Machine Learning (PG)

Monsoon 2020

TOTAL MARKS: 100 | ASSIGNMENT 6 | DUE DATE: DEC 4, 2020

Instructions:

(1) The assignment is to be attempted in groups.
(2) You can use only Python as the programming language.
(3) You are free to use math libraries like Numpy, Pandas, SciPy etc.; any library is allowed for visualizations; and utility libraries like os, pickle etc. are fine.
(4) Usage instructions regarding the other libraries is provided in the questions. **Do not use any ML module that is not allowed.**
(5) Create a '.pdf' report that contains your approach, pre-processing, assumptions etc. Add all the analysis related to the question in the written format in the report, **anything not in the report will not be marked**. Use plots wherever required.
(6) Implement code that is modular in nature. Only python (*.py) files should be submitted.
(7) Submit code, readme and analysis files in ZIP format with naming convention '**A6_groupno.zip**' (one submission per group). This nomenclature has to be followed strictly.
(8) You should be able to replicate your results during the demo, failing which will fetch zero marks.
(9) There will be no deadline extension under any circumstances. According to course policies, no late submissions will be considered. So, start early.

Save your best models. During demo, you must be able to load your saved models and replicate the reported results.

You are free to use Tensorflow, Pytorch or any other DL library. For beginners, we recommend using Keras. As deep learning models often require higher computation and memory, you can try using an online platform like Google Colaboratory or Kaggle Notebooks.

Question 1:

Use the CIFAR-10 dataset for all the experiments. In this question, you will implement a fully functioning CNN for classification. Use a 70:30 data split.
Compile the model by calling `model.compile(optimizer = "...", loss = "categorical_crossentropy' metrics= ["accuracy"])`.
Train the model on train data by calling `model.fit(x = ..., y = ..., epochs = ...)`. *Note that if you run fit() again, the model will continue to train with the parameters it has already learnt instead of reinitializing them.*
Test the model on test data by calling `model.evaluate(x = ..., y = ...)`.

The model architecture is provided in the figure. Perform the following tasks:

(1) Use all three channels for the classification. Use input shape (32 x 32 x 3).
(2) Use SGD optimizer with suitable learning rate, and suitable activations in the required layers.
(3) Try out all the following variations in the architecture of Fig.1. Report the performance of all models. **40 Points**
   (a) No BatchNormalization
   (b) Two Dense Layers. (*Note: The last Dense layer should have 10 nodes as there are 10 classes. For the Dense layer before that, use 64 nodes.)*
   (c) 2 blocks of Cov2D -> BatchNorm2D->MaxPooling2D

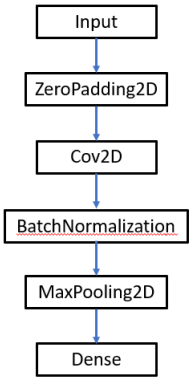

FIGURE 1. CNN architecture

(d) 3 blocks of Cov2D -> BatchNorm2D->MaxPooling2D
Add a table contrasting the performance of the given architecture with all above variations. State your analysis.

(4) Save the best model and plot the accuracy vs epoch. Show the model architecture using `model.summary()` **10 Points**

Question 2: RNN

Use the dataset for carrying out a sentiment classification task.
Details for the dataset:

(1) The format of the dataset is
Sentence Tag
(2) The tag is labelled as 0 and 1 for negative and positive sentiment respectively.

Perform the following tasks:

(1) Download the dataset given and implement RNN for binary classification.
(2) Use pre-trained GloVe glove.6B.50d.txt for vectorizing the text. (Helpful resource) (Note: If you face too many problems while using this, please leave the 5 points and use a Tf-Idf vectorizer instead.) **5 Points**
(3) Report your choice of hyper parameters taken and also justify the network architecture which you implemented(You are free to use architecture of your own choice).**30 Points**
(4) Report the training and the validation accuracy. Save your best model and visualize the model's performance with appropriate plots for loss and accuracy over the epochs. **15 Points**