

TENSORFLOW ON IOS

尹航 *h4x@Google*

TENSORFLOW ON IOS

尹航 *h4x@Google*

机器学习框架

Caffe



MINERVA

mxnet

DL4J
Deeplearning4j

K
KERAS

Microsoft
CNTK

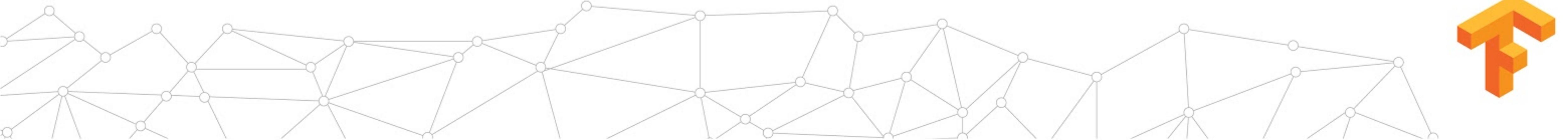
MatConvNet

Purine

The TensorFlow logo features a stylized orange 'F' shape composed of three interlocking arrows, with the word "TensorFlow" in orange and grey below it.

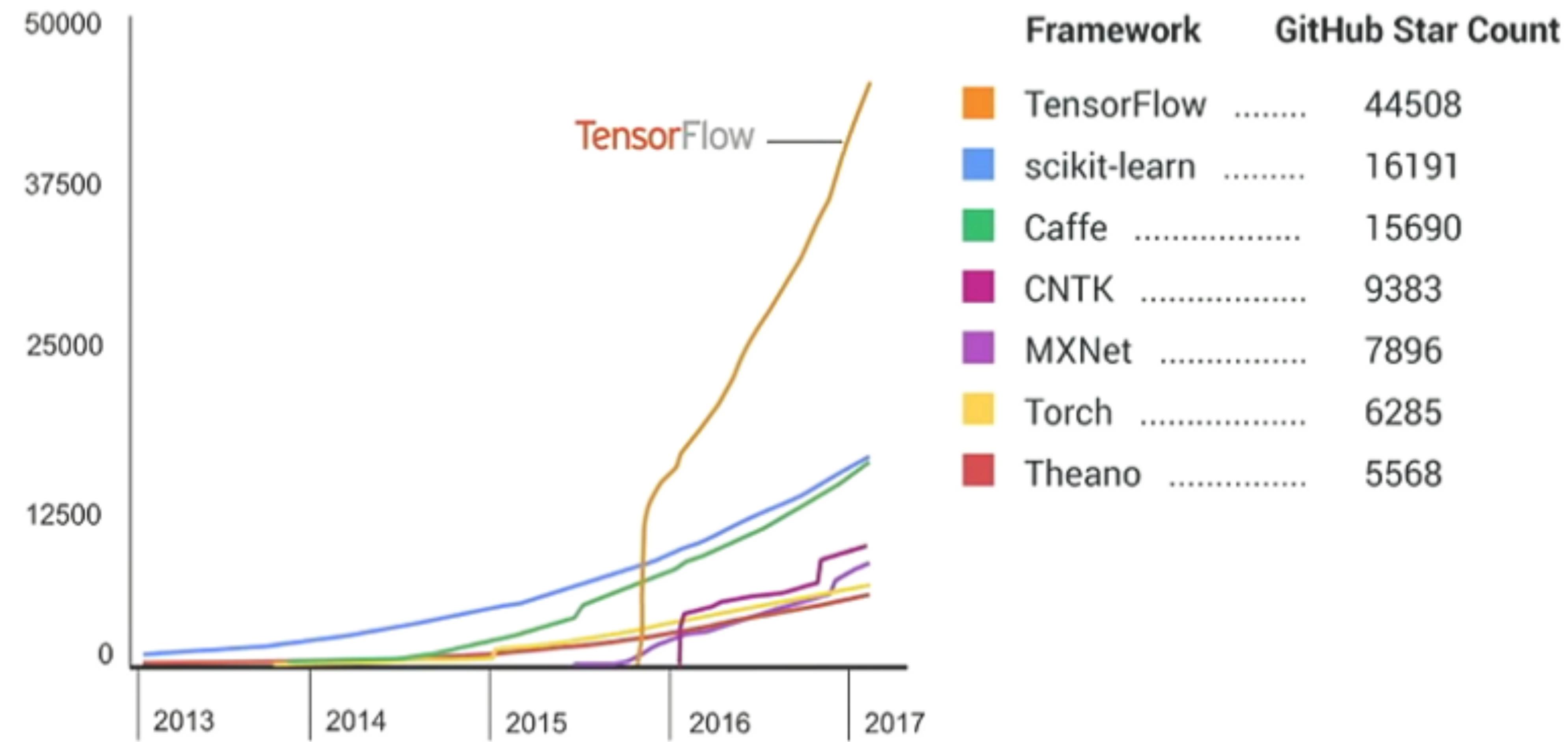
theano

The torch logo is a black stylized molecule or neural network structure, with the word "torch" in a grey sans-serif font to its right.



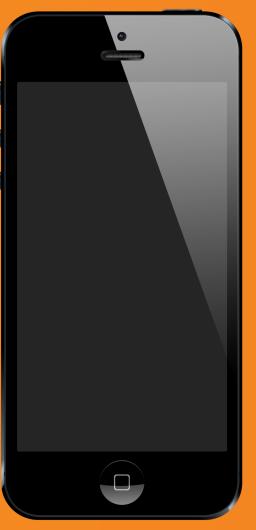
WHY TENSORFLOW?





