University of Helsinki
Data Science for the Internet of Things (DS4IoT) - Spring 2024

Exercise 6

Due on 29th April 2024 by 23:59, Helsinki time.

Instruction: All course participants are requested to submit their solutions (in English) through Moodle by the due date. Please take into account the following guidelines about including AI-generated text: https://studies.helsinki.fi/instructions/article/using-ai-support-learning

Submission: You can submit your homework using one of the two following options:

- *Option 1:* submit both the report of non-programming tasks and the solution of the programming tasks in Jupyter Notebook format (*.ipynb*).
- *Option 2:*
  - Submit the report of non-programming tasks in PDF file format.
  - Submit the solutions and the source code of the programming tasks in Python using Jupyter Notebook format (*.ipynb*).

Use the following format for naming the files: *[last name_first name]_[your file name], (i.e., Nguyen_Ngoc_Exercise 1 solution.ipynb).*

Assessment: Participants are encouraged to review course materials to answer the problems and in some cases write computer programs to derive solutions. In all the exercises, do not just give the answers, but also justify, derive and contextualise the answers in Data Science for the Internet of Things contexts. The assessment parameters for each of the tasks are: solution (25%), justification and derivation (50%), contextualisation (25%).

1 Compulsory Tasks

1.1 Learning diary (4 pts.)

Choose two concepts from this week's lectures (one from Lecture 11 and one from Lecture 12) and describe **what** they mean and **why** they are relevant for the Data Science for the Internet of Things. For full points give an example (or examples) of the concept in a Data Science for the Internet of Things context or otherwise illustrate their importance. Do not copy the definitions from Wikipedia or other sources, but explain the concepts in your own words and relate them with Data Science for the Internet of Things. Use about 100–150 words to describe each of the concepts.

2 Feedback survey (2 pts.)

Fill out the feedback survey found in the material of the last week in Moodle.

3 Non-programming Tasks

3.1 Learning in Deep Learning (4 pts.)

Consider the network shown in Figure 1 and the activation functions given in the figure (σ = sigmoid). Assume the initial weights are $(0.5094, -0.4479, 0.3594)$. As the loss function consider binary cross-entropy, i.e., assume the network is responsible for a binary classification task where the output corresponds to the probability of the positive / negative outcome. Assume you are given data $(-2, 4, 2)$ belonging to the positive class (i.e., $y = 1$ below). Answer the following questions about the network:

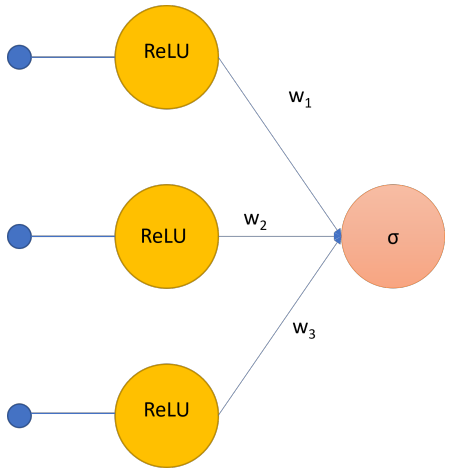

Figure 1: Deep learning model

(a) What is the output of the network?

(b) What is the loss of the network for the given sample?

(c) What is the value of the gradient of the loss?

(d) Assume the learning rate $\eta = 0.01$. Calculate updated weights for the network given the loss.

Use the binary cross-entropy as the loss function:

$$L\left(f(x,w),y\right) = -y\log\left(f(x,w)\right) + (1-y)\log\left(1-f(x,w)\right) \tag{1}$$

Here $f(x,w)$ is the value of the activation function for a given input x and weight vector w and y is the ground truth value.

The gradient of the Loss function is given by:

$$\Delta L = (f(x, w) - y)\, x \tag{2}$$

3.2 Deep Learning Architecture and Frameworks (4 pts.)

Consider the DS4IoT applications described below. Which type of deep learning network(s) and TensorFlow version(s) would be best suited for the applications and why?

