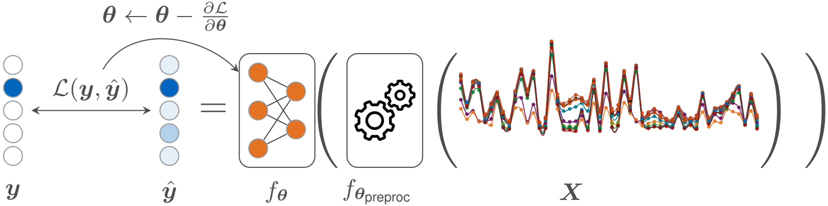
- $f_{\theta_{\text{sel}}}(\boldsymbol{X})$ temporal selection (not considering winter period) where $\boldsymbol{\theta}_{\text{sel}} = \{t_{\text{start}}, t_{\text{end}}\}$
- $f_{\theta_{\text{atm}}}(\boldsymbol{X})$ atmospheric correction ($\boldsymbol{X}_{\text{top-of-atmosphere}} \rightarrow \boldsymbol{X}_{\text{bottom-of-atmosphere}}$)
- $f_{\theta_{\text{cl}}}(\boldsymbol{X})$ cloud/cloud- shadow classification (F-Mask, MAJA, CNNs, Cloud Clustering (go FDL!))
- $f_{\text{int}}(\boldsymbol{X})$ temporal interpolation to generate equal sample times
- $f_{\theta_{\dots}}$ **many more problem-specific chained building blocks**

Deep Learning Models are trained differently...

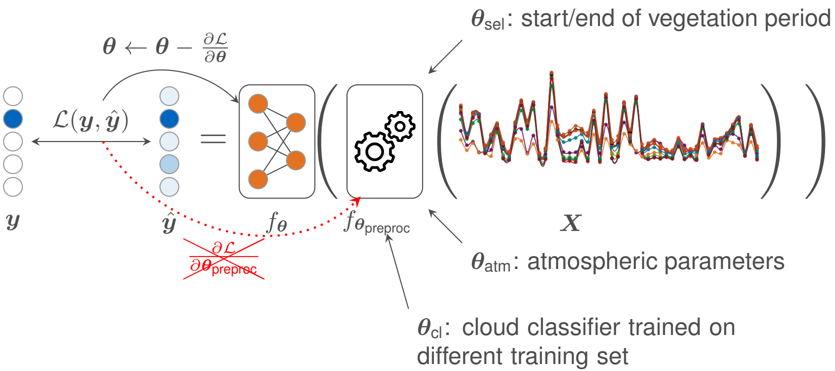
End-to-End Learning of Deep Neural Networks



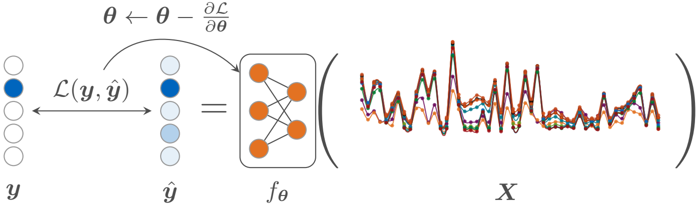
End-to-End Learning of Deep Neural Networks



End-to-End Learning of Deep Neural Networks



End-to-End Learning of Deep Neural Networks



Let’s look at some quantitative results...

