Using Specialty Layers and Functions

Jerry Kurata
CONSULTANT

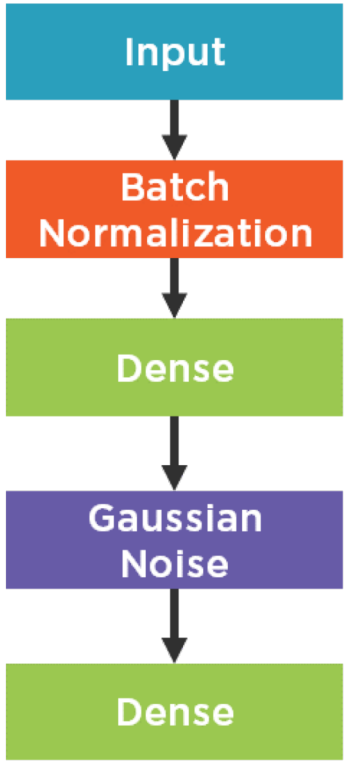
@jerrykur

PreProcessing

pad_sequence()

skipgrams()

make_sampling_table()



Normalization Layer

- Batch Normalization

Noise Layers

- Gaussian Noise
- Gaussian Dropout
- Alpha Dropout

Keras Datasets

Save time

Standards

Reduces your workload

Keras Datasets

Image	MNIST Fashion MNIST CIFAR 10 CIFAR 100
Text	IMDB – Sentiment Analysis Reuters – Topic Classification
Regression	Boston Housing Price

Keras Pre-trained Models

Image Classification

Use directly for trained classes

Retrained through Transfer Learning

Optimized via Fine-Tuning

Pre-trained Image Models

Model	Size	Parameters	Depth
Xception	88 MB	22,910,480	126
VGG16	528 MB	138,357,544	23
VGG19	549 MB	143,667,240	26
ResNet50	99 MB	25,636,712	168
InceptionV3	92 MB	23,851,784	159
InceptionResNetV2	215 MB	55,873,736	572
MobileNet	17 MB	4,253,864	88
DenseNet121	33 MB	8,062,504	121
DenseNet169	57 MB	14,307,880	169
DenseNet201	80 MB	20,242,984	201

Functional API – A Second Look

```
cm = Conv2D(128,(4,4))
cm = MaxPool()(cm)
cm = Conv2D(64,(4,4))(cm)
cm = MaxPool()(cm)
```

◀ Inline definition

```
def conmax(f, k):
    x = Conv2D(f,
           kernel_size=k)
    x = MaxPool()(x)
    return x

cm = conmax(128,(4,4))
cm = conmax(64,(4,4))(cm)

cm = conmax(72,(4,4))(cm)
```

◀ Function definition

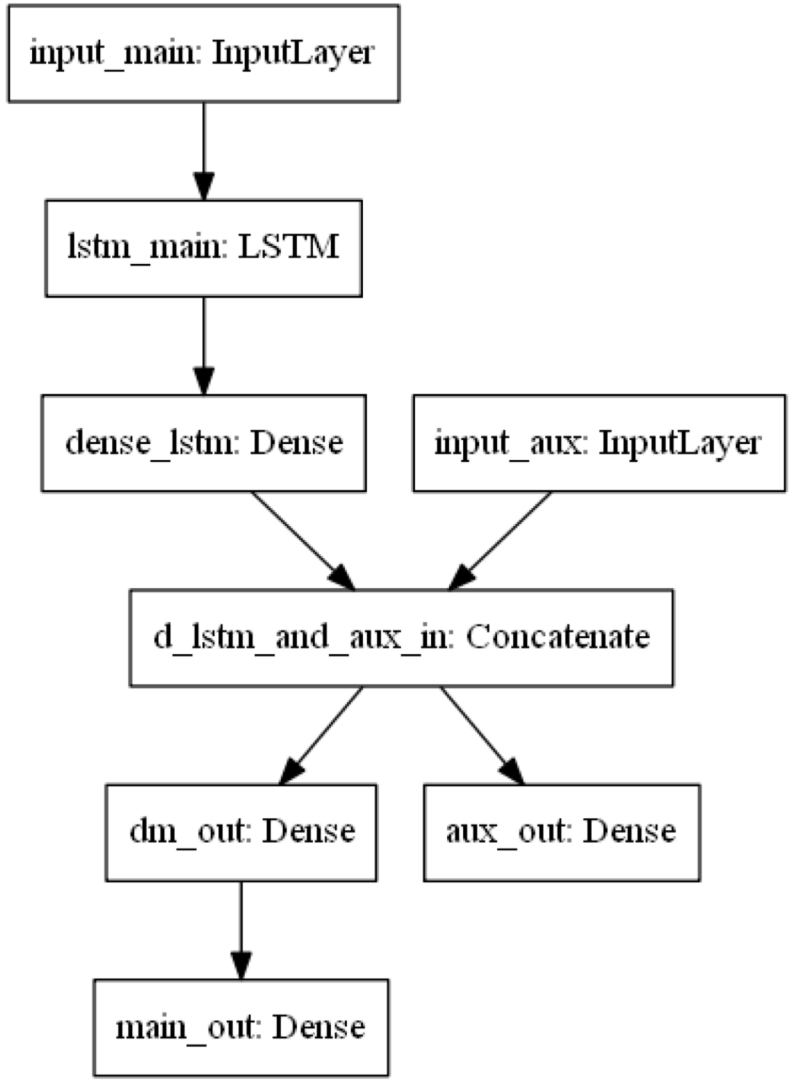
◀ Create Conv and MaxPool via function

```
im = Input(shape=(100,200),
    name='input_main')
ls = LSTM(10,
    name='lstm_main')(im)
dl = Dense(5,
    name='dense_lstm')(ls)
ix = Input(name='input_aux')
cc = concatenate([dl,ix],
    name='d_lstm_and_aux_in')
x = Dense(32,name='dm_out')
        (cc)
mo = Dense(1,name='main_out')(x)
ao = Dense(2,name='aux_out')(cc)

m = Model(inputs=[im,ix],
          outputs=[mo, ao])
```

- Main input
- LSTM main
- Dense LSTM
- Auxiliary input
- Concatenate dense and aux input
- Dense with concatenated
- Main output
- Aux output
- Model with multiple inputs and outputs

Multiple Input and Multiple Output Model



Summary

- Pre-processing
- Normalization and Noise layers
- Included datasets
- Pre-trained models
- More on Functional API