Memory Trace Analysis Using Machine Learning

Presented by:
Braeden Slade
post-bacc, HPC-DES, USRC

LA-UR-20-26022

Mentor:
Nathan DeBardeleben

Outline

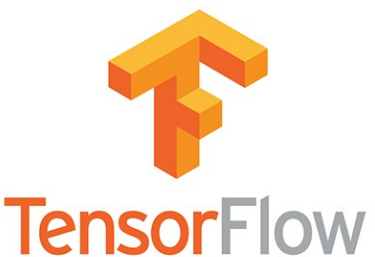
- Tools Used
- Basic Overview of the data
- Prediction Types
  - Classifications vs Regression
- Methods of Predicting
  - Data Splitting
  - Trees
  - RNN
- Results Using Trees
  - Read or Write(Classification)
  - Address(Regression)
- Results Using RNNS
  - Read or Write
- Clustering
- Conclusion/Future Work

Outline

- Tools Used
- Basic Overview of the data
- Prediction Types
  - Classifications vs Regression
- Methods of Predicting
  - Data Splitting
  - Trees
  - RNN
- Results Using Trees
  - Read or Write(Classification)
  - Address(Regression)
- Results Using RNNS
  - Read or Write
- Clustering
- Conclusion/Future Work

Tools used

- Data Analysis
  - Pandas
  - Spark
  - Seaborn
  - Pyspark
  - Numpy
- Machine Learning
  - Sklearn
  - Pyspark Machine Learning
  - Tensorflow
  - Keras



Outline

- Tools Used
- Basic Overview of the data
- Prediction Types
  - Classifications vs Regression
- Methods of Predicting
  - Data Splitting
  - Trees
  - RNN
- Results Using Trees
  - Read or Write(Classification)
  - Address(Regression)
- Results Using RNNS
  - Read or Write
- Clustering
- Conclusion/Future Work

Understanding the data

- Where does it come from?
  - ArmIE Executed
    - Run applications chosen
    - HPC benchmark traces
  - Binary trace file

Understanding the data

- Where does it come from?
  - ArmIE Executed
    - Run applications we choose
    - HPC benchmark traces
  - Binary trace file

	trace_id	trace_state	is_sve	is_write	size_in_bytes	addr_int
0	15	Start	False	False	0	0
1	15	Tracing	False	False	16	281474976676336
2	15	Tracing	False	False	8	281474976676912
3	15	Tracing	False	False	16	281474976676928
4	15	Tracing	False	False	8	281474976676984
...	...	...	...	...	...	...
1049381	15	Tracing	False	False	4	281474976676504
1049382	15	Tracing	False	True	8	281474976676328
1049383	15	Tracing	False	False	8	281474976676328
1049384	15	Tracing	False	False	4	4408096
1049385	15	End	False	False	0	0

Outline

- Tools Used
- Basic Overview of the data
- Prediction Types
  - Classifications vs Regression
- Methods of Predicting
  - Data Splitting
  - Trees
  - RNN
- Results Using Trees
  - Read or Write(Classification)
  - Address(Regression)
- Results Using RNNS
  - Read or Write
- Clustering
- Conclusion/Future Work

Classification

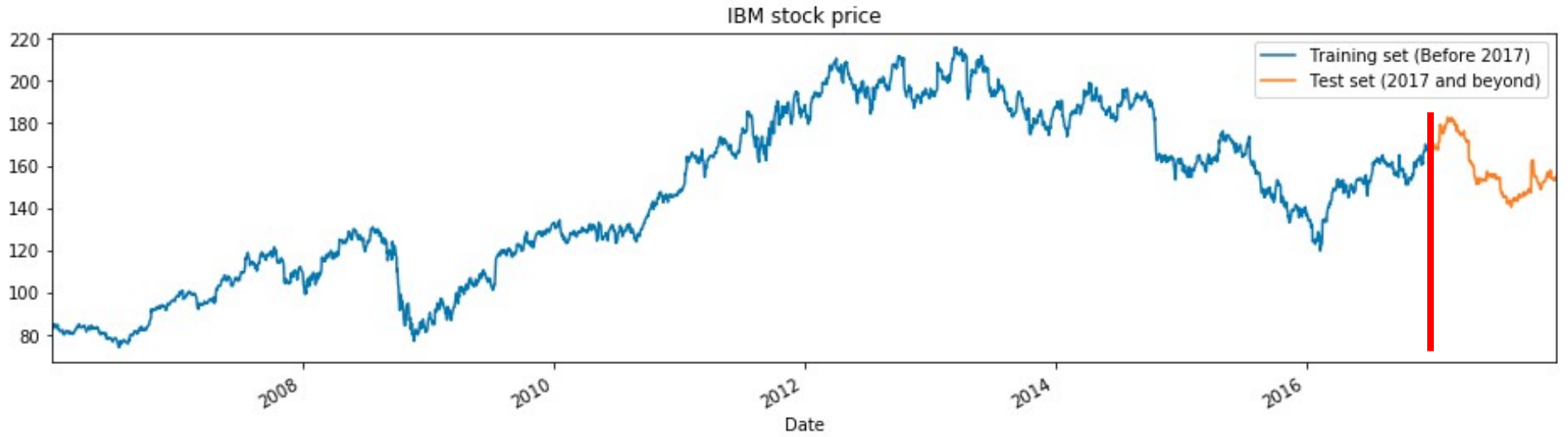
- Probability falls into certain category
- Dog vs Cat
- Read vs Write
- Accuracy: (0-1)

