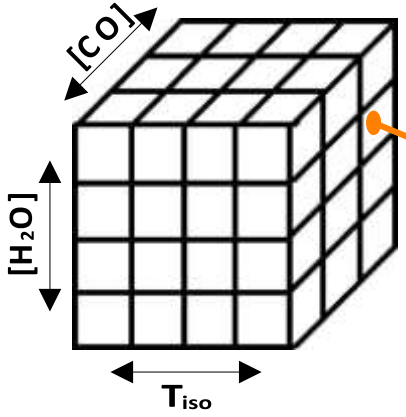
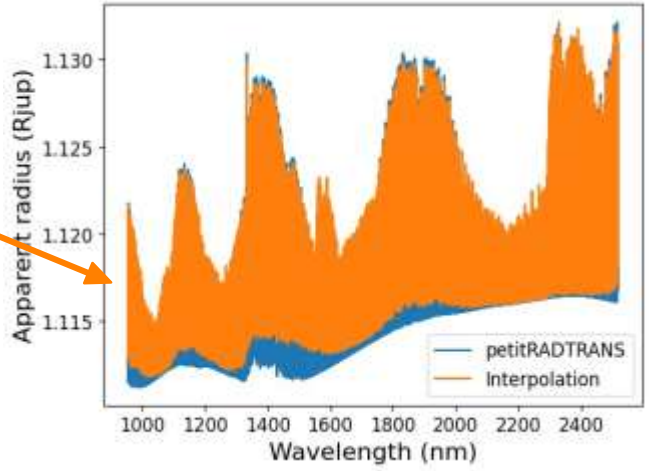
Why not use interpolation grids ?

Precompute a grid of models and interpolate :





petitRADTRANS: ~1-10s

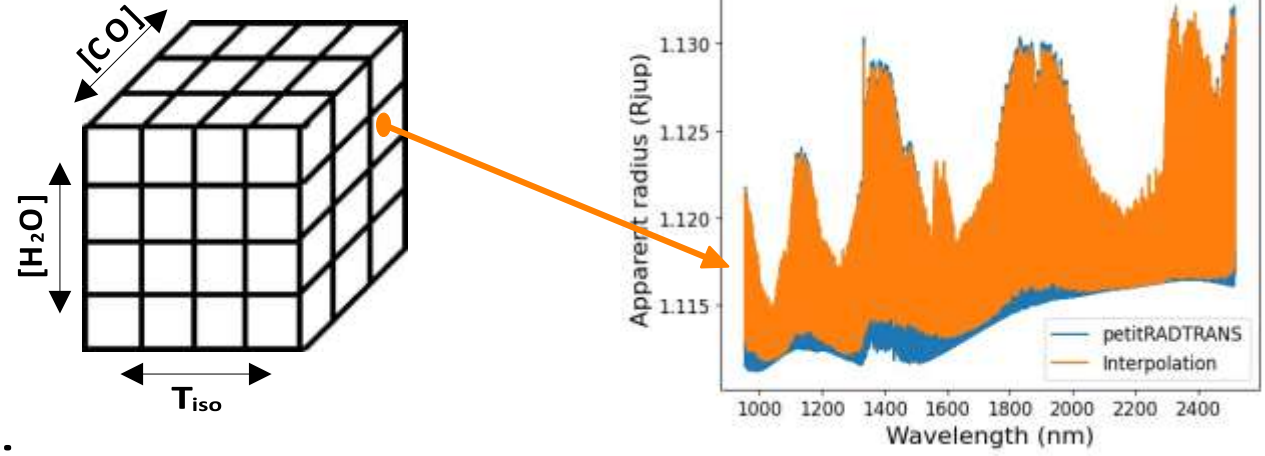
Interpolation : ~10-100ms

BUT :

- High-resolution spectra are **heavy** (**~1Mb** per model)
- Sampling **10 points in 5D** (T_{iso}, [H2O], [CO], [CH4], P_{clouds}) 10→ ^{5} x 1Mb ~ **100Gb** (storage, RAM…)

Why not use interpolation grids ?

