Reviev

DI Librario

Introductio
Symbolic
Programming
Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Up Next

# Deep Learning with TensorFlow Deep Learning — Units 7 & 8

Dr. Jon Krohn
jon@untapt.com

Slides available at jonkrohn.com/talks

April 6, 2019



Review

DI Librarie

Introduction Symbolic Programming Graphs

Fitting Model
Eight Data Points
Eight Million Point
Dense Nets
ConvNets

TF 2.0

Project Improvemer Where We Are Ten Tunings

In Nev

- Review Take-Home Exercise
- Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow
  Symbolic Programming
  Programming TensorFlow Graphs
  Neurons in TensorFlow
- 4 Fitting Models
   Eight Data Points
   Eight Million Points
   Dense Nets
   Convolutional Nets
- **5** TF 2.0
- 6 Deep Learning Project IV: Improving
  Where We Are
  Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



\_. . . . . . .

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project
Improvemen
Where We Are
Ten Tunings

Up Nex

#### **Outline**

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow Symbolic Programming Programming TensorFlow Graphs Neurons in TensorFlow
- 4 Fitting Models
  Eight Data Points
  Eight Million Points
  Dense Nets
  Convolutional Nets
- **5** TF 2.0
- 6 Deep Learning Project IV: Improving
  Where We Are
  Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



- Review
- DL Librarie
- Introductio
  Symbolic
  Programming
  Graphs
- Fitting Model
  Eight Data Points
  Eight Million Points
  Dense Nets
  ConvNets
- TF 2.0

Project Improvement Where We Are

Hn Ne

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow
  Symbolic Programming
  Programming TensorFlow Graphs
  Neurons in TensorFlow
  - 4 Fitting Models
     Eight Data Points
     Eight Million Points
     Dense Nets
     Convolutional Nets
- **5** TF 2.0
- 6 Deep Learning Project IV: Improving Where We Are Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



Review

L Librarie

Introductio Symbolic Programming Graphs Neurons

Fitting Model: Eight Data Points Eight Million Points Dense Nets ConvNets

TF 2.0

Project Improvemer Where We Are Ten Tunings

Un Ne

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow
  Symbolic Programming
  Programming TensorFlow Graphs
  Neurons in TensorFlow
- 4 Fitting Models
  Eight Data Points
  Eight Million Points
  Dense Nets
  Convolutional Nets
- **5** TF 2.0
- 6 Deep Learning Project IV: Improving
  Where We Are
  Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



Review

L Librarie

Introductio
Symbolic
Programming
Graphs
Neurons

Fitting Model: Eight Data Points Eight Million Points Dense Nets ConvNets

TF 2.0

Project Improvemer Where We Are Ten Tunings

Up Ne

- Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow
  Symbolic Programming
  Programming TensorFlow Graphs
  Neurons in TensorFlow
  - 4 Fitting Models
    Eight Data Points
    Eight Million Points
    Dense Nets
    Convolutional Nets
- **5** TF 2.0
- 6 Deep Learning Project IV: Improving
  Where We Are
  Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



- Review
- L Librarie
- Introductio
  Symbolic
  Programming
  Graphs
  Neurons
- Fitting Model
  Eight Data Points
  Eight Million Point
  Dense Nets
  ConvNets
- TF 2.0

Project Improvemer Where We Are

Up Ne

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow
  Symbolic Programming
  Programming TensorFlow Graphs
  Neurons in TensorFlow
- 4 Fitting Models

Eight Data Points
Eight Million Points

Dense Nets

Convolutional Nets

- **5** TF 2.0
- 6 Deep Learning Project IV: Improving

Where We Are

Ten Hyperparameter-Tuning Steps





Reviev

L Librarie

Introductio Symbolic Programming Graphs Neurons

Fitting Model
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Jp Ne

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow
  Symbolic Programming
  Programming TensorFlow Graphs
  Neurons in TensorFlow
- 4 Fitting Models
  Eight Data Points
  Eight Million Points
  - Dense Nets
  - Convolutional Nets
- **5** TF 2.0
- 6 Deep Learning Project IV: Improving
  - Where We Are
    - Ten Hyperparameter-Tuning Steps
- 7 Up Next: Advanced Topics



#### Review

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Model
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