Is this a cat or a dog?

Cats vs Dogs

Regression

- Numerical values
- Value of Stock
- Accuracy: RMSE
- Address Accessed



Outline

- Tools Used
- Basic Overview of the data
- Prediction Types
  - Classifications vs Regression
- Methods of Predicting
  - Data Splitting
  - Trees
  - RNN
- Results Using Trees
  - Read or Write(Classification)
  - Address(Regression)
- Results Using RNNS
  - Read or Write
- Clustering
- Conclusion/Future Work

Data Splitting

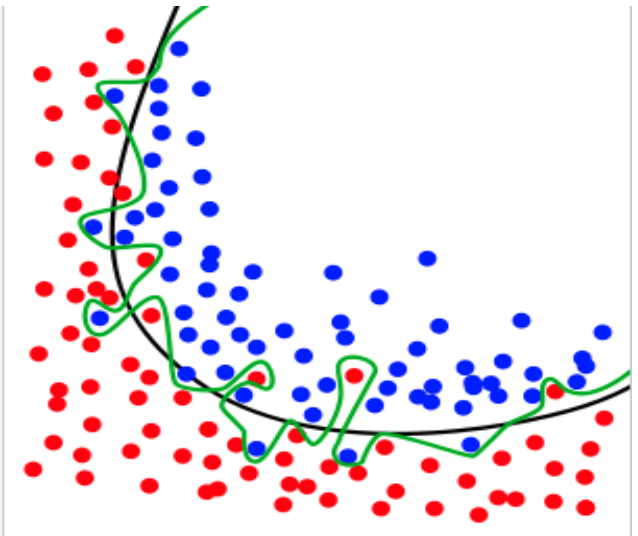
- Single Trace used

- 70/30 Random Split

Data Splitting

- Single Trace used

- 70/30 Random Split

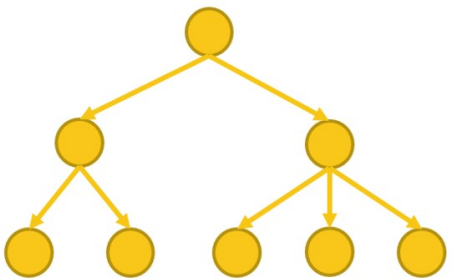


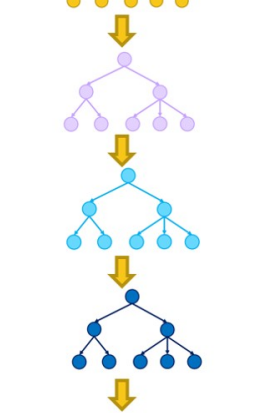
- Overfitting possible
- Overfits based on data, will not perform well on new data

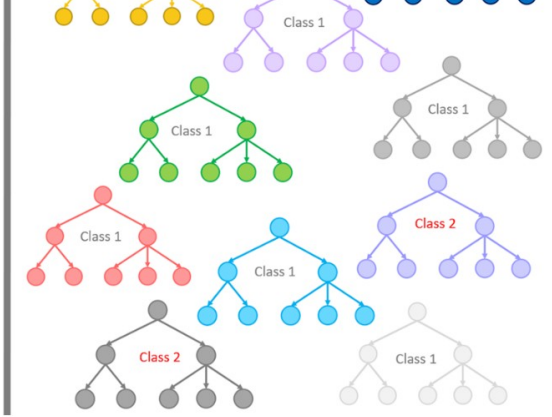
Green = Overfitting
Black = Model we want

