Machine Learning Optimization of Photometric Redshift

Kate Sautel — Senior Talk 1

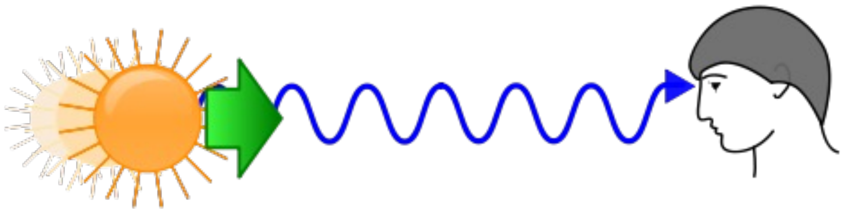
October 25, 2023

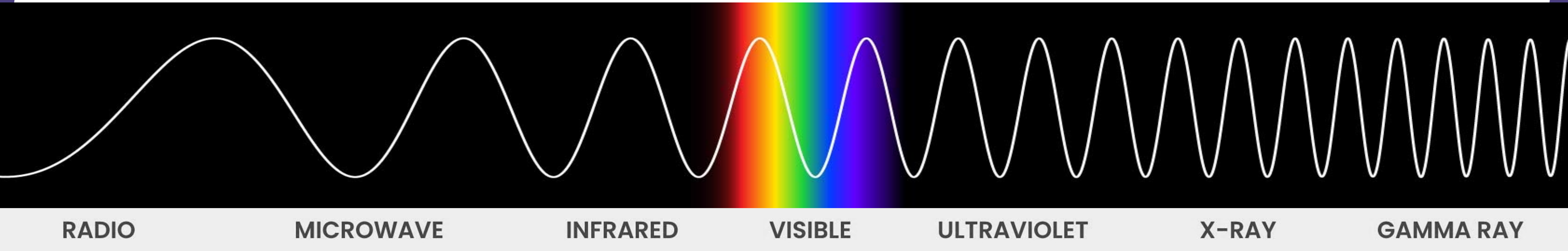
Overview

- Redshift
- Photometric redshift
- Machine learning
  - Neural networks
  - PyTorch
- My project

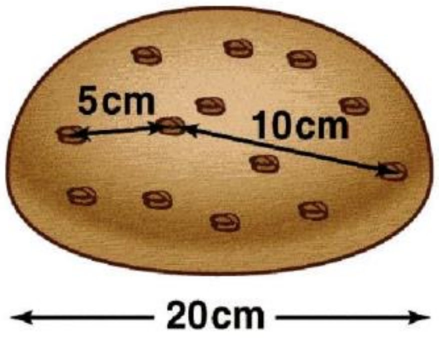
Redshift

- Redshift/blueshift relevant in astrophysics and cosmology
- Describes relative motion of celestial objects
- Due to Doppler, wavelength of light emitted from moving object stretches (toward red side of electromagnetic spectrum)



