

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# Deep Learning with TensorFlow

## Deep Learning — Units 7 & 8

Dr. Jon Krohn

jon@untapt.com

December 2nd, 2017

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics



# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# Outline

## Review

### DL Libraries

### Introduction

Symbolic  
Programming  
Graphs  
Neurons

### Fitting Models

Eight Data Points  
Eight Million Points

### Dense Nets

### ConvNets

### Project Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

Review

DL Libraries

Introduction

- Symbolic Programming
- Graphs
- Neurons

Fitting Models

- Eight Data Points
- Eight Million Points

Dense Nets

ConvNets

Project Improvement

- Where We Are
- Ten Tunings

Up Next

# Assessing Your Deep Learning Project III



# Assessing

## Your