Methods of Testing (Trees)

- Trees
  - Decision Trees
  - Gradient Boosted Trees
  - Random Forests

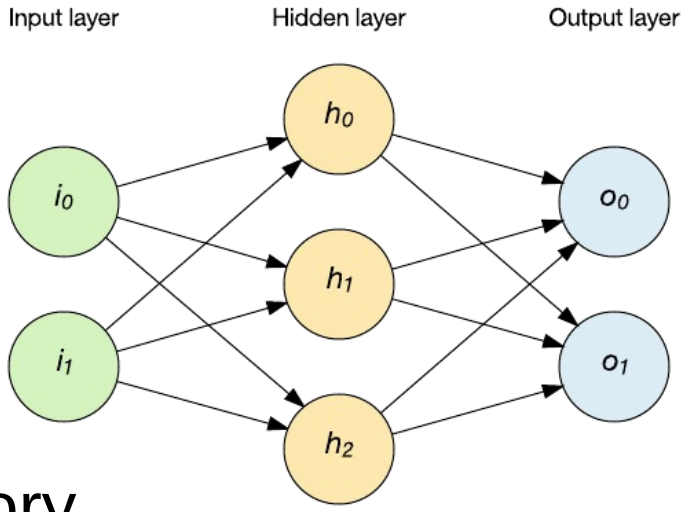






Methods of Testing (RNN)

- RNN(Recurrent Neural Networks)
  - Previous inputs predict future inputs
- LSTM Neural Networks (Future)
  - RNN with better “Logical Memory management”



Example “Memory Management”

- the clouds are in the…

Example “Memory Management”

- the clouds are in the (sky),

RNN =

LSTM =

Example 2 “Memory Management”

- I grew up in The United States. I am 22 years old. I play soccer. I speak fluent…

Example 2 “Memory Management”

- I grew up in The United States. I am 22 years old. I play soccer. I speak fluent (English)

RNN =

LSTM =

Outline

- Tools Used
- Basic Overview of the data
- Prediction Types
  - Classifications vs Regression
- Methods of Predicting
  - Data Splitting
  - Trees
  - RNN
- Results Using Trees
  - Read or Write(Classification)
  - Address(Regression)
- Results Using RNNS
  - Read or Write
- Clustering
- Conclusion/Future Work

Predict Read/Write (Classification)

```
DTC
0.5935198086247566
RFC
0.8219450537957972
GBT
0.8219450537957972
```

	is_sve	size_in_bytes	addr_int
0	0	0	0
1	0	16	281474976676336
2	0	8	281474976676912
3	0	16	281474976676928
4	0	8	281474976676984
...	...	...	...
1049381	0	4	281474976676504
1049382	0	8	281474976676328
1049383	0	8	281474976676328
1049384	0	4	4408096
1049385	0	0	0

- Function calculates Accuracies
- Trees
  - Decision Tree Classifier: 59%
  - Random Forest Classifier: 82%
  - Gradient Boosted Tree: 82%

Predict Read/Write (Classification)

