**Supplementary appendix – resources**

This appendix does not provide an exhaustive review of the available software and resources, but discusses examples, from the author’s experience, that are particularly useful.

**1. MACHINE LEARNING RESOURCES**

As a brief review section, many aspects of machine learning, especially the numerous techniques used in this field, have not been discussed. For those readers looking for more information, a useful resource for actuaries is the lecture notes by Wüthrich and Buser (2018), which provide a mathematical perspective on machine learning with many applications in actuarial science, and the papers by Parodi (2012a, 2012b) which examine machine learning in the context of non-life insurance. Chapter 12 in Parodi (2014) discusses some issues of the machine learning process applied in the insurance pricing context. Recent applications of machine learning to individual claims reserving is Wüthrich (2018) and to mortality modelling is Deprez, Shevchenko, and Wüthrich (2017).

Considering more general sources, the book by James, Witten, Hastie, and Tibshirani (2013) provides an introductory look at many aspects of machine learning, including code in the R language (R Core Team, 2018), and the book by Friedman, Hastie, and Tibshirani (2009) provides more theoretical background. Both of these are written from a statistician’s perspective, which might be more easily understood by actuaries than similar books written by computer scientists. Kuhn and Johnson (2013) describes the predictive modelling process in the R language and provides case studies on applying many machine learning algorithms[[1]](#footnote-1). Finally, the book by Efron and Hastie (2016) provides an interesting and up to date discussion of the interplay between statistical and machine learning methods.

In terms of software, the caret package (Kuhn, 2008) in R provides a unified interface to machine learning algorithms in other packages, and, in Python, scikit-learn (Pedregosa et al., 2011) is an integrated package containing many machine learning algorithms and utilities.

**2. DEEP LEARNING RESOURCES**

**2.1 Software**

Several open-source software packages allowing for the application of deep learning are currently available. Amongst these, the TensorFlow (Abadi et al., 2016) and PyTorch (Paszke et al., 2017) packages provide low level interfaces for defining and fitting deep neural networks, both of which are accessible in the Python programming language and provide support for fitting neural networks on CPUs as well as GPUs, the latter being recommended for deep neural networks involving the application of specialized layers. Some of the workflow used when defining TensorFlow models in the form of programming graphs might appear very foreign compared to the same workflow in PyTorch, which, anecdotally, is said to be more intuitive for users of Python.

A notable high level interface is Keras (Chollet, 2015), which provides an intuitive domain specific language for easily designing and fitting neural networks, which are then run on a low level interface, such as TensorFlow. Keras abstracts the many of the finer details away from the user and requires less technical knowledge than the low-level packages mentioned before. An interface between Keras and the R language (R Core Team, 2018) has been made available (Allaire & Chollet, 2018), which is notable since R appears to be more popular among actuaries than Python. For the code examples, this paper uses the R interface to Keras, running on the TensorFlow back-end.

**2.2 Hardware**

Modern neural networks require a significant amount of computing power to fit. Graphics processing units (GPUs) are often used to speed up the fitting of deep networks, and the software packages just discussed provide the option to use GPUs. Several web-based platforms offer GPU access in the cloud if local access to a GPU is not available.

**2.3 Resources**

The book by Goodfellow, Bengio, and Courville (2016) is currently the most comprehensive resource providing background, theory and perspective on all aspects of modern neural networks, from the vantage point of computer science. Chollet (2017) provides a practical guide to the Keras package in Python and a similar guide to the Keras package in R is Chollet and Allaire (2018). Both of these resources discuss practical aspects of fitting deep neural networks and provide example code in the respective programming languages. A concise introduction to deep learning from a statistical viewpoint is Section 18 in Efron and Hastie (2016). Beyond books, the online course by Ng (2017) provides theory, practical advice and code examples in Python, and the Deep Learning Nanodegree (Udacity, 2018) combines theory with several projects applying deep learning.

**3. REFERENCES**

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., . . . Isard, M. (2016). *TensorFlow: A System for Large-Scale Machine Learning.* Paper presented at the OSDI.

Allaire, J., & Chollet, F. (2018). R interface to Keras: RStudio, Google. Retrieved from <https://cloud.r-project.org/web/packages/keras/index.html>

Chollet, F. (2015). Keras Retrieved from keras.io

Chollet, F. (2017). *Deep learning with Python*: Manning Publications Co.

Chollet, F., & Allaire, J. (2018). *Deep Learning with R*: Manning Publications Co.

Deprez, P., Shevchenko, P., & Wüthrich, M. (2017). Machine learning techniques for mortality modeling. *European Actuarial Journal, 7*(2), 337-352.

Efron, B., & Hastie, T. (2016). *Computer Age Statistical Inference* (Vol. 5): Cambridge University Press.

Friedman, J., Hastie, T., & Tibshirani, R. (2009). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. New York: Springer-Verlag.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*: MIT Press.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning* (Vol. 112): Springer.

Kuhn, M. (2008). Caret package. *Journal of Statistical Software, 28*(5), 1-26.

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling* (Vol. 26): Springer.

Ng, A. (2017). Deep Learning Specialization. Retrieved from <https://www.coursera.org/specializations/deep-learning>

Parodi, P. (2012a). Computational intelligence with applications to general insurance: a review: I – The role of statistical learning. *Annals of Actuarial Science, 6*(2), 307-343. doi:10.1017/S1748499512000036

Parodi, P. (2012b). Computational intelligence with applications to general insurance: a review: II. Dealing with uncertain knowledge. *Annals of Actuarial Science, 6*(2), 344-380. doi:10.1017/S1748499512000048

Parodi, P. (2014). *Pricing in general insurance*: CRC Press.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., . . . Lerer, A. (2017). Automatic differentiation in PyTorch.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*(Oct), 2825-2830.

R Core Team. (2018). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from <https://www.R-project.org>

Udacity. (2018). Deep Learning. Retrieved from <https://eu.udacity.com/course/deep-learning-nanodegree--nd101>

Wüthrich, M. (2018). Machine learning in individual claims reserving. *Scandinavian Actuarial Journal*, 1-16.

Wüthrich, M., & Buser, C. (2018). Data analytics for non-life insurance pricing. Retrieved from <https://ssrn.com/abstract=2870308>

1. Note, however, that the works just cited provide a somewhat out of date view of neural networks, and the resources at the end of the following section provide a helpful counterpoint to critically examine these parts of the books. [↑](#footnote-ref-1)