Molecular Classfication

Andrea Paparella

1 Introduction

1.1 Problem Statement

The classification of molecular structures based on their geometric properties is a critical task in computational chemistry and molecular biology. This project focuses on categorizing molecules into two distinct groups: $functional$ and $nonfunctional$. The categorization is based on their dynamic behavior, as observed in simulations, leveraging their atomic interaction patterns. The underlying motivation for this study is to develop a binary classification model capable of discerning the functionality of molecules. Such models have significant implications in various scientific fields, including drug discovery and materials science.

1.2 State-of-the-Art Approaches

The current landscape in molecular classification is marked by innovative approaches that leverage advanced computational techniques, including logistic regression, deep neural networks (DNNs), and recurrent neural networks, specifically Long Short-Term Memory (LSTM) networks. These methods are suitable at handling and interpreting large datasets of molecular structures.

In molecular dynamics, LSTM has been adapted to model the temporal evolution of chemical and biophysical trajectories. By interpreting these trajectories as sequences of characters, LSTM networks can learn their evolution, effectively capturing both kinetic and thermodynamic aspects. This approach relies on training the LSTM network to understand path entropy, a concept integral to the kinetics and thermodynamics of dynamical trajectories in chemical and biological physics. The practical implementation of this technique has been validated across various test systems, including both simple model molecules and more complex biological systems, demonstrating its reliability and potential in decoding complex stochastic molecular systems. [1]

In the arena of cancer classification, logistic regression has adopted a new approach by using a Bayesian method for gene selection in microarray-based classification. This method addresses the challenge posed by the high number of genes versus limited experimental conditions. It combines logistic regression with gene expression data, employing Gibbs sampling and Markov chain Monte Carlo methods for identifying significant genes. This methodology involves deriving a posterior distribution of selected genes from observed data and then applying logistic regression for classification. What sets this approach apart is its flexibility in the number of genes selected,

[1] https://www.nature.com/articles/s41467-020-18959-8

achieved by assigning a prior distribution over them. This innovation addresses the challenges of high-dimensional data. The efficacy of this Bayesian logistic regression approach is highlighted by its application to large microarray datasets, where it has successfully identified important genes in various cancer types, achieving high classification accuracy and offering a step forward in microarray-based cancer classification.[2]

Deep neural networks (DNNs) have significantly impacted cancer classification by leveraging deep learning to understand complex non-linear relationships between features. This approach has revolutionized the handling of complex data, including the utilization of unlabeled data in cancer classification. A notable development in this field is the transfer learning technique using DNNs. This technique involves feature selection and normalization, coupled with sparse auto-encoders, to enhance feature representation. It utilizes data from different tumor types in an unsupervised manner, leading to improved feature representation. Benchmark datasets have shown that this deep learning-based approach surpasses traditional cancer classification methods in performance.[3]

[2]https://www.sciencedirect.com/science/article/pii/S1532046404000772
[3]https://pubmed.ncbi.nlm.nih.gov/29993662/

2 Method

2.1 Approach and Model Selection

This study investigates two primary machine learning models: Logistic Regression and Long Short-Term Memory (LSTM), to understand their performance in the context of molecular structure analysis. Logistic Regression offers a baseline for performance comparison. It is particularly effective for binary classification problems, making it suitable for classifying molecular structures based on their properties. In contrast, LSTM models excel at handling sequential data, making them a natural choice for datasets containing temporal or sequential patterns. The study compares these models in two scenarios: a smaller dataset with 500 samples and a larger dataset with 20000 samples.

2.2 Data and Preprocessing

The dataset contains simulations of 24 distinct molecules. Based on their atomic interactions and spatial configurations over time, classification will be accomplished.

2.2.1 Dataset Composition

The dataset includes two types of simulation files:

1. **Short Simulation Files (`A000500s*`):** Representing simulations with 500 time steps, these files provide a snapshot of molecular dynamics over a brief duration. Each file contains a 58x500 matrix detailing the x and y coordinates of 29 atoms at each time step.
2. **Long Simulation Files (`A020000s*`):** Analogous to the short simulation files in structure, but extending over 20000 time steps, these files offer a more detailed view of molecular behavior.

Each file's naming convention encodes specific molecular characteristics, leading to 24 unique molecular variations. Using the provided MATLAB script, 312 matrices have been generated for the 20000-sample dataset and 288 matrices for the 500-sample dataset.

2.2.2 Preprocessing Steps

The preprocessing workflow involves several steps:

- **Data Extraction and Labeling:** Flattening each data matrix and labeling molecules as functional or nonfunctional based on predefined criteria.
- **Data Splitting:** Dividing the dataset into training and test sets with a 70:30 ratio.
- **Normalization:** Applying feature scaling using *StandardScaler* to normalize the features.

2.3 Hyperparameter Search Space and Selection

The models' configurations were tuned through a systematic hyperparameter search. For Logistic Regression models the learning rate was set to 1×10^{-4} for the 500 samples dataset and 1×10^{-6} for the 20000 samples dataset. For the LSTM models, the complexity varied from a basic configuration to a more complex setup involving multiple LSTM layers. Learning rate was set to 1×10^{-6} for both LSTM models. These values were chosen to balance the trade-off between learning speed and the risk of overshooting the minimum loss.

3 Results and Discussion

3.1 Model 1: Logistic Regression for 500 samples dataset

3.1.1 Model Description

This model is a simple linear classifier that uses a sigmoid activation function. L2 regularization is applied to prevent overfitting by penalizing large weights.

Parameter	**Value**
Model Type	Logistic Regression
Output Layer Activation	Sigmoid
Kernel Regularization	L2 (0.1)
Optimizer	Adam (0.0001 LR)
Loss Function	Binary Crossentropy
Training Batch Size	64

Table 1: Model 1

3.1.2 Accuracy and Loss Plots

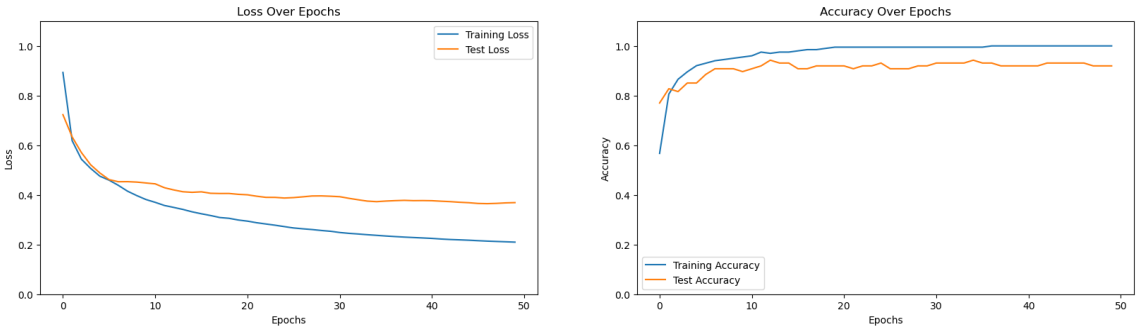

Figure 1: Loss and Accuracy plots for Model 1

3.1.3 Discussion

The model's accuracy on the training data is impressively high, starting at 56.72% and reaching 100% by the 37th epoch. This high level of accuracy is maintained through the remaining epochs. While the validation accuracy starts at a decent 77.01% and improves to 94.25% by the 35th epoch, it then fluctuates and slightly decreases, ending at 91.95%. This plateau and subsequent minor decrease could suggest that the model is not generalizing as effectively to unseen data towards the later epochs. The model shows a steady decline in loss throughout the training and validation process. Starting from an initial loss of 0.8931 on the training set and 0.7230 on the validation set, it ends with a loss of 0.2098 and 0.3688, respectively. This steady reduction indicates that the model is effectively learning and improving its predictions over time.

3.2 Model 2: Logistic Regression for 20000 samples dataset

3.2.1 Model Description

This model is similar to the first one but without regularization. It maintains a linear decision boundary. The use of a larger dataset (20000 time steps) potentially reduces the risk of overfitting, which justifies the absence of regularization.