Precompute a grid of models and interpolate :



petitRADTRANS: ~1-10s

Interpolation : ~10-100ms

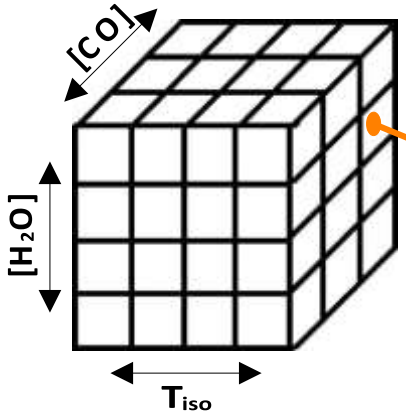
BUT :

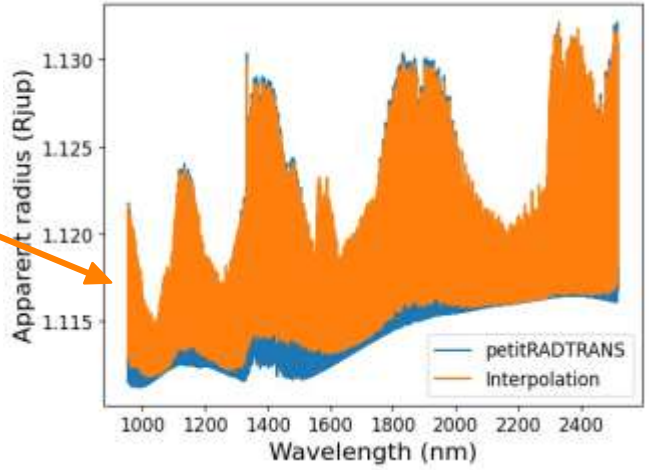
- High-resolution spectra are **heavy** (**~1Mb** per model)
- Sampling **10 points in 5D** (T_{iso}, [H2O], [CO], [CH4], P_{clouds}) 10→ ^{5} x 1Mb ~ **100Gb** (storage, RAM…) •Number of **species** and **parametrized profile** are **strongly limited**...

Why not use interpolation

- Need to compute and store **one grid per target** and **set of parameters**

Precompute a grid of models and interpolate :





petitRADTRANS: ~1-10s

Interpolation : ~10-100ms

BUT :

- High-resolution spectra are **heavy** (**~1Mb** per model)

Why not use interpolation

- Sampling **10 points in 5D** (T_{iso}, [H2O], [CO], [CH4], P_{clouds}) 10→ ^{5} x 1Mb ~ **100Gb** (storage, RAM…) •Number of **species** and **parametrized profile** are **strongly limited**... •Need to compute and store **one grid per target** and **set of parameters** • Radiatif transfer is **not linear**...

(T_{iso}, [H_2O], [CO], [CH_4], P_{clouds})

petitRADTRANS

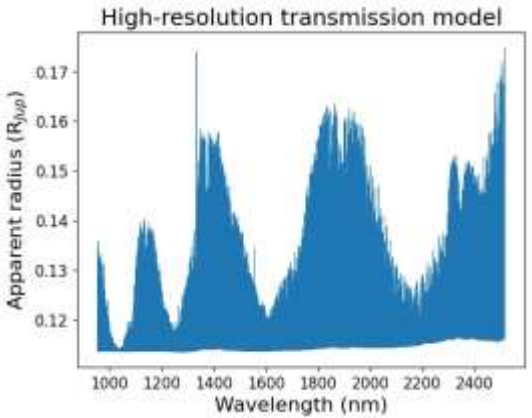
Expensive line-by-line radiatif transfer

(T_{iso}, [H_2O], [CO], [CH_4], P_{clouds})

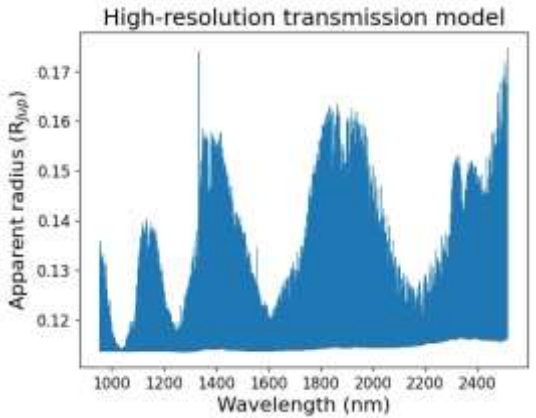
Neural Network

Approximate complex physical problem with thousands of simple individual operations.

