Brain cancer gene expression using Neural Networks

Presented by:

Shivika Prasanna

Ahmad Alhonainy

Abdulmateen Adebiyi

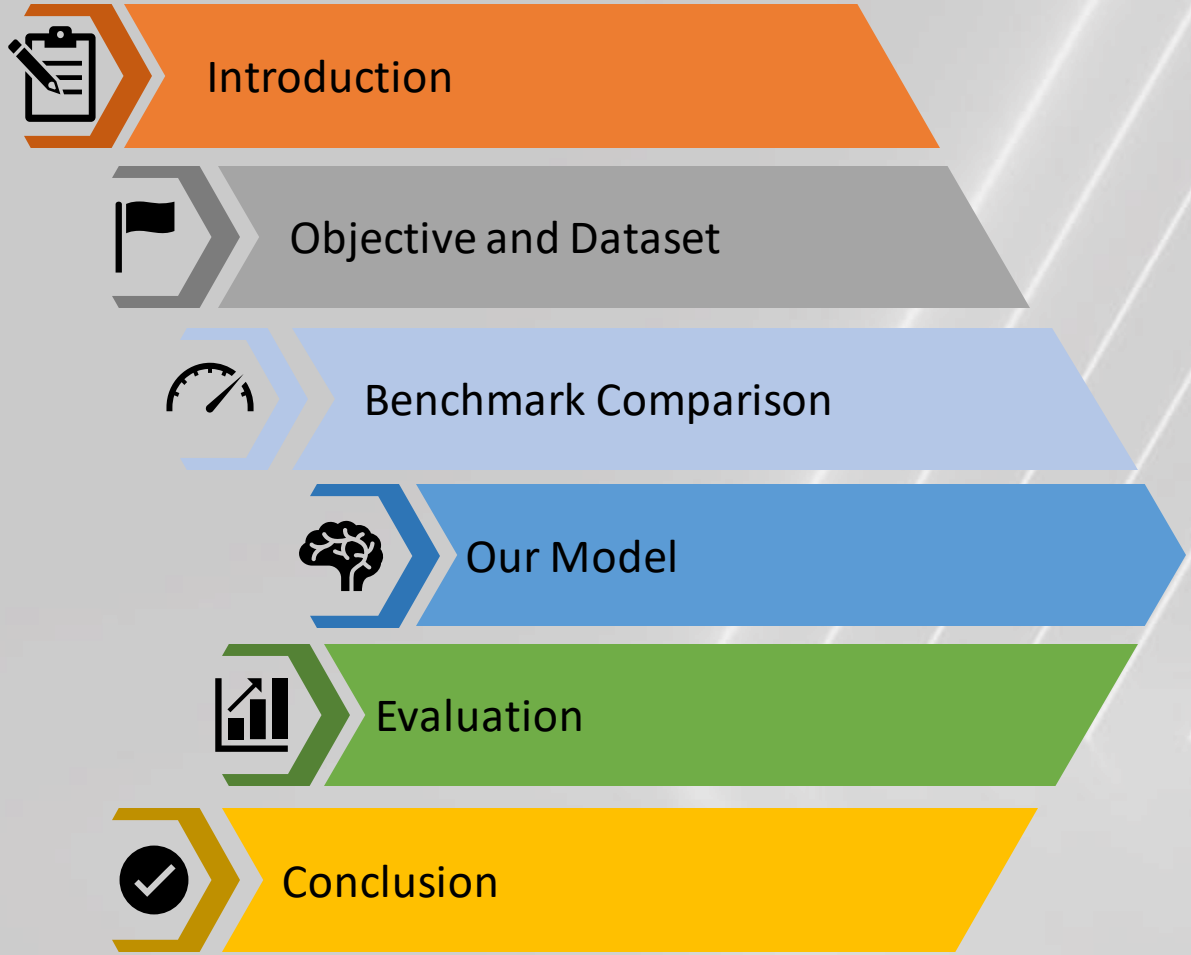
Hulayyil Alshammari

For Machine Learning for Biomedical Informatics

Images have been taken from sources cited on the slides.

