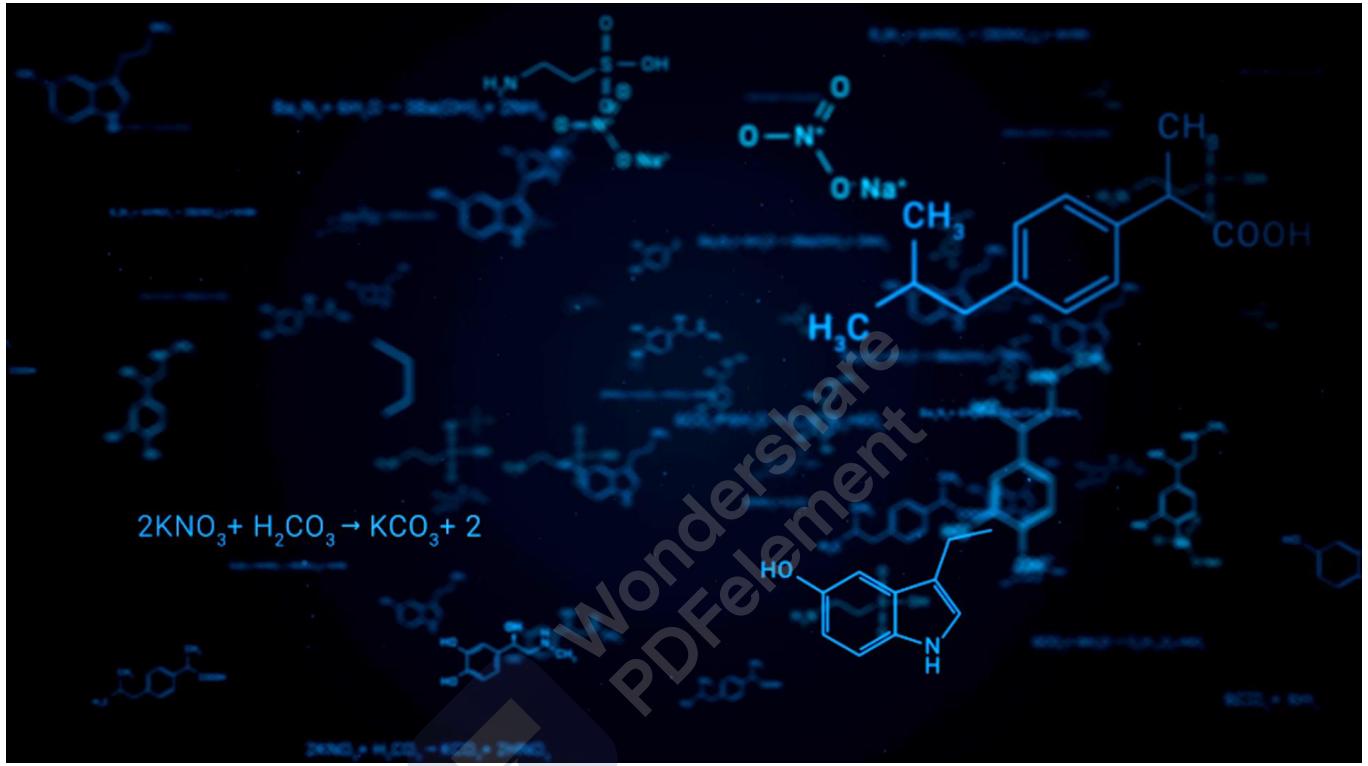

Modulation Classification

MOHAMED SAIED

Problem : A synthetic dataset, generated with GNU Radio, consisting of 11 modulations. This is a variable-SNR dataset with moderate LO drift, light fading, and numerous different labeled SNR increments for use in measuring performance across different signal and noise power scenarios.

Dataset : http://opendata.deepsig.io/datasets/2016.10/RML2016.10b.tar.bz2

Feature Space Data : due to limited vram I made 6 feature spaces (raw / integration / diff / raw&int / raw&diff / diff&int)

Models made: CNN,RNN,LSTM

CODE :-

- *started with tensors configurations and assign vram.*
- *splitted dictionary to values & keys and then plotted some samples*
- *Encoded the modulations (10 classes)*
- *Then some helpful functions (add_integral , add_der , evaluate ,etc …)*
- *Started with rnn model with 6 features space then lstm and cnn at the end*

Notes : due to limited time and I don't have team , I could have made more epochs << better accuracy and results but I had shortage in a time so sorry about that

Notes :-

- *due to limited time and I don't have team , I could have made more epochs << better accuracy and results but I had shortage in a time so sorry about that*
- *All weights of models are saved in the same folder so you can load and use them if you want to check something*
- *There is a readme file is important*

configurations

```
.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'       #verbose INFO, WARNING, and ERROR messages are not printed
config = ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.75      #3G vram
session = InteractiveSession(config=config)
session.close()                          #to release resourses held by other interactive sessions
physical_devices = tf.config.list_physical_devices('GPU')

#If memory growth is enabled for a PhysicalDevice, the runtime initialization will not allocate all memory on the device. Memory growth can\not be configured on a
PhysicalDevice with virtual devices configured.
try:
  tf.config.experimental.set_memory_growth(physical_devices[0], True)
except:
  # Invalid device or cannot modify virtual devices once initialized.
  pass
```


split

```
open_file = open("RML2016.10b.dat",'rb')
data = pickle.load(open_file, encoding='latin1')
x=np.array(list(data.values()),dtype=np.float32)       #values     #(200, 6000, 2, 128)
y=np.array(list(data.keys()))                    #keys tubles (mod,snr)      #(200, 2)
```

Encoder:

```
le = preprocessing.LabelEncoder()   #encoder
snr=y[:,1:]            #separate snr
snr=np.ravel(snr)      ##
snr_classes=np.unique(snr)
y=np.delete(y, 1, 1)   ##drop snr
classes=np.unique(y)
y=np.ravel(y)      ##
y=le.fit_transform(y)   ##encode mods
```

Aux functions :


```
ef evaluate(x):
    results = model.predict(x)
    return results
#%%
def  add_der(x):          #add derivative
    features=np.zeros((x.shape[0],x.shape[1]+2,128),dtype=np.float32)
    #print("features :",features.shape)
    for i in range(x.shape[0]):
        diff=np.diff(x[i][0:2,:])
        c=np.empty((2,1))
        c[0][0]=diff[0][126]
        c[1][0]=diff[1][126]
        diff=np.append(diff,c,axis= 1)
        features[i]=np.concatenate((x[i],diff), axis=0)
    x=None
    x=features
    features=None
    return x
#%%
def  add_integral(x):          #add integration
    from scipy import integrate
    features=np.zeros((x.shape[0],x.shape[1]+2,128),dtype=np.float32)
    #print("features :",features.shape)
    for i in range(x.shape[0]):
        integral=integrate.cumtrapz(x[i][0:2], initial=0)
        features[i]=np.concatenate((x[i],integral), axis=0)
    x=None
    x=features
    features=None
    return x
```

Models in ipynb file: