Gas Chromatography - Mass Spectrometry (GC-MS) Prediction Models

Two deep learning models using the Tensorflow framework:

1. spectra2formula: predict molecular formula from GC-MS spectra
2. spectra2smiles: predict SMILES (simplified molecular-input line-entry system notation) from GC-MS spectra. (www.daylight.com/dayhtml/doc/theory/theory.smiles.html)

Data

This supervised training relies on the NIST data set (https://www.nist.gov/srd) which contains 242466 spectrum-label pairs. This training data contains a total of 71 different elements/isotopes and to initially reduce the complexity of the problem, only covalently bound compounds containing (C, O, H, N) were selected reducing the number of samples to 116354. Out of these 116354 samples, 6354 were saved for a final testing set and 10000 were used as a validation set to assist in hyper-parameter picking and to prevent overfitting. This resulted in 100000 final training samples.

1 Spectrum Input: The spectral input is represented as a two-dimensional numerical matrix with the rows equal to the number of samples and the number of columns equal to the highest mass integer in the spectra (1760). The intensities for each spectral peak in the NIST data range from 0.0 to 999.0 and are subsequently normalized to a range of 0.0 to 1.0.

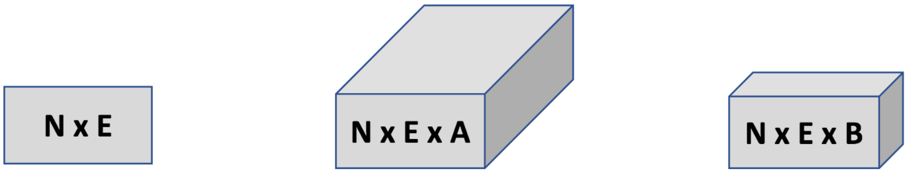
2a Formula Labels: The formula label is represented as a two-dimensional matrix with the rows equal to number of samples and the number of columns equal to the number of chemical elements. For predicting the formulas from only the organic compounds, the number of elements equals 4 (Carbon, Hydrogen, Nitrogen, Oxygen). The value of the matrix is an integer representing the number of atoms of the respective element contained in the compound and thus, predicting the formula is a multi-class regression problem.

2b Formula Labels (Classification): Instead of representing the number of each element as a real number and treating the problem as multi-class regression, the problem can be converted into a multi-class classification problem. An advantage to this approach is that the answer is associated with a probability of being correct which regression does not allow. The label is represented as a three-dimensional matrix with the first dimension equal to the number of samples, the 2nd dimension equal to the number of elements and the last dimension equal to the largest number of any element found in the data set. The largest number of any element found in any sample in the organic data set is 154 hydrogen atoms. Thus, the 3rd dimension is size 154 with the corresponding elements being a 1.0 if that number of the element is found in that sample or 0.0 elsewise. This is now a four-class classification problem with each class taking 154 possibilities.

2c Formula Labels (Classification - Binary): To reduce the number of possibilities for each of the four classes just mentioned down from 154, the label can be expressed with the number of each element represented in binary notation instead of decimal. Thus, with 8 digits (2^8=256), the number of each element can be represented more compactly.

Formula Label Variants

N= Number of Samples
E = Number of Chemical Elements
A= Maximum number of any Chemical Element in the Data Set
B = Log base2 of the Maximum number of any Chemical Element in the Data Set (Rounded Up)



