Marilyn CHAHINE, 21517286
Jean MACQUET, 3532408

TP2-a: Transfer Learning through feature extraction from a CNN

Summary:

In this practical, we created a feature extractor VGG16relu7 by truncating VGG16 to be able to use the output of the last layer before classification as a representation of the input. That allowed us to use the VGG16 model's pretrained weights on ImageNet classification as a backbone for the task we actually want to complete, which is image classification on the smaller 15scenes dataset. The truncated VGG16 model was thus used to generate representations of the 15scenes dataset images, and then an SVM was trained using the VGG16relu7 representations of the 15scenes dataset.

After training and testing the SVM classifier, we try other methods. Fine-tuning the SVM C hyperparameter and applying feature extraction on the first fully connected layer of vgg16 instead of the second yielded better model performances, while replacing the VGG16 classification head with a new one and training it on 15 Scene data yielded slightly worse performance than using an independent SVM.

Questions:

1. Parameter count of a fully connected layer is (input neurons)x(output neurons + 1 for bias):
   - Parameters in fully connected layer 1: (7x7x512)x(4096 + 1) = 102,785,536
   - Parameters in fully connected layer 2: (4096)x(4096 + 1) = 16,781,312
   - Parameters in fully connected layer 3: (4096)x(1000 + 1) = 4,100,096
   - Total fully connected layers parameters = 123,666,944

   The number of parameters in each convolutional layer is $(kxkxC_{in} + 1)xN$ with k the kernel size (k=3 in VGG16), C_{in} the number of input channels, and N the number of filters.

   - Parameters in convolutions block 1:
     - (3x3x3 + 1)x64 + (3x3x64 + 1)x64 = 38,720
   - Parameters in convolutions block 2:
     - (3x3x64 + 1)x128 + (3x3x128 + 1)x128 = 221,440
   - Parameters in convolutions block 3:
     - (3x3x128 + 1)x256 + 2x((3x3x256 + 1)x256) = 1,475,328
   - Parameters in convolutions block 4:
     - (3x3x256 + 1)x512 + 2x((3x3x512 + 1)x512) = 5,899,776
   - Parameters in convolutions block 5:
     - (3x3x512 + 1)x512 + 2x((3x3x512 + 1)x512) = 7,079,424
   - Total convolutional layers parameters: 14,714,688

   ➔ VGG16 has approximately 138,381,632 parameters (~138M)

Marilyn CHAHINE, 21517286
Jean MACQUET, 3532408

2. The output size of the last layer of VGG16 is of size 1,000 and it corresponds to the classification head, each of the 1,000 output neurons corresponds to one ImageNet class and outputs a logit, which can be converted into a probability of belonging to each class using a SoftMax layer.

3.
   a) We normalize the inputted ImageNet images before running them through the model because VGG16 was trained on the normalized ImageNet training dataset, so we normalize the testing data to ensure the input distribution matches the training distribution, which is essential for pretrained weights to behave correctly.
   b) We set the model to eval mode when we are testing (rather than training), because the eval mode disables the dropout layers since we want to evaluate the current weights of the model without removing any of them.

4. Bonus

5. VGG16 is built to output 1000 output classes corresponding to the 1000 classes of the ImageNet dataset. To be able to train VGG16 on the 15 Scene dataset, it is necessary to replace the classification head with another classification layer or another classification model that can be adapted to the 15 Scene dataset. Training the model directly on the 15 scene dataset also makes it possible to run into catastrophic forgetting, where training on new data can change the previous model's weights in a way that makes it unable to classify the images from the dataset it was trained on.

6. Pre-training on ImageNet can help 15 Scene classification because the model will already distinguish the low-level features of the 15 Scene dataset images, since it has already learned to classify them through ImageNet training. For instance, the model will already recognize basic shapes and multiple types of objects.

7. A potential limit for feature extraction could be the risk of ignoring details of the scene that are relevant for classification, it could be a better fit for object classification since scenes often require knowledge of global context.

8. The choice of layer we extract the representation from is important because it determines the level of abstraction included in the representation. Early layers will contain the low-level information of the image such as basic shapes, textures, etc., while the later layers will contain higher level information such as semantic information. We can chose different layers based on the task we are transferring to. For scene classification, relu7 is a good choice because it can already have detected relevant semantic information from the scene that can be useful for scene classification.

Marilyn CHAHINE, 21517286
Jean MACQUET, 3532408

9. We can represent each 15 Scene dataset image by an image that has three input channels (for RGB) but the three channels will all contain the same value for each pixel, and that value will correspond to the value of the black and white channel in the original image.

10. Yes, we can just use the neural network instead of training another independent classifier, by adding a final fully connected layer after the relu7 layer that has input size 4096 and output size 15 (for the 4096 output size of the relu7 and the 15 classes of the 15 Scene dataset). After adding those layers, we fine-tune the model on the 15 Scene dataset since the current weights are pre-trained on ImageNet.

11.

a) Hyperparameter tuning for C of the SVM:

The default SVM setting is C=1, which results in a hard margin SVM. It can be useful to finetune C to check if allowing some misclassifications improves the accuracy of the model.

We tested SVM with C = [0.001, 0.01, 0.1, 0.5, 1]. The optimal value for C was 0.001 with an accuracy of 88.9%. But the results for all the tested parameter values were between 88.5% and 89%, so we achieve similar performance using hard or soft margin SVM.

b) Replacing the SVM by a fully-connected layer for classification:

We replace the SVM by an extension of the VGG16relu7 model that adds a (4096, 15) layer for classification. This method simplifies the pipeline by relying on a single neural network instead of a feature extractor and classifier. But it achieved slightly lower performance compared to the SVM-based approach, with an accuracy of 85.5% after 10 epochs.

This can be explained by the limited size of the 15 Scene dataset, which increases the risk of overfitting when training additional network parameters. We have also only trained one layer on the classification task, with all other layers' parameters being frozen, which might not have been enough depth for the accuracy to be as high as for the SVM-based classifier.

c) Using the first fully connected layer as feature extractor instead of the second one:

This evaluates the impact of the depth of the feature extractor on the results. We used the output of the first fully connected layer fc6/relu6 instead of the second one fc7/relu7. Features extracted from fc6 are less specialized and capture more generic visual information, while features from fc7 are more abstract and semantically rich.

We observed a slightly higher accuracy when using relu6 instead of relu7, with an accuracy of 90.15%.

This could be due to low-level features having more common ground across the two datasets. The ability to recognize low-level information of the image might be more useful for scene categorization than being able to recognize semantics that aren't necessarily present in scenes. Low-level visual information is more generalizable across different image domains.