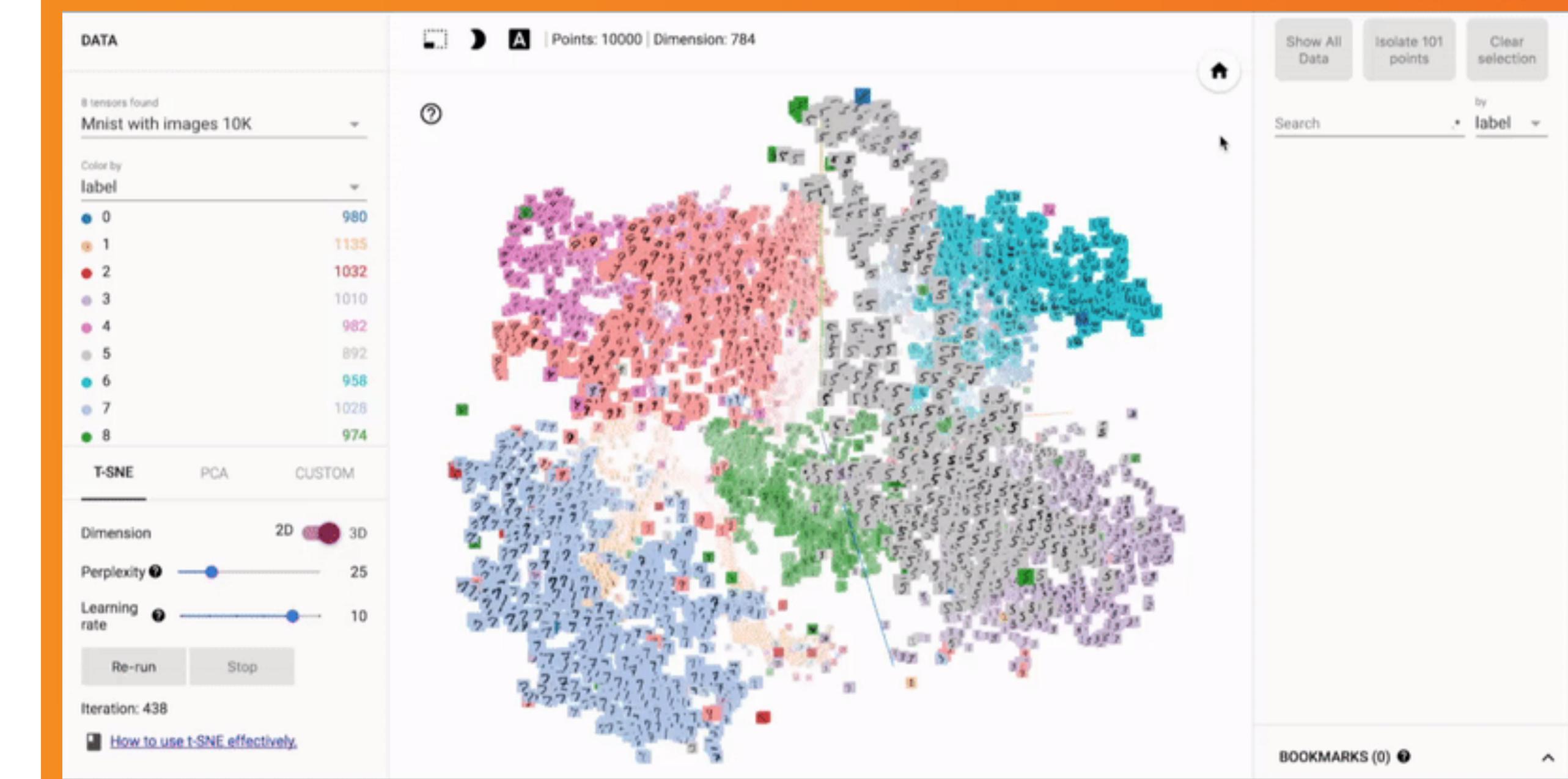
WHY TENSORFLOW

- 全平台支持
- 服务器集群
- GPU、TPU加速
- CPU
- 移动端



WHY TENSORFLOW

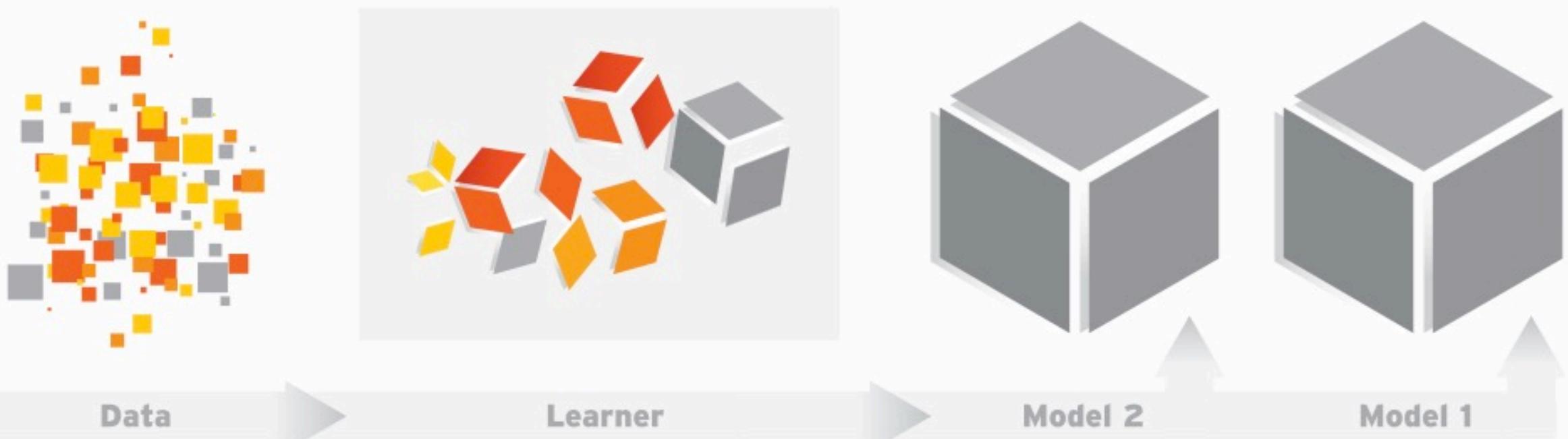
- 全平台支持
- 丰富的调试工具
 - TensorBoard



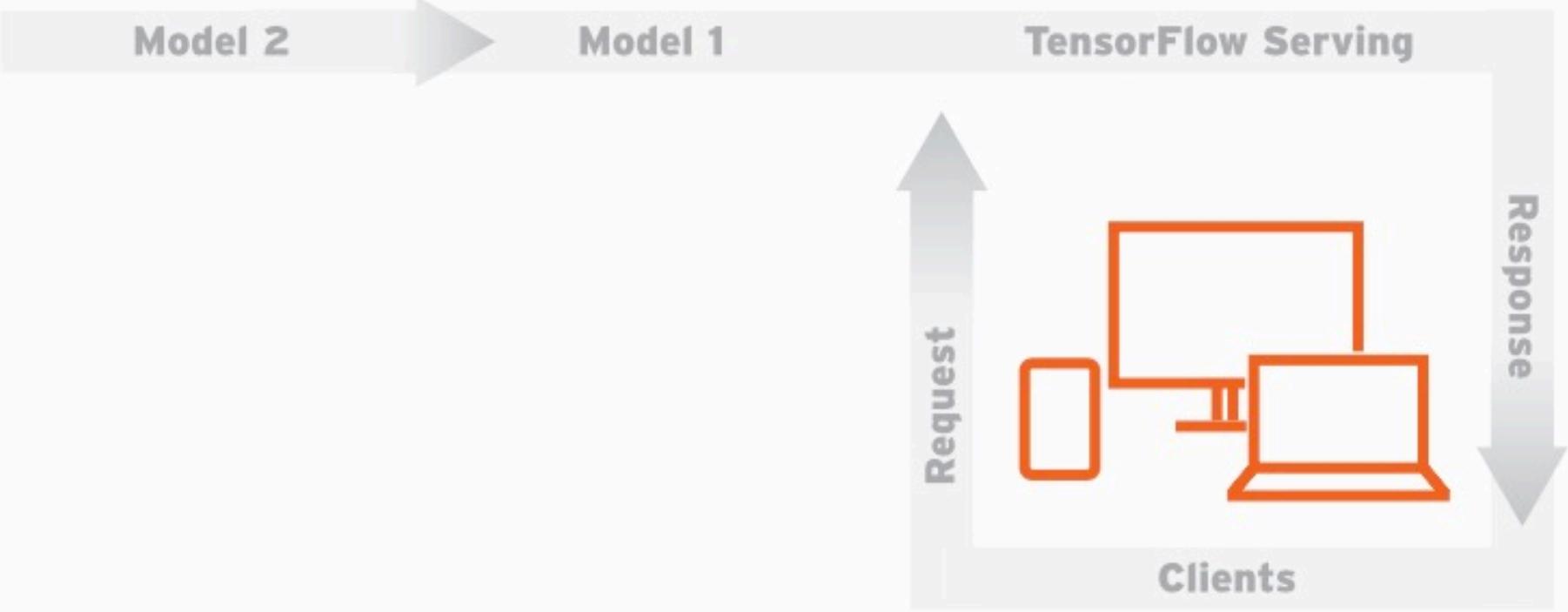
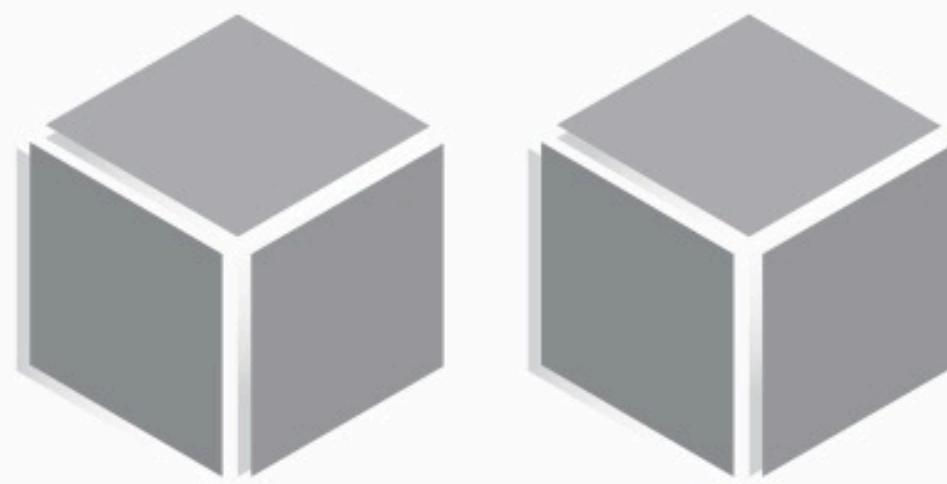
WHY TENSORFLOW

- 全平台支持
- 丰富的调试工具
- 产品化
 - TensorFlow Serving
 - Google Cloud

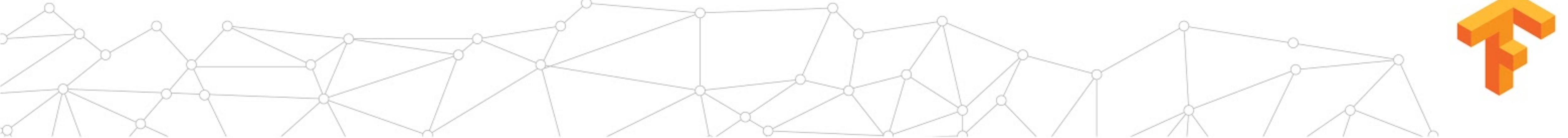
CONTINUOUS TRAINING PIPELINE

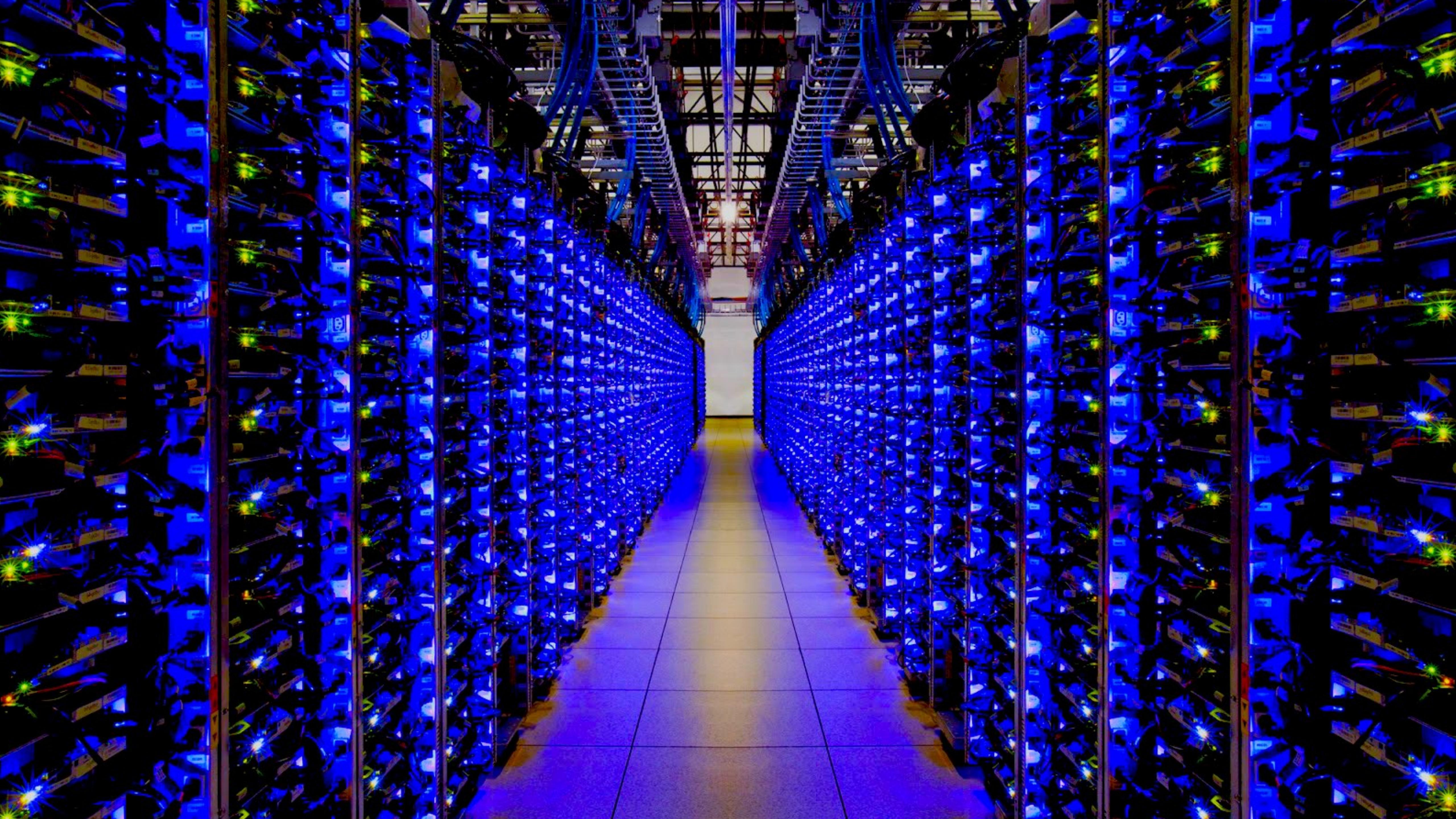


SERVING



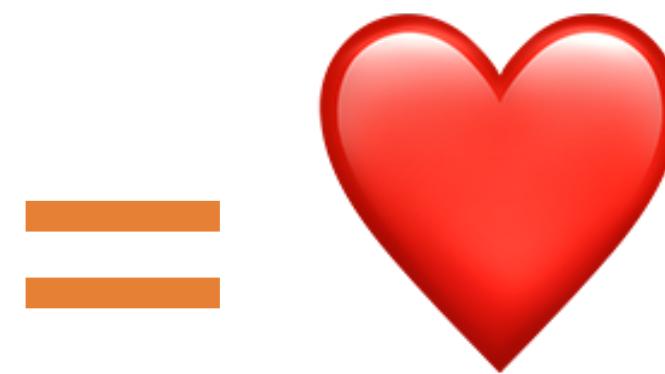
WHY MOBILE?







