=> to be discussed (Option - Validating Neural Network Packages in R with NNbenchmark)

by Author One, Author Two

Abstract An abstract of less than 150 words. Possible format: 1) the overall purpose of the study and the research problem(s) you investigated; 2) the basic design of the study; 3) major findings or trends found as a result of your analysis; and, 4) a brief summary of your interpretations and conclusions.

Introduction

R Statistical Software, as any opensource platform, has relied on its contributors to keep it up to date with the latest developments. Neural networks is one of these advancements pertaining to a particular class of models in machine learning. Formerly a theory with not much practical implementation due to the complicated calculations of its algorithms, neural networks are now one of the most actively explored in its field. The reason it used to be so demanding is because neural networks use the gradient of the cost to step into the direction of the optimal solution. It updates the parameters of neurons accordingly to make better predictions until a certain number of iterations. Manually computing gradients for large, or even small datasets, was simply too hard. With the help of computers, neural network algorithms can be calculated through a few lines of code.

However, this capability of computers is often not used to the greatest extent. Instead of calculating first order derivatives, and moving incrementally forward by a predetermined learning rate, it is faster to adjust the size of each step according to its curvature. Curvature can be learned about through calculating the second order derivative. Some algorithms take this further by numerical methods. The Hessian, a matrix of the second derivatives, is approximated. Such methods are called Quasi-Newton. Broyden-Fletcher-Goldfarb-Shanno (BFGS) is a popular example of an algorithm from this class. We believed that these second order algorithms are also better than first order algorithms in terms of finding the optimal solution. Regardless of our belief, it was important to conduct a thorough examination to assess the quality of these training algorithms in R. There is much code, but barely enough comparison. At the very least our research may be used as a framework for future research.

In R, code for new neural network algorithms, better code for existing ones, or code to import algorithms from other platforms/languages is submitted and shared in the form of packages. As of August the 25th, 2019, there were 77 packages in CRAN with this keyword. Those are: **AMORE** (Limas et al., 2014), **ANN2** (Lammers, 2019), **appnn** (Família et al., 2015), **autoencoder** (Dubossarsky and Tyshetskiy, 2015), **automl** (Boulangé, 2019), **BNN** (Jia, 2018), **brnn** (Rodriguez and Gianola, 2018), **Buddle** (Kim, 2018), **CaDENCE** (Cannon, 2017a), **cld2** (Ooms, 2018), **cld3** (Ooms, 2019), **condmixt** (Carreau, 2012), **DALEX2** (Biecek, 2018), **DamiaNN** (Siniakowicz, 2016), **DChaos** (Sandubete and Escot, 2019), **deepnet** (Rong, 2014), **deepNN** (Taylor, 2019), **DNMF** (Jia and Zhang, 2015), **elmNNRcpp** (Mouselimis and Gosso, 2018), **ELMR** (Petrozziello, 2015), **EnsembleBase** (Mahani and Sharabiani, 2016), **evclass** (Denoeux, 2017), **gamlss.add** (Stasinopoulos et al., 2016), **gcForest** (Jing, 2018), **GMDH** (Dag and Yozgatligil, 2016), **GMDH2** (Dag et al., 2019), **GMDHreg** (Tilve, 2019), **grnn** (Chasset, 2013a), **h2o** (LeDell et al., 2019), **hybridEnsemble** (Ballings et al., 2015), **isingLenzMC** (Suzen, 2016), **keras** (Allaire and Chollet, 2019), **kerasformula** (Mohanty, 2018), **kerasR** (Arnold, 2017), **leabRa** (Titz, 2017), **learNN** (Quast, 2015), **LilRhino** (Barton, 2019), **minpack.lm** (Elzhov et al., 2016), **MachineShop** (Smith, 2019), **monmlp** (Cannon, 2017b), **neural** (Nagy, 2014), **neuralnet** (Fritsch et al., 2019), **NeuralNetTools** (Beck, 2018), **NeuralSens** (Portela González and Muñoz San Roque, 2019), **NlinTS** (Hmamouche, 2019), **nlsr** (Nash and Murdoch, 2018), **nnet** (Ripley, 2016), **nnetpredint** (Ding, 2015), **nnfor** (Kourentzes, 2019), **onnx** (Tang and ONNX Authors, 2018), **OptimClassifier** (Pérez-Martín et al., 2018), **OSTSC** (Dixon et al., 2017), **pnn** (Chasset, 2013b), **polyreg** (Matloff et al., 2019), **predictoR** (with contributions from Diego Jimenez A. and D., 2019), **qrnn** (Cannon, 2019), **QuantumOps** (Resch, 2019), **quarrint** (Barthelemy et al., 2016), **radiant.model** (Nijs, 2019), **rasclass** (Wiesmann and Quinn, 2016), **rcane** (Suresh et al., 2018), **regressoR** (Rodriguez R., 2019), **rminer** (Cortez, 2016), **rnn** (Quast and Fichou, 2019), **RSNNS** (Bergmeir, 2018), **ruta** (Charte et al., 2019), **simpleNeural** (Dernoncourt, 2015), **snnR** (Wang et al., 2017), **softmaxreg** (Ding, 2016), **Sojourn.Data** (Hibbing and Lyden, 2019), **spnn** (Ebrahimi, 2018), **TeachNet** (Steinbuss, 2018), **tensorflow** (Allaire and Tang, 2019), **tfestimators** (Allaire et al., 2018), **trackdem** (Bruijning et al., 2019), **TrafficBDE** (Chatzopoulou et al., 2018), **tsensembler** (Cerqueira, 2019), **validann** (Humphrey, 2017), **zFactor** (Reyes, 2019).

