Assignment 8: Define CNN Architecture

40 Points

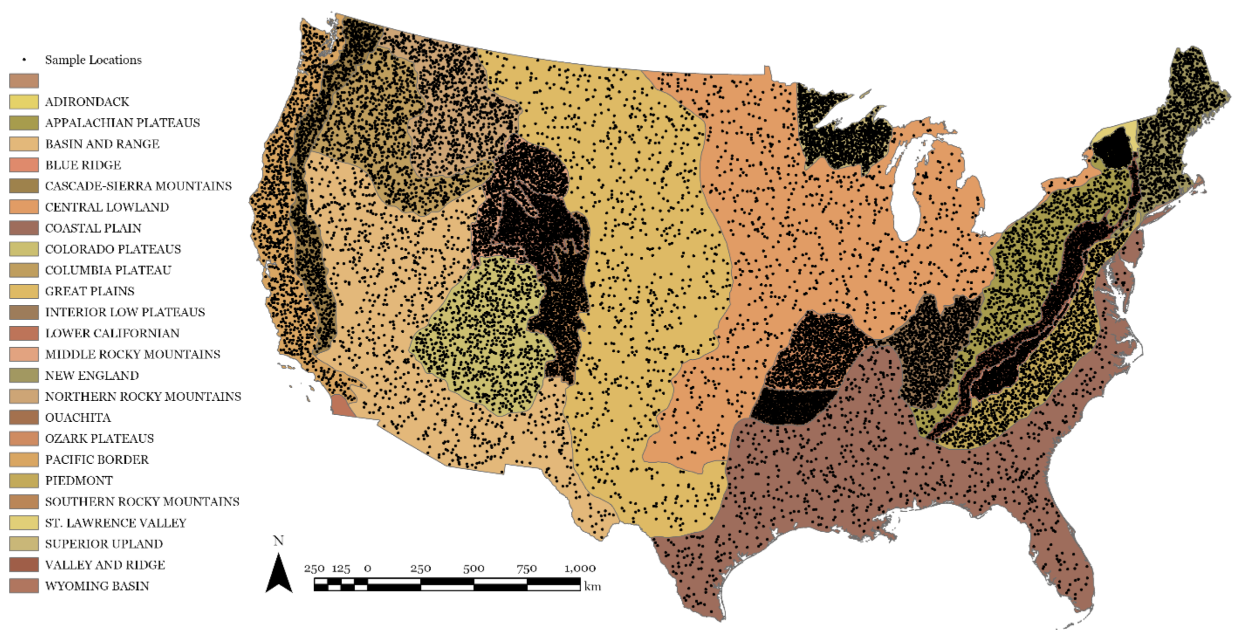
Deliverable: Notebook (.ipynb file) with all required code to complete the stated tasks. Answer all questions in Markdown cells.

Overview: The goal of this assignment is to build a CNN architecture for a scene classification task.

Data:

This assignment makes use of the *physioDL* dataset, which is available on FigShare:

Maxwell, A.E., 2024. physioDL: A dataset for geomorphic deep learning representing a scene classification task (predict physiographic region in which a hillshade occurs). https://doi.org/10.6084/m9.figshare.26363824.



The task presented in this dataset is to predict the physiographic province of an area based on a hillshade image. Terrain data were derived from the 30 m (1 arc-second) 3DEP product across the entirety of CONUS. Each chip has a spatial resolution of 30 m and 256 rows and columns of pixels. As a result, each chip measures 7,680 meters-by-7,680 meters. Two datasets are provided. Chips in the *hs* folder represent a multidirectional hillshade while chips in the *ths* folder represent a tinted multidirectional hillshade. You will use the data in the *hs* folder. Data are represented in 8-bit (0 to 255 scale, integer values). Data are projected to the Web Mercator projection relative to the WGS84 datum. Data were split into training, test, and validation partitions using stratified random sampling by physiographic province. 70% of the samples per region were selected for training, 15% for testing, and 15% for validation. There are a total of 16,325 chips. The following 22 physiographic regions are represented: "ADIRONDACK" , "APPALACHIAN PLATEAUS", "BASIN AND RANGE", "BLUE RIDGE", "CASCADE-SIERRA MOUNTAINS", "CENTRAL LOWLAND", "COASTAL PLAIN", "COLORADO PLATEAUS", "COLUMBIA PLATEAU", "GREAT PLAINS", "INTERIOR LOW PLATEAUS", "MIDDLE ROCKY MOUNTAINS", "NEW ENGLAND", "NORTHERN ROCKY MOUNTAINS", "OUACHITA", "OZARK PLATEAUS", "PACIFIC BORDER", and "PIEDMONT", "SOUTHERN ROCKY MOUNTAINS", "SUPERIOR UPLAND", "VALLEY AND RIDGE", and "WYOMING BASIN".

physioDL.csv: Table listing all image chips and associated physiographic province (id = unique ID for each chip; region = physiographic province; fnameHS = file name of associated chip in *hs* folder; fnameTHS = file name of associated chip in *ths* folder; set = data split (train, test, or validation).

chipCounts.csv: Number of chips in each data partition per physiographic province.

Background Questions

B1: Explain the purpose of the stride parameter when configuring a convolutional layer. (2.5 Points)

B2: Explain the purpose of the padding parameter when configuring a convolutional layer. (2.5 Points)

B3: Why is it necessary to flatten the tensor between the CNN and fully connected components of a CNN architecture for scene classification? (2.5 Points)

B4: Describe another way to decrease the size of tensors in the spatial dimensions other than using pooling layers. (2.5 Points)

Tasks:

T1: Define a CNN architecture by subclassing nn.Module. The architecture should meet the following criteria: (10 Points)

1. Accepts a mini-batch of tensors of shape (mini-batch, channels, width, height) where the channels dimension has a length of 1 and the width and height dimensions have a length of 256.
2. Has a total of 4 convolutional layers that generate 4, 8, 16, and 32 feature maps, respectively. Each convolutional layer should use a 3×3 kernel, a stride of 1, and a padding of 1.
3. Each CNN layer should be followed by a batch normalization layer, a ReLU activation function, and a max pooling operation. All max pooling operations should use a kernel size of 2×2 and a stride of 2.

4. The fully connected component of the architecture should include 3 fully connected layers. The first fully connected layer must accept an input size equal in length to the flattened array produced at the end of the CNN-component of the architecture and generate 128 outputs. The second layer should accept 128 inputs and generate 256 outputs. The last layer should accept 128 inputs and generate 22 outputs, one for each differentiated class.
5. Include batch normalization layers and a ReLU activation function after the first two fully connected layers.
6. The model should output 22 raw logits. Do not rescale the logits using a softmax activation within the model.

T2: Instantiate an instance of the model. (5 Points)

T3: Print a model summary using TorchInfo. List out the number of trainable parameters associated with: (10 Points)

1. Weights in the CNN kernels
2. Biases in the CNN kernels
3. Gain and shift parameters in all batch normalization layers
4. Weights in the fully connected layers
5. Biases in the fully connected layers

How many total trainable parameters are included in the model? What is the estimated file size of the model if saved to disk?

T4: Using your results from the prior module, predict a mini-batch of training data with the model to confirm that it accepts data in the correct shape. (5 Points)