Project Description

Contents

Deliverables

- Working phyton scripts that meet the objectives.

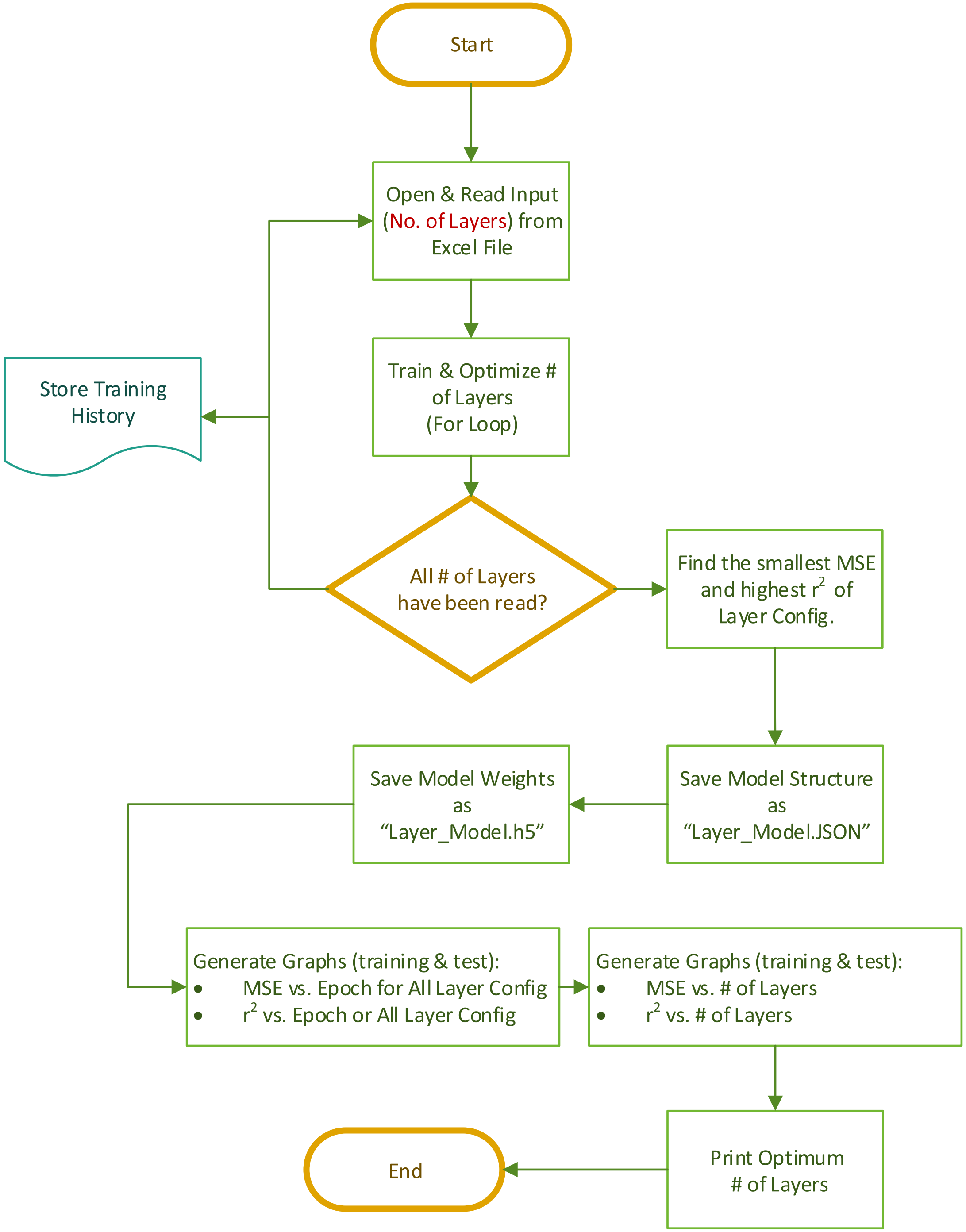
Objective

1. Improve accuracy of the current model in validation and prediction dataset
2. Provide the evidence of model optimization through the workflow

Flowcharts

Script #1

Layers Optimization



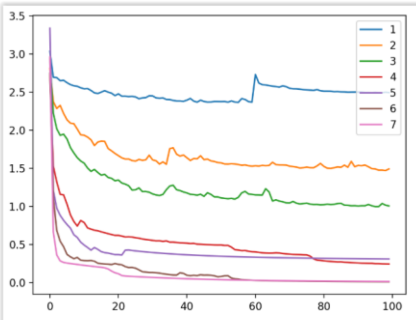
All Training & Test History Must be Stored in Excel

Graph Example:

Test result must be generated as well

Training

MSE

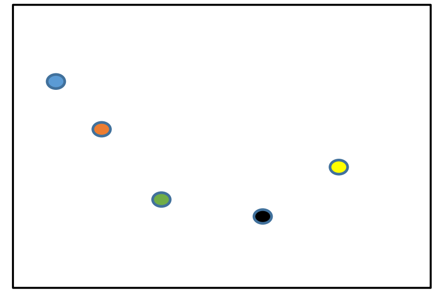


Epoch

# of Layers

Training

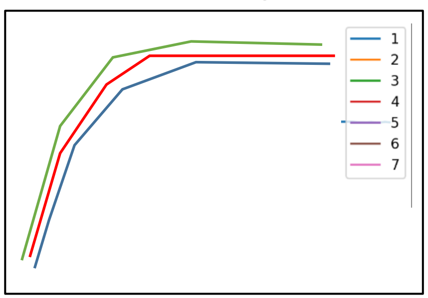
MSE



# of Layers

Training

r^2

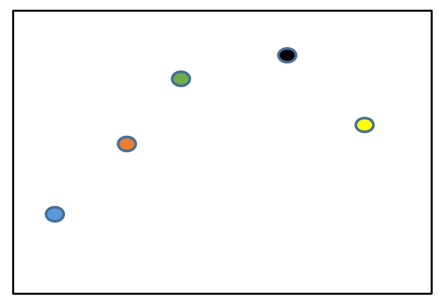


Epoch

# of Layers

Training

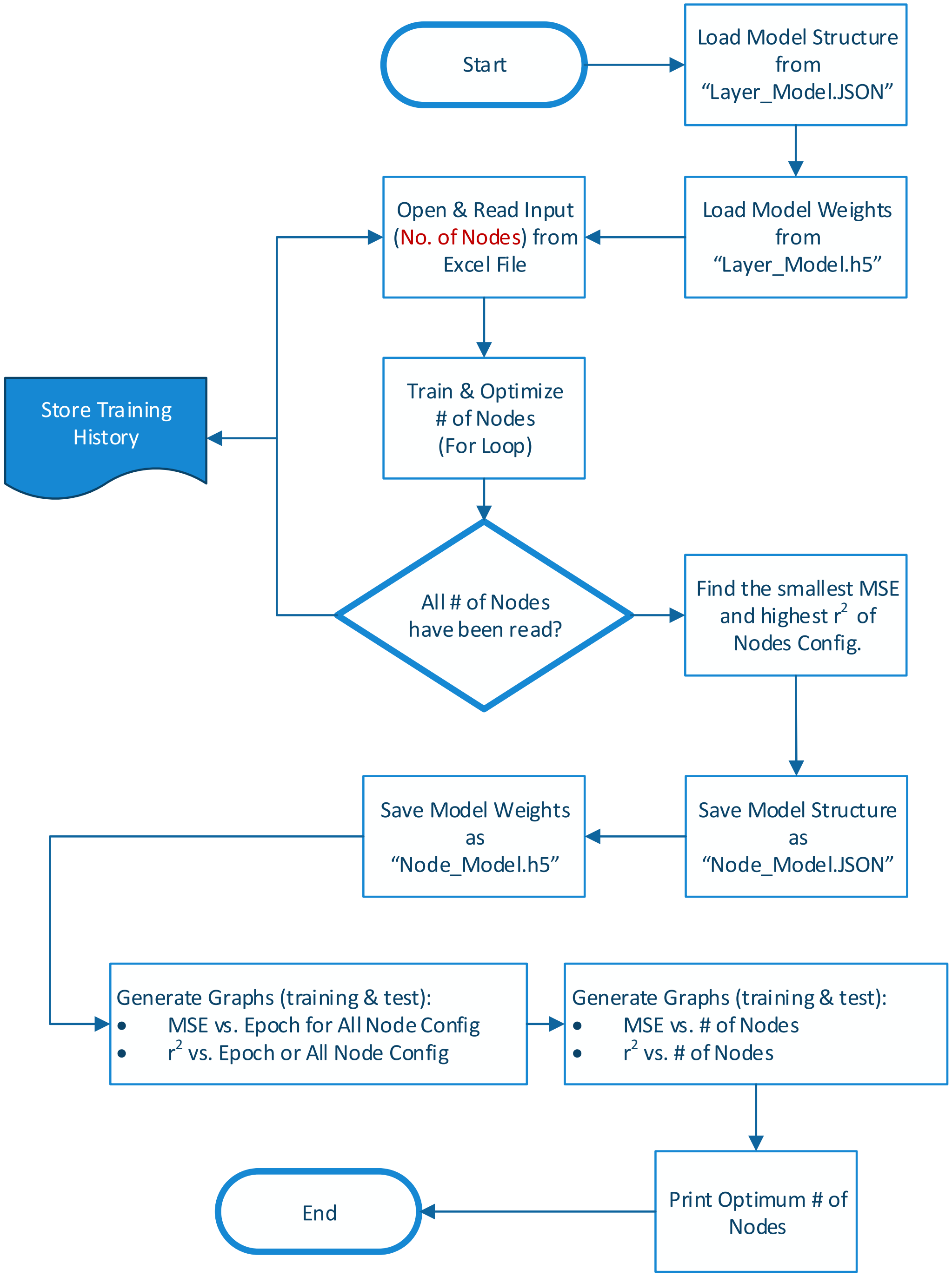
r^2



# of Layers

Script #2

Nodes Optimization



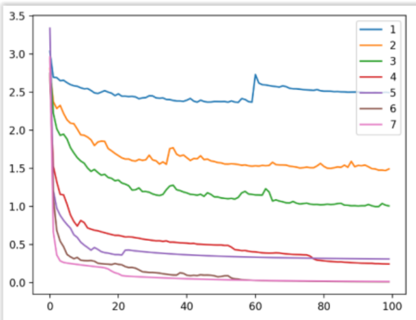
All Training & Test History Must be Stored in Excel