Regression
Values are integers

Classification
Values are either 1.0 or 0.0

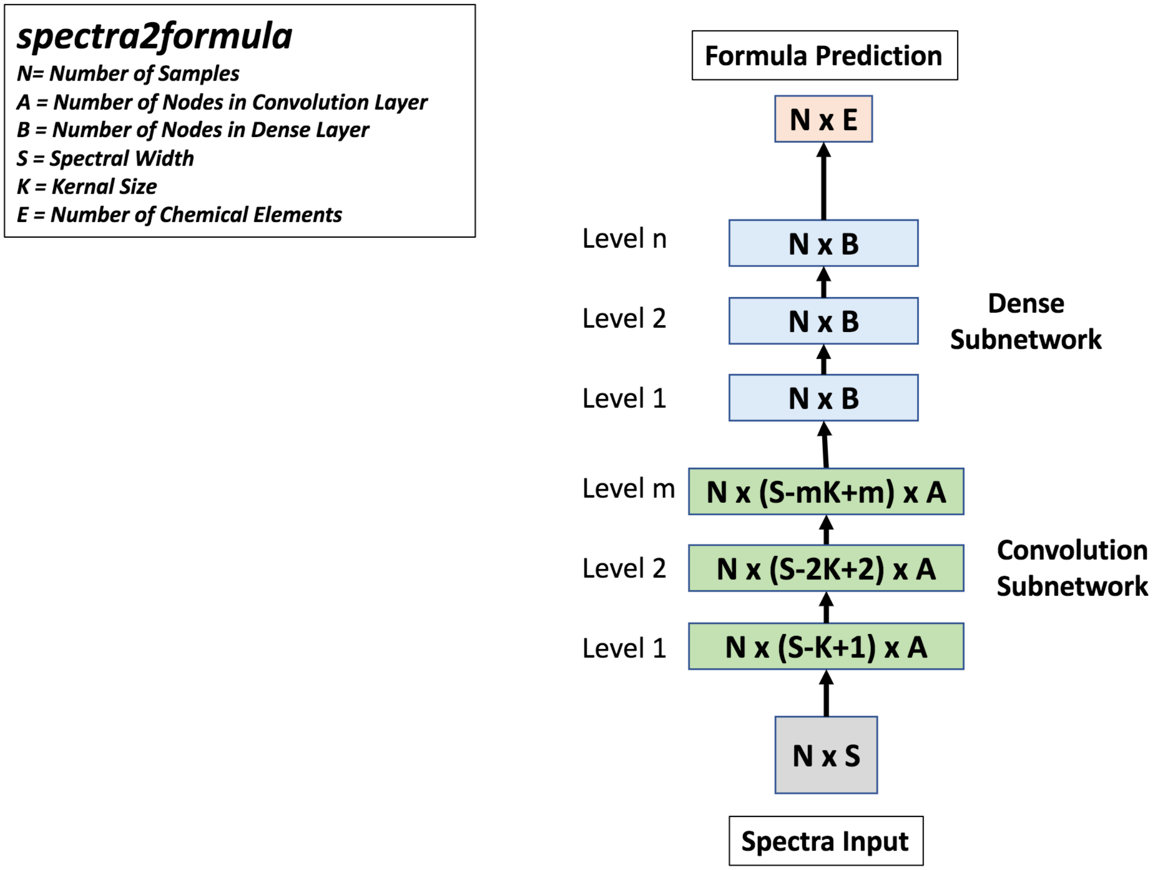
Classification (Binary)
Values are either 1.0 or 0.0

3 SMILES Labels: The canonical SMILES maintain all the structural information of the molecule within a string. The SMILES label is represented as a three-dimensional matrix in the form (number of samples x length of SMILES x vocabulary size of SMILES). A value of 1.0 is added if the character is present in that location in the SMILES string and a 0.0 elsewise. For the NIST data, the length of the longest SMILES is 185 and there are 17 possible choices for the vocabulary. Note that covalent Hydrogen is not needed as an explicit character since the location of hydrogen in a compound is easily determined from the canonical SMILES and elementary bonding rules. The 17 characters used are: C, O, N, =, #, (,), 1, 2, 3, 4, 5, 6, 7, 8, 9, End of String

