**Lab Assignment #4 – Use CNNs in image classification problems**

Due Date: By the dropbox deadline.

Purpose: The purpose of this Lab assignment is to:

* Build CNN models for solving image recognition problems
* Design and implement Python applications that incorporate CNN solutions
* Use TensorFlow for implementing CNN models and evaluate their accuracy

References: Read reference books, Stanford tutorial on logistic regression (http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/), and the lecture slides. This material provides the necessary information that you need to complete the exercises.

* Your submission **MUST** include the **dataset files (if any)** that you may have used. Instead of using a dataset file, **if you need call some function to fetch a dataset**, then that should be fine too.
* Your submission **must** be a **zip** file **if compressed**. Any other compression **is not accepted**.
* You MUST run the program of an exercise in the relevant .ipynb file and **MUST retain** **the output that gets generated** (Note that the textual output of a program in an .ipynb file stays in that .ipynb file if you do not delete the output).
* You **MUST** create a **demo video** of less than 3 minutes of your solution. **Do not** **show yourself** in the demo video. Upload your video in your personal youtube account or google-drive account and share its link with the instructor through the **Comments** **box** of submission pageas mentioned next (Do not share the video publicly).
* During submission at the dropbox, you should see a **Comments** **box** present near the bottom of the submission page. **Write the link of your video** in this **Comments** **box**.
* Next, upload your solution and submit.

**Exercise 1: Convolutional Neural Networks**

In this exercise you will implement a CNN model for digit classification using tensorflow and MNIST dataset. The site <https://www.tensorflow.org/api_docs/python/tf/keras/datasets/mnist/load_data> will help you to figure out how to fetch the data using a special tensorflow function. Overall, use convolution layers and pooling layers that are eventually followed by a densely connected output layer. Here is the architecture of the network:

* Use Conv2D for two hidden layers. They are activated by ReLU activation function
* After each Conv2D layer, use a mean pooling operation.
* Use Flatten operation.
* Use Dense for the output layer. This layer is to be activated by softmax function.
* Do not use Dropout operation in this architecture.

**Do not use one-hot encoding of the target.**

Compile the model using the loss function **SparseCategoricalCrossentropy** (You need to use this loss function **since you** **must not use** **one-hot encoding of the target**). Train the model using the **top 60 rows** out of 60000 rowsofthe training data (present in x\_train; see below how to obtain the training data in x\_train). Test the model using **top 10 rows** out of 10000 rows oftest data (present in x\_test; see below how to obtain the test data in x\_test). Evaluate the accuracy of the model. **Compare the result with those of the SVM model used in lab2.**

**Note:** If you go down the webpage <https://www.tensorflow.org/api_docs/python/tf/keras/datasets/mnist/load_data> you will see an example usage of the method keras.datasets.mnist.load\_data. It returns four objects of type *ndarray*. Those four objects are x\_train, x\_test, y\_train, y\_test. Another file that could be of help is in a News item under the string **8.1.using.built-in.dataset.of.sklearn.digit.images.for.tensorflow.Conv2D.model.ipynb**

It also has an associated video in the same News item under the string **8.1.using.built-in.dataset.of.sklearn.digit.images.for.tensorflow.Conv2D.model**

Here is another tensorflow website that shows the current way of creating a CNN architecture:

<https://www.tensorflow.org/tutorials/images/classification>

(5 marks)

**Exercise 2: Convolutional Neural Networks**

In this exercise you will build a CNN model for photo classification using tensorflow and CIFAR-10 dataset. The site <https://www.tensorflow.org/api_docs/python/tf/keras/datasets/cifar10/load_data> will help you to figure out how to fetch the data using a special tensorflow function. Use the same technique as in Exercise 1 above to create the CNN architecture. **Train** the model using the **top 50 rows** out of 50000 rowsofthe training data (present in x\_train; see below how to obtain the training data in x\_train). **Test** the model using **top 10 rows** out of 10000 rows oftest data (present in x\_test; see below how to obtain the test data in x\_test). **Evaluate** the accuracy of the model.

**Note:** If you go down the webpage <https://www.tensorflow.org/api_docs/python/tf/keras/datasets/cifar10/load_data> you will see an example usage of the method keras.datasets.cifar10.load\_data. It returns four objects of type *ndarray*. Those four objects are x\_train, x\_test, y\_train, y\_test. Another file that could be of help is in a News item under the string **8.1.using.built-in.dataset.of.sklearn.digit.images.for.tensorflow.Conv2D.model.ipynb**

It also has an associated video in the same News item under the string **8.1.using.built-in.dataset.of.sklearn.digit.images.for.tensorflow.Conv2D.model**.

Here is another tensorflow website that shows the current way of creating a CNN architecture:

<https://www.tensorflow.org/tutorials/images/classification>

(5 marks)

**Evaluation:**

|  |  |
| --- | --- |
| **Design and Functionality:**  Correct design and implementation of requirements  Code explanation if asked | 90% |
| **Documentation of code using comments:**  At least a single-line comment for each functionality | 10% |
| **Total** | 100% |

You must name your Jupyter notebook file(s) according to the following rule:

**YourFullname\_COMP377Labnumber\_Exercisenumber**.ipynb

Example: **JohnSmith\_COMP377Lab1\_Ex1**.ipynb

**Submission rules:**

Submit your solution as a **zip file** that is named according to the following rule:

**YourFullname\_COMP377Labnumber.zip**

Example: **JohnSmith\_COMP377Lab1.zip**

Use 7-zip to compress files (https://www.7-zip.org/download.html).