TENSORFLOW + IOS



A TensorFlow Demo



PROBLEM

- Emoji输入法
 - 输入：一段短文本
 - 输出：预测合适的Emoji

1834 640228491 482315469 461912952 430206917 364
175
145
123
966
802
648
561
472
360
326
270
236
210
184
170
152
141
130
114
100

1834 640228491 482315469 461912952 430206917 364
175
145
123
966
802
648
561
472
360
326
270
236
210
184
170
152
141
130
114
100

 FACE WITH TEARS OF JOY

unified id: 1F602
shorthand: :joy:
popularity: #1

live tweets

- > Pretty much every song on this years #Eurovision has been a rip off of another song. But like... proper blatant! 😂
— Redz@SamRedz92
- > Thought it was Titanium at the start 😂 #ger #Eurovision
— Nikki @_nikkilp
- > @Pandy_RU Lmao hay man bendfuna ukuthi xa befuna iMakeup 😂
— missjuicybaby @molose_mihle
- > S/O to @christiand !! All I have to do is watch me some Christian Delgrosso and my day is 10x better. #WeLoveYouChristian
#&KriSSaaaaalll 😂
— KBre 💪 @Kelseyyyy2016
- > @susannecc @one_mrs_k @gavmacn 😂
— Dorothy Aidulis @Dorothy_Aidulis
- > Snepceti gecen sene indirip hiç bişey anlamayıp silmiştim bu sene tam tersi oldu resmen snepchet delisi oldum 😂

PROBLEM

- Emoji输入法
 - 输入：一段短文本
 - 输出：预测合适的Emoji
- 有没有简单的办法...
 - 比如匹配关键字？

“Happy New Year”



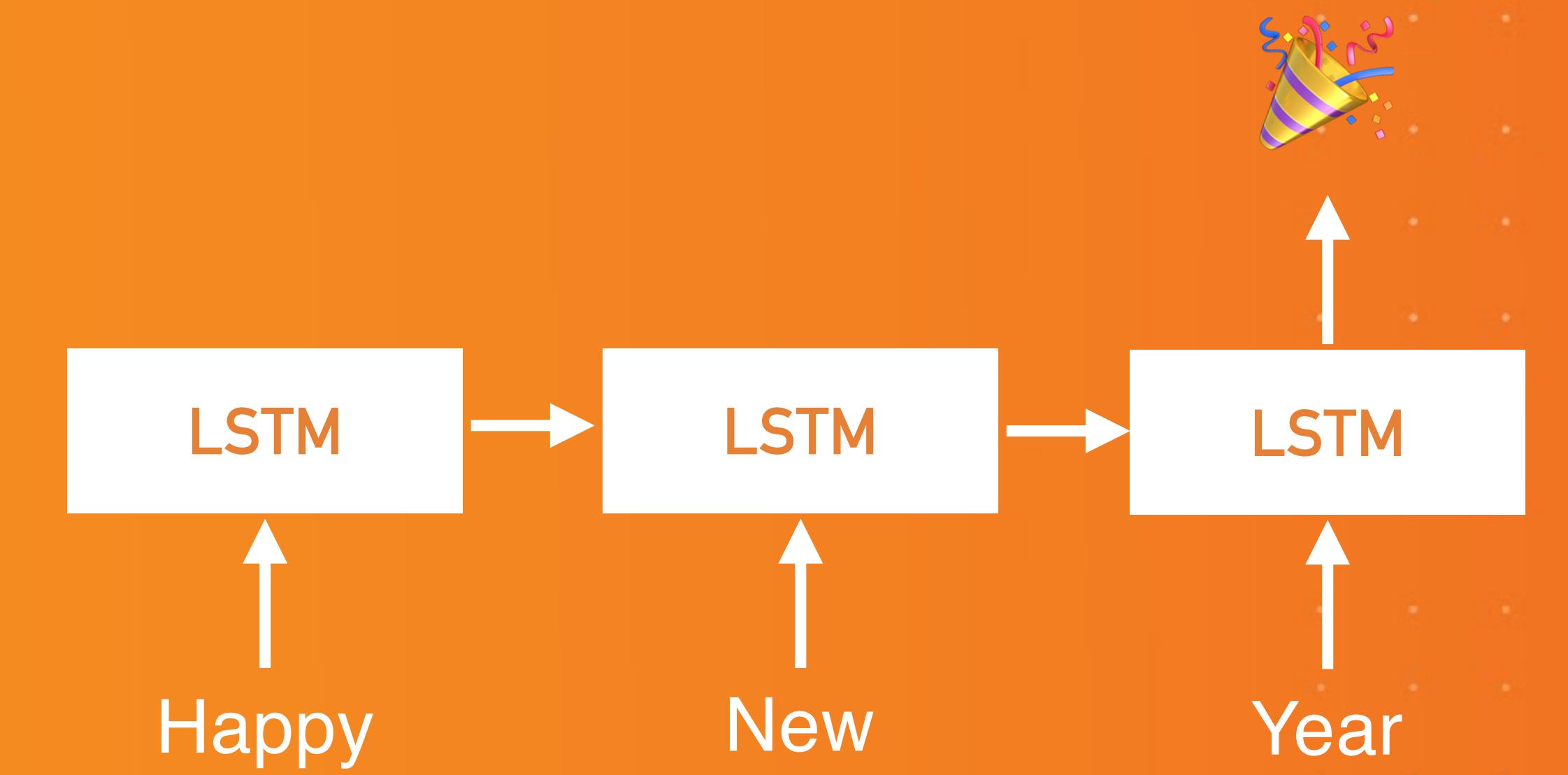
准备数据

- Twitter 2017年1月数据
 - 144字限制
 - 网络语言
- 预处理
 - 统计Top-100 Emoji
 - 100,000条英文推文



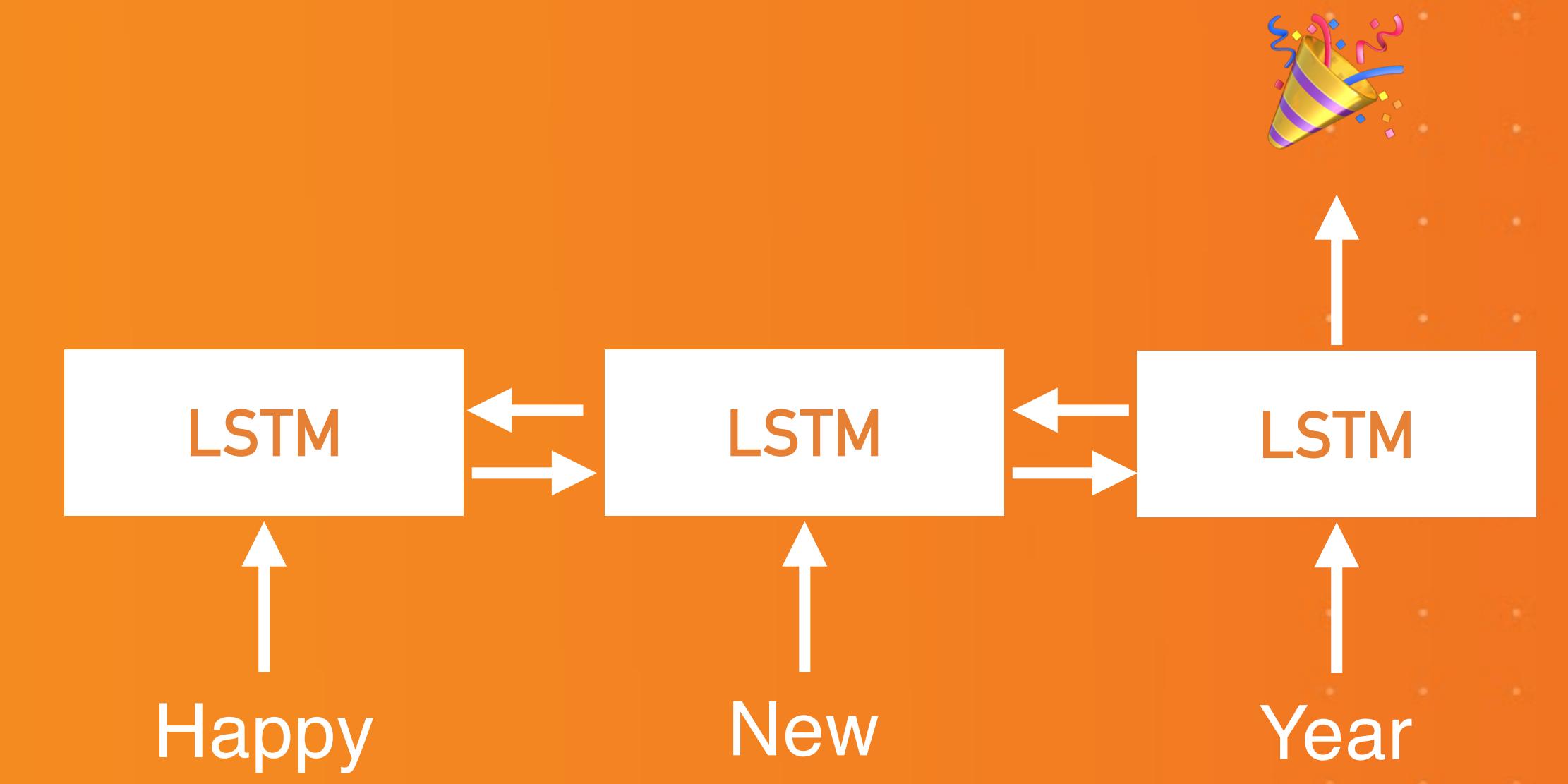
神经网络模型

- 基本思想: RNN
 - 接受任意长输入
 - 取最后一个输出作为结果
- LSTM
 - Long Short Term Memory
 - 一种适合文本的RNN



