ASSIGNMENT ON GENOMIC SEQUENCE

Submitted by Akshaya Karthikeyan | Roll# 20171016

Models Implemented:

1. 2D Convolutional Neural Network

2. LSTM

3. Bi-LSTM

---

1. Methodology and Implementation Details:

2D Convolutional Neural Network:

Flow structure:

- Extracting the sequences
- Embedding the sequences using One Hot Encoder and Label Encoder
- Making data loaders
- Creating the model along with parameters
- Training the model
- Choosing the best model through validation
- Plotting loss vs epochs graphs
- Calculating accuracy, F1 score and AUPRC
- Testing

Number of parameters:

```
The model has 427,134 trainable parameters
```

Hyperparameters:

num_epochs = 40

batch_size = 5

learning rate: 0.001

momentum: 0.9

Output dimension: 2

Model summary:

```
Cnn(
  (conv1): Conv2d(1, 6, kernel_size=(1, 5), stride=(1, 1))
  (pool1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, 
ceil_mode=False)
  (conv2): Conv2d(6, 16, kernel_size=(1, 5), stride=(1, 1))
  (fc1): Linear(in_features=3552, out_features=120, bias=True)
  (fc2): Linear(in_features=120, out_features=2, bias=True)
  (softmax): Softmax(dim=1)
)
```

Loss: CrossEntropyLoss

Optimizer: Stochastic Gradient Descent

LSTM:

Flow structure:

- Extracting the sequences
- Embedding the sequences using Label Encoder
- Making data loaders
- Creating the model along with parameters
- Training the model
- Choosing the best model through validation
- Plotting loss vs epochs graphs
- Calculating accuracy, F1 score and AUPRC
- Testing

Number of parameters:

```
The model has 2,163,802 trainable parameters
```

Hyperparameters:

num_epochs = 40

batch_size = 5

learning rate: 0.0005

momentum: 0.9

Input dimension: 900

hidden dimension: 400

Number of layers: 1

Output dimension: 2

Model summary:

```
LSTMModel(
  (solv_lstm): LSTM(900, 400, batch_first=True)
  (fc1): Linear(in_features=400, out_features=200, bias=True)
  (fc2): Linear(in_features=200, out_features=2, bias=True)
  (softmax): Sigmoid()
)
```

Loss: CrossEntropyLoss

Optimizer: Stochastic Gradient Descent

Bi-LSTM:

Flow structure:

- Extracting the sequences
- Embedding the sequences using Label Encoder
- Making data loaders
- Creating the model along with parameters
- Training the model
- Choosing the best model through validation
- Plotting loss vs epochs graphs
- Calculating accuracy, F1 score and AUPRC
- Testing

Number of parameters:

The model has 4,327,002 trainable parameters

Hyperparameters:

num_epochs = 40

batch_size = 5

learning rate: 0.0005

momentum: 0.9

Input dimension: 900

hidden dimension: 400

Number of layers: 1

Output dimension: 2

Model summary:

```
LSTMModel(
  (solv_lstm): LSTM(900, 400, batch_first=True, bidirectional=True)
  (fc1): Linear(in_features=800, out_features=200, bias=True)
  (fc2): Linear(in_features=200, out_features=2, bias=True)
  (softmax): Sigmoid()
)
```

Loss: CrossEntropyLoss

Optimizer: Stochastic Gradient Descent

---

2. Performance:

2D Convolutional Neural Network:

Accuracy: 80%

F1 score: 0.74866

AUPRC: 0.74866

LSTM:

Accuracy: 78%

F1 score: 0.71964

AUPRC: 0.71964

Bi-LSTM:

Accuracy: 78%

F1 score: 0.71964

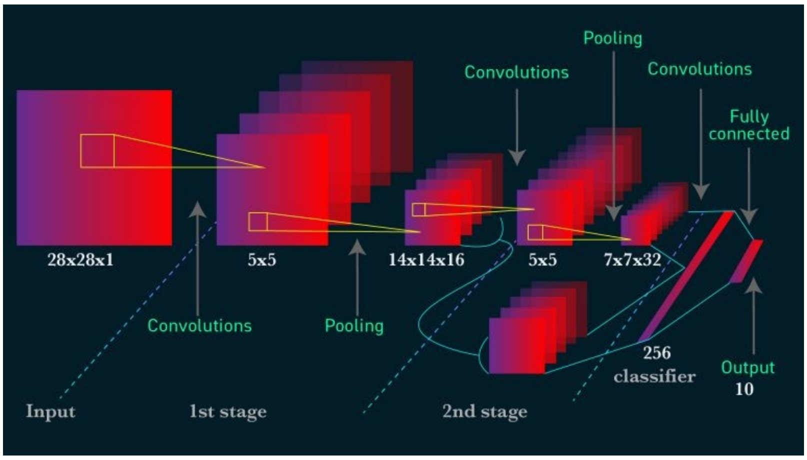
AUPRC: 0.71964

---

3. Rationale for design decisions:

2D Convolutional Neural Network:

