Objectives

This document is trying to build a RNN model for viscoelasticity (VE), and viscoplasticity (VP). The idea is to use the similarity between RNN and history carrying material modeling, such as VE and VP. The internal state variables (ISV) resemble the hidden state variables in RNN units. This feature makes it naturally fit for the modeling of memory carrying material laws, for example VE and VP.

In [1]:

```
import numpy as np
from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.layers import LSTM, Dense, TimeDistributed
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from sklearn.model_selection import train_test_split
import random
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
# import the data
stress = np.loadtxt(’../../stress_3d_uniaxial.txt’)
strain = np.loadtxt(’../../strains_3d_uniaxial.txt’)
```

In [3]:

```
strain.shape
```

Out[3]:

```
(13000, 120)
```

In [4]:

```
stress.shape
```

Out[4]:

```
(13000, 120)
```

In [5]:

```
# train test split
X_train, X_test, y_train, y_test = train_test_split(strain, stress, test_size=0.1, ra
```

In [6]:

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

Out[6]:

```
((11700, 120), (1300, 120), (11700, 120), (1300, 120))
```

In [7]:

```
# define parameters
batch_size = 128
timesteps = 20
Features = 6
```

In [8]:

```
# reshape the data for fitting
X_train = np.reshape(X_train, (X_train.shape[0], timesteps, Features))
X_test = np.reshape(X_test, (X_test.shape[0], timesteps, Features))
```

In [9]:

```python
# define model
RNNmodel = Sequential()
RNNmodel.add(LSTM(50, return_sequences=True, input_shape=(timesteps, Features))) # ou
RNNmodel.add(LSTM(50, return_sequences=True)) # out put units = 20 without dense
RNNmodel.add(TimeDistributed(Dense(Features)))
```

Notes

1. The first parameter in LSTM function is the dimention of the hidden state variable (h), but not the time steps or the number of memory cells.Thus, from the mechanical point of view, if HSV is a strain-like state variable, then the dimension of it should be the same as the inputs of the model (namely the strain).
2. The number of features: in this case a 1-D VE problem is studied. In a multi-dimensional problem, the n_features can be 3 or 6 (stress/strain in 2D and 3D).

The hidden state variable (and cell state) can be obtained by:

```
lstm1, state_h, state_c = LSTM(1, return_state=True)
```

In [10]:

```python
RNNmodel.compile(loss='mse',
                optimizer='adam',
                metrics=['mae'])
```

In [11]:

```python
RNNmodel.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm (LSTM)                  (None, 20, 50)            11400     
_________________________________________________________________
lstm_1 (LSTM)                (None, 20, 50)            20200     
_________________________________________________________________
time_distributed (TimeDistri (None, 20, 6)             306       
=================================================================
Total params: 31,906
Trainable params: 31,906
Non-trainable params: 0
_________________________________________________________________
```

In [12]:

```python
# also needs to reshape y to use TimeDistributed
y_train = np.reshape(y_train, (y_train.shape[0], timesteps, Features))
y_test = np.reshape(y_test, (y_test.shape[0], timesteps, Features))
```

In [13]:

```
# add callbacks
# es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=10)
# mc = ModelCheckpoint('./best_model.h5', monitor='val_loss', mode='min', verbose=1, s

history = RNNmodel.fit(X_train, y_train, validation_split=0.2, epochs=1000, batch_siz
                                    #, callbacks=[es, mc])
```

```
Epoch 978/1000
9360/9360 [==============================] - 1s 100us/sample - loss: 4.1633e-07 - ma
e: 3.6944e-04 - val_loss: 2.6511e-07 - val_mae: 3.9789e-04
Epoch 979/1000
9360/9360 [==============================] - 1s 100us/sample - loss: 1.7384e-07 - ma
e: 2.5082e-04 - val_loss: 2.0165e-07 - val_mae: 3.0998e-04
Epoch 980/1000
9360/9360 [==============================] - 1s 98us/sample - loss: 4.8086e-07 - mae:
3.9714e-04 - val_loss: 4.6758e-08 - val_mae: 1.5595e-04
Epoch 981/1000
9360/9360 [==============================] - 1s 101us/sample - loss: 1.5012e-07 - ma
e: 2.1145e-04 - val_loss: 6.0419e-08 - val_mae: 1.5750e-04
Epoch 982/1000
9360/9360 [==============================] - 1s 101us/sample - loss: 5.1078e-07 - ma
e: 3.9688e-04 - val_loss: 3.5156e-07 - val_mae: 4.2247e-04
Epoch 983/1000
9360/9360 [==============================] - 1s 99us/sample - loss: 1.2437e-07 - mae:
2.0438e-04 - val_loss: 1.9921e-07 - val_mae: 3.0481e-04

Epoch 984/1000
9360/9360 [==============================] - 1s 103us/sample - loss: 4.4170e-07 - ma
```

In [14]:

```
# evaluate model
loss,mae = RNNmodel.evaluate(X_test, y_test)
loss,mae
```

```
1300/1300 [==============================] - 0s 203us/sample - loss: 3.2745e-07 - mae:
3.7912e-04
```