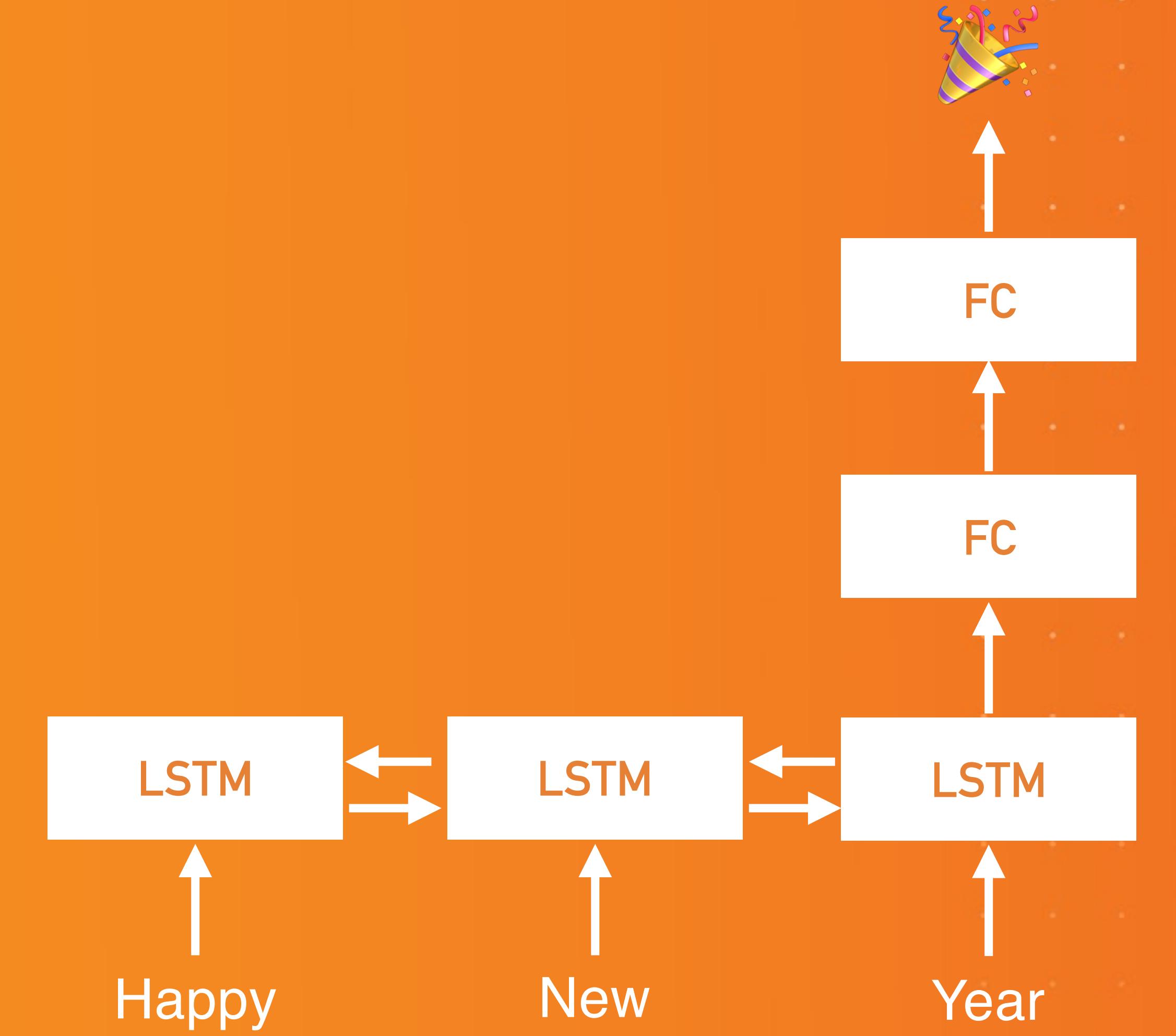
神经网络模型-改进

► 双向LSTM



神经网络模型-改进

- 双向LSTM
- 更深的网络



CHAR-CNN编码器

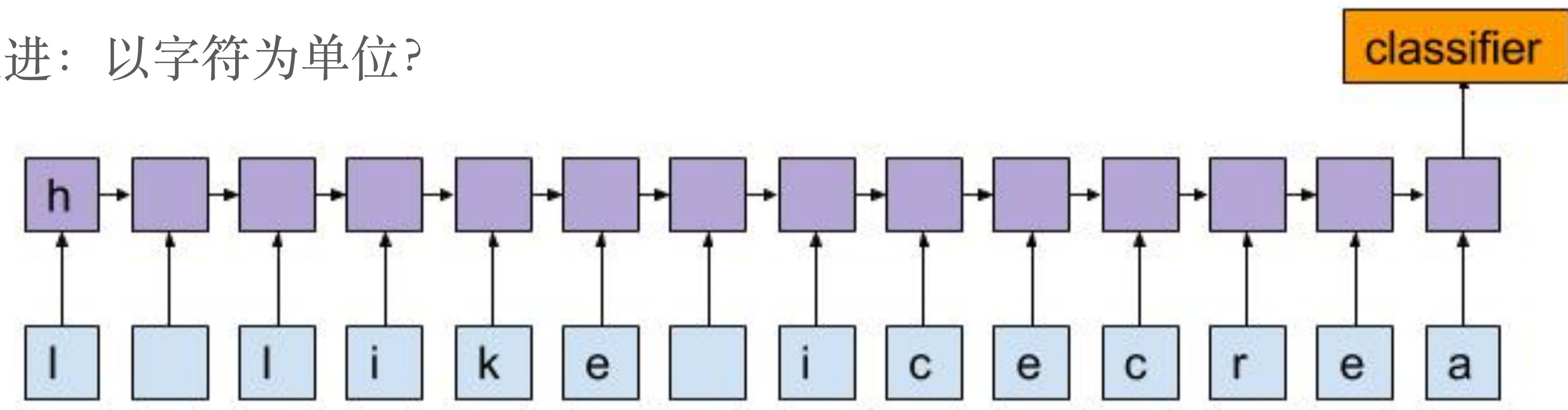
- 以词为单位的问题
- 词典尺寸太大
- 不规范用词：网络用语、拼写错误

$$100,000 \text{ words} * 128 \text{ dimension} * 4 \text{ bytes} = 51.2\text{MB}$$



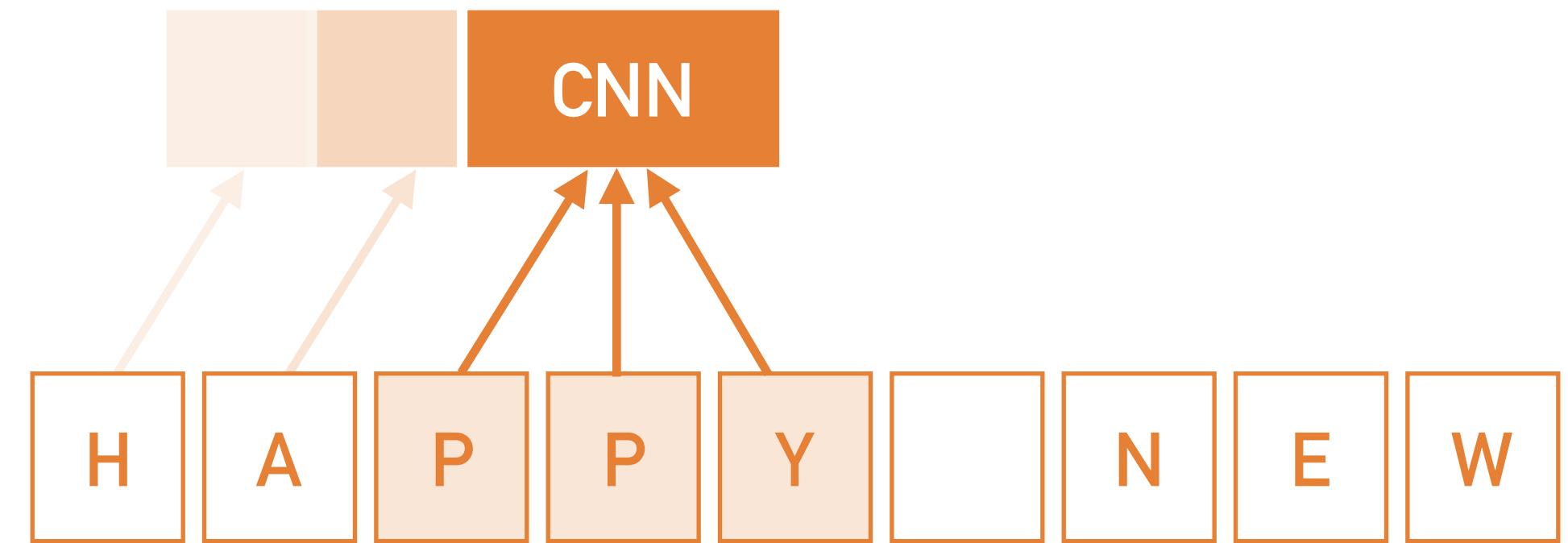
CHAR-CNN编码器

- 以词为单位的问题
- 词典尺寸太大
- 不规范用词：网络用语、拼写错误
- 改进：以字符为单位？



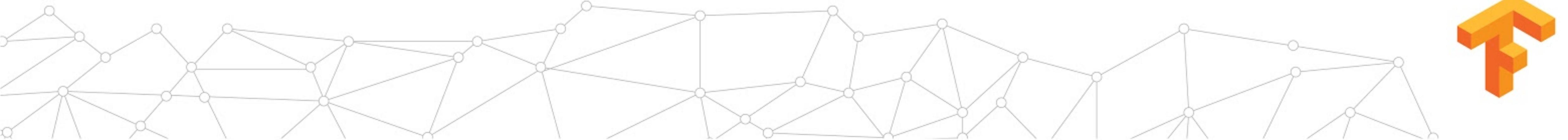
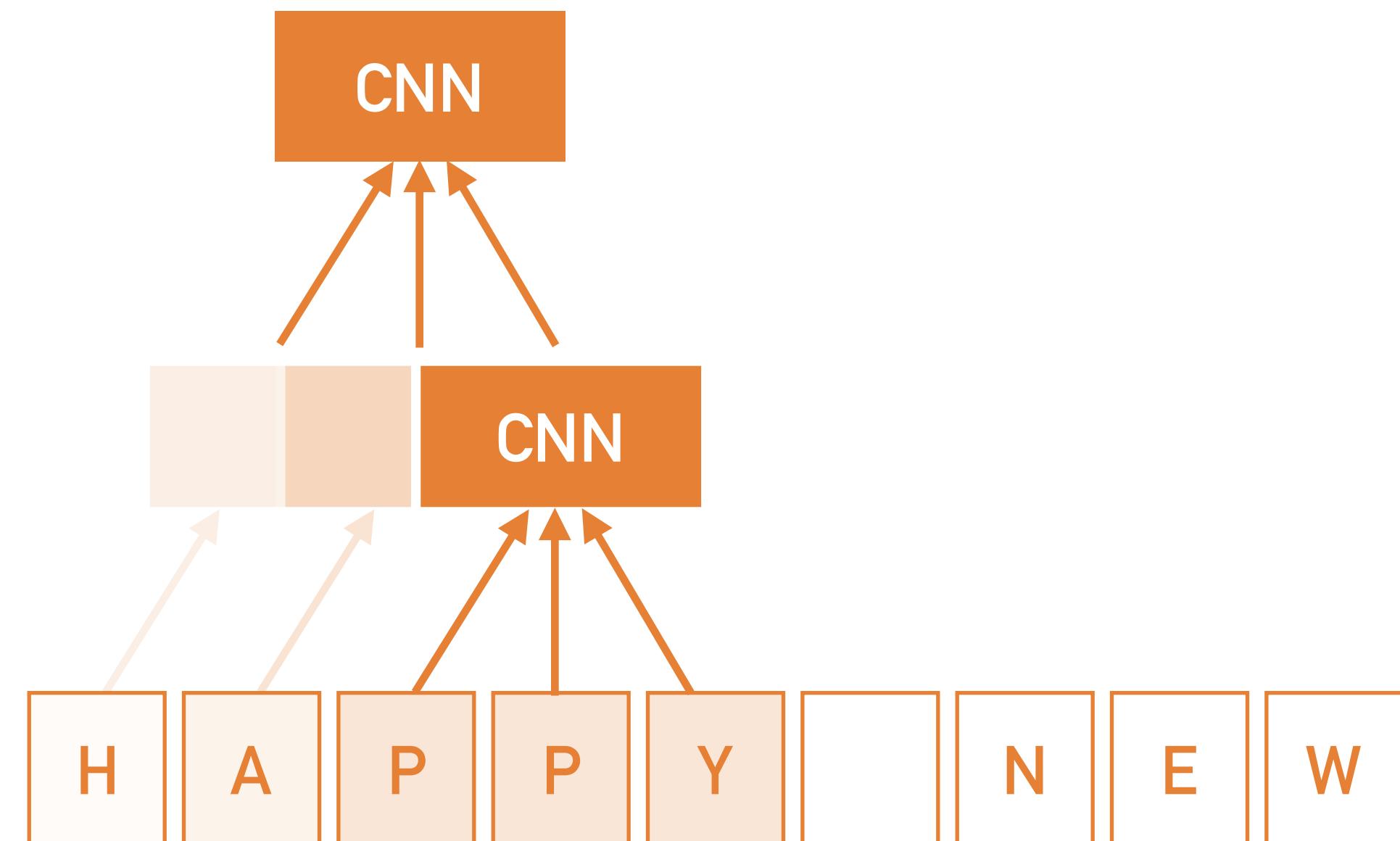
CHAR-CNN编码器

