

## Model Description:

### CNN

Conv1D along features, 128 units, kernel size 5, relu activation  
Max Pool, stride = kernel = 3  
Conv1D with 64 units, kernel size 5, relu activation  
Average Pool, stride = kernel = 3  
Conv1D with 32 units, kernel size 5, relu activation  
Max pool output = 1  
0.3 Dropout in training  
Dense layer with 128 units, relu activation  
Dense layer with 64 units, relu activation  
Softmax output layer  
Adagrad optimizer with categorical crossentropy loss

### RNN

4 layers of Bidirectional GRUs with 384 units  
1 Dense layer with 512 units with relu activation  
1 Softmax output layer  
0.5 Dropout in training  
RMSprop optimizer with categorical crossentropy loss

## Performance Improvement

### Preprocessing

Normalize inputs  
Switch label for padding to silence

### Training / Model

Decrease training epochs : prevent overfitting  
Increase units per layer: 128 -> 512/1024  
Increase layers: 2 -> 4 (5 shows no improvement)

## Postprocessing

Hard cut off by input sequence length

Filter out phones with less 3 per chain (from observed minimum chain length of 3 in training data)

## Experiments:

### RNN with LSTM

Refuses to learn anything

### RNN with Conv2DLSTM

Performance decreases

### CNN + RNN:

Results similar to only CNN

### Dense -> RNN/CNN -> Dense

Performance decreases

### Ensemble

Performance same as the best model with ensemble

### Pure Tensorflow

Headaches, no visible gains from Grid LSTM / Bidirectional Grid LSTMs