Out[14]:

```
(3.274522355066158e-07, 0.0003791195)
```

In [15]:

```
# save the history
history.history.keys()

history_df = pd.DataFrame.from_dict(history.history)
history_df.to_csv("history.csv", index=True)
```

In [16]:

```
history.history.keys()
```

Out[16]:

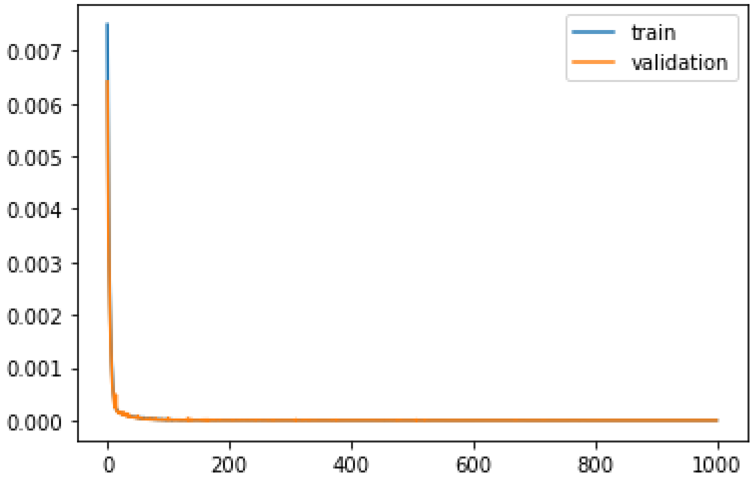
```
dict_keys(['loss', 'mae', 'val_loss', 'val_mae'])
```

In [17]:

```
# make plot
%matplotlib inline
plt.plot(history.history["loss"]) #['acc', 'loss'])
plt.plot(history.history["val_loss"])
plt.legend(['train', 'validation'])
```

Out[17]:

```
<matplotlib.legend.Legend at 0x15fb03b0b88>
```

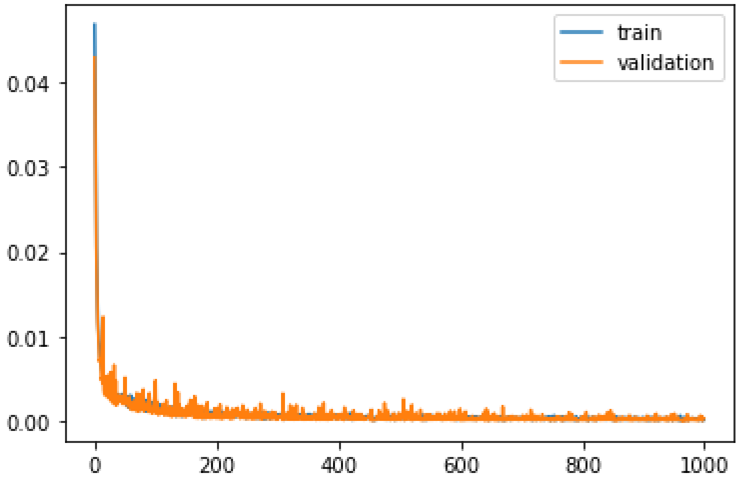


In [18]:

```
# make plot
%matplotlib inline
plt.plot(history.history["mae"])
plt.plot(history.history["val_mae"])
plt.legend(['train', 'validation'])
```

Out[18]:

```
<matplotlib.legend.Legend at 0x15fb03b03c8>
```



In [19]:

```
# make predictions to test the model
y_pred = RNNmodel.predict(X_test)
```

In [20]:

```
y_pred.shape
```

Out[20]:

```
(1300, 20, 6)
```

In [21]:

```
y_test.shape
```

Out[21]:

```
(1300, 20, 6)
```

In [22]:

```
X_test.shape, X_train.shape
```

Out[22]:

```
((1300, 20, 6), (11700, 20, 6))
```

In [23]:

```
# save the test data for future usage
np.save("strain_test.npy", X_test)
np.save("stress_test.npy", y_test)
np.save("stress_pred.npy", y_pred)
```

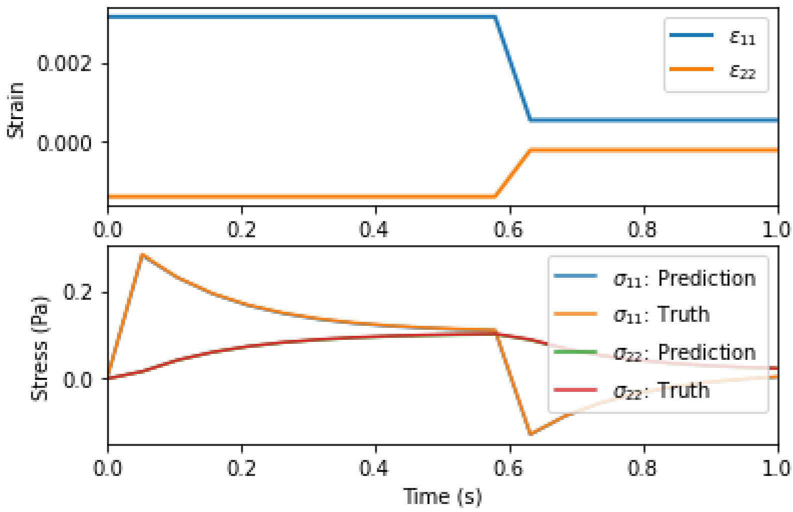
In [24]:

```
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three comp
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[936]
```