In particular, packages that provide neural network of the perceptron type (one input layer, one normalized layer, one hidden layer with a nonlinear activation function that is usually tanh(), one normalized layer, one output output layer) for regression purpose (i.e. NN(X1, ., Xn) = E[Y] were the focus of this research.

Methodology

Our research process can be divided into 3 phases: preparation, exploration, and results. In practice, these three phases overlapped as in any other research.

Phase 1 - Preparation

Datasets
All the datasets used were nonlinear. Linear data sets are more simple and can even be solved with OLS regression. This is why we believe to truly set apart the ability of neural networks we needed to go beyond linear regression. Varying difficulties between data sets helped to classify further package's algorithms accuracy. One site was used for 3 of the multivariate data sets. Sonja Surjanovic and Derek Bingham of Simon Fraser University created this resourceful website to evaluate the design and analysis of computer models. Links to each dataset and their level of difficulty:
- `http://www.sfu.ca/~ssurjano/fried.html` (Friedman - average)
- `http://www.sfu.ca/~ssurjano/detpep10curv.html` (Dette - medium)
- `http://www.sfu.ca/~ssurjano/ishigami.html` (Ishigami - high)
The other multivariate dataset, Ref153, was taken from . . . 3 of the univariate datasets were taken from NIST at `https://www.itl.nist.gov/div898/strd/nls/nls_main.shtml`. Gauss1 and Gauss2 have a low level of difficulty to solve. Gauss3 is average. Dmod1, Dmod2 Dreyfus1, Dreyfus2 NeuroOne from . . .
Packages
Searching manually through the thousands of packages title or the package description of thousands packages would have taken a long time. With **RWsearch** (Kiener, 2019) we were able to automate the process. All packages that have "neural network" as a keyword in the package title or in the package description, that are mentioned in the introduction, were included.

Phase 2 - Exploration of each package and development of template

Exploration
However, not all packages that had the keyword were fit for the scope of our research. Some didn't have any functions to make neural networks. They were simply meta-packages. Others were not regression neural networks of the perceptron type or were only made for specific purposes. We learned this through reading documentation and trying out example code. **Template**
As we inspected the packages, we developed a template for benchmarking. This template's structure is as follows:
(1) Set up of environment - loading packages, setting directory, options;
(2) Summary of datasets;
(3) A loop over datasets which contained (a) setting parameters for a specific dataset (b) selecting benchmark options (c) the training of a neural network with a package's tuned functions (d) calculation of RMSE and MAE (e) plot each training over one initial graph, then plot the best result (f) adding results to the appropriate *.csv file and (g) clearing up environment for next loop; and
(4) Clearing up the environment for the next package. (5) It is optional to print warnings.
This process was made easier with tools from the NNbenchmark package. It is not on CRAN yet and can instead be found at `https://github.com/pkR-pkR/NNbenchmark`. Our templates for each package can be found in the companion repository, `https://github.com/pkR-pkR/NNbenchmarkTemplates`.

Phase 3 - Collection of and analysis of results

Collection After the templates were finished, the packages were looped on all datasets. Results were collected in the directory of the templates repository. **Analysis** We manipulated the results with the following rating scheme:

Results

The following is the final table of results. Further components of the rating can be found at the end.

Table 1: Results of Benchmarking

No	Name (package::algorithm)	Rating	Comment
1	**AMORE**::train.ADAPTgd	*	
	AMORE::train.ADAPTgdwm	*	
	AMORE::train.BATCHgd	rating	
	AMORE::train.BATCHgdwm	rating	
2	**automl**	rating	
3	**ANN2**::neuralnetwork.sgd	rating	
	ANN2::neuralnetwork.adam	rating	
	ANN2::neuralnetwork.rmsprop	rating	
4	**brnn**	***	
5	**CaDENCE**	rating	
6	**deepnet**::gradientdescent	rating	
7	**elmNNRcpp**	rating	
8	**ELMR**	rating	
9	**h2o**::deeplearning	rating	bad on univariate datasets
10	**keras**	rating	high level API for tensorflow
11	**kerasformula**	rating	
12	**kerasR**	rating	
13	**minpack.lm**::nlsLM	**	Requires hand-made formulas + scaling
14	**MachineShop**::fit.NNetModel()	rating	uses nnet
15	**monmlp**::fit.BFGS	rating	optimx, 200 iterations
	monmlp::fit.Nelder-Mead	rating	optimx, 10000 iterations
16	**neural**::mlptrain	rating	more appropriate for classification
17	**neuralnet**::backprop	rating	not good, most NA's
	neuralnet::rprop+	rating	100,000 iter
	neuralnet::rprop-	rating	above
	neuralnet::sag	rating	above
	neuralnet::slr	rating	above
18	**nlsr**::nlxb	**	Requires hand-made formulas + scaling
19	**nnet**::nnet.BFGS	rating	optim, 250 iterations, good
20	**qrnn**::qrnn.fit	rating	might be the best, 2000 iters?
21	**radiant.model**::radiant.model	rating	uses nnet, 10000 iters
22	**rcane**::rlm	rating?	linear, not appropriate
23	**rminer**::fit	rating	uses nnet
24	**RSNNS**::BackpropBatch	rating	
	RSNNS::BackpropChunk	rating	
	RSNNS::BackpropMomentum	rating	
	RSNNS::BackpropWeightDecay	rating	
	RSNNS::Quickprop	rating	
	RSNNS::Rprop	rating	
	RSNNS::SCG	rating	
	RSNNS::Std-Backpropagation	rating	
25	**ruta**	rating	
26	**simpleNeural**::sN.MLPtrain	rating	NA's, works for Ref153-NeuroOne
27	**snnR**	rating	classification
28	**softmaxreg**	rating	linear
29	**tensorflow**::AdadeltaOptmizer	rating	
	tensorflow::AdagradOptmizer	rating	
	tensorflow::AdamOptmizer	rating	
	tensorflow::FtrlOptmizer	rating	
	tensorflow::GradientDescent	rating	
	tensorflow::MomentumOptmizer	rating	
30	**tfestimators**	rating	
31	**tsensembler**	rating	uses nnet, chooses "optimal" weights
32	**validann**::Nelder-Mead	rating	10000 iter
	validann::BFGS	rating	good
	validann::CG	rating	good BUT slow, sys hangs
	validann::L-BFGS-B	rating	good
	validann::SANN	rating	10000 iter
	validann::Brent	rating	for one dimension, not looped