~1 - 10s



~10 - 100ms



Advantages:

- **FAST**
- Train on planet parameters (R_p, M_P, …) to get **a general model for any planet**.
- One general model : **a few Mb** !
  - **Easy to share**
  - **RAM saving** for parallel computation

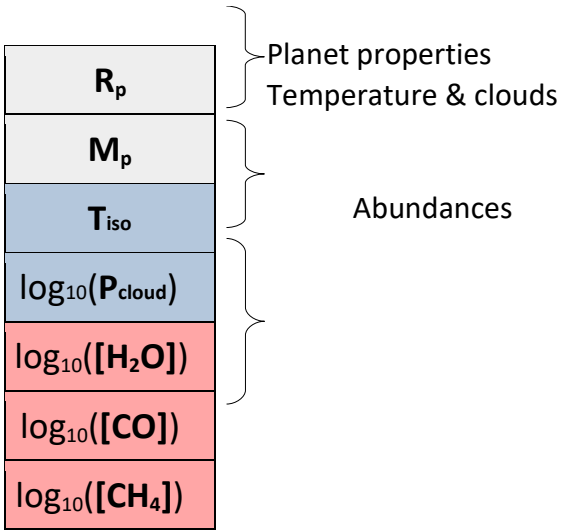
- Neural Networks are **differentiable**
  - Could allow for HMC/NUTS methods… **BUT :**
- Need quality **training data** (**a lot!**)

Generated **two datasets** with petitRADTRANS in **transmission** mode.

Generating the datasets

Generated **two datasets** with petitRADTRANS in **transmission** mode.

- **‘Easy mode’ :**
  - **50K** spectra (250k bins) **YJHK bands** (0,9 - 2,5 µm) **7 parameters** :



Generating the datasets

Generated **two datasets** with petitRADTRANS in **transmission** mode.

Defining the network structure

- **'Easy mode' :** tra (250k bins)
  - **50K** (0,9 - 2,5 µm)
  - **YJHK bands** ters :
  - **7 param**

R_p	Planet properties
M_p	Temperature & clouds
T_{iso}	
$\log_{10}(P_{cloud})$	Abundances
$\log_{10}([H_2O])$	
$\log_{10}([CO])$	
$\log_{10}([CH_4])$	

- **'Hard mode' :**
  - **100K** spectra (500k bins)
  - **VIS+YJHK** (0,3 - 2,5 µm)
  - **33 parameters :**

R_p	NH_3	Ca	Li	TiO
M_p	CO_2	Ca^+	Mg	V
T_{iso}	H_2S	Cr	Mg^+	V_+
P_{cloud}	HCN	Fe	Na	VO
H_2O	C_2H_2	FeH	Si	Y
CO	OH	Fe^+	SiO	
CH_4	Al	K	Ti	

Defining the network structure

Naïve approach : train it to directly predict high-resolution spectra...

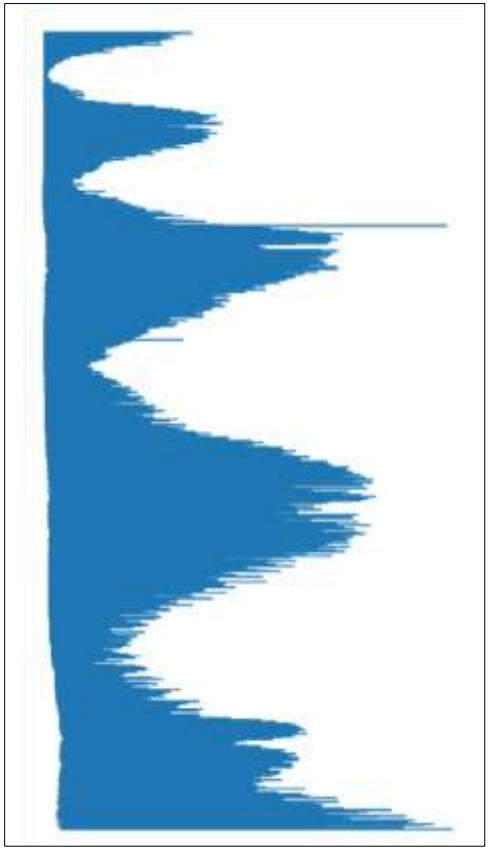
R_p
M_p
T_{iso}
$\log_{10}(P_{cloud})$
$\log_{10}([H_2O])$
$\log_{10}([CO])$
$\log_{10}([CH_4])$