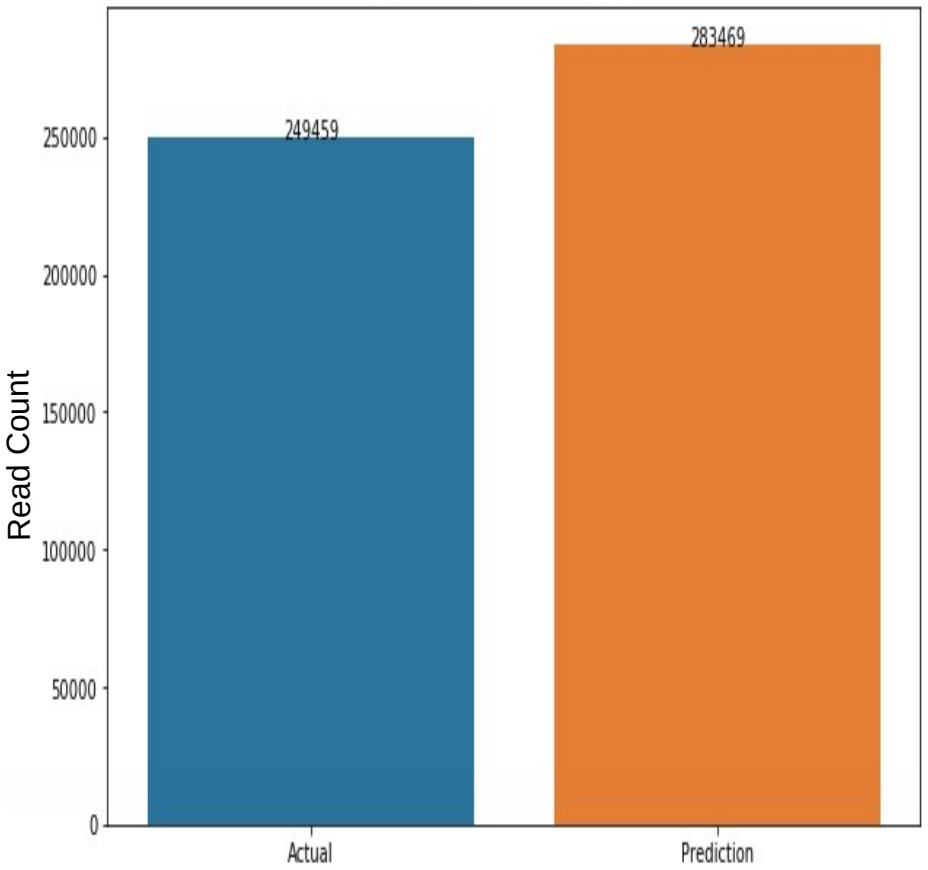
```
DTC
0.5935198086247566
RFC
0.8219450537957972
GBT
0.8219450537957972
Feature Influence on label column
(3,[0,1,2],[0.34430663818548224,0.5695918210358937,0.0861015407786240
Most influencial column is: addr_int with a value of: 0.5695918210358
```

- Train on all the columns to predict read or write
- Which column had the most influence

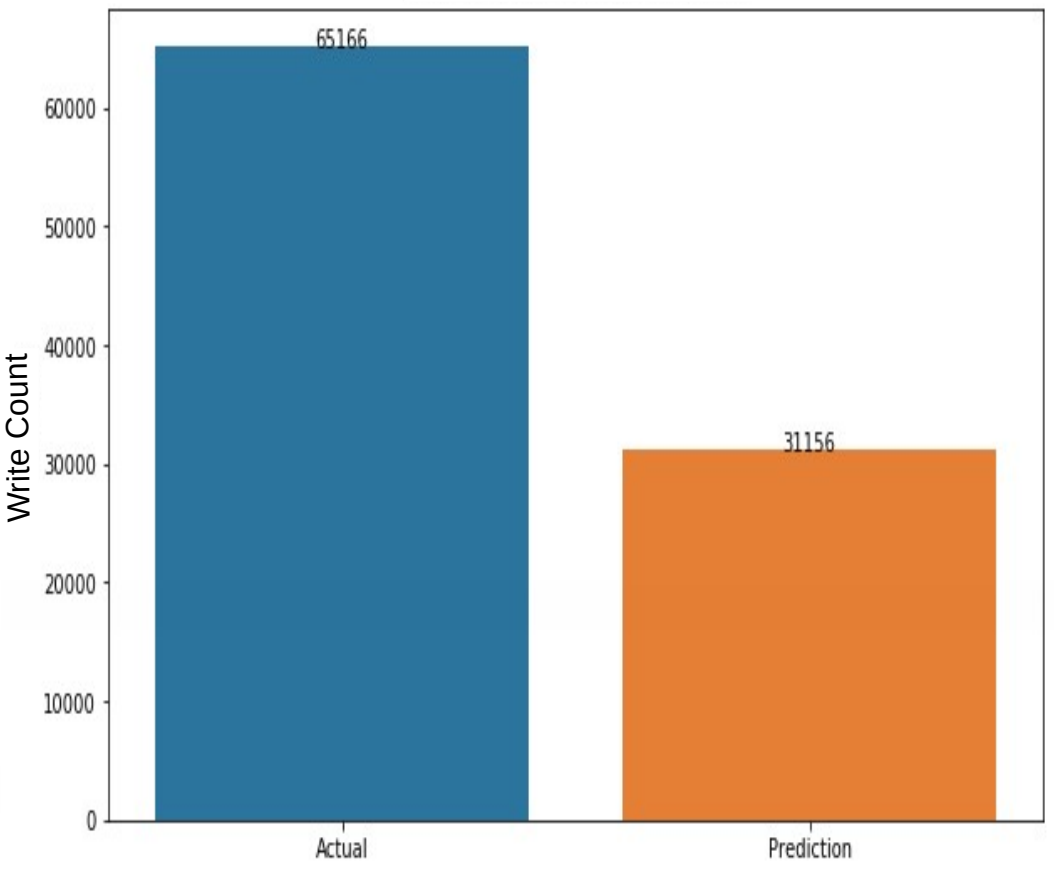
	is_sve	size_in_bytes	addr_int
0	0	0	0
1	0	16	281474976676336
2	0	8	281474976676912
3	0	16	281474976676928
4	0	8	281474976676984
...	...	...	...
1049381	0	4	281474976676504
1049382	0	8	281474976676328
1049383	0	8	281474976676328
1049384	0	4	4408096
1049385	0	0	0

