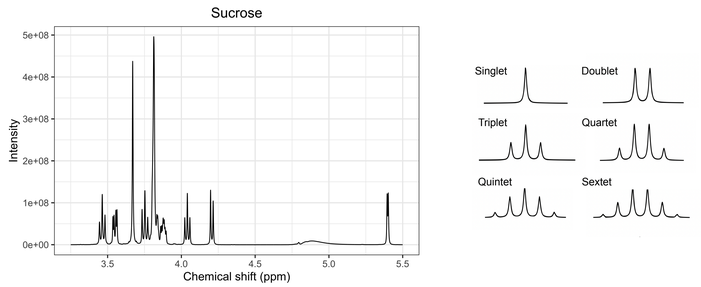
Compound Classification Neural Network for Nuclear Magnetic Resonance Spectra

Ruis MacDonald
Department of Chemical Engineering, Dalhousie University

Background

Peaks in an NMR spectrum represent the magnetic fields around the nuclei in a sample. Pure compounds have a unique multiplet pattern (multiplets at 65536 possible ppm positions). Multiplets are 18 peak groups with specific area ratios and separation distances. 6 multiplet types are shown below.



Objective

The model is a supervised single-channel source separation network inspired by Cov-TasNet https://ieeexplore.ieee.org/abstract/document/8707065. The model separates a mixture of 3 single compounds into the single compounds and classifies each.

Motivation

The long-term objective is a network that classifies compounds in an NMR spectrum. The network will be trained on database multiplet data for 877 pure compounds. The input to the network will be a mixture of multiplet sequences of 50-100 pure compounds, and the model will separate the mixture into the single compounds and classify each.

An accurate automatic NMR classification method would be extremely valuable because manual compound identification is time-consuming, challenging and expert dependent. There are consistent weaknesses in existing automatic approaches that would be overcome using neural networks. Current approaches are slow, expensive (commercial), requires user input, and/or results in high false positives and negatives. Neural networks would improve on current methods with high computation speed, robustness and accuracy and low user dependence.

Datasets

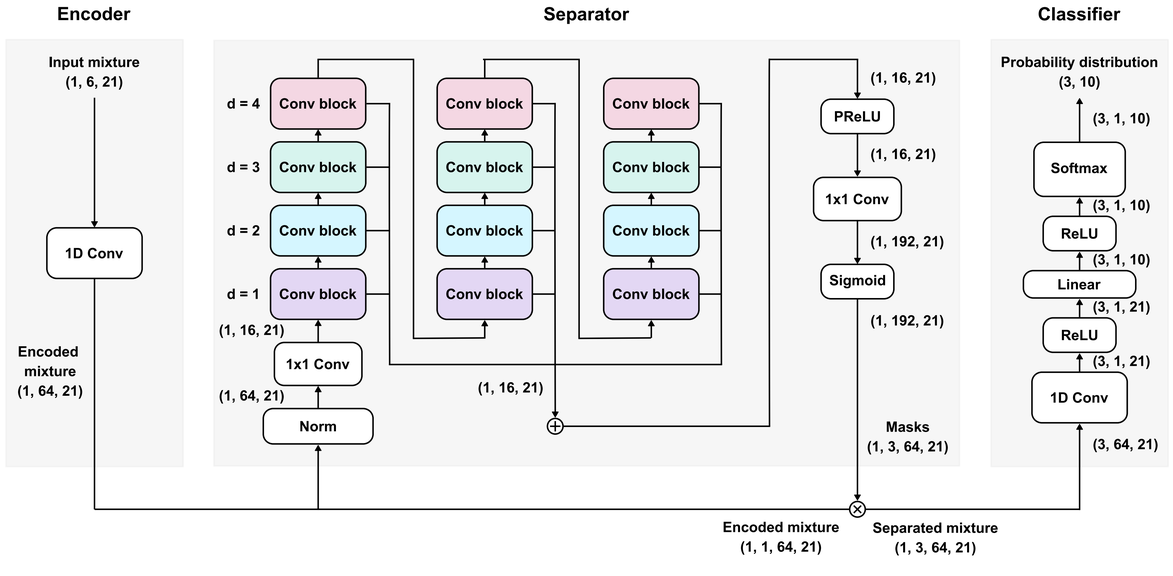
The synthetic data is a simplified abstraction of the real NMR data. Multiplets are (1, 6) arrays with elements representing area ratios between peaks. There are 6 different multiplets representing singlets, doublets, triplets, quartets, quintets and sextets. Single compounds are the addition of 2 sets of multiplets occupying a percentage of 21 possible ppm positions. Addition is used to represent multiplets of 2 compounds occupying the same position. Mixed compounds are the addition of 3 single compounds. The testing and training datasets each have 10 mixed compounds.