Code available on: https://github.com/ShivikaPrasanna/brain_cancer_gene_expression.git

Outline:


Introduction
Objective and Dataset
Benchmark Comparison
Our Model
Evaluation
Conclusion

Introduction

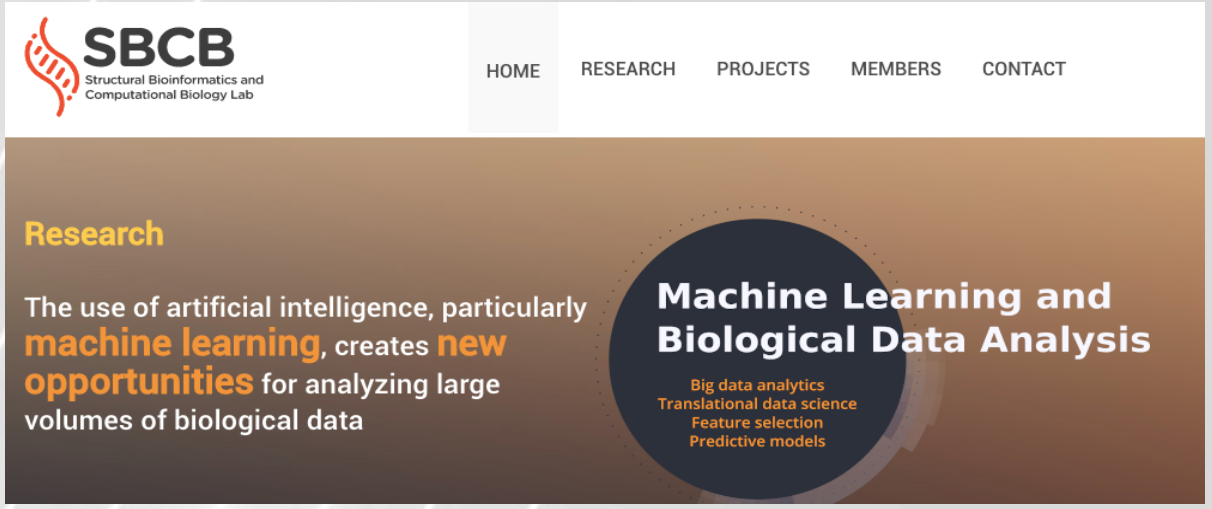
- Level of gene expression depends on factors
  - Type of cell
  - Developmental stage of the organism
  - Presence of external signals or environmental stimuli

- Regulation of gene expression critical for maintaining proper balance of proteins and other molecules within cells

Objective and Dataset

Goal:
Predict the class of gene expression in Brain Cancer dataset

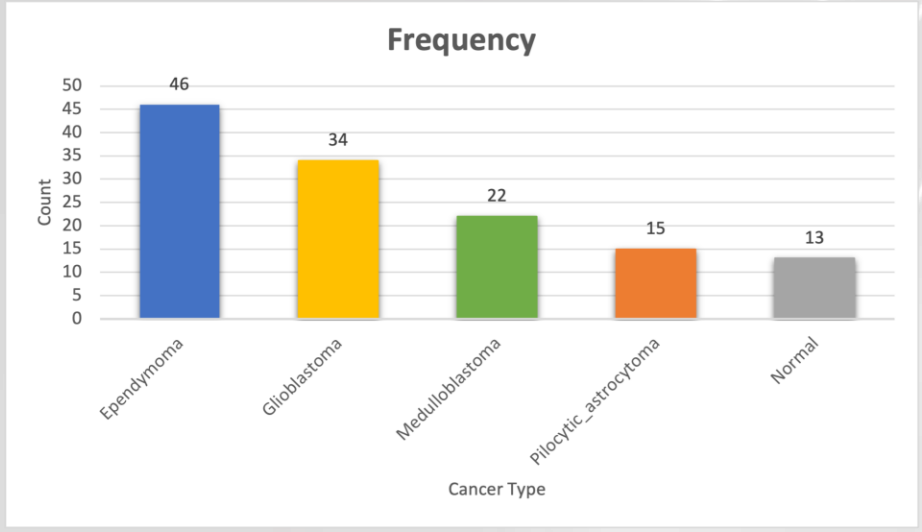
- SBCB lab focuses on the computational analysis of biological structures
- Curated Microarray Database (CuMiDa) containing 78 cancer datasets curated from 30,000 studies
  - Provide standardized preprocessing and benchmark results
  - Facilitate machine learning studies in cancer research



Source: https://sbcb.inf.ufrgs.br/

Objective and Dataset

- GSE50161 on brain cancer gene expression:
  - 5 classes
  - 54,676 genes
  - 130 samples
- Distribution of the 5 classes:



	samples	type	1007_s_at	1053_at	117_at	121_at	1255_g_at	1294_at	1316_at	1320_at	...
0	834	ependymoma	12.498150	7.604868	6.880934	9.027128	4.176175	7.224920	6.085942	6.835999	...
1	835	ependymoma	13.067436	7.998090	7.209076	9.723322	4.826126	7.539381	6.250962	8.012549	...
2	836	ependymoma	13.068179	8.573674	8.647684	9.613002	4.396581	7.813101	6.007746	7.178156	...
3	837	ependymoma	12.456040	9.098977	6.628784	8.517677	4.154847	8.361843	6.596064	6.347285	...
4	838	ependymoma	12.699958	8.800721	11.556188	9.166309	4.165891	7.923826	6.212754	6.866387	...
...	...	...	...	...	...	...	...	...	...	...	...
125	959	pilocytic_astrocytoma	12.658228	8.843270	7.672655	9.125912	5.495477	8.603892	7.747514	5.828978	...
126	960	pilocytic_astrocytoma	12.812823	8.510550	8.729699	9.104402	3.967228	7.719089	7.092496	6.504812	...
127	961	pilocytic_astrocytoma	12.706991	8.795721	7.772359	8.327273	6.329383	8.550471	6.613332	6.308945	...
128	962	pilocytic_astrocytoma	12.684593	8.293938	7.228186	8.494428	6.049414	8.214729	7.287758	5.732710	...
129	963	pilocytic_astrocytoma	12.397722	8.843524	8.825100	8.551541	5.002072	8.547894	6.920827	5.738159	...

130 rows × 54677 columns

```
1  data.samples.count()
```

```
130
```

```
1  data.type.unique()
```

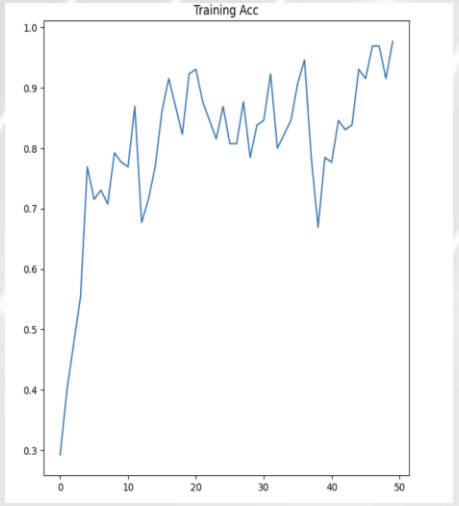
```
array(['ependymoma', 'glioblastoma', 'medulloblastoma', 'normal',
       'pilocytic_astrocytoma'], dtype=object)
```

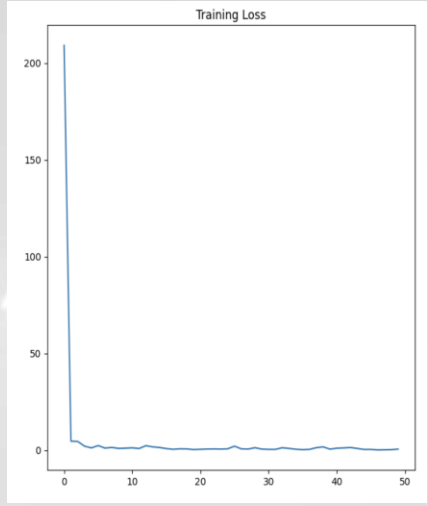
Source: https://sbcb.inf.ufrgs.br/

Benchmark Comparison

- Model:
  - Train-test split of *100-0*
  - ResMLP (Sequential)
  - Dense layer (Activation: ReLU)
  - Batch normalization
- Optimizer: Adam
- Loss: Sparse Categorical Cross Entropy
- Model Training Parameters:
  - Epochs: 50
  - Batch size: 16

```
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                multiple                  55988224

batch_normalization (BatchN  multiple                  4096
ormalization)

sequential (Sequential)      (None, 4096)              153147392

dense_23 (Dense)             multiple                  20485

=================================================================
Total params: 209,160,197
Trainable params: 209,158,149
Non-trainable params: 2,048
_________________________________________________________________
```





Source: https://www.kaggle.com/code/jiaowoguanren/brain-cancer-gene-expression-tf-resmlp-0-98

Our Model

- Neural Network
- Model definition:
  - Train-test split of 90-10 due to small size
  - Sequential model
  - 4 dense layers (16, 8, 4, 1 units)
  - Activations:
    - Layers 1-3: ReLU
    - Layer 4: Sigmoid

```
# Define the model architecture
model = Sequential()
model.add(Dense(16, activation='relu', input_dim=X_train.shape[1]))
model.add(Dense(8, activation='relu'))
model.add(Dense(4, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Our Model

- Loss: Binary Cross Entropy
- Optimizer: Adam
- Metrics: Accuracy

```
# Compile the model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Our Model

