# CS 131 - Week 9

TA : Tanmay Sardesai

# How to find these slides

Piazza -> CS 131 -> Resources -> Discussion 1B

# Announcements

- Email tanmays@cs.ucla.edu
- Office Hours Thursday 1:30 pm - 3:30 pm. Bolter Hall 3256S-A
- Python Project extended March 12th (Tuesday) 11:55pm
- HW6 is due March 15th
  - Last day to submit is March 15th. No late submissions
- Evaluations due 8:00 AM Saturday, March 16th
- Next discussion will be preparing for the final so please review past content

# Topics covered today

- Project Questions?
- HW6
- Introduction to Tensorflow
- Tensorflow architecture overview
- Tensorflow language bindings
- Java vs Ocaml vs Kotlin

Project questions?

# Project

- https://web.cs.ucla.edu/classes/fall18/cs131/hw/pr.html

# Homework 6

- https://web.cs.ucla.edu/classes/fall18/cs131/hw/hw6.html

# Homework 6

- https://web.cs.ucla.edu/classes/fall18/cs131/hw/hw6.html
- Java vs Ocaml vs Kotlin

# Homework 6

- https://web.cs.ucla.edu/classes/fall18/cs131/hw/hw6.html
- Java vs Ocaml vs Kotlin
- Imagine you are running application server herd on a large set of virtual machines and your application heavily relies on Tensorflow

# Homework 6

- https://web.cs.ucla.edu/classes/fall18/cs131/hw/hw6.html
- Java vs Ocaml vs Kotlin
- Imagine you are running application server herd on a large set of virtual machines and your application heavily relies on Tensorflow
- You notice that small queries take reallllllly long to finish

# Homework 6

- https://web.cs.ucla.edu/classes/fall18/cs131/hw/hw6.html
- Java vs Ocaml vs Kotlin
- Imagine you are running application server herd on a large set of virtual machines and your application heavily relies on Tensorflow
- You notice that small queries take reallllllly long to finish
- Bottleneck is python code to setup the models

# Homework 6

- https://web.cs.ucla.edu/classes/fall18/cs131/hw/hw6.html
- Java vs Ocaml vs Kotlin
- Imagine you are running application server herd on a large set of virtual machines and your application heavily relies on Tensorflow
- You notice that small queries take reallllllly long to finish
- Bottleneck is python code to setup the models
- What language do you use to improve performance, ease of use, flexibility, generality and reliability

# Introduction to TensorFlow - high level python apis

https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/keras/basic_classification.ipynb

https://github.com/tensorflow/docs/blob/master/site/en/tutorials/keras/basic_classification.ipynb

# TensorFlow - What is a tensor?

- Central unit of data in TensorFlow
- It an n-dimensional array

```
3. # a rank 0 tensor; a scalar with shape [],
[1., 2., 3.] # a rank 1 tensor; a vector with shape [3]
[[1., 2., 3.], [4., 5., 6.]] # a rank 2 tensor; a matrix with shape [2, 3]
[[[1., 2., 3.]], [[7., 8., 9.]]] # a rank 3 tensor with shape [2, 1, 3]
```

# Tensorflow - low level

```
import numpy as np
import tensorflow as tf
a = tf.constant(3.0, dtype=tf.float32)
b = tf.constant(4.0) # also tf.float32
implicitly
total = a + b
print(a)
print(b)
print(total)
```

Output:

```
Tensor("Const:0", shape=(), dtype=float32)
Tensor("Const_1:0", shape=(), dtype=float32)
Tensor("add:0", shape=(), dtype=float32)
```