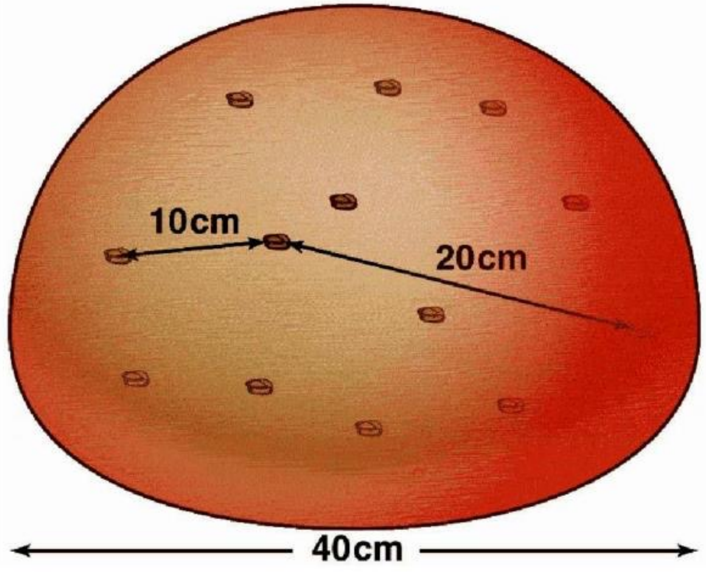


Cosmological Redshift

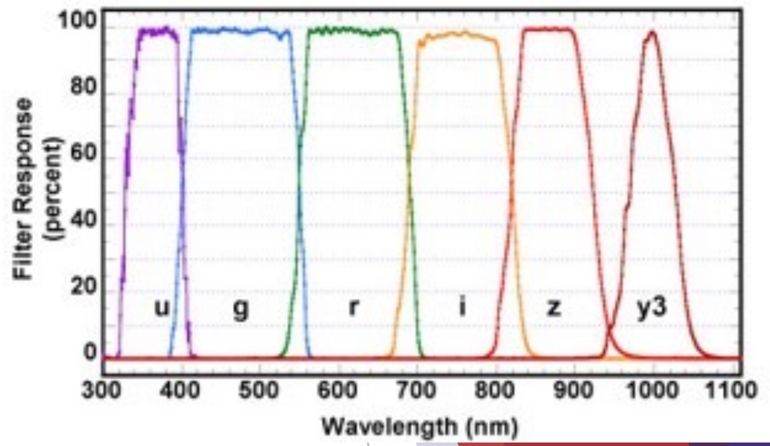


- Measure redshift ↔ distance from observer
- Hubble's law: speed ∝ distance
  - $v = H_0 D$
- Redshift definition:
  - $1 + Z \stackrel{\text{def}}{=} \dfrac{f_{emitted}}{f_{measured}} = \dfrac{\lambda_{measured}}{\lambda_{emitted}}$

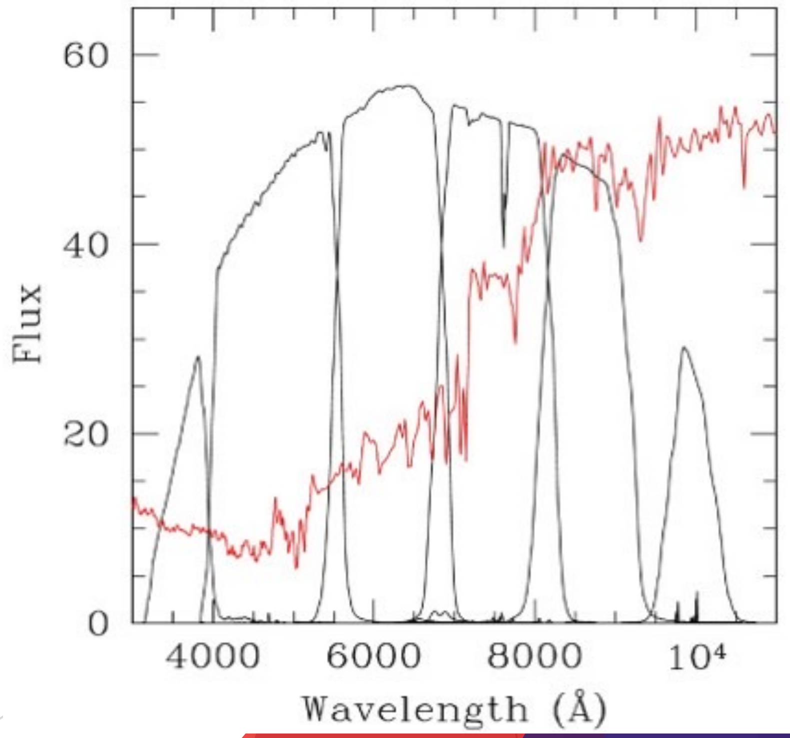
Z = redshift



Photometric Redshift & Photometric Bands



- Photometry: measure intensity (flux) of light emitted by object viewed through different filters
  - Photometer converts light into electric current through photoelectric effect