Project Improvemer Where We Are Ten Tunings

Un Ne

- Review Take-Home Exercise
- Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow
  Symbolic Programming
  Programming TensorFlow Graphs
  Neurons in TensorFlow
- 4 Fitting Models
  Eight Data Points
  Eight Million Points
  Dense Nets
  Convolutional Nets
- **5** TF 2.0
- 6 Deep Learning Project IV: Improving Where We Are Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



#### Review

al Libraria

#### Introduction

Symbolic Programmin Graphs

#### Fitting Mod

Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

#### Project

Where We Are

Up Next

# Assessing Your Deep Learning Project III





#### Review

DL Librarie

Introduction Symbolic Programming

Programming Graphs Neurons

Fitting Models
Eight Data Points
Eight Million Points

TF 2.0

Project Improveme Where We Are

Up Next

# Assessing Your Deep Learning Project III

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- ② Building and assessing architecture
  - get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark



#### Review

DL Librari

Introductio Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points

ConvNets

11 2.

Project Improvemer Where We Are Ten Tunings

Up Next

# Assessing Your Deep Learning Project III

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- 2 Building and assessing architecture
  - get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark



#### Review

DL Librarie

Introductio
Symbolic
Programming
Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2 0

Project Improvemen Where We Are Ten Tunings

Up Next

# Assessing Your Deep Learning Project III

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- 2 Building and assessing architecture
  - get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark



#### Review

DL Librari

Introduction Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvement Where We Are Ten Tunings

Up Next

# Assessing Your Deep Learning Project III

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- 2 Building and assessing architecture
  - get above chance (simplifying problem, if necessary)
     do existing performance benchmarks exist?
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark



#### Review

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Up Next

# Assessing Your Deep Learning Project III

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- 2 Building and assessing architecture
  - get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark



Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Up Next

# Assessing Your Deep Learning Project III

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- 2 Building and assessing architecture
  - get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark



#### Review

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Up Next

# Assessing Your Deep Learning Project III

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- 2 Building and assessing architecture
  - get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark



#### Review

DL Librario

Introduction Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvemer Where We Are Ten Tunings

Up Next

# Assessing Your Deep Learning Project III

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- 2 Building and assessing architecture
  - get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark



Review

**DL** Libraries

Introduction Symbolic Programming Graphs

Fitting Model
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

Project
Improvemen
Where We Are
Ten Tunings

In Nev

- Heview Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
  - 3 Introduction to TensorFlow
    Symbolic Programming
    Programming TensorFlow Graphs
    Neurons in TensorFlow
  - 4 Fitting Models
    Eight Data Points
    Eight Million Points
    Dense Nets
    Convolutional Nets
  - **5** TF 2.0
- 6 Deep Learning Project IV: Improving
  Where We Are
  Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Up Nex

# Leading DL Libraries

	Caffe	Torch	MXNet	TensorFlow
Language	Python, Matlab	Lua, C	Python, R, C++ Julia, Matlab JavaScript, Go Scala, Perl	Python, R, C++ C, Java, Go
Programming Style	Symbolic	Imperative	Imperative	Symbolic
Parallel GPUs: Data	Yes	Yes	Yes	Yes
Parallel GPUs: Model		Yes	Yes	Yes
Pre-Trained Models	Model Zoo	ModelZoo	Model Zoo	github.com/tensorflow/ models
For RNNs				Best
High-Level APIs		PyTorch	in-built	Keras, TFLearn



Pearson, Inc.

Review

DI Librarie

Introduction

Symbolic Programming Graphs Neurons

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

Project Improvemen Where We Are

I In Ne

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow

Programming
Programming TensorFlow Graphs
Neurons in TensorFlow

- 4 Fitting Models
  Eight Data Points
  Eight Million Points
  Dense Nets
- **5** TF 2.0
- 6 Deep Learning Project IV: Improving
  Where We Are
  Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



Review

DL Librarie

Introductio
Symbolic
Programming
Graphs
Neurons

Fitting Model:
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Un Ne

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow Symbolic Programming Programming TensorFlow Graphs Neurons in TensorFlow
- 4 Fitting Models
  Eight Data Points
  Eight Million Points
  Dense Nets
  Convolutional Nets
- **5** TF 2.0
- 6 Deep Learning Project IV: Improving Where We Are Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



Symbolic Programming

# TensorFlow Graphs

- 1 build graph



#### Review

DI Librarie

#### Introduction

Symbolic Programming Graphs

#### Fitting Models

Eight Data Points
Eight Million Points

ConvNe

TF 2.

