1 Introduction to Keras

Keras is a high-level neural networks API, written in Python, and capable of running on top of TensorFlow, CNTK, or Theano. It's designed to enable fast experimentation.

1.1 Installation

```
pip install keras
```

1.2 Importing Keras

```
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout
```

2 Building Models

2.1 Sequential Model

```
model = Sequential()
model.add(Dense(64, activation='relu', input_shape=(32,)))
model.add(Dense(10, activation='softmax'))
```

2.2 Functional API

```
from keras.models import Model
from keras.layers import Input, Dense

inputs = Input(shape=(32,))
x = Dense(64, activation='relu')(inputs)
outputs = Dense(10, activation='softmax')(x)

model = Model(inputs=inputs, outputs=outputs)
```

3 Compiling Models

```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

4 Training Models

```
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_split=0.2)
```

5 Evaluating Models

5.1 Model Evaluation

```
loss, accuracy = model.evaluate(X_test, y_test)
```

5.2 Making Predictions

```
predictions = model.predict(X_new)
```

6 Model Regularization

6.1 Dropout Layer

```
model.add(Dropout(0.5))
```

6.2 L2 Regularization

```
from keras.regularizers import l2

model.add(Dense(64, activation='relu', kernel_regularizer=l2(0.01)))
```

7 Advanced Layers

7.1 Convolutional Layers

```
from keras.layers import Conv2D, MaxPooling2D

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

7.2 Recurrent Layers

```
from keras.layers import LSTM

model.add(LSTM(128, activation='tanh'))
```

8 Working with Data

8.1 Image Data

```
from keras.preprocessing.image import ImageDataGenerator

datagen = ImageDataGenerator(rescale=1./255)
train_generator = datagen.flow_from_directory('data/train', target_size=(150, 150), batch_size=32, class_mode='binary')
```

8.2 Text Data

```
from keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer(num_words=10000)
tokenizer.fit_on_texts(texts)
sequences = tokenizer.texts_to_sequences(texts)
```

9 Callbacks

9.1 Early Stopping

```
from keras.callbacks import EarlyStopping

early_stopping = EarlyStopping(monitor='val_loss', patience=3)
model.fit(X_train, y_train, epochs=10, callbacks=[early_stopping])
```

9.2 TensorBoard

```
from keras.callbacks import TensorBoard

tensorboard = TensorBoard(log_dir='logs')
model.fit(X_train, y_train, epochs=10, callbacks=[tensorboard])
```

10 Saving and Loading Models

10.1 Saving a Model

```
model.save('model.h5')
```

10.2 Loading a Model

```
from keras.models import load_model

model = load_model('model.h5')
```

11 Model Optimization

11.1 Learning Rate Scheduling

```
from keras.callbacks import LearningRateScheduler

def scheduler(epoch, lr):
    return lr * 0.1

lr_scheduler = LearningRateScheduler(scheduler)
model.fit(X_train, y_train, epochs=10, callbacks=[
    lr_scheduler])
```

11.2 Batch Normalization

```
from keras.layers import BatchNormalization

model.add(BatchNormalization())
```

12 Transfer Learning

12.1 Pre-trained Models

```
from keras.applications import VGG16

vgg_model = VGG16(weights='imagenet', include_top=
    False, input_shape=(224, 224, 3))
for layer in vgg_model.layers:
    layer.trainable = False

model.add(vgg_model)
model.add(Dense(256, activation='relu'))
model.add(Dense(10, activation='softmax'))
```

13 GPU Utilization

13.1 Enabling GPU Support

```
import tensorflow as tf

physical_devices = tf.config.list_physical_devices
    ('GPU')
if physical_devices:
    tf.config.experimental.set_memory_growth(
    physical_devices[0], True)
```

13.2 Mixed Precision Training

```
from keras.mixed_precision import experimental as
    mixed_precision

policy = mixed_precision.Policy('mixed_float16')
mixed_precision.set_policy(policy)
```

14 Resources

14.1 Official Documentation

- Keras Documentation

14.2 Courses

- Keras in TensorFlow Course

14.3 Books

- *Deep Learning with Python* by François Chollet