Table 2: Review of Ommitted Packages

No	Name (package)	Category	Comment
1	**appnn**	-	
2	**autoencoder**	-	
3	**BNN**	-	
4	**Buddle**	-	
5	**cld2**	-	
6	**cld3**	-	
7	**condmixt**	-	
8	**DALEX2**	-	
9	**DamiaNN**	-	
10	**DChaos**	-	
11	**deepNN**	-	
12	**DNMF**	-	
13	**EnsembleBase**	-	
14	**evclass**	-	
15	**gamlss.add**	-	
16	**gcForest**	-	
17	**GMDH**	-	
18	**GMDH2**	-	
19	**GMDHreg**	-	
20	**grnn**	-	
21	**hybridEnsemble**	-	
22	**isingLenzMC**	-	
23	**leabRa**	-	
24	**learNN**	-	
25	**LilRhino**	-	
26	**NeuralNetTools**	-	tools for neural networks
27	**NeuralSens**	-	tools for neural networks
28	**NlinTS**	NA	Time Series
29	**nnetpredint**	-	confidence intervals for NN
30	**nnfor**	NA	Times Series, uses neuralnet
31	**onnx**	-	provides an open source format
32	**OptimClassifier**	NA	choose classifier parameters, nnet
33	**OSTSC**	-	solving oversampling classification
34	**pnn**	NA	Probabilistic
35	**polyreg**	-	polyregression ALT to NN
36	**predictoR**	NA	shiny interface, neuralnet
37	**QuantumOps**	NA	classifies MNIST, Schuld (2018)
38	**quarrint**	NA	specified classifier for quarry data
39	**rasclass**	NA	classifier for raster images, nnet?
40	**regressoR**	NA	a manual rich version of predictoR
41	**rnn**	NA	Recurrent
42	**Sojourn.Data**	NA	sojourn Accelerometer methods, nnet?
43	**spnn**	NA	classifier, probabilistic
44	**TeachNet**	NA	classifier, selfbuilt, slow
45	**trackdem**	NA	classifier for particle tracking
46	**TrafficBDE**	NA	specific reg, predicting traffic
47	**zFactor**	NA	'compressibility' of hydrocarbon gas

Possible things to talk about: What is running under the packages? Dependencies, base functions / optimization algorithms nnet, neuralnet optim() | **optimx** (Nash and Varadhan, 2018)

Conclusion

Future work

As the alogrithms for neural networks continue to grow, there will always be more to validate. For current algorithms in R, our research should be extended to encompass more types of neural networks and their data formats (classifiers NN's, recurrent NN's, and so on). Different rating schemes and tunings of package functions can also be tried out.

Acknowledgements

This work was possible due to the support of Google during Summer of Code 2019.

Bibliography

J. Allaire and F. Chollet. *keras: R Interface to 'Keras'*, 2019. URL https://CRAN.R-project.org/package=keras. R package version 2.2.4.1. [p1]

J. Allaire and Y. Tang. *tensorflow: R Interface to 'TensorFlow'*, 2019. URL https://github.com/rstudio/tensorflow. R package version 1.14.0.9000. [p1]

J. Allaire, Y. Tang, K. Ushey, and K. Kuo. *tfestimators: Interface to 'TensorFlow' Estimators*, 2018. URL https://CRAN.R-project.org/package=tfestimators. R package version 1.9.1. [p1]

T. Arnold. *kerasR: R Interface to the Keras Deep Learning Library*, 2017. URL https://CRAN.R-project.org/package=kerasR. R package version 0.6.1. [p1]

M. Ballings, D. Vercamer, and D. Van den Poel. *hybridEnsemble: Build, Deploy and Evaluate Hybrid Ensembles*, 2015. URL https://CRAN.R-project.org/package=hybridEnsemble. R package version 1.0.0. [p1]

J. Barthelemy, T. Carletti, L. Collier, V. Hallet, M. Moriame, and A. Sartenaer. *quarrint: Interaction Prediction Between Groundwater and Quarry Extension Using Discrete Choice Models and Artificial Neural Networks*, 2016. URL https://CRAN.R-project.org/package=quarrint. R package version 1.0.0. [p1]

T. Barton. *LilRhino: For Implementation of Feed Reduction, Learning Examples, NLP and Code Management*, 2019. URL https://CRAN.R-project.org/package=LilRhino. R package version 1.1.0. [p1]

M. W. Beck. *NeuralNetTools: Visualization and Analysis Tools for Neural Networks*, 2018. URL https://CRAN.R-project.org/package=NeuralNetTools. R package version 1.5.2. [p1]

C. Bergmeir. *RSNNS: Neural Networks using the Stuttgart Neural Network Simulator (SNNS)*, 2018. URL https://CRAN.R-project.org/package=RSNNS. R package version 0.4-11. [p1]

P. Biecek. *DALEX2: Descriptive mAchine Learning EXplanations*, 2018. URL https://CRAN.R-project.org/package=DALEX2. R package version 0.9. [p1]

A. Boulangé. *automl: Deep Learning with Metaheuristic*, 2019. URL https://CRAN.R-project.org/package=automl. R package version 1.2.8. [p1]