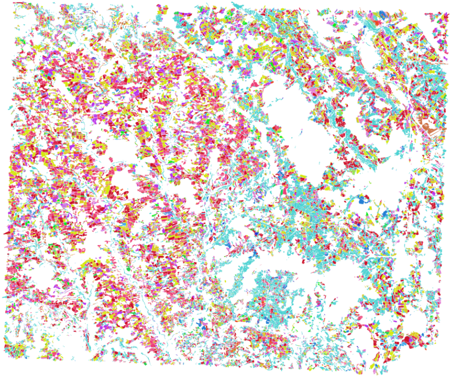
Crop Type Dataset northern Bavaria

Common project with GAFAG

crop type labels by the
Bavarian Ministry of Agriculture

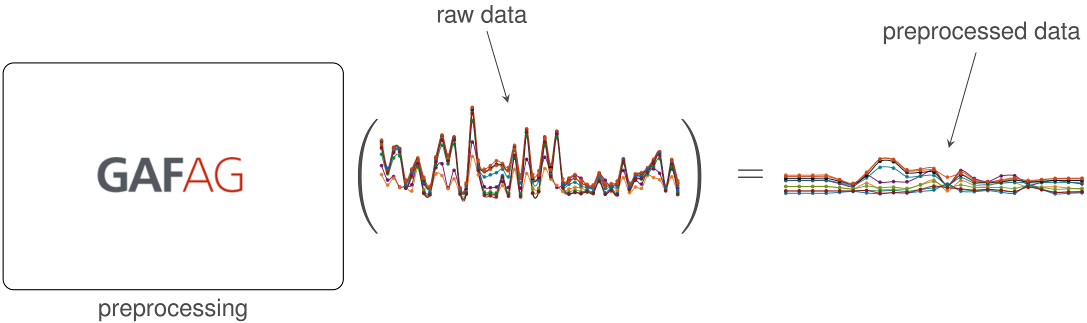
49k field **parcels** of 2018

34 crop categories



Parcels colored by crop type (40 km $\times$ 40 km)

Two Datasets: Raw and Preprocessed from the same Examples



In this case: Preprocessing Engine of GAFAG

Four state-of-the-art deep Models for Time Series Classification

	LSTM-RNN[1]	Transformer[1]	MS-ResNet[3]	TempCNN[4]
Mechanism	Recurrence	Self-Attention	Convolution	Convolution
Parameters	100k	600k	2M	433k

[1] **Hochreiter, S., & Schmidhuber, J. (1997)**. Long short-term memory. Neural computation, 9(8), 1735-1780.
[2] **Vaswani**, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. **(2017)**. Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
[3] **Wang**, F., Han, J., Zhang, S., He, X., & Huang, D. **(2018)**. Csi-net: Unified human body characterization and action recognition. arXiv preprint arXiv:1810.03064.
[4] **Pelletier, C.**, Webb, G. I., & Petitjean, F. **(2019)**. Temporal convolutional neural network for the classification of satellite image time series. Remote Sensing, 11(5), 523.

Preprocessed versus Raw Data

accuracy	RNN (LSTM)[2]	Transformer[3]	MS-ResNet[1]	TempCNN[4]
preprocessed	**.804** $\pm.0031$	.804 $\pm.0011$	**.849** $\pm.0041$	**.836** $\pm.0012$
raw	.801 $\pm.0026$	**.842** $\pm.0043$	.836 $\pm.0033$	.799 $\pm.0027$
Δ	.003 $\pm.0041$	-.038 $\pm.0045$	.013 $\pm.0055$	.038 $\pm.0029$

kappa	RNN (LSTM)[2]	Transformer[3]	MS-ResNet[1]	TempCNN[4]
preprocessed	**.759** $\pm.0037$	.759 $\pm.0017$	**.816** $\pm.0048$	**.799** $\pm.0015$
raw	.756 $\pm.0037$	**.808** $\pm.0052$	.800 $\pm.0039$	.750 $\pm.0036$
Δ	.003 $\pm.0048$	-.049 $\pm.0054$	.016 $\pm.0060$	.049 $\pm.0036$

Experiments:

mean $\pm$ standard deviation of 10 models trained from different random initialization

Findings:

remarkably similar results on preprocessed and raw data ($\Delta \leq 5\%$)

Attention

Photo by zhengtao tang on Unsplash

Self-Attention in Deep Learning

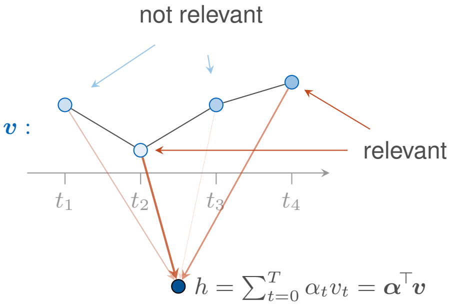
Photo by zhengtao tang on Unsplash

Attention

Given **values** $\boldsymbol{v}$ as a sequence of observations.

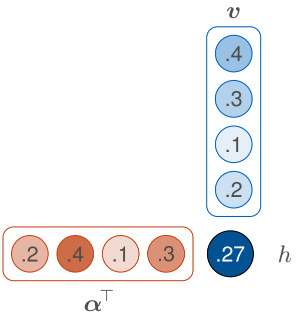
We want to **calculate a result** $\boldsymbol{h}$ based only on **classification-relevant** observations.

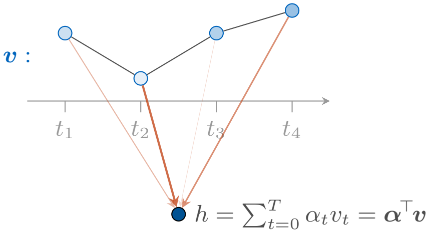
This is realized by an weighted sum over **attention scores** $\boldsymbol{\alpha}$



$$\boldsymbol{h} = \text{Attention}(\boldsymbol{\alpha}, \boldsymbol{v}) = \boldsymbol{\alpha}^\top \boldsymbol{v} = \sum_{t=0}^{T} \alpha_t v_t, \quad \boldsymbol{\alpha} \in [0,1]^{T=4}, \boldsymbol{v} \in \mathbb{R}^T$$

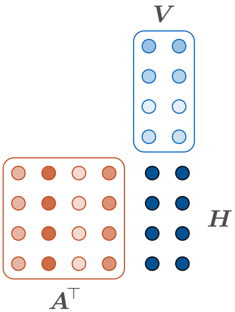
Attention

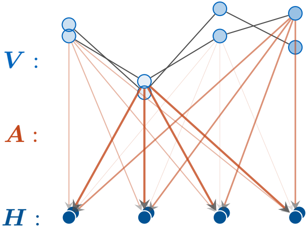




$$\boldsymbol{h} = \text{Attention}(\boldsymbol{\alpha}, \boldsymbol{v}) = \boldsymbol{\alpha}^\top \boldsymbol{v} = \sum_{t=0}^{T} \alpha_t v_t, \quad \boldsymbol{\alpha} \in [0,1]^{T=4}, \boldsymbol{v} \in \mathbb{R}^T$$

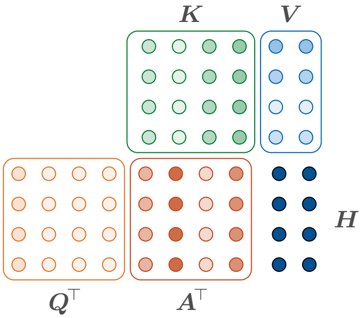
Attention

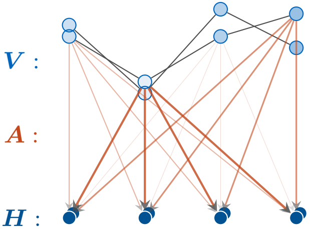




$$\boldsymbol{H} = \text{Attention}(\boldsymbol{A}, \boldsymbol{V}) = \boldsymbol{A}^\top \boldsymbol{V}, \quad \boldsymbol{A} \in [0,1]^{T_{in} \times T_{out}}, \boldsymbol{V} \in \mathbb{R}^{T \times D_v}$$

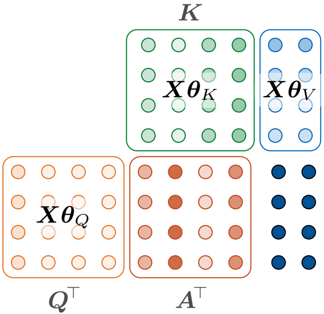
Attention

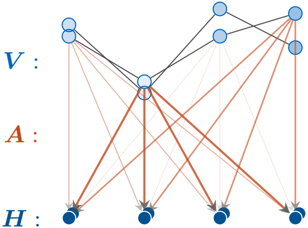




$$\text{Attention}(\boldsymbol{K}, \boldsymbol{Q}, \boldsymbol{V}) = \overbrace{\text{softmax}\left(\boldsymbol{Q}^\top \boldsymbol{K}\right)}^{\boldsymbol{A}^\top} \boldsymbol{V}, \quad \boldsymbol{V} \in \mathbb{R}^{T \times D_v}, \boldsymbol{Q}, \boldsymbol{K} \in \mathbb{R}^{D_k \times T}, \boldsymbol{A} \in \mathbb{R}^{T_{in} \times T_{out}}$$

