Master in Computer Vision Barcelona

UAB UOC UPC upf.

Module: **M3. Machine learning for computer vision** **Final exam**

- Books, lecture notes, calculators, phones, etc. are not allowed.
- All sheets of paper should have your name.
- Answer each problem in a separate sheet of paper.
- All results should be demonstrated or justified.

Question 1:

The SIFT feature extractor obtains rotation invariance by...

a) normalizing the vector resulting from the concatenated histograms to have unit length.
b) constructing the histograms of 4×4 pixel subregions centered at each key point.
c) sorting the histogram components with respect to the dominant orientation.
d) weighting the orientations by the magnitude of the gradients.

Answer:
c). Although all the responses are present in the SIFT algorithm, only the c) is related to the rotation invariance property. Normalizing vectors to have unit variance improves the descriptor against illumination changes, and b) and d) are effective steps on the algorithm, but do not help to get rotation invariance, as the final histograms would change after a simple image rotation.

Question 2:

In the Principal Component Analysis algorithm...

a) we do not use the labels that show the class membership of the samples.
b) we sort the eigenvectors of the covariance matrix, and use the ones that have minimum eigenvalue as the projection matrix.
c) the algorithm finds the linear projection that bests separates the input samples in two classes.
d) we find the linear projection that minimizes the variance in the projected space.

Answer:
a). Answer b) is incorrect, we take the eigenvectors with larger eigenvalue, d) is also incorrect, the projection maximizes the variance on the projected space, and c) is also incorrect, as we do not take into account class membership in PCA, which is exactly what it is stated in a).

Question 3:

Describe briefly the main differences between hard and soft assignment in the BoW frameworks. Which is the main motivation for using soft assignment instead of hard assignment?

Answer:
In hard assignment every local descriptor is assigned only to one of the visual words, while in soft assignment every local descriptor is assigned to all visual words with a relative weight that depends on the distance of the descriptor to the word representative. In hard assignment two very close descriptors could be assigned to different visual words or two very different descriptors assigned to the same visual word. In soft assignment we obtain a more uniform distribution of descriptors to visual words and thus, we can obtain a final representation more robust to noise.

Question 4:

Describe briefly how we compute the final image representation when we use the standard configuration of Spatial Pyramids approach in the BoW framework.

Answer:
The representation is computed at different levels of an image pyramid. At the first level, the standard BoW histogram is obtained from the whole image. At the second level, the image is partitioned using a regular grid with four cells. For each of these cells, also a standard BoW histogram is obtained. At the third level, every cell at the second level is again partitioned using a grid of four cells, and a standard BoW histogram is obtained for each of the resulting 16 cells. We can repeat the process for as many levels as necessary. Finally, all the BoW histograms obtained for all cells at all levels are concatenated into the final representation. All histograms are weighted with a weight that is larger for histograms derived at finer levels of the representation.

Question 5:

In a SVM, the use of the slack variables has the following implications in the solution of the optimization problem:

a) The reduction of the dimensionality of the feature space.
b) The increment of the dimensionality of the feature space.
c) The elimination of the maximal margin constraint.
d) The introduction of a regularization parameter.

Answer:
d) The regularisation parameter C allows some misclassifications in the final result.

Question 6:

The kernel trick allows

a) The generalization of the SVM to non-separable datasets.
b) The inclusion of some misclassifications in the final result.
c) The implementation of the SVM in a linear time with respect to the number of dimensions.
d) To get the Bayesian error.

Answer:
a) By using the kernel trick, the dual problem is solved in a high dimensional space where the dataset can be separated linearly.

Question 7:

The bagging approach implies

a) Bootstrap sampling with substitution.
b) Random sampling with substitution.
c) Bootstrap sampling without substitution.
d) Montecarlo sampling without substitution.

Answer:
a) Bootstrap sampling with substitution.

Question 8:

The classical classifier of a boosting algorithm must be...

a) A strong classifier with a performance slightly over 50%.
b) A strong classifier with the maximum performance achievable.
c) A weak classifier with a performance slightly over 50%.
d) A weak classifier with the maximum performance achievable.

Answer:
c) A weak classifier with a performance slightly over 50%.

Question 9:

For any given classification problem, the minimum error achievable is always…

a) 0.
b) The training error.
c) The Bayesian error.
d) The test error.

Answer:
c) The Bayesian error.

