Loan Prediction using Multi-Layer Perceptron (MLP) model in Python & MATLAB

Mohd Arifullah & Mohammad Talib Akhtar

Initial Analysis

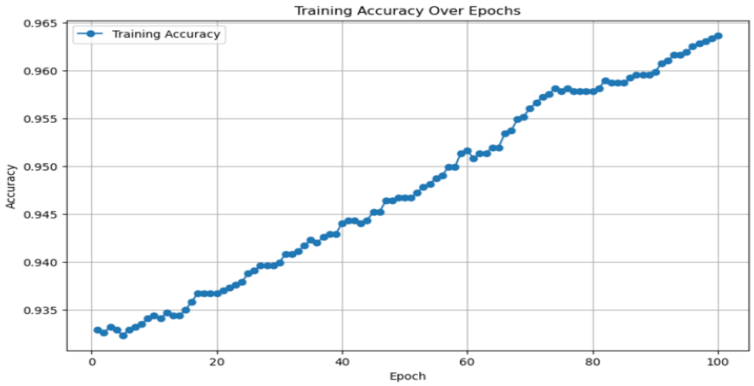
This dataset is ideal for developing predictive models to assess loan eligibility, enabling financial institutions to better gauge credit risk and make informed lending decisions. The dataset contains 4,268 records with 12 attributes related to banking and customer information, such as demographics, financial status, and loan details. Attributes include binary indicators for education and self-employment, continuous variables like income and asset values, and a binary target variable for loan status. We have implemented a Multi-Layer Perceptron with standard Backpropagation algorithm to predict Loan Approval. We have used a Kaggle dataset for this task.

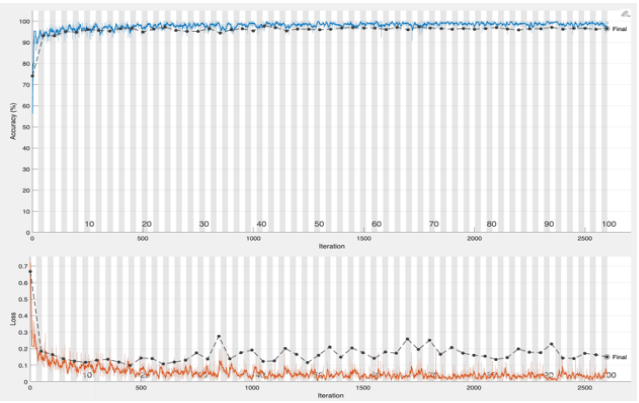
Comparison & Critical Evaluation

	Hidden Sizes	64	64	64	[64, 32]	[64,32]	[64,32]	[64,32,16]	[64,32,16]	[64,32,16]
	Learning Rate	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
Python	Test Set Accuracy	0.96	0.94	0.87	0.95	0.94	0.92	0.96	0.96	0.91
	Training Time /s	0.07	0.04	0.04	0.07	0.07	0.07	0.08	0.08	0.91
MATLAB	Test Set Accuracy	0.96	0.95	0.96	0.95	0.96	0.95	0.62	0.96	0.94
	Training Time /s	7.4	5.5	5.4	5.9	6.1	6.9	6.5	6.5	6.9

Result Analysis

As we can see that Python generally achieves higher accuracy i, e Python: 0.9637, MATLAB: 0.96483 due to differences in numerical computing and optimization routines. Both are sensitive to learning rate changes, with MATLAB showing more pronounced impacts. Python has shorter training times, possibly due to optimized matrix operations. It can be observed that Python exhibits less variability in accuracy, suggesting more stable training or better library support. Increasing hidden layers doesn't consistently improve accuracy in MATLAB, while Python shows a slight trend towards higher accuracy with increased complexity, except for the highest learning rate.





The first image is a Python generated plot of training accuracy increasing steadily over 100 epochs, suggesting good learning progress without any indication of overfitting. The second image, from MATLAB, presents both training and validation accuracy over iterations with minimal overfitting, as indicated by the close performance between training and validation, ending at a validation accuracy of 96.48%.

Lessons learned

When building a loan approval model with neural networks, we learned that setting up the right number of layers and choosing how quickly the model learns is important. We used three layers and tried different speeds of learning, which changed how well the model worked. Also, we found that the model worked differently in Python and MATLAB, probably because of how each programming language does its calculations. This taught us that building a good model means carefully adjusting these settings to get the best results.

Future work

Expanding the dataset, either by acquiring more data or through techniques such as synthetic data generation, could help in building a more robust model. A larger dataset would be particularly useful for training more complex models without overfitting. Implementing systematic approaches to hyperparameter tuning, such as grid search, random search, or Bayesian optimization, could fine-tune the model's performance.

References

[1] Y. Abakarim, M. Lahby and A. Attioui, "Towards An Efficient Real-time Approach To Loan Credit Approval Using Deep Learning," 2018 9th International Symposium on Signal, Image, Video and Communications (ISIVC), Rabat, Morocco, 2018, pp. 306-313, doi: 10.1109/ISIVC.2018.8709173https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8709173&isnumber=8709081

[2] Kumar Arun, Garg Ishan, Kaur Sanmeet "Loan Approval Prediction based on Machine Learning Approach"IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 3, Ver. I (May-Jun. 2016), PP 79-81 www.iosrjournals.org

[3] Hamayel, Mohammad & Moreb, Mohammed & Abumohsen, Mobarak. (2021). Improvement of personal loans granting methods in banks using machine learning methods and approaches in Palestine. 10.1109/ICIT52682.2021.9491636. [6]Andy Liaw and Matthew Wiener. Classification and Regression by randomForest. R News(http://CRAN.R-project.org/doc/Rnews/), 2(3):9–22, 2002.