NEURAL NETWORKS
Human anatomy inspired learning network.
Neural Networks

dendrites
nucleus

axon
cell body
axon terminals

in$_1$
in$_2$
... 
in$_n$

out
Neural Networks
Neural Network – A Peek Inside

Input signals

$\Sigma$

Activation function

Output $y_k$

Synaptic weights

Bias $b_k$

Summing junction

$w_{k1}, w_{k2}, \ldots, w_{km}$

$x_1, x_2, \ldots, x_m$
Deep Neural Network
PREDICTION AND INFERENCE

How it works today. How it shall work tomorrow.
“Transfer to Infer” Approach
Why On-Device Prediction

- Data Privacy
- Poor internet connections
- Questionable user experience
To The Rescue...

TensorFlow
TensorFlow

• Tensor: N Dimensional Arrays
• Open source software library for numerical computation using data flow graphs.
TensorFlow – Data Flow Graphs

- Nodes represent mathematical functions
- Edges represent tensors.
Tensorflow – “Deferred Execution” Model

• Graph first. Computation Afterward.

```python
import tensorflow as tf
x = tf.constant(10)
y = tf.Variable(x + 5)
print(y)
```
Tensorflow – “Deferred Execution” Model

• Graph first. Computation Afterward.

```python
import tensorflow as tf

x = tf.constant(10)
y = tf.Variable(x + 5)

model = tf.global_variables_initializer()

with tf.Session() as session:
    session.run(model)
    print(session.run(y))
```
Packaging the App and the Model
QUANTIZATION

Compress. And Compress More.
Quantization

• **Round it up**
  • Transform: `round_weights`
  • Compression rates: \(~8\%\) \(\Rightarrow\) \(~70\%\)

• **Shrink down node names**
  • Transform: `obfuscate_names`

• **Eight bit calculations**
Quantization - Eight Bit Calculations

Input (float) → Relu → Output (float)

Quantize → QuantizedRelu → Dequantize → Output (float)

Min → Max
Quantization - Eight Bit Calculations

QuantizedMatMul

Dequantize

Min

Max

Quantize

QuantizedRelu

Dequantize

Output (float)

QuantizedMatMul

Dequantize

Output (float)
IMPLEMENTATION

Code Away! 😊
Implementation

build.gradle

```groovy
buildscript {
    repositories {
        jcenter()
    }

    dependencies {
        classpath 'com.android.tools.build:gradle:2.3.0'
    }
}
```
Implementation

1. Load
2. Feed
3. Run
4. Fetch
Implementation

1. Load the model

```java
TensorFlowInferenceInterface inferenceInterface =
    new TensorFlowInferenceInterface(assetManager, modelFile);
```

2. Feed in the input
3. Run the model
4. Fetch the output
Implementation

1. Load the model

2. Feed in the input

```java
// feed(String s, float[] floats, long... longs)
inferenceInterface.feed(inputName, floatValues, 1, inputSize, inputSize, 3);
```

3. Run the model

4. Fetch the output
Implementation

1. Load the model
2. Feed in the input
3. Run the model
   
   ```java
   inferenceInterface.run(outputNames);
   ```
4. Fetch the output
Implementation

1. Load the model
2. Feed in the input
3. Run the model
4. Fetch the output

// fetch(String s, float[] floats)
inferenceInterface.fetch(outputName, outputs);
APPLICATIONS

Awesomeness.
1. Take a picture

2. Process the image
   1. detect & locate
   2. crop
   3. normalize

3. Extract the text
   A trained Neural Network predicts the 14 pincode values

4. Display the results
   Bingo! A valid pincode was predicted
   Close! We're off by a few characters, ask the user to correct them

- TensorFlow
Thank you.
GREAT INDIAN DEVELOPER SUMMIT 2018

Conference and Deep Dive Sessions
April 24-28, IISc Bangalore

Register early and get the best discounts!