#### \_ .

Improvement Where We Are Ten Tunings

Up Nex

# TensorFlow Graphs

- build graph
- 2 initialize session
- g fetch and feed data



#### Review

DL Librarie

#### Introduction

Symbolic Programming Graphs

### Fitting Models Eight Data Points

Eight Million Point

ConvNet

#### TF 2.

Improvemer
Where We Are
Ten Tunings

Up Next

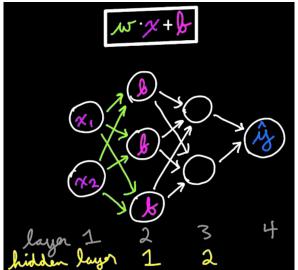
## TensorFlow Graphs

- build graph
- 2 initialize session
- 3 fetch and feed data



# A Familiar Equation

Symbolic Programming







#### Review

DI Libraria

Introductio

Symbolic Programming

Graphs

Fitting Models
Eight Data Points
Eight Million Points

ConvN

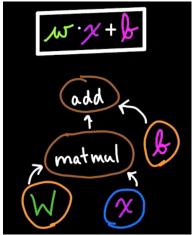
TF 2.0

Project Improveme

Where We Are

Up Next

### TensorFlow Graphs



©2018 Pearson, Inc.



Review

DL Librarie

Introductio
Symbolic
Programming
Graphs
Neurons

Fitting Model
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Up Ne:

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow Symbolic Programming Programming TensorFlow Graphs

Neurons in TensorFlow

- 4 Fitting Models
  Eight Data Points
  Eight Million Points
  Dense Nets
  Convolutional Nets
- **5** TF 2.0
- 6 Deep Learning Project IV: Improving
  Where We Are
  Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



Graphs

## TensorFlow Graph **Programming**

[ first TensorFlow graphs notebook ]



Review

DL Librarie

Introductio
Symbolic
Programming
Graphs
Neurons

Fitting Model
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Un Ne

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow
  Symbolic Programming
  Programming TensorFlow Graphs
  Neurons in TensorFlow
- 4 Fitting Models
  Eight Data Points
  Eight Million Points
  Dense Nets
  Convolutional Nets
- **5** TF 2.0
- 6 Deep Learning Project IV: Improving Where We Are Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



# A Familiar Equation

Review

DI Disease

Introduction

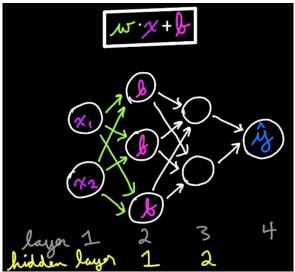
Symbolic Programmin Graphs Neurons

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TFo

Project Improvemer Where We Are

Up Next







Review

DL Librario

Introduction

Symbolic Programmir Graphs Neurons

Eitting Mode

Eight Data Points Eight Million Points Dense Nets

TF 2 C

Project Improvemen Where We Are Ten Tunings

Up Next

# Neurons in TensorFlow Programming

[ first TensorFlow neurons notebook ]



Review

DI Librarie

Introduction Symbolic Programming Graphs

Fitting Models

Eight Data Points

ConvNet

Project Improvemen Where We Are

Un Ne

- Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow
  Symbolic Programming
  Programming TensorFlow Graphs
  Neurons in TensorFlow
- 4 Fitting Models

Eight Million Points
Dense Nets

Convolutional Nets

- **5** TF 2.0
- 6 Deep Learning Project IV: Improving
  Where We Are
  Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



- Review
- DL Librarie
- Introduction Symbolic Programming Graphs
- Fitting Model
  Eight Data Points
  Eight Million Points
  Dense Nets
  ConvNets
- TF 2.0

Project Improvemer Where We Are Ten Tunings

l In Ne

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
  - 3 Introduction to TensorFlow Symbolic Programming Programming TensorFlow Graphs Neurons in TensorFlow
  - 4 Fitting Models
    Eight Data Points
    Eight Million Points
    Dense Nets
    Convolutional Nets
  - **5** TF 2.0
  - 6 Deep Learning Project IV: Improving Where We Are Ten Hyperparameter-Tuning Steps
  - Up Next: Advanced Topics



Review

DI Librarie

Introduction

Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvemer Where We Are Ten Tunings

Up Next

# Fitting Eight Points

[ point by point intro to TensorFlow notebook ]



Review

DL Librario

Introduction

Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvement Where We Are Ten Tunings

Up Next

# Fitting Eight Points with Tensors

[ tensor-fied intro to TensorFlow notebook ]



# **Outline**

Review

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Up Ne

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
  - 3 Introduction to TensorFlow Symbolic Programming Programming TensorFlow Graphs Neurons in TensorFlow
  - 4 Fitting Models

Eight Data Points

**Eight Million Points** 

Dense Nets Convolutional Nets

- **5** TF 2.0
- 6 Deep Learning Project IV: Improving
  Where We Are
  Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



Review

DI Librarie

Introduction

Programmin Graphs

Fitting Mode

Eight Data Points

Eight Million Points

Dense Nets ConvNets

TF 2.0

.. \_..

Improvemen Where We Are

Up Next

# Fitting Eight Million Points

[ intro to TensorFlow times a million notebook ]



# **Outline**

I ICVICW

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Mode
Eight Data Points
Eight Million Point
Dense Nets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Up Ne:

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
  - 3 Introduction to TensorFlow
    Symbolic Programming
    Programming TensorFlow Graphs
    Neurons in TensorFlow
  - 4 Fitting Models

Eight Data Points

Eight Million Points

Dense Nets

Convolutional Nets

- **5** TF 2.0
- 6 Deep Learning Project IV: Improving
  Where We Are
  Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



## **Dense Nets**

Dense Nets

[ intermediate net in TensorFlow notebook ]



## **Dense Nets**

Reviev

DI Librarie

Introduction

Programmin Graphs

Fitting Mode

Eight Data Points

Dense Nets

ConvNets

-----

Improveme Where We Are

Where We Are Ten Tunings

Up Next

[ deep net in TensorFlow notebook ]



# **Outline**

Review

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Model
Eight Data Points
Eight Million Point
Dense Nets
ConvNets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Un Ne

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow Symbolic Programming Programming TensorFlow Graphs Neurons in TensorFlow
- 4 Fitting Models
  Eight Data Points
  Eight Million Points
  Dense Nets
  Convolutional Nets
- 5 TF 2 0
- 6 Deep Learning Project IV: Improving
  Where We Are
  Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



Review

DI Librario

Introduction

Symbolic Programming Graphs

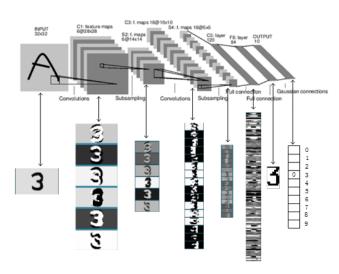
Fitting Models
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

Project Improveme

l In Nev

# LeNet-5 LeCun et al. (1998)





Review

DI Librarie

Introduction

Programming Graphs

Fitting Models

Eight Million Points

Dense Nets

ConvNets

TF 2

Improveme

Where We Are Ten Tunings

Up Next

# LeNet-5

LeCun et al. (1998)

[ LeNet in TensorFlow notebook ]



# **Outline**

Review

DI Librarie

Introduction Symbolic Programming Graphs

Fitting Model: Eight Data Points Eight Million Points Dense Nets ConvNets

TF 2.0

Project Improvement Where We Are Ten Tunings

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
  - 3 Introduction to TensorFlow Symbolic Programming Programming TensorFlow Graphs Neurons in TensorFlow
  - 4 Fitting Models
     Eight Data Points
     Eight Million Points
     Dense Nets
     Convolutional Nets
  - **5** TF 2.0
  - 6 Deep Learning Project IV: Improving Where We Are Ten Hyperparameter-Tuning Steps
  - Up Next: Advanced Topics



Review

DI Librari

Introductio

Programming
Graphs
Neurons

Fitting Models

Eight Data Points

ConvNets

TF 2.0

Improvemen
Where We Are

Up Next

## · Eager execution by default

- subclassing for unlimited flexibility
- TF Serving: high-performance systems on servers
- TF Lite: for mobile or embedded devices
- TensorFlow.js: for web browsers



Review

DL Librari

Introduction

Programming
Graphs
Neurons

Fitting Models

Eight Data Points

Eight Data Points
Eight Million Points

TF 20

Project Improveme

Improvement Where We Are Ten Tunings

- Eager execution by default
- subclassing for unlimited flexibility
- TF Serving: high-performance systems on servers
- TF Lite: for mobile or embedded devices
- TensorFlow.js: for web browsers



