Deep Ensemble predicts Exoplanets' Atmospheres Composition

Ariel Data Challenge

Ondřej Podsztavek

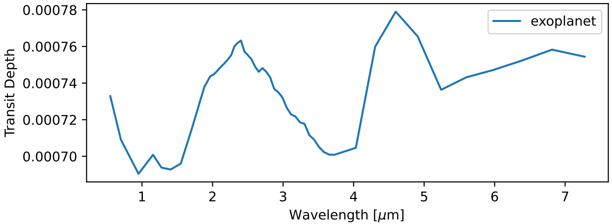
Faculty of Information Technology
Czech Technical University in Prague

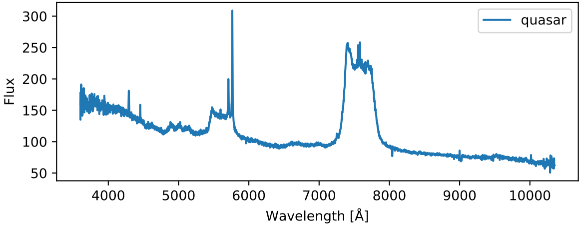
NeurIPS 2022

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Motivation from Personal Research Perspective

- Research of **predictive uncertainty**.
- Spectra of exoplanetary atmospheres and quasars are similar.





Training Set

Training set

- Includes: annotated spectra with auxiliary data.
- Excludes: unannotated data.

My task's simplification

Predict $\mathcal{N}(\mu_{ij}, \sigma_{ij}^2)$ where

- μ_{ij} are weighted means and σ_{ij}^2 are weighted variances,
- i indexes exoplanets and j indexes the six targets.

The Simplification

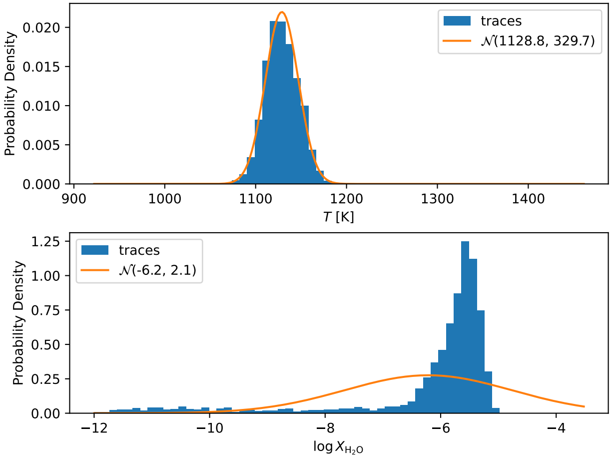


Figure: Exoplanet #2313 traces of T and $\log X_{\mathrm{H_2O}}$ with fitted $\mathcal{N}(\mu_{ij}, \sigma_{ij}^2)$.

Data Preparation

Auxiliary data

Standardisation of **each feature** (star distance, etc.).

Spectra scaling

Standardisation of **each spectrum's transit depths**.

Spectra Scaling's Effect

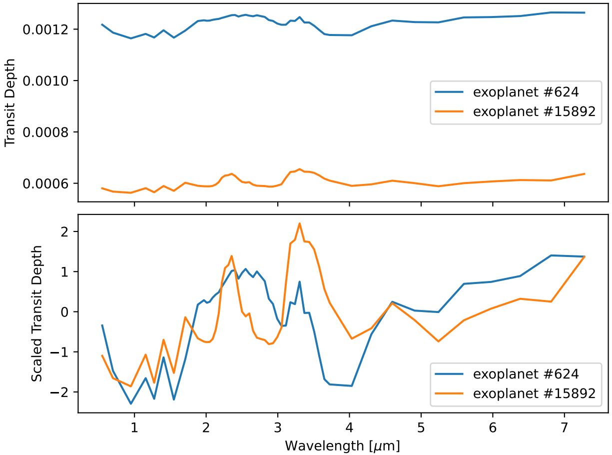


Figure: Two spectra before (top) and after (bottom) spectra scaling.

My Convolutional Neural Network (CNN)

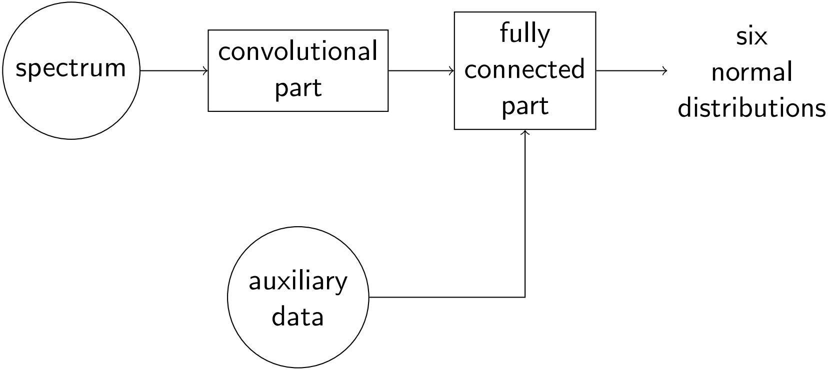


Figure: Schema of the CNN's prediction process.

CNN's Output: Six Normal Distributions

1. $\mathcal{N}(\hat{\mu}_{i1}, \hat{\sigma}^2_{i1})$ for T
2. $\mathcal{N}(\hat{\mu}_{i2}, \hat{\sigma}^2_{i2})$ for $\log X_{\mathrm{H_2O}}$
3. $\mathcal{N}(\hat{\mu}_{i3}, \hat{\sigma}^2_{i3})$ for $\log X_{\mathrm{CO_2}}$
4. $\mathcal{N}(\hat{\mu}_{i4}, \hat{\sigma}^2_{i4})$ for $\log X_{\mathrm{CH_4}}$
5. $\mathcal{N}(\hat{\mu}_{i5}, \hat{\sigma}^2_{i5})$ for $\log X_{\mathrm{CO}}$
6. $\mathcal{N}(\hat{\mu}_{i6}, \hat{\sigma}^2_{i6})$ for $\log X_{\mathrm{NH_3}}$

- Output units of $\hat{\mu}_{ij}$ are identities.
- Output units of $\hat{\sigma}^2_{ij}$: $\mathrm{softplus}(z) = \log(1 + e^z)$.

Loss Function Variants

- point value to distribution
  - negative log-likelihood (NLL)
  - continuous ranked probability score (CRPS)
- distribution to distribution
  - Kullback–Leibler (KL) divergence
  - Wasserstein distance

Final Training

- KL divergence worked best.
- The whole training set is used to train final models!

Deep Ensembles[1]

- Deep ensemble of 20 CNNs.
- Each CNN predicts six normal distributions.
- Generate 250 samples from predicted normal distributions.
- **Regular track**: Output is the 5000 samples in total.
- **Light track**: Compute quartiles of those 5000 samples.

[1] Lakshminarayanan, B., Pritzel, A., Blundell, C., 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles, in: Advances in Neural Information Processing Systems 30.

Problem: Simulated Data

- Performance will be worse on **real** data.
- Distributions of simulated and real data (likely) differ.
- Machine learning assume that data are **i.i.d.**

Active Domain Adaptation (ADA)

- ADA mitigates a difference between **source** and **target** data.

$$\begin{tikzcd}
\text{simulated spectra} \arrow[r] & \text{deep ensemble} \arrow[r] & \text{real spectra} \arrow[d] \\
& \text{annotated real spectra} \arrow[u] & \text{slow standard method} \arrow[l]
\end{tikzcd}$$

Conclusion

If you are uncertain what to do, use an uncertain deep ensemble:

- Deep ensembles are powerful at estimating **predictive uncertainty**.
- There is potential for **active domain adaptation**.

Code

See `https://github.com/podondra/ariel-data-challenge`.

Thanks

Thank the organisers for organising this challenge.

Acknowledgement

I want to acknowledge the support of my supervisor **prof. P. Tvrdík**, my co-supervisor **dr. P. Škoda**, **dr. K. Polsterer**, the European Regional Development Fund project "Research Center for Informatics" (No. CZ.02.1.01/0.0/0.0/16_019/0000765), and the Grant Agency of the Czech Technical University in Prague (No. SGS20/212/OHK3/3T/18).

CNN's Architecture: Modification of VGG Net-A[2]

Type of Layer	Hyperparameters
convolutional	8 kernels
max pooling	
convolutional	16 kernels
max pooling	
convolutional	32 kernels
convolutional	32 kernels
max pooling	
convolutional	64 kernels
convolutional	64 kernels
max pooling	
$6 \times$ fully connected	1024 neurons
fully connected	12 neurons

[2] Simonyan, K., Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

Loss Function Variants

- negative log-likelihood (NLL):

$$\ln \hat{\sigma}_{ij}^2 + \frac{(\mu_{ij} - \hat{\mu}_{ij})^2}{\hat{\sigma}_{ij}^2}$$

- continuous ranked probability score (CRPS):

$$\hat{\sigma}_{ij} \left\{ \frac{\mu_{ij} - \hat{\mu}_{ij}}{\hat{\sigma}_{ij}} \left[2\Phi \left(\frac{\mu_{ij} - \hat{\mu}_{ij}}{\hat{\sigma}_{ij}} \right) - 1 \right] + 2\varphi \left(\frac{\mu_{ij} - \hat{\mu}_{ij}}{\hat{\sigma}_{ij}} \right) - \frac{1}{\sqrt{\pi}} \right\}$$

- Kullback–Leibler (KL) divergence:

$$\ln \frac{\hat{\sigma}_{ij}^2}{\sigma_{ij}^2} + \frac{\sigma_{ij}^2 + (\hat{\mu}_{ij} - \mu_{ij})^2}{\hat{\sigma}_{ij}^2}$$

- Wasserstein distance:

$$(\hat{\mu}_{ij} - \mu_{ij})^2 + (\hat{\sigma}_{ij} - \sigma_{ij})^2$$