Predict Read/Write (Classification)

Actual vs Predicted Reads



Actual vs Predicted Write



Outline

- Tools Used
- Basic Overview of the data
- Prediction Types
  - Classifications vs Regression
- Methods of Predicting
  - Data Splitting
  - Trees
  - RNN
- Results Using Trees
  - Read or Write(Classification)
  - Address(Regression)
- Results Using RNNS
  - Read or Write
- Clustering
- Conclusion/Future Work

Predict Address (Regression)

```
Linear Regression Checks
Root Mean Squared Error:  97486276721688.88
R-Squared:  0.09924256091497718
```

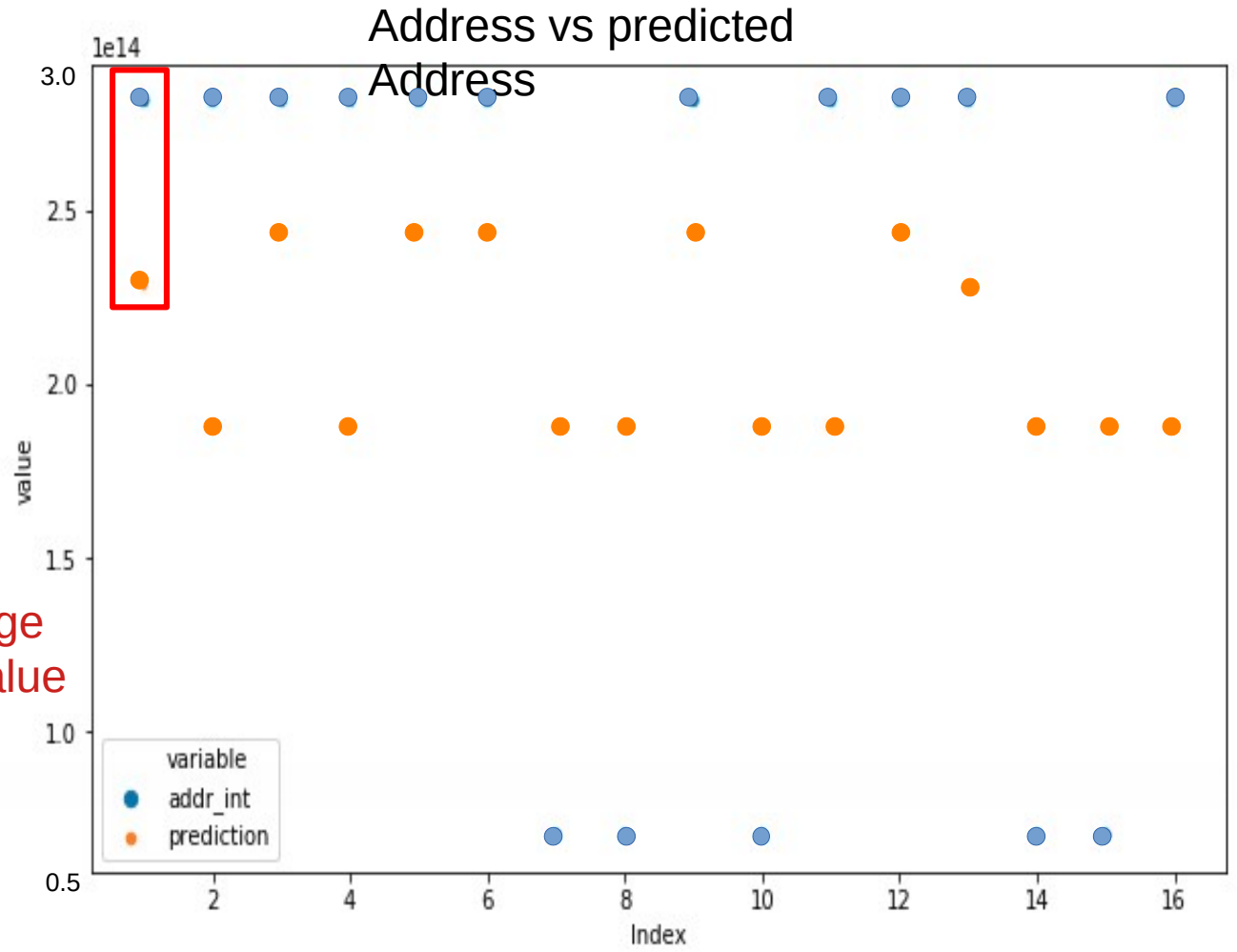
- Regression Specific Information

Predict Address (Regression)

```
Linear Regression Checks
Root Mean Squared Error:  97486276721688.88
R-Squared:  0.09924256091497718
DTR
95902893908584.58
RFR
96739102296671.98
GBT
96739102296671.98
Feature Influence on label column
(3,[0,1,2],[0.27508678657457536,0.5763997614364617,0.14851345198896299])
Most influencial column is: is_write with a value of: 0.5763997614364617
```

- Same structure as Read and Write
  - Trees
  - Most influential feature

Predict Address (Regression)