In machine learning, Convolutional Neural Networks (CNN or ConvNet) are complex feed forward neural networks. CNNs are used for classification and recognition because of its high accuracy. The CNN follows a hierarchical model which works on building a network, like a funnel, and finally gives out a fully-connected layer where all the neurons are connected to each other and the output is processed.

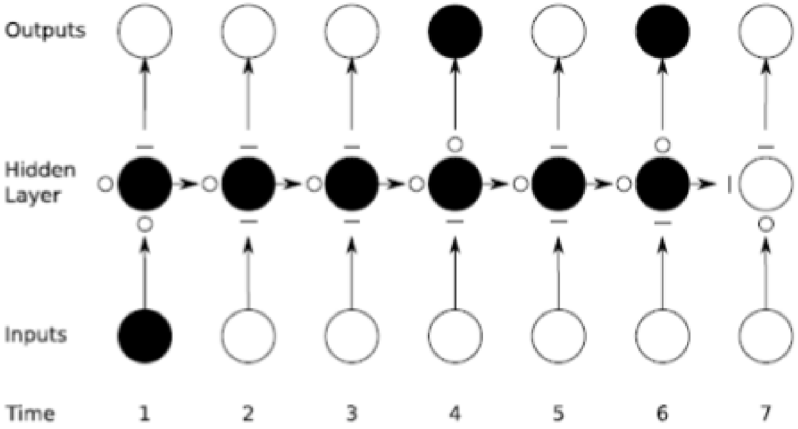


LSTM:

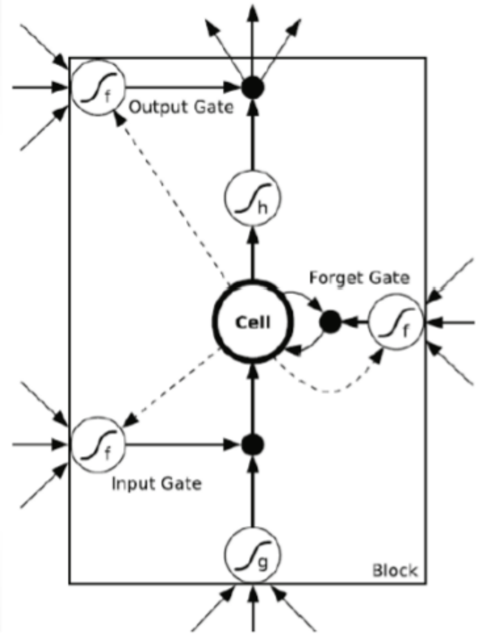
LSTM is basically considered to avoid the problem of vanishing gradient in RNN. It also works very well for long sequences.

An LSTM allows the preservation of gradients. The memory cell remembers the first input as long as the forget gate is open and the input gate is closed.

The output gate provides finer control to switch the output layer on or off without altering the cell contents.

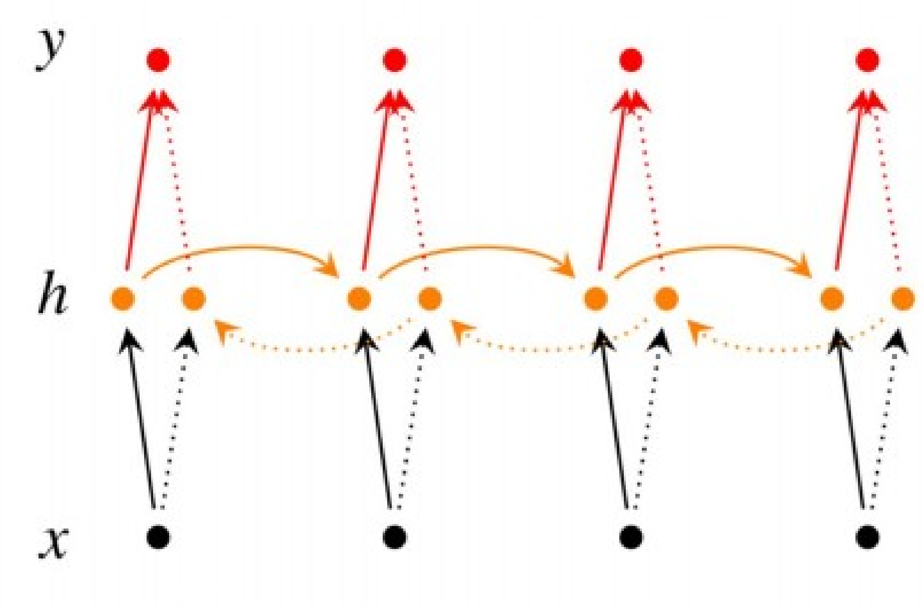


Internal architecture of the LSTM is:



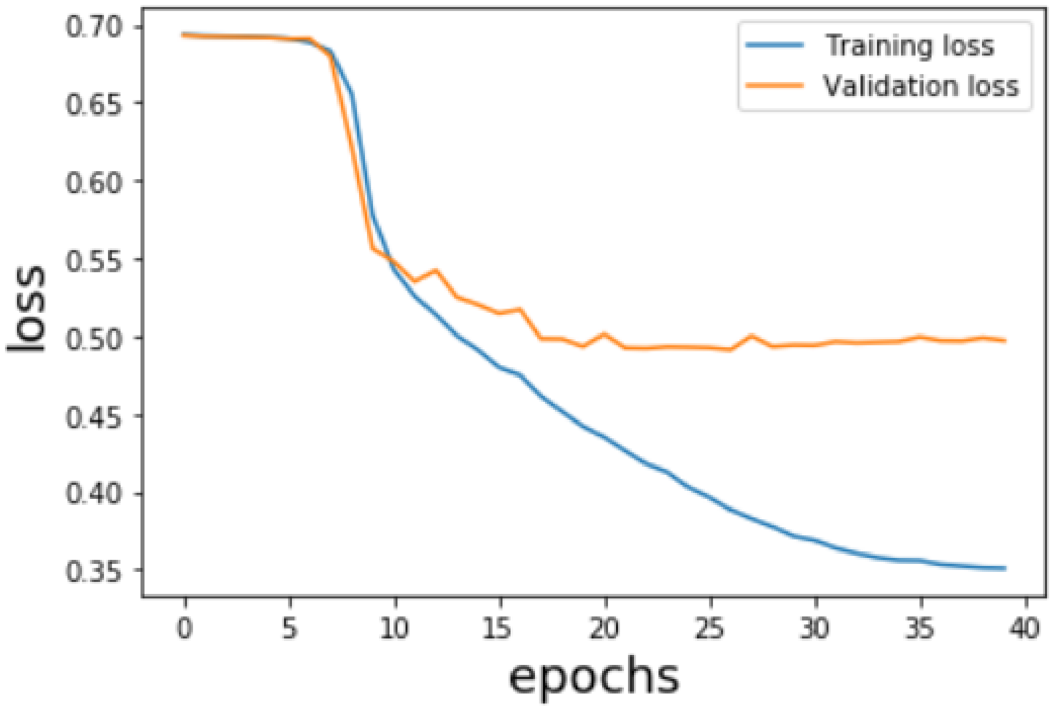
Bi-LSTM:

This is a bidirectional LSTM. It has two networks, one access information in forward direction and another access in the reverse direction. These networks have access to the past as well as the future information and hence the output is generated from both the past and future context. Therefore it remembers more of the long sequences.

