Final Project – COM S 574

IOWA STATE UNIVERSITY

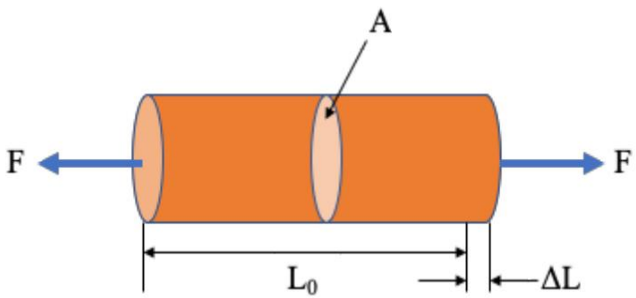
Convolutional Neural Network (CNN) to predict the Elastic Modulus of Particulate Composite Microstructures

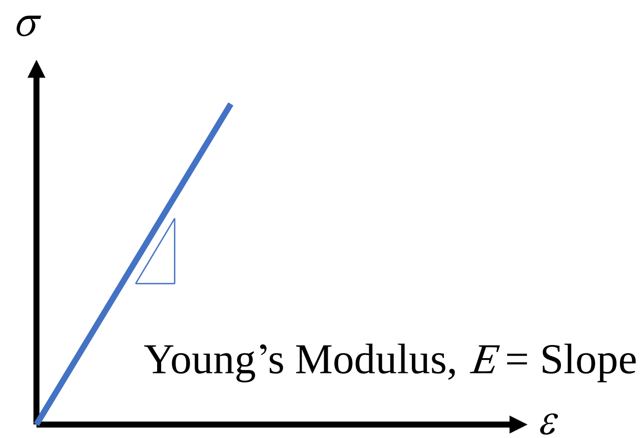
Roozbeh Eghbalpoor

May 4, 2023

Introduction

- What is Elastic Modulus (or Young's Modulus)?
  - ✓ A Mechanical Property represented by *E*
  - ✓ Tells how a material deforms according to the applied load
  - ✓ Higher Young's Modulus, less deformation
  - ✓ $$\underbrace{\frac{\Delta L}{L_0}}_{\text{change in length, } \varepsilon} = \underbrace{\frac{1}{E}}_{\text{elasticity}} \times \underbrace{\frac{F}{A}}_{\text{pressure, } \sigma}$$





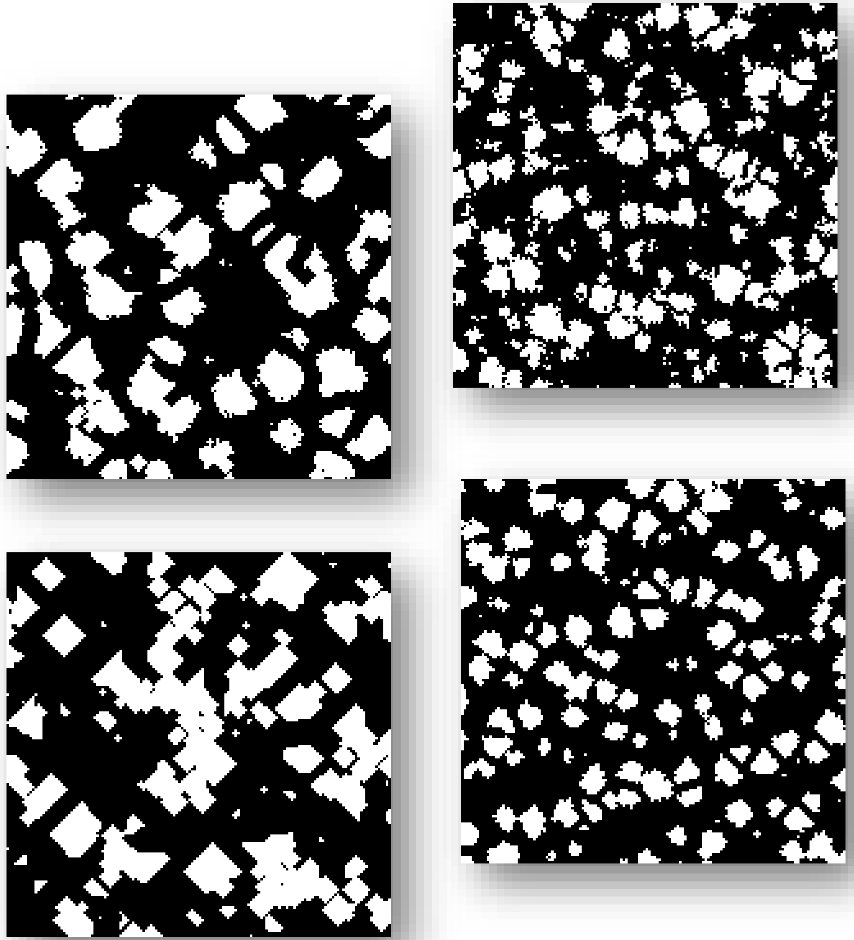
Introduction

- What is a Composite Material?
  - ✓ A combination of two (or more) materials with different physical and chemical properties
  - ✓ They are produced for a specific job like higher strength-to-weight ratio, higher (or lower) electric conductivity, higher (or lower) heat transfer characteristic
  - ✓ Fiber-reinforced composites and Particulate Composites

