ASSIGNMENT 1

Computer Vision 20 May 2024

You are free to use any deep learning library. I recommend Keras or PyTorch. For submission, please prepare a condensed and neatly edited report or notebook. Include short descriptions of what you did, results, and brief discussion/interpretation. What you submit must be your own work (not group work!) and sources other than the lecture slides must be cited.

---

1. Training basic CNNs from scratch (40%)

CIFAR-10 is a popular image classification dataset, with 50,000 training samples and 10,000 test samples, and can be loaded from libraries like Keras and PyTorch. Your task is to set up a basic CNN (similar to the one at the end of Lecture 1), train it from scratch on the CIFAR-10 training set, experiment with a few different architectures and regularisation strategies, and ultimately evaluate your final model on the hold-out test set.

Importantly, do not look at your model's classification performance on the test set while experimenting with different architectures and hyperparameters. Instead, use a portion of the training set for validation throughout, and only evaluate your final model on the test set once, for an unbiased indication of how the model would generalise.

2. Classification with transfer learning (30%)

The task is again to train a classification model for CIFAR-10, but this time with transfer learning. Load a model with pre-trained weights into your environment. You may choose any one of the available models (VGG, Inception, ResNet, DenseNet, EfficientNet, etc.), but give a short motivation for your choice. Also find out and explain how the difference in image dimensions between ImageNet and CIFAR-10 can be handled.

Replace the pre-trained model's classification layers with one or two new layers, freeze the weights of the feature extraction (convolutional) layers, and train the new layers. You may also experiment with fine-tuning all the convolutional layers. Compare training times and test accuracies with your best model from Problem 1 above.

3. Input masking (30%)

When a trained CNN classifies an image of some object, can we determine whether it is actually focussing on the image regions containing that object, as opposed to contextual cues surrounding the object? One simple approach is to systematically occlude portions of a correctly classified image, and noting how the output probability of the correct class changes. The result is a "saliency map" of the image, that we can visualise for an indication of what the CNN might be focussing on. This idea has been used by Zeiler and Fergus (2013); see section 4.2 and figure 7d of their paper available here: `https://arxiv.org/abs/1311.2901`.

Load any network with weights pre-trained on ImageNet into your coding environment, and then generate saliency maps for a handful of random images by sliding an occluded square over the image and recording the network's output probability of the correct class at each square's location (similar to the results shown in figure 7d of the paper).

Deadline: Saturday 25 May 2024, 9pm