Machine Learning Engineer Nanodegree

Capstone Proposal

Elena Bushmanova
August and September, 2019

Proposal

Domain Background

Peptidic natural products (PNPs) are small bioactive compounds consisting of amino acids connected via peptide bonds. A PNP may be represented as a graph with amino acids as nodes and bonds as edges. These graphs have either linear, cyclic, or more complex structure. PNPs are important for medicine since many of them are active against bacteria i.e. could be **antibiotics**. One of the main ways to study PNPs is through mass spectrometry. For each PNP you can get a **spectrum** (intensity as a function of the mass-to-charge ratio) or a few by examining it in a black box -- mass spectrometer. These spectra can further be compared against databases of previously characterized compounds using computational methods such as **DEREPLICATOR** (Mohimani H. et al., 2017).

Understanding which spectra correspond to which types of PNPs structure will significantly speed up the DEREPLICATOR since it will be possible to search through smaller sets (cyclic spectra only against cyclic compounds and linear only against linear). At the same time it will increase precision of the algorithm because initial DEREPLICATOR compares any spectrum with any compound and thereby can get such false positive matching as linear spectrum to nonlinear compound and nonlinear spectra to linear compound (not present in an improved algorithm). Also knowledge about the structure itself (separately from DEREPLICATOR) tells scientists some biological properties of the compound represented by its own spectrum. Cyclic PNPs are more stable and biologically more active on average so we can focus on studying of only such spectra thereby saving our resources.

Problem Statement

The problem of this Capstone project is to **categorize PNPs spectra** into spectra corresponding to **cyclic** and **linear** compounds (branch-cyclic and complex classes can also be considered). Thus the program requires spectrum of the unknown compound as input and defines type of the compound structure as output.

Datasets and Inputs

There is already a huge amount of publicly available mass spectra of natural products. It turned out to be possible to detect natural products by their mass spectra and also find new ones missing in the database using a high-throughput technology built on computational algorithms such as DEREPLICATOR.

I'm going to use this one hundred million tandem mass spectra in the Global Natural Products Social (GNPS) molecular networking infrastructure (Wang M. et al., 2016) to select peptide compounds and classify them using Machine learning algorithms. The labels can be taken from molecular structures from GNPS library (trustworthy labels manually obtained by biologists) or from highly-reliable DEREPLICATOR identifications. In both cases it's **several hundred cyclic and non-cyclic structures** and **several thousand spectra** related to them (3-5 different spectra for the structure on average).

Each spectrum is in the MGF Format consisting of list of pairs of mass-to-charge ratio and intensity (see `data/spectra/*.mgf`, `data/spectra_REG_RUN/*.mgf` or `data/GNPS-LIBRARY.mgf`). The compound structure for spectrum from GNPS library is in the Molfile containing information about the atoms, bonds, connectivity and molecular coordinates (see `data/mols/*.mol`). Information about spectra structures identified by DEREPLICATOR can be found in tab-separated values `data/REG_RUN_GNPS/regrun_fdr0_complete.tsv`.

Solution Statement

It's Supervised learning task because example input-output (namely spectrum-structure) pairs exists. I will start with the simplest **Neural network** model. The advantage of Neural networks approach is the possibility of non-linear models with respect to the features. I plan to try various data representations and then do some preprocessing steps. There are two ways to work with these continuous space of input data: **discretize** the raw spectra or directly **approximate** them by functions. For discretization most likely I will use **CNN** to utilize spatial information and for function approximation -- **usual NN**. Of course I also will try a different models (various layers and etc.) and most **Keras** optimizers. The solution can be measured by common metrics such as **AUC**, **precision**, **recall** and more since there is labeled data.

Benchmark Model

A good result that relates to the domain of Natural products identification would be less elapsed time and less FP at the same time obtained by **target matching DEREPLICATOR** (cyclic spectra against cyclic compounds and linear against linear) than by current DEREPLICATOR pipeline. It will mean that the model correctly classify the spectra by their structures into two groups. Thus the benchmark model is **current DEREPLICATOR** results.