Microstructures

IOWA STATE UNIVERSITY

- Design parameters of a particulate composites:
  - ✓ Volume Fraction (VF)
  - ✓ Aspect Ratio (AR)
  - ✓ Average particle size
  - ✓ Distribution of particles



Data generation

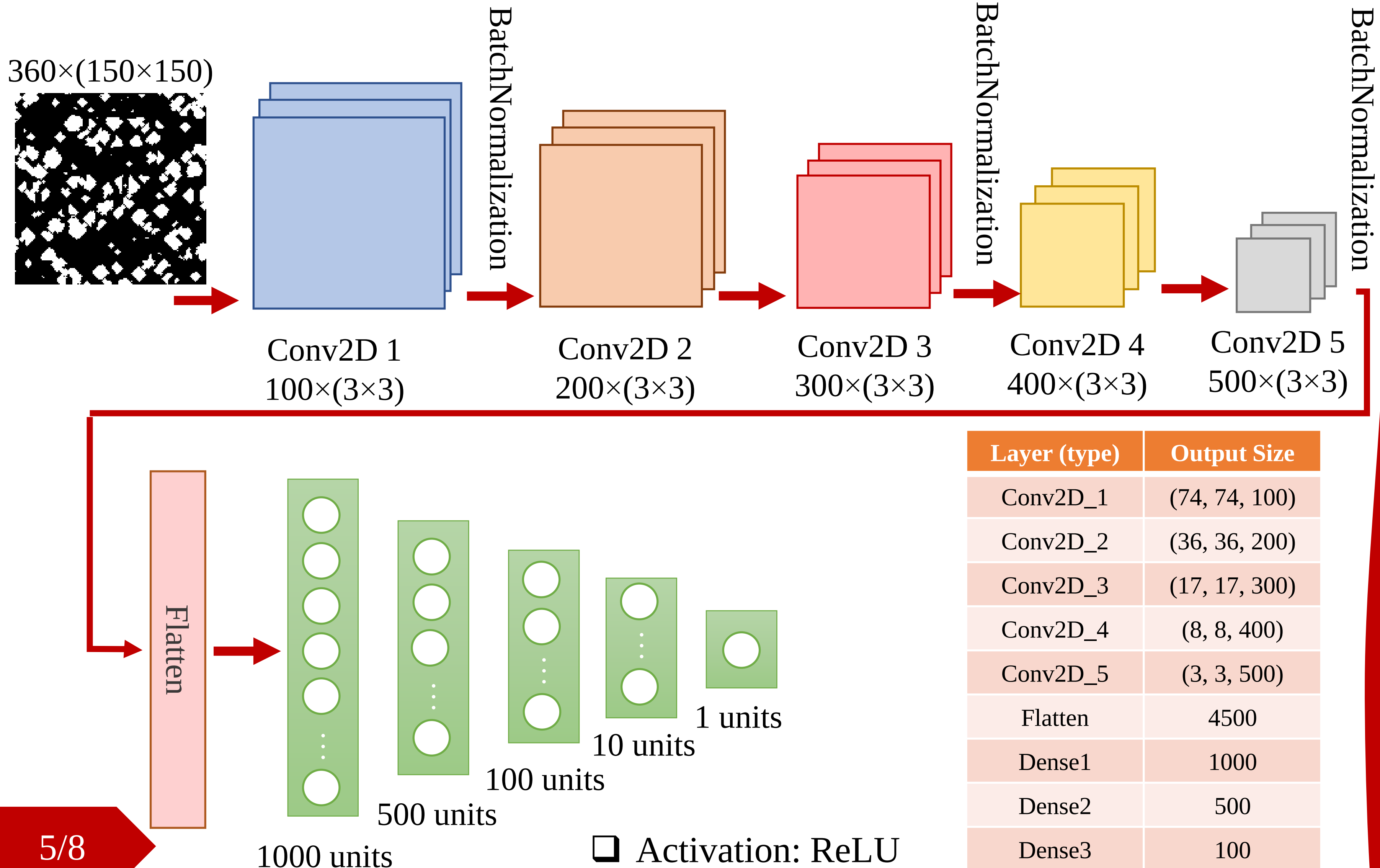
IOWA STATE UNIVERSITY

- Each microstructure (VF30%) is analyzed with a computational framework called PeriDynamics (PD)
- A small deformation is applied and Boundary force is measured
- Elastic Modulus (E) is obtained by previous equation

