horizontal line

**Apella Technology, Inc.**

ML Engineering Assessment

**Revision: October 15, 2020**

Time 120 minutes

PROJECT DESCRIPTION

* We are acquiring a time-series of 2D images of a hospital scene, and are looking to infer, in real-time, what is happening in this scene.
* To simulate this challenge, your project is as follows:
  + You hypothesize that there are a number of key objects in the scene - a circle and a square -, which can indicate to us in what “phase” the system is, i.e. what is happening in the scene.
  + Using deep learning, you will train an algorithm to detect the presence of key objects, as they move in and out of the scene (*Task 1*).
  + Once you have reached sufficient accuracy on the object detection task, you would process the obtained results - presence and absence of objects - to train a machine learning model to infer in what phase the system is.
  + In *Task 2*, you are provided with some hypothetical results and will assess what the next steps in algorithm training might be.

# TASK 1 (WITH CODE)

## Data Description

* Several (5) sequences of noisy 150 gray-scale images (128 x 100), with two objects coming in and out:
  + a bright circle of radius 15 pixels.
  + a dim square of width 15 pixels.
* Phases of the image sequence:
  + Phase 1: the circle is present alone before the square shows up (Figure 1A)
  + Event 1: the square enters the field-of-view (Figure 1B)
  + Phase 2: both objects present (Figure 1C)
  + Event 2: the square leaves the field-of-view (Figure 1D)
  + Phase 3: the circle is present for some duration after the square has left
  + Event 3: the circle leaves the field-of-view (Figure 1E)
  + Phase 4: both objects absent (Figure 1F)

Figure 1: Sample Images

|  |  |  |
| --- | --- | --- |
| A | B | C |
| D | E | F |

Notice how Phase 1 and Phase 3 both are both characterized by presence of the circle and absence of the circle. However, the order in which objects appear and disappear in a scene is also critical in determining the phase.

## Task Setup

* We have provided some initial code to load the 5 sequences of data, split them into train/validate/test sets, build a simple deep learning architecture and train on the data.
* To access the code and data, unzip the file into a location of your choice
* Install the required python modules on your local machine and/or environment
  + pandas, numpy, tensorflow (2.0 or above), opencv-python are the only required packages, along with python > 3.6.
  + Data size and complexity of the algorithm is low, so you should be able to train rapidly on the CPU (~30-60 seconds per training run)
* Run python train.py at the command line to verify that the training procedure runs - even though results will be sub-optimal.
* Describe any issues you encounter, if any, and how those were fixed.
* **You will be asked to write some code and submit it as part of this exercise. Add any comments within the code if needed, or provide a separate README file as needed.**

Questions

1. Briefly read through the code that builds the tensorflow/keras model, and describe at a high level the neural network architecture you see defined. (Maximum 100 words)
2. What is the receptive field of the first conv-layer in the default architecture?
3. What is the receptive field of the second conv-layer in the default architecture?
4. What is a reasonable range of filters (default 16, see line 203 of train.py) for the first conv-layer? Why? (Maximum 50 words)
5. Why are we using sigmoid activation instead of soft-max in the last dense layer? (Maximum 50 words)
6. How would you propose to augment the training data / images? (Maximum 50 words)
7. Run the model as it is with default parameters, by running python train.py
8. What can you say about the performance (train vs validate vs final test) as reported by the accuracy metric? (Maximum 100 words)
9. Adjust the command-line parameters of the model (python train.py --help to see what they are) to improve the results. Please provide a table of parameters you’ve tested, and the results you obtained. Explain the values and parameters you chose (based on empirical results or theoretical understanding). (Maximum 200 words)
10. Accuracy can be, at times, a misleading metric. What would you propose as a performance metric other than accuracy to circumvent some of its pitfalls? (Maximum 100 words)
11. Modify the network code itself to improve the results if you think further improvements are possible. If you do so, explain and justify your choices. (Maximum 200 words). As previously, please include a table of parameters and structures you’ve tested and the associated results. This modified version of the code will be submitted as train\_v1.py.
12. Create an additional copy of train.py called train\_v2.py. In this version, modify the command-line argument parser to add one parameter for the choice of optimizer, either SGD or Adam. Adjust the remainder of the code to integrate this option in the training routine.
13. Create one more copy of train.py called train\_v3.py. In this version, you will be making specific changes to the network architecture:

* Insert a dropout layer after each dense layer of the network, with a dropout probability of 20%.
* Make the 3rd convolutional block a residual block, similar to what is found in the ResNet architecture.

# TASK 2

## Task Setup

Assume Task 1 results in a model with sufficient accuracy to detect the presence of key objects (circle and square as defined earlier). During real-time inference of the model on real data, you obtain raw object detection model results such as this, which indicate the confidence in the presence of each object:

|  |  |  |
| --- | --- | --- |
| Frame / time index | Probability of circle present | Probability of square present |
| 0 | 0.89 | 0.11 |
| 1 | 0.91 | 0.32 |
| 2 | 0.88 | 0.79 |
| 3 | 0.8 | 0.88 |
| 4 | 0.81 | 0.89 |
| 5 | 0.90 | 0.2 |
| 6 | 0.95 | 0.05 |
| 7 | 0.94 | 0.01 |
| 8 | 0.87 | 0.10 |
| … | … | … |

The aim is now to train a real-time “phase model” that can use this data to infer when a transition between various phases happens.

## Questions

1. Can this tabulated data (excluding the first column) be used as a set of training feature vectors for a traditional ML model such as a logistic regression, Random Forest or support vector machine to assess the phase of the image? In other words, do we have sufficient information in those 2 columns? Explain your answer (Maximum 150 words)
2. Assume now that you use this raw data, along with 20 additional hand-crafted columns, to train a Random Forest model. You get the following results:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model # | Number of decision trees | Training accuracy | Validation accuracy | Test accuracy |
| 1 | 10 | 75% | 72% | 60% |
| 2 | 30 | 80% | 77% | 73% |
| 3 | 50 | 85% | 78% | 77% |
| 4 | 100 | 88% | 79% | 76% |
| 5 | 200 | 89% | 79% | 77% |

3. Which model would you choose and why? (Maximum 150 words)

4. What kind of hand-crafted columns or processing steps would you envisage as input to your phase model? (Maximum 200 words)

5. How do you explain the difference in performance accuracy between the validation and the test set? If you can’t explain it, what are some ideas of how you might address that question? (Maximum 200 words)

6. If this image data capture processing, and phase change processing, were happening in real-time with real people doing real things in a complex environment, how would you adjust some of your architectures in Task 1 and Task 2? (Maximum 300 words)