**Proposal structure:**

Abstract

Introduction

What is open clusters, why they are important, what are the available data

What is stellar evolution, why is it important, astroseismology (do we have to include these)

Hendriks & Aerts 2018

Method

What is machine learning, what is neural networks, what is tensorflow/keras

hierarchical Bayesian modelling of stellar ages in an open cluster

HMC, NUTS

to get the ages to feed into NUTS, need astroseismic data, neural net

Results?

D. Stello et. al 2016 gave masses for 33 red giants in M67

Found mu and sigma for M67 based on D. Stello

Implication

Good stuff about approximating stellar models/grids with neural net: much much faster, possibly cant account for loss function

Good stuff for getting age and spread of cluster’s star age:

Work flow

Week 1-4 background research and proposal

Week 5-7 HMC and NUTS on M67

Week 8-11 research for grid/stellar models for neural net

Term break building neural net

Week 1-5 building and testing neural net with validating program

Week 6-8 extra stuff

Week 9-11 write up

References

##############################################

**Outline of project:**

Sciency bit:

* trying to get age distribution (mean and spread) of some well documented open cluster, current candidate is M67 (other candidates: NGC6819, NGC6791, M4 (GC not OC but ...), Ruprecht 147)
* with age distribution and perhaps a good estimation of the individual star’s ages, can work on stuff related to Mg/Fe, which is believed to have a direct relationship with how old the stars are (due to supernova history of the Milky way/star cluster) (e.g. if find any form of correlation between age, either by direct plotting of individual stars, or having the Mg/Fe range increase with increase age range on different open clusters, are super good results)

Method stuff:

* forward modelling = from stellar properties/parameters/fundamentals (Guy uses fundamentals) to observables
* backward modelling = from observables to stellar parameters
* forward is better than backward cause much much easier to do hieratical modelling and Bayesian inference on stuff that follows the flow of physics
* Overarching structure is a massive hieratical model through HMC, with layers “hyper prior”, “prior”, “model population’s true values, inputs and outputs”, “model population’s observed values”, “real observed values”, “posterior”
* For each stellar parameter that we are trying to estimate, we set up one hyper prior, with a guess from previous research, mu and uncertainty del mu, and a spread sigma and an uncertainty on spread del sigma
* A whole bunch of priors that consist of (mu\_i, sigma\_i) is created that correspond to the normal distributions given by mu, del mu, sigma and del sigma
* For each of those priors, a population of N stars is created, each with a set of stellar parameters to be pumped into the already trained NN to get the observables of that modelled star
* The modelled star population’s observables will have observational noise added to them (according to some noise manually specified)
* The modelled star population’s noise added observables will be compared to the real observed values, to give a posterior, an evaluation of how “look alike” the two populations are
* With a large number of priors made, HMC will sample a bunch of mu and sigma values to construct the posterior distribution against mu and sigma. Peak of that distribution is the number with the highest probability to be the real value (of a certain stellar parameter) for the population of stars in concern
* done

##############################################

European CoRoT

(Auvergne et al. 2009) and the NASA Kepler (Koch et al.

2010) space telescopes

List of machine learning on stellar population papers:

Verma 2016

Bellinger 2016

Hendriks & Aerts 2018

**Stuff got from Hendriks & Aerts 2018:**

A prominent application of machine learning (ML) to lowmass star asteroseismology based on damped pressure modes was developed by Bellinger et al. (2016).

Why does non-linearity has to be added to neural network

Overfitting -> penalty term for high weights regularization (Ng 2004)

The input data for the neural network are 8D: the six stellar parameters (M, X, Z, Xc, fov, and Dmix) and two integer mode quantum numbers connected with the frequency of the zonal mode: l and npg. The output of the network is the mode frequency f i th for all modes i with degree 0, 1, or 2 and for all radial orders npg Î- + [ 50, 5] (see Figure 1).

After trained model from grids, now with a trained neural network, to go backwards and find the corresponding stellar parameters with a given set of mode freq, used a cost function under a particle finder (PF). PF looks for the local region in the 6D input space and finds the values that gives the least error compared to the mode freqs, under some layered loops of constraining functions.

Forward modelling in astroseismology = making a model that gives observables (del nu, nu max, individual mode freq) from stellar parameters (mass, radius, age)

Kendall & Gal (2017): procedures to turning a deep neural network into a Bayesian deep neural network that also gives uncertainties

To site tensorflow: <https://www.tensorflow.org/about/bib>