16 individual random addresses.

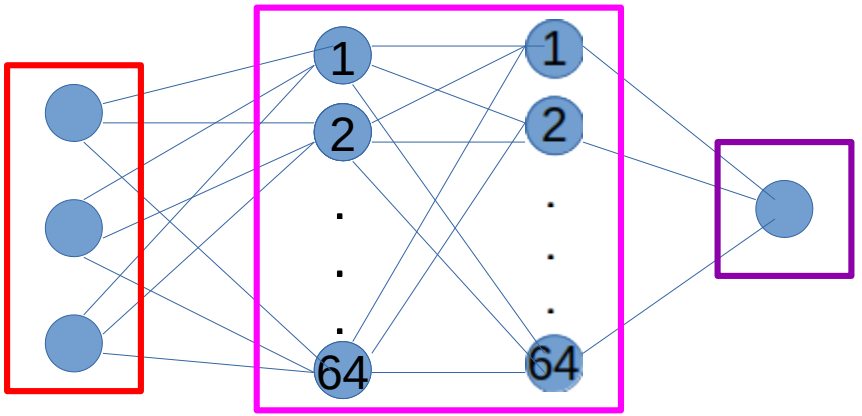
Blue = Actual
Orange = predicted

NOTE: Huge address value range 100 Trillions

Outline

- Tools Used
- Basic Overview of the data
- Prediction Types
  - Classifications vs Regression
- Methods of Predicting
  - Data Splitting
  - Trees
  - RNN
- Results Using Trees
  - Read or Write(Classification)
  - Address(Regression)
- Results Using RNNS
  - Read or Write
- Clustering
- Conclusion/Future Work

Predict is_write (RNN)

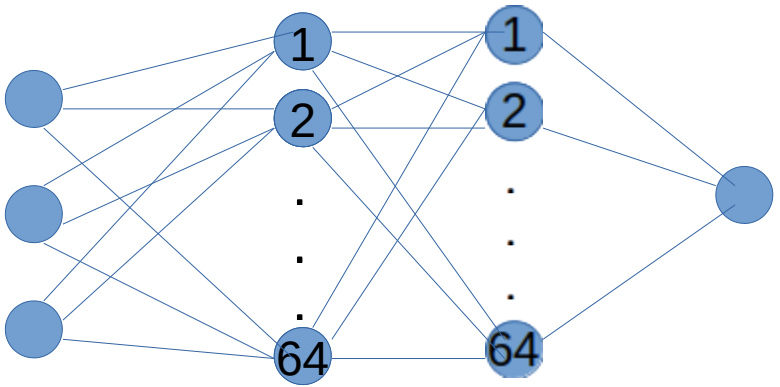


Red = Input Layer (is_sve, size_in_bytes, address)

Pink = Hidden Layer (nonlinear transformations on inputs)

Purple = Output Layer

Predict is_write (RNN)




```
In [29]: EPOCHS = 1

history = model.fit(
  normed_train_data, train_labels,
  epochs=EPOCHS, verbose=1,
  callbacks=[tfdocs.modeling.EpochDots()])
```

```
22931/22956 [============================>.] - ETA: 0s - loss: 0.4541 - accuracy: 0.7990
Epoch: 0, accuracy:0.7990,  loss:0.4541,
22956/22956 [==============================] - 20s 864us/step - loss: 0.4541 - accuracy: 0.7990
```