Out[24]:

```
Text(0, 0.5, 'Stress (Pa)')
```



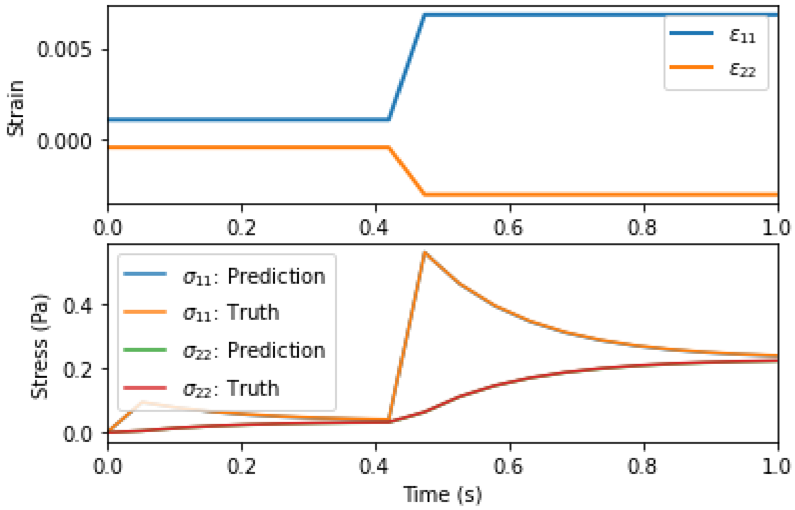
In [25]:

```
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three comp
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[480]
```

Out[25]:

```
Text(0, 0.5, 'Stress (Pa)')
```



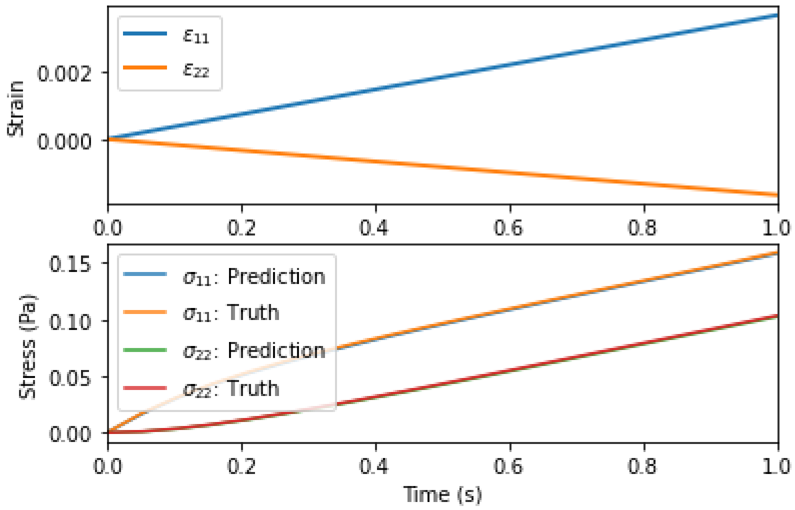
In [26]:

```python
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three com
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[151]
```

Out[26]:

```
Text(0, 0.5, 'Stress (Pa)')
```



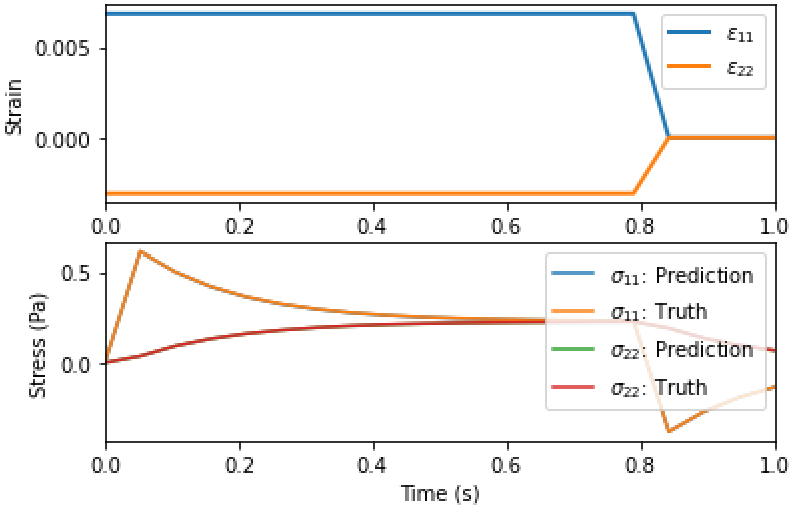
In [27]:

```python
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three comp
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[829]
```