# microstructures	Min. E (Pa)	Max. E (Pa)
406	317×10^6	552×10^6

CNN Architecture

Iowa State University

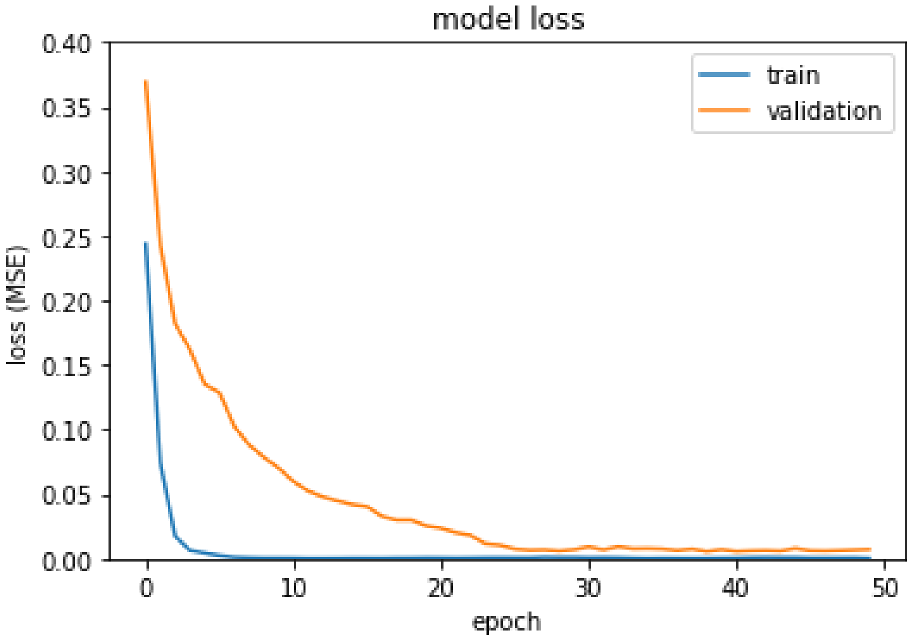


- Activation: ReLU
- Loss: MSE

Layer (type)	Output Size
Conv2D_1	(74, 74, 100)
Conv2D_2	(36, 36, 200)
Conv2D_3	(17, 17, 300)
Conv2D_4	(8, 8, 400)
Conv2D_5	(3, 3, 500)
Flatten	4500
Dense1	1000
Dense2	500
Dense3	100
Dense4	10

Model train and Output

- 10% test and 15% validation set
- Adam optimizer with 5.0e-4 learning rate
- Batch size: 20
- Train MSE = 4.9437e-04
- Validation MSE = 0.0055
- Test MSE = 0.0073



Code Structure


```python
from PIL import Image
from matplotlib import pyplot as plt
import numpy as np
import os
import tensorflow as tf
from tensorflow.keras.models import Sequential
```

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

tf.keras.optimizers.Adam(learning_rate=0.0005)

history = model.fit(x_train, y_train, epochs=50, batch_size=20, validation_split=0.15)
```

```python
image_dir = 'C:/Users/reghb/OneDrive/Desktop/images30%/'
images = []
for filename in os.listdir(image_dir):
    if filename.endswith('.png'):
        image = Image.open(os.path.join(image_dir, filename))
        image_data = np.asarray(image)
        images.append(image_data)
x_data = np.array(images)
y_data = np.loadtxt('stiffness.txt')
y_data = y_data/max(y_data)
y_data = y_data.reshape((-1, 1))
x_train=x_data[0:360,:,:]
y_train=y_data[0:360]
x_test=x_data[360:,:]
y_test=y_data[360:]
```

```python
test_loss= model.evaluate(x_test, y_test)
y_test_pred = model.predict(x_test)

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss (MSE)')
plt.xlabel('epoch')
plt.ylim([0, 0.4])
plt.legend(['train', 'validation'], loc='upper right')
plt.show()
```

```python
model = Sequential([
    Conv2D(100, (3,3), strides=(2, 2), activation='relu', input_shape=(150, 150, 1)),
    BatchNormalization(),
    Conv2D(200, (3,3), strides=(2, 2), activation='relu'),
    Conv2D(300, (3,3), strides=(2, 2), activation='relu'),
    BatchNormalization(),
    Conv2D(400, (3,3), strides=(2, 2), activation='relu'),
    Conv2D(500, (3,3), strides=(2, 2), activation='relu'),
    BatchNormalization(),
    Flatten(),
    Dense(1000, activation='relu'),
    Dense(500, activation='relu'),
    Dense(100, activation='relu'),
    Dense(10, activation='relu'),
    Dense(1, activation='linear')
])
```

Future works

- Increase the number of input data to have a more efficient trained model.
- Use convolutional Neural Network model proposed in this project to predict other mechanical properties like maximum strength.
- Compare model performance with other machine learning methods.

https://github.com/roozbeh191/COMS574.git

Thank you for your attention!