Question 10:

The p-value

a) Must be typically more than 0.50 to be associated to statistical significance.
b) Must be typically less than 0.50 to be associated to statistical significance.
c) Must be typically more than 0.95 to be associated to statistical significance.
d) Must be typically less than 0.05 to be associated to statistical significance.

Answer:
d) Must be typically less than 0.05 to be associated to statistical significance.

Question 10:

Explain the advantages of convolutional neural networks over feed forward or multilayer percentrons that justify their wide adoption for computer vision applications.

Answer: slides 68-70. Problems with NN: –Massive number of parameters –Fixed input size –Each point is connected to every other point

Question 11:

Name as much layer models as possible used in a typical convolutional net.

Answer: Input layers, convolutional layers, activation functions (RELU, …), pooling layers, softmax layer, loss function (Categorical cross entropy, …), regularization (Dropout, l2...), Normalization, ...

Question 12:

What is overfitting? How can you prevent it?

Question 13:

Suppose you are building a convolutional network. How would you best initialize the weights of the convolutional layers? Best mean better than simply a constant or random values.

Answer: slide 63

Question 14:

What is an activation function? Tell me some of them

Answer: Activation definition. Sigmoid, Tanh, RELU, Leaky relu, ...

Question 15:

Historically one of the first activation functions was the sigmoid. Because of some problems, now ReLU and variants are preferred. Explain at least two reasons for that.

Answer:
Slides block 2 39-53. Properties: 1.Does not saturate 2.Fast Convergence 3.Computationally efficient Problems: 1.Non-zero entered 2.Gradient of negative values?

Question 16:

The typical parameters of a convolutional layer are the kernel shape, the stride and the padding. Given the following Keras code implementing the first layers of a SegNet network, calculate a) the shape of the output of each layer ('maxpX_enc') and b) the size of its receptive field of view. All of them in 3 dimension.

```
kernel = 3
pool_size = 2
depths = np.array([64, 128, 256, 512], dtype=np.int)
do = 'tf' # tensorflow
input_shape = (360, 240, 3) # rows, columns, channels, ordering is 'tf'

inputs = Input(input_shape)

""" encoding layers """
conv1_enc = Convolution2D(depths[0], kernel, kernel, border_mode='same',
                dim_ordering=do, trainable=True)(inputs)
bn1_enc = BatchNormalization()(conv1_enc)
act1_enc = Activation('relu')(bn1_enc)
maxp1_enc = MaxPooling2D(pool_size=(pool_size, pool_size), strides=(2, 2),
               dim_ordering=do)(act1_enc)

conv2_enc = Convolution2D(depths[1], kernel, kernel, border_mode='same',
                dim_ordering=do, trainable=True)( maxp1_enc)
bn2_enc = BatchNormalization()(conv2_enc)
act2_enc = Activation('relu')(bn2_enc)
maxp2_enc = MaxPooling2D(pool_size=(pool_size, pool_size), strides=(2, 2),
               dim_ordering=do)(act2_enc)

zp3_enc = ZeroPadding2D(padding=(pad,pad), dim_ordering=do)(maxp2_enc)
conv3_enc = Convolution2D(depths[2], kernel, kernel, border_mode='same',
                dim_ordering=do, trainable=True)( maxp2_enc)
bn3_enc = BatchNormalization()(conv3_enc)
act3_enc = Activation('relu')(bn3_enc)
maxp3_enc = MaxPooling2D(pool_size=(pool_size, pool_size), strides=(2, 2),
               dim_ordering=do)(act3_enc)
```

Question 17:

What is the idea of grid search for hyper-parameters? and random search? Why is random search often much more efficient?

Answer: No closed answer

Question 18:

Name at least 5 hyper-parameters and link it to the task it is involved in (ie: weight initialization, SGD minimization, …)

Answer: No closed answer

Question 19:

Understanding and Visualizing CNNs can be done through different points of analysis:

1. Weight visualizations
2. Visualizing layer activations
3. Analyzing a set of Image Patches that maximally activate a neuron
4. Generating artificial images that maximize the activation of a neuron
5. Describing their neurons using selectivity indexes.

Choose two of these methodologies. Compare them by explaining what are they trying to describe from a trained CNNs (main idea), how they are able to describe a neuron (used methodology), the strong and weak points, ...

Answer: No closed answer

Question 20:

You have seen different methods that try to describe, understand and visualize the main features learned by a trained CNN. What can we conclude from all of the works that are focused on understanding the neuron behavior? Which kind of features are learned in each level of the hierarchy?

Answer: No closed answer