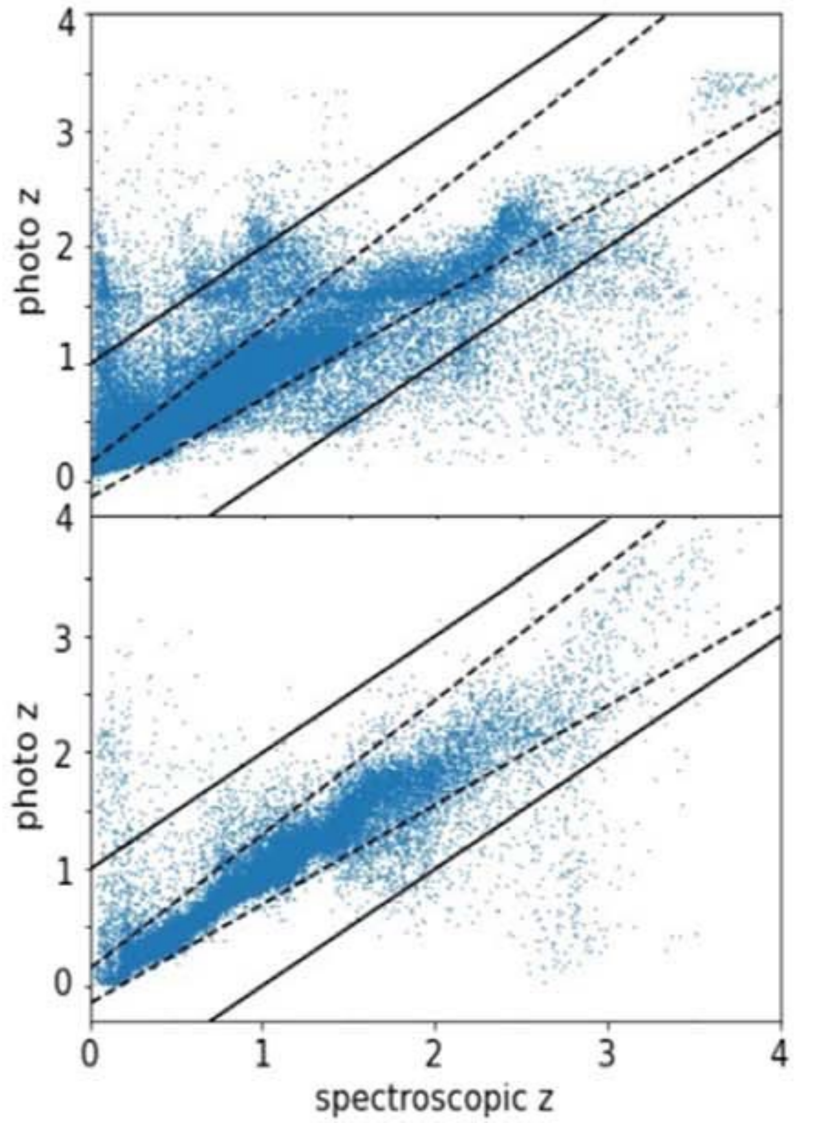
Redshift changes → Spectrum shift (in f or λ) → Flux in photo bands changes

Photometric vs. Spectroscopic Redshift

- **Photometric:** estimation, determined through ML
  - Use known redshifts of galaxies to train/predict unknown redshifts, given parameters
- **Spectroscopic:** actual measurement of redshift
  - Observe frequency/wavelength of spectral lines
- Large-sky surveys (LSST: 100,000,000 + galaxies surveyed, ~500 petabytes of data)
  - Study galaxies statistically instead of for individual properties
  - Photometric > spectroscopic

Accuracy of Photometric Redshifts

- Run tests with galaxies of known (spectroscopic) redshifts
- Classify estimations into 3 categories:
  - Non-outlier (NO)
  - Outlier
  - Catastrophic outlier (CO)
- NOT outliers in traditional/statistical sense
  - Specifically motivated by what would affect scientific analyses and conclusions