For cyclic-linear classification itself **random model** will be used as benchmark model.

Evaluation Metrics

AUC (instead of *accuracy* since the dataset can't be considered fully balanced), **precision**, **recall**, **F1 score** and **FP** as the primary metric are a good choice for evaluation metrics that can be used to quantify the performance of both the current DEREPLICATOR (in the sense of benchmark model) and the Target matching DEREPLICATOR. Here FP means that DEREPLICATOR got a structure that actually doesn't match input spectrum.

Also we can simply compare results of our model with random model results on test set from GNPS library using **FP** metric where false means that spectrum corresponds to other cyclicality than the model got.

Project Design

I will use **Python 3** with **pandas**, **NumPy**, **scikit-learn** and mainly **Keras**. Some steps have already been done in `capstone.ipynb` to get input data representation.

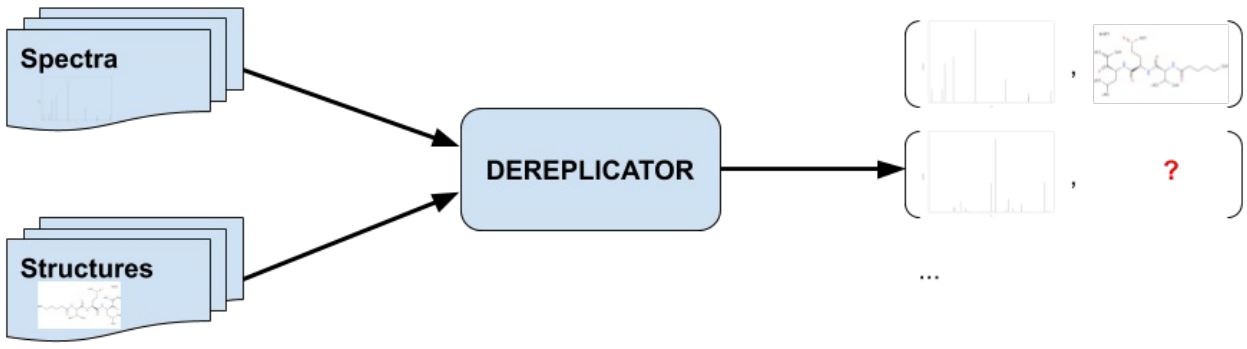

Fig. 1. Spectra and compound structures are passed to DEREPLICATOR. DEREPLICATOR outputs spectrum-compound pairs. Not all PNPs could be identified in this way.

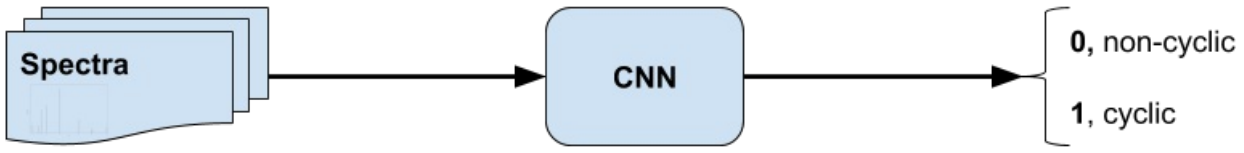

Fig. 2. Spectra are passed to CNN, CNN outputs cyclicality vector whose components are the probabilities of being cyclic or non-cyclic structures.