Out[27]:

```
Text(0, 0.5, 'Stress (Pa)')
```



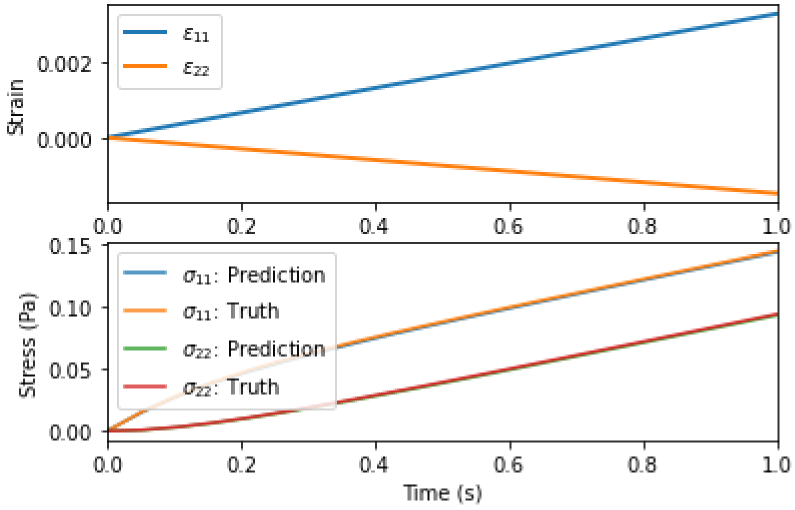
In [28]:

```python
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three com
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[509]
```

Out[28]:

```
Text(0, 0.5, 'Stress (Pa)')
```



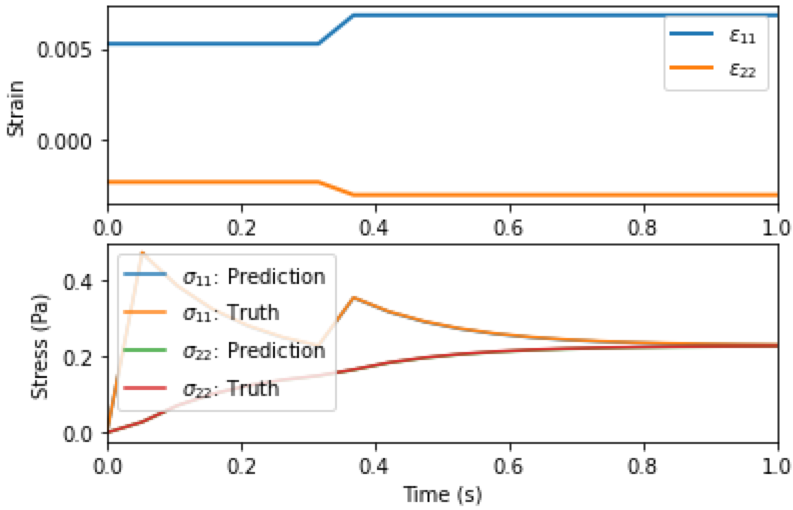
In [29]:

```
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three com
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[680]
```

Out[29]:

```
Text(0, 0.5, 'Stress (Pa)')
```



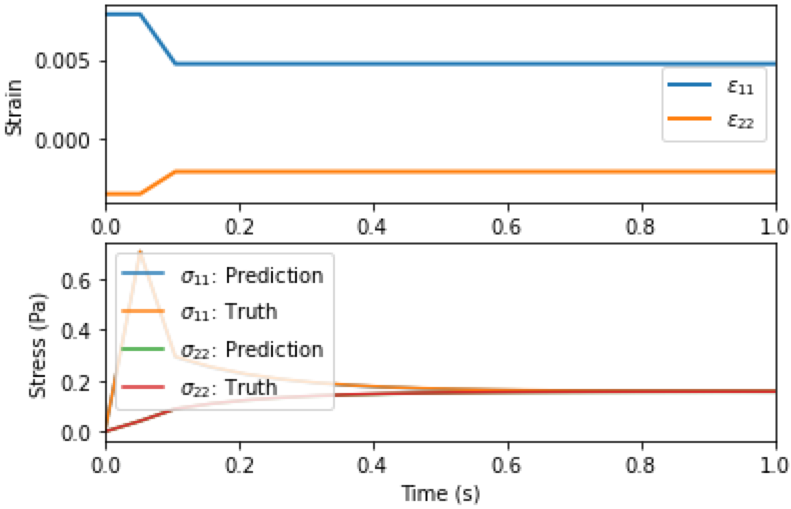
In [30]:

```python
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three com
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[934]
```