- 以词为单位的问题
- Char-CNN
 - 输入：字母序列
 - 卷积神经网络



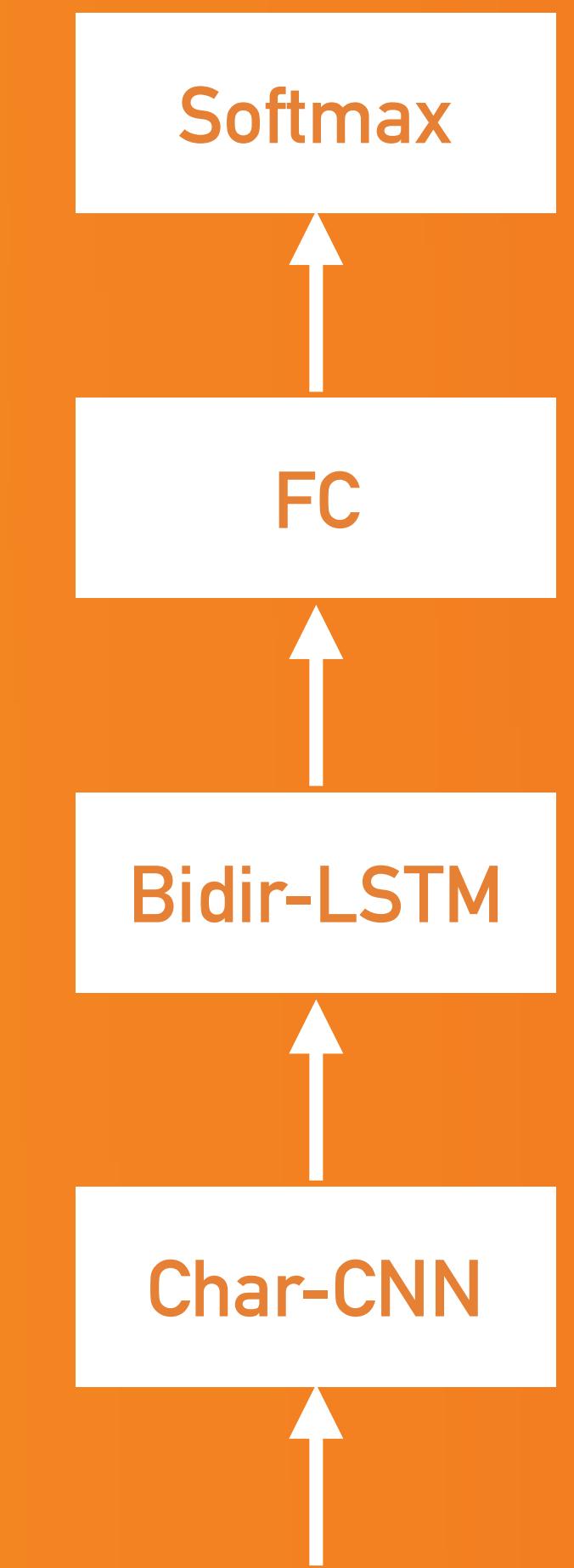
CHAR-CNN编码器

- 以词为单位的问题
- Char-CNN
 - 输入：字母序列
 - 卷积神经网络
 - 多层CNN：从字母到单词



神经网络模型

- 输入：字符序列
- Char-CNN 字符卷积网络
- 双向LSTM
- 隐含层
- 输出：预测Emoji



Happy New Year

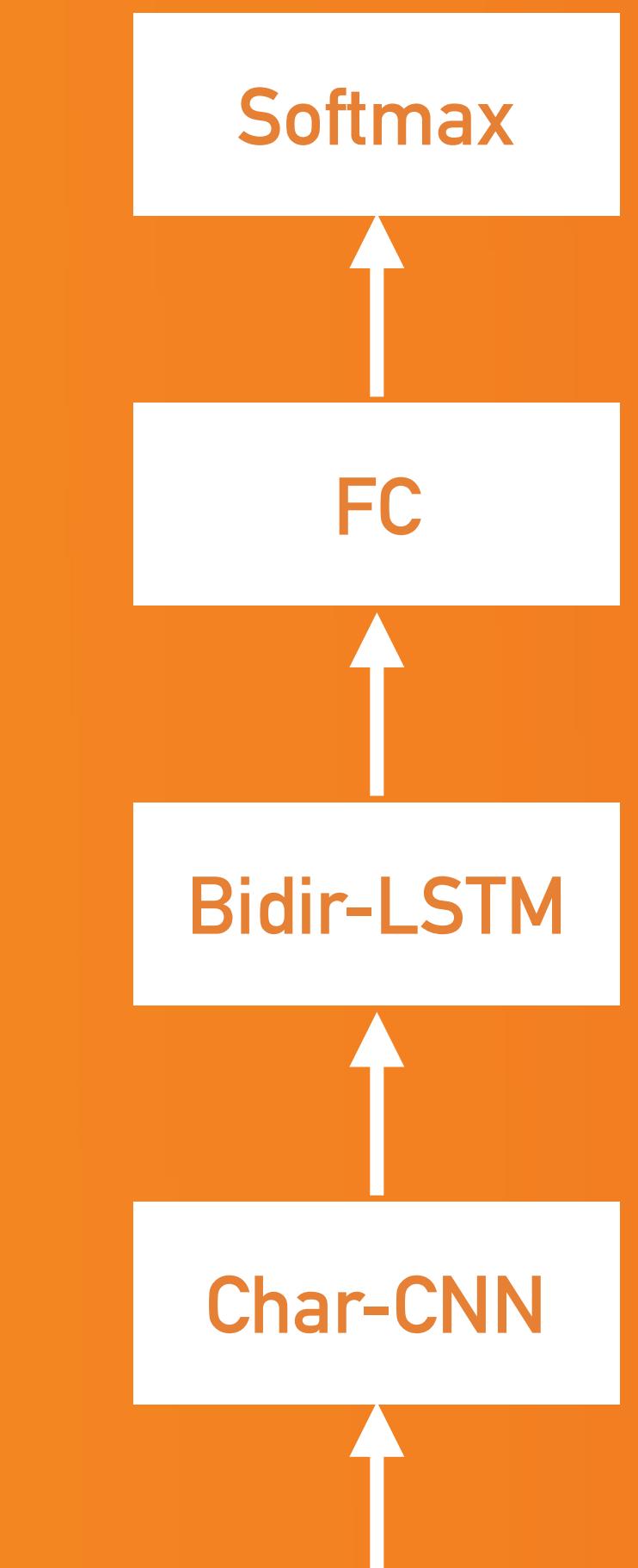


神经网络模型-KERAS实现

- Keras: 一个对人类友好的TensorFlow前端API



<http://keras.io>

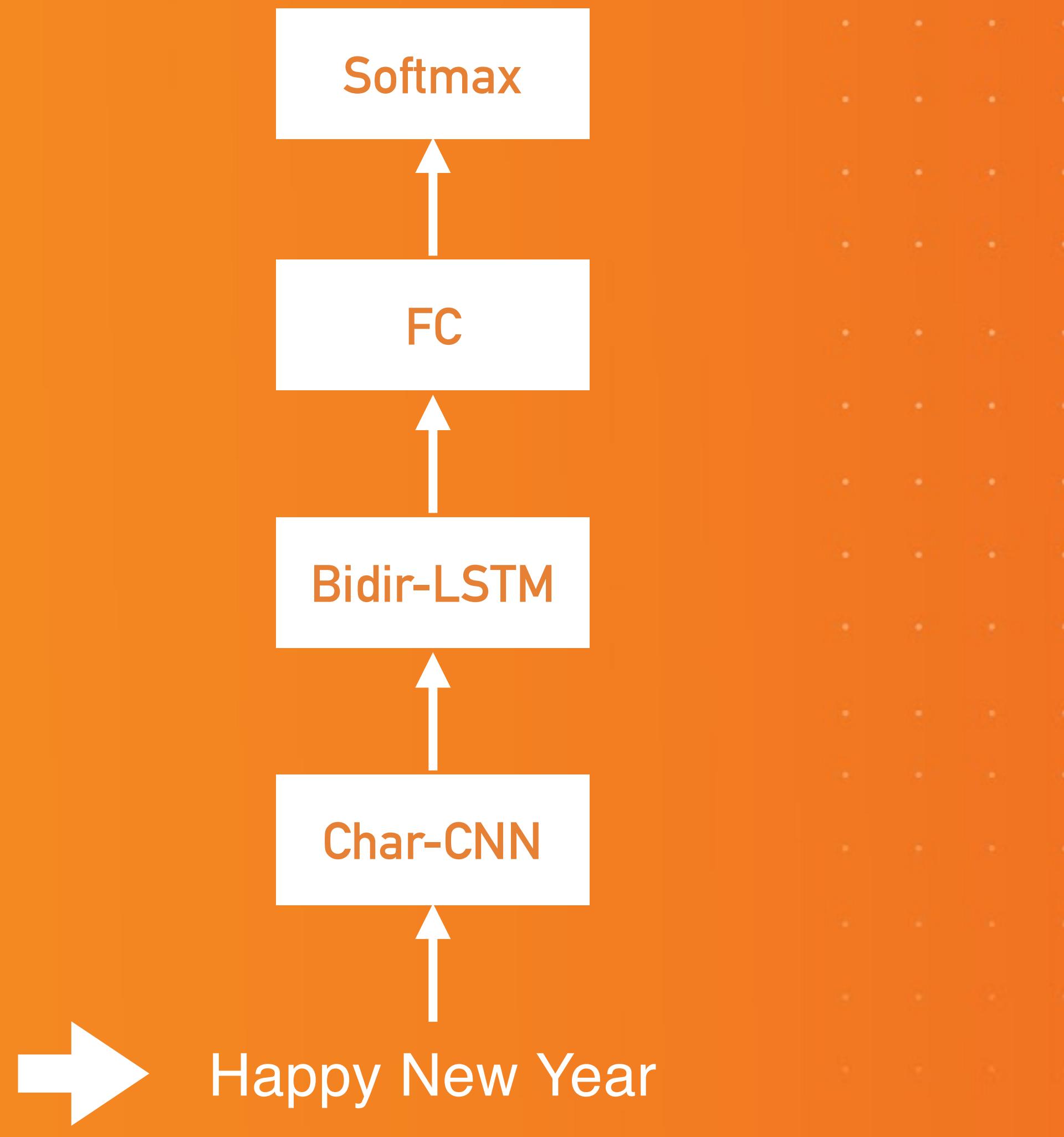


Happy New Year



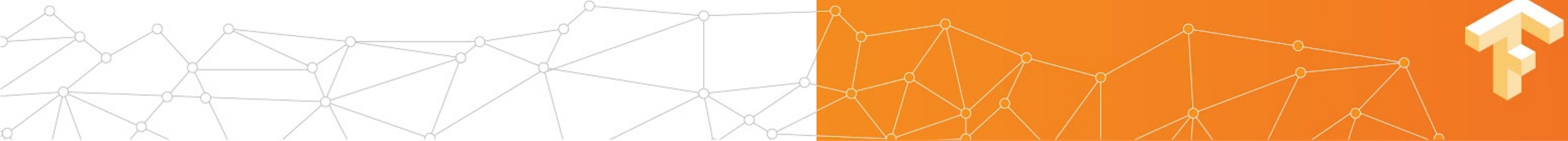
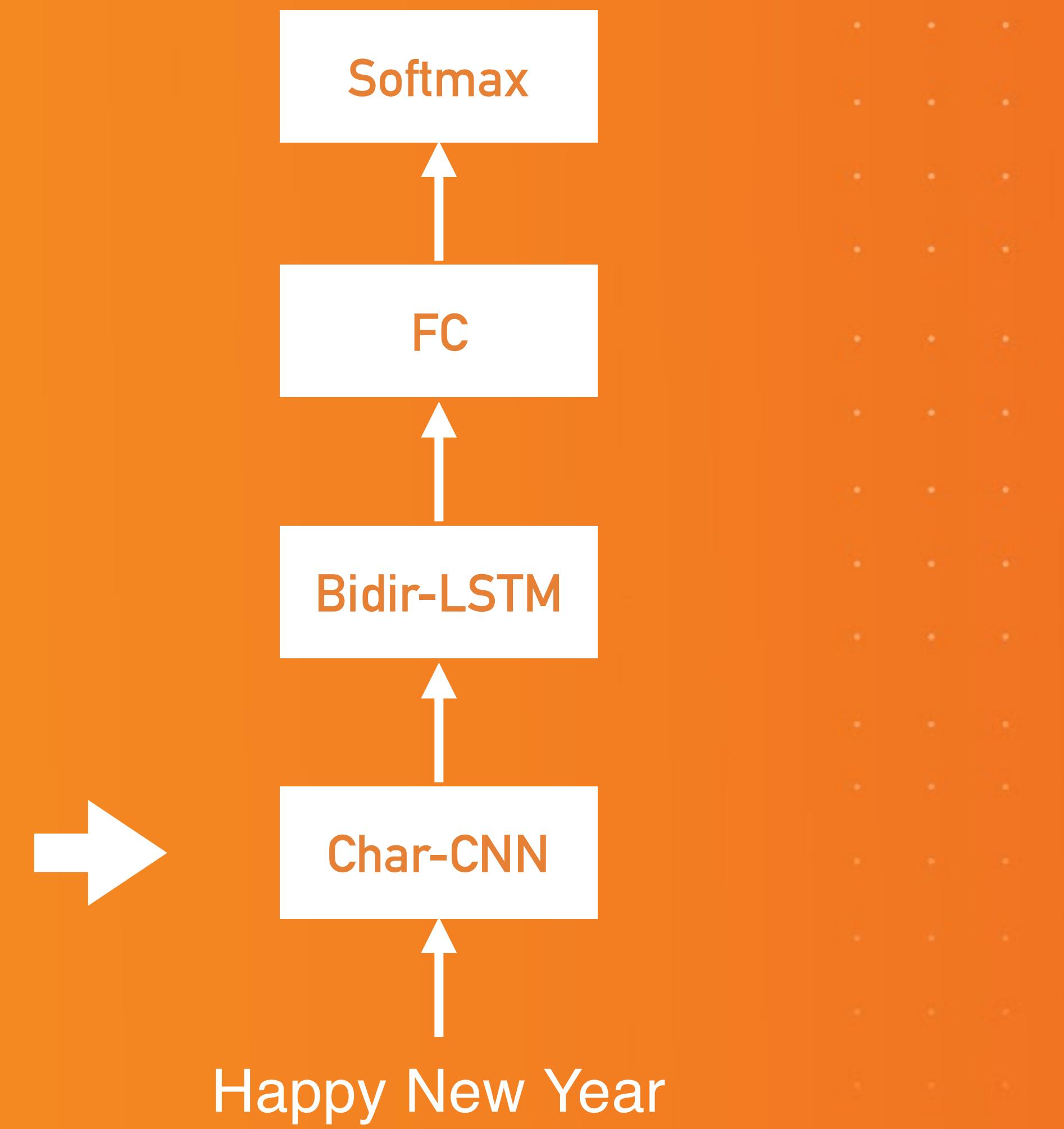
神经网络模型-KERAS实现

```
MAXLEN = 120  
