Week #5

ML Metrics and Overfitting

EECE-4710 IoT and tinyML

Cristinel Ababei
Electrical and Computer Engr., Marquette University

1. Objective

To study commonly used ML metrics. Investigate splitting the dataset into train, validation, test portions. Learn about overfitting and methods to address that. Learn about data augmentation (image rotations, flips, cropping, contrast and brightness changes, etc.) as a way to counteract overfitting. Learn about Early Stopping and DropOut as means to counteract overfitting and at dealing with class imbalance.

2. Assignment

Example 1:
This example is inside the Jupyter Notebook:
TF_MNIST_Classification_v2.ipynb

It is again about classification of the ten digits in images from MNIST dataset. You will investigate the impact of learning rate on accuracy. Start Colab, then, click on File->Upload Notebook and select from your computer the said Notebook. Once you uploaded the Notebook, do first Edit->Clear All Outputs. Then, work through all its code cells.
The example builds a Dense NN:

```
    model = Sequential([Flatten(input_shape=(28,28)),
                        Dense(20, activation='relu'),
                        Dense(10, activation='softmax')])
```

In this part of the assignment, you must investigate five different values for the learning rate:

```
    opt = Adam(learning_rate=0.0001)
```

from the interval [0.00001 – 0.1]. Compare results by discussing “accuracy & val_accuracy vs. epoch” plots you get in each case. This discussion and the plots must be included in the report. Report also the train/validation/test accuracy percentages in all cases.

Example 2:
This example is in this Jupyter Notebook:
Classification_Report.ipynb

It is a very simple example that shows how to generate the confusion matrix for given Actual and Predicted values resulted from a Classification problem with two classes. You have seen this before, but, it is here as a reference for future uses when you will need it.
For this assignment, you only need to execute the entire notebook; you do not need to include anything in your report from this example.

Example 3:
You must also work through this Jupyter Notebook:
IESTI01_data_augmentation.ipynb

This notebook is a modified Google tutorial demonstrate data augmentation: a technique to increase the diversity of the training set by applying random (but realistic) transformations, such as image rotation. It shows how to apply data augmentation in two ways:

(1) Use the Keras preprocessing layers, such as `tf.keras.layers.Resizing`, `tf.keras.layers.Rescaling`, `tf.keras.layers.RandomFlip`, and `tf.keras.layers.RandomRotation`.

(2) Use the `tf.image` methods, such as `tf.image.flip_left_right`, `tf.image.rgb_to_grayscale`, `tf.image.adjust_brightness`, `tf.image.central_crop`, and `tf.image.stateless_random*`.

As part of this assignment, you must:

- Modify the notebook to code your own simple CNN model to do classification (try to re-use code from previously studied notebooks in this class).
- Train, evaluate and test your model using the two different approaches:
  1--Constructing your model as presented at the end of "PART 1: Use Keras preprocessing layers to do Augmentation (The Convenient Approach)". That is, your model will look like this:

  ```
  model = tf.keras.Sequential([
    # Add the preprocessing layers you created earlier:
    resize_and_rescale,
    data_augmentation,
    # Add the usual model layers here:
    # ...
  ])
  ```

  2--Using train_ds, vl_ds, and test_ds from the end of section "Option 1: Using tf.data.experimental.Counter" of "PART 2: Using tf.image to do Augmentation (The Finer Control Approach)".
- Get in both cases the accuracy during train, eval, and test. Compare and discuss your results. Include figures/plots you may create during these experiments.

Example 4:
You must also work through this Jupyter Notebook:
Breast_Cancer_Classification.ipynb

This example does breast cancer binary classification using a Dense NN. We use the Breast Cancer Wisconsin (Diagnostic) dataset. The example shows first how the model is overfitting. Then, it uses Early Stopping and DropOut Layers to address that.
For this assignment, you only need to execute the entire notebook; you do not need to include anything in your report from this example.

3. Deliverables

You must write (typed) a report and upload it as a PDF file on D2L. The report should be named "**LastName_hw5.pdf**". The report should include the following sections:

1) Title + course info + your name
2) **Summary.** Describe in one paragraph what the objective of the assignment is.
3) **Exploring Learning Rate.** Describe your experiments from Example 1. Include the discussion and the plots you obtained.

4) **Data Augmentation.** Describe your experiments from Example 3. Include the discussion and the plots you obtained.
5) **Conclusion.** Present your conclusions and describe what issues you encountered and how you solved them.
6) **References.** Include all references that you used, as a numbered list. Cite them in the report itself; do not just list them here.