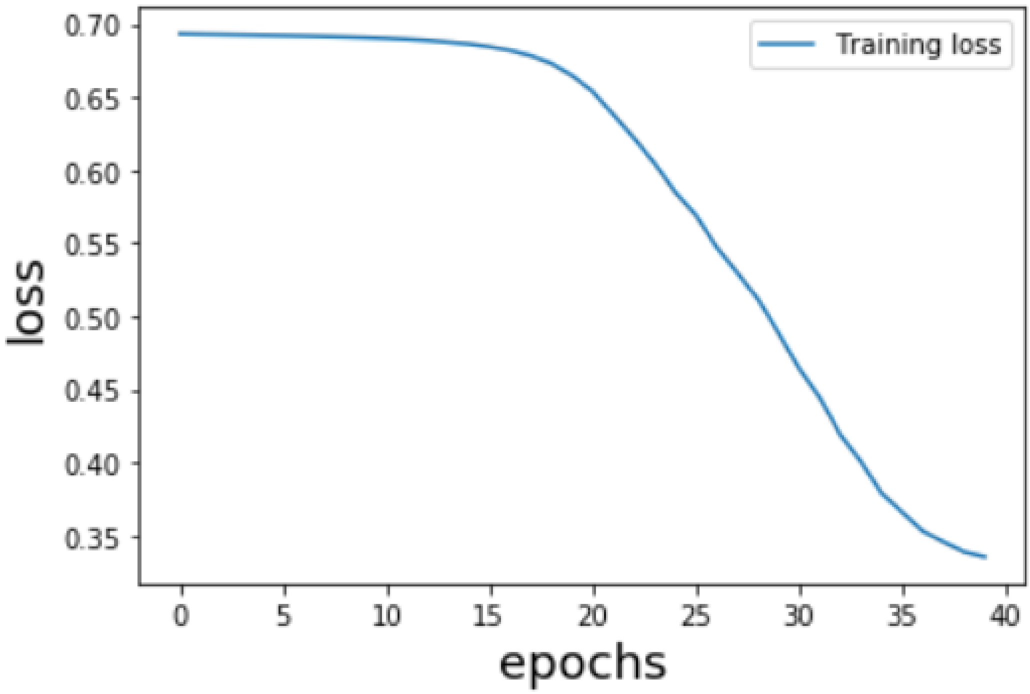


4. Training curve (loss vs epochs):

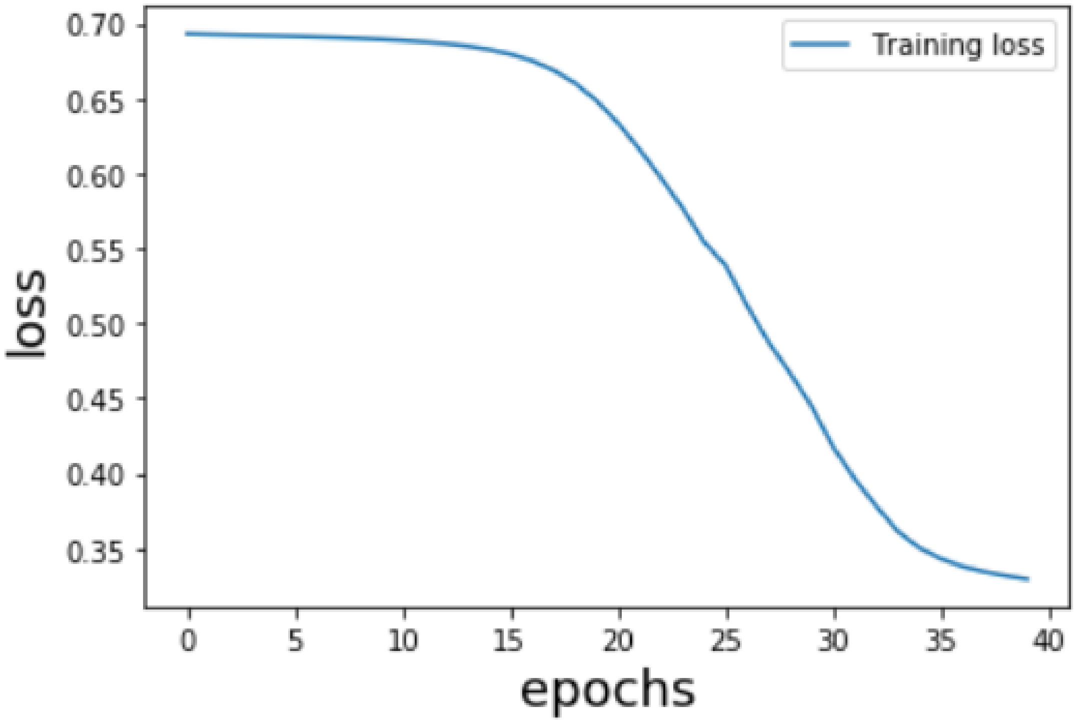
2D Convolutional Neural Network:


0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
loss
0
5
10
15
20
25
30
35
40
epochs
Training loss
Validation loss

LSTM:


0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
loss
0
5
10
15
20
25
30
35
40
epochs
Training loss

Bi-LSTM:



5. Predictions on test set:

Given in the test_results folder.