in_sentence = Input(shape=(MAXLEN,), dtype='int32')
```



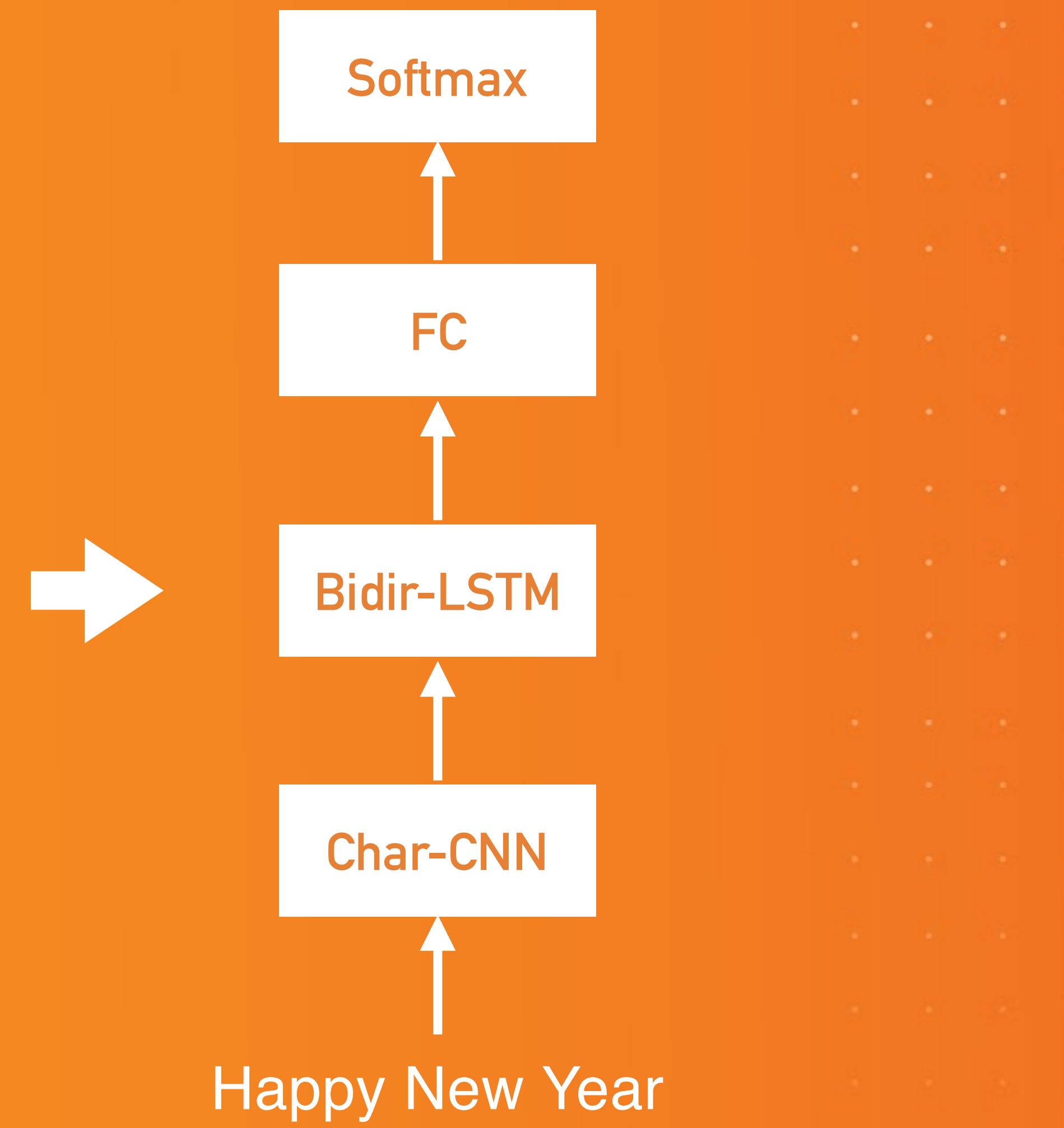
神经网络模型-KERAS实现

```
filter_length = [3, 3, 1]
nb_filter = [196, 196, 256]
pool_length = 2
for i in range(len(nb_filter)):
    embedding = Conv1D(filters=nb_filter[i],
                        kernel_size=filter_length[i],
                        padding='valid',
                        activation='relu',
                        kernel_initializer='glorot_normal',
                        strides=1)(embedding)
embedding = MaxPooling1D(pool_size=pool_length)(embedding)
```