Parameter	**Value**
Model Type	Logistic Regression
Output Layer Activation	Sigmoid
Optimizer	Adam (0.000001 LR)
Loss Function	Binary Crossentropy
Training Batch Size	64

Table 2: Model 2

3.2.2 Accuracy and Loss Plots

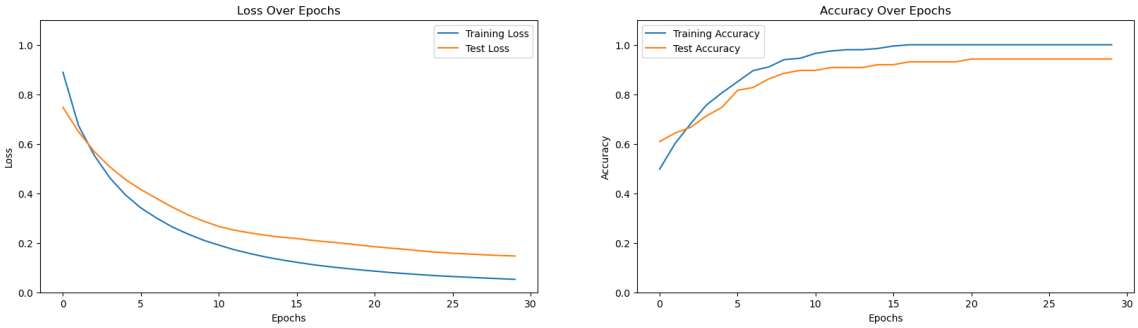

Figure 2: Loss and Accuracy plots for Model 2

3.2.3 Discussion

The model showed a remarkable increase in accuracy during training. Starting at an accuracy of 49.75% in the training set and 60.92% in the validation set, it reached 100% in training accuracy by the 17th epoch and maintained this level until the end. Validation accuracy also improved consistently, reaching 94.25% by the final epoch. The fact that the training accuracy reached 100% while the validation accuracy lagged behind at 94.25% indicates overfitting. The model exhibits a rapid decrease in loss from the first to the last epoch, both in training and validation. The training loss reduced from 0.8890 to 0.0525, and the validation loss went from 0.7472 to 0.1467. This shows that the model is effective in minimizing the error in its predictions as training progressed.

3.3 Model 3: Basic LSTM for 20000 samples dataset

3.3.1 Model Description

This model includes an LSTM layer followed by a fully connected neural network layer.

Parameter	**Value**
Model Type	Basic LSTM
LSTM Units	10
Output Layer Activation	Sigmoid
Optimizer	Adam (0.000001 LR)
Loss Function	Binary Crossentropy
Training Batch Size	64

Table 3: Model 3

3.3.2 Accuracy and Loss Plots

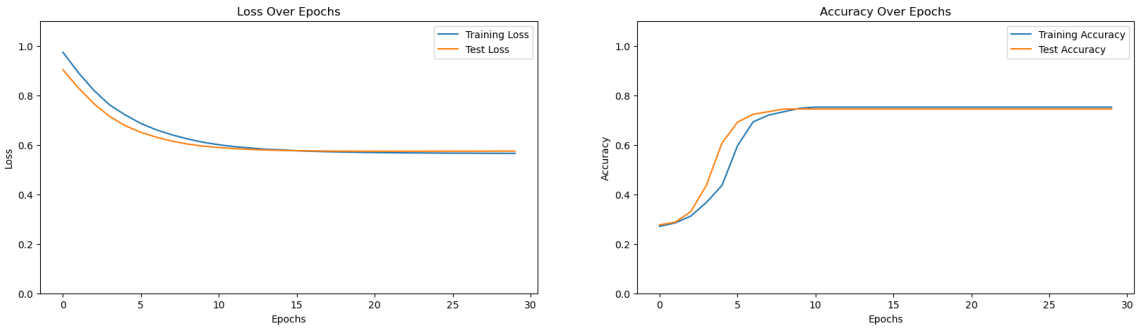

Figure 3: Loss and Accuracy plots for Model 3

3.3.3 Discussion

The accuracy of the model on both training and validation sets increased notably. It started at around 27% for training and 28% for validation in the first epoch and reached approximately 75% for both by the 30th epoch. Around the 10th epoch, both the training and validation accuracy and loss rates began to plateau. This plateau suggests that the model reached its capacity for learning from the provided data. The model shows a consistent decrease in loss over the epochs for both training and validation. Starting at 0.7127 and 0.6936 respectively, the loss figures reduce to 0.6107 and 0.6146 by the 30th epoch.