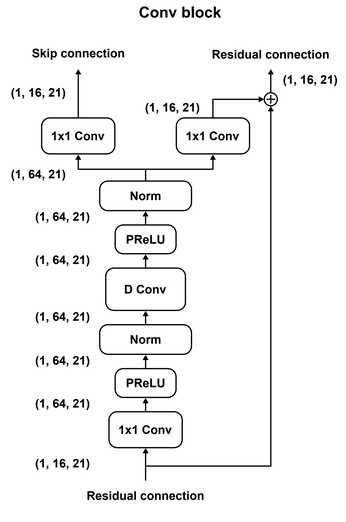
Model

The model uses a temporal convolutional network with stacked dilated 1D convolution blocks to estimate the mask for each compound. The model consists of an encoder, separation module and classification module.

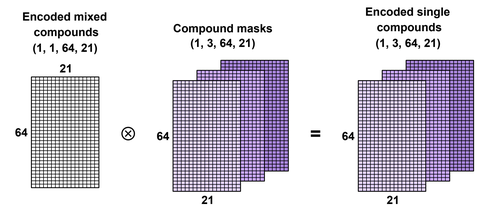
Encoder: The encoder uses 1D convolution to transform the (1, 6, 21) (effectively one-dimensional because each multiplet is (1, 6)) input to a (1, 64, 21) (multi-dimensional) representation.

Separation: The model then uses layer normalization to normalize each channel and pointwise convolution to reduce the number of channels (bottleneck layer).





The model estimates the masks using 3 stacks of 4 layers of convolution blocks with increasing dilation factors. Dilation factors are used to capture the long-range dependence of the sequence. Each convolution block uses depthwise separable convolution (depthwise then pointwise convolution, which reduces the number of parameters thus the model size). The skip connections are summed and the 3 masks are estimated using a pointwise convolution and a Sigmoid activation function. The mixture is separated into the single compounds by element-wise multiplication of the encoded input mixture and the estimated masks.

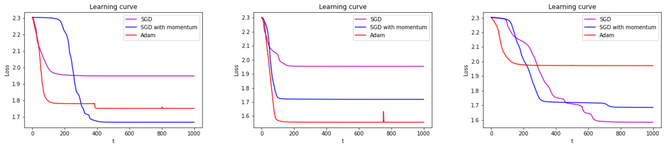


Classification: 1D convolution is used to transform the multi-dimensional separated mixture to one-dimensional, a linear layer is used to transform the third dimension from 21 (ppm positions) to 10 (compound classes), and a Softmax layer is used to determine the probability distribution across the classes (the output).

The model is trained using cross entropy loss and 1000 epochs. The stochastic gradient descent, stochastic gradient descent with momentum of 0.99 and Adam optimizer were used, each with a weight decay of 1e-5. These parameters were chosen because they result in the lowest losses.

Results

The optimizer that results in the lowest training loss and highest testing accuracy changes each time the model is run. I'm not sure why this happens because I use a random seed so that the data stays the same. Example results are 1.9537 training loss and 0.8286 testing accuracy for SGD (results are always around these values). The learning curve for 3 different runs are shown below.



Future work

Immediate future work is to solve the results changing with each run, and to improve performance by adjusting parameters. Other future work is to modify the network so the number of single compounds in the mixture does not need to be known, because it is unknown for real data. One idea I had is to set the number to greater than the maximum possible value and to cluster the estimated masks to get the true single compound masks. Other future work is to use both the raw spectral data and derived multiplet data as inputs. This may improve performance because 1D convolution works better on continuous data, and the network will have 2 layers of paired information to work with. The model will then be scaled up and used with real data.