Exam in SSY097

August 19th, 2019

Allowed materials : Pen/pencil, eraser.

The exam consists of six problems. Make sure that you have them all.

- Motivate all answers carefully.
- Use a new paper for each new numbered problem.
- Write on one side of the papers only.
- Write your anonymous number on each new page.
- Avoid using a red pen.
- If you want the result registered as SSY096, write this on the cover page.

The dates for the exam review will be announced on PingPong.

Grades

$\geq$ **8 points** Grade: 3

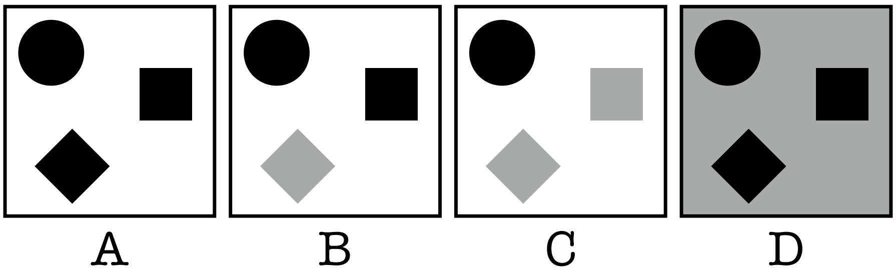
$\geq$ **11 points** Grade: 4

$\geq$ **14 points** Grade: 5

1 SIFT, 3 points

(a) Given a patch centered around a keypoint, SIFT computes a descriptor from that patch that can be used for image matching.
Describe how SIFT computes these descriptors: Name the key steps in the descriptor computation process and describe each in 1-2 sentences.

(b) Which of the following patches will result in the same SIFT descriptor? Justify your answer.

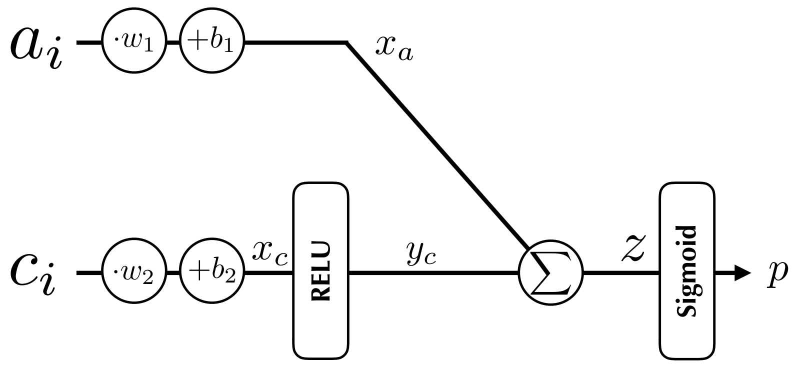


(c) Prior to descriptor computation, SIFT aligns the patch from which the descriptor is extracted to the orientation assigned to the keypoint.
Describe in 2-3 sentences how the orientation of the keypoint is determined.

2 Statistical Learning, 3 points

Consider the neural network shown below that performs binary classification on a tuple (a_i, c_i) of two scalar input values $a_i, c_i \in \mathbb{R}$. The RELU and Sigmoid functions are given as $\text{RELU}(x) = \max(0, x)$ and $\text{Sigmoid}(z) = \frac{e^z}{1+e^z}$. The derivative of the RELU function is given as

$$\frac{\partial \text{RELU}(x)}{\partial x} = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{otherwise} \end{cases} .$$



(a) Provide the loss functions L_i for when the example (a_i, c_i) is positive and when it is negative. The loss should depend on the probability p estimated by the Sigmoid layer. As in the lecture and the lab, use the negative log-likelihood loss.

(b) Given a positive example $(a_i, c_i) = (1, 2)$, compute the derivatives

$$\frac{\partial L_i}{\partial w_1}, \quad \frac{\partial L_i}{\partial b_1}, \quad \frac{\partial L_i}{\partial w_2}, \quad \frac{\partial L_i}{\partial b_2}$$

through the backpropagation algorithm. To this end, first perform a forward pass to compute values for x_a, x_c, y_c, z, and p. Next, use the chain rule to derive formulas for the derivatives in the backward pass. Compute the actual values for the derivatives using your equations and the values for x_a, x_c, y_c, z, and p computed during the forward pass.
The current values for w_1, b_1, w_2, and b_2 are

$$w_1 = -15, \qquad b_1 = 5, \qquad w_2 = 4, \qquad b_2 = 2 \ .$$

(c) Using a learning rate of 0.1, what is the next value for w_2?