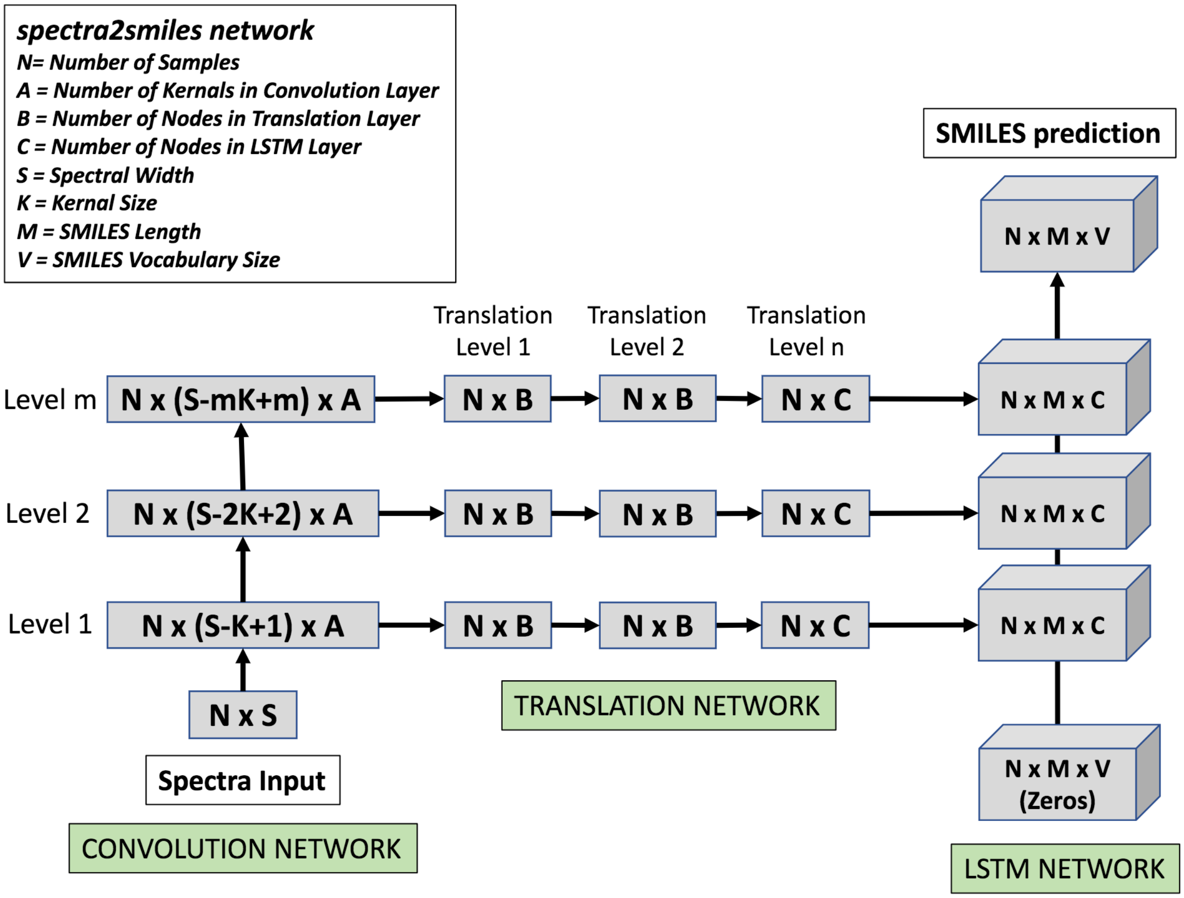
Model

(see spectra2formula_model.pdf and spectra2smiles_model.pdf for visualization)

1. spectra2formula: The spectra input is fed into a variable level convolution subnetwork that follows the following repeating structure: Convolution-> Pooling-> Batch Norm-> Dropout-> Elu Activation. The resulting state is fed into a fully connected variable level dense subnetwork that follows the following repeating structure: State-> Batch Norm -> Dropout -> Elu Activation. A final set of weights transforms the last state to the regression results for the four elements. The model is shown below. The classification variants of spectra2formula are the same except for the shape of the final output.



2. spectra2smiles: The spectra input is fed into a variable level convolution subnetwork that follows the following repeating structure: Convolution-> Pooling-> Batch Norm-> Dropout-> Elu Activation. The state from each convolution layer is fed into a fully connected variable level "translation" subnetwork. The translated convolution states are then fed into their corresponding LSTM layer. In addition to the translated convolution states, the LSTM subnetwork is input with a zero matrix the size of the SMILES labels. A final set of weights transforms the output of the LSTM subnetwork to the SMILES prediction. The general model is shown below.

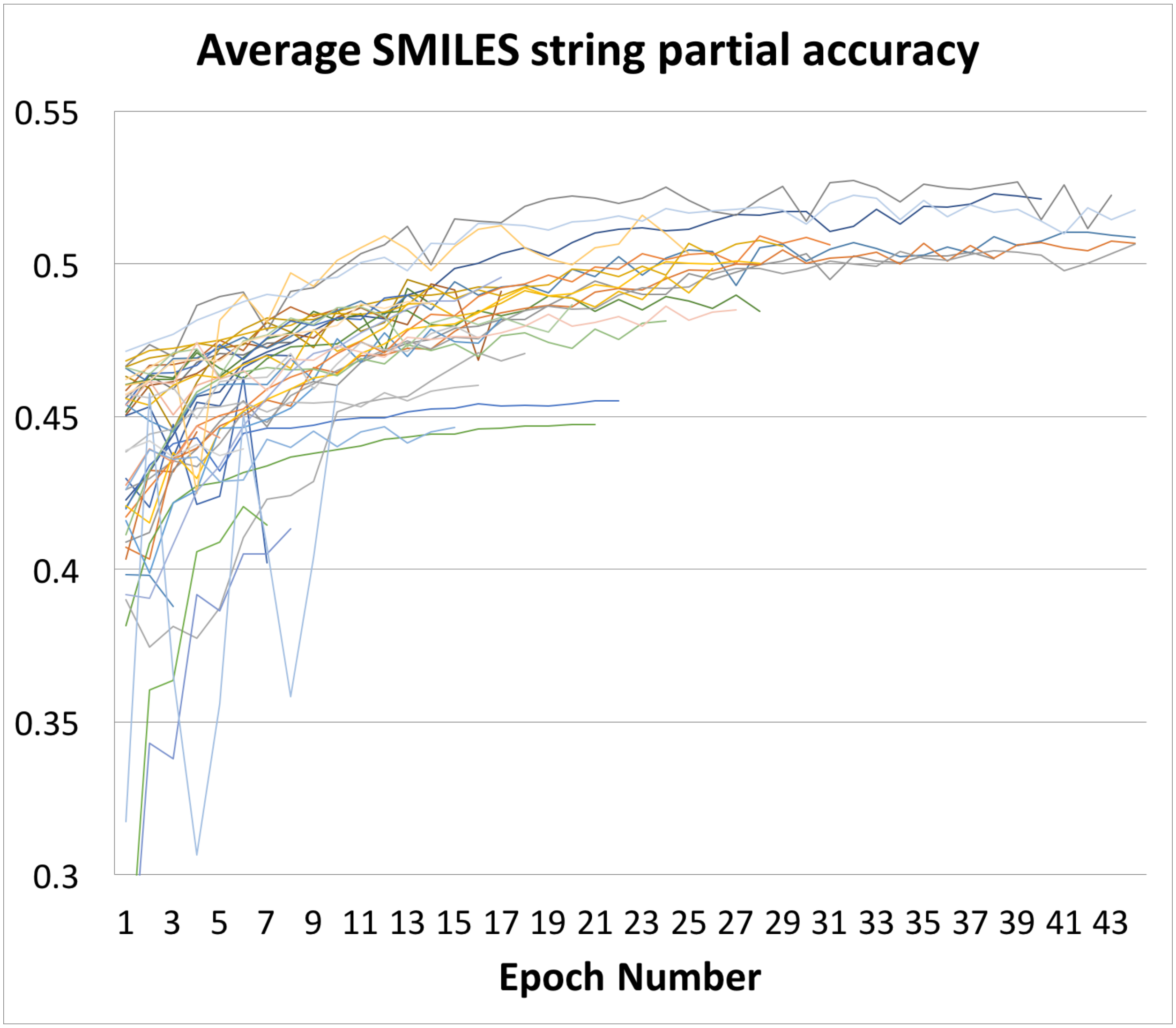


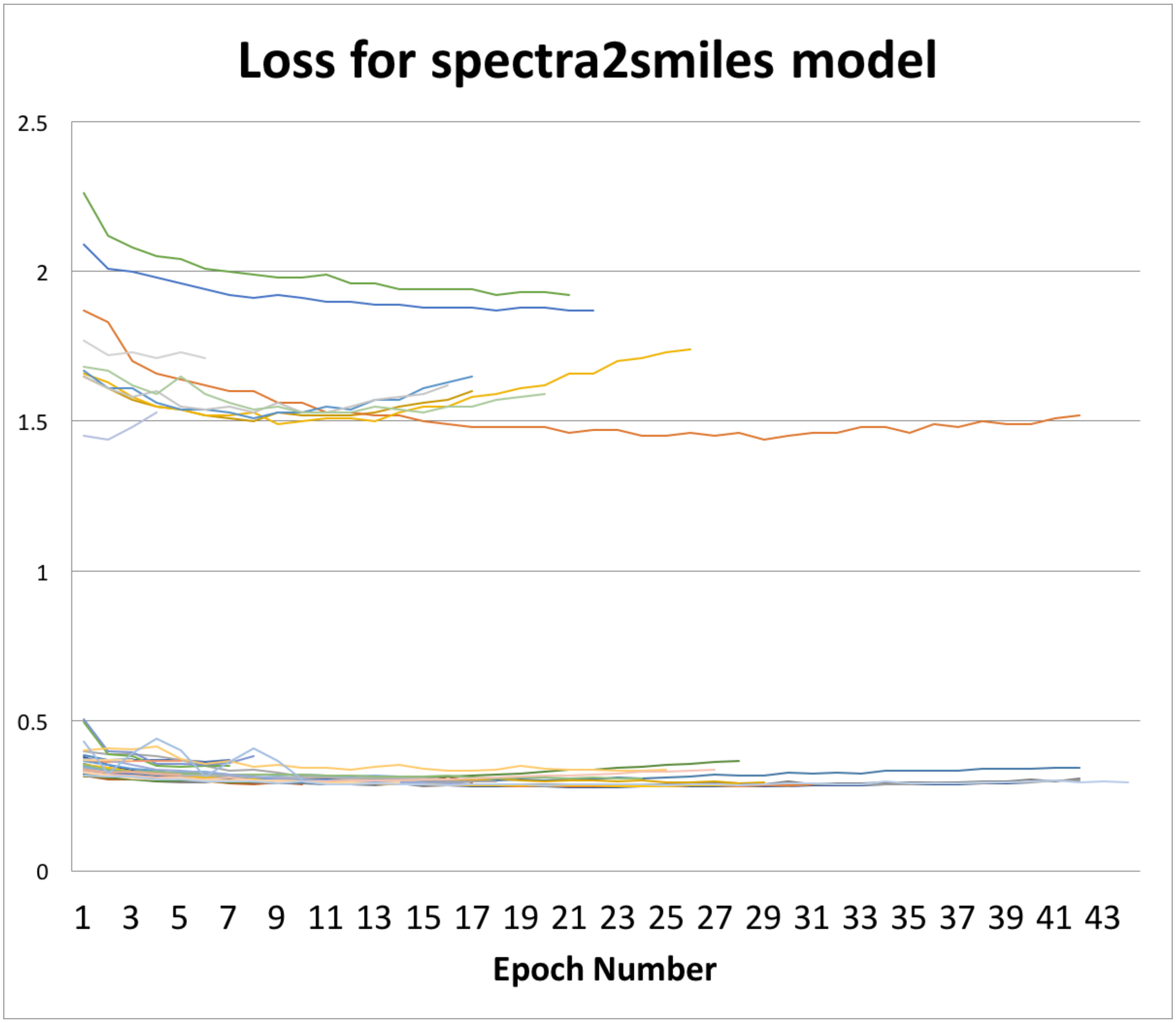
Regularization

Several techniques were tried to improve generalization performance. These include the standard L2-regularization and node dropout methods as well as adding Gaussian noise to the gradients during back propagation. (Arvind Neelakantan, Luke Vilnis, Quoc V. Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, James Martens, "Adding Gradient Noise Improves Learning for Very Deep Networks", arXiv:1511.06807)

Results

For the spectra2smiles model, the accuracy results for the validation set were very low reaching around 1% which was not unexpected (see discussion below). Due to the difficulty in predicting the entire SMILES, I instead chose to investigate what was the average percentage of each SMILES that could be predicted. For example, if the real SMILES was 'CCON' and my model predicted 'CCOC', then the partial accuracy would be 75% while the authentic accuracy for this sample would be a dismal 0%. As can be seen in the following plot which displays this partial accuracy for some of the best models, the partial accuracy for validation samples only reaches around 53%. It is important to note that even the training samples only reach an average SMILE partial accuracy of 70% for my best model. Furthermore, the validation sample loss illustrated that overfitting has already occurred for most of the models (note that the different scales for loss are due to both L2 and gradient noise being added to certain models)





Discussion

The objective of this project was to predict either the chemical formula or the SMILES string from the GC-MS experimental spectra. Two models were created to solves these problems and are called spectra2formula and spectra2smiles respectively.