Graph Example:

Test result must be generated as well

Training

MSE

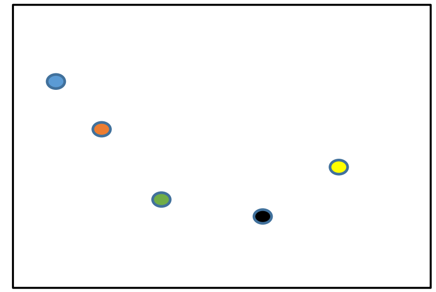


Epoch

# of Nodes

Training

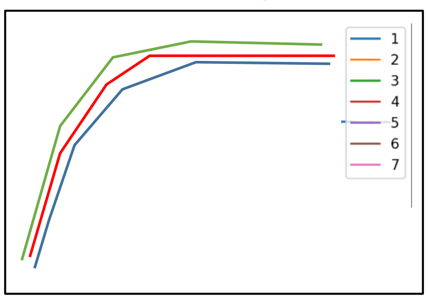
MSE



# of Nodes

Training

r^2

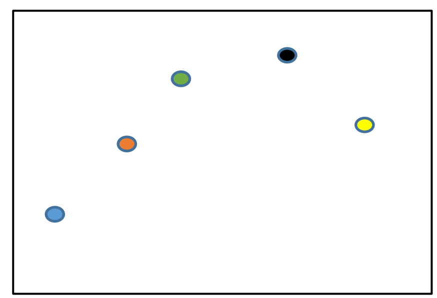


Epoch

# of Nodes

Training

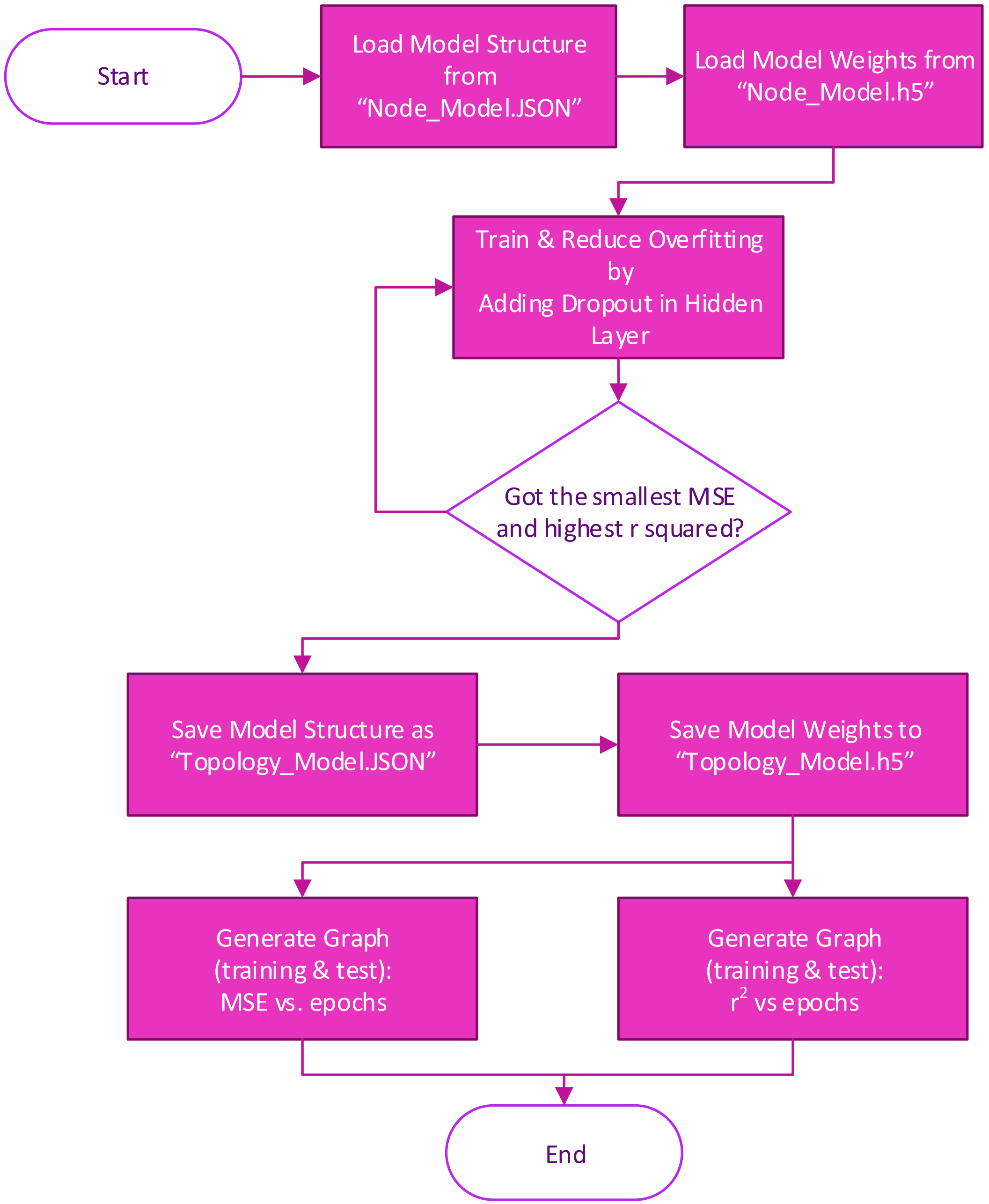
r^2



# of Nodes

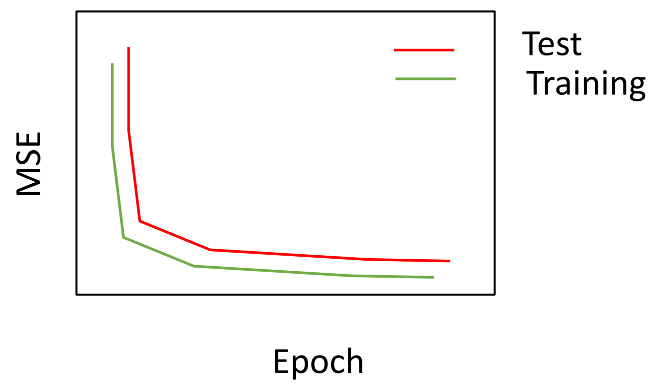
Script #3a

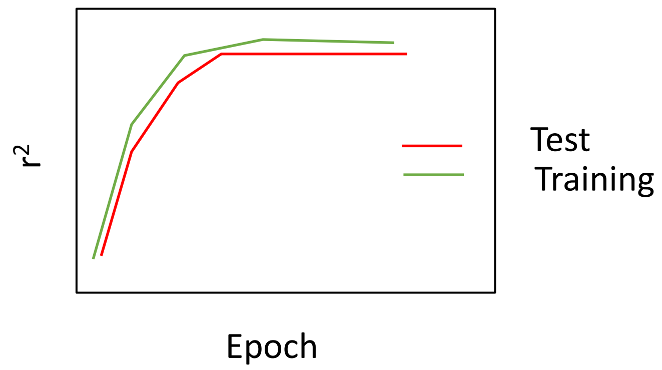
