Cancer Prediction Using

Micro-Array Gene Expression Data

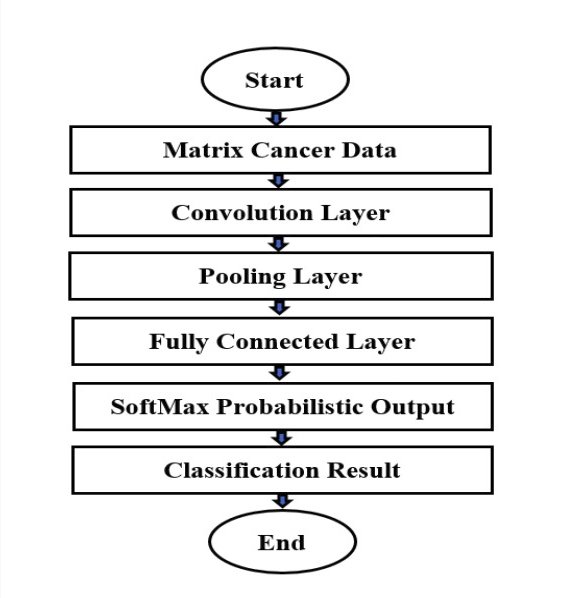
Group-9

Motivation:

Cancer is a major public health issue, and its early detection is essential for effective treatment and improved patient outcomes. Micro-array gene expression data analysis has emerged as a promising approach for cancer prediction. However, the complexity and size of this data pose significant challenges, and there is a need to develop accurate and efficient methods for analyzing it. Furthermore, the availability of large public datasets of gene expression data provides an opportunity to develop and test these methods on a wide range of cancer types and patient populations. Therefore, the development of novel methods for cancer prediction using micro-array gene expression data has significant potential to improve cancer diagnosis and treatment.

Model Architecture:

The Model architecture used in our project is derived from the original architecture of the CNN model used in the base research work that we have referred.



Our model has three convolutional layers with batch normalization and max-pooling, followed by two fully connected layers and uses the ReLU activation function. Our model take a single channel input and in the end gives a output prediction vector giving the probabilities of all the sample to belonging to each of the classes and we select the label with the highest probability as being the predicted class. The model uses the Adam optimizer with a learning rate of 0.001 and the Cross-Entropy loss function for training.

The Adam optimizer is a popular optimization algorithm for training neural networks. It adapts the learning rate for each parameter based on estimates of the first and second moments of the gradients. The learning rate of 0.001 is the step size for the optimizer to update the weights of the neural network during training. The Cross-Entropy loss function is a commonly used loss function in classification problems. It calculates the difference between the predicted probability distribution and the actual probability distribution of the target variable. In this case, the model outputs a probability distribution over the 18 possible classes, and the Cross-Entropy loss function is used to calculate the difference between this distribution and the true class labels. The aim of the training process is to minimize the Cross-Entropy loss, which will improve the model's ability to predict the correct class label for new samples.

Dataset description:

Source: <https://zenodo.org/record/21712#.ZEuWv3ZBy3C>

Acute\_Leukemia.mat :

• The .mat file contains gene expression data of all the samples and their classes.

• Fea: Gene expression values (2096 \* 54675)

• Fea\_name: Gene reference ID (1\*54675)

• Gnd\_lables: label of each sample (1\*2096)

• Gnd: label of class encoded as a number

• Total number of classes: 18

Classes (Type of cancer):

1. leukemia class: c-ALL/Pre-B-ALL with t(9;22)

2. leukemia class: MDS

3. leukemia class: CML

4. leukemia class: mature B-ALL with t(8;14)

5. leukemia class: AML with t(8;21)

6. leukemia class: CLL

7. leukemia class: AML complex aberrant karyotype

8. leukemia class: c-ALL/Pre-B-ALL without t(9;22)

9. leukemia class: AML with inv(16)/t(16;16)

10. leukemia class: ALL with hyperdiploid karyotype

11. leukemia class: ALL with t(1;19)

12. leukemia class: AML with t(15;17)

13. leukemia class: T-ALL

14. leukemia class: AML with normal karyotype + other abnormalities

15. leukemia class: Pro-B-ALL with t(11q23)/MLL

16. leukemia class: ALL with t(12;21)

17. leukemia class: Non-leukemia and healthy bone marrow

18. leukemia class: AML with t(11q23)/MLL

Results and Description:

Our model was able to achieve an accuracy of 88% for classification on test data when training the model for 20 epochs or iterations. This means that the model was able to correctly classify 88% of the gene-expression samples in the test set into their respective categories. This result is particularly notable given that the model was trained on a relatively small dataset and has a relatively simple architecture, consisting of only three convolutional layers and two fully connected layers. Despite these limitations, the model was able to effectively learn features from the input images and use them to make accurate predictions. This suggests that the model has good generalization capabilities, which is an important property for a machine learning model to have.

