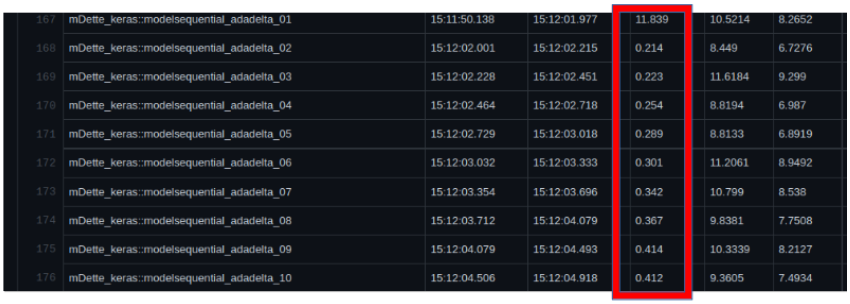
Review: A Review of R Neural Network Packages

March 2022

Although all of my comments were addresses by the authors, I'm not convinced by the revision. The manuscript does not present a comprehensive benchmark or comparison of neural network packages in R. That is for five major reasons:

1. Evaluation on training data (overfitting, not useful in practice)
2. Restriction to single-layer networks with tanh activation (doesn't reflect state of the art)
3. No hyperparameter tuning (hyperparameters need to be tuned for a fair comparison)
4. Too few datasets (insufficient to draw conclusions and benchmark suites are available; see initial review)
5. Inadequate evaluation of computational performance (The second criterion - "speed of the training algorithm" - includes package initialization and therefore depends on the complexity of the package; see example below)



167	mDette_keras::modelsequential_adadelta_01	15:11:50.138	15:12:01.977	11.839	10.5214	8.2652
168	mDette_keras::modelsequential_adadelta_02	15:12:02.001	15:12:02.215	0.214	8.449	6.7276
169	mDette_keras::modelsequential_adadelta_03	15:12:02.228	15:12:02.451	0.223	11.6184	9.299
170	mDette_keras::modelsequential_adadelta_04	15:12:02.464	15:12:02.718	0.254	8.8194	6.987
171	mDette_keras::modelsequential_adadelta_05	15:12:02.729	15:12:03.018	0.289	8.8133	6.8919
172	mDette_keras::modelsequential_adadelta_06	15:12:03.032	15:12:03.333	0.301	11.2061	8.9492
173	mDette_keras::modelsequential_adadelta_07	15:12:03.354	15:12:03.696	0.342	10.799	8.538
174	mDette_keras::modelsequential_adadelta_08	15:12:03.712	15:12:04.079	0.367	9.8381	7.7508
175	mDette_keras::modelsequential_adadelta_09	15:12:04.079	15:12:04.493	0.414	10.3339	8.2127
176	mDette_keras::modelsequential_adadelta_10	15:12:04.506	15:12:04.918	0.412	9.3605	7.4934

The keras package has a huge Python dependency that is built only when necessary (lazy-evaluation). This causes the first repetition to take much longer than all subsequent ones, only because of package and function initialization (image from `https://github.com/pkR-pkR/NNbenchmarkTemplates/blob/master/results_2019_gsoc2020/mDette-results.csv#L167`).

Most of these points have already been raised in the first round of reviews and while the authors provided answers to the issues, none of it was fixed or improved in the revision.

Minor issues:

1. In the keras package, the loss function MSE is used instead of RMSE. Additionally, the calculation of the metric (MAE) and the default batch size (32) could have an impact on the evaluation time.

2. p.1: "[...] we stick to the multilayer perceptron because it is still the most used NN structure [...]" - A very daring statement. Either a reference is needed here or it must be specified more precisely in which context they are most frequently used.

3. Since the R variant `torch` of PyTorch is becoming increasingly popular in the R community recently and is implemented directly in C++ without a Python dependency, I miss this very comprehensive package in your benchmark.