Overfitting Reduction-Training



All Training & Test History Must be Stored in Excel

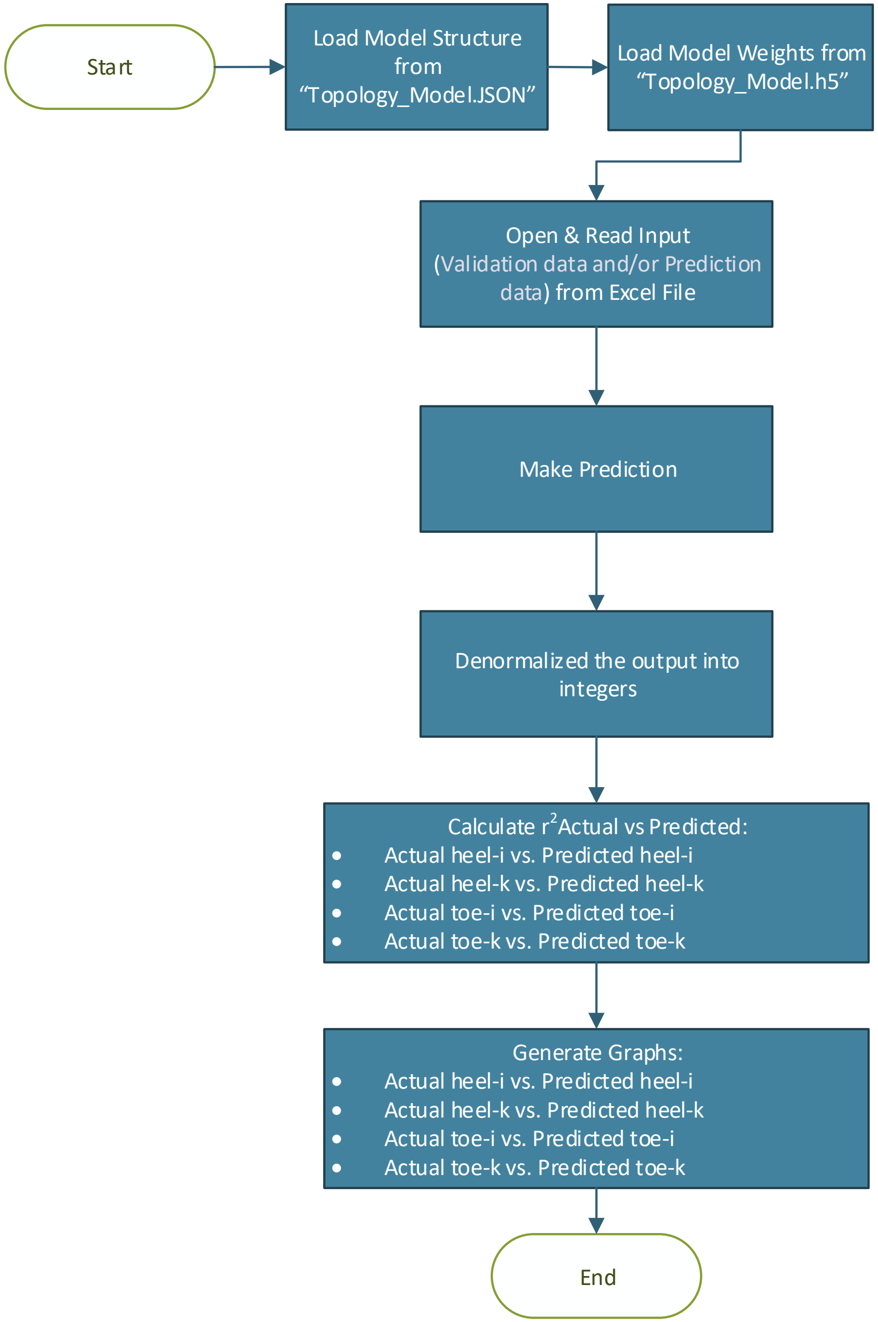
Graph Example:





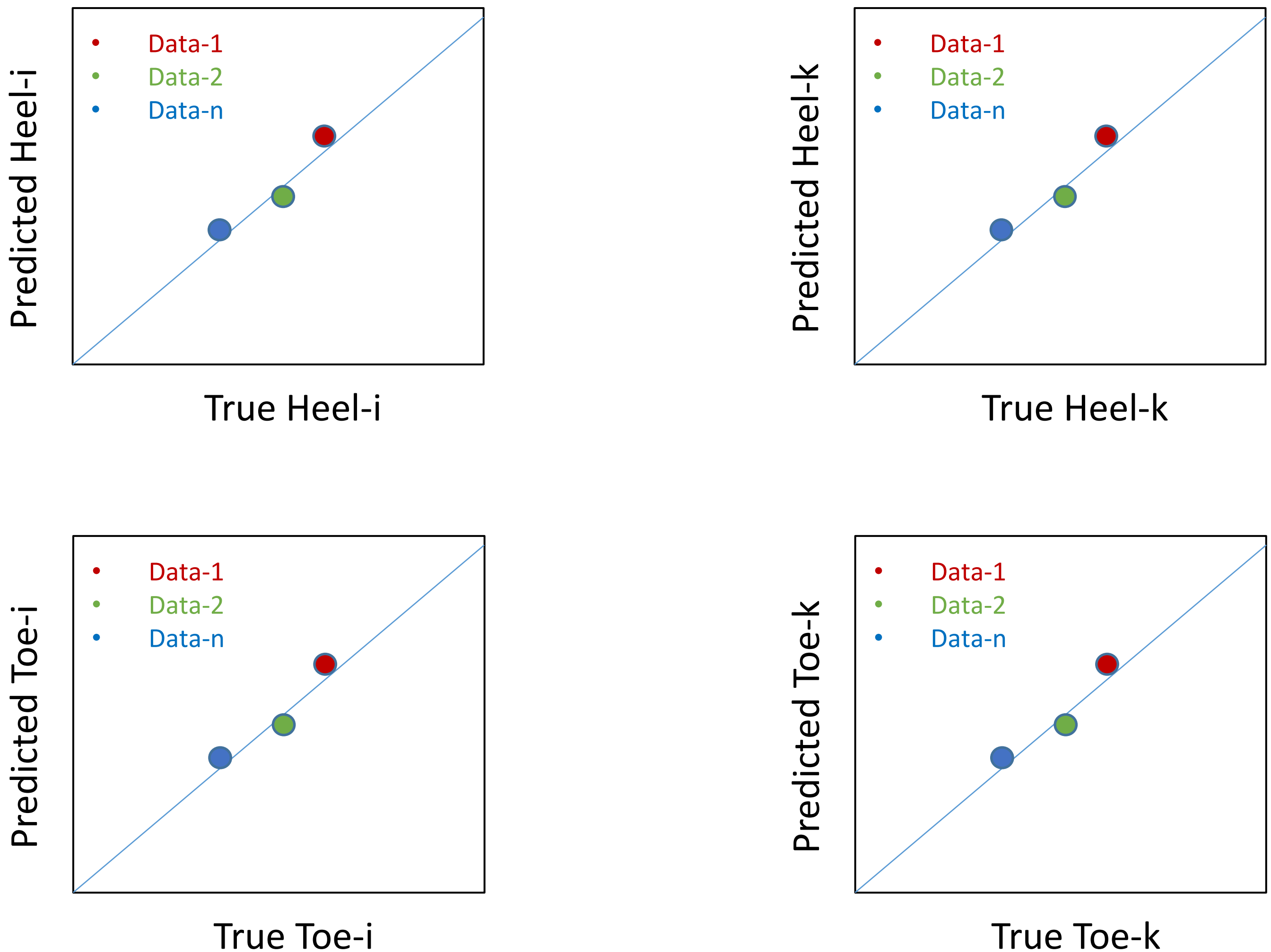
Script #3b

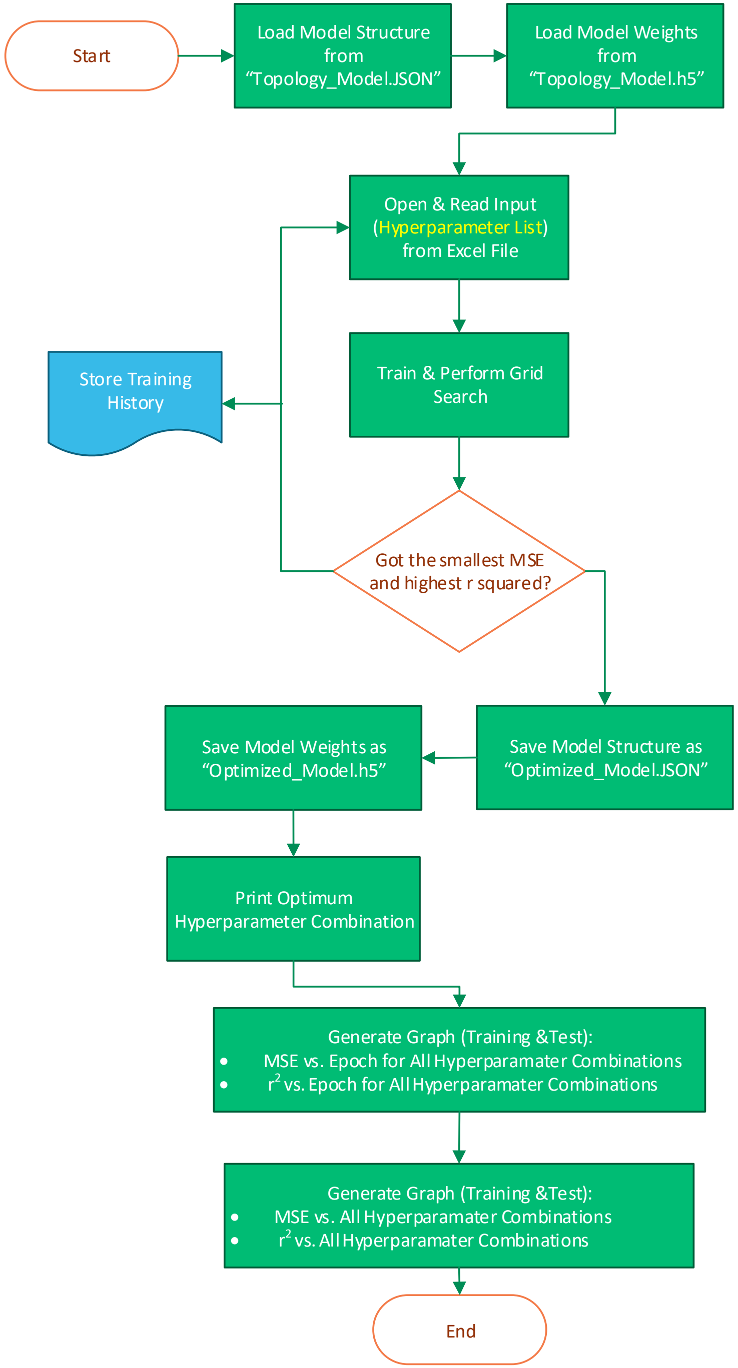
Overfitting Reduction-Prediction



Graph Example:

Validation Dataset and Prediction Dataset





Script #4a

Grid Search-Training

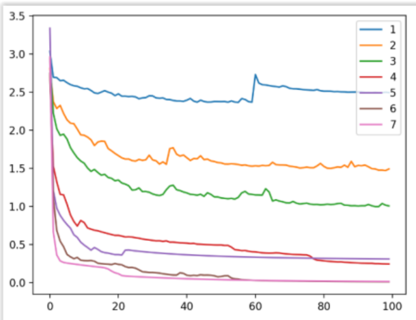
All Training & Test History Must be Stored in Excel