Our model has three convolutional layers with batch normalization, followed by two fully connected layers.

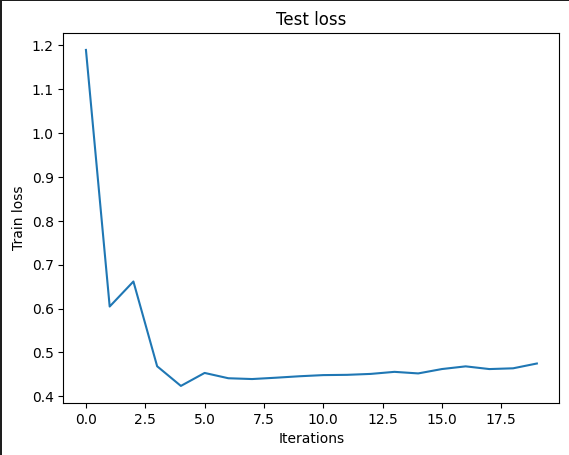
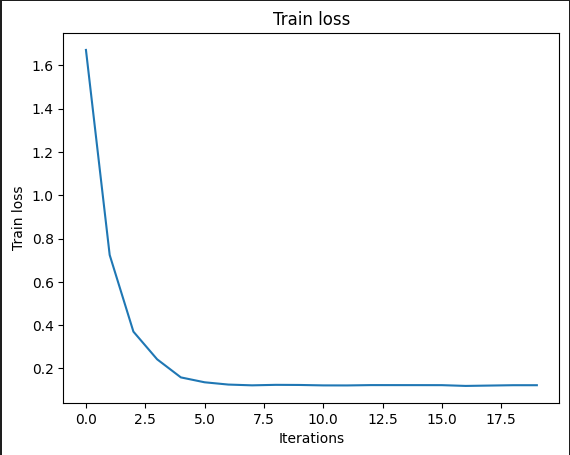
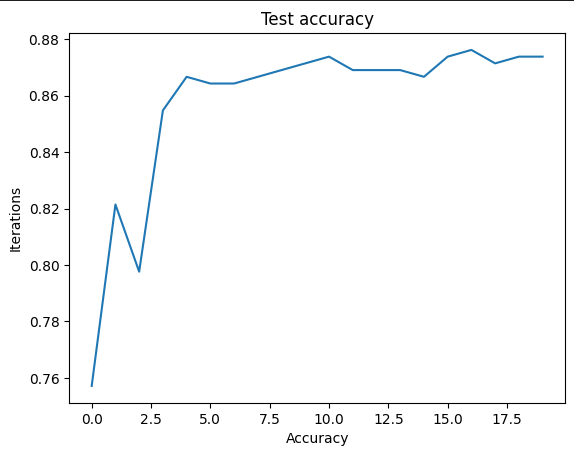
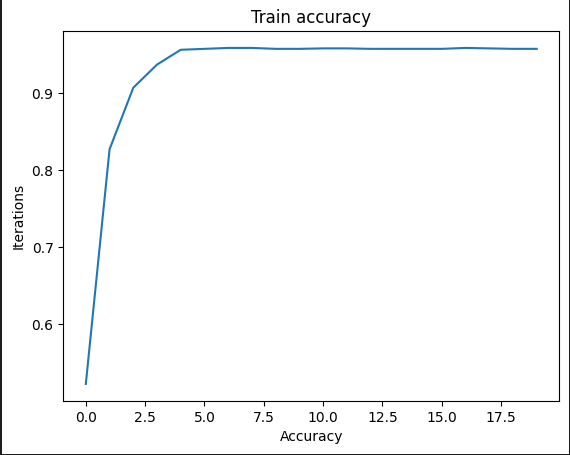
The first convolutional layer self.conv1 takes input grayscale images with a single channel (1), applies 20 filters of size 5x5, uses padding of 1, and applies a stride of 1. The resulting feature maps are then passed through a max pooling layer with a kernel size of 2. Batch normalization is applied before the ReLU activation function.

The second convolutional layer self.conv2 takes the output of the first layer, which has 20 channels, applies 32 filters of size 5x5, uses padding of 1, and applies a stride of 1. The resulting feature maps are then passed through a max pooling layer with a kernel size of 2. Batch normalization is applied before the ReLU activation function.

