Assignment 12s

Applied Machine Learning

In this assignment, we will continue working on image classification using PyTorch.

- Download the intel image dataset from Kaggle.
- We will use the OpenCV image feature extraction library.
  (`conda install -c conda-forge opencv`)

1. [10 pts] Download the dataset, unzip and explore the file folders. Load the image dataset with training and testing grouped. (Note, `cv2` reads and saves in BGR channel order)

```
import cv2

IMGSIZE = (128, 128)
CNAMES = ['buildings', 'forest', 'glacier', 'mountain', 'sea', 'street']

X_tr, y_tr, X_ts, y_ts = [], [], [], []
for label in _labels:
    path = _path + '/seg_train/seg_train/' + label
    for f in sorted([_ for _ in os.listdir(path) if
_.lower().endswith('.jpg')]):
        X_tr += [cv2.resize(cv2.imread(os.path.join(path,f)), IMGSIZE)]
        y_tr += [CNAMES.index(label)]
```

   Display a few images. How many color channels are there?

2. [30 pts] Convert the imageset to `numpy` array, such as the array size:
   `(14034, 128, 128, 3)`
   Scale the imageset to [0-1].
   Build a regular fully connected neural network and report its performance on this dataset.

3. [40 pts] Create a convolutional neural network (CNN) to train and report its performance on the testing portion of the dataset. 95% reclassification and 75% testing performance should be achievable without any hyperparameter tuning. (Hint: My model, which is similar to the model in module notebook, took around 10 minutes to train 10 epochs without a GPU.)

4. [20 pts] Add regularization and/or drop-out features to your CNN. Report your model's best performance. As the performance standard deviation decreases the model is deemed to be more robust. Why?