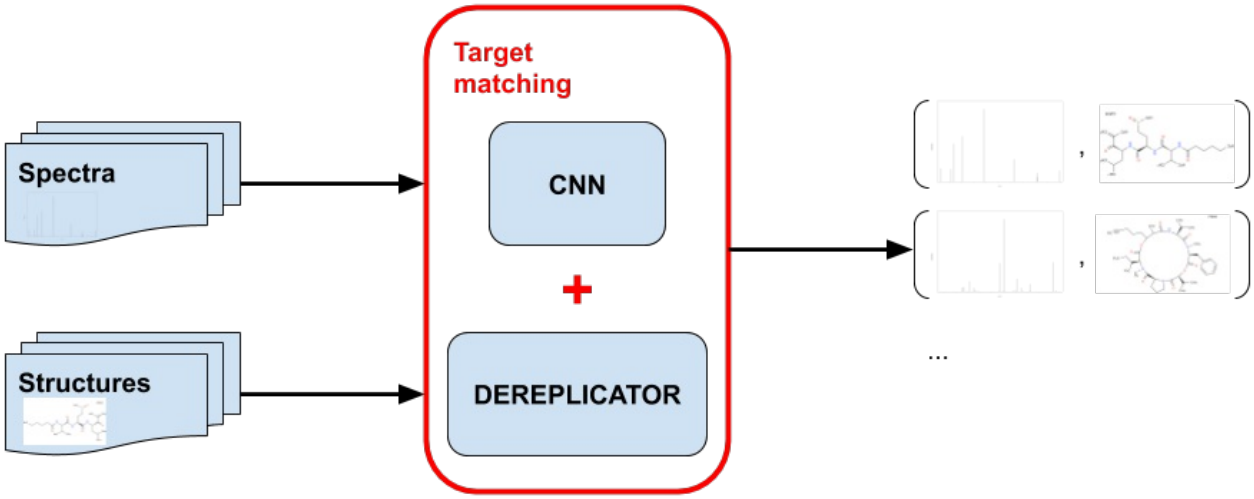

Fig. 3. Spectra and compound structures are passed to Target matching DEREPLICATOR (spectra are passed to CNN; spectra, compound structures, and cyclicality vector are passed to DEREPLICATOR). Target matching outputs more pairs than DEREPLICATOR alone since CNN helps identify more spectra.

The workflow for approaching a solution given the problem includes

- **Collect** the data. Choose peptide not complex compounds from GNPS Public Spectral Library and also the same highly-reliable DEREPLICATOR identifications. GNPS library alone contains *443 peptidic* spectra (*85 linear*, *82 cyclic*, *71 branch-cyclic* and *205 complex*) and DEREPLICATOR identifies *7505 peptidic* spectra (*3101 linear*, *2681 cyclic*, *1692 branch-cyclic* and *31 complex*).
- **Preprocess** the data. It's necessary to think thoroughly here about a representation of the input spectra since what features will consider our algorithm completely depends on it. Each spectrum can be converted into intensity vector by tiny step discretization in which mass-to-charge ratios are indices and intensities are values (let the length be 50-150 thousand). Also spectrum can be approximated by basis functions like RBF. Maybe it will be meaningful to use some data augmentation to increase the set of input data. After that when I understand the data I will identify what kind of preprocessing is needed: scaling, normalization and so on.
- **Split** the data into training, validation and test sets such that both linear and cyclic compounds fall into each of these sets in acceptable proportions.
- **Choose**, **train** and **tune** the model. The initial CNN could include 2 convolutional layers (anyway up to 4 due to the large length of intensity vector), each with 4 filters of size 1×4 and two fully connected layers of 512 and 2 (number of output categories) neuron units. We also use tanh or ReLU activation, max-pooling, and dropout to prevent overfitting. Then I get some intuitions about how these networks work on spectra data by testing them and plotting some scores, change initial model varying layers and other hyperparameters, use different optimizers and also try any more models. There are some articles about Deep learning on mass spectra data into which I want to dig deeper (mainly Tran N. H. et al., 2017 and 2019)
- **Evaluate** the solution. After getting two groups of spectra by approved network run DEREPLICATOR for cyclic spectra against cyclic compounds and linear against linear separately. Compare FP and elapsed time for these results and for DEREPLICATOR on full set of spectra. Also compare with random model and compute FP for approved network without considering DEREPLICATOR pipeline.