In [1]:

```
import numpy as np
from random import shuffle
from collections import Counter
import keras
from keras.utils import np_utils
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation, Flatten
from keras.layers.convolutional import Convolution2D, MaxPooling2D, MaxPooling
from time import time


def LoadData ():
    # X = np.loadtxt ('UnscaledX.txt')
    X = np.loadtxt ('ScaledX.txt')
    Y = [l.strip() for l in open ('Y.txt').xreadlines()]
    print 'loaded a X and Y matrices of shapes: ', X.shape, len(Y)
    return X, Y

def RandomizeXY (X, Y):
    Z = list(zip(X, Y))
    shuffle(Z)
    X, Y = zip(*Z)
    print 'randomized zip (X,Y)'
    return X, Y

def GetTrainTestSplit(X, Y, TrainPercent=0.8):
    TrCutOff = int(len(Y) * TrainPercent - 1)
    TeCutoff = TrCutOff - len(Y)
    print 'training and test cutoff:', TrCutOff, TeCutoff

    X_train = np.array(X[:TrCutOff])
    Y_train = np.array(Y[:TrCutOff])
    X_test = np.array(X[TeCutoff:])
    Y_test = np.array(Y[TeCutoff:])

    print 'X_train, Y_train, X_test, Y_test shapes'
    print X_train.shape
    print Y_train.shape
    print X_test.shape
    print Y_test.shape

    return X_train, Y_train, X_test, Y_test

def MakeYAsNum (Y):
    Ytmp = []
    for y in Y:
        if 'positive' == y:
            Ytmp.append (1)
        elif 'negative' == y:
            Ytmp.append(2)
        elif 'conflict' == y:
            Ytmp.append(3)
        else:
            Ytmp.append(0)
    return Ytmp
```

```
Using Theano backend.
```

In [2]:

```
X, Y = LoadData()
X, Y = RandomizeXY (X, Y)
X_train, Y_train, X_test, Y_test =  GetTrainTestSplit(X, Y, TrainPercent=0.8)

MaxFeatsWithPadd = 69
WordVecDims = 300

X_train = X_train.reshape(X_train.shape[0],1, MaxFeatsWithPadd, WordVecDims)
X_test = X_test.reshape(X_test.shape[0],1, MaxFeatsWithPadd, WordVecDims)
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
print('X_train shape:', X_train.shape)
print('X_test shape:', X_test.shape)
print(X_train.shape[0], 'train samples')
print(X_test.shape[0], 'test samples')


Y_train = MakeYAsNum(Y_train)
Y_test = MakeYAsNum(Y_test)

print(Counter(Y_train))
print(Counter(Y_test))

nb_classes = 4

Y_train = np_utils.to_categorical(Y_train, nb_classes)
Y_test = np_utils.to_categorical(Y_test, nb_classes)
```

```
loaded a X and Y matrices of shapes:  (3698, 20700) 3698
randomized zip (X,Y)
training and test cutoff: 2957 -741
X_train, Y_train, X_test, Y_test shapes
(2957, 20700)
(2957,)
(741, 20700)
(741,)
('X_train shape:', (2957, 1, 69, 300))
('X_test shape:', (741, 1, 69, 300))
(2957, 'train samples')
(741, 'test samples')
Counter({1: 1717, 2: 657, 0: 510, 3: 73})
Counter({1: 446, 2: 150, 0: 127, 3: 18})
```

In [3]:

```
import os
os.environ["THEANO_FLAGS"] = "mode=FAST_RUN,device=gpu,floatX=float32"
import theano

nb_filters = 100
nb_pool = 3
model = Sequential()

model.add(Convolution2D(nb_filters, 3, 3,
                        border_mode='valid',
                        input_shape=(1, 69, 300)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
# model.add(Convolution2D(nb_filters*2, 3, 3))
# model.add(Activation('relu'))
# model.add(MaxPooling2D(pool_size=(2, 2)))
# model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy',
              optimizer='adadelta',
              metrics=['accuracy'])
```

