Models For MSCRCM

1. CNN

**90.05**

**60.05**

from keras.models import Sequential

from keras.layers import Dense, Dropout, Activation, Flatten, LSTM

from keras.layers import Convolution1D, Conv1D, MaxPooling1D, GlobalAveragePooling1D

from keras.optimizers import Adam

from keras.utils import np\_utils

from sklearn import metrics

num\_rows = 20

num\_columns = 2

num\_channels = 1

x\_train = x\_train.reshape(x\_train.shape[0], num\_rows, num\_columns)

x\_test = x\_test.reshape(x\_test.shape[0], num\_rows, num\_columns)

num\_labels = yy.shape[1]

filter\_size=1

# Construct model

model = Sequential()

model.add(Conv1D(filters=32, kernel\_size=2, input\_shape=(num\_rows, num\_columns), activation='relu'))

model.add(MaxPooling1D(pool\_size=1))

model.add(Dropout(0.25))

model.add(Conv1D(filters=64, kernel\_size=2, activation='relu'))

model.add(MaxPooling1D(pool\_size=1))

model.add(Dropout(0.25))

model.add(Conv1D(filters=128, kernel\_size=2, activation='relu'))

model.add(MaxPooling1D(pool\_size=1))

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(64, activation='relu'))

model.add(Dense(num\_labels, activation='softmax'))

**92.5**

**57.5**

from keras.models import Sequential

from keras.layers import Dense, Dropout, Activation, Flatten, LSTM,BatchNormalization

from keras.layers import Convolution1D, Conv1D, MaxPooling1D, AveragePooling1D

from keras.optimizers import Adam

from keras.utils import np\_utils

from sklearn import metrics

num\_rows = 20

num\_columns = 2

num\_channels = 1

x\_train = x\_train.reshape(x\_train.shape[0], num\_rows, num\_columns)

x\_test = x\_test.reshape(x\_test.shape[0], num\_rows, num\_columns)

num\_labels = yy.shape[1]

filter\_size=1

# Construct model

model = Sequential()

model.add(Conv1D(filters=32, kernel\_size=2, input\_shape=(num\_rows, num\_columns), activation='relu'))

model.add(MaxPooling1D(pool\_size=1))

model.add(Dropout(0.25))

model.add(AveragePooling1D(pool\_size=1))

model.add(Conv1D(filters=64, kernel\_size=2, activation='relu'))

model.add(MaxPooling1D(pool\_size=1))

model.add(Dropout(0.25))

model.add(AveragePooling1D(pool\_size=1))

model.add(Conv1D(filters=128, kernel\_size=2, activation='relu'))

model.add(MaxPooling1D(pool\_size=1))

model.add(Dropout(0.25))

model.add(AveragePooling1D(pool\_size=1))

model.add(Flatten())

model.add(Dense(64, activation='relu'))

model.add(Dense(num\_labels, activation='softmax'))

CRNN

from keras.models import Sequential

from keras.layers import Dense, Dropout, Activation, Flatten, LSTM,BatchNormalization

from keras.layers import Convolution1D, Conv1D, MaxPooling1D, AveragePooling1D

from keras.optimizers import Adam

from keras.utils import np\_utils

from sklearn import metrics

from keras import regularizers

num\_rows = 20

num\_columns = 2

num\_channels = 1

x\_train = x\_train.reshape(x\_train.shape[0], num\_rows, num\_columns)

x\_test = x\_test.reshape(x\_test.shape[0], num\_rows, num\_columns)

num\_labels = yy.shape[1]

filter\_size=1

# Construct model

model = Sequential()

model.add(Conv1D(filters=32, kernel\_size=2, input\_shape=(num\_rows, num\_columns), kernel\_regularizer=regularizers.l2(0.01),activation='relu'))

model.add(BatchNormalization())

model.add(MaxPooling1D(pool\_size=1))

model.add(Dropout(0.25))

model.add(Conv1D(filters=32, kernel\_size=2, input\_shape=(num\_rows, num\_columns),kernel\_regularizer=regularizers.l2(0.01), activation='relu'))

model.add(BatchNormalization())

model.add(MaxPooling1D(pool\_size=1))

model.add(Dropout(0.25))

model.add(Conv1D(filters=32, kernel\_size=2, input\_shape=(num\_rows, num\_columns),kernel\_regularizer=regularizers.l2(0.01), activation='relu'))

model.add(BatchNormalization())

model.add(MaxPooling1D(pool\_size=1))

model.add(Dropout(0.25))

model.add(LSTM(96,kernel\_regularizer=regularizers.l2(0.01)))

model.add(Dropout(0.25))

model.add(Dense(64,kernel\_regularizer=regularizers.l2(0.01), activation='relu'))

model.add(Dropout(0.25))

model.add(Dense(num\_labels, activation='softmax'))