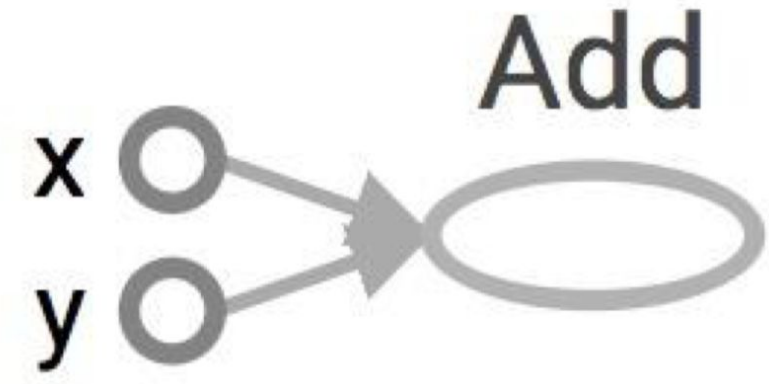
```
sess = tf.Session()
print(sess.run(total))
```

Output:

```
7.0
```

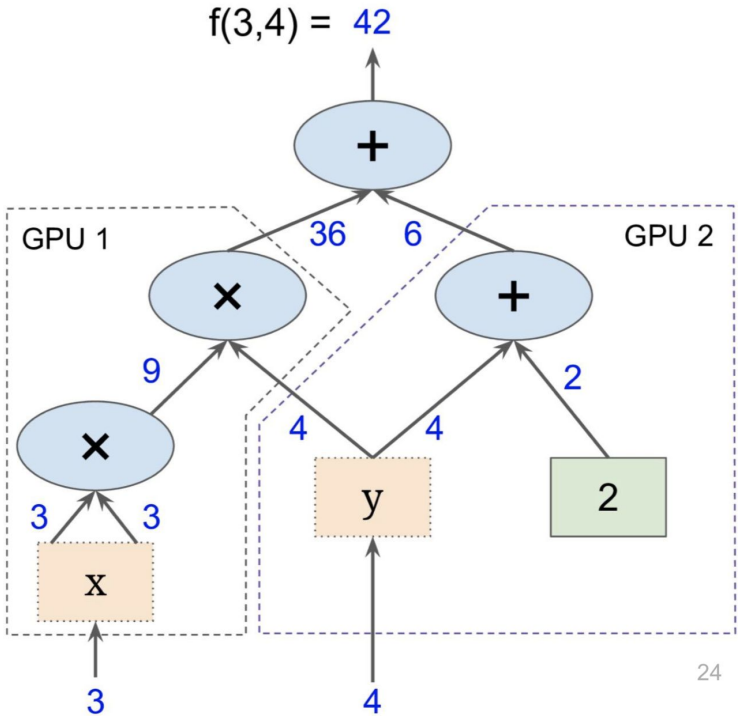
# Dataflow Graphs

Tensorflow separates definition of the model from the computation. The model definition is in the form of a graph.