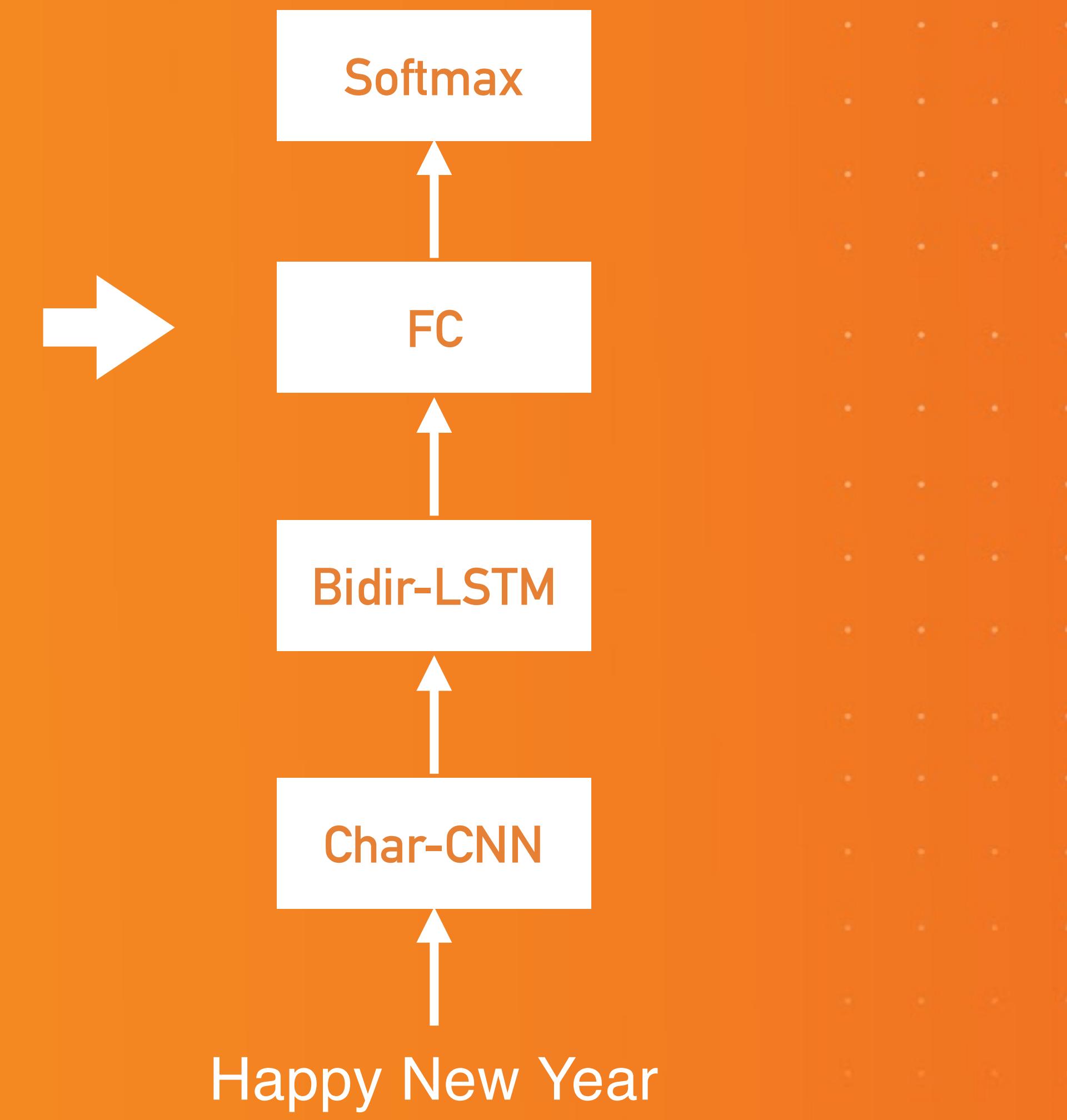
神经网络模型-KERAS实现

```
hidden = Bidirectional(LSTM(  
    128, dropout=0.2, recurrent_dropout=0.2))(embedding)
```



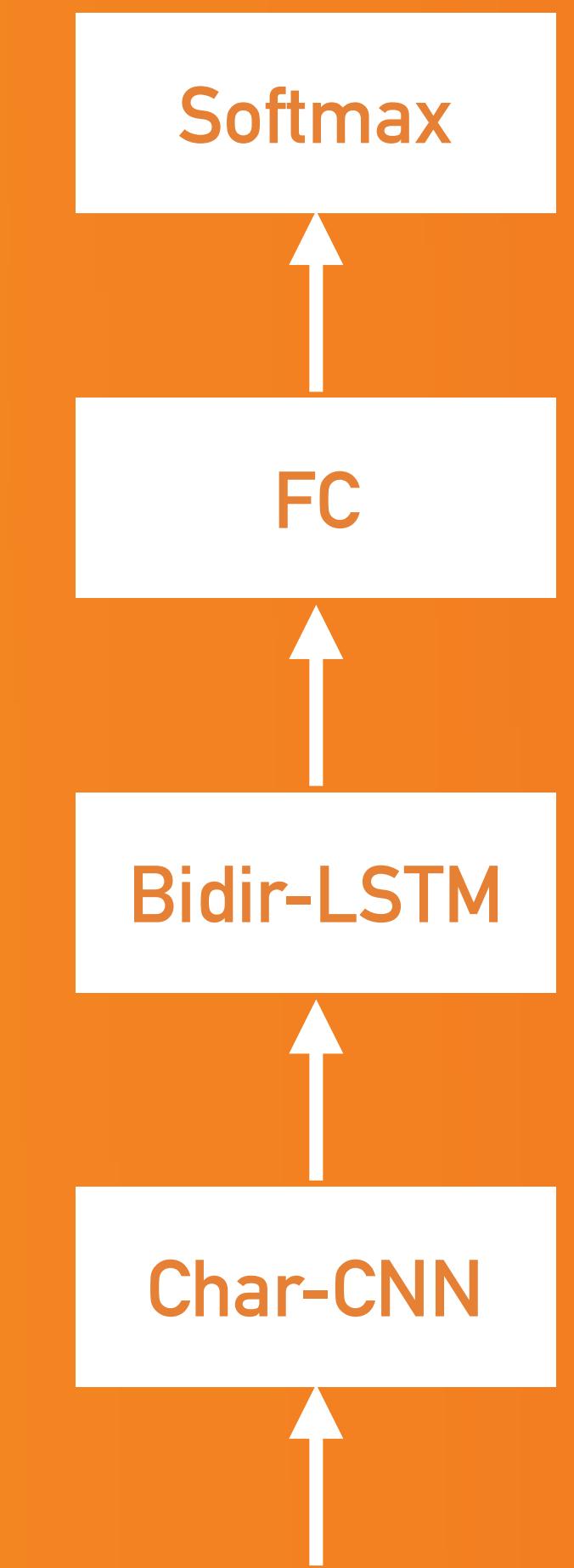
神经网络模型-KERAS实现

```
hidden = Dense(128, activation='relu')(hidden)
hidden = Dropout(0.2)(hidden)
output = Dense(num_cat, activation='softmax')(hidden)
```



神经网络模型-KERAS实现

```
model = Model(inputs=in_sentence, outputs=output)
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy', 'top_k_categorical_accuracy'])
```



Write a regex to create a tag group X

Show data download links

Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing

Horizontal Axis

STEP RELATIVE

WALL

Runs

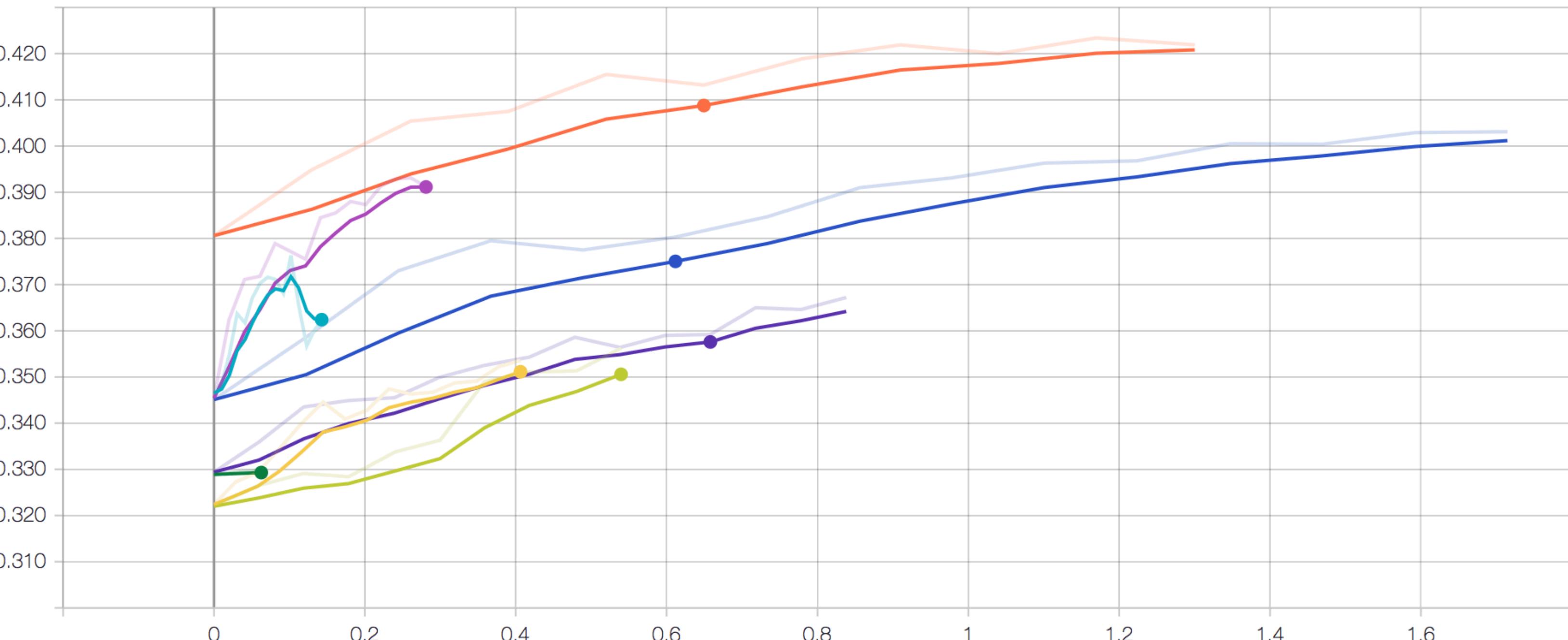
Write a regex to filter runs

- Graph
- tb-chcnn-blstm
- tb-chcnn-blstm-2fc
- tb-chcnn-slim-blstm-2fc
- tb-naive-embed-blstm
- tb-naive-noembd
- tb-naive-noembd-2lstm
- tb-naive-noembd-blstm