M. Bruijning, M. D. Visser, C. A. Hallmann, and E. Jongejans. *trackdem: Particle Tracking and Demography*, 2019. URL https://CRAN.R-project.org/package=trackdem. R package version 0.4.3. [p1]

A. J. Cannon. *CaDENCE: Conditional Density Estimation Network Construction and Evaluation*, 2017a. URL https://CRAN.R-project.org/package=CaDENCE. R package version 1.2.5. [p1]

A. J. Cannon. *monmlp: Multi-Layer Perceptron Neural Network with Optional Monotonicity Constraints*, 2017b. URL https://CRAN.R-project.org/package=monmlp. R package version 1.1.5. [p1]

A. J. Cannon. *qrnn: Quantile Regression Neural Network*, 2019. URL https://CRAN.R-project.org/package=qrnn. R package version 2.0.4. [p1]

J. Carreau. *condmixt: Conditional Density Estimation with Neural Network Conditional Mixtures*, 2012. URL https://CRAN.R-project.org/package=condmixt. R package version 1.0. [p1]

V. Cerqueira. *tsensembler: Dynamic Ensembles for Time Series Forecasting*, 2019. URL https://CRAN.R-project.org/package=tsensembler. R package version 0.0.5. [p1]

D. Charte, F. Charte, and F. Herrera. *ruta: Implementation of Unsupervised Neural Architectures*, 2019. URL https://CRAN.R-project.org/package=ruta. R package version 1.1.0. [p1]

P.-O. Chasset. *grnn: General regression neural network*, 2013a. URL https://CRAN.R-project.org/package=grnn. R package version 0.1.0. [p1]

P.-O. Chasset. *pnn: Probabilistic neural networks*, 2013b. URL https://CRAN.R-project.org/package=pnn. R package version 1.0.1. [p1]

A. Chatzopoulou, K. Koupidis, and C. Bratsas. *TrafficBDE: Traffic Status Prediction in Urban Places using Neural Network Models*, 2018. URL https://CRAN.R-project.org/package=TrafficBDE. R package version 0.1.0. [p1]

P. Cortez. *rminer: Data Mining Classification and Regression Methods*, 2016. URL https://CRAN.R-project.org/package=rminer. R package version 1.4.2. [p1]

O. Dag and C. Yozgatligil. *GMDH: Short Term Forecasting via GMDH-Type Neural Network Algorithms*, 2016. URL https://CRAN.R-project.org/package=GMDH. R package version 1.6. [p1]

O. Dag, E. Karabulut, and R. Alpar. *GMDH2: Binary Classification via GMDH-Type Neural Network Algorithms*, 2019. URL https://CRAN.R-project.org/package=GMDH2. R package version 1.5. [p1]

T. Denoeux. *evclass: Evidential Distance-Based Classification*, 2017. URL https://CRAN.R-project.org/package=evclass. R package version 1.1.1. [p1]

D. Dernoncourt. *simpleNeural: An Easy to Use Multilayer Perceptron*, 2015. URL https://CRAN.R-project.org/package=simpleNeural. R package version 0.1.1. [p1]

X. Ding. *nnetpredint: Prediction Intervals of Multi-Layer Neural Networks*, 2015. URL https://CRAN.R-project.org/package=nnetpredint. R package version 1.2. [p1]

X. Ding. *softmaxreg: Training Multi-Layer Neural Network for Softmax Regression and Classification*, 2016. URL https://CRAN.R-project.org/package=softmaxreg. R package version 1.2. [p1]

M. Dixon, D. Klabjan, and L. Wei. *OSTSC: Over Sampling for Time Series Classification*, 2017. URL https://CRAN.R-project.org/package=OSTSC. R package version 0.0.1. [p1]

E. Dubossarsky and Y. Tyshetskiy. *autoencoder: Sparse Autoencoder for Automatic Learning of Representative Features from Unlabeled Data*, 2015. URL https://CRAN.R-project.org/package=autoencoder. R package version 1.1. [p1]

R. Ebrahimi. *spnn: Scale Invariant Probabilistic Neural Networks*, 2018. URL https://CRAN.R-project.org/package=spnn. R package version 1.1. [p1]

T. V. Elzhov, K. M. Mullen, A.-N. Spiess, and B. Bolker. *minpack.lm: R Interface to the Levenberg-Marquardt Nonlinear Least-Squares Algorithm Found in MINPACK, Plus Support for Bounds*, 2016. URL https://CRAN.R-project.org/package=minpack.lm. R package version 1.2-1. [p1]

