CV Homework 3 Report

106062202 陳騰鴻

December 12, 2019

Problem A: Line Fitting

Did Not Have A Basic Understanding of PyTorch

Solution: Mainly this tutorial (link), and also the PyTorch documentation.

- `Dataset`: A dataset container which needs `__getitem__` to be implemented.
- `nn.Module`: The model class from which the model inherits.
- `DataLoader`: A dataloading object that loads data with given batch size.
- `.to(device)`: Specify where data is assigned to: CPU or GPU (for parallel computation).
- Update paramters and gradients are wrapped into `loss.backward()` and `optimizer.step()`.

Plotting Gradient Descent

Solution: Extensive Googling and Stackoverflow.

- `plt.plot_surface`: Requires 3 sets of 2D vectors. `x` and `y` axes can be created by `np.meshgrid`.
- Calculate the losses of all possible weights and biases.
- The set of `z` axis needs to be a function of 2D vectors `x` and `y`, so it will also be 2D.
- Reshape `z` axis back to needed shape using `z.reshape(X.shape)`.

Problem B: License Plate Localization

Loading Data and Converting Them into Tensors

Solution: PyTorch documentation.

- `PIL.Image.open()`: Load images to a `PIL_Image` object of dimensions `(W, H, 3)`.
- `torchvision.transforms`: Contains all kinds of conversion.
  - `transforms.ToTensor()`: Converts a `PIL_Image` object into a tensor of dimensions `(3, W, H)`.
  - `transforms.Compose([...])`: Performs tranformations on an object with the given tranformations.
  - `transforms.Resize()`: Resizes images.
- `torch.from_numpy()`: Create tensor from a numpy array.
- `Path` objects are overloaded with operator `/`, which is analogous to the ones in paths.

Rescaling Pictures Back and Convert Results to Answer Format

Solution: PyTorch discussion forums and Pandas documentation and Stackoverflow.

- `enumerate()`: Packs the original iterators into a tuple and adds a counter in a for loop.
- `torch.tensor()` can be initialized with a list as dimensions.
- SomeTensorObject`.repeat(a, b)`: Repeats the object for a times in row and b times in column.
- SomeTensorObject`.tolist()`: Returns the tensor values as a list (dimensions still kept).
- `pd.DataFrame(list, columns=[])`: Create a `DataFrame` with names specified in `columns`.
- List operations: `[ x ]` creates a list with object `x`. Lists can be concatenated using `+`.