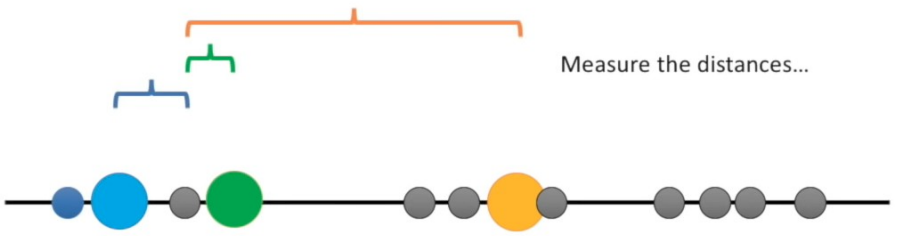
- RNN = 79.9% Accuracy

Outline

- Tools Used
- Basic Overview of the data
- Prediction Types
  - Classifications vs Regression
- Methods of Predicting
  - Data Splitting
  - Trees
  - RNN
- Results Using Trees
  - Read or Write(Classification)
  - Address(Regression)
- Results Using RNNS
  - Read or Write
- Clustering
- Conclusion/Future Work

Clustering

- Clustering
  - K-means most effective
  - Several Others
    - Affinity Propagation
    - SVM
    - DBSCAN



K-Means in Action

```
centers = km.cluster_centers_
print(centers)
```

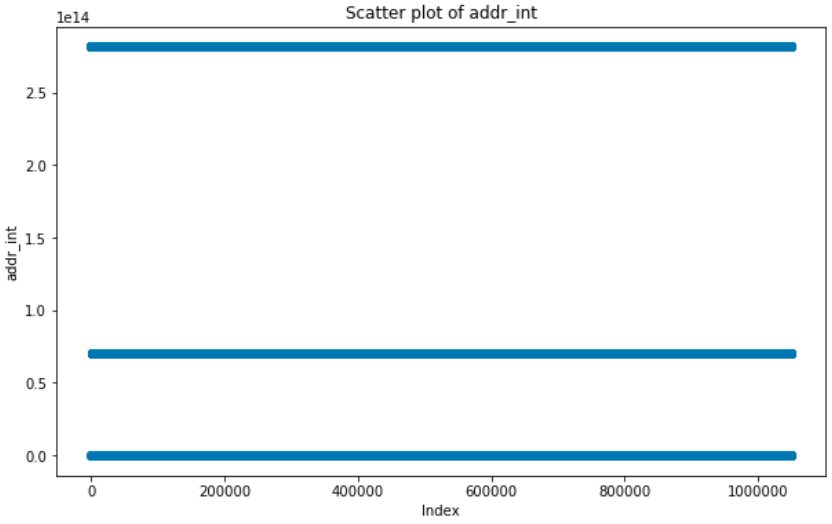
```
[[4.52325600e+06]
 [2.81474977e+14]
 [7.03687779e+13]]
```

- Provide number of clusters

K-Means in Action

```
centers = km.cluster_centers_
print(centers)
```

```
[[4.52325600e+06]
 [2.81474977e+14]
 [7.03687779e+13]]
```

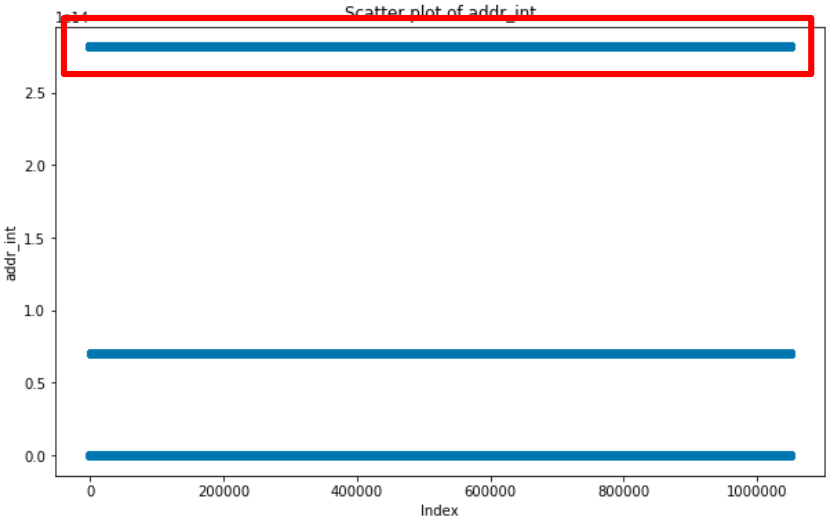


- Graph Shows Addresses Accessed

K-Means in Action

```
centers = km.cluster_centers_
print(centers)
```

```
[[4.52325600e+06]
 [2.81474977e+14]
 [7.03687779e+13]]
```

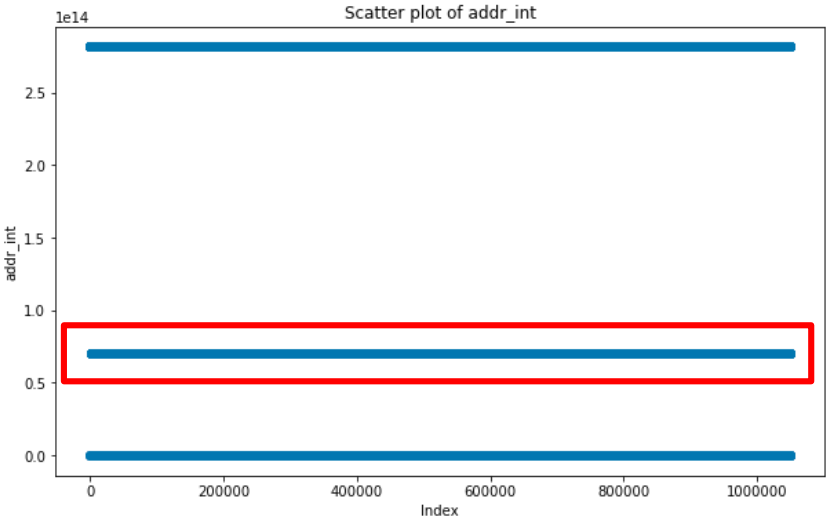


- Centers Match with graph

K-Means in Action

```
centers = km.cluster_centers_
print(centers)
```

```
[[4.52325600e+06]
 [2.81474977e+14]
 [7.03687779e+13]]
```

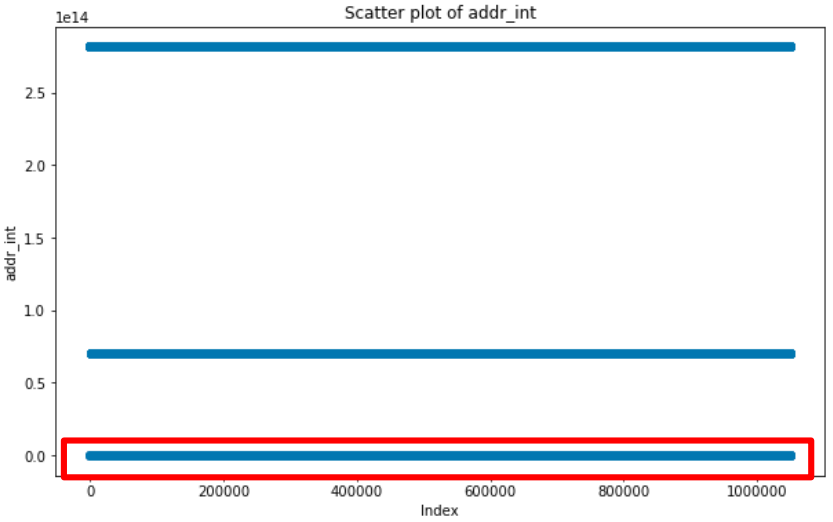


- Centers Match with graph

K-Means in Action

```
centers = km.cluster_centers_
print(centers)
```

```
[[4.52325600e+06]
 [2.81474977e+14]
 [7.03687779e+13]]
```



- Centers Match with graph

Outline

- Tools Used
- Basic Overview of the data
- Prediction Types
  - Classifications vs Regression
- Methods of Predicting
  - Data Splitting
  - Trees
  - RNN
- Results Using Trees
  - Read or Write(Classification)
  - Address(Regression)
- Results Using RNNS
  - Read or Write
- Clustering
- Conclusion/Future Work

To Conclude

- Early Stages
  - Reasonable predictions on individual traces
  - One Trace is not sufficient

- Moving Forward
  - Different NNs (LSTM)
  - Alter and expand training and testing data