Jaidon Lybbert
EEP596
Homework 3
11/09/2022

Questions and Answers

1. **If I want to design a CNN based image classifier to recognize the following 19 different kinds of dog breeds (great dane harlequin, dalmatian, dobermann bitch, leonberger, French bulldog, german dodge, etc) , as well as their sex (male and female), 8 hair colors and 3 different sizes (small, medium and large), i.e., each image needs to find one class from the above four attributes (breed, sex, color and size), but not all the possible combinations are available (do not design a system with 19x2x8x3=912 output classes), how will you design such a system?**

   My CNN would take each image as an input, and output a vector of length 32 (19+2+8+3=32). The labels for the images would be vectors of this length. The first 19 entries in the label would indicate breed, and one element would have a value of 1., with all other 18 elements equal to 0.

   The next 2 elements of the label vector (or the output of the CNN) would indicate sex, one element of the label would equal 0., and the other 1.

   The next 8 elements would indicate hair color. One element of the label would equal 1., and the other 7 would equal 0.

   The last 3 elements would indicate size. One element of the label would equal 1., and the other 2 would equal 0.

   I would use the cross-entropy as my loss function, and interpret the output of the CNN by slicing the output vector along the category boundaries, and classifying each category by taking the largest value in that category.

2. **Face Verification: Given an AlexNet, pre-trained by 1000-class ImageNet dataset, and the Oxford VGG 2622-identity Celebrity Face Dataset, you want to use this AlexNet to train a face verification system to design a camera-based automatic gate entry system for the 20 students in your lab, where you have collected one photo from each of them. Please describe the training and inference procedures, and the corresponding loss functions, of your designed system**

   For this problem, I am assuming the AlexNet has an output of length 3,622 to classify images as an object class in the ImageNet set or a celebrity in the celebrity dataset.

   I would use the output of the AlexNet network as the input to my neural network. The assumption is that each of the 20 students will trigger a unique identifying output from the AlexNet, based on their similarities to celebrities. I would use several fully connected layers with dropout to connect the 3,622 input to the 20 length output.

To prepare the 20 images, I would apply several data-augmentation methods to simulate different orientations, and lighting conditions, and expand the training set.

I would train the system using cross-entropy loss on the augmented labeled set.

3. **In Ally Complimentary Experts (ACE) method (see Figure below), what is the purpose of introducing the regularization term as shown in the overall loss function. Why not just simply force all the IC outputs to be zero since we are only interested in TC outputs? Without this term, what will be the bad side effect?**

   The regularization term is known as the complement loss. The purpose of the complement loss is to suppress the effect of other experts' responses on the classes they haven't seen. If we set these interfering categories (IC) outputs to zero, they would compete with the target categories during the training, so we instead suppress them with the complement loss. Without the term, we would see more interference between experts'.

4. **The generator network used in the anime face generation, as shown below, takes a vector of 100 random numbers drawn from a uniform distribution as input and outputs an anime face image of size 64 × 64 × 3. The network architecture consists of a fully connected layer reshaped to size 4 × 4 × 1024 and four fractionally-strided convolutional layers to up-sample the image with a 5 × 5 kernel size. A fractionally-strided convolution (known also as 'deconvolution') can be interpreted as expanding the pixels by inserting zeros in between them. Please determine how many trainable parameters are involved in this architecture?**

   The first fully connected layer involves 4x4x1024x100 trainable weights. The first convolutional layer involves 1x1x1024x512 weights, the second convolutional layer involves 5x5x512x256 weights, the third convolutional layer involves 5x5x256x128 weights, and the final convolution layer involves 5x5x128x3 weights. In total that is 6,268,288 trainable parameters.

5. **During the autonomous driving, we have to face various bad weather condition, such as foggy days or raining days. Can you design a de-fogging scheme to systematically remove fog systematically from foggy image frames? What kind of data will you collect and what kind of training procedures will you design?**

   It can't be known for certain what is behind the fog if it is very dense, but it can be predicted via a generative model. I will take driving down a long highway as the simplest case. There will be a large amount of data collected of images driving down similar highways, with consistent markers such as the white line on one side, and yellow stripes in the middle, and asphalt ahead. We can use these images to input to a GAN, and train a discriminator with the labeled clear images. A generator can take the very foggy images and attempt to fill in the blanks.