POLI
[TECH)
NIKA
Czestochowa
University
of Technology

Faculty of Mechanical Engineering
and Computer Science

Computer vision, pattern recognition and image retrieval

Laboratory 10

Topic: *Convolutional Neural Networks (CNN) - Understanding Results*

Teacher: Joanna Kulawik, PhD

Technical support for Matlab is available on the website: http://www.mathworks.com/

Exercise 1

In Lab 09, you gained insight into a basic CNN model. Now, let's implement a new CNN model architecture in the 'Lab10.m' script. The goal is to classify digits 2, 4, 6 and 8. Prepare a dataset from the MNIST set (from the last laboratory) using the 'imageDatastore' as previously learned. Ensure your model comprises at least 3 convolutional layers, and adhere to the fundamental rules for creating layers (choose filter sizes appropriate to the image or data from the preceding layer).

As in previous classes, conduct the learning and testing processes. To assess the model's performance, calculate relevant metrics using the confusion matrix. Compute accuracy for each set, along with specificity, recall, precision, F1-score, and MFC metric.

Additionally, to evaluate the model's effectiveness, randomly select and display 12 images from the test set, showing the obtained prediction results.

Required commands:

confusionmat – Compute confusion matrix for classification problem

confusionchart – Create confusion matrix chart for classification problem

In response, please send:

- the file 'Lab10.m',

- a screenshot of the graphical presentation of the confusion matrix,

- a screenshot of the test results generated for 12 images.

Additional materials

You can find information about the basic layers on CNN at the link https://uk.mathworks.com/help/deeplearning/ug/layers-of-a-convolutional-neural-network.html

Information about the available types of layers in the Matlab environment can be found at the link https://uk.mathworks.com/help/deeplearning/ug/list-of-deep-learning-layers.html

In classifiers, besides computing accuracy, it is common to utilize metrics derived from a confusion matrix, wherein:

TP - is the number of true positive results,

FP - is the number of false positives,

TN - is the number of true negative results,

FN - is the number of false negatives

		Predicted	
		Positive	**Negative**
Real	**Positive**	**TP**	**FN**
	Negative	**FP**	**TN**

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + FP + TN + FN}}$$

Recall

It is a metric that quantifies how frequently a sample is accurately classified in relation to true positive responses. It assumes values in the range ⟨0, 1⟩: the higher its value, the higher the sensitivity of the tested network model.

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}}$$

Specificity

It is a metric that measures the model's ability to minimize false positives. It assumes values in the range ⟨0, 1⟩: the higher the value, the more specific the model is.

$$\boldsymbol{Specificity} = \frac{\boldsymbol{TN}}{\boldsymbol{TN} + \boldsymbol{FP}}$$

Precision

It is a metric that assesses the accuracy of predicted positive responses, indicating the proportion of correct positive predictions (positive predictive value). It takes values in the range ⟨0, 1⟩, where a higher value reflects better precision achieved by the network model.

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}}$$

F1-score

It is the harmonic mean of precision and recall, generally serving as a model quality indicator. In binary classification, it determines the accuracy of the model but is most commonly used to assess the quality of information retrieval models. It takes values in the range ⟨0, 1⟩, with the indicator being better the closer its value is to 1.

$$\boldsymbol{F1} = \frac{\boldsymbol{2 \cdot TP}}{\boldsymbol{2 \cdot TP + FP + FN}}$$

Matthews Correlation Coefficient (MCC)

Is a metric that simultaneously analyzes data from all sets of confusion matrices (true positives, true negatives, false positives, and false negatives), similar to accuracy but with a crucial feature: it is insensitive to significant class imbalance. It takes values in the range ⟨-1, +1⟩, where -1 indicates complete disagreement in predictions, and +1 signifies perfect prediction.

$$\boldsymbol{MCC} = \frac{\boldsymbol{TP \cdot TN - FP \cdot FN}}{\sqrt{\boldsymbol{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}}$$