Graph Example:

Test result must be generated as well

Training

MSE

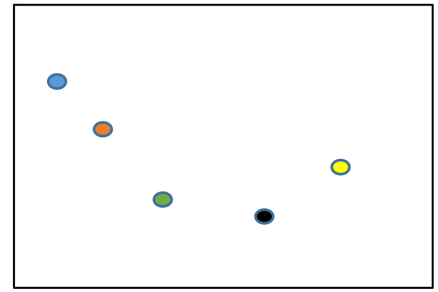


Epoch

HyperPara Comb-1
HyperPara Comb-2

Training

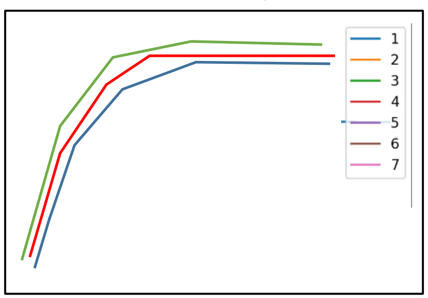
MSE



Hyperparameter Combination

Training

r^2

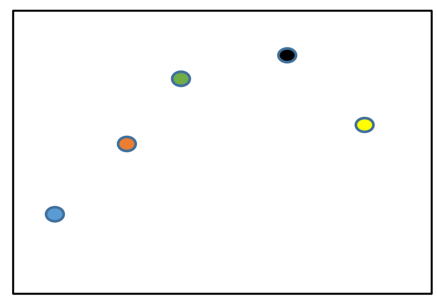


Epoch

HyperPara Comb-1
HyperPara Comb-2

Training

r^2

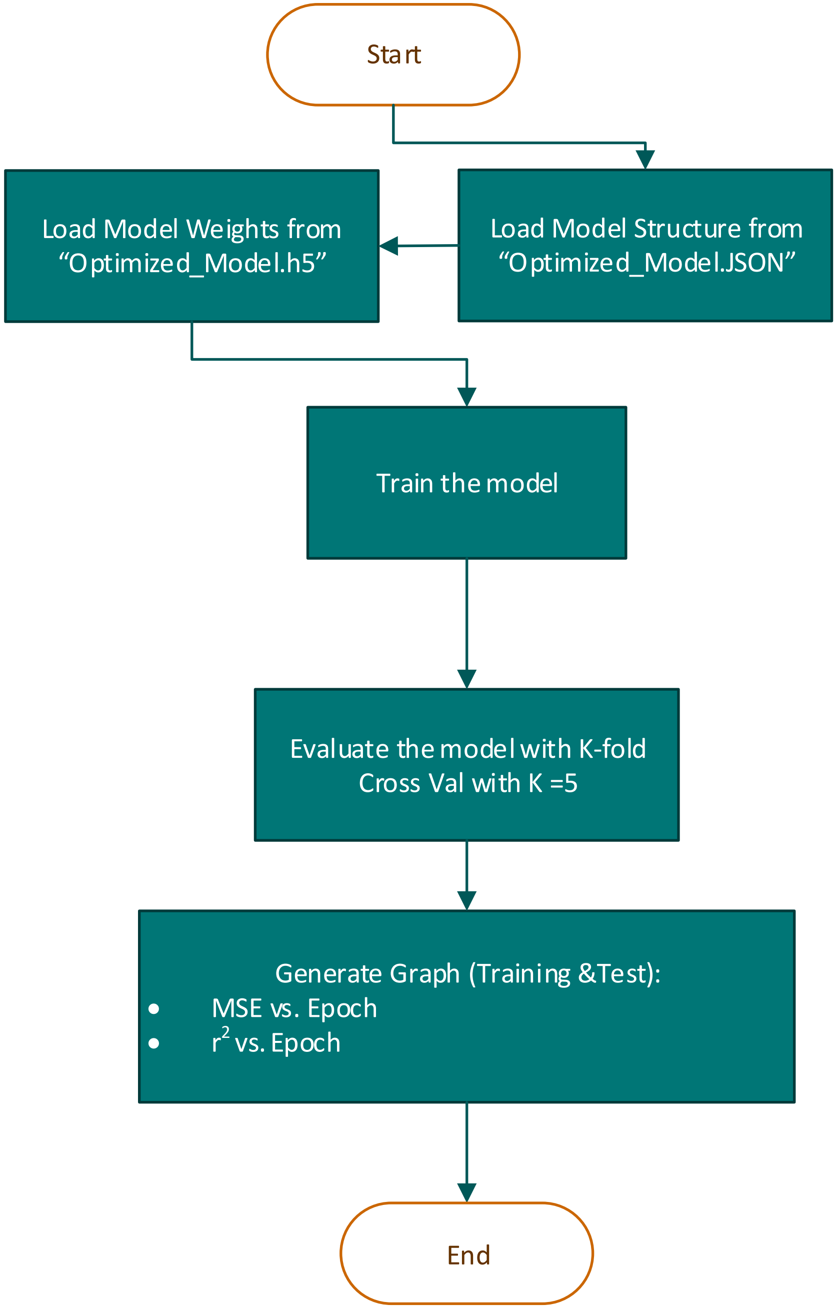


Hyperparameter Combination

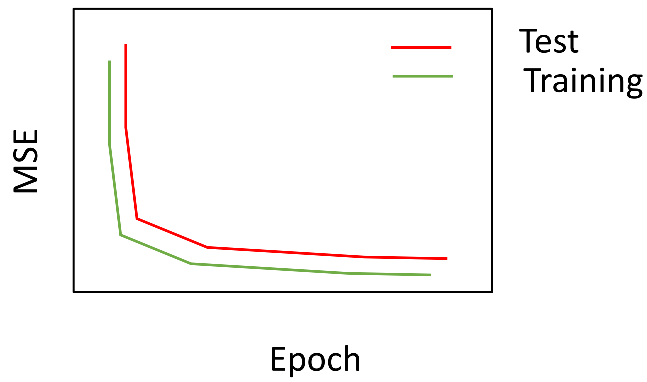
e.g. HyperPara Comb-1: 0.001, glorot uniform, Adam, 50, 100

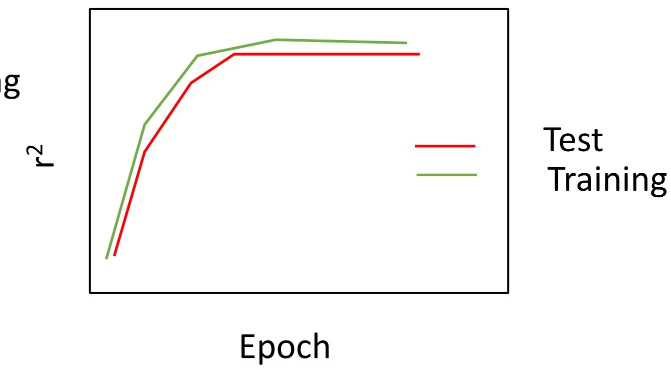
Script #4b

Grid Search-Training
with Cross Validation

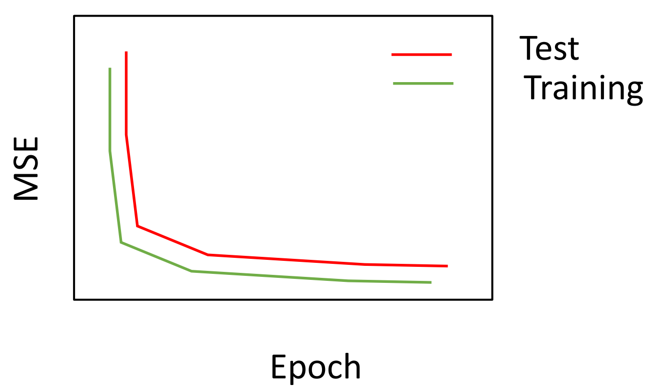


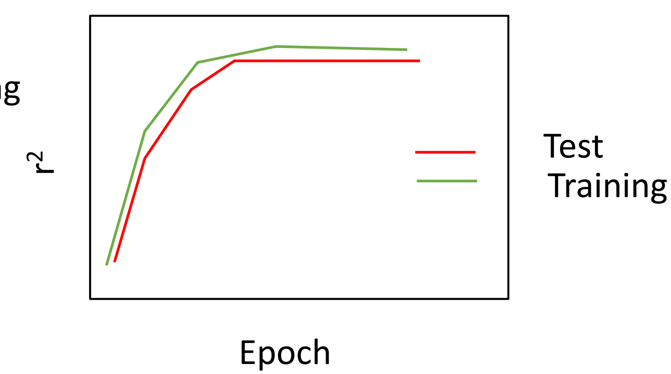
Graph Example:





K-1

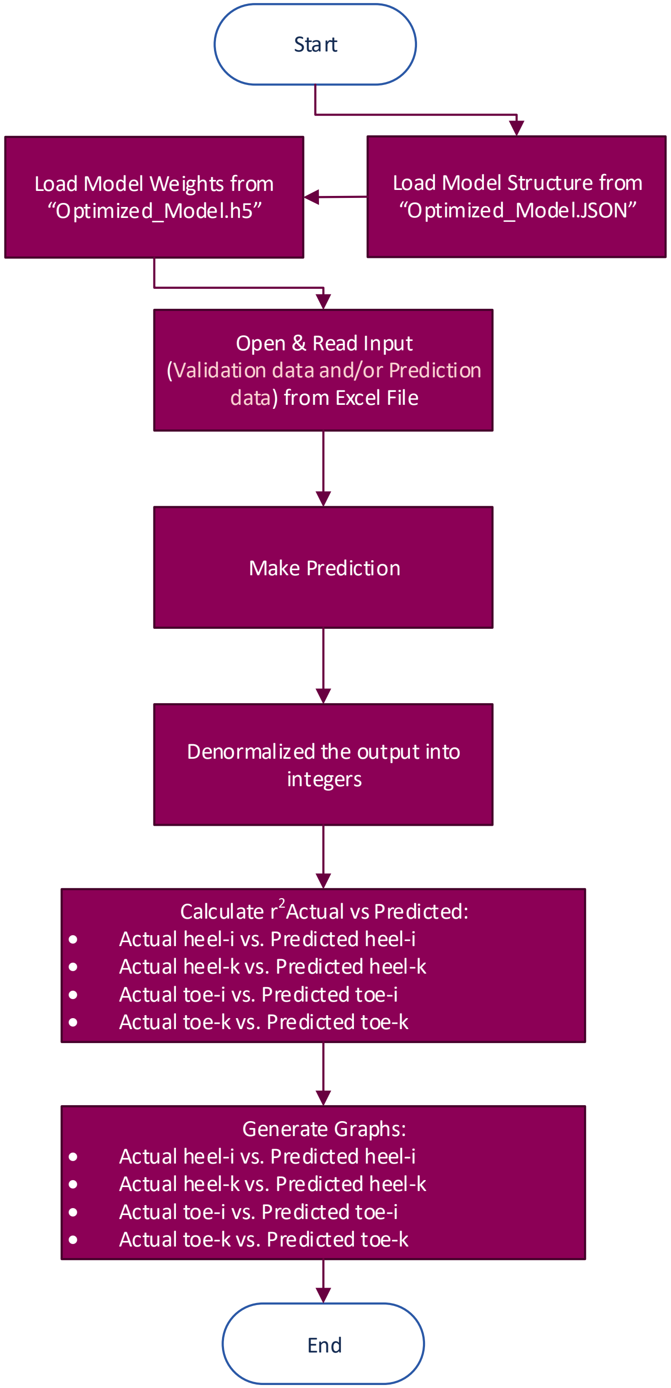




K-5

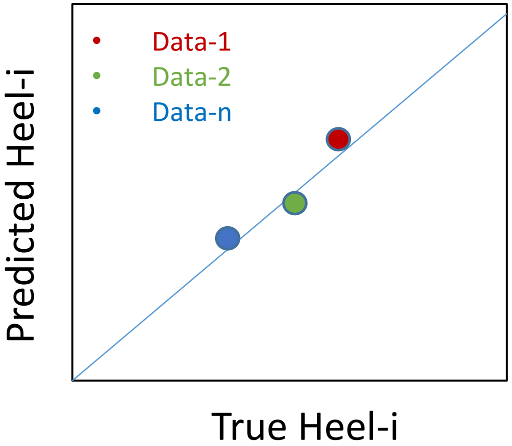
Script #4c

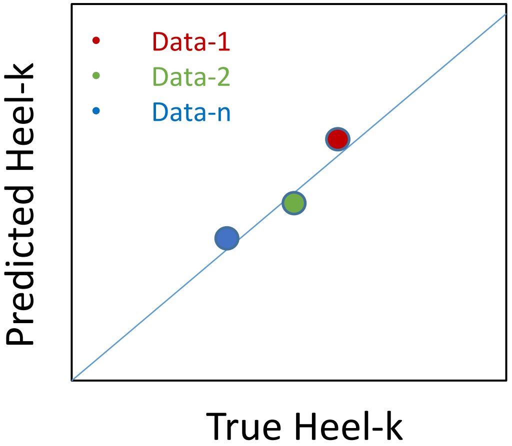
Grid Search-Prediction

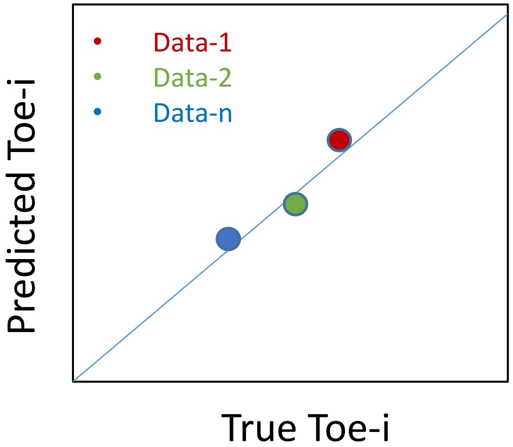


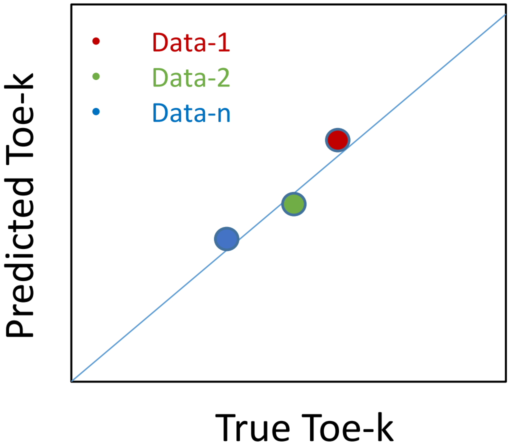
Graph Example:

Validation Dataset and Prediction Dataset



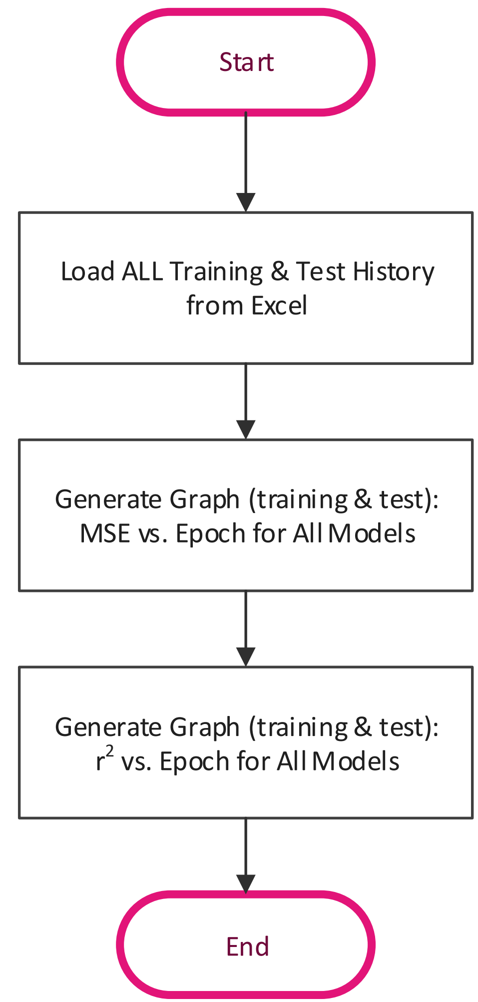






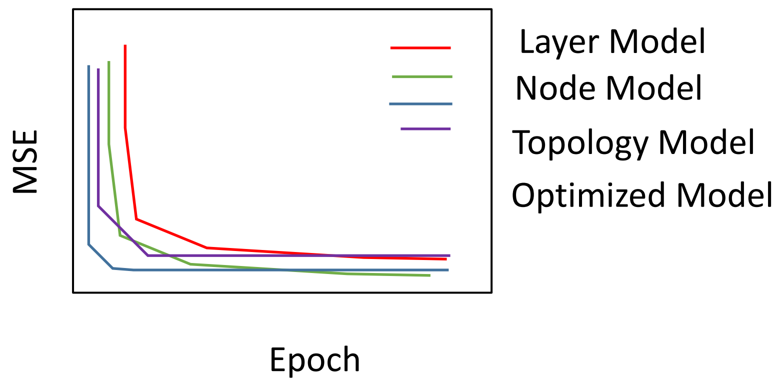
Script #5

Compilation for All Models

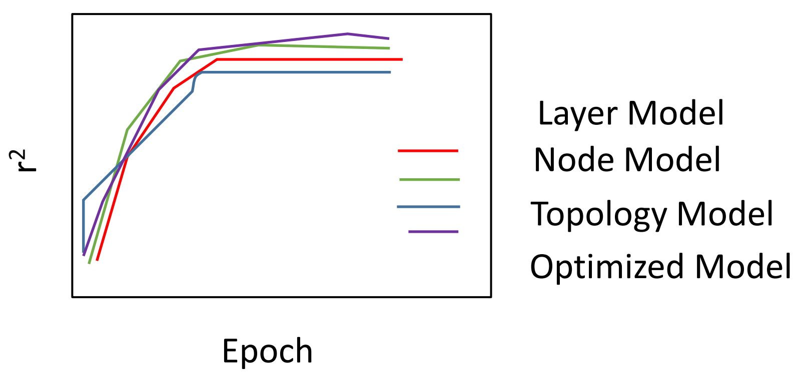


Graph Example:

Training & Test



Training & Test



Dataset

Dataset

- Stored in Excel ("Dataset.xlsx") and Consists of:
    1. Training and test dataset – Tab "Data"
    2. Validation dataset – Tab "Validation"
    3. Prediction dataset – Tab "Prediction"
    4. Not normalized target & Model # – Tab "Raw Data"
    5. Example calculation of denormalization for validation and prediction dataset – Tab "Denormalization Example"