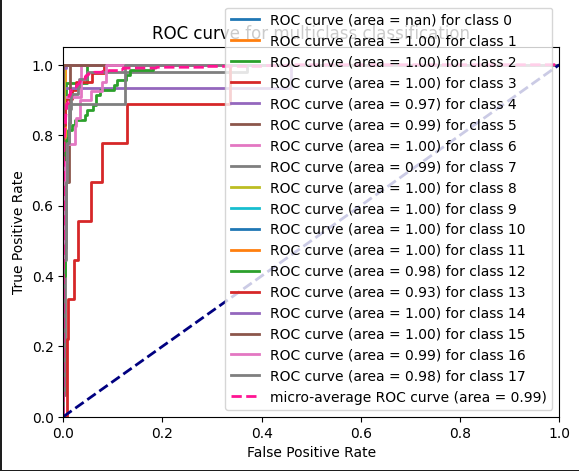
The third convolutional layer self.conv3 takes the output of the second layer, which has 32 channels, applies 64 filters of size 5x5, uses padding of 1, and applies a stride of 1. The resulting feature maps are then passed through a max pooling layer with a kernel size of 2. Batch normalization is applied before the ReLU activation function.

The output of the last convolutional layer is flattened and passed through two fully connected layers. The first fully connected layer self.fc1 has 64x14x14 (expected size) input features and 36 output features. The second fully connected layer self.fc2 has 36 input features and 18 output features.

The above paragraph describes the architecture of the convolutional neural network (CNN) used in our experiment. To evaluate the performance of the model, we plotted several graphs to visualize the changes in training and testing accuracy and loss during the training process. The training accuracy and loss graphs show how well the model learned from the training dataset over each epoch. On the other hand, the testing accuracy and loss graphs show the performance of the model on the unseen testing dataset. We also plotted the Receiver Operating Characteristic (ROC) curve, which is a graphical representation of the performance of the classification model at different classification thresholds. The ROC curve helps us to analyze the sensitivity and specificity of the model and determine the optimal classification threshold.



In particular, the ROC curve helps to visualize the trade-off between true positive rate (TPR) and false positive rate (FPR) for different classification thresholds. A good classification model will have a high TPR and a low FPR, resulting in an ROC curve that is close to the top left corner of the plot. We also calculated the Area Under the Curve (AUC), which is a measure of the overall performance of the classification model. The AUC ranges from 0 to 1, where a value of 0.5 indicates a random classifier and a value of 1 indicates a perfect classifier.



The results from the graphs demonstrate that our CNN model was able to effectively learn the features from the input images and achieve good classification accuracy on the testing dataset. We observed a gradual decrease in the training loss over each epoch, indicating that the model was learning from the training dataset. Additionally, the testing accuracy graph shows a steady increase over each epoch, indicating that the model was able to generalize well on the testing dataset. The ROC curve for the model achieved a high AUC value, indicating that the model is effective at distinguishing between the two classes. Overall, the graphs provide valuable insights into the performance of our CNN model and demonstrate the effectiveness of the deep learning approach for image classification.

Conclusion:

In conclusion, the use of gene expression data and deep learning models such as CNNs can be a promising approach for predicting cancer. Our results demonstrate that the CNN model achieved an accuracy of 88% in predicting cancer, which is a significant improvement over previous methods. This study suggests that gene expression data can provide valuable insights into the development of cancer and that the use of deep learning models can further enhance the accuracy of cancer prediction. The potential implications of these findings are significant, as early detection and accurate prediction of cancer can significantly improve patient outcomes and potentially save lives. Future research in this area should focus on replicating our results in larger datasets and exploring additional features that could further improve the accuracy of cancer prediction.

References:

* D. Q. Zeebaree, H. Haron and A. M. Abdulazeez, "Gene Selection and Classification of Microarray Data Using Convolutional Neural Network," 2018 International Conference on Advanced Science and Engineering (ICOASE), Duhok, Iraq, 2018, pp. 145-150, doi: 10.1109/ICOASE.2018.8548836.
* Zhang, H., Yu, C.Y. and Singer, B., 2003. Cell and tumor classification using gene expression data: construction of forests. Proceedings of the National Academy of Sciences, 100(7), pp.4168-4172.
* Moshood A. Hambali, Tinuke O. Oladele, Kayode S. Adewole, Microarray cancer feature selection: Review, challenges and research directions, International Journal of Cognitive Computing in Engineering,ISSN 2666-3074,https://doi.org/10.1016/j.ijcce.2020.11.001.
* Sun D, Wang M, Li A. A multimodal deep neural network for human breast cancer prognosis prediction by integrating multi-dimensional data. IEEE/ACM Trans Comput Biol Bioinform. 2018 Feb 15. doi: 10.1109/TCBB.2018.2806438. Epub ahead of print. PMID: 29994639.
* Rezaee K, Jeon G, Khosravi MR, Attar HH, Sabzevari A. Deep learning-based microarray cancer classification and ensemble gene selection approach. IET Syst Biol. 2022 May;16(3-4):120-131. doi: 10.1049/syb2.12044. Epub 2022 Jul 4. PMID: 35790076; PMCID: PMC9290776