COs and NOs

- Outlier:

$$O: \frac{|Z_{phot} - Z_{spec}|}{1 + Z_{spec}} > 0.15$$

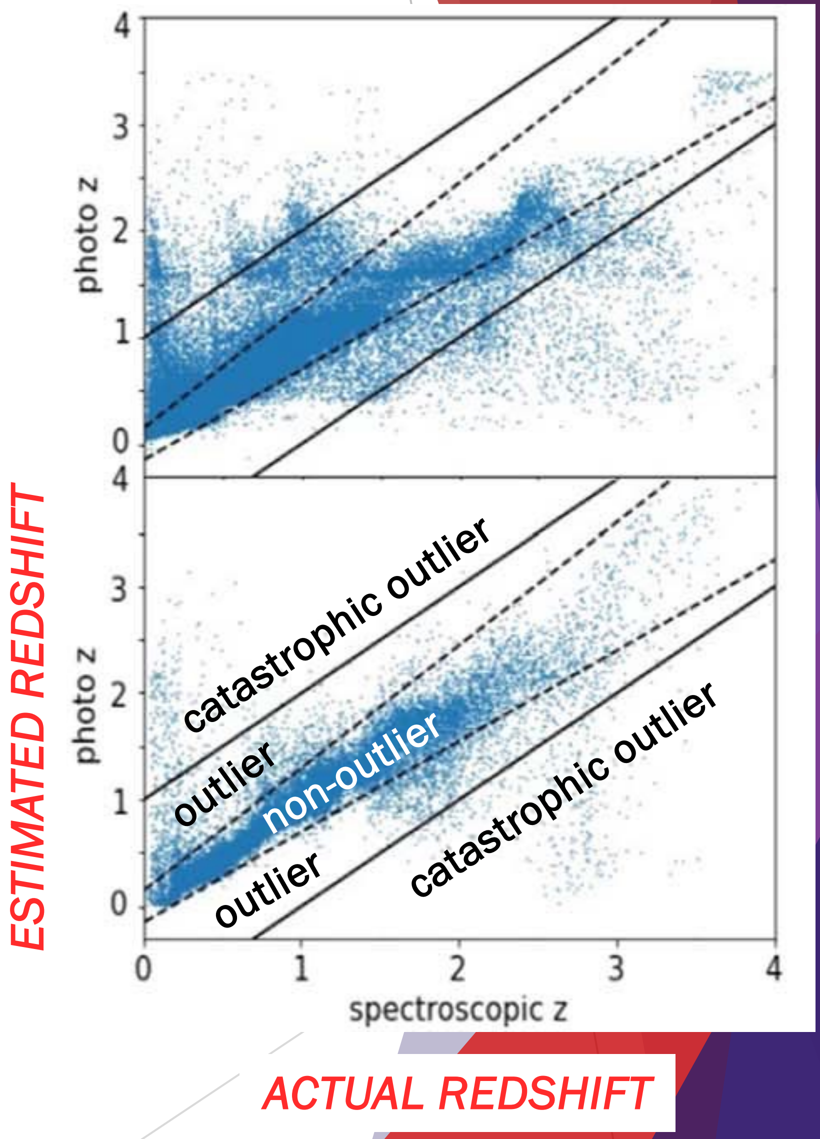
e.g. Hildebrandt, H. et al. 2010, *A&A*, 523, 832

- Catastrophic outlier:

$$O_c: |Z_{phot} - Z_{spec}| > 1.0$$

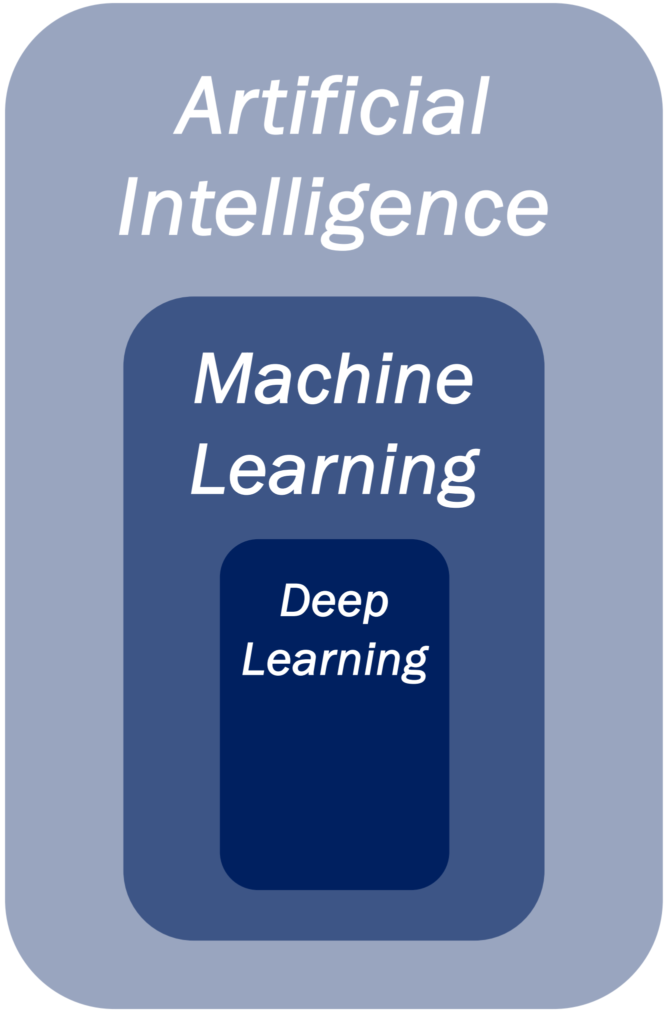
Bernstein, G. & Huterer, D. 2010, *MNRAS*, 401, 1399

