**Property Determinations of Open Clusters using Hierarchical Bayesian Modelling and Neural Networks**

H. Westwood, ID: 1556764  
H. Leung, ID:

**Abstract:**

Context:   
- open cluster are good for evolution tracing  
- current methods are time consuming and rely on previous work  
Aims:   
- determine accurate properties of OCs using NNs and HBMs  
Methods:  
- NN trained on grid.  
- NN put into HBM  
- sampled different clusters with the help from data from the literature.   
Results:  
- Mean ages, Mean [Fe/H], maybe mean Y0 (and MLT?) for each cluster, some additional interpretation of results.

Estimated Word Count:

Key words: neural network, HBM, M67

Contents?  
Nomenclature?

**Introduction:**  
- Open clusters are useful for tracing galactic evolution because: same age+metallicity etc.  
- Why our method is expected to be good for getting results that can improve galactic archaeology, based on/compared to previous works  
- outline the structure of the paper

**Target Selection:**  
- looking for a spread in properties, mainly in age and [Fe/H]  
- outline the main targets (M67, NGC6791, NGC6819, NGC2158, NGC188, NGC752)  
- cite membership paper and describe how we selected the most stars flagged as part of some cluster such that the uncertainty per star was < 1% such that we could be confident in the cluster members and that they wouldn’t cause grievances with the HBM’s sampling.  
- Luminosity calculations and the parts therein. (we ignore reddening + treat extinction error as 0)  
- rigorous star removal (blue stragglers etc.)

**Stellar evolution grid (training data):**  
- how was it made  
- what does it consist of i.e. variable ranges and spacing.  
- Tracks are evolved from pre-main sequence to before core-He burning.

**Neural Networks:**  
- NN Theory: effectively NN introduction  
 - Cybenko’s universal function approximation theorem is why neural networks provide a  
 good avenue for investigation  
 - further reasons why they are useful for our means and how we intend to improve upon  
 previous work.  
 NN method:  
 - what we did to improve the NNs and how that links back to the theory for NNs  
 - validation+shuffling, batch\_sizes, architecture, regularization, batch normalisation,   
 (relu v elu?, MSE v MAE?, L1 v L2?, Nadam v SGD?)  
 - underfitting/overfitting, RGB problem+limitation due to grid spacings  
 - training pipeline, accuracy achieved and loss, thus graphs.

Validation amount is around 10% to 20%, hold back a random selection from the data, which is why shuffling the data is important, as it prevents some chunk from some part of the data distribution being reserved for validation which prevents the validation being done holistically on the data range.

Why we chose the NUTS sampler over other samplers and reference some paper that explains NUTS samplers.

Data augmentation = adding data to neural network training  
Paper may also contain information on how the neural networks improves with increasing the number of data points.   
You increase the number of data points using Gaussian processes (which is much faster than by calculating more stellar evolutionary tracks.)

From a small batch size and small learning rate, you will move quickly but eventually you will be exploring the noise of the data due to each batch not being representative of the whole data. Therefore, you increase the batch size, and so that it doesn’t take forever you also need to DELICATELY increase the learning rate. In order to converge quickly but keep stability

Getting small details requires large batch sizes

Although Guy is of the opinion that you need to use different batch sizes to accomplish different tasks along training. So initially if you had batches with only 1 data point the neural net would just try to pull in to that point but then when it looks at the next single point batch try to instead pull to that point with the result of the neural net just pinging about everywhere not accomplishing anything. However, if you immediately were to use full batch then the neural net wouldn’t have any idea of where to start as it can’t choose a particular feature to start learning from. The initial just generalizing batch size allows the neural net to choose a particular feature to view the trend in and as you increase the batch it can use the trend it has already learned and then see how the additional data effects that trend.   
We should say in the paper why you shouldn’t put the batch size too high or too low, and to describe how we came to this conclusion.

Conclusions: if we get a results like this we can get that

Implications: doing this for more clusters would yield better results

Acknowledgements: Guy, Tanda, Alex