Defining the network structure

Outputs

Defining the network structure

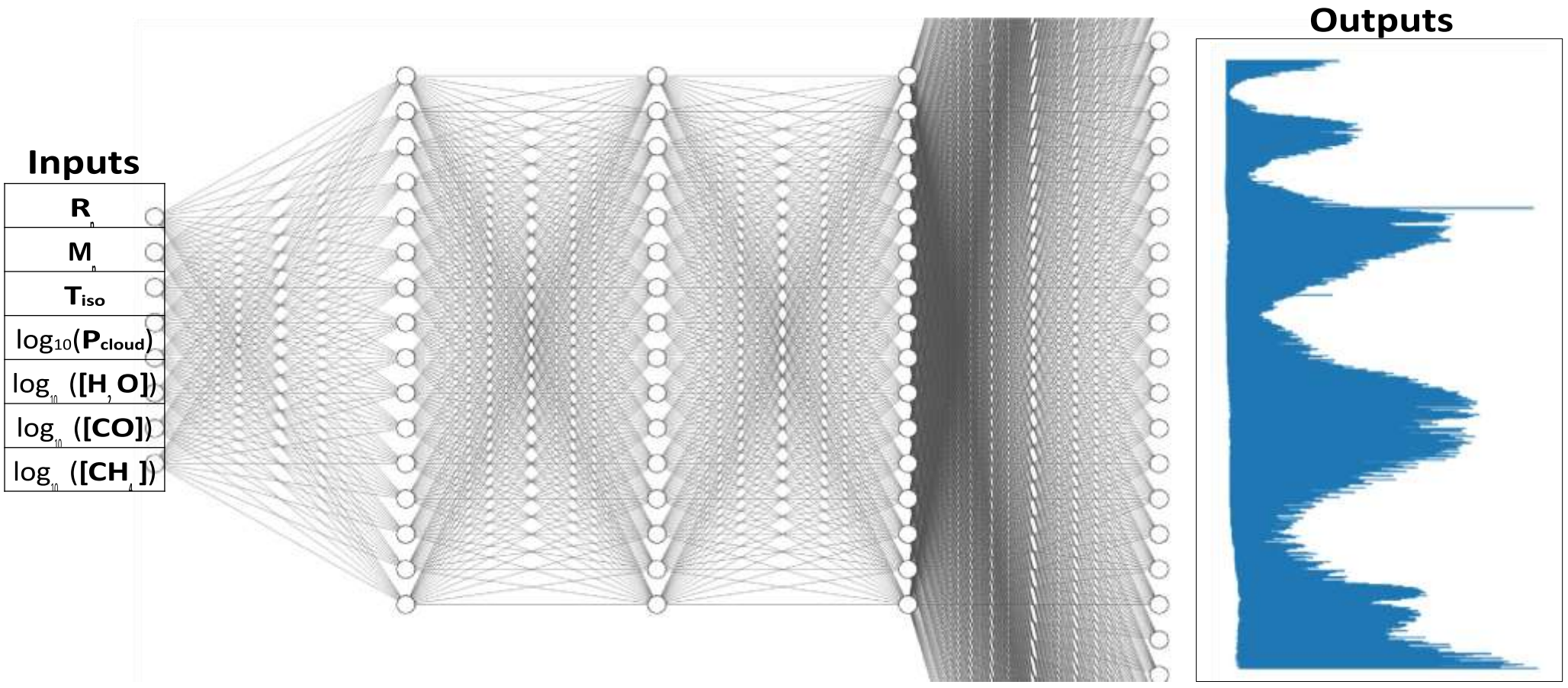
Inputs



Defining the network structure

Naïve approach : train it to directly predict high-resolution spectra...

Defining the network structure



Defining the network structure

Naïve approach : train it to directly predict high-resolution spectra...

Defining the network structure

Outputs