Out[30]:

```
Text(0, 0.5, 'Stress (Pa)')
```



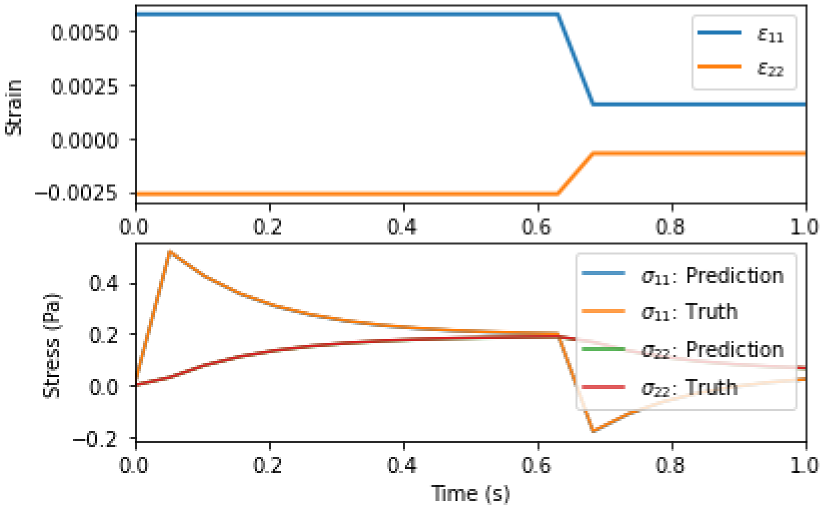
In [31]:

```python
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three comp
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[666]
```

Out[31]:

```
Text(0, 0.5, 'Stress (Pa)')
```



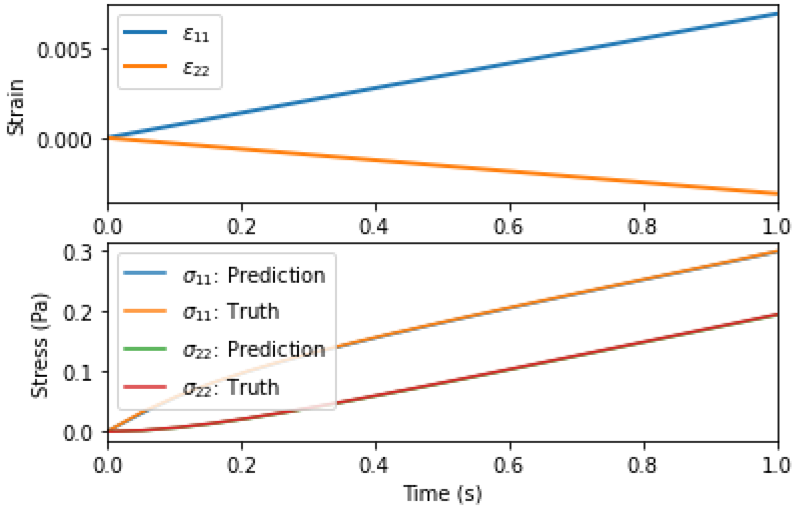
In [32]:

```
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three comp
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[492]
```

Out[32]:

```
Text(0, 0.5, 'Stress (Pa)')
```



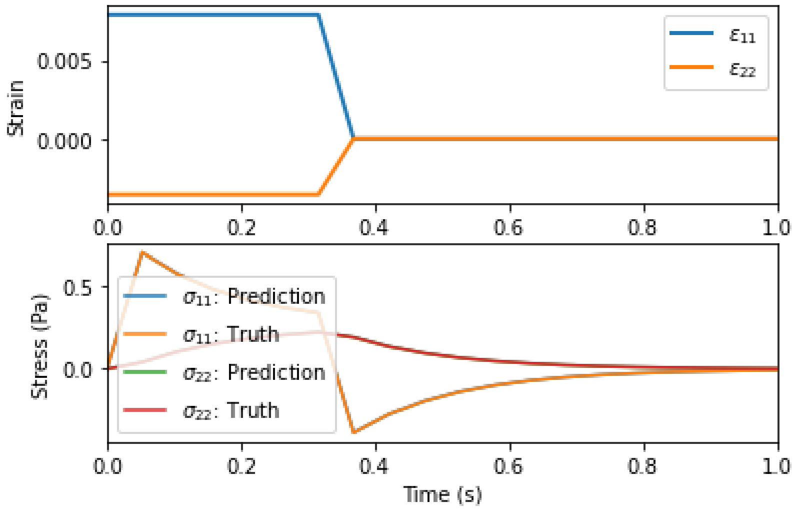
In [33]:

```python
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three com
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[894]
```