#### Revie

DL Librari

## Introduction

Programming
Graphs
Neurons

# Fitting Models Eight Data Points Eight Million Points

ConvNets

### TF 2.0

Improvement
Where We Are
Ten Tunings

- · Eager execution by default
- subclassing for unlimited flexibility
- TF Serving: high-performance systems on servers
  - TF Lite: for mobile or embedded devices
- TensorFlow.js: for web browsers



#### Review

DL Librari

# Introduction Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvement Where We Are Ten Tunings

- · Eager execution by default
- subclassing for unlimited flexibility
- TF Serving: high-performance systems on servers
- TF Lite: for mobile or embedded devices
- TensorFlow.js: for web browsers



### Revie

DL Librari

# Introduction Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

### Project Improvemen Where We Are Ten Tunings

- · Eager execution by default
- subclassing for unlimited flexibility
- TF Serving: high-performance systems on servers
- TF Lite: for mobile or embedded devices
- TensorFlow.js: for web browsers



Review

DI Librarios

Introduction

Programming
Graphs
Neurons

Fitting Mode

Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvemen Where We Are

Up Next

[ new TF 2.0 repo ]



# **Outline**

Review

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

### Project Improvement

Where We Are Ten Tunings

Up Ne

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
  - 3 Introduction to TensorFlow Symbolic Programming Programming TensorFlow Graphs Neurons in TensorFlow
  - 4 Fitting Models
    Eight Data Points
    Eight Million Points
    Dense Nets
    Convolutional Nets
  - **5** TF 2.0
  - 6 Deep Learning Project IV: Improving
    Where We Are
    Ten Hyperparameter-Tuning Steps
  - Up Next: Advanced Topics



# **Outline**

Review

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Model
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

Project Improvemen Where We Are Ten Tunings

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow Symbolic Programming Programming TensorFlow Graphs Neurons in TensorFlow
- 4 Fitting Models
   Eight Data Points
   Eight Million Points
   Dense Nets
   Convolutional Nets
- **5** TF 2.0
- 6 Deep Learning Project IV: Improving Where We Are
  Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



#### Review

al Libraria

Introduction

Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points

TF 2.0

Project Improveme Where We Are

Up Nex

# Improving Your Deep Learning Project IV





Review

DL Librarie

Introduction

Symbolic Programming Graphs Neurons

Fitting Model

Eight Data Points

Dense Nets ConvNets

TF 2.0

Improvemer
Where We Are

Up Nex

# Improving Your Deep Learning Project IV

# Splitting your data

- training set (80% for optimizing parameters)
- validation set (10% for hyperparameters)
- test set (10% don't touch yet!)
- 2 Building and assessing architecture
  - get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark
- Improving performance & tuning hyperparameters in ten steps...



Review

DL Librari

Introduction
Symbolic
Programming
Graphs

Fitting Model
Eight Data Points

TF 2.0

Project

Improvemen
Where We Are
Ten Tunings

Up Nex

# Improving Your Deep Learning Project IV

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- ② Building and assessing architecture
  - get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark
- Improving performance & tuning hyperparameters in ten steps...



Review

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points

TF 2.0

Project Improvemer Where We Are Ten Tunings

Up Next

# Improving

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
    - test set (10% don't touch yet!)
- ② Building and assessing architecture
  - get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark
- Improving performance & tuning hyperparameters in ten steps...



Review

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Model

Eight Data Points

Fight Million Points

TF 2.0

Project Improvemer Where We Are

Up Next

# Improving

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- ② Building and assessing architecture
  - get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark
- Improving performance & tuning hyperparameters in ten steps...



#### Review

DL Librari

Introduction Symbolic Programming Graphs

Fitting Model

TF 2.0

Project Improvemer Where We Are

Up Nex

# Improving Your Deep Learning Project IV

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- 2 Building and assessing architecture
  - · get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark
- Improving performance & tuning hyperparameters in ten steps...



Reviev

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Model

TF 2.0

Project Improvemen Where We Are

Up Next

# Improving

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- 2 Building and assessing architecture
  - · get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark
- Improving performance & tuning hyperparameters in ten steps...



Review

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Model: Eight Data Points Eight Million Points

TF 2.0

Project Improvemen Where We Are Ten Tunings

Up Next

# Improving

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- 2 Building and assessing architecture
  - · get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark
- Improving performance & tuning hyperparameters in ten steps...