- Model Training Parameters
  - Early Stopping
    - Monitor: loss
    - Patience: 10
    - Mode: min
  - Model Checkpoint
    - Monitor: loss
    - Save best model with min loss
  - Learning Rate
    - Monitor: loss
    - Factor: 0.1
    - Mode: min

```
earlyStopping = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=10, verbose=0, mode='min', restore_best_weights=True)
mcp_save = tf.keras.callbacks.ModelCheckpoint('.mdl_wts.hdf5', save_best_only=True, monitor='loss', mode='min')
reduce_lr_loss = tf.keras.callbacks.ReduceLROnPlateau(monitor='loss', factor=0.1, patience=7, verbose=1, min_delta=1e-4, mode='min')
```

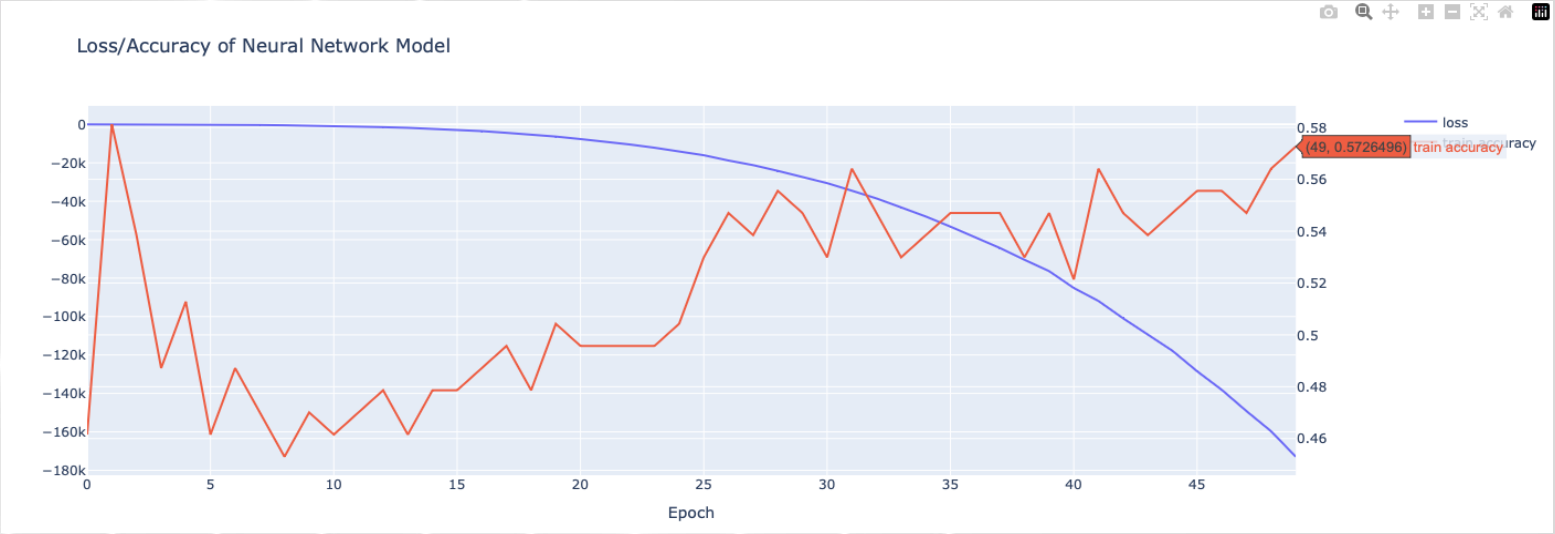
Model Summary

```
1  model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 16)                874832

 dense_1 (Dense)             (None, 8)                 136

 dense_2 (Dense)             (None, 4)                 36

 dense_3 (Dense)             (None, 1)                 5

=================================================================
Total params: 875,009
Trainable params: 875,009
Non-trainable params: 0
_________________________________________________________________
```

Evaluation

Metrics	Value
Training Accuracy	0.57
Test Accuracy	0.61
Epochs	50
Batch size	16




```
score, acc = model.evaluate(X_test, y_test, batch_size=32, verbose=2)
print("Accuracy: ", acc)
```

```
1/1 - 0s - loss: -1.3199e+05 - accuracy: 0.6154 - 99ms/epoch - 99ms/step
Accuracy:  0.6153846383094788
```

Conclusion

- Neural networks can be used to classify the gene expression
  - Into 3 classes for Brain Cancer dataset due to extreme class imbalance, namely ependymoma, glioblastoma, medulloblastoma
- Data was split into 90-10 due to dataset size
- Model developed had fewer layers to accommodate the data size
- Multiple experiments were carried out to finalize the best model
  - SMOTE for additional synthetic data points yielded similar test accuracy of 59%
  - Smaller model of 2 layers yielded test accuracy of 60%
- Hyperparameter tuning on the batch size, epochs and split ratio