Dataset-1

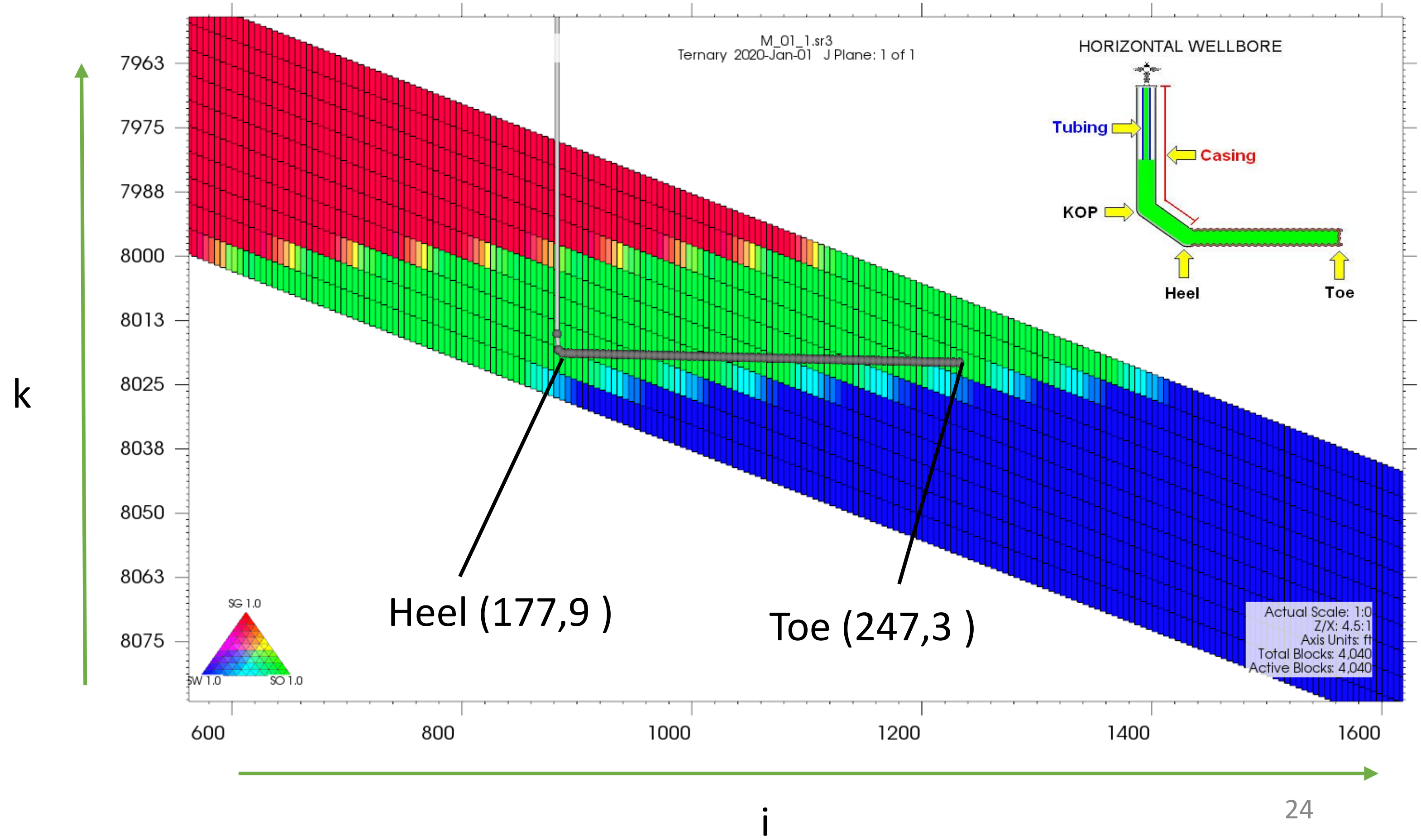
- Total number of data: 5,832

No.	Features	Unit	Type
1	Reservoir Thickness (h)	ft	Continuous
2	Oil Column Thickness (h_oil)	ft	Continuous
3	Dip	degree	Continuous
4	Gas Viscosity	cP	Continuous
5	Oil Viscosity	cP	Continuous
6	Ratio Gas Volume to Oil Volume (Vg/Vo)	rbb/rbb	Continuous
7	Ratio Water Volume to Oil Volume (Vw/Vo)	rbb/rbb	Continuous
8	Ratio Vertical to Horizontal Permeability (KvKh)	-	Continuous
9	Oil Density	lb/ft^3	Continuous

No.	Targets	Unit	Type
Well Location:			
1	Heel-i	-	**Integer**
2	Heel-k	-	**Integer**
3	Toe-i	-	**Integer**
4	Toe-k	-	**Integer**

Dataset-2

- Example illustration of well location (i,k)- taken from data No.1:



Dataset-3

- Features Matrix:

Reservoir Thickness (h)	Oil Column Thickness (h_oil)	Dip	Gas Viscosity	Oil Viscosity	Ratio Gas Volume to Oil Volume (Vg/Vo)	Ratio Water Volume to Oil Volume (Vw/Vo)	Ratio Vertical to Horizontal Permeability	Oil Density
50	25	5	0.01	0.5	3	3	0.01	0.7
50	25	5	0.01	0.5	3	3	0.01	0.85
50	25	5	0.01	0.5	3	3	0.1	0.7
50	25	5	0.01	0.5	3	3	0.1	0.85
50	25	5	0.01	0.5	3	3	0.5	0.7
50	25	5	0.01	0.5	3	3	0.5	0.85
50	25	5	0.01	0.5	3	5	0.01	0.7
50	25	5	0.01	0.5	3	5	0.01	0.85
50	25	5	0.01	0.5	3	5	0.1	0.7
50	25	5	0.01	0.5	3	5	0.1	0.85
50	25	5	0.01	0.5	3	5	0.5	0.7
50	25	5	0.01	0.5	3	5	0.5	0.85
50	25	5	0.01	0.5	5	3	0.01	0.7
50	25	5	0.01	0.5	5	3	0.01	0.85
50	25	5	0.01	0.5	5	3	0.1	0.7
50	25	5	0.01	0.5	5	3	0.1	0.85

Dataset-4

Warning: Targets have been normalized using specific domain knowledge – you don't need to normalize using machine learning method

- Target Matrix:

Well Location			
Heel		Toe	
i	k	i	k
177	9	247	3
179	9	249	3
171	10	241	4
174	10	244	4
175	10	245	4
175	10	245	4
179	9	249	3
179	9	249	3
171	10	241	4
171	10	241	4
175	10	245	4
175	10	245	4
171	10	241	4
180	9	250	3
171	10	241	4

Normalized Well Location			
Heel		Toe	
i	k	i	k
0.15	0.808	0.75	0.832
0.15	0.843	0.75	0.867
0.05	0.903	0.65	0.927
0.05	0.955	0.65	0.980
0.05	0.973	0.65	0.997
0.05	0.973	0.65	0.997
0.15	0.843	0.75	0.867
0.15	0.843	0.75	0.867
0.05	0.903	0.65	0.927
0.05	0.903	0.65	0.927
0.05	0.973	0.65	0.997
0.05	0.973	0.65	0.997
0.05	0.903	0.65	0.927
0.15	0.860	0.75	0.885
0.05	0.903	0.65	0.927

Normalization and Denormalization

Normalization Formula: Target

- As you have been warned in the previous slide, the target/output was normalized/transformed based on the domain specific knowledge
- This normalization will transform the integers well location into floating data type
- You have to feed the normalized target into the neural net.
- After running the prediction, you have to **denormalized** the output to convert back into integers

Normalization Formula-1

Based on Domain Knowledge

Heel:

$$Heel_{(i)} = 1 - 0.1 \times Heel_{(k)original} + 0.05$$ → *Only for Model (M): M_01 to M_09*

$$Heel_{(k)} = \frac{[DTOP_M + Zstep_M \times \left(Heel_{(i)original} - 1\right) + 5 \times Heel_{(k)original} - 8002.5]}{h\_oil}$$

Toe:

$$Toe_{(i)} = 1 - 0.1 \times Toe_{(k)original} + 0.05$$ → *Only for Model (M): M_01 to M_09*

$$Toe_{(k)} = \frac{[DTOP_M + Zstep_M \times \left(Toe_{(i)original} - 1\right) + 5 \times Toe_{(k)original} - 8002.5]}{h\_oil}$$

Normalization Formula-2

Based on Domain Knowledge

Formula for $\mathrm{Heel}_{(i)}$, $\mathrm{Toe}_{(i)}$ are based on Model Number:

$$Heel_{(i)} = 1 - 0.1 \times Heel_{(k)original} + 0.05$$

$$Toe_{(i)} = 1 - 0.1 \times Toe_{(k)original} + 0.05$$

Only for Model (M): M_01 to M_09

$$Heel_{(i)} = 1 - 0.05 \times Heel_{(k)original} + 0.025$$

$$Toe_{(i)} = 1 - 0.05 \times Toe_{(k)original} + 0.025$$

Only for Model (M): M_10 to M_18

$$Heel_{(i)} = 1 - 0.025 \times Heel_{(k)original} + 0.0125$$

$$Toe_{(i)} = 1 - 0.025 \times Toe_{(k)original} + 0.0125$$

Only for Model (M): M_19 to M_27

Normalization Formula-3

Based on Domain Knowledge

Formula for $Heel_{(k)}$, $Toe_{(k)}$ are based on DTOP and Zstep of *each* Model:

$$Heel_{(k)} = \frac{[DTOP_M + Zstep_M \times \left(Heel_{(i)original} - 1\right) + 5 \times Heel_{(k)original} - 8002.5]}{h\_oil}$$

$$Toe_{(k)} = \frac{[DTOP_M + Zstep_M \times \left(Toe_{(i)original} - 1\right) + 5 \times Toe_{(k)original} - 8002.5]}{h\_oil}$$