Review

DL Librarie

Introduction Symbolic Programming Graphs Neurons

Fitting Models
Eight Data Points
Eight Million Points

TF 2.0

Project Improvemen Where We Are Ten Tunings

Up Next

# Improving

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- 2 Building and assessing architecture
  - · get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark
- Improving performance & tuning hyperparameters in ten steps...



Review

DL Librarie

Introduction
Symbolic
Programming
Graphs
Neurons

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Up Next

# Improving

- Splitting your data
  - training set (80% for optimizing parameters)
  - validation set (10% for hyperparameters)
  - test set (10% don't touch yet!)
- 2 Building and assessing architecture
  - get above chance (simplifying problem, if necessary)
  - do existing performance benchmarks exist?
  - if not, use a simple architecture as benchmark
- Improving performance & tuning hyperparameters in ten steps...



## **Outline**

Review

DI Librarie

Introduction Symbolic Programming Graphs

Fitting Model
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

Project Improveme Where We Are Ten Tunings

l In Ne

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
- 3 Introduction to TensorFlow
  Symbolic Programming
  Programming TensorFlow Graphs
  Neurons in TensorFlow
- 4 Fitting Models
  Eight Data Points
  Eight Million Points
  Dense Nets
  Convolutional Nets
- **5** TF 2.0
- 6 Deep Learning Project IV: Improving
  Where We Are
  Ten Hyperparameter-Tuning Steps
- Up Next: Advanced Topics



Reviev

DL Librario

Introduction

Symbolic Programming Graphs Neurons

Fitting Models

Eight Data Points

Eight Million Poin Dense Nets ConvNets

TF 2.0

Project Improveme Where We Are Ten Tunings

Un Next

## 1. Initialization

...in lenet\_in\_tensorflow.ipynb:

```
Set neural network hyperparameters

epochs = 20
batch_size = 128
display progress = 40 # after this many batches, output progress to screen
wt_init = tf.contrib.layers.xavier_initializer() # weight initializer
```



### \_ .

DI Libraria

#### Introduction

Symbolic Programming Graphs

# Eight Data Points Fight Million Points

Eight Million Points
Dense Nets
ConvNets

TF 2.0

### Project Improveme

Improvemen Where We Are Ten Tunings

Up Next

## 2. Get Above Chance

## If your accuracy is below chance, try:

- simplifying the problem
- simplifying the network architecture
- reducing your training set size (to iterate more quickly)



Fitting Models
Eight Data Points
Eight Million Points
Dense Nets

TF 2.0

Project Improvement Where We Are Ten Tunings

Up Next

## 2. Get Above Chance

If your accuracy is below chance, try:

- simplifying the problem
- simplifying the network architecture
- reducing your training set size (to iterate more quickly)



# Ten Tunings

### 2. Get Above Chance

If your accuracy is below chance, try:

- simplifying the problem
- simplifying the network architecture
- reducing your training set size (to iterate more quickly)



Ten Tunings

## Experiment with varying:

- number of layers



# 3. Layers

Review

DL Librarie

Introductio

Programming
Graphs
Neurons

Eight Data Points
Eight Million Points

Dense Nets
ConvNets

1F 2.

Project Improveme

IMPROVEMEN Where We Are Ten Tunings

Up Next

## Experiment with varying:

- · number of layers
- type of layers
- layer width (by powers of two)



# 3. Layers

Reviev

DL Librari

Introductio

Programming
Graphs
Neurons

Eight Data Points
Eight Million Points

TF 2.

Project Improvement Where We Are Ten Tunings

Up Next

## Experiment with varying:

- number of layers
- · type of layers
- layer width (by powers of two)



Fitting Models
Eight Data Points
Eight Million Points

TF 2.0

Improvem

Where We Are
Ten Tunings

Up Nex

...in lenet in keras.ipynb:

#### Configure model

model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

...in lenet in tensorflow.ipynb:

#### Define model's loss and its optimizer

cost = tf.reduce\_mean(tf.nn.softmax\_cross\_entropy\_with\_logits(logits=predictions, labels=y)
optimizer = tf.train.AdamOptimizer().minimize(cost)



### Review

DL Librarie

### Introduction Symbolic Programming

Symbolic Programming Graphs Neurons

# Eight Data Points Eight Million Points Dense Nets ConvNets

TF 2.0

### Project Improvement Where We Are Ten Tunings

Up Next

## 5. Avoid Overfitting

If validation cost begins to increase or validation accuracy begins to decrease, consider:

- stopping training earlier
- dropout



### Review

DL Librarie

# Introduction Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

Project Improvemer Where We Are Ten Tunings

Up Next

## 5. Avoid Overfitting

If validation cost begins to increase or validation accuracy begins to decrease, consider:

- stopping training earlier
- dropout



Units 7 and 8 - Tensorflow

Ten Tunings

## 5. Avoid Overfitting

## ...in lenet in keras.ipynb:

```
model = Sequential()
model.add(Conv2D(32, kernel size=(3, 3), activation='relu', input shape=(28, 28, 1)))
model.add(Conv2D(64, kernel size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128. activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(n classes, activation='softmax'))
```

## ...in lenet in tensorflow.ipynb:

```
# convolutional and max-pooling layers:
                                conv 1 = conv2d(square x, weights['W c1'], biases['b c1'])
# max pooling laver:
                                conv 2 = conv2d(conv 1, weights['W c2'], biases['b c2'])
pool size = 2
                               pool 1 = maxpooling2d(conv 2, mp psize)
mp layer dropout = 0.25
                                pool 1 = tf.nn.dropout(pool 1, 1-mp dropout)
# dense layer:
                                # dense laver:
n dense = 128
                                flat = tf.reshape(pool 1, [-1, weights['W dl'].get shape().as list()[0]])
dense laver dropout = 0.5
                               dense 1 = dense(flat, weights['W dl'], biases['b dl'])
                                dense 1 = tf.nn.dropout(dense 1, 1-dense dropout)
```



Fitting Model
Eight Data Points

Eight Data Points
Eight Million Point

TF 2.0

Project

Where We Are

Un Next

...in lenet in keras.ipynb:

Configure model

model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

...in lenet\_in\_tensorflow.ipynb:

Define model's loss and its optimizer

cost = tf.reduce mean(tf.nn.softmax cross entropy with\_logits(logits=predictions, labels=y))
optimizer = tf.train.AdamOptimizer().minimize(cost)



Pearson, Inc.

02018

### Review

DI Libraria

### Introductio

Programmin Graphs Neurons

## Fitting Model

Eight Million Points
Dense Nets

### TF 2.

1 - 2.1

## Improvem

Where We Are
Ten Tunings

Up Nex

## 7. Epochs

## ...in lenet in keras.ipynb:

```
Train!
model.fit(X_train, y_train, batch_size=128, epochs=20, verbose=1, validation_data=(X_test, y_test))
```

## ...in lenet in tensorflow.ipynb:

```
Set neural network hyperparameters

| pochs = 20 |
| baccn_mixe = 128 |
| display_progress = 40 # after this many batches, output progress to screen
| wt_init = tf.contrib.layers.xavier_initializer() # weight initializer

# loop over epochs;
| for epoch in range epoch epoc
```



## 8. Regularization $\lambda$

Review

DL Librari

Introductio

Programmir Graphs

Fitting Models
Eight Data Points
Eight Million Points

Eight Million Points
Dense Nets
ConvNets

TF 2.0

Improvement
Where We Are
Ten Tunings

Up Next

If using L1 or L2 regularization, consider:

• adjusting  $\lambda$  by orders of magnitude



Units 7 and 8 — Tensorflow

### Review

DI Libraria

### Introduction

Symbolic Programming Graphs

## Fitting Mode

Eight Data Points
Eight Million Points
Dense Nets

TF 2.

### IF 2.

Improveme
Where We Are
Ten Tunings

Up Nex

## 9. Batch Size

...in lenet\_in\_keras.ipynb:

```
Train!

model.fit(X_train, y_train, batch_size=128, epochs=20, verbose=1, validation_data=(X_test, y_test))
```

## ...in lenet in tensorflow.ipynb:

```
Set neural network hyperparameters
```

```
enochs = 20
batch_size = 128
display_progress = 40 # after this is
wt init = tf.contrib.layers.xavier is
```

```
# loop over all batches of the epoch:
n batches = int(mist.train.num_examples / batch_size
for i in range(n batches):
    # to reassure you something's happening!
    if i & display progress == 0:
        print("Step", i.t., of "n_batches, " in epoch ", epoch+1, ".", sep='')
    batch_x, batch_y = mnist.train.next_batch_batch_size
```



# Ten Tunings

## 10. Automation

## For grid search of hyperparameters, consider:

- sampling values instead of looping over fixed values



Ten Tunings

## 10. Automation

For grid search of hyperparameters, consider:

- sampling values instead of looping over fixed values
- using [ Spearmint ]



### Review

DL Librarie

### Introduction

Symbolic Programming Graphs Neurons

## Fitting Models Eight Data Points

Eight Million Points
Dense Nets
ConvNets

TF 2.0

## Project

Improvemen Where We Are Ten Tunings

Up Nex

- Initialization
- 2 Get Above Chance
- 3 Layers
- 4 Cost
- 6 Avoid Overfitting
- 6 Learning Rate
- Epochs
- f 8 Regularization  $\lambda$
- 9 Batch Size
- Automation



- Initialization
- Get Above Chance



- Initialization
- Get Above Chance
- 3 Layers



- Initialization
- Get Above Chance
- 3 Layers
- 4 Cost



### Review

DL Librarie

## Introduction

Programming
Graphs
Neurons

# Fitting Models Eight Data Points Eight Million Points

Eight Million Points
Dense Nets
ConvNets

TF 2.0

### Project Improvement

Improvemen Where We Are Ten Tunings

Up Nex

- Initialization
- 2 Get Above Chance
- 3 Layers
- 4 Cost
- 6 Avoid Overfitting
- 6 Learning Rate
- Epochs
- 8 Regularization  $\lambda$
- 9 Batch Size
- Automation



### Review

DL Librarie

## Introduction

Symbolic Programming Graphs Neurons

## Eight Data Points Eight Million Points

Eight Million Points
Dense Nets
ConvNets

TF 2.0

## Project Improveme

Improvemen Where We Are Ten Tunings

Up Ne

- Initialization
- 2 Get Above Chance
- 3 Layers
- 4 Cost
- 6 Avoid Overfitting
- 6 Learning Rate
- 7 Epochs
- 8 Regularization  $\lambda$
- 9 Batch Size
- Automation



### Review

DL Librari

## Introduction Symbolic

Symbolic Programming Graphs Neurons

# Eight Data Points Eight Million Points Dansa Nate

Convinets

### Project Improveme

Improvemer Where We Are Ten Tunings

Up Ne

- 1 Initialization
- ② Get Above Chance
- 3 Layers
- 4 Cost
- 6 Avoid Overfitting
- **6** Learning Rate
- pochs
- 8 Regularization  $\lambda$
- 9 Batch Size
- Automation



- Initialization
- Get Above Chance
- 3 Layers
- 4 Cost
- 6 Avoid Overfitting
- 6 Learning Rate
- Epochs
- **8** Regularization  $\lambda$



### Review

DL Librari

### Introduction Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

### Project Improvement Where We Are Ten Tunings

Up Nex

- Initialization
- 2 Get Above Chance
- 3 Layers
- 4 Cost
- 6 Avoid Overfitting
- 6 Learning Rate
- Epochs
- 8 Regularization  $\lambda$
- Batch Size
- Automation



### Review

DL Librari

### Introductio Symbolic Programming Graphs

Fitting Models
Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.0

### Project Improvement Where We Are Ten Tunings

Up Nex

- Initialization
- 2 Get Above Chance
- 3 Layers
- 4 Cost
- 6 Avoid Overfitting
- **6** Learning Rate
- Epochs
- 8 Regularization  $\lambda$
- Batch Size
- Automation



## **Outline**

Review

DL Librarie

Introduction Symbolic Programming Graphs

Fitting Model
Eight Data Points
Eight Million Point
Dense Nets
ConvNets

TF 2.0

Project Improvemen Where We Are Ten Tunings

Up Next

- 1 Review Take-Home Exercise
- 2 Comparison of the Leading Deep Learning Libraries
  - 3 Introduction to TensorFlow Symbolic Programming Programming TensorFlow Graphs Neurons in TensorFlow
  - 4 Fitting Models
    Eight Data Points
    Eight Million Points
    Dense Nets
    Convolutional Nets
  - **5** TF 2.0
  - 6 Deep Learning Project IV: Improving Where We Are Ten Hyperparameter-Tuning Steps
  - Up Next: Advanced Topics



Units 7 and 8 — Tensorflow

### Review

DI Librari

### Introduction

Symbolic Programmin Graphs

## Fitting Mode

Eight Data Points
Eight Million Points
Dense Nets
ConvNets

TF 2.

## Project Improveme

Where We Are
Ten Tunings

Up Next

# Generative Adversarial Networks





Units 7 and 8 — Tensorflow

Up Next

## Reinforcement Learning