In [4]:

```
import os
os.environ["THEANO_FLAGS"] = "mode=FAST_RUN,device=gpu,floatX=float32"
import theano
```

In [5]:

```
model.fit(X_train, Y_train, batch_size=10, nb_epoch=100,
          verbose=1, validation_data=(X_test, Y_test))
score = model.evaluate(X_test, Y_test, verbose=1)
```

```
Train on 2957 samples, validate on 741 samples
Epoch 1/100
  70/2957 [..............................] - ETA: 622s - loss: 1.
3957 - acc: 0.3571
```

```
---------------------------------------------------------------
----------
KeyboardInterrupt                         Traceback (most recent
call last)
<ipython-input-5-3c111856a6c7> in <module>()
      1 model.fit(X_train, Y_train, batch_size=10, nb_epoch=100,
----> 2           verbose=1, validation_data=(X_test, Y_test))
      3 score = model.evaluate(X_test, Y_test, verbose=1)

/root/anaconda2/lib/python2.7/site-packages/keras/models.pyc in
it(self, x, y, batch_size, nb_epoch, verbose, callbacks, validat
on_split, validation_data, shuffle, class_weight, sample_weight,
**kwargs)
    618                                 shuffle=shuffle,
    619                                 class_weight=class_weight,
--> 620                                 sample_weight=sample_weigh
t)
    621
    622     def evaluate(self, x, y, batch_size=32, verbose=1,

/root/anaconda2/lib/python2.7/site-packages/keras/engine/trainin
g.pyc in fit(self, x, y, batch_size, nb_epoch, verbose, callback
s, validation_split, validation_data, shuffle, class_weight, sam
le_weight)
   1102                                 verbose=verbose, callbacks
callbacks,
   1103                                 val_f=val_f, val_ins=val_i
s, shuffle=shuffle,
-> 1104                                 callback_metrics=callback_
etrics)
   1105
   1106     def evaluate(self, x, y, batch_size=32, verbose=1, s
mple_weight=None):

/root/anaconda2/lib/python2.7/site-packages/keras/engine/trainin
g.pyc in _fit_loop(self, f, ins, out_labels, batch_size, nb_epoc
h, verbose, callbacks, val_f, val_ins, shuffle, callback_metrics
    820                 batch_logs['size'] = len(batch_ids)
    821                 callbacks.on_batch_begin(batch_index, ba
ch_logs)
--> 822                 outs = f(ins_batch)
    823                 if type(outs) != list:
    824                     outs = [outs]

/root/anaconda2/lib/python2.7/site-packages/keras/backend/theano
backend.pyc in __call__(self, inputs)
    670     def __call__(self, inputs):
```

```
    671         assert type(inputs) in {list, tuple}
--> 672         return self.function(*inputs)
    673
    674

/root/anaconda2/lib/python2.7/site-packages/theano/compile/funct
on_module.pyc in __call__(self, *args, **kwargs)
    857         t0_fn = time.time()
    858         try:
```

In [ ]:

```
from keras.layers import merge, Convolution2D, MaxPooling2D, Input

input_img = Input(shape=(1, 69, 300))

tower_1 = Convolution2D(64, 1, 1, border_mode='same', activation='relu')(input
tower_1 = Convolution2D(64, 3, 3, border_mode='same', activation='relu')(tower

tower_2 = Convolution2D(64, 1, 1, border_mode='same', activation='relu')(input
tower_2 = Convolution2D(64, 5, 5, border_mode='same', activation='relu')(tower

tower_3 = MaxPooling2D((3, 3), strides=(1, 1), border_mode='same')(input_img)
tower_3 = Convolution2D(64, 1, 1, border_mode='same', activation='relu')(tower

output = merge([tower_1, tower_2, tower_3], mode='concat', concat_axis=1)
```