Lab 4 - Keyboard Acoustic Emanations

Due on Mar. 10^{th} (T)

1 Description

In addition to power consumption and EM emanations, other physical side channels including acoustic and thermal map can also be effective in leaking information. In this lab, you will implement an attack to steal the *secret* from an audio recording of keyboard typings. The purpose of this lab is to demonstrate the advancements of modern machine learning techniques, which we can leverage to implement side-channel attacks more efficiently. Despite the advantages over traditional methods, machine learning is not magic. We still need carefully prepare the data in a form that is machine solvable.

You will need a profile of keystroke characteristics - preprocessing a set of audio recordings of each character. You will use this profile to train a neural network and use the audio recording (for the secret typing) for inference by the neural network. In the end, you will submit the secret flag you have stolen from the given audio recording.

All audio recordings used in this lab are collected on a Dell keyboard and a microphone on an iPhone headphone. These audio files are collected from one channel of data at 44.1KHz with 16-bit samples. Each file in the training set records one key typing for more than 100 times. The audio file contains the recording of *secret* typings - an 8-character passphrase. Your job is to create a neural net using files in the training set, and then use your trained neural net to recover the passphrase. You may not recover all eight characters, but you can make a guess based on what you have recovered.

Python is recommended for this lab. The Python package scikit-learn is a fantastic tool for data science beginners (click the turquoise box and follow the link to learn more about the package). It hides the complications but is more than sufficient to complete this lab.

Here is a list of files provided to you:

1. Dataset: 26 keys with 100 clicks per key (example file name: *a.wav*); a secret audio recording (file name *secret.wav*).
2. *DataProcessing.py* Python script.
3. Some environmental setting hint files.

2 Processing Raw Data (20 pts)

In this step, you will preprocess your raw data and prepare them for neural nets inputs. The input for a neural net consists of one observation and label. The observation is an audio recording of a key typing event. The label is a character, but you can also assign a number to each of the characters. For example, you can assign *0* to *a* and *1* to *b*.

Each file in the training set contains more than 100 observations, and its filename is the label. You will need to segment each file to 100 parts and extract the pushing peak to 100 inputs and feed them all to your neural net.

This might be too much for you. Hence, a Python 3 script that processes the raw data is provided, which consists of two classes: KeyExtractor and Data. The KeyExtractor class extracts a given number of push peaks from a WAV file. The Data class generates the dataset and saves it as *train.npz*. Usage:

```
python3 DataProcessing.py
```

Question 1. Read and understand *DataProcessing.py*. Explain the plots that appear in the data folder after you executed the code. (e.g., the x-axis represents..., the y-axis represents..., how many points in each plot.) Give one possible reason that we want to generate these plots.

Extra credit (20 pts) Unlike the attack in the orginal paper, the script does not utilize information from the frequency domain and deals with time series directly. To receive the extra credit, implement the data processing method in the original paper, and use your processed data. (e.g., Fig. 4 of the article, use FFTs of the audios as data points.) If you chose the extra credit, you could skip Question 1.

3 Preprocessing Data (20 pts)

In the previous section, we have segmented each click file to 100 data points. In this step, we will standardize the dataset. Recall that in class, we have discussed that for typing the same key, different pressing strengths will generate signals with different amplitude even though their profiles are similar. Therefore the entire data set needs to be standardized. Generally, machine learning algorithms benefit from the standardization of the dataset.

Question 2. Learn about the data preprocessing methods. Test the methods that you think is suitable for our dataset. Which one performs the best? Why?

4 Evaluating Estimator Performance (40 pts)

Ideally, your model should neither underfit nor overfit the dataset. Cross-validation is one of the most effective ways of assessing the performance of your model. In this step, please use 5-fold validation on the entire dataset (which means 80% of the data is used for training, and 20% of the data is used for validation for five times). A two-layer multilayer perceptron (MLP) is recommended for this part, where the size of the input layer is determined by the number of features in your input files, and the size of the output layer is decided by the number of labels (26 in this case). The only hyperparameters of the model you need to tune are the sizes of the two hidden layers. To simplify the process, you can set the two layers at the same size

Question 3. Play with the hyperparameters of the MLP. Report the average and the standard deviation of your highest cross-validation scores.

Question 4. Use the parameters obtained from Question 3, plot the confusion matrix of the 5-fold validation. Analyze the confusion matrix, which letter has the highest accuracy, which letter has the lowest accuracy? Try to explain the performance difference.

5 Attack (20 pts)

Once the previous steps found the hyperparameters, you are going to use your entire data set to train the model to set the parameters. Now your model is fully trained.

You will need to apply your trained model for inference now - to extract eight peaks from the secret file, *secret0.wav*. The process is the same as how you extract data points for training your neural net (import the KeyExtracter class from *DataProcessing.py* and apply it to *secret.wav*). With secret features extracted, use your neural net to recover the secret. **Hint: the secret is an English word repeated twice**.

Question 5. Report the secret. Explain if you did not achieve a complete recovery.

6 What You Need to Turn In

To receive credit on this lab, you will need to turn in a PDF that contains answers to the five questions. You will also need to submit a zipped folder of your source codes. You should also provide a README file that contains all the necessary information to run your code.