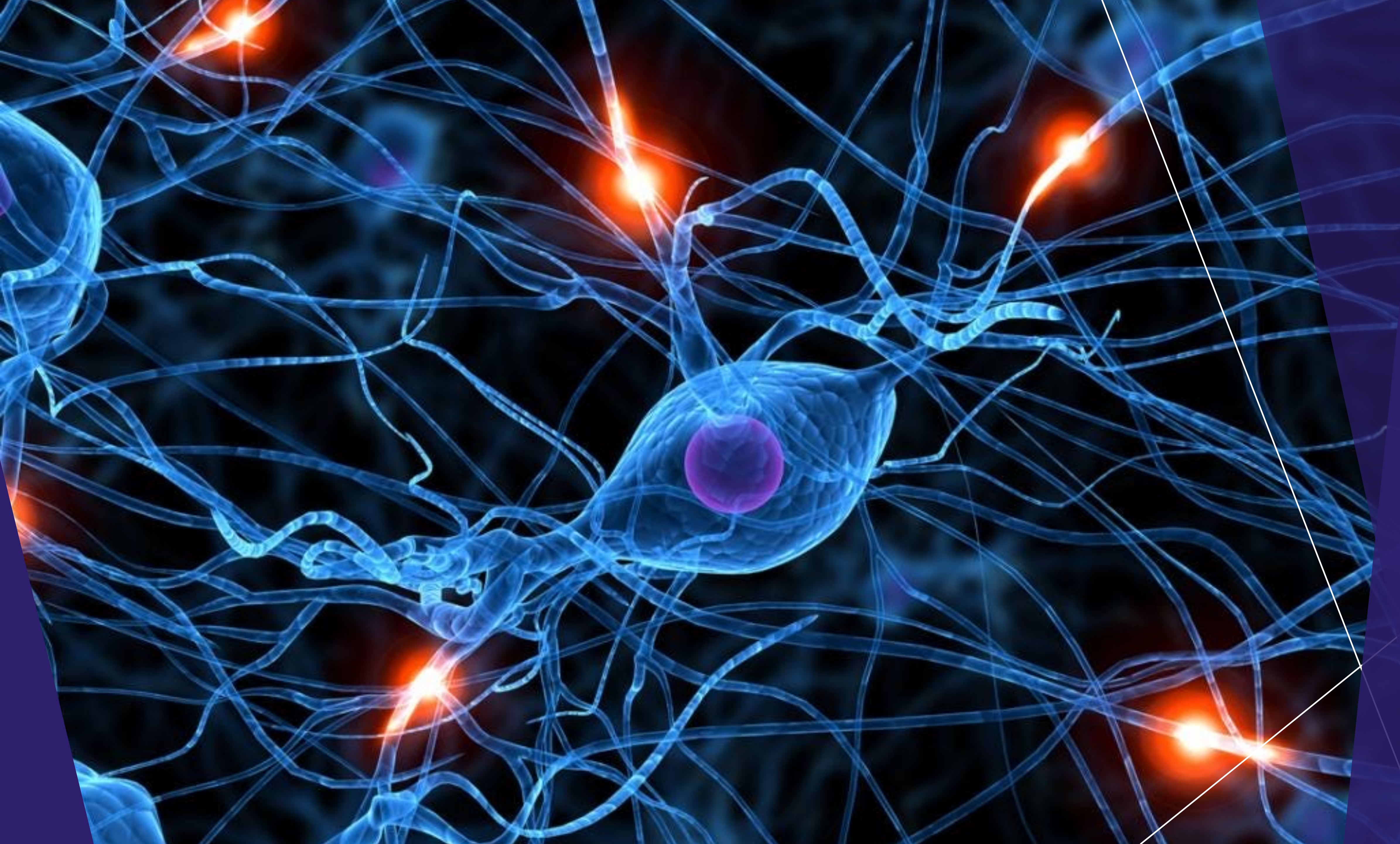
Z_{phot} = photometric redshift, Z_{spec} = spectroscopic redshift

