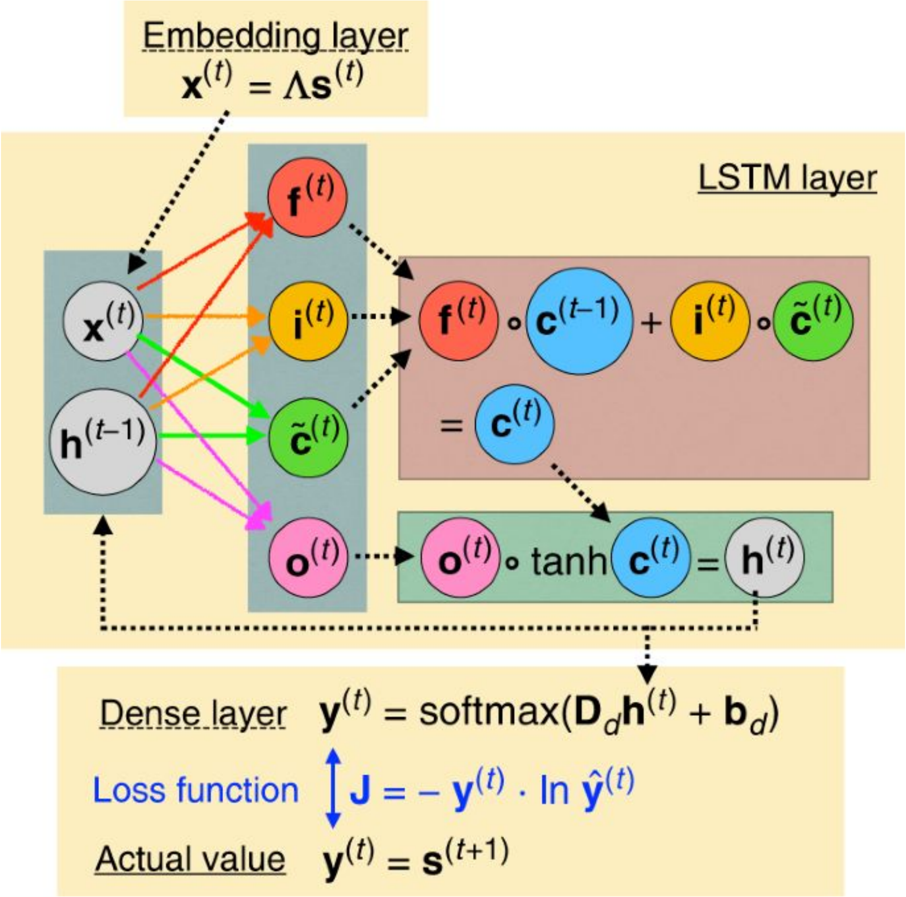
Nested Bidirectional LSTMs for Alanine Dipeptide and Chignolin Molecular Dynamic Forecasting

Franklin Ruan

Key Results from Paper

Learning molecular dynamics with simple language model built upon long short-term memory neural network

By: Sun-Ting Tsai, En-Jui Kuo & Pratyush Tiwary [published on Nature 09 Oct 2020]


Embedding layer
$\mathbf{x}^{(t)} = \Lambda \mathbf{s}^{(t)}$
LSTM layer
$\mathbf{x}^{(t)}$
$\mathbf{h}^{(t-1)}$
$\mathbf{f}^{(t)}$
$\mathbf{i}^{(t)}$
$\tilde{\mathbf{c}}^{(t)}$
$\mathbf{o}^{(t)}$
$\mathbf{f}^{(t)} \circ \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} \circ \tilde{\mathbf{c}}^{(t)}$
$= \mathbf{c}^{(t)}$
$\mathbf{o}^{(t)} \circ \tanh \mathbf{c}^{(t)} = \mathbf{h}^{(t)}$
Dense layer $\mathbf{y}^{(t)} = \mathrm{softmax}(\mathbf{D}_d \mathbf{h}^{(t)} + \mathbf{b}_d)$
Loss function $\mathbf{J} = -\mathbf{y}^{(t)} \cdot \ln \hat{\mathbf{y}}^{(t)}$
Actual value $\mathbf{y}^{(t)} = \mathbf{s}^{(t+1)}$

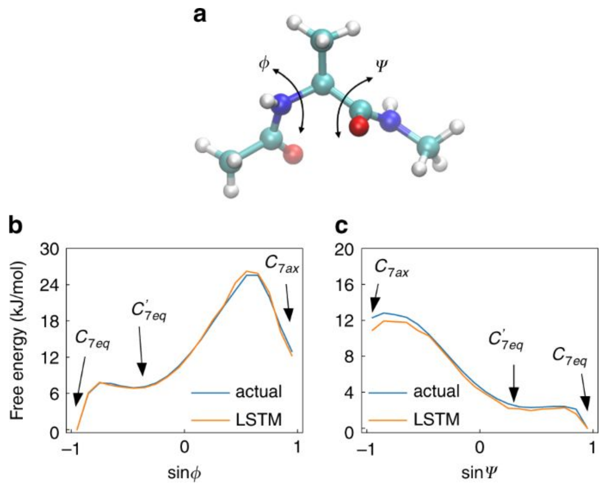
Learning molecular dynamics with simple language model built upon long short-term memory neural network

By: Sun-Ting Tsai, En-Jui Kuo & Pratyush Tiwary [published on Nature 09 Oct 2020]

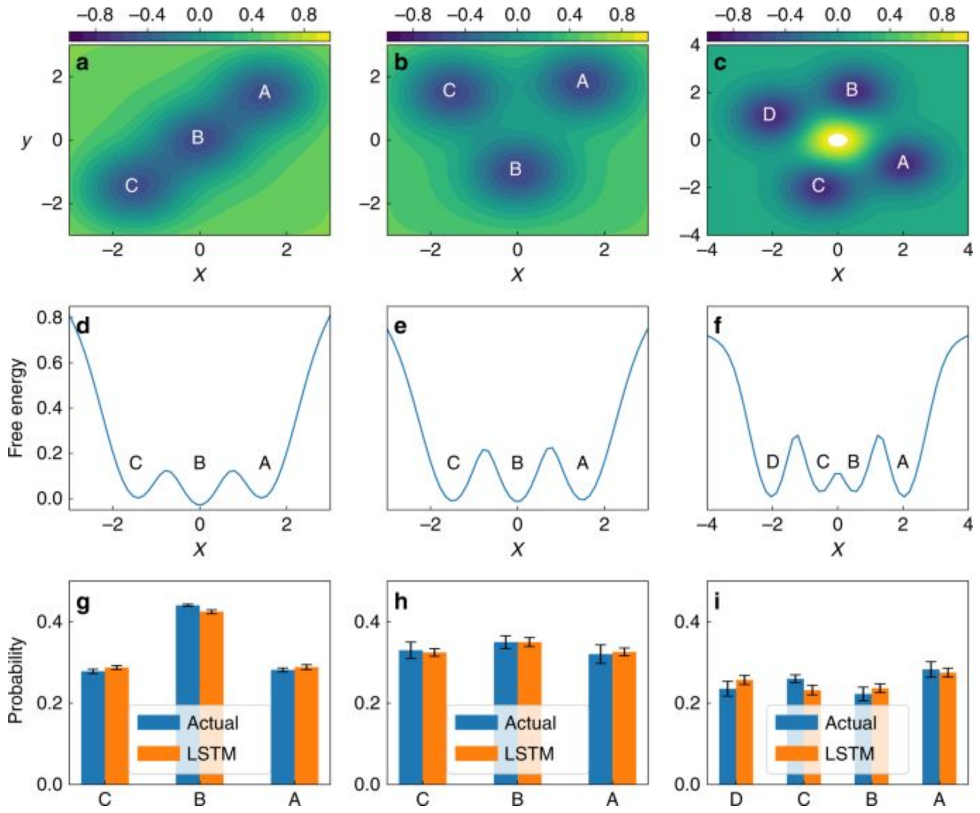
Cross Entropy:

$$J' = -\sum_{\mathbf{x}^{(T)}\ldots\mathbf{x}^{(0)}} P(\mathbf{x}^{(T)}\ldots\mathbf{x}^{(0)}) \ln Q(\mathbf{x}^{(T)}\ldots\mathbf{x}^{(0)})$$

Alanine Dipeptide:



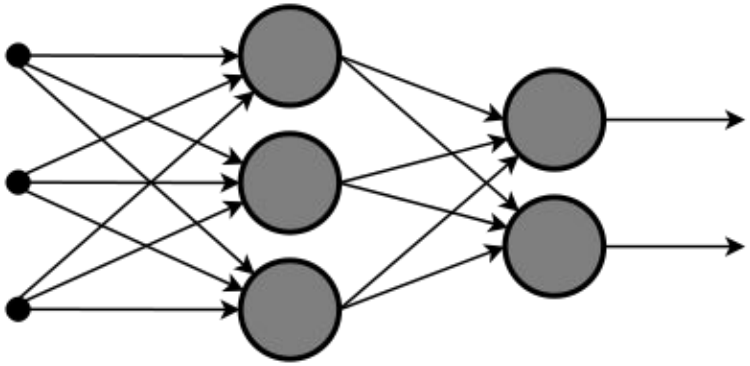
Boltzmann Statistics



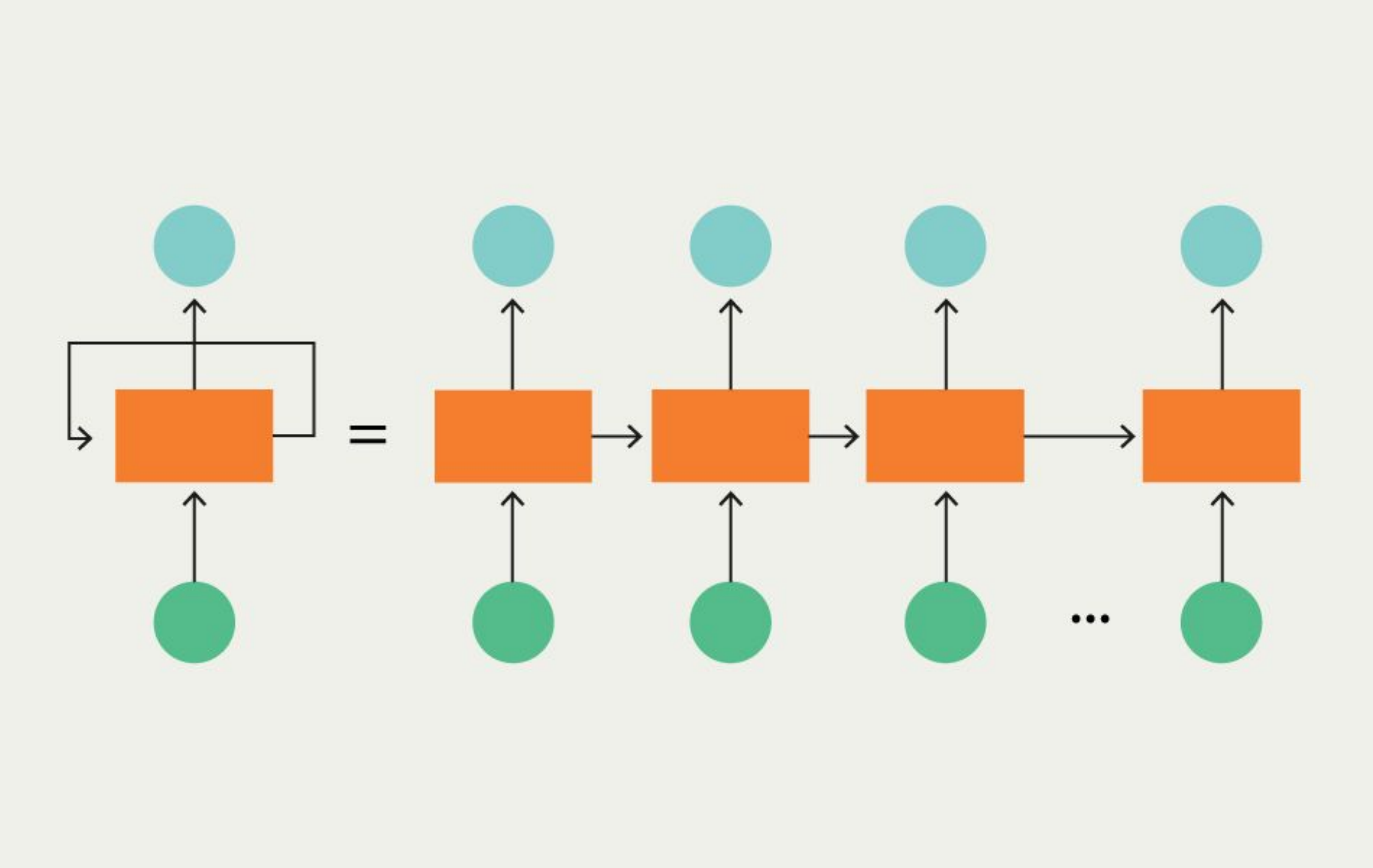
Theory

RNN

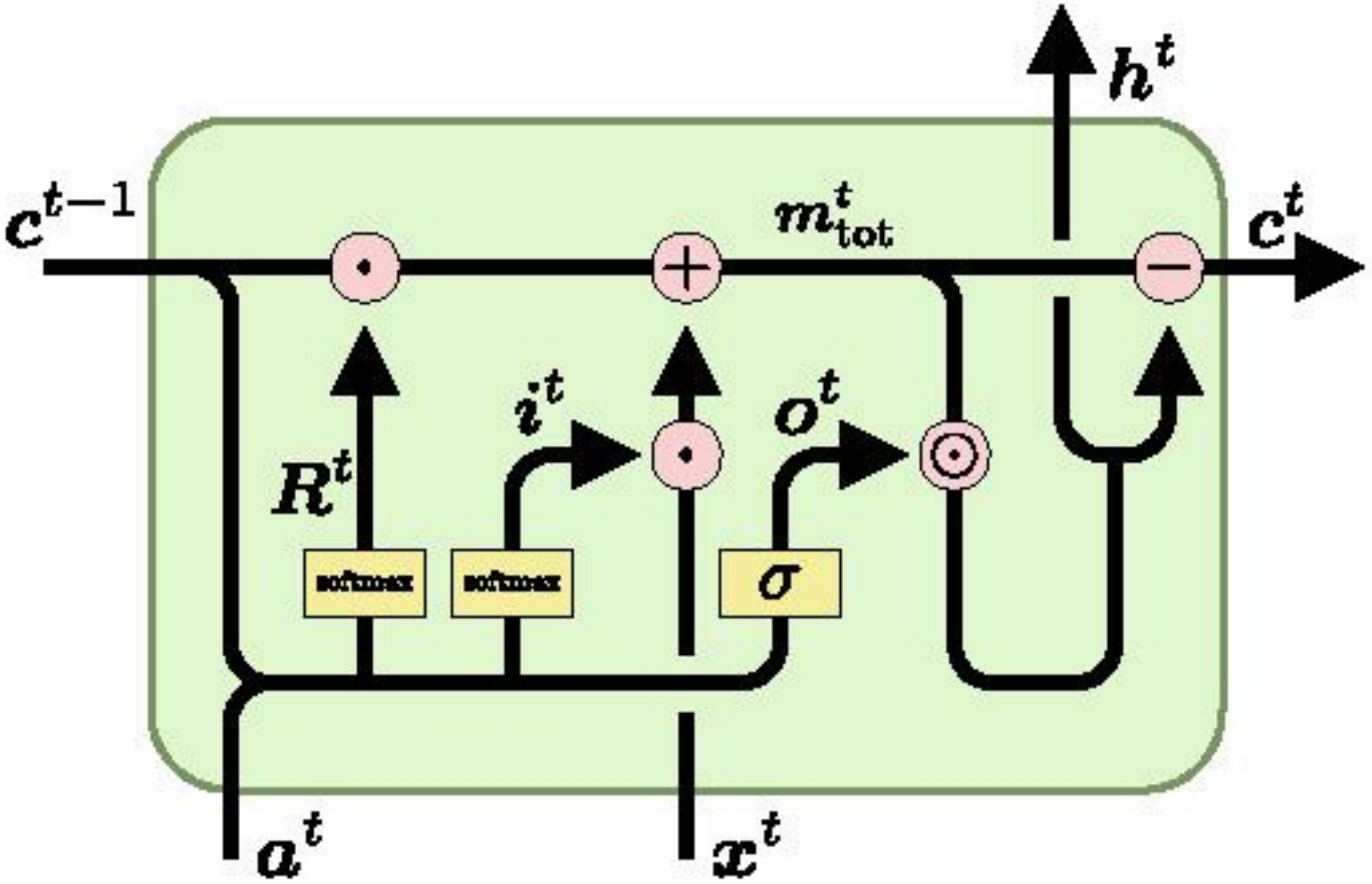
Simple Feed Forward Network:



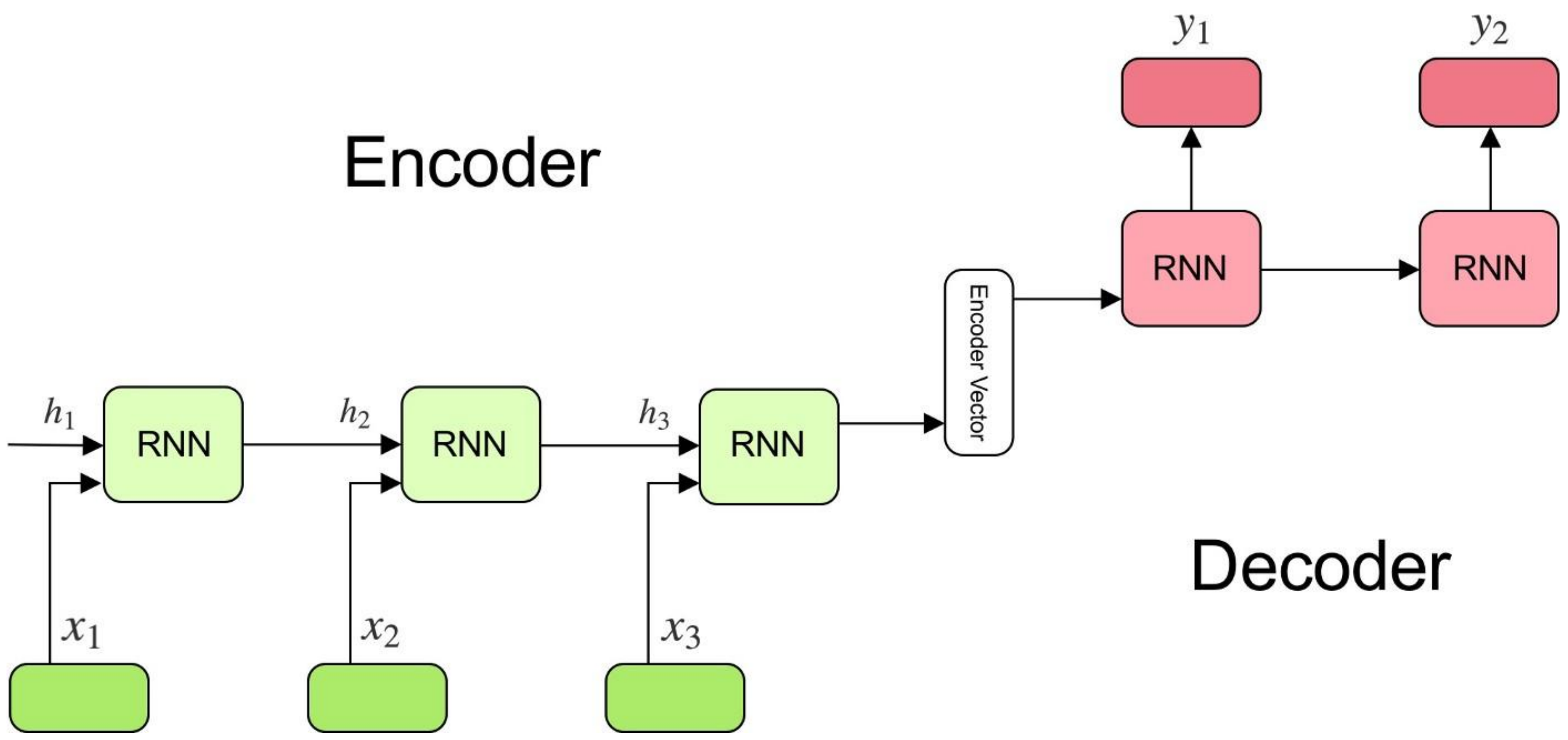
RNN



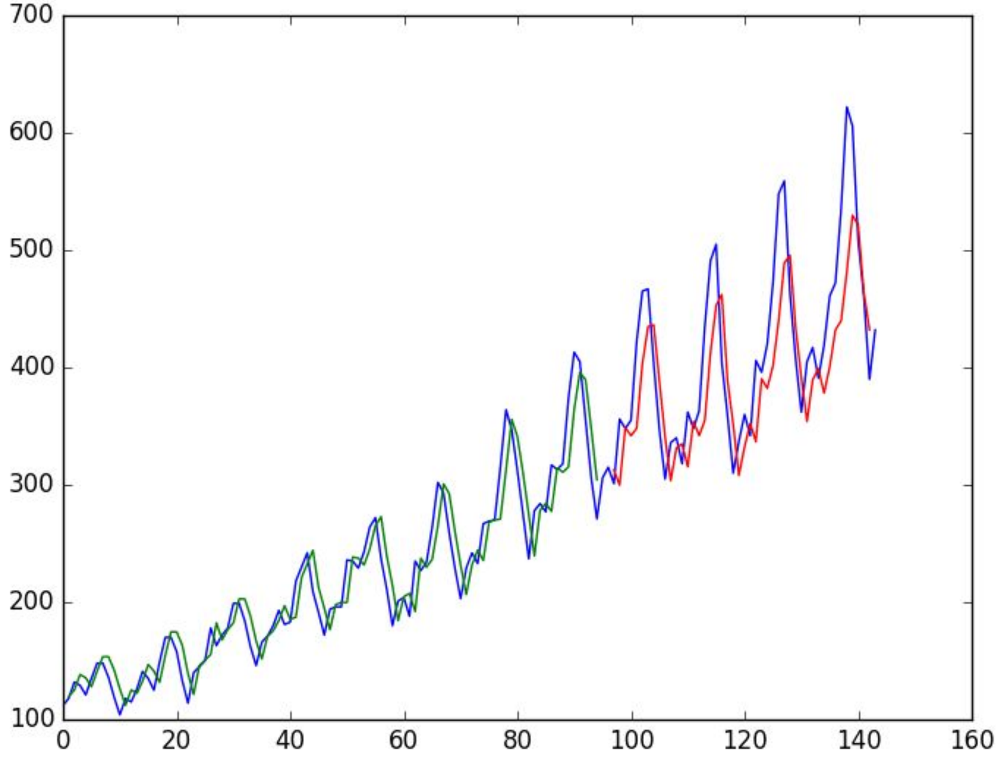
LSTM



Encoding and Decoding



Forecasting



Software

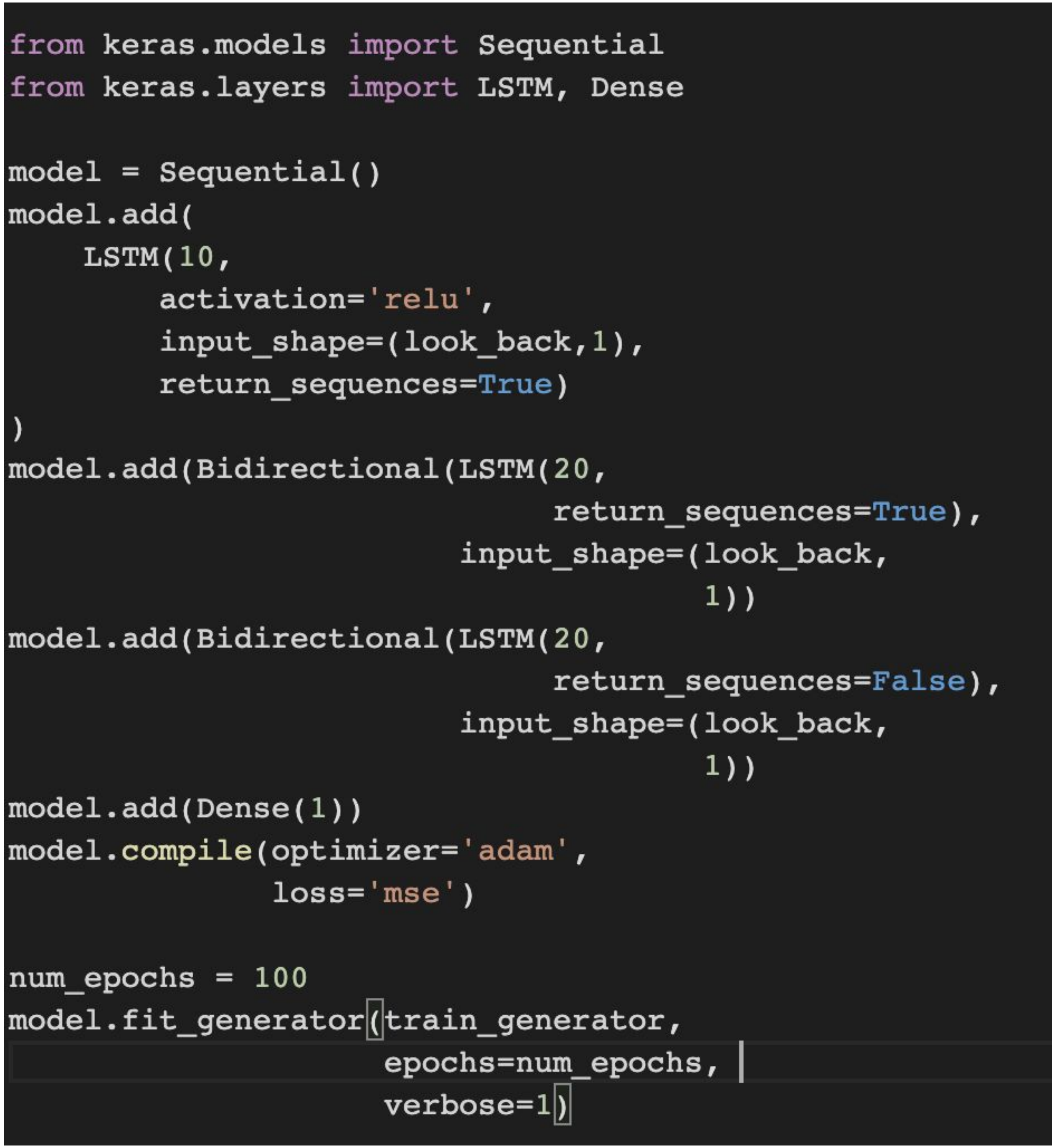
Software

Tensorflow

Keras

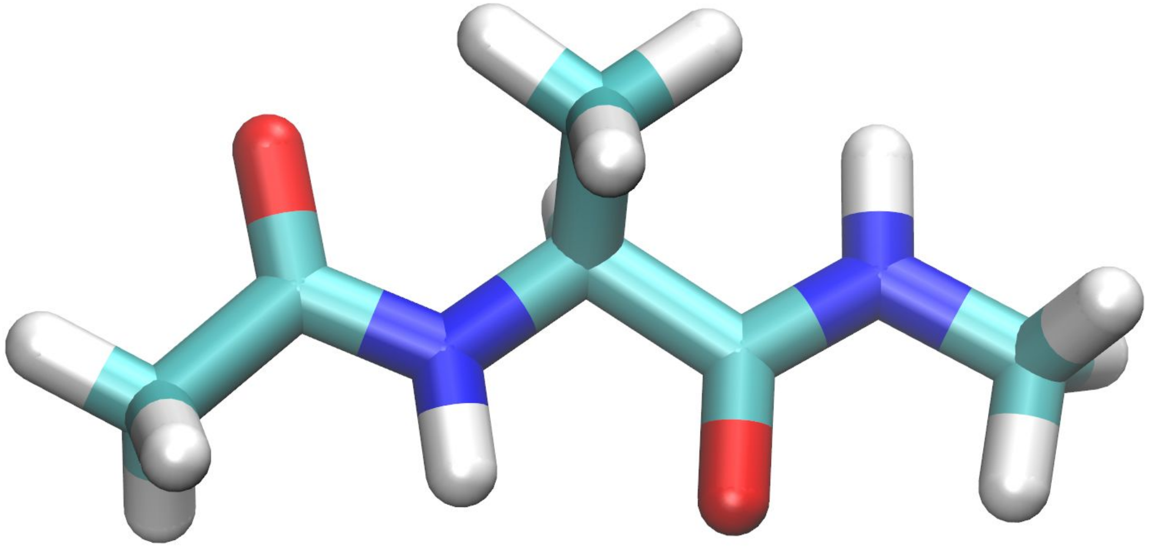
Time Series Generator

How to use TF and Keras

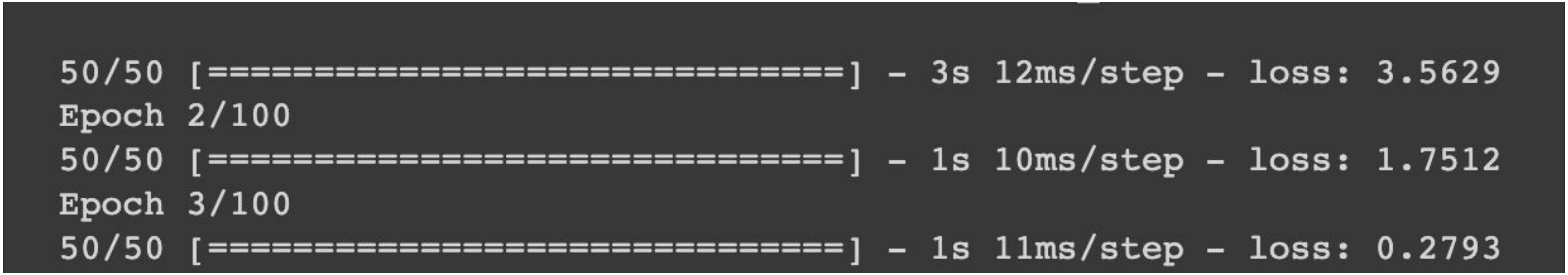


```
from keras.models import Sequential
from keras.layers import LSTM, Dense

model = Sequential()
model.add(
    LSTM(10,
        activation='relu',
        input_shape=(look_back,1),
        return_sequences=True)
)
model.add(Bidirectional(LSTM(20,
                             return_sequences=True),
                        input_shape=(look_back,
                                     1)))
