Uncertainty Estimation in Selective Classification on Skin Lesions

AC40001

Mid-term Report

Daniel Blackley - 160007728

# **Summary and Aims**

Machine learning algorithms have been shown to perform exceptionally well on the classification of skin cancer, matching the accuracy of experts in the field[1]. Due to the nature of the medical domain being safety critical, it is important that any predictions made by these algorithms have a quantifiable measure of uncertainty and correctly consider the cost of misclassifying certain Skin Lesions.

I hope to investigate epistemic uncertainty in machine learning Algorithms by comparing a baseline Softmax response against Yarin Gal’s Monte Carlo Dropout method, using the efficient net architecture and Entropy across predictions as a measure for uncertainty.

# **Background Research**

The two most common types of uncertainty that you can have with regards to machine learning are epistemic and aleatoric[2]. Aleatoric uncertainty is a measure of unnatural (i.e. artificially added white noise) and out of sample data, whereas epistemic uncertainty is uncertainty in regards to things that a machine could in theory learn, but has not due to a lack of training samples. There are many papers looking into both types of uncertainty, and many different methods of discerning a measure of uncertainty from machine learning algorithms[5][6].

Bayesian Networks usually learn a distribution over the weights, which requires significant modification to the training procedure and can be computationally expensive, There are many alternatives to traditional Bayesian neural networks; One popular method is called Deep Ensembles[3], which, on a basic level, consists of training multiple networks and combining them into one more powerful Network. Another Method that was proposed by Yarin Gal was MC Dropout, it uses dropout at run time, which he proved was equivalent to approximate Varational Inference[4], this method is less computationally expensive and is also easy to implement due to the prevalence of dropout as a regularisation technique. Another, even simpler method, is using 1 –

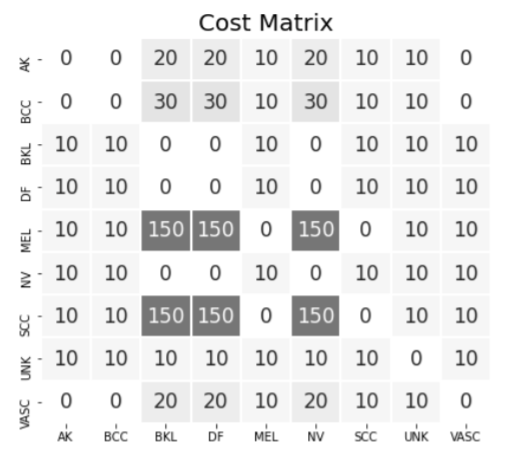


Figure 1: Cost matrix showing the cost of misclassification with Predictions on the Y axis and True labels on the X axis, taken from [9]

max softmax probability, this was shown to be a rather effective method for calculating uncertainty [8] and serves as a good baseline to compare other methods to. In my project I plan to compare MC Dropout to a baseline softmax response, primarily due to the comparatively computationally inexpensive nature of the proposed methods. Both have shown to have comparable results on various datasets[7], and Skin Lesion classification[5].

Not all classifications of Skin Lesions are equal, some misclassifications can be particularly more dangerous than others. A cost matrix can help counteract this, Figure 1 shows the implementation I will be using.

# **Main Features**

I plan to implement a cost sensitive and risk aware efficient net b0 model using pytorch, From this I hope to retrieve useful evaluation metrics that can help compare the softmax baseline and MC Dropout to provide a informative report on which model provides a better prediction of uncertainty

# **Progress to date**

So far I have managed to implement an efficientNetb0 model capable of outputting both a basic softmax prediction as well as a range of predictions by applying dropout at

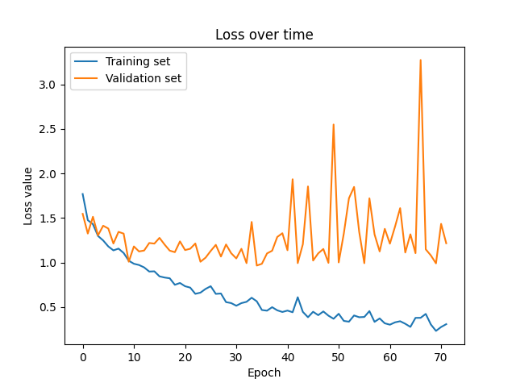
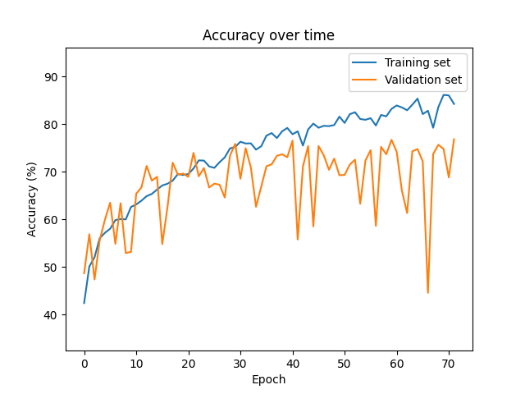


Figure 2a – Learning curve Figure 2b – Accuracy over each epoch

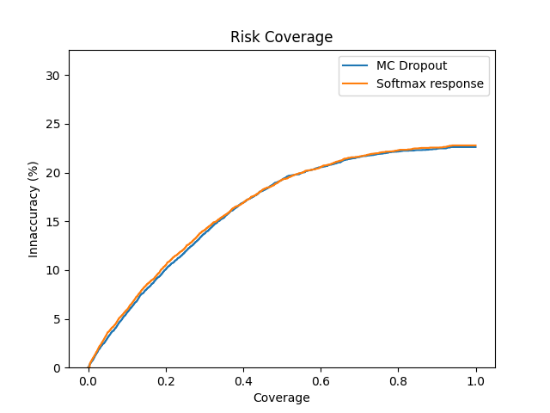


Figure 3 – Risk Coverage curve showing accuracy as the threshold for uncertainty is increased, entropy threshold values were obtained by finding the lowest uncertainty value across all uncertainties, plotting accuracy, then finding the second lowest uncertainty plotting accuracy…

run time. These models gets a accuracy of 77% on the ISIC 2019 Dataset without any sort of risk consideration. Figure 2a illustrates the Learning curve while Figure 2b shows the accuracy over the epochs. In terms of how I obtained this model, I trained an efficientnet model, with an extra 512 linear layer, on the training data until my learning loss hit a gradient of roughly 0 (80 epochs) and save the model that scored the best on the validation set over all epochs. Then during test time I apply dropout to the 512 neuron layer, or don’t, depending on which results I need.

I have also obtained some preliminary metrics for the evaluation of uncertainty; Figure 3 shows a risk coverage curve and Figure 4a and 4b shows the entropy across incorrect and correct predictions.

In terms of discoveries and challenges, I have found pytorch particularly easy to work with, after reading so much into the theory on the subject I was a little overwhelmed, expecting to have to manually implement everything on my own, but found out the pytorch already supplies plenty of optimisers, loss functions and deep learning architecture for you to use and adapt.

# **Personal Reflection**

I’ve certainly noticed the progress I’ve made so far, there is definitely a considerable difference in my knowledge and skillset between at the start of semester 1 and now. In terms of if I’m on track, I am literally, according to Figure 5, on track for getting work done on time, in terms of the quality of work/ the grade I hope to achieve, I think its going to be particularly hard to tell until I begin writing up my findings. I think my understanding of some things has been confused, but the current cycle of implementation, meeting up and then finding out which parts can be improved on has been the fastest way to see exactly which parts I don’t understand. Even now I presume there will be some mistakes in the earlier sections of this report that will be interesting to go over.

# **Plans for Remainder of Project**

Figure 5 is the Gantt chart I’ve put together, as of now, my next major milestone should be finishing up implementation in the coming weeks and begin looking into writing up my report. After that the report has been split into more minor milestones as shown in Figure 5, (i.e. literature review, data analysis).

I don’t expect I will be following this Gantt chart exactly as I don’t know how big of a challenge writing the report will be, due to this being my first time writing something of this nature. The best way to minimise this challenge is going to be making sure I start looking at how other papers interpret their results and paying close attention to any technical details that they talk about, hopefully catching any problems that I might run into before I reach them.

# **References**

[1] Tschandl P, Codella N, Akay BN, Argenziano G, Braun RP, Cabo H, Gutman D, Halpern A, Helba B, Hofmann-Wellenhof R, Lallas A, Lapins J, Longo C, Malvehy J, Marchetti MA, Marghoob A, Menzies S, Oakley A, Paoli J, Puig S, Rinner C, Rosendahl C, Scope A, Sinz C, Soyer HP, Thomas L, Zalaudek I, Kittler H. Comparison of the accuracy of human readers versus machine-learning algorithms for pigmented /skin lesion classification: an open, web-based, international, diagnostic study. Lancet Oncol. 2019 Jul;20(7):938-947. doi: 10.1016/S1470-2045(19)30333-X. Epub 2019 Jun 12. PMID: 31201137.

[2] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? Structural safety, 31(2):105–112, 2009.

[3] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scal-able Predictive Uncertainty Estimation using Deep Ensembles. 2017. arXiv:1612 .01474 [stat.ML].

[4] Yarin Gal and Zoubin Ghahramani.Dropout as a Bayesian Approximation: Repre-senting Model Uncertainty in Deep Learning. 2016. arXiv:1506.02142 [stat.ML]

[5] Leibig C. et al. “Leveraging uncertainty information from deep neural networksfor disease detection”. In: Sci Rep 7.17816 (2017). DOI:https://doi.org/10.1038/s41598-017-17876-z.

[6] Marc Combalia et al. “Uncertainty Estimation in Deep Neural Networks for Der-moscopic Image Classification”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. June 2020.

[7] Geifman, Yonatan & El-Yaniv, Ran. (2017). Selective Classification for Deep Neural Networks.

[8] Dan Hendrycks and Kevin Gimpel. “A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks”. In:CoRRabs/1610.02136(2016). arXiv:1610.02136. <URL:http://arxiv.org/abs/1610.02136>.

[9] Vasileios Aidonis. “Cost-Sensitive Deep Learning for Skin Lesion Diagnosis”. 2020. [University of Dundee MsC]

# **Appendix**

# No description available.

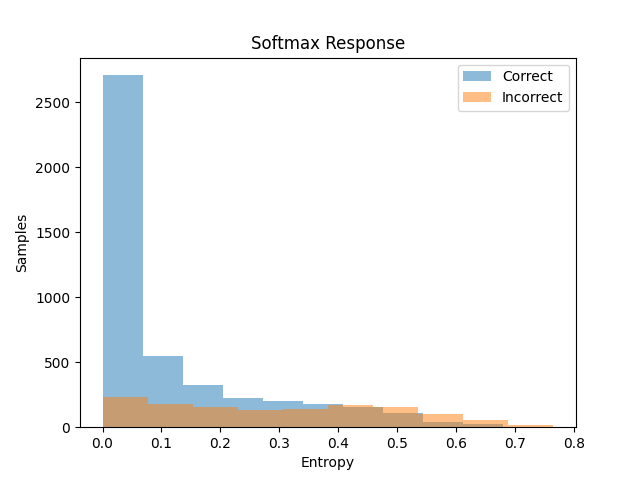


Figure 4a/b, histograms showing number of correct/incorrect samples and their entropies

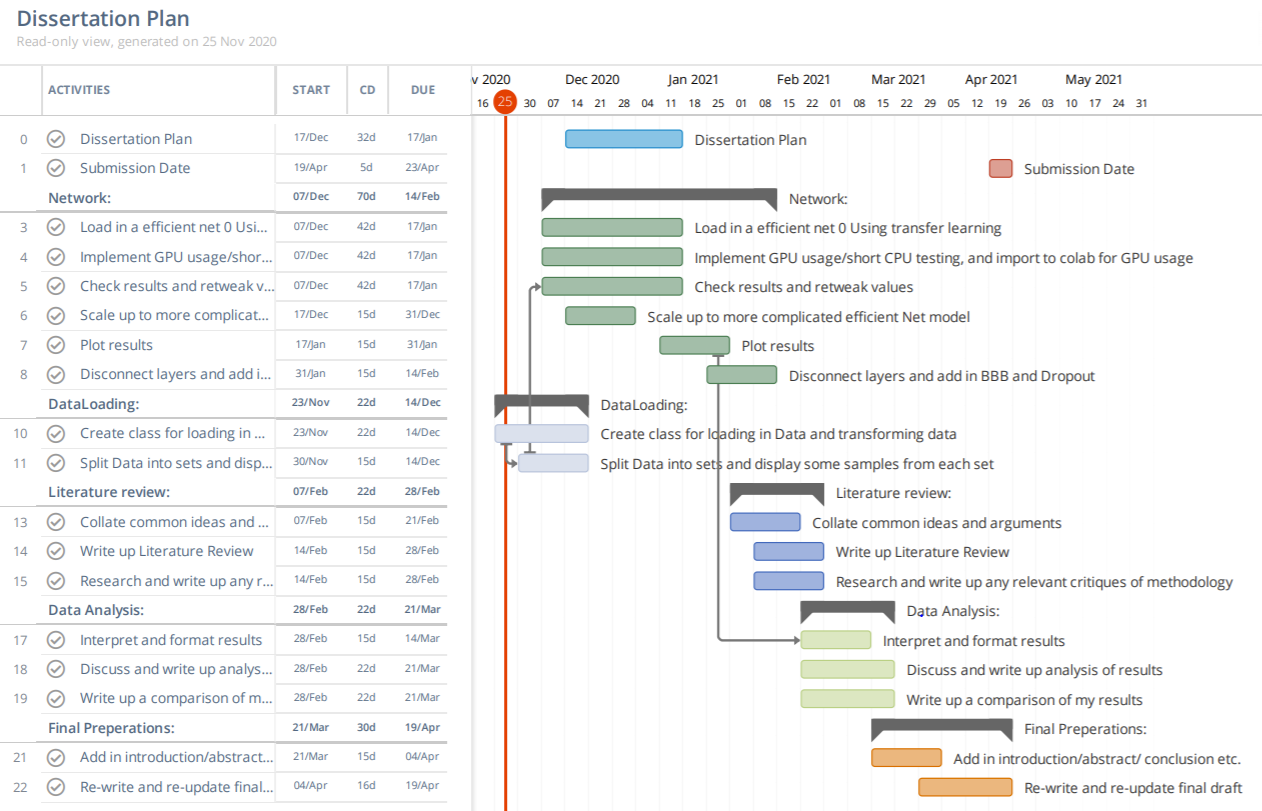


Figure 5, Gantt chart showing dates and predicted deadlines.