Self-Attention

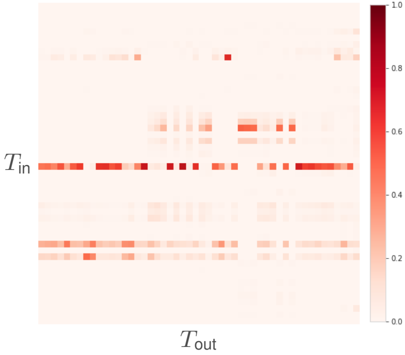




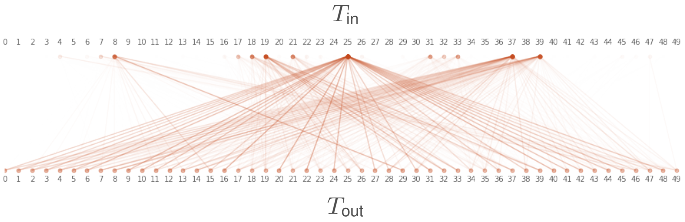
$$\text{Self-Attention}_{\boldsymbol{\theta}}(\boldsymbol{X}) = \text{Attention}(\underbrace{\boldsymbol{X}\boldsymbol{\theta}_K}_{\boldsymbol{K}}, \underbrace{\boldsymbol{X}\boldsymbol{\theta}_Q}_{\boldsymbol{Q}}, \underbrace{\boldsymbol{X}\boldsymbol{\theta}_V}_{\boldsymbol{V}}) = \text{softmax}\left(\left(\boldsymbol{X}\boldsymbol{\theta}_Q\right)\left(\boldsymbol{X}\boldsymbol{\theta}_K\right)\right)\left(\boldsymbol{X}\boldsymbol{\theta}_V\right)$$