(a) Sun flare monitoring using x-rays and optical light sensors.

(b) Mushroom classification using smartphone camera.

(c) Satellite images for crowd density estimation.

(d) Accelorometer, gyroscope, electrodermal activity, oxygen saturation and skin temperature data from wearables for emotion recognition and heart rate prediction.

4 Programming Tasks

4.1 Deep Learning CNN Programming (4 pts.)

The solar flare prediction image dataset is an open access dataset that contains solar images taken by the Solar Dynamics Observatory (SDO) satellite and are used to detect and predict solar flares. The file *solar_flares.zip* in Moodle contains 1074 images taken from the SDO dataset. The pictures are organised in two folders that represent the occurrence of solar flares (i.e., $flare$, $noflare$).

(a) Use TensorFlow to implement a deep learning model that classifies the images for occurrence of solar flares. Figure 2 describes the architecture of the model. Use 80/20 hold-out to split the dataset into training and validation subsets. Report the model summary and plot the model architecture. (**Hint:** You can find a full example of how to implement CNNs in TensorFlow in the following link: *https://www.tensorflow.org/tutorials/images/cnn*. Use *tf.keras.utils.image_dataset_from_directory* to split the dataset automatically).

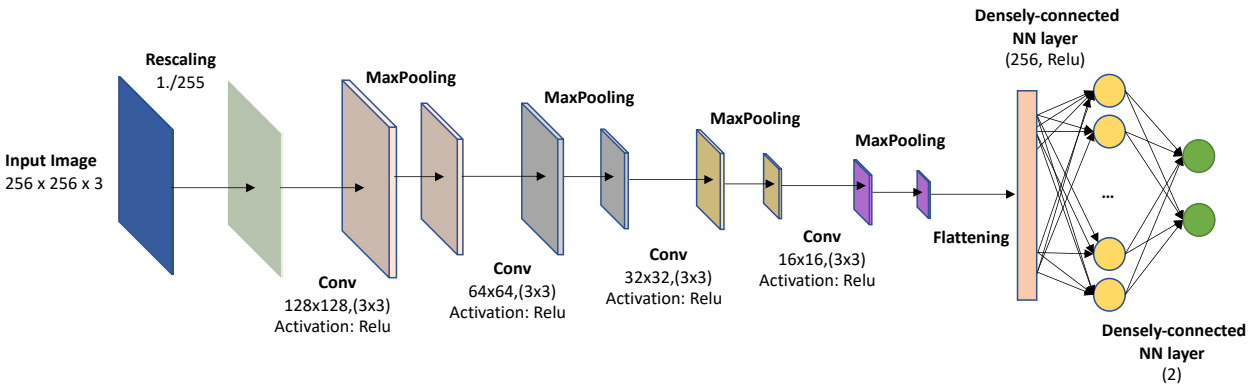

Figure 2: Deep learning model architecture

(b) How many epochs are necessary to obtain an accuracy higher than 90%? Report the epoch-accuracy plot to get full points. (**Hint:** Use *adam* optimizer and $tf.keras.losses.SparseCategoricalCrossentropy$ loss when compiling the model. The output of the method fit provides detailed information of the changes in the model performance during the training and validation).

(c) Data augmentation can reduce the overfitting caused by having a small number of samples. The experts warn that excepting vertical flip, other data augmentation methods will alter the underlying physics. Add a data augmentation layer into the model that satisfies this warning. Compare the model performance before and after adding data augmentation. How about adding a dropout layer (rate = 0.5) before flattening instead of using an augmentation layer? (**Hint:** Use $layers.RandomFlip$ to perform data augmentation). *Note:* Use 20 epochs to train the model.

(d) Use Sigmoid instead of Relu as the activation function. How does this change impact the results? Why? Use the model architecture described in Figure 2.

* **Note:** In all the cases use 20 epochs to train the model.