C. Família, S. R. Dennison, A. Quintas, and D. A. Phoenix. *appnn: Amyloid Propensity Prediction Neural Network*, 2015. URL https://CRAN.R-project.org/package=appnn. R package version 1.0-0. [p1]

S. Fritsch, F. Guenther, and M. N. Wright. *neuralnet: Training of Neural Networks*, 2019. URL https://CRAN.R-project.org/package=neuralnet. R package version 1.44.2. [p1]

P. R. Hibbing and K. Lyden. *Sojourn.Data: Supporting Objects for Sojourn Accelerometer Methods*, 2019. URL https://CRAN.R-project.org/package=Sojourn.Data. R package version 0.1.0. [p1]

Y. Hmamouche. *NlinTS: Non Linear Time Series Analysis*, 2019. URL https://CRAN.R-project.org/package=NlinTS. R package version 1.3.7. [p1]

G. B. Humphrey. *validann: Validation Tools for Artificial Neural Networks*, 2017. URL https://CRAN.R-project.org/package=validann. R package version 1.2.1. [p1]

B. Jia. *BNN: Bayesian Neural Network for High-Dimensional Nonlinear Variable Selection*, 2018. URL https://CRAN.R-project.org/package=BNN. R package version 1.0.2. [p1]

Z. Jia and X. Zhang. *DNMF: Discriminant Non-Negative Matrix Factorization*, 2015. URL https://CRAN.R-project.org/package=DNMF. R package version 1.3. [p1]

X. Jing. *gcForest: Deep Forest Model*, 2018. URL https://CRAN.R-project.org/package=gcForest. R package version 0.2.7. [p1]

P. Kiener. *RWsearch: Lazy Search in R Packages, Task Views, CRAN, the Web. All-in-One Download*, 2019. URL https://CRAN.R-project.org/package=RWsearch. R package version 4.6. [p2]

J. Kim. *Buddle: A Deep Learning Package for Statistical Classification Analysis*, 2018. URL https://CRAN.R-project.org/package=Buddle. R package version 1.0. [p1]

N. Kourentzes. *nnfor: Time Series Forecasting with Neural Networks*, 2019. URL https://CRAN.R-project.org/package=nnfor. R package version 0.9.6. [p1]

B. Lammers. *ANN2: Artificial Neural Networks for Anomaly Detection*, 2019. URL https://CRAN.R-project.org/package=ANN2. R package version 2.3.2. [p1]

E. LeDell, N. Gill, S. Aiello, A. Fu, A. Candel, C. Click, T. Kraljevic, T. Nykodym, P. Aboyoun, M. Kurka, and M. Malohlava. *h2o: R Interface for 'H2O'*, 2019. URL https://CRAN.R-project.org/package=h2o. R package version 3.26.0.2. [p1]

M. C. Limas, J. B. O. Mere, A. G. Marcos, F. J. M. de Pison Ascacibar, A. V. P. Espinoza, F. A. Elias, and J. M. P. Ramos. *AMORE: A MORE flexible neural network package*, 2014. URL https://CRAN.R-project.org/package=AMORE. R package version 0.2-15. [p1]

A. S. Mahani and M. T. Sharabiani. *EnsembleBase: Extensible Package for Parallel, Batch Training of Base Learners for Ensemble Modeling*, 2016. URL https://CRAN.R-project.org/package=EnsembleBase. R package version 1.0.2. [p1]

N. Matloff, X. Cheng, P. Mohanty, B. Khomtchouk, M. Kotila, R. Yancey, R. Tucker, and A. Zhao. *polyreg: Polynomial Regression*, 2019. URL https://CRAN.R-project.org/package=polyreg. R package version 0.6.4. [p1]

P. Mohanty. *kerasformula: A High-Level R Interface for Neural Nets*, 2018. URL https://CRAN.R-project.org/package=kerasformula. R package version 1.5.1. [p1]

L. Mouselimis and A. Gosso. *elmNNRcpp: The Extreme Learning Machine Algorithm*, 2018. URL https://CRAN.R-project.org/package=elmNNRcpp. R package version 1.0.1. [p1]

A. Nagy. *neural: Neural Networks*, 2014. URL https://CRAN.R-project.org/package=neural. R package version 1.4.2.2. [p1]

J. C. Nash and D. Murdoch. *nlsr: Functions for Nonlinear Least Squares Solutions*, 2018. URL https://CRAN.R-project.org/package=nlsr. R package version 2018.1.28. [p1]

J. C. Nash and R. Varadhan. *optimx: Expanded Replacement and Extension of the 'optim' Function*, 2018. URL https://CRAN.R-project.org/package=optimx. R package version 2018-7.10. [p4]

V. Nijs. *radiant.model: Model Menu for Radiant: Business Analytics using R and Shiny*, 2019. URL https://CRAN.R-project.org/package=radiant.model. R package version 1.0.0. [p1]

J. Ooms. *cld2: Google's Compact Language Detector 2*, 2018. URL https://CRAN.R-project.org/package=cld2. R package version 1.2. [p1]

J. Ooms. *cld3: Google's Compact Language Detector 3*, 2019. URL https://CRAN.R-project.org/package=cld3. R package version 1.2. [p1]

A. Petrozziello. *ELMR: Extreme Machine Learning (ELM)*, 2015. URL https://CRAN.R-project.org/package=ELMR. R package version 1.0. [p1]

J. Portela González and A. Muñoz San Roque. *NeuralSens: Sensitivity Analysis of Neural Networks*, 2019. URL https://CRAN.R-project.org/package=NeuralSens. R package version 0.0.5. [p1]

A. Pérez-Martín, A. Pérez-Torregrosa, M. Vaca-Lamata, and A. J. Verdú-Jover. *OptimClassifier: Create the Best Train for Classification Models*, 2018. URL https://CRAN.R-project.org/package=OptimClassifier. R package version 0.1.4. [p1]

B. Quast. *learNN: Examples of Neural Networks*, 2015. URL https://CRAN.R-project.org/package=learNN. R package version 0.2.0. [p1]

B. Quast and D. Fichou. *rnn: Recurrent Neural Network*, 2019. URL https://CRAN.R-project.org/package=rnn. R package version 0.9.8. [p1]

S. Resch. *QuantumOps: Performs Common Linear Algebra Operations Used in Quantum Computing and Implements Quantum Algorithms*, 2019. URL https://CRAN.R-project.org/package=QuantumOps. R package version 2.5.2. [p1]

A. R. Reyes. *zFactor: Calculate the Compressibility Factor 'z' for Hydrocarbon Gases*, 2019. URL https://CRAN.R-project.org/package=zFactor. R package version 0.1.9. [p1]

B. Ripley. *nnet: Feed-Forward Neural Networks and Multinomial Log-Linear Models*, 2016. URL https://CRAN.R-project.org/package=nnet. R package version 7.3-12. [p1]

P. P. Rodriguez and D. Gianola. *brnn: Bayesian Regularization for Feed-Forward Neural Networks*, 2018. URL https://CRAN.R-project.org/package=brnn. R package version 0.7. [p1]

O. Rodriguez R. *regressoR: Regression Data Analysis System*, 2019. URL https://CRAN.R-project.org/package=regressoR. R package version 1.1.7. [p1]

X. Rong. *deepnet: deep learning toolkit in R*, 2014. URL https://CRAN.R-project.org/package=deepnet. R package version 0.2. [p1]

J. E. Sandubete and L. Escot. *DChaos: Chaotic Time Series Analysis*, 2019. URL https://CRAN.R-project.org/package=DChaos. R package version 0.1-2. [p1]

D. Siniakowicz. *DamiaNN: Neural Network Numerai*, 2016. URL https://CRAN.R-project.org/package=DamiaNN. R package version 1.0.0. [p1]

B. J. Smith. *MachineShop: Machine Learning Models and Tools*, 2019. URL https://CRAN.R-project.org/package=MachineShop. R package version 1.5.0. [p1]

M. Stasinopoulos, B. Rigby, V. Voudouris, and D. Kiose. *gamlss.add: Extra Additive Terms for GAMLSS Models*, 2016. URL https://CRAN.R-project.org/package=gamlss.add. R package version 5.0-1. [p1]

G. Steinbuss. *TeachNet: Fits Neural Networks to Learn About Backpropagation*, 2018. URL https://CRAN.R-project.org/package=TeachNet. R package version 0.7.1. [p1]