Out[33]:

```
Text(0, 0.5, 'Stress (Pa)')
```



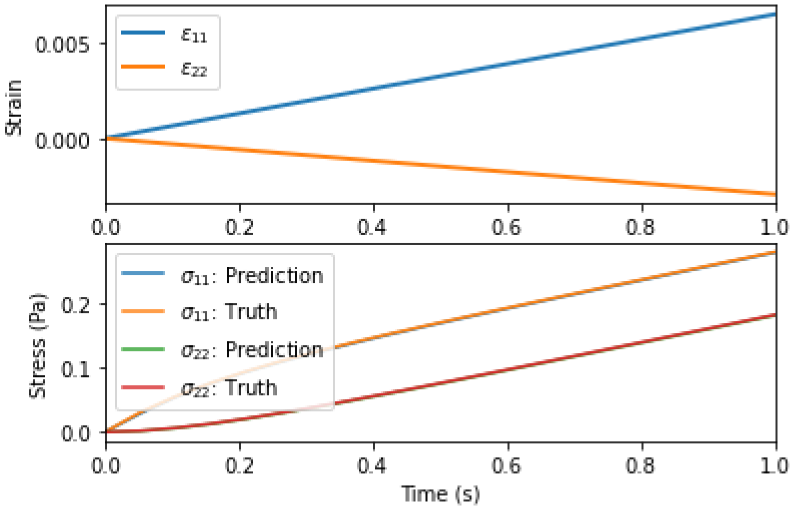
In [34]:

```python
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three comp
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[760]
```

Out[34]:

```
Text(0, 0.5, 'Stress (Pa)')
```



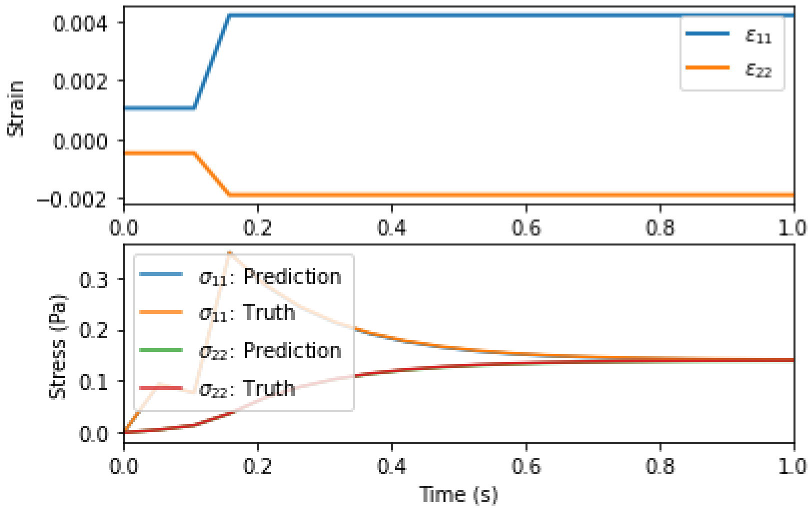
In [35]:

```python
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three comp
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[976]
```

Out[35]:

```
Text(0, 0.5, 'Stress (Pa)')
```



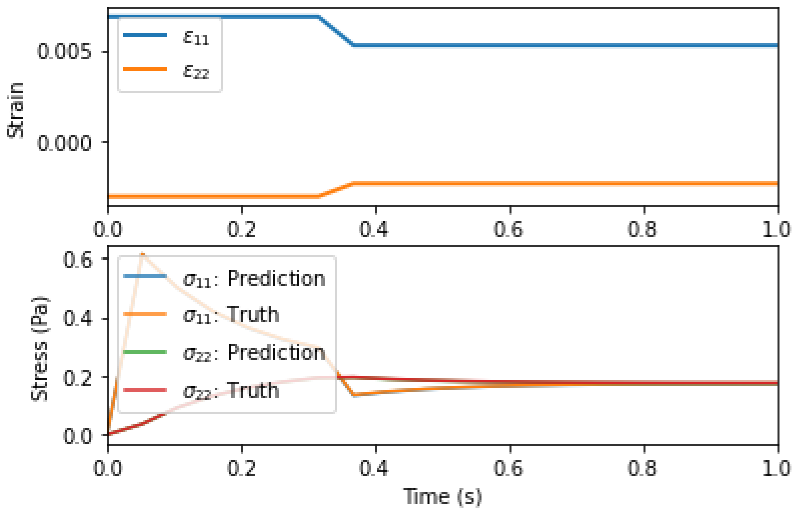
In [36]:

```python
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three com
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[665]
```

Out[36]:

```
Text(0, 0.5, 'Stress (Pa)')
```



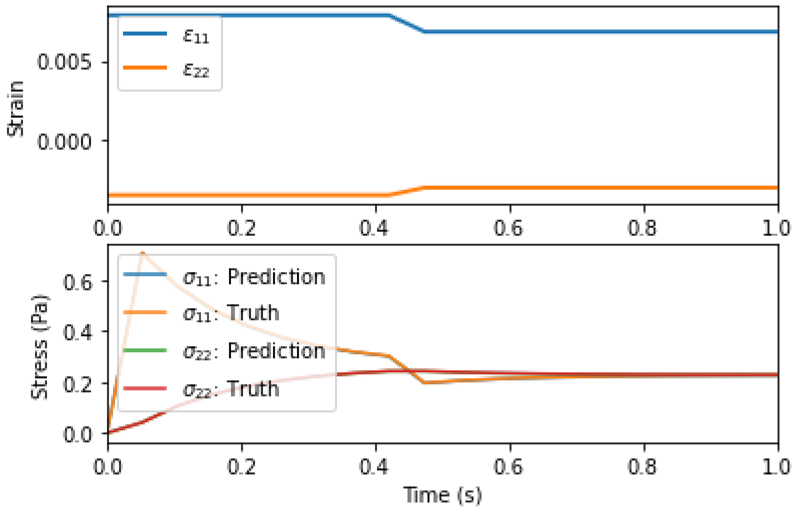
In [37]:

```
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three comp
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[284]
```