model.add(Bidirectional(LSTM(20,
                             return_sequences=False),
                        input_shape=(look_back,
                                     1)))
model.add(Dense(1))
model.compile(optimizer='adam',
              loss='mse')

num_epochs = 100
model.fit_generator(train_generator,
                    epochs=num_epochs,
                    verbose=1)
```

Results

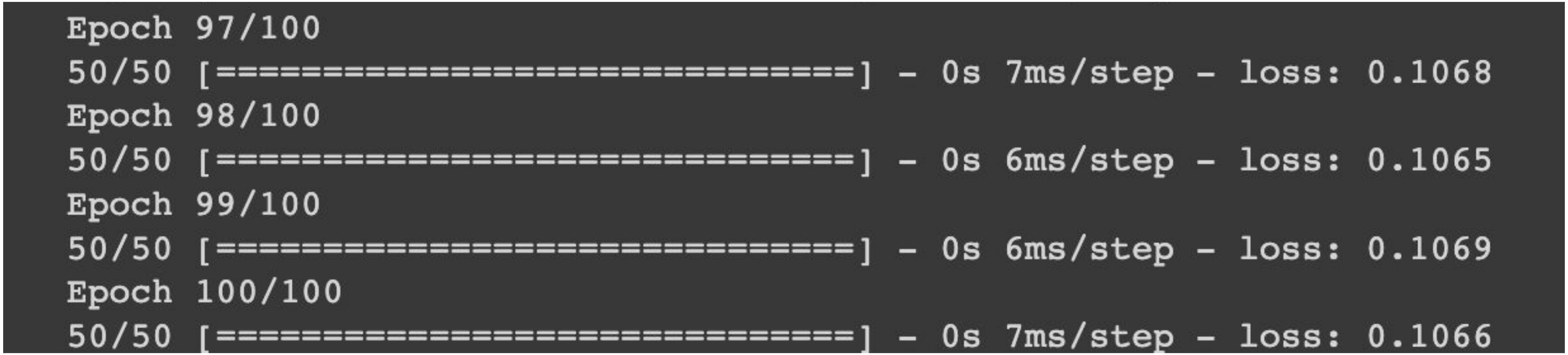


Univariate Results: Phi Angles



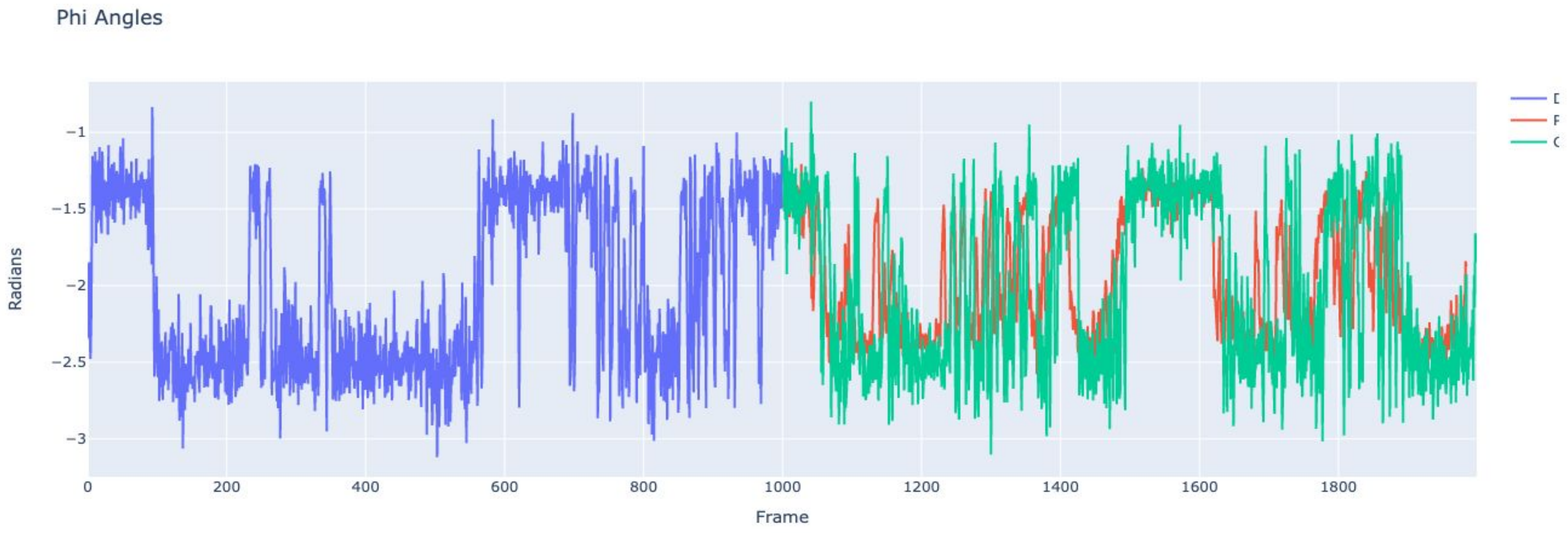
```
50/50 [==============================] - 3s 12ms/step - loss: 3.5629
Epoch 2/100
50/50 [==============================] - 1s 10ms/step - loss: 1.7512
Epoch 3/100
50/50 [==============================] - 1s 11ms/step - loss: 0.2793
```

A little bit later...

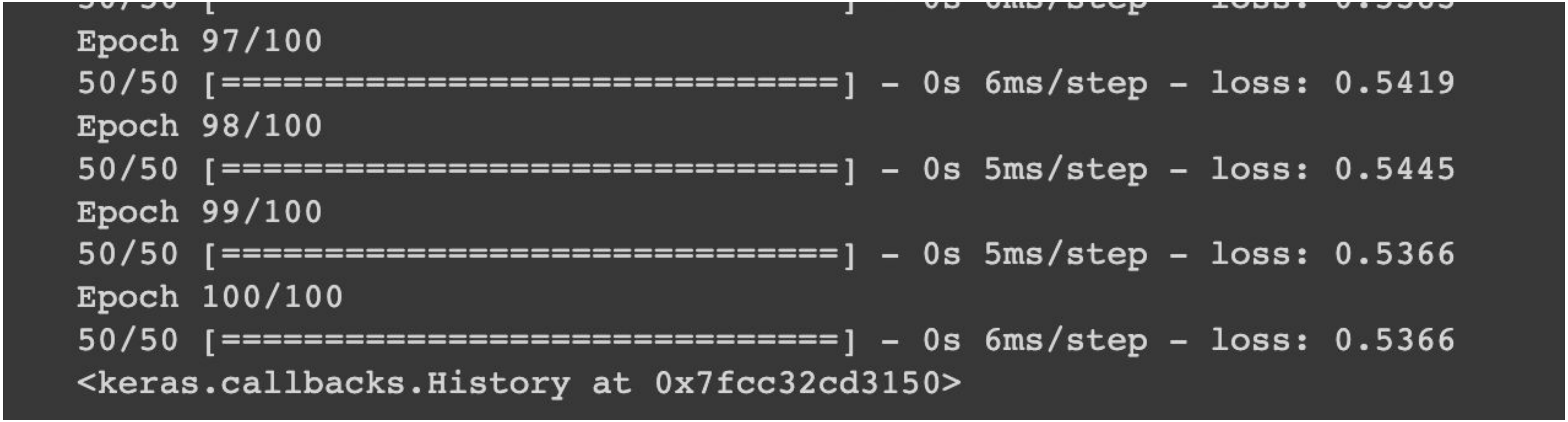


```
Epoch 97/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1068
Epoch 98/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1065
Epoch 99/100
50/50 [==============================] - 0s 6ms/step - loss: 0.1069
Epoch 100/100
50/50 [==============================] - 0s 7ms/step - loss: 0.1066
```

Univariate Results: Phi Angles

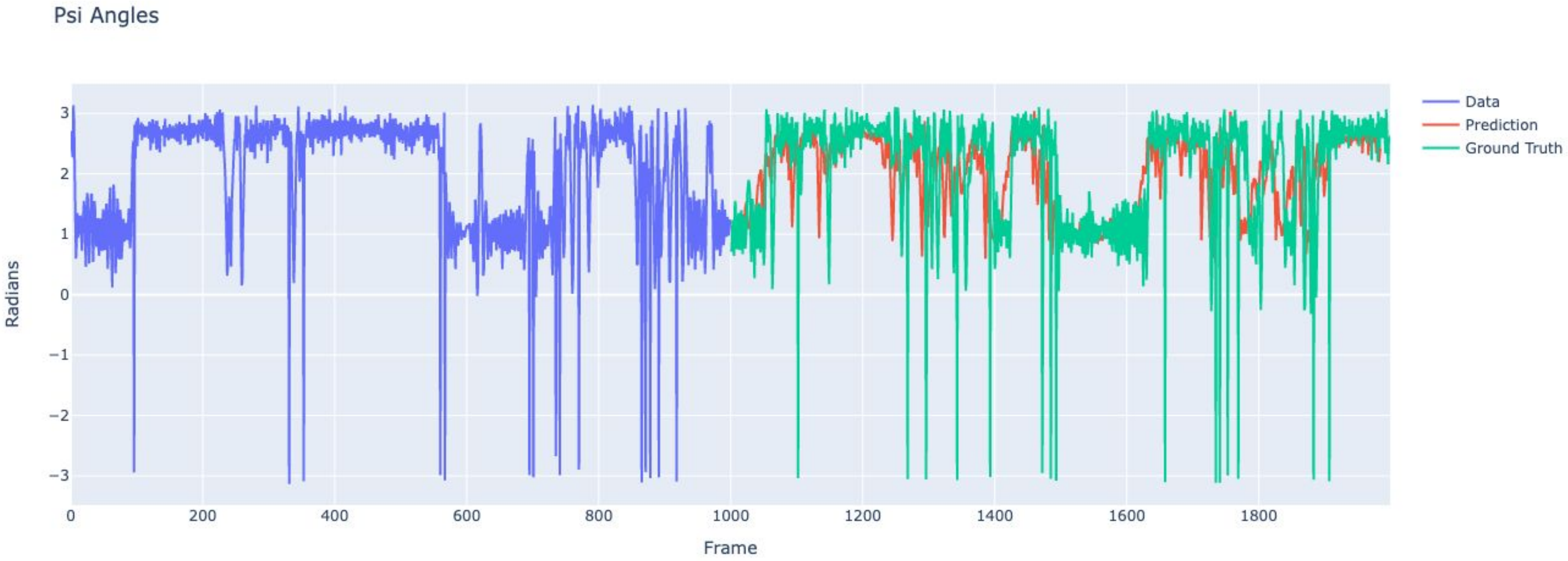


Univariate Results: Psi Angles



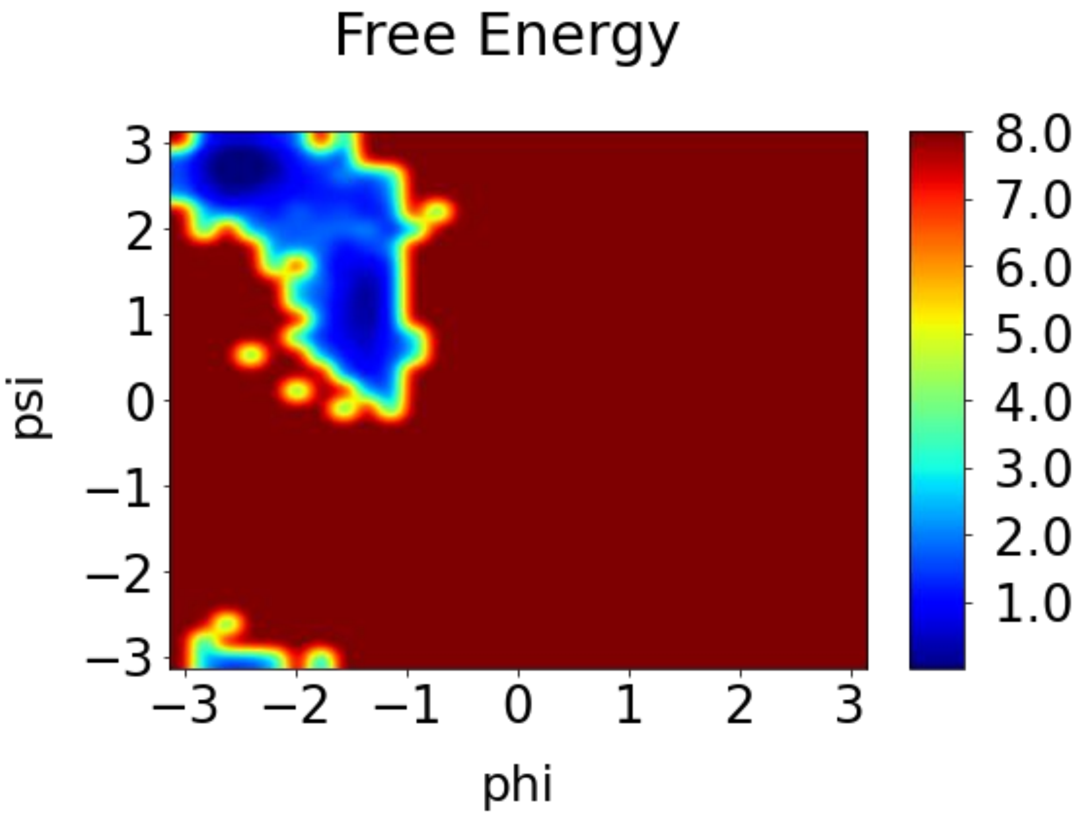
```
Epoch 97/100
50/50 [==============================] - 0s 6ms/step - loss: 0.5419
Epoch 98/100
50/50 [==============================] - 0s 5ms/step - loss: 0.5445
Epoch 99/100
50/50 [==============================] - 0s 5ms/step - loss: 0.5366
Epoch 100/100
50/50 [==============================] - 0s 6ms/step - loss: 0.5366
<keras.callbacks.History at 0x7fcc32cd3150>
```

