15.06.2017

DNN hands on challenge - HW 5

Due date: 24.6.17

Goal for this HW: Understand the data better through what your trained model has learnt

General Submission Notes

1. Submission in **PDF** format only. **File name** should **include your name**.
2. Submission in pairs (one per group).
3. Send everything to both: niv.haim at weizmann ac il & hadar.gorodissky at weizmann ac il

Understanding the data through your trained net

We would like to understand which features in the data are the most significant for predicting the future glucose levels.
We can do this by analysing what our trained model has learnt. Here are some ways to get you started:

Sensitivity of the model to one feature:

- Choose some features, make a data set where each sample is a zero-vector except for one feature which is set to a specific value (e.g. 1). Plot your model predictions as a function the features (e.g.: set all glucose values to zero except for one time-stamp which you set to 1 and plot your network predictions as a function of the timestamp).
- Plot your model predictions as a function of only one feature while holding the rest of the features constant/zero (e.g.: hold all glucose levels (and other input data) constant except for only one glucose entry. Plot the predictions for different values of this specific glucose entry).
- Which features triggers your model the most? Can you understand why these and not others?

Deep dreaming inputs:

- Understand what type of input would make a very high/low prediction.
  Consider the optimization problem where the weights of your trained model are fixed, and now you maximize the prediction with respect to the input (we can also do the same trick for other neurons or even a group of neurons). See how your model “prefers” its input in order to predict very high/low values (or any other values that you are interested in).

You can follow similar implementation with Keras from here: keras/deep_dream.py and this paper.

- How does your model prefers its input for certain values? Explain some of the results if possible?

Analysing feature vectors:

- Each layer of your model is a function $F : R^n \rightarrow R^m$ where n is the dimension of the model-input and m is the dimension of the output of this layer. Each input-sample would result in a "feature vector" for this layer. This can be thought of a different (hopefully better) representation of your data in an m-dimensional space.
- You can generate feature vectors for many data samples and analyse the resulting vectors. (use this as a motivation: colah's visualizing MNIST)
- A popular approach is to use t-sne from the mD space to a smaller space (2D or 3D) and see how this looks. (this is a good presentation of the algorithm by van der Maaten)
- You can use your favourite clustering algorithm or analyse this feature space in any way you see fit.
- What layer made a good representation and why you think this layer did the job better than other layers? What interesting clusters did you get (can you trace certain features in the input that these cluster represent?)

Any other idea you have that makes you understand better why your model predict some samples that way or another.

Write Report

A. Describe the model that you used for the analysis.

B. Describe and attach your results of your analysis. (remember: a picture/graph is worth a thousand words :)

**** We would ask those of you with an interesting story to present your work to the class (with your approval) as a basis for discussion and for the benefit of all ****

Have fun! :)

Hadar & Niv