Out[37]:

```
Text(0, 0.5, 'Stress (Pa)')
```



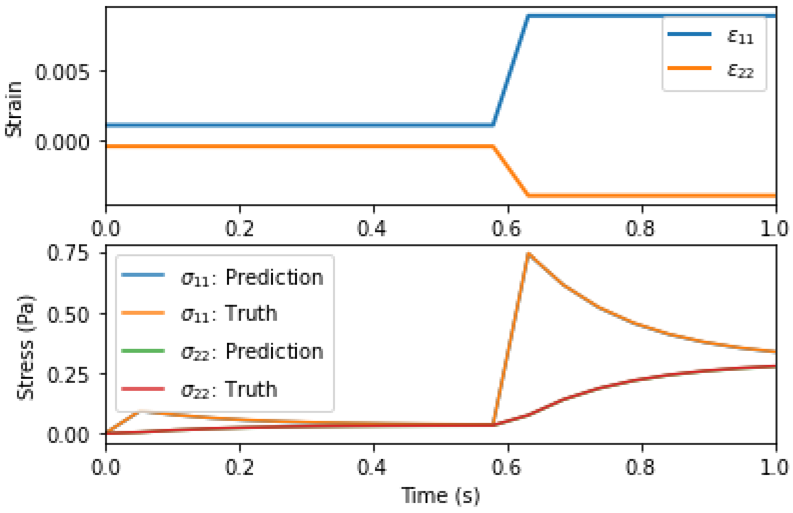
In [38]:

```python
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three com
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[177]
```

Out[38]:

```
Text(0, 0.5, 'Stress (Pa)')
```



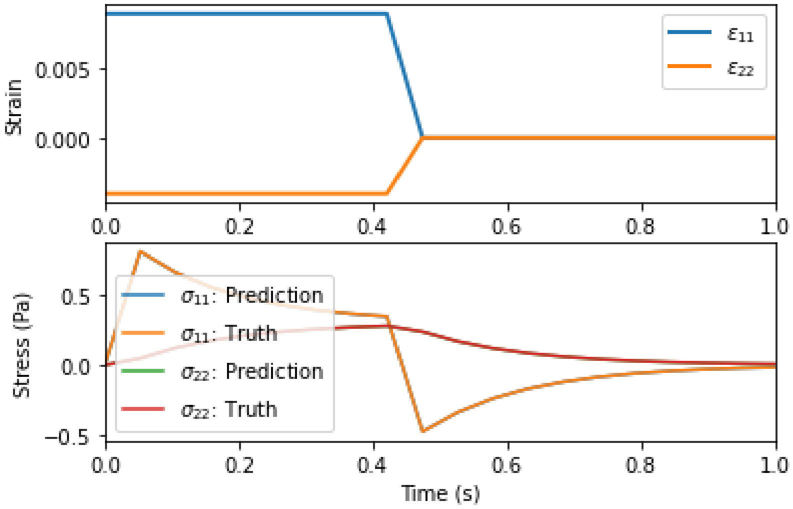
In [39]:

```python
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three comp
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[1059]
```

Out[39]:

```
Text(0, 0.5, 'Stress (Pa)')
```



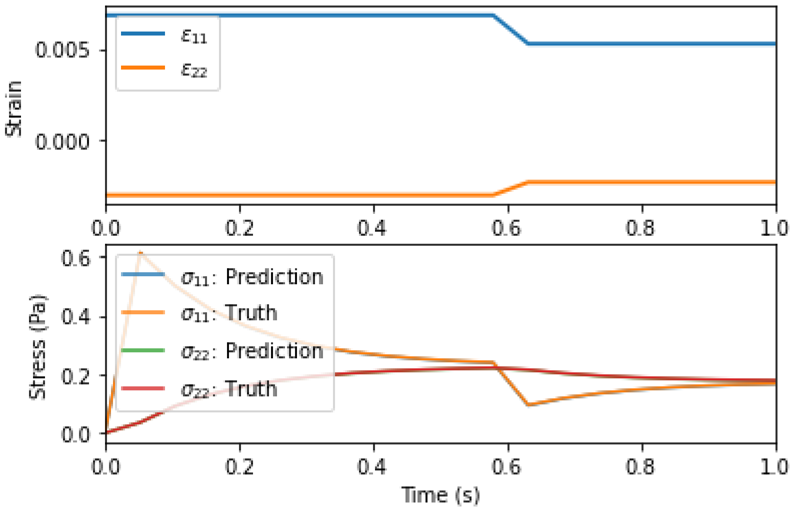
In [40]:

```python
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three comp
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[850]
```

Out[40]:

```
Text(0, 0.5, 'Stress (Pa)')
```



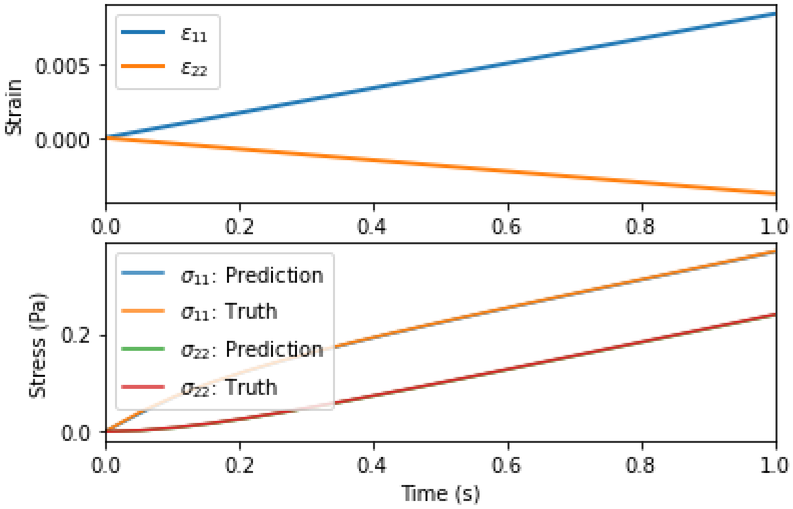
In [41]:

```
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three comp
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[175]
```

Out[41]:

```
Text(0, 0.5, 'Stress (Pa)')
```



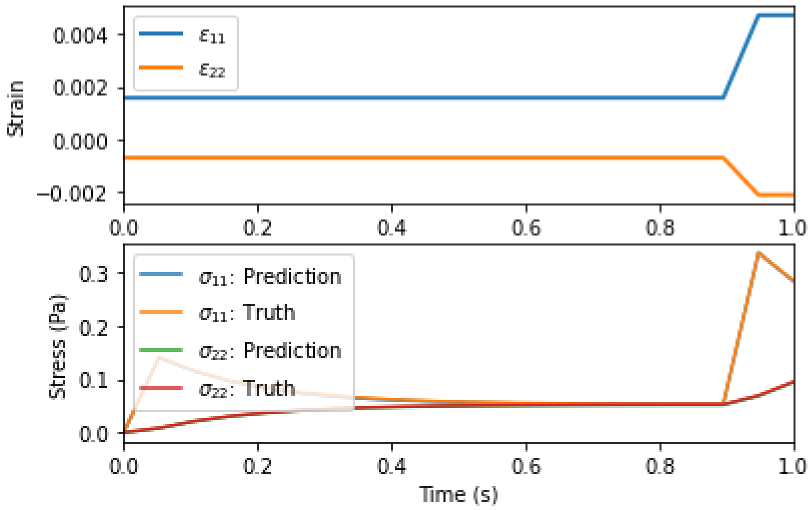
In [42]:

```python
# select random index
idx = random.sample(range(0,y_test.shape[0]), 1)
print(idx)

# time sequence
time_vec = np.linspace(0, 1, 20, endpoint=True)
strain_vec = np.reshape(X_test[idx, :, 0:3], (timesteps,3))
stress_pred_vec = np.reshape(y_pred[idx,:, 0:3], (timesteps,3)) # the first three com
stress_test_vec = np.reshape(y_test[idx,:, 0:3], (timesteps,3))

fig, axs = plt.subplots(2)
axs[0].plot(time_vec, strain_vec[:,0], lw=2.0, label="$\epsilon_{11}$")
axs[0].plot(time_vec, strain_vec[:,1], lw=2.0, label="$\epsilon_{22}$")
axs[0].legend()
axs[0].set_xlim(0,1)
axs[0].set_ylabel("Strain")
axs[1].plot(time_vec, stress_pred_vec[:,0], label="$\sigma_{11}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,0], label="$\sigma_{11}$: Truth")
axs[1].plot(time_vec, stress_pred_vec[:,1], label="$\sigma_{22}$: Prediction")
axs[1].plot(time_vec, stress_test_vec[:,1], label="$\sigma_{22}$: Truth")
# axs[1].plot(time_vec, stress_pred_vec[:,2], label="$\sigma_{33}$: Prediction")
# axs[1].plot(time_vec, stress_test_vec[:,2], label="$\sigma_{33}$: Truth")
axs[1].legend()
axs[1].set_xlim(0,1)
axs[1].set_xlabel("Time (s)")
axs[1].set_ylabel("Stress (Pa)")
```

```
[1294]
```

Out[42]:

```
Text(0, 0.5, 'Stress (Pa)')
```



This results show that the performance is more improved with 1000 epochs compared to that of the previous model (with early stopping and with 500 epochs training)

In [43]:

```
# save model
RNNmodel.save("best_model.h5")
```

In [ ]:

```

```