Deep learning Practical Assignment #3:

Convolutional Neural Networks

Lab date : December 13th, 2021
Deadline to submit : December 31th, 2021 - 11:59 PM (23:59)

Instructions

Read all the instructions below carefully before you start working on the assignment.

- You must do this assignment in groups of **at most 3** students.
- You must implement all algorithms **from scratch**.
- Please submit a jupyter notebook that contains:
  - A section (multiple cells) for your answer to the theoretical questions.
  - A section (multiple cells) for the installation of the used packaged. please specify the exact package version to avoid possible conflicts.
  - A section (multiple cells) for you implementation.
- Please submit your notebook to your corresponding git branch in our git work-space.
- Late submissions will be graded as follows:
  - On time submission grade SG.
  - $SG = 0.95 * SG$ for each day. Capped at $\frac{2}{3} * SG$
- Early submissions will be graded as follows:
  - On time submission grade SG.
  - $SG = 1.02 * SG$ for each day. Capped at $\frac{5}{4} * SG$
- Code that does not work will be graded accordingly. Please focus on the algorithms you are implementing.

Practical assignment objective

- To create a CNN based model.
- Train different model on different dataset.

Datasets

In this lab we’ll use different datasets

- IRIS dataset
- Digit dataset
- Olivetti Faces dataset

1 Implementing CNN architecture !!

In this segment we implement the perceptron algorithm.

In the last lab we have used different datasets with a fully connected network.
By the end of that lab, you had a segment titled **Working towards the CNN**.

1.1 Training form scratch

After using any data augmentation technique and splitting the data to Train and Test.
Implement some CNN architectures seen in class (LE-NET5, VGG-16)
Using different metrics: accuracy, recall, precision, ..., present your best architecture you have created.
Please comment the results.

1.2 Using keras

```
from keras.models import Sequential
from keras.layers import Dense

...
#loading and understanding the data
...

# Sample code to create a fully connected network
model = Sequential()
model.add(Dense(12, input_dim=8, activation=’relu’))
model.add(Dense(5, activation=’relu’))
model.add(Dense(3, activation=’relu’))
...

# compile the keras model
model.compile(loss=’binary_crossentropy’, optimizer=’adam’, metrics=[’accuracy’])
...

# fit the keras model on the dataset
model.fit(X, y, epochs=150, batch_size=10)
```

2 Fine-tuning a pretrained model

Keras present mutilple pretrained famous model. Use few of these models to create a more powerful model for you classification task.
Be careful, you should freeze most of the model and only train few CNN layer or only train the FC segment of the model
To freeze all the model use the following code.

```
for layer in model.layers:
    layer.trainable = False
```

Then you can add few layers to your final model.

2.1 Selecting best model

Using different metrics: accuracy, recall, precision, ..., present your best architecture you have created.
Please comment the results.
For your best model please provide the characteristics of its architectures:

- NB Layer
- NB Neurons
- Activation functions (Non linearity functions)
- Error function (Cost function)
- optimization algorithm

3 CNN vs FC

Using the results from your last lab, please a comparison between the fully connected layer architecture and the CNN architecture.
In addition the metrics you have defined, **be as creative as you want** in the selection of your comparison criteria