Visualizing Attention Scores of a Pretrained Transformer Model

As (adjacency) Matrix A



As bipartite graph

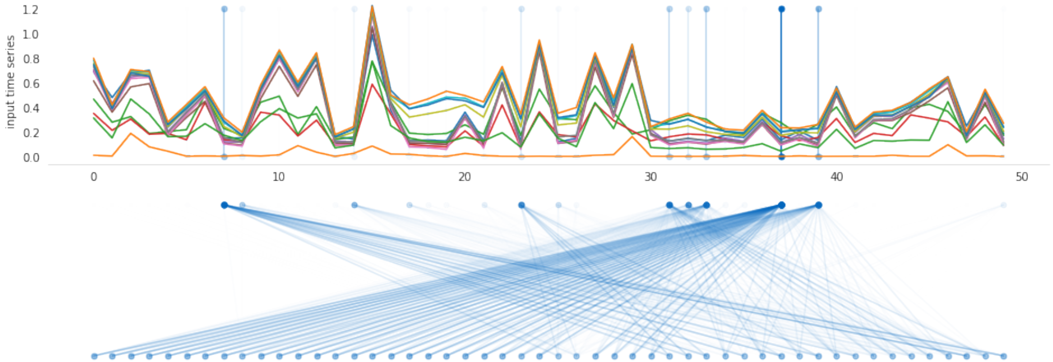


Attention Scores in Context of Input Time Series

Head 1

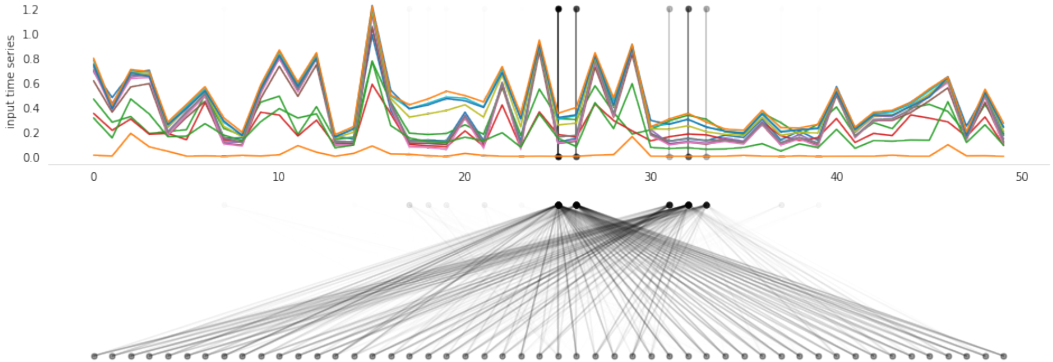
Attention Scores in Context of Input Time Series

Head 2



Attention Scores in Context of Input Time Series

Head 3



Let’s summarize...

Summary

What did we look at?

Summary

What did we look at?

end-to-end learning in combination with **preprocessing**

Summary

What did we look at?

end-to-end learning in combination with **preprocessing**

quantitative **results** on models with **raw** and **preprocessed** data

Summary

What did we look at?

end-to-end learning in combination with **preprocessing**

quantitative **results** on models with **raw** and **preprocessed** data

a qualitative example on the **self-attention** mechanism

Conclusion

What were the outcomes of these experiments?

Conclusion

What were the outcomes of these experiments?

classification results on raw and preprocessed data were **remarkably similar**

Conclusion

What were the outcomes of these experiments?

classification results on raw and preprocessed data were **remarkably similar**

do we **need** extensive **preprocessing** for deep learning models on time series data?

Conclusion

How did deep learning models get robust to noise (e.g., clouds)?

Conclusion

How did deep learning models get robust to noise (e.g., clouds)?

we saw how **self-attention** is used to **focus** on cloud-free observations

Conclusion

How did deep learning models get robust to noise (e.g., clouds)?

we saw how **self-attention** is used to **focus** on cloud-free observations

gates in recurrent networks work similar (see previous work)[1,2]

[1] Rußwurm, M., & Körner, M. (**2018**). Multi-temporal land cover classification with sequential recurrent encoders. ISPRS **International Journal of Geo-Information**, 7(4), 129.

[2] Rußwurm, M., & Körner, M. (**2018**). Convolutional LSTMs for Cloud-Robust Segmentation of Remote Sensing Imagery. **NeurIPS2018 Workshop on Spatiotemporal Modeling**

Thank you

Marc Rußwurm & Marco Körner
TUM Chair of Remote Sensing Technology
Computer Vision Research Group

clone the repo to the attention experiments!
github.com/marccoru/phiweek19 Open in Colab

TUM Chair:
www.bgu.tum.de/en/lmf/vision/

our official account:
github.com/tum-lmf

in cooperation with GAFAG

follow **@marccoru** for updates.

github.com/marccoru for code.

and **marccoru.github.io**

Bonus slides: Results on Recurrent Neural Networks

see Recurrence.ipynb in the repository

Input feature importance analysis through gradient backpropagation

if we change the input $\boldsymbol{X}$...

$$\frac{\partial \max(\hat{\boldsymbol{y}})}{\partial \boldsymbol{X}}$$

Input feature importance analysis through gradient backpropagation

if we change the input $\boldsymbol{X}$...

$$\frac{\partial \max(\hat{\boldsymbol{y}})}{\partial \boldsymbol{X}}$$

... how would the highest predicted score $\max(\hat{\boldsymbol{y}})$ change?

Can be implemented in four lines of code

```
In [18]: x_ = torch.autograd.Variable(x[None,:,:], requires_grad=True)
         logprobabilities = model.forward(x_)
         logprobabilities.exp().max().backward()
         dydx = x_.grad
```

Gradients from $\max(\hat{\boldsymbol{y}})$ to $\boldsymbol{X}$