Neural Networks for Image Segmentation

This laboratory explores a specific application of machine learning in Materials Science and Engineering (MSE) — image segmentation (in this case, phase identification) of X-ray tomographic data using Convolutional Neural Networks (CNNs). In this lab, students perform by-hand/scripting exercises to explore the foundations of neural networks (NNs), as well as utilize Google Colab and its Graphics Processing Units (GPUs) to train neural networks on real data.

This exercise is adapted from the work of Research Associate Dr. Tiberiu Stan and Prof. Peter Voorhees [STV20] at Northwestern University.

This laboratory was created with the contributions from the following individuals:

- Laboratory design and problem writing: Jonathan Emery and Tiberiu Stan (Northwestern)
- MATCNN initial collaboration development: Elizabeth Holm and Bo Lei (CMU)
- PYTHON development: Joshua Pritz (Northwestern) and Jiwon Yeom (KAIST)
- Adaptation to Google Colab: Nathan Pruyne, Jim James, Marcus Schwarting (Argonne)
Exercise #1: Segmentation by Hand

In lecture we practiced segmentation using the image manipulation program GIMP. Here, you are tasked with creating your own segmentations. The graders will evaluate your efforts for accuracy and precision.

A. Discuss the importance of establishing accurate segmentations and boundaries with regards to studying phase transformations in materials science (such as solidification of metallic materials).

B. In your own words, briefly enumerate the steps necessary to segment the XCT dataset, as discussed in class.

C. Use GIMP to create segmentations for your two assigned images. The instructor will provide these images.

   Keep track of how long it took you to segment each image! Submit your segmentations as PNG files with correctly formatted naming conventions. Make sure you include the images that you were assigned, as well as the segmented label images in the report itself, as well as uploading the .PNGs themselves.

D. How long did it take you to create each of the segmentations? XCT datasets of solidification experiments may contain as many as 30,000 images! How long would segmentation of the entire large dataset take if you did it manually in GIMP?
Exercise #2: Classification with a Neural Network

Here you will build the simple neural network discussed in lecture (Fig. 2) to classify 2×2 pixel images in which each pixel can be either red (+1), blue (-1), or gray (0). You will construct a Matlab/Mathematica/Python script that takes an input vector \( a^1 \) representing a 4-pixel pattern and classifies it as “solid”, “horizontal”, “vertical”, or “diagonal” with some probability.

**Figure 2:** An example neural network for the classification of a 4-pixel image. This network matches the one presented in the lecture.

A. Neural Network Construction

i. Consider a 4-pixel input. Write (in matrix form) the generalized equation that relates the activities in the Input layer \( (a^1) \) to the activities in the Sigmoid Layer \( (a^2) \) via a weighting matrix \( (w^1) \), as shown in Fig. 2. Your Input and Sigmoid layers should not yet have numerical activities, but your \( w^1 \) matrix should contain the numerical values from Fig. 2.

ii. Finish constructing the Sigmoid layer by applying the sigmoid function below to the equation derived in (i.).

\[
\sigma(x) = \frac{2}{1 + \exp(-10x)} - 1,
\]

\( \alpha \) Briefly, explain the purpose of the weighting matrix.

\( \beta \) Briefly, explain the purpose of the sigmoid layer.
iii. Add the second Sigmoid Layer to your network by repeating Parts (i.) and (ii.).
Write the equation for $a^3$ using the weights from Fig. 2.

iv. Add the ReLU layer and define $a^4$. Note, there exist functions in the programs
mentioned above to perform this operation. For example, in Mathematica, the
function is Ramp, in Matlab it is reluLayer. Or, just define your own with an
if statement! Hint: $w^3$ should be a 4-column, 8-row matrix.

α) What does the ReLU layer do? What is the purpose of adding a ReLU
layer?

v. Finally, add a “Normalization” layer to compute probabilities, as discussed in
lecture. Write the matrix equation that connects $a^4$ to $a^5$. Again, use Fig. 2 as
a guide.

B. Use what you completed in (A.) to construct a script that accepts 4-pixel inputs
($a^1$) and outputs probability vectors ($a^5$). This defines the likelihood of this network
to classify the input image as “solid”, “horizontal”, “vertical”, or “diagonal”.

i. Use your neural network to classify the five images in Fig. 3. Report $a^5$ for each
of the images. How well did your network perform? Are you surprised by any
of the results?

![Figure 3: Five 2×2 pixel pixels to be classified](image)

ii. Use a random number generator (values between -1 and 1) to replace all of the
weights in $w^1$ and $w^2$. Try this new neural network on the five 2×2 images
in Fig. 3 and report the values of $a^5$. Are these values different than those
reported in (i)? Which network is more predictive — the supplied network from
part (A.) or the random network?

iii. Typically, we start with random weights when using CNNs for classification.
Explain in your own words how weights are adjusted during network training
to formulate predictive networks.
Exercise #3: Segmentation using MatCNN

In this exercise you will explore the MatCNN tool to segment an x-ray computed tomography (XCT) dataset of a solidification experiment of an Al-Zn alloy. Refer to the lectures, the descriptions in Google Colab, and the paper by Stan et al. [STV20] as resources as you work through these problems. Note that CNN training takes roughly 1 minute per epoch so plan ahead! The MatCNN link is below:

https://colab.research.google.com/drive/1VzV95t9y52ut3IzYwh1ir5S5mQhSBb38?usp=sharing

A. In your own words, define each of the following terms with respect to their usage in neural networks NNs.

- Learning Rate
- Epoch
- Training Dataset
- Validation Dataset
- Test Dataset
- Accuracy
- IoU
- BF1 Score

B. Carefully follow the directions in the MatCNN-Tutorial to train a convolutional neural network with the following parameters:

```
add_version('tomography', model='unet', epoch=10, optimizer='Adam',
            batch_size=1, lr=0.001, aug=False, shuffle=True, patience=11,
            balance=False)
```

i. Once training is finished, the resulting training plot is saved in Google Drive. Submit it with your write-up. You can use the `display_plot` command to visualize the training plot in MatCNN directly.

What do the lines Train Accuracy, Validate Accuracy, Train Loss, and Validate Loss communicate? What are the final Validation Accuracy, Validation Loss, and Training Time?

ii. Evaluate your trained CNN on the two test images. Submit the two overlays and the segmentation metrics generated by the `(display_metric)` command.

Visually compare the raw test images with the overlays. What areas of the microstructure did the CNN struggle to segment correctly?

Now compare the segmentation metrics between the two test images. Which of the CNN segmentations had better scores? Why?

iii. Apply the CNN to all images in the (tomography_full) folder and submit the resulting segmentations. You have now segmented a real dataset! How long did it take to segment each image using the CNN? How long would it take to segment an entire 30,000 image dataset? Compare this to how long it would take you to segment the images by hand (Exercise 2D).
C. Train three networks with the same parameters as in part (B.), but with different learning rates (i.e. learning rates of \( lr = 0.1, 0.001, \) and \( 0.000001 \)).

i. Compare the training plots of the three networks, and submit them. Which CNN achieved the highest validation accuracy? Which CNN converged to a high value the quickest? Why?

ii. Apply each of the three CNNs to the two test images, and submit the resulting overlays and segmentation metrics. Which CNN performed the best?

iii. The quality of the neural network has consequences on data interpretation. In the context of studying phase transformations (like solidification), how might poor segmentations influence scientific interpretation of the data?
References