3 (Robust) Model Fitting, 3 points

(a) A 2D affine transformation maps a 2D point $p_1 = (x_1, y_1)$ to a 2D point $p_2 = (x_2, y_2)$ via

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \quad . \tag{1}$$

Explain how to construct a solver that can be used inside RANSAC to estimate a transformation between two sets of 2D points. To this end, derive a linear system $\mathtt{M}\theta = \mathbf{v}$, where the vector θ contains the parameters of the transformation.
It is sufficient to derive the equations for a single correspondences between p_1 and p_2.

How many correspondences are needed for your minimal solver?

(b) How does Eq. 1 change when estimating a homography rather than an affine transformation?
Make sure to explain all variables.

(c) Methods for least-squares model fitting implicitly assume that the noise on the data points follows a Gaussian / Normal distribution. Thus, they are not able to handle outlier measurements as these do not follow a Gaussian / Normal distribution.
Explain in 2-3 sentences how robust cost functions address this problem.

4 RANSAC, 3 points

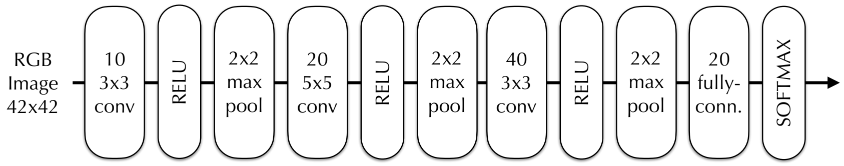
In the following, we will consider the problem of fitting a 2D line through a set of 2D points in the presence of outliers. We will use a minimal solver that fits a line through two points.

(a) Given an inlier ratio of ε, derive a formula for the number of RANSAC iterations k required such that the probability of missing the best model is at most 1%.
Explain how you obtain your formula.

(b) Implement RANSAC for 2D line fitting in pseudo code or MATLAB code (or a mixture of the two). You can assume that you have a function `line = minimal_solver(S)` that computes a line hypothesis from a minimal sample `S` and a function `squared_error = evaluate(line, c)` that returns the squared error when evaluating a line hypotheses `line` on a 2D point correspondences `c`.
Make sure that you explain all variables that you use.

(c) In a general context (not necessarily for line fitting), assume that a minimal solver requires only a single data point as input.
How many RANSAC iterations are necessary to ensure that RANSAC finds the best solution with probability 100%? Justify your answer!

5 Deep Learning, 3 points



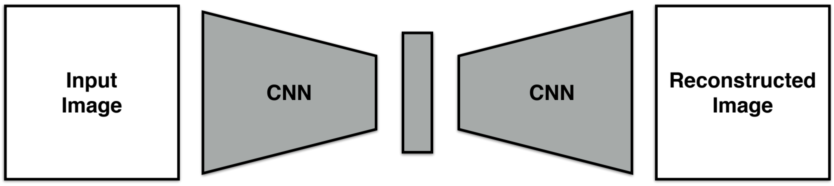
(a) Consider the convolutional neural network shown above. For each layer of the network with trainable parameters, write down the number of trainable parameters.
Assume that the convolutional layers do not use padding and that no bias terms are used.

(b) Assume that you replace the 20 5×5 conv. block in the network above by a 20 7×7 conv. block. What is the number of trainable parameters in the fully connected layer?

(c) Assume that you replace the fully connected layer in the network from a) by a fully convolutional layer to be able to apply the network on RGB input images of arbitrary dimensions. You do the replacement such that applying the new network without padding on a 42×42 RGB image results in the same output as applying the original network.
How many trainable parameters does the new fully convolutional layer have? Explain your answer!

6 Generative Networks, 3 points

(a) The general structure of a Variational Autoencoder is given in the figure below.



Name the different parts (marked in gray) and explain their purpose (a single sentence per component is sufficient).
Which parts are required during training? Which parts are used during testing?

(b) The training objective of a generative model is given by

$$\min_G \max_D E_{\mathbf{x}\sim p_{\text{data}}}[\log(D(\mathbf{x})] + E_{\mathbf{z}\sim p_{\text{model}}}[\log(1 - D(G(\mathbf{z})))] \quad . \tag{2}$$

Is Eq. 2 the training objective of a VAE or a Generative Adversarial Network (GAN)?
Explain all terms in Eq. 2.

(c) Assume that you are given a GAN that can generate images of dogs and cats. How can this GAN be used to help train a convolutional neural network that classifies images into cats and dogs?
Do you expect your approach to solve the problem of obtaining large training sets of labelled data in general? Explain your answer!