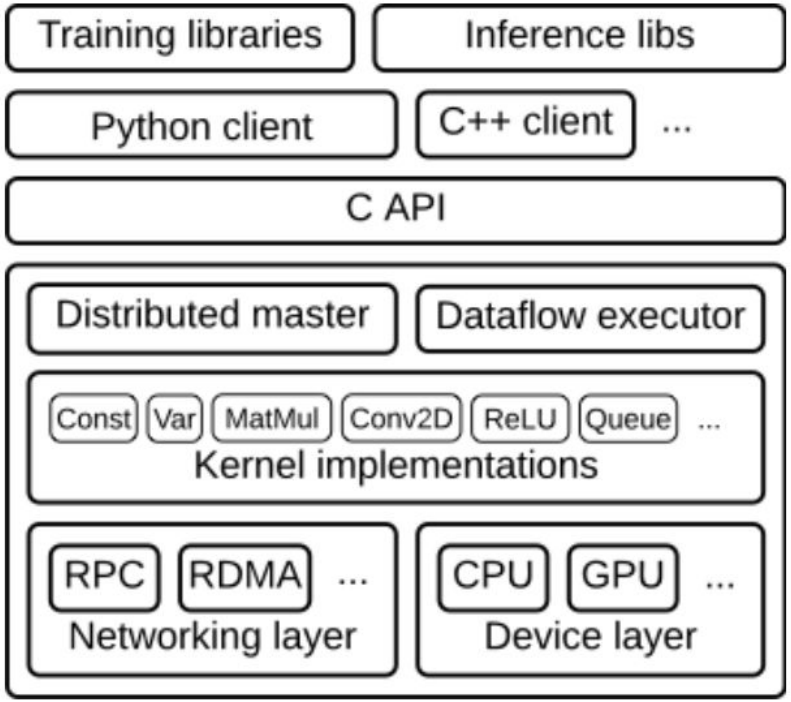
# Dataflow Graphs

Important because subgraphs can be used to optimize computations



# TensorFlow architecture

https://www.tensorflow.org/guide/extend/architecture



# Language bindings

- Most libraries are built in system programming languages like C or C++
- To use these libraries in other languages we have to create a binding using foreign function interface
- Why?

# Language bindings

- Most libraries are built in system programming languages like C or C++
- To use these libraries in other languages we have to create a binding using foreign function interface
- Why?
  - Code reuse
  - Somethings are better done at a lower-level

# TensorFlow in other languages

- Python client was one of the first high-level client. Currently supports most features
- TensorFlow community is moving more API implementations to C
- You can use TensorFlow C API to add support to any language
- Java - https://github.com/tensorflow/tensorflow/tree/master/tensorflow/java
- Ocaml - https://github.com/LaurentMazare/tensorflow-ocaml/
- Kotlin - https://juliuskunze.com/tensorflow-in-kotlin-native.html

# Tensorflow in other languages

https://www.tensorflow.org/guide/extend/bindings

# Other Tasks

- Write a program in Kotlin for everyNth which returns a list of every Nth element in a given input list
- Asyncio alternatives in these languages
- Your summary should focus on the technologies' effects on ease of use, flexibility, generality, performance, reliability; thie idea is to explore the most-important technical challenges in doing the proposed rewrite

# Kotlin

- In July 2011 JetBrains release Kotlin
- First official stable release in 2015
- 2017 Google announced first-class support for Kotlin on Android
- Coroutines for asynchronous programming were added in 2018

# Kotlin Features

- Efficiency
  - Syntax is intuitive and easier to understand than Java
  - Reduces code length. Less code -> Less bugs

- Extension functions
  - In most programming languages to add an extension to the class you have to create a new class that extends a superclass and then add functions to this new class.
  - Kotlin allows you to extend functionality
  - Example: `fun String.removeFirstLastChar(): String = this.substring(1, this.length – 1)`

# Kotlin Features

- Interoperability
  - Java code works with Kotlin
  - Kotlin code works with Java

- Nullable and non-nullable
  - Kotlin types are strict and do not allow null initialization to reduce run-time errors.
  - var name: String? = null
  - var name: String = null ; ERROR at compile time

- Safe and reliable
  - Because of the above features and small amount of code

# Kotlin Features

- Low cost to use
  - Open source
  - 100% interoperability with Java
  - Safe and reliable
  - Easy to use
  - Efficient

# Kotlin design

- Semicolons are optional
- Use val keyword for immutable variables
- Use var keyword for mutable variables
- Class members are public by default
- Classes are final by default
- Kotlin not only supports object oriented programming but also procedural programming
- Like C and C++ the entry point for Kotlin is main()

# Kotlin basic code

```
fun main(args: Array<String>) {
  val scope = "World"
  println("Hello, $scope!")
}
```

# References

- https://kotlinlang.org
- https://kotlinlang.org/docs/tutorials/
- https://play.kotlinlang.org/koans/overview
- https://en.wikipedia.org/wiki/Kotlin_(programming_language)
- https://instabug.com/blog/kotlin-features/amp/
- https://kotlinlang.org/docs/reference/comparison-to-java.html
- https://github.com/tensorflow/swift/blob/master/docs/WhySwiftForTensorFlow.md (This doc argues why swift is good for tensorflow use this to figure out what are some criterias you should look at when discussing Kotlin. This also talks about weaknesses and strengths of python)

# References

- https://web.stanford.edu/class/cs224n/lectures/lecture6.pdf - Good tensorflow resource