Lee Xiong An
Personal Project 1
Predicting Cell Penetrating Peptides.

Background

There are three main classes of medicinal drugs with very different sizes (0.1 - 150kda). These drugs are small molecules (<0.9kDa) , peptides ~1kda and large antibodies ~150kda. Small molecules easily enter cells but are less specific due to their small size, while large antibodies are very specific but do not enter cells at all. Peptides are the middle ground being specific yet small enough to enter the cells.

However, the mechanism in which peptides enter cells are unknown and under active research. I wish to see if deep learning can find features or patterns explaining why certain peptides enter cells and others do not.

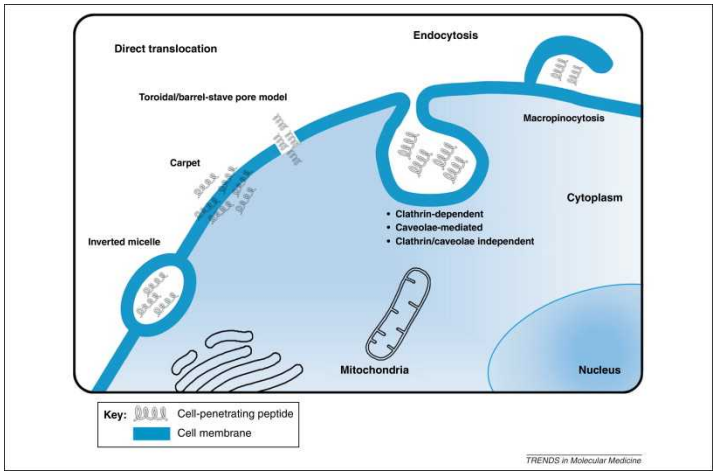

Figure : Cell penetration by peptides via postulated mechanisms.

Data format

Data are present as amino acid sequences. Amino acids contain 20 non-unique characters (ACDEFGHIKLMNPQRSTVWY), and these sequences are usually 4-30 amino acids long. Furthermore, as amino acids are well characterized experimentally, additional thermodynamic features will be given (4 features). Outputs will be permeable or non-permeable classification.

Data collection :

Our dataset after preprocessing has 500 positive and 9200 negative sequences.

Since there was a ratio of 1 : 18 positive to negative data-points, I duplicated the positive class 18 times for the training set to balance the skewed class ratio during training. This improved the AUC metric notwithstanding that AUC is already a balanced metric.

A public dataset containing 110 positive and 34 negative examples will be given due to non-disclosure agreements.

Note* The code given will reweigh the public dataset to 1980pos : 34neg and appear nonsensical.

Model :
I used a two models: A) LSTM and B) Convolution NN since these models accept peptide sequences of varying lengths. Both models accept the same inputs and use a similar dense output layer and contain similar number of parameters ~800k.

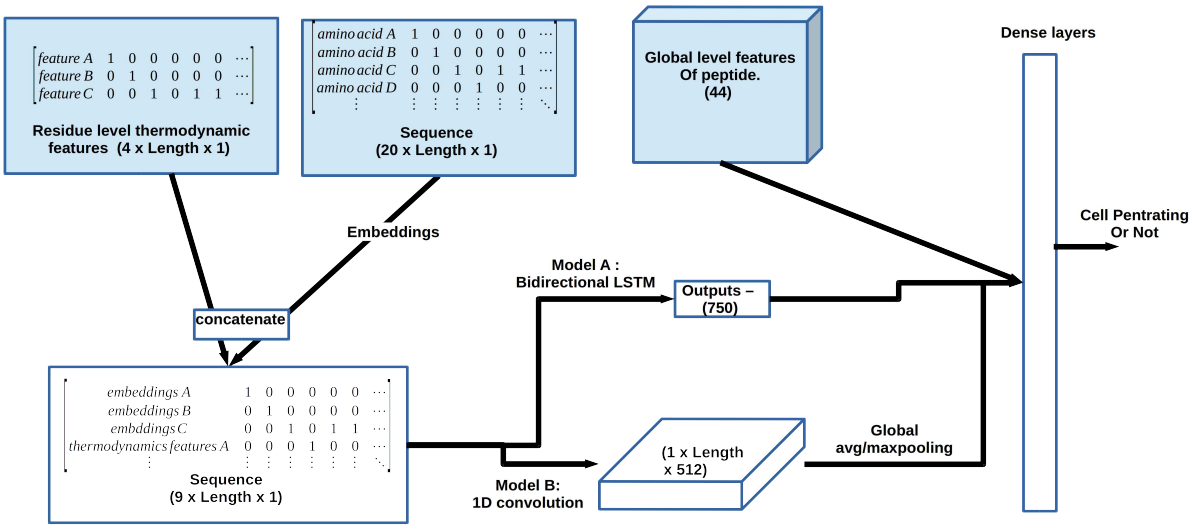

Figure 2 : Inputs colored solid blue, which are residue level thermodynamics features, sequence and global features of peptide. Global features are sequence and thermodynamic features averaged over the entire length of the peptide while residue features are thermodynamics features over each amino acid.

Model training :
The data was split into train, validation and test in ratios 3:1:1 with a total of 20 combinations of train/validation/test. The model was trained with stochastic gradient descent (batch for RNN) to sample the global minima of the cost function. To sample the the local minima, the learning rate has a cosine decay over each epoch. Models were picked based on validation AUC, and the final performance of the model based on the test set was noted down.

To prevent over-fitting of the test set, the validation set was used to pick the best model, and the performance was recorded on the test set. However, I believe over-fitting may have occurred and this will be explained in the next section.

The CNN model does not require a GPU and takes 6 hours to complete while the RNN model takes 18 hours on a K40 GPU. This is probably an overkill.

Models AUC: (train AUC ~ 0.999 – 0.99999)
Convolution NN – 0.930
Xgboost (only global features)– 0.954
LSTM – 0.935
Ensemble model – 90% xgboost, 5% CNN/LSTM – 0.958

I find it interesting that xgboost seems to perform much better then deep learning and takes much less to train. It could be due to the relatively less complex data (this problem only contains 20 amino acids). However, when doing inference on sequences of interest , the CNN and xgboost models (LSTM not attempted) show great disparity.

One possible reasons is due to over-fitting since the train and test set are very similar, being selected with systematic sampling. As a benchmark against unseen data, I will cluster the data into two separate clusters, training the model on one cluster and predicting on the other. The AUC will give a gauge against over-fitting the test set. This is subjected to time constraints and included in the next paragraph (if done).