Univariate Results: Psi Angles

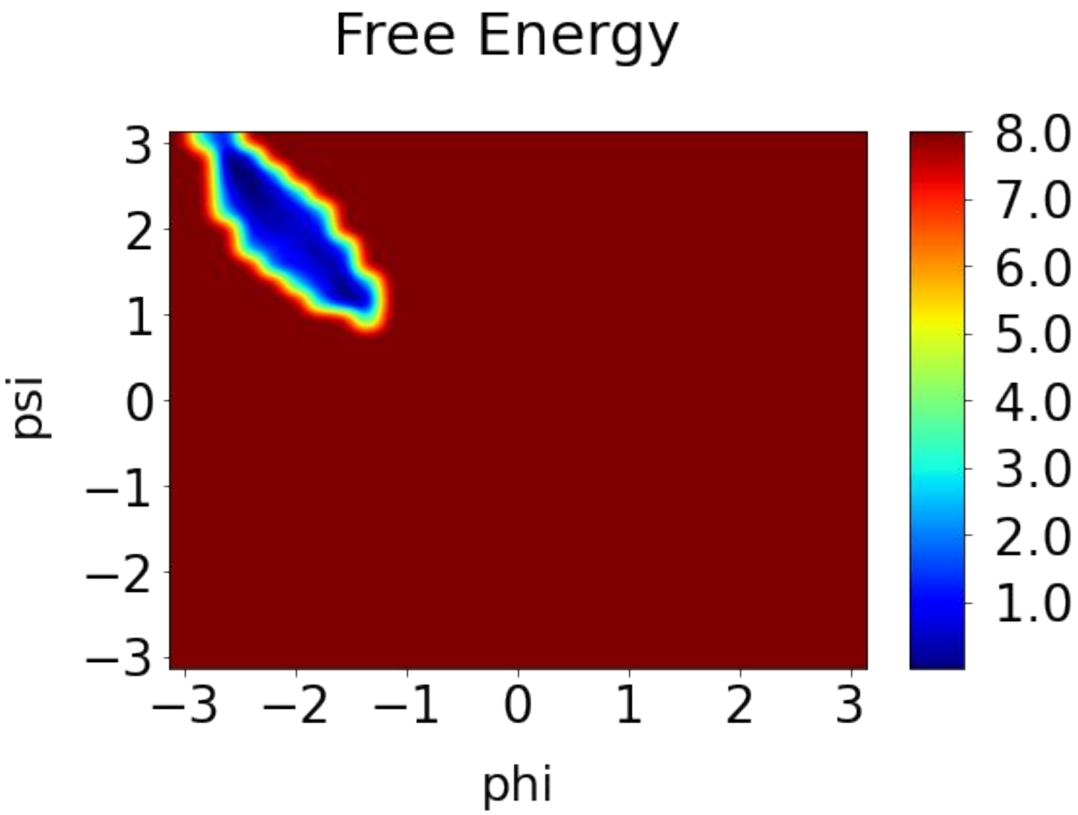


Free Energy Plot

Actual

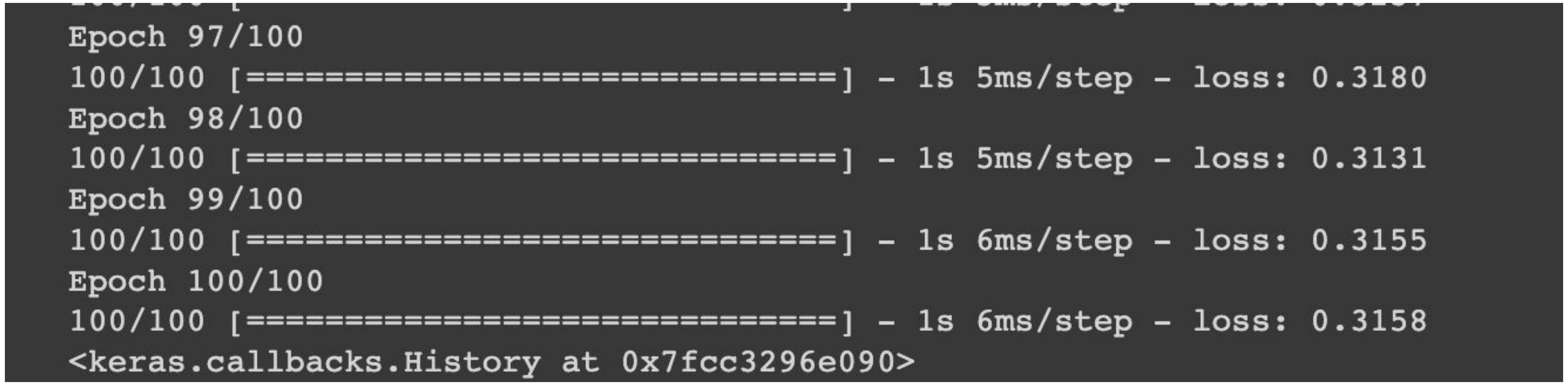


LSTM



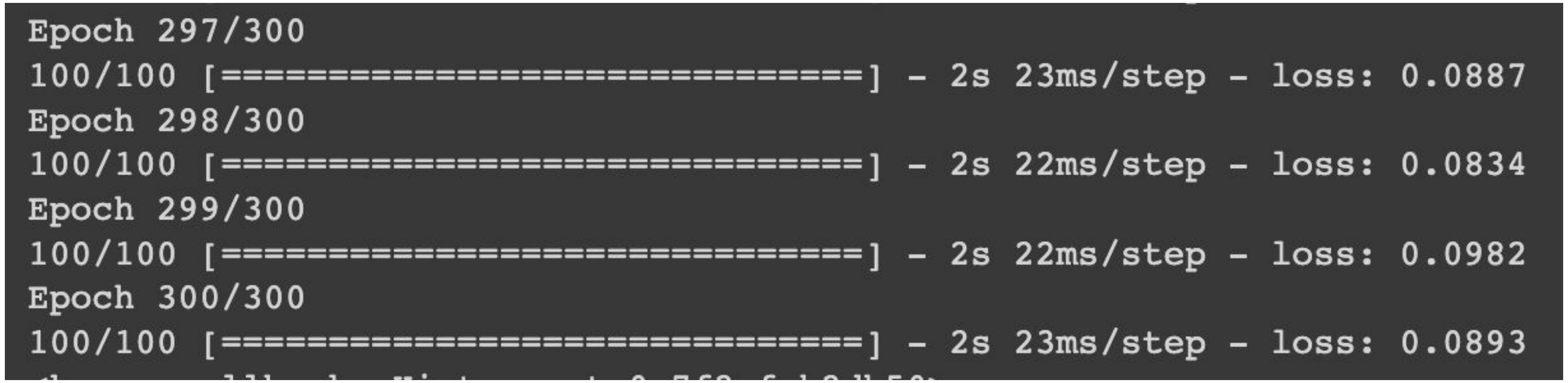
Improvements: Psi Angles

Multivariate LSTM:



```
Epoch 97/100
100/100 [==============================] - 1s 5ms/step - loss: 0.3180
Epoch 98/100
100/100 [==============================] - 1s 5ms/step - loss: 0.3131
Epoch 99/100
100/100 [==============================] - 1s 6ms/step - loss: 0.3155
Epoch 100/100
100/100 [==============================] - 1s 6ms/step - loss: 0.3158
<keras.callbacks.History at 0x7fcc3296e090>
```

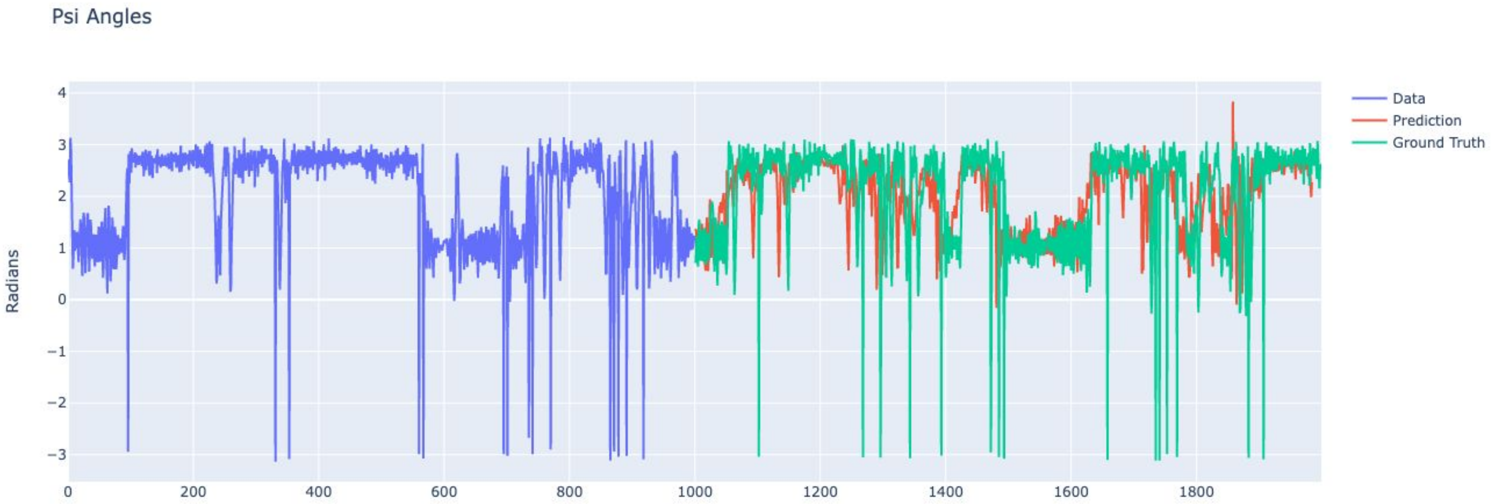
Multivariate stacked Bidirectional LSTM:



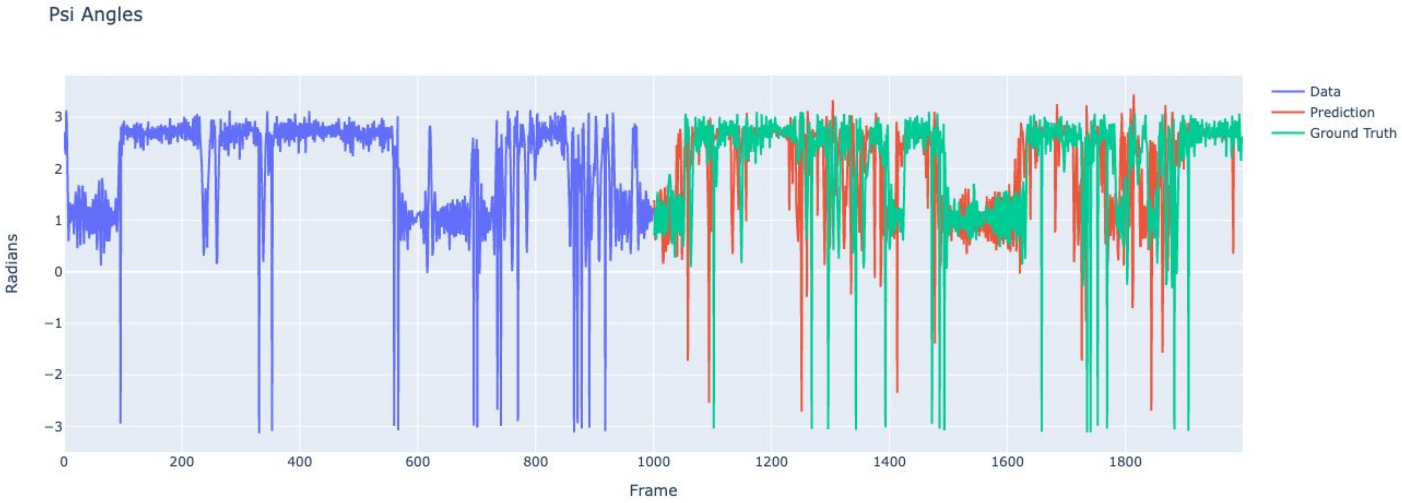
```
Epoch 297/300
100/100 [==============================] - 2s 23ms/step - loss: 0.0887
Epoch 298/300
100/100 [==============================] - 2s 22ms/step - loss: 0.0834
Epoch 299/300
100/100 [==============================] - 2s 22ms/step - loss: 0.0982
Epoch 300/300
100/100 [==============================] - 2s 23ms/step - loss: 0.0893
```

Improvements: Psi Angles

Multivariate LSTM:

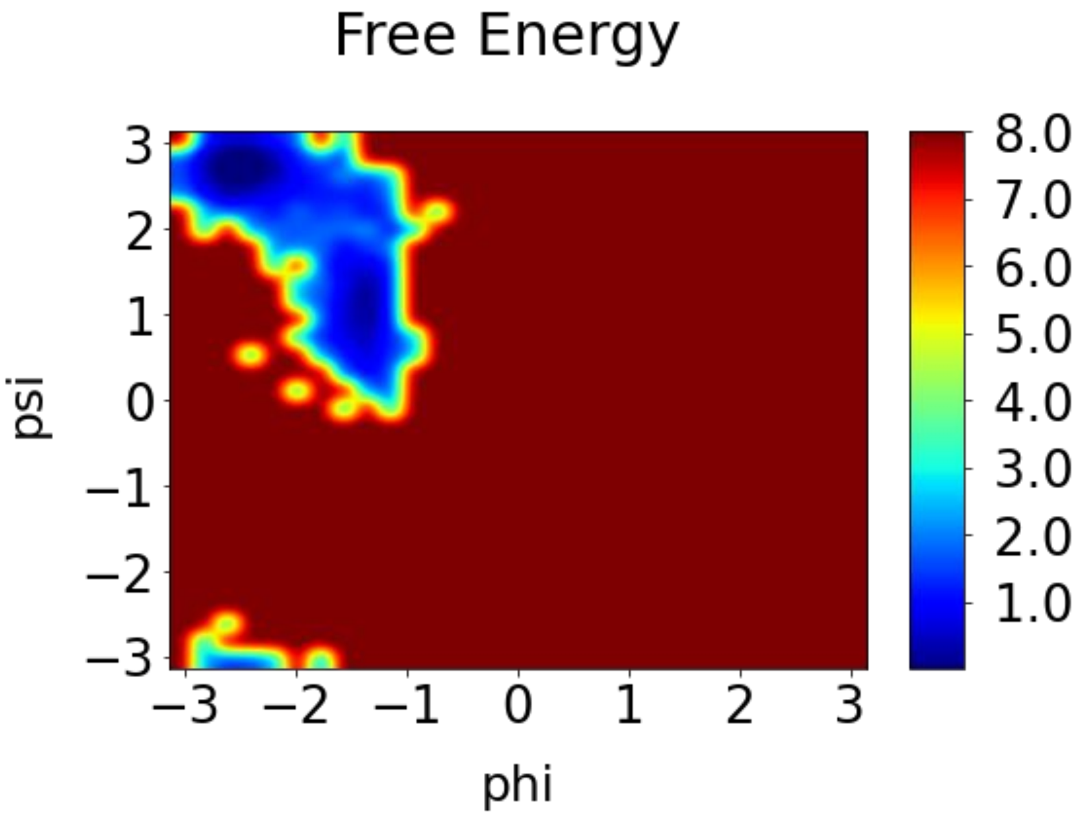


Multivariate Bidirectional LSTM:

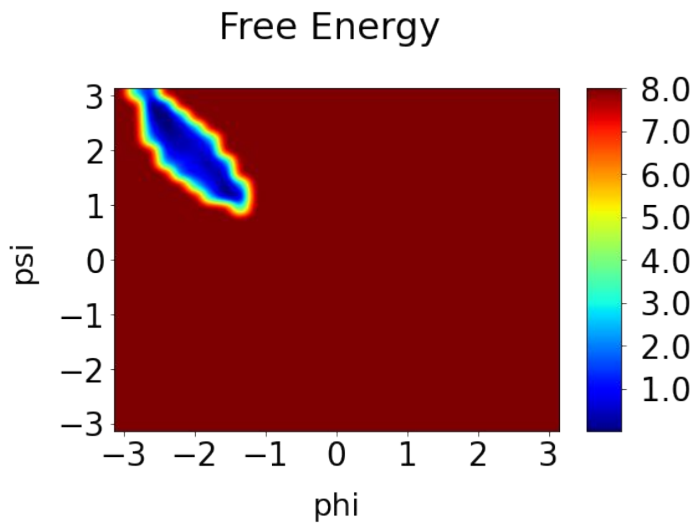


Better Free Energy Plot

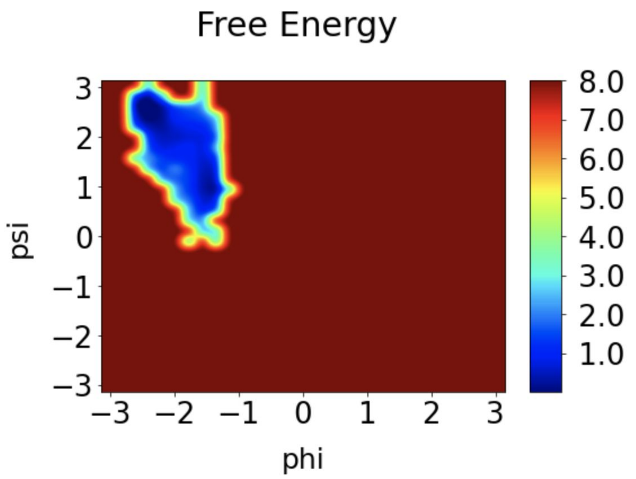
Actual



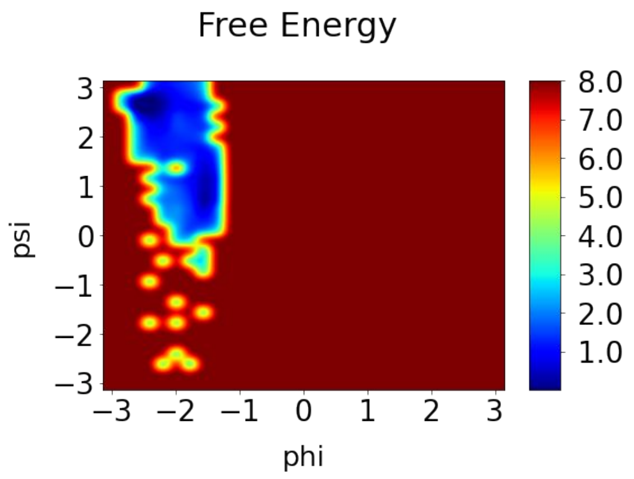
Vanilla LSTM



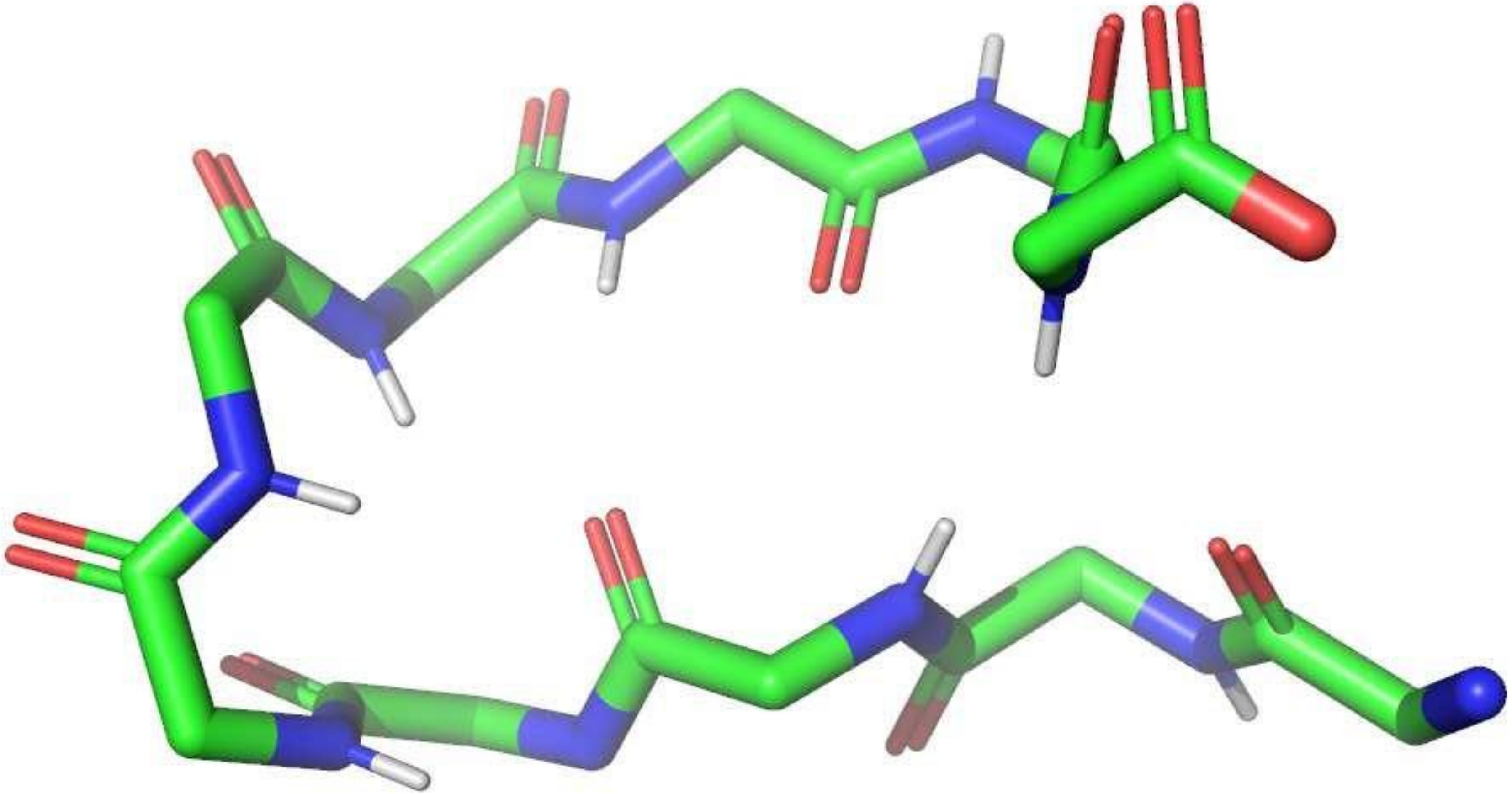
Multivariate LSTM



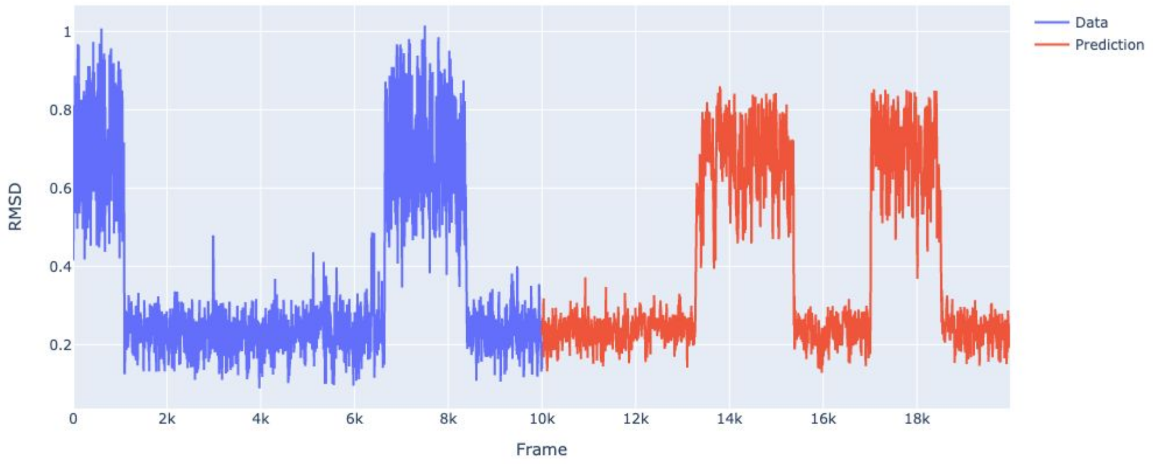
Multivariate SB-LSTM



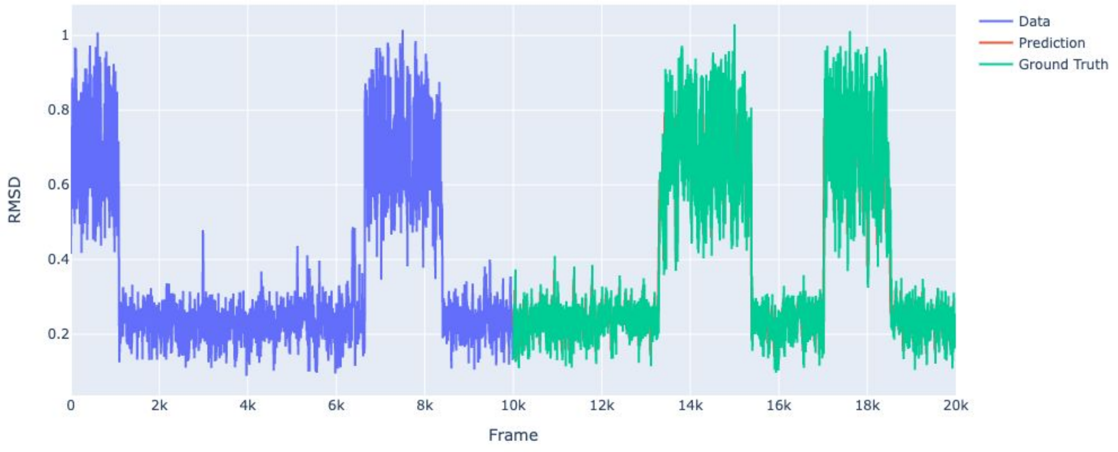
Chignolin



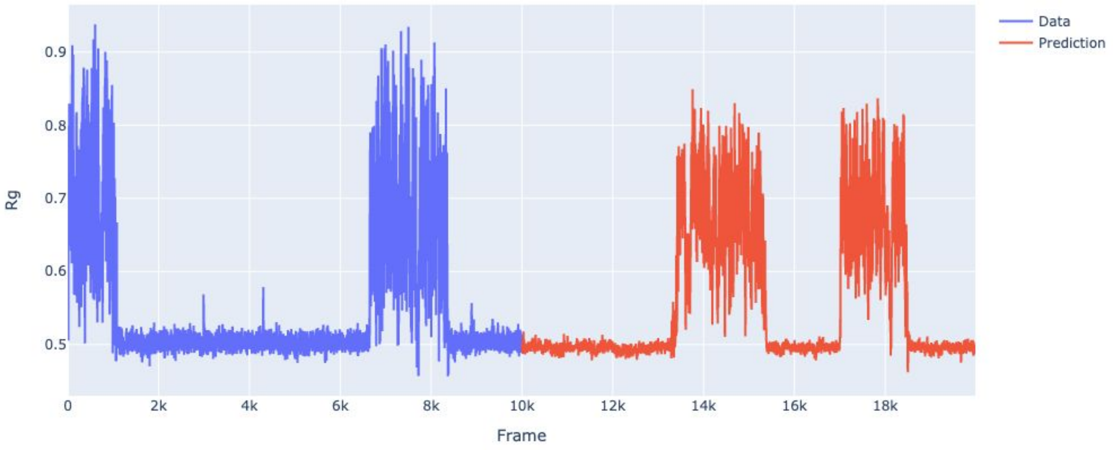
RMSD



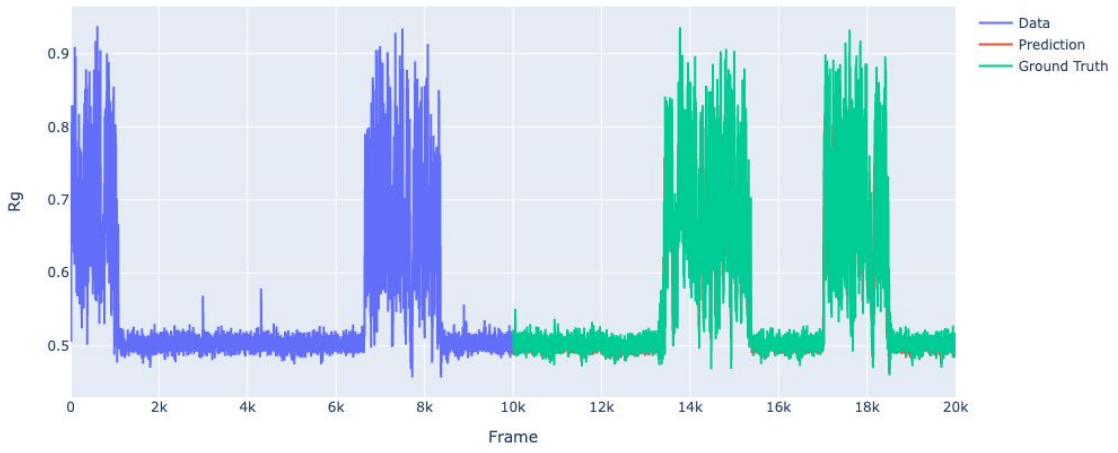
RMSD



Rg



Rg



Further Work

Longer Term predictions

Ab Initio Predictions

Transformer Model/ CNN-BS-LSTM Model

Tensor Processing Units

Thank you