# Model	DTOP	z_step
M_1	7900.700	0.437443
M_2	7900.860	1.339746
M_3	7901.850	2.886751
M_4	7825.460	0.437443
M_5	7827.173	1.339746
M_6	7826.796	2.886751
M_7	7748.033	0.437443
M_8	7749.468	1.339746
M_9	7751.739	2.886751
M_10	7875.766	0.437443
M_11	7875.403	1.339746
M_12	7875.869	2.886751
M_13	7800.526	0.437443
M_14	7801.718	1.339746
M_15	7803.701	2.886751
M_16	7722.661	0.437443
M_17	7724.012	1.339746
M_18	7725.759	2.886751
M_19	7775.592	0.437443
M_20	7777.602	1.339746
M_21	7780.607	2.886751
M_22	7750.657	0.437443
M_23	7752.147	1.339746
M_24	7751.739	2.886751
M_25	7673.230	0.437443
M_26	7674.442	1.339746
M_27	7676.684	2.886751

How to Transform back to Integer

Denormalization

Example for Heel$_{(k)original}$ when making prediction:

- Calculate first:

$$Heel_{(i)} = 1 - 0.1 \times \boxed{Heel_{(k)original}} + 0.05 \quad \Rightarrow \quad \text{Got } Heel_{(k)original}$$

Obtained from Neural Net

- Then calculate KOP$_{(k)}$ by substituting KOP$_{(i)}$:

$$Heel_{(k)} = \frac{[DTOP_M + Zstep_M \times \boxed{(Heel_{(i)original} - 1)} + 5 \times Heel_{(k)original} - 8002.5]}{h\_oil} \quad \Rightarrow \quad \text{Got } Heel_{(i)original}$$

Obtained from Neural Net

Substituted from above

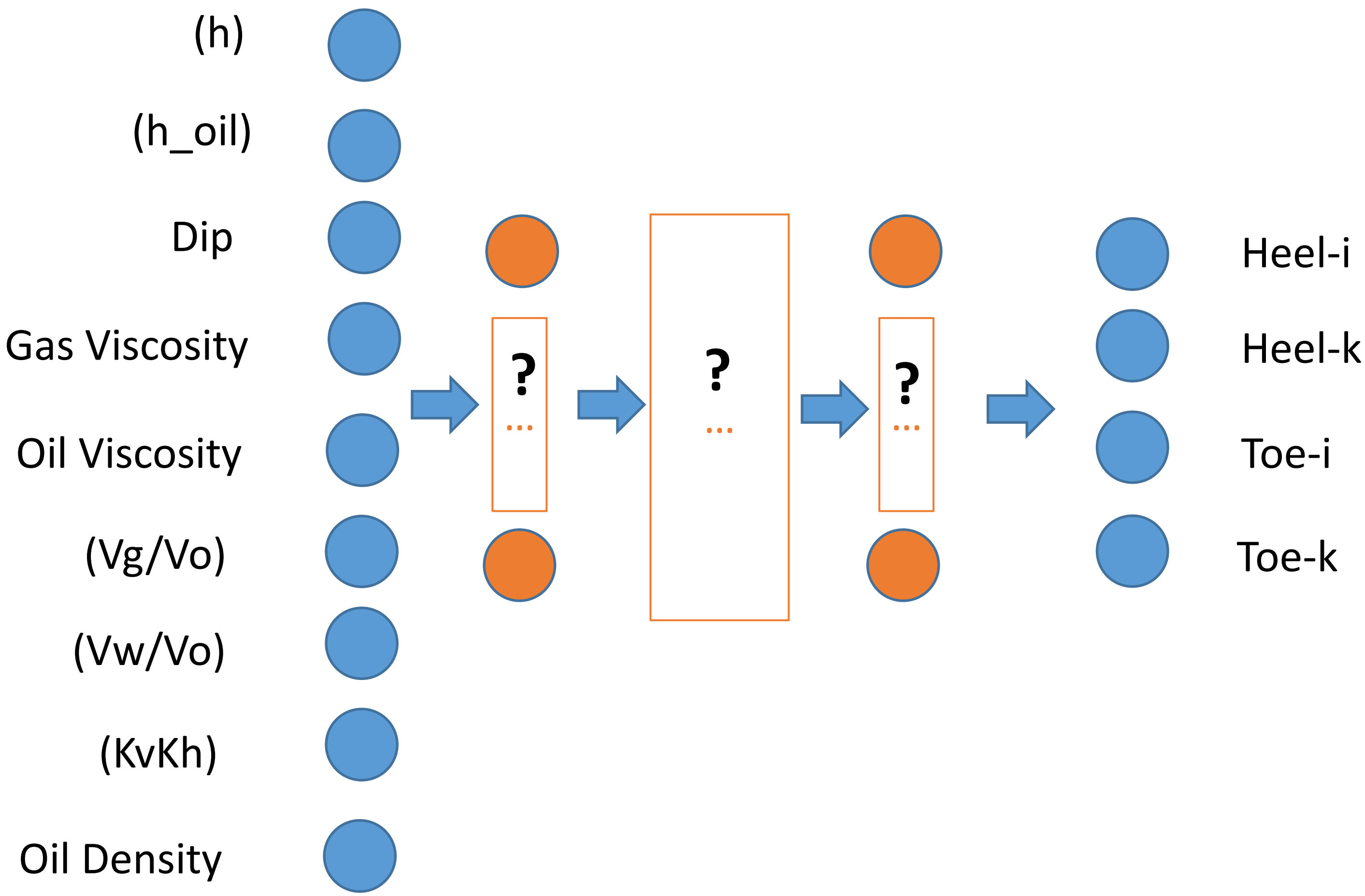
Obtained from input data

Provided in test data or prediction data

Deep Learning Model

Deep Learning Model-1

- Problem type: regression with multioutput **integer**
- Model type: Multilayer Perceptrons (MLP)



Deep Learning Model-2

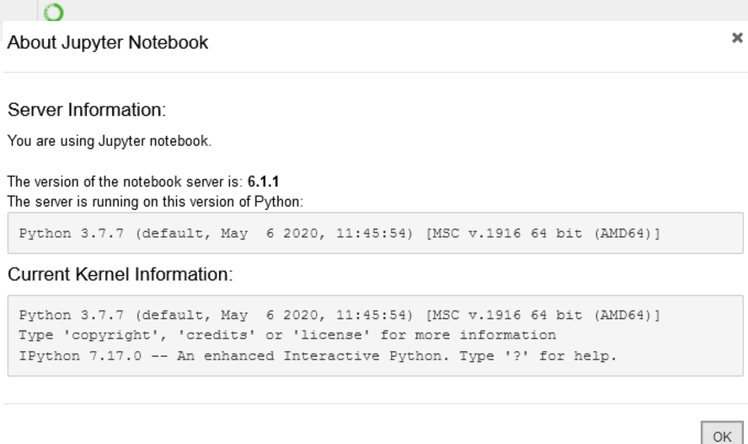
- Dataset:
  - Training = 80%
  - Test = 20%
  - Small batch
- Model evaluation:
  - Manual Verification Dataset
  - K-fold cross val only for script# 4b
- Loss function:
  - MSE
- Metrics:
  - RMSE
  - R squared

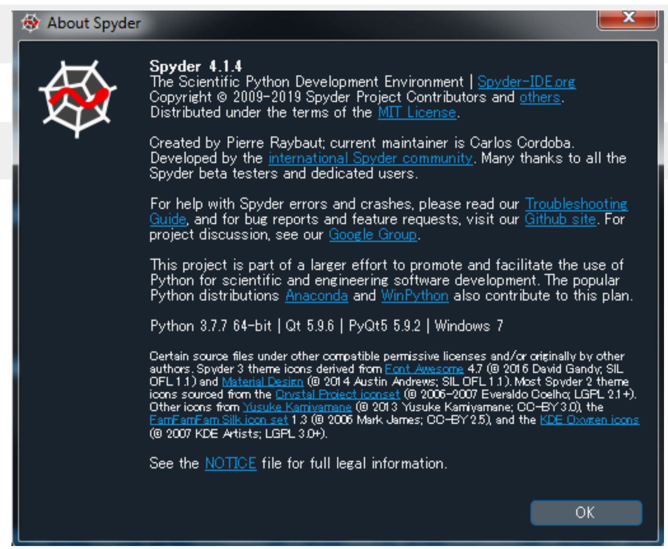
Library and IDE

- Suggested to use Keras with backend tensorflow.
- Spyder or jupyter Notebook

Name	T	Description	Version
keras		Deep learning library for theano and tensorflow	2.3.1
keras-applications		Applications module of the keras deep learning library.	1.0.8
keras-base			2.3.1
keras-preprocessing		Data preprocessing and data augmentation module of the keras deep learning library	1.1.0

Name	T	Description	Version
keras		Deep learning library for theano and tensorflow	2.3.1
tensorflow		Tensorflow is a machine learning library.	2.1.0
tensorflow-base		Tensorflow is a machine learning library, base package contains only tensorflow.	2.1.0
tensorflow-estimator			2.1.0





HINTS

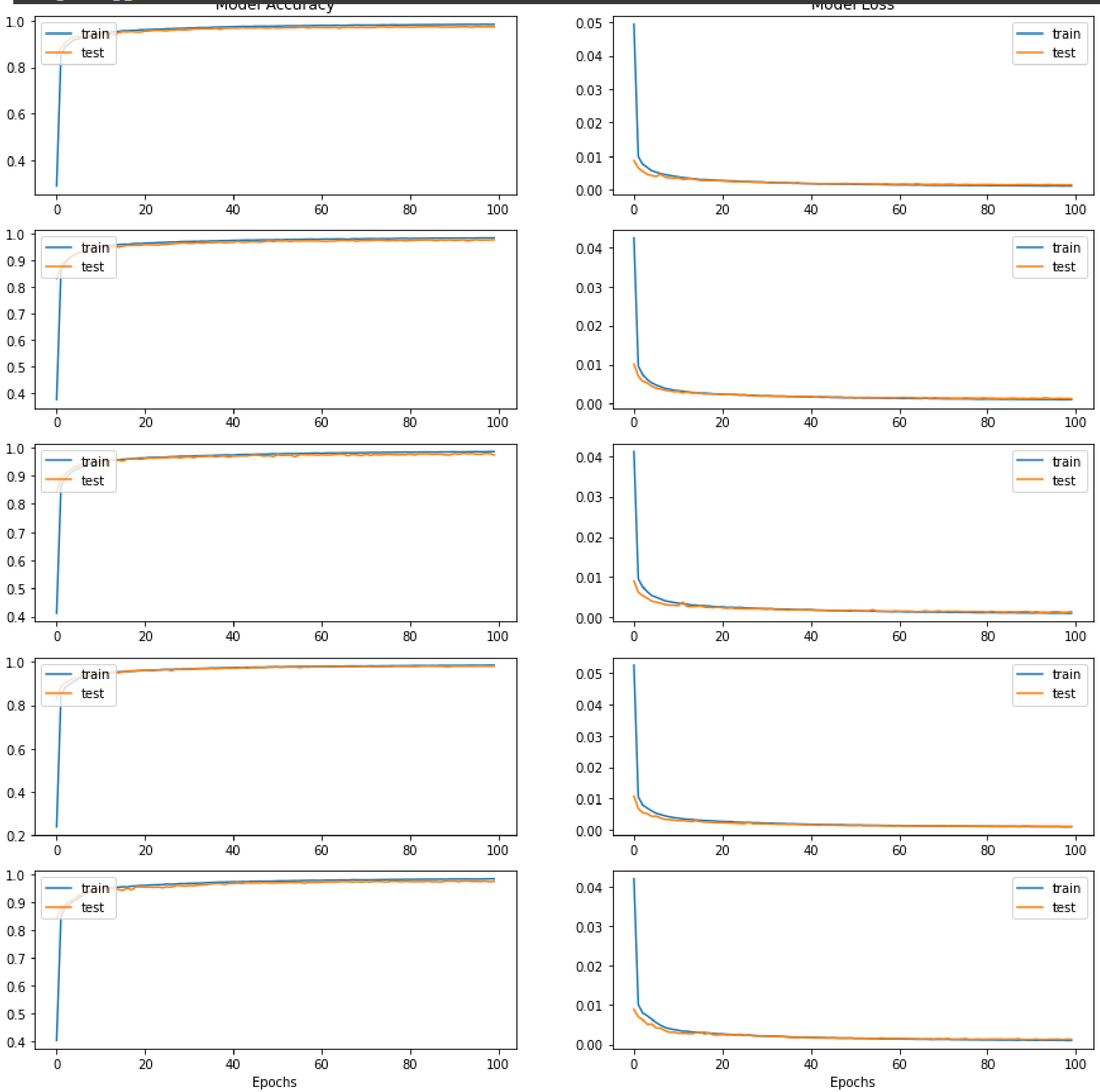
My Model

```
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Dense(128, kernel_initializer="glorot_uniform", activation='relu', input_dim=9))
model.add(tf.keras.layers.Dropout(0.1))
model.add(tf.keras.layers.Dense(128, kernel_initializer="glorot_uniform", activation='relu'))
#model.add(tf.keras.layers.Dropout(0.1))
#model.add(tf.keras.layers.Dense(10, kernel_initializer="glorot_uniform", activation='relu'))
#model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(4))
```

```
def topology_model(optimizer=tf.keras.optimizers.Adam, init='ones', lr=0.001):
```

```
epochs=100, batch_size=48, verbose=2)
```

```
25/25 - 0s - loss: 0.0013 - mse: 0.0013 - root_mean_squared_error: 0.0365 - r_squared: 0.9756
Topology Model: -0.0013 (0.0001) MSE
```



My Result

Actual

Model #	Model ID	Reservoir Thickness (h)	Oil Column Thicknes	Dip	Gas Viscosity	Oil Viscosity	Ratio Gas Volume to Oil	Ratio Water Volume	Ratio Vertical to	Oil Density	Well Location				Transformed Well Location			
											Heel		Toe		Heel		Toe	
											i	k	i	k	i	k	i	k
M_01	M_01_1	50	25	5	0.01	0.5	3	3	0.01	0.7	177	9	247	3	0.15	0.808	0.75	0.832
M_05	M_05_152	50	50	15	0.05	0.5	3	5	0.01	0.85	130	9	147	4	0.15	0.850	0.65	0.806
M_10	M_10_43	100	25	5	0.01	2	5	5	0.01	0.7	140	17	245	8	0.175	0.763	0.625	0.800
M_15	M_15_38	100	50	30	0.01	2	5	3	0.01	0.85	58	15	78	4	0.275	0.815	0.825	0.870
M_20	M_20_209	200	25	15	0.05	4	5	3	0.5	0.7	94	24	162	6	0.4125	0.788	0.8625	0.832
M_25	M_25_58	200	75	5	0.01	4	3	5	0.1	0.85	620	23	783	9	0.4375	0.753	0.7875	0.771

Prediction

+/- 1-2 prediction off

Validation Dataset

Model #	Model ID	Reservoir Thickness (h)	Oil Column Thicknes	Dip	Gas Viscosity	Oil Viscosity	Ratio Gas Volume to Oil	Ratio Water Volume	Ratio Vertical to	Oil Density	Well Location				Transformed Well Location			
											Heel		Toe		Heel		Toe	
											i	k	i	k	Heel_i	Heel_k	Toe_i	Toe_k
M_01	M_01_1	50	25	5	0.01	0.5	3	3	0.01	0.7	179	9	243	3	0.165699	0.809234	0.720401	0.828579
M_05	M_05_152	50	50	15	0.05	0.5	3	5	0.01	0.85	130	9	148	4	0.166553	0.832059	0.674344	0.801567
M_10	M_10_43	100	25	5	0.01	2	5	5	0.01	0.7	141	17	244	8	0.176272	0.766929	0.619003	0.799395
M_15	M_15_38	100	50	30	0.01	2	5	3	0.01	0.85	57	16	77	4	0.248444	0.805516	0.808466	0.855115
M_20	M_20_209	200	25	15	0.05	4	5	3	0.5	0.7	93	24	162	6	0.402034	0.799794	0.861934	0.852817
M_25	M_25_58	200	75	5	0.01	4	3	5	0.1	0.85	620	23	784	9	0.431341	0.769184	0.782971	0.789626

```
Loaded model from disk
1/1 [==============================] - 0s 224ms/step - loss: 2.0913e-04 - mse: 2.0913e-04 - root_mean_squared_error: 0.0145 - r_squared: 0.9964
WARNING:tensorflow:Model was constructed with shape (None, 9) for input KerasTensor(type_spec=TensorSpec(shape=(None, 9), dtype=tf.float32, name='dense_47_input'), na
WARNING:tensorflow:5 out of the last 5 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7f54ee4f2a60> triggered tf.function retracing. Tr
Predicting Well Location For First Data Point: [[0.165699   0.80923355 0.7204012  0.82857925]
 [0.1665529  0.83205926 0.6743443  0.80156696]
 [0.17627174 0.7669289  0.6190033  0.7993945 ]
 [0.24844411 0.80551577 0.8084664  0.85511506]
 [0.40203425 0.7997942  0.8619341  0.852817  ]
 [0.43134066 0.7691839  0.7829712  0.7896255 ]]
Topology Model Accuracy:  {'loss': 0.00020913417392876, 'mse': 0.00020913417392876, 'root_mean_squared_error': 0.01446147169917821 9, 'r_squared': 0.9963684678077698}
```

Actual

Model #	Reservoir Thickness (h)	Oil Column Thicknes	Dip	Gas Viscosity	Oil Viscosity	Ratio Gas Volume to Oil	Ratio Water Volume	Ratio Vertical to	Oil Density	Well Location Heel i	Well Location Heel k	Well Location Toe i	Well Location Toe k	Transformed Well Location Heel i	Transformed Well Location Heel k	Transformed Well Location Toe i	Transformed Well Location Toe k
Pred_1	160	40	25	0.02	3.5	3.5	4	0.3	0.73	53	22	80	10	0.328125	0.764	0.703125	0.837788
Pred_2	70	60	10	0.04	3	4.5	3.5	0.06	0.82	231	13	270	6	0.107143	0.835934	0.607143	0.825663
Pred_3	120	30	20	0.03	1	4	4.5	0.15	0.78	43	20	74	9	0.1875	0.823558	0.645833	0.870738

Prediction

Prediction Dataset

Model #	Reservoir Thickness (h)	Oil Column Thicknes	Dip	Gas Viscosity	Oil Viscosity	Ratio Gas Volume to Oil	Ratio Water Volume	Ratio Vertical to	Oil Density	Well Location Heel i	Well Location Heel k	Well Location Toe i	Well Location Toe k	Transformed Well Location Heel i	Transformed Well Location Heel k	Transformed Well Location Toe i	Transformed Well Location Toe k
Pred_1	160	40	25	0.02	3.5	3.5	4	0.3	0.73	56	21	84	8	0.363174	0.781521	0.755929	0.843054
Pred_2	70	60	10	0.04	3	4.5	3.5	0.06	0.82	233	12	267	6	0.187092	0.773697	0.599143	0.793344
Pred_3	120	30	20	0.03	1	4	4.5	0.15	0.78	49	18	79	7	0.266696	0.85216	0.720286	0.903623

+/- 1-4 prediction off