A. Suresh, S. Acharekar, H. Chao, and S. Y. Biradar. *rcane: Different Numeric Optimizations to Estimate Parameter Coefficients*, 2018. URL https://CRAN.R-project.org/package=rcane. R package version 1.0. [p1]

M. Suzen. *isingLenzMC: Monte Carlo for Classical Ising Model*, 2016. URL https://CRAN.R-project.org/package=isingLenzMC. R package version 0.2.5. [p1]

Y. Tang and ONNX Authors. *onnx: R Interface to 'ONNX'*, 2018. URL https://CRAN.R-project.org/package=onnx. R package version 0.0.2. [p1]

B. Taylor. *deepNN: Deep Learning*, 2019. URL https://CRAN.R-project.org/package=deepNN. R package version 0.3. [p1]

M. V. Tilve. *GMDHreg: Regression using GMDH Algorithms*, 2019. URL https://CRAN.R-project.org/package=GMDHreg. R package version 0.2.0. [p1]

J. Titz. *leabRa: The Artificial Neural Networks Algorithm Leabra*, 2017. URL https://CRAN.R-project.org/package=leabRa. R package version 0.1.0. [p1]

Y. Wang, P. Lin, Z. Chen, Z. Bao, and G. J. M. Rosa. *snnR: Sparse Neural Networks for Genomic Selection in Animal Breeding*, 2017. URL https://CRAN.R-project.org/package=snnR. R package version 1.0. [p1]

D. Wiesmann and D. Quinn. *rasclass: Supervised Raster Image Classification*, 2016. URL https://CRAN.R-project.org/package=rasclass. R package version 0.2.2. [p1]

O. R. R. with contributions from Diego Jimenez A. and A. N. D. *predictoR: Predictive Data Analysis System*, 2019. URL https://CRAN.R-project.org/package=predictoR. R package version 1.1.0. [p1]

Supplementary Materials

Table 3: Value of Components from Each Rating

No	Name (package::algorithm)	Docs	UF	CQ	Time	Final
1	**AMORE**::train.ADAPTgd					
	AMORE::train.ADAPTgdwm					
	AMORE::train.BATCHgd					
	AMORE::train.BATCHgdwm					
2	**automl**					
3	**ANN2**::neuralnetwork.sgd					
	ANN2::neuralnetwork.adam					
	ANN2::neuralnetwork.rmsprop					
4	**brnn**					
5	**CaDENCE**					
6	**deepnet**::gradientdescent					
7	**elmNNRcpp**					
8	**ELMR**					
9	**h2o**::deeplearning					
10	**keras**					
11	**kerasformula**					
12	**kerasR**					
13	**minpack.lm**::nlsLM					
14	**MachineShop**::fit.NNetModel()					
15	**monmlp**::fit.BFGS					
	monmlp::fit.Nelder-Mead					
16	**neural**::mlptrain					
17	**neuralnet**::backprop					
	neuralnet::rprop+					
	neuralnet::rprop-					
	neuralnet::sag					
	neuralnet::slr					
18	**nlsr**::nlxb					
19	**nnet**::nnet.BFGS					
20	**qrnn**::qrnn.fit					
21	**radiant.model**::radiant.model					
22	**rcane**::rlm					
23	**rminer**::fit					
24	**RSNNS**::BackpropBatch					
	RSNNS::BackpropChunk					
	RSNNS::BackpropMomentum					
	RSNNS::BackpropWeightDecay					
	RSNNS::Quickprop					
	RSNNS::Rprop					
	RSNNS::SCG					
	RSNNS::Std-Backpropagation					
25	**ruta**					
26	**simpleNeural**::sN.MLPtrain					
27	**snnR**					
28	**softmaxreg**					
29	**tensorflow**::AdadeltaOptmizer					
	tensorflow::AdagradOptmizer					
	tensorflow::AdamOptmizer					
	tensorflow::FtrlOptmizer					
	tensorflow::GradientDescent					
	tensorflow::MomentumOptmizer					
30	**tfestimators**					
31	**tsensembler**					
32	**validann**::Nelder-Mead					
	validann::BFGS					
	validann::CG					
	validann::L-BFGS-B					
	validann::SANN					
	validann::Brent					

note: Documentation = Docs, Utility Functions = UF, Convergence Quality = CQ, Convergence Time = Time, Final Rating = Final.

Author One

line 1
line 2

Author Two
Affiliation
line 1
line 2