val_top_k_categorical_accuracy

val_top_k_categorical_accuracy



Graph

Smoothed Value Step Time

Relative

0.4088 0.4132 5.000 Sat May 6, 23:29:13 38m 57s

```
In [ ]: from keras.models import Model, load_model, model_from_config
from keras import backend as K
from tensorflow.contrib.session_bundle import exporter
from tensorflow.python import saved_model
import tensorflow as tf
```

```
In [ ]: sess = tf.Session()
K.set_session(sess)
K.set_learning_phase(0) # all new operations will be in test mode from now on
```

```
In [ ]: orig_model = load_model('p5-40-test.hdf5')
weights = orig_model.get_weights()
model = model_from_config({
    'class_name': 'Model',
    'config': orig_model.get_config(),
})
model.set_weights(weights)
```

```
In [ ]: tf.train.write_graph(sess.graph_def, 'export/p5-40-test-serving', "graph-serving.pb", True)
```

```
In [ ]: saver = tf.train.Saver()
```

```
In [ ]: saver.save(sess, 'export/p5-40-test-serving/model-ckpt')
```

```
In [21]: run('I wanna go home and go to sleep')
```

```
Out[21]: ['😭', '😩', '👀', '😂', '😴']
```

```
In [22]: run('happy new year! God Bless')
```

```
Out[22]: ['🎉', '❤️', '🎈', '😊', '😘']
```

```
In [23]: run('HAPPY NEW YEAR here\\'s to many more amazing memories')
```

```
Out[23]: ['🎉', '❤️', '😘', '😊', '💕']
```

```
In [24]: run('day 1 of 365 thank you God for allowing me to see this day')
```

```
Out[24]: ['❤️', '🙏', '😊', '😘', '💕']
```

```
In [26]: run('The art of knowing is knowing to "IGNORE". Good morning')
```

```
Out[26]: ['❤️', '😊', '😍', '😘', '💕']
```

MOVE TO IOS



HOW TO

- 编译TensorFlow for iOS

```
tensorflow/contrib/makefile/  
build_all_ios.sh
```



HOW TO

- 编译TensorFlow for iOS
- 转换模型
 - 裁剪模型
 - 压缩权值 (Quantization)

```
python3 -m tensorflow.python.tools  
freeze_graph \  
--input_graph="graph-serving.pb" \  
--input_checkpoint="model.ckpt" \  
--output_graph="frozen.pb" \  
--output_node_names="dense_2/Softmax"
```



HOW TO

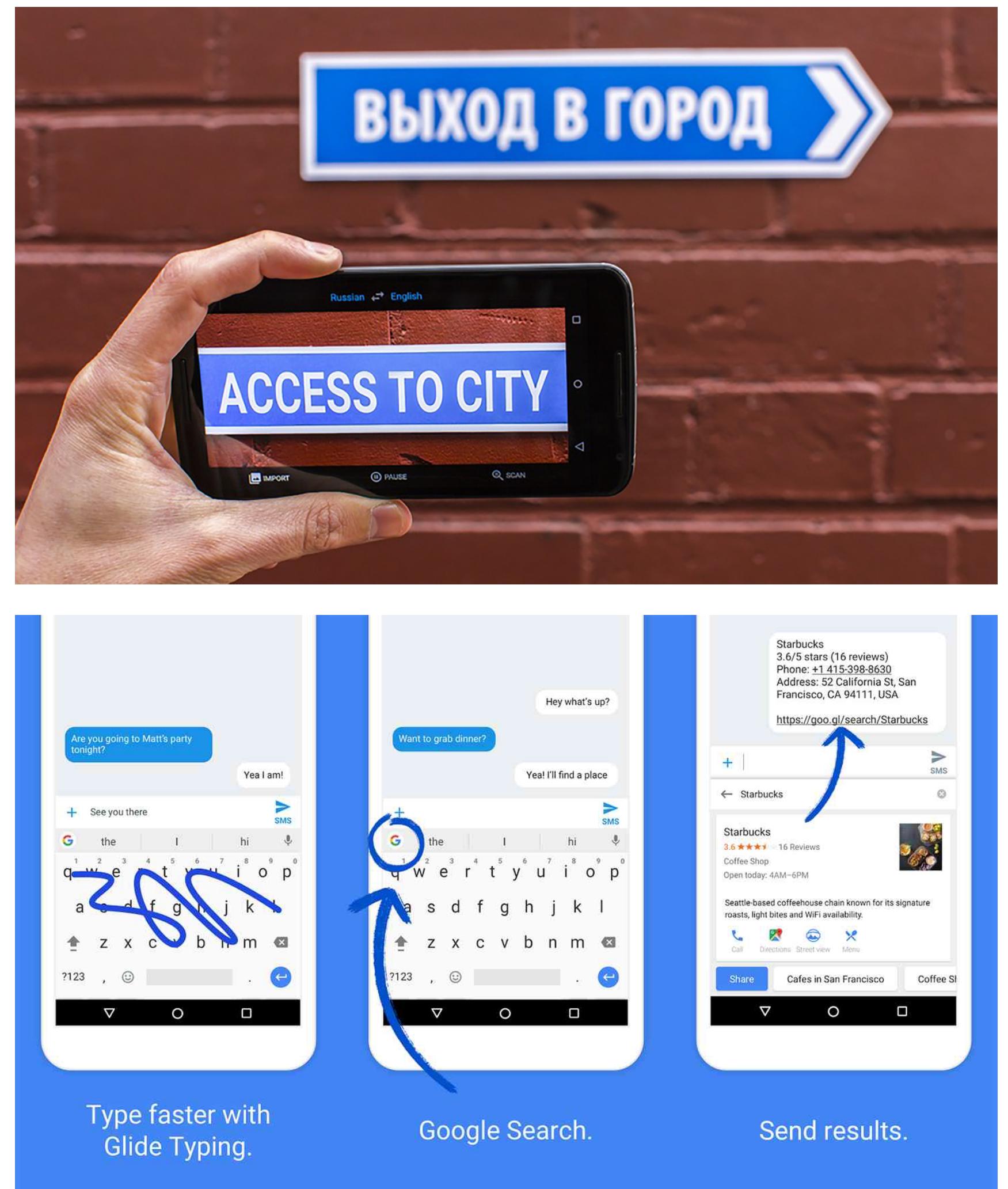
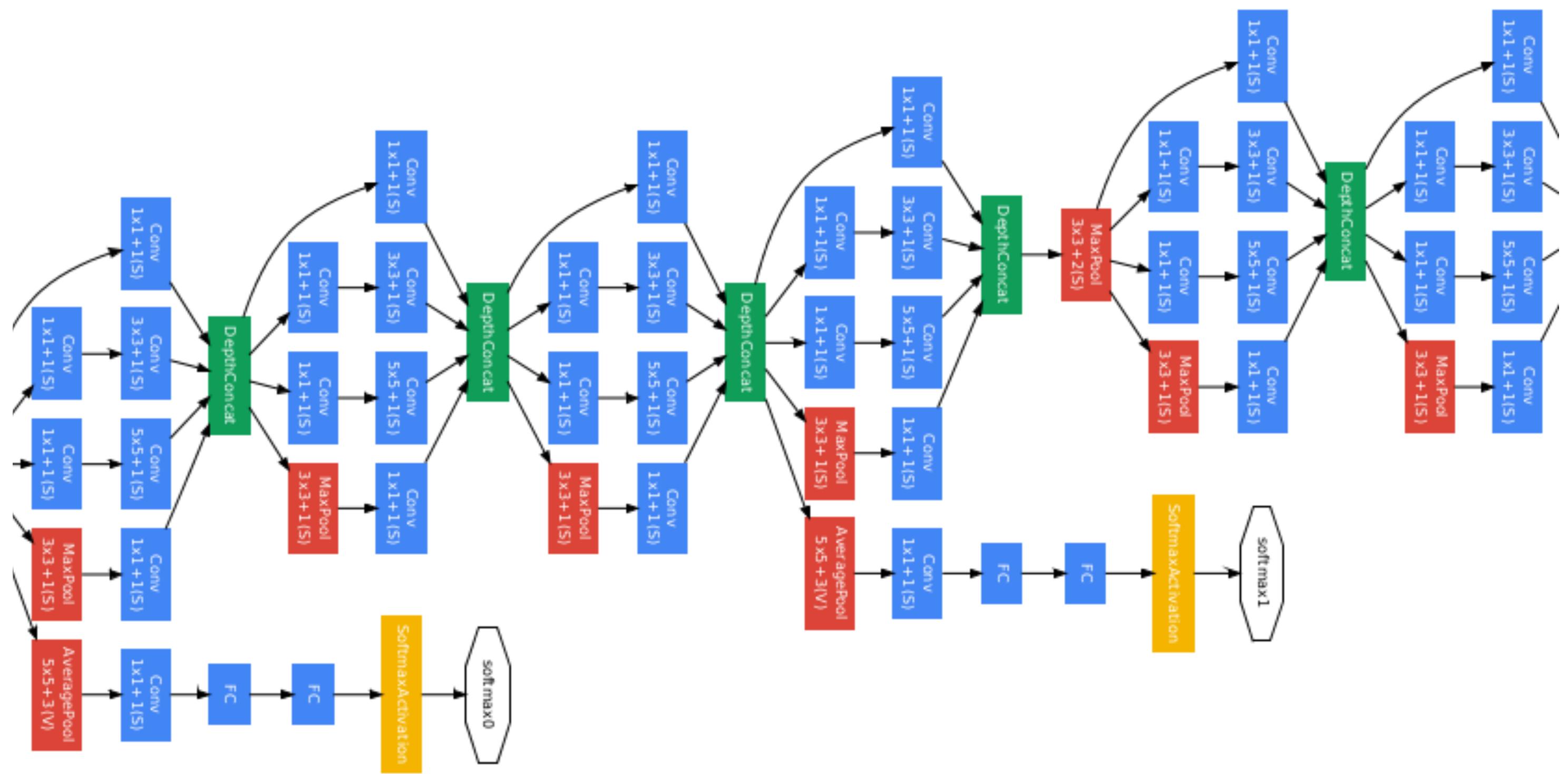
- 编译TensorFlow for iOS
- 转换模型
- 在iOS使用TensorFlow C++ API

```
tensorflow::Session* sess;
tensorflow::GraphDef graph;
PortableReadFileToProto(
    network_path, &graph);
tensorflow::NewSession(options,
    &session_pointer);
sess->Create(graph);
```



DEMO





TensorFlow被用于诸多App: Google Translate, GBoard, Google Photo...

BINARY SIZE

- 默认编译12MB
- 全功能编译100+MB
- 最小化编译（InceptionV3） 2MB



THE UGLY

- 缺少TensorFlow Serving
- 缺少GPU支持
- 一些遇到的坑
 - build_all_ios.sh
 - graph_optimizer.py
 - “No OpKernel found”错误



ENJOY AND MAKE YOUR APPS