3.4 Model 4: Complex LSTM for 20000 samples dataset

3.4.1 Model Description

This model included two LSTM layers followed by a fully connected neural network layer.

Parameter	**Value**
Model Type	Complex LSTM
First LSTM Layer Units	10
Second LSTM Layer Units	8
Output Layer Activation	Sigmoid
Optimizer	Adam (0.000001 LR)
Loss Function	Binary Crossentropy

Table 4: Model 4

3.4.2 Accuracy and Loss Plots

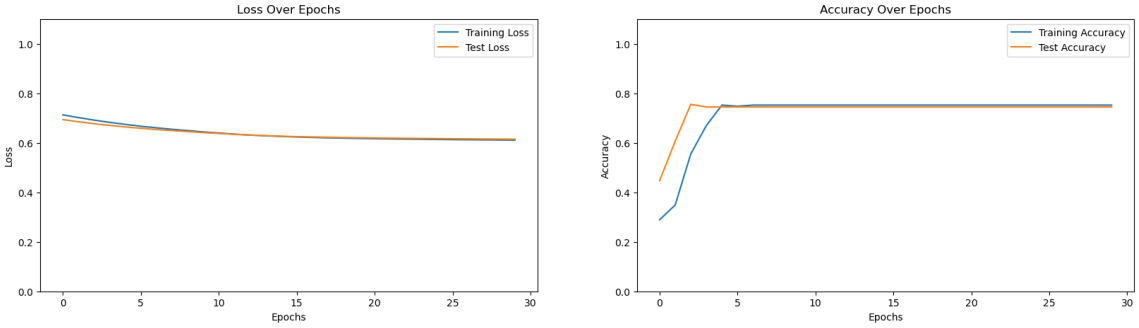

Figure 4: Loss and Accuracy plots for Model 4

3.4.3 Discussion

The model's accuracy improved significantly from the start of training. Initially, the training accuracy was around 29% and the validation accuracy was around 45%. Both improved to

approximately 75% by the end of the training. It also shows a plateau in both accuracy and loss, particularly after the seventh epoch. Both training and validation loss consistently decreased over the epochs.

3.5 Comparative Analysis

Both logistic regression models learn faster and achieve higher accuracies compared to LSTM models. However, they show signs of overfitting. The regularization in Model 1 helps mitigate this to some extent. The LSTM models, especially the complex LSTM, potentially offer better generalization due to their ability to capture temporal dependencies, but they are limited in their learning capacity as shown by the plateau in accuracy. Logistic regression models seem more suitable for situations where a simpler model is adequate, while LSTMs are better for complex sequential data, although with limitations in their current configurations. An important consideration in this analysis is the absence of an LSTM model for the 500-sample dataset, which was not included due to unstable results. This decision underscores the critical requirement of a larger dataset for LSTM models to function effectively, a factor that significantly influences the choice of model in practical scenarios.

4 Conclusion

In this study on molecular classification using Logistic Regression and Long Short-Term Memory models, we have gained valuable insights into the application of machine learning techniques in molecular biology. Logistic Regression models demonstrated high accuracy and rapid learning for simpler classification tasks but tended to overfit, while LSTM models were capable of capturing temporal dependencies in sequential data, although with a slower learning rate and a plateau in accuracy. Future research should emphasize enhancing these models. For Logistic Regression, the incorporation of more robust regularization techniques and the implementation of cross-validation could mitigate overfitting. In the case of LSTM models, exploring more complex architectures, hybrid models, or incorporating attention mechanisms might address the learning plateau and improve their ability to focus on relevant features. Additionally, considering models like Convolutional Neural Networks or Graph Neural Networks, known for their effectiveness in spatial data representation, could provide new insights, especially given the unique characteristics of molecular structure data.