There are several challenges that presented themselves which are unique to mass spectrometry prediction and that are the stumbling blocks to good accuracy scores. Firstly, while 71 different elements are represented in the NIST dataset, many of these are found in only 1-4 samples rendering them non-ideal for serious feature extraction.

By focusing on only the organic compounds, the number of samples for training was effectively cut in half from 200 thousand to 100 thousand. Secondly, the GC-MS experimental data from the is low resolution with each mass/charge peak taking whole integer values. This means that separate peaks representing unique atoms are combined obscuring good features. Thirdly, there is a large disparity in the ratio of different elements with the decreasing order of hydrogen, carbon, oxygen, nitrogen. For the specra2formula model, this is not a factor since for each of the four elements, there is a distinct regression output. However, for the spectra2smiles model, the choice of which character to predict at EACH location of the SMILES string is dominated by choosing carbon (covalent hydrogen is not an option for SMILES string as they can be implicitly determined). It was noted that for the spectra2smiles model, the average percentage of each SMILES that was predicted correctly jumped to around 46% and then slowly increased to a maximum of around 53%. The 46% was easily obtained by the network assigning all characters to be chosen as carbon and it was only very slowly that this degeneracy was disrupted. Fourthly, the size of the samples is not uniformly distributed with some SMILES strings as small as three characters and as large as 185 characters while the bulk of the samples fall in the middle around 70 characters. The difficulty in getting high accuracy is not surprising given that as many as 185 sequential characters must ALL be predicted correctly for the SMILES string to be considered correct.