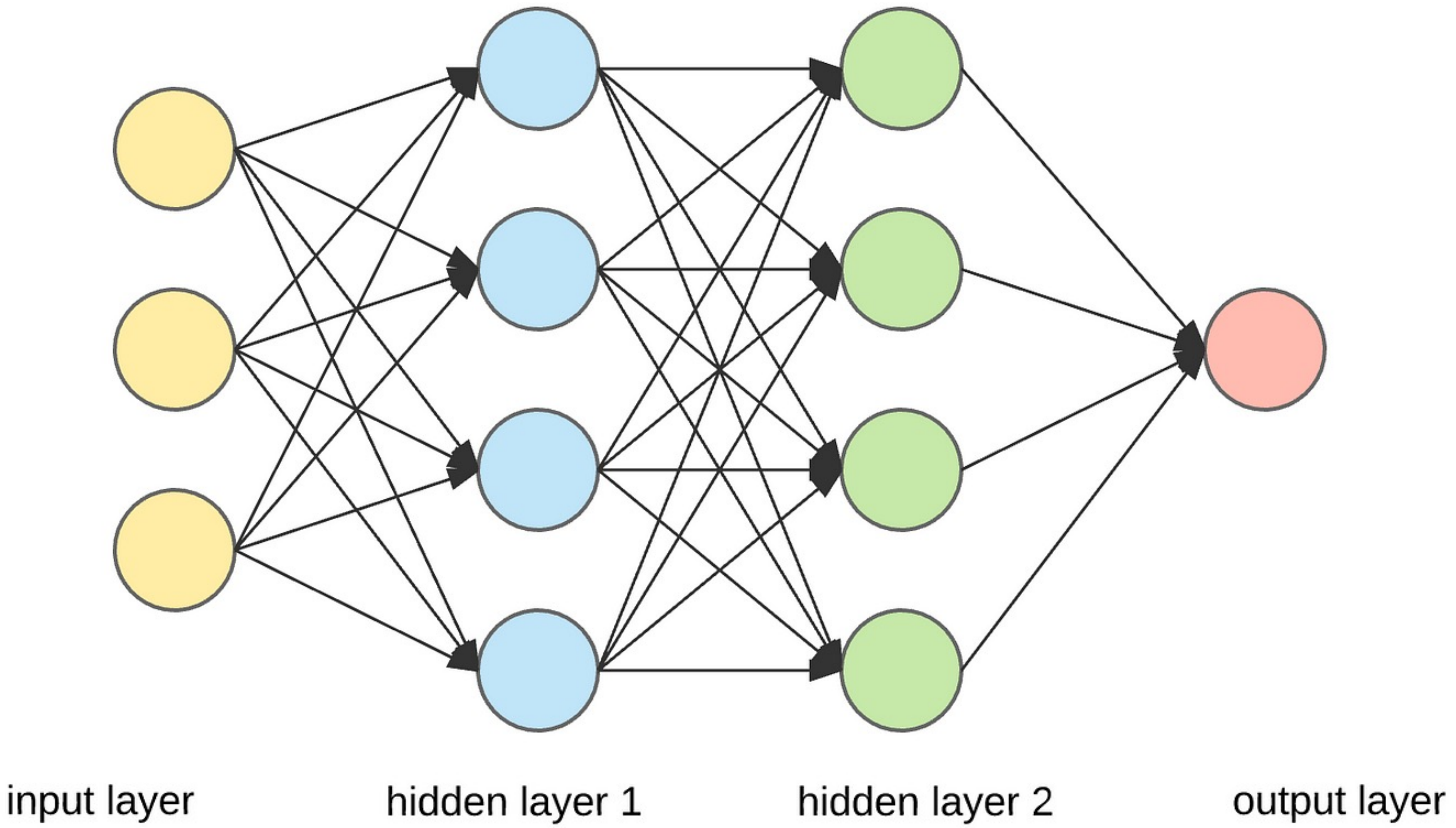


Intro to Machine Learning

- Use large quantities of data to train machine to make simple correlations
  - Avoid *explicitly* coding for individual circumstances
  - Handwriting
  - FaceID
- NNs are often backbone/structure
  - Most common






input layer
hidden layer 1
hidden layer 2
output layer

Input neurons: process/analyze/categorize input data and pass onto next layer

Hidden layer(s): further process input data or data from other hidden layers, weights attributed to neurons

Output layer: gives final "conclusion" of processing (0 or 1 or more complex)

Training

- Training involves updating the weights attributed with neurons in the hidden layers
- Weights updated iteratively
  - Updated automatically, given learning rate of model
  - Objective is to minimize loss function
  - Loss function: difference between expected output and actual output of model

Broad NN Categories

- Feedforward NN
  - Unidirectional
  - Neurons can only transmit info to neurons in next layer
- Recurrent NN
  - Bidirectional
  - Neurons can transmit info to other neurons in the same layer
  - Complicated

Data Dictionary

- Both publicly available
- Both 5-band spectroscopic (actual) measurements
- HSC: 286,401
- COSMOS: 58,619

HSC Data

	A	B	C	D	E	F
1	u_mag	g_mag	r_mag	i_mag	zed_mag	z
2	25.426	25.567	25.2166	24.5832	24.0859	0.9966
3	26.0628	26.0293	25.419	25.1661	24.6804	1.8231
4	26.03	25.9201	24.8829	24.4989	24.438	0.5484
5	26.4852	26.0375	25.416	24.8233	24.2776	1.5998

COSMOS2015 Data

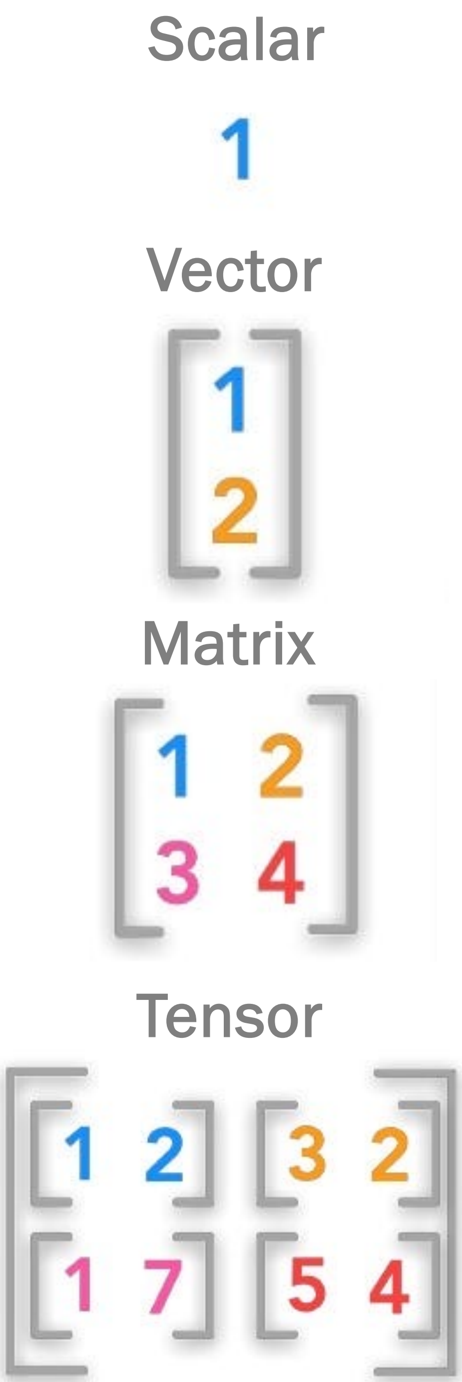
	A	B	C	D	E	F
1	g_cmodel_mag	r_cmodel_mag	i_cmodel_mag	z_cmodel_mag	y_cmodel_mag	specz_redshift
2	20.9795723	19.9471302	19.0677147	18.58988	18.398716	0.548910022
3	21.9359283	20.2798767	19.2970181	18.8702316	18.6586075	0.548210025
4	18.2886353	17.6349106	17.292387	17.095787	16.9314461	0.069250003
5	22.0716896	20.4420376	19.3423538	18.947113	18.7577057	0.565800011

Parameters & Hyperparameters

- Parameters: known inputs to input layers
  - Photo band magnitudes
- Hyperparameters: properties of the NN
  - Hidden layers
  - Neurons per layer
  - Loss metric:
    - SGD
    - Adam
    - AdGrad
  - Parameters of training procedure:
    - Momentum
    - Learning rate

PyTorch

- Torch library in Python
- ML framework
  - Computer vision, NLP, training NN
- Analogous to TensorFlow
  - PyTorch $\longrightarrow$ academia
  - TensorFlow $\longrightarrow$ industry



My Project

- Build NN redshift estimator in Python using PyTorch
- Optimize hyperparameters with Optuna
  - Limit number of COs
  - Apply RMS as a metric of error

- Hyperparameters: properties of NN
  - Hidden layers
  - Neurons per layer
  - Training procedure/code framework
    - SGD
    - Adam
    - Adgrad
  - Parameters of training procedure:
    - Momentum
    - Learning rate

Optuna

- Hyperparameter optimization framework
- In PyTorch:
  - Wrap Optuna around model and let it learn
  - Will quickly and efficiently try new values of hyperparameters
  - Goal will be to find best combination of hyperparameters

Optuna

- Potential framework:
  - Use 50% of galaxies as training set
  - Use other 50% as testing set
- Unsure about metric of success…
  - Limit number of COs
  - Apply RMS:

$$\mathrm{RMS} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Z_{phot\,i})^2 - (Z_{spec\,i})^2}$$

Sources

- https://en.wikipedia.org/wiki/Photometry_(astronomy)
- https://en.wikipedia.org/wiki/Photometric_redshift
- https://aws.amazon.com/what-is/neural-network/
- https://www.superannotate.com/blog/guide-to-gradient-descent-algorithms
- https://optuna.org
- https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide
- https://www.youtube.com/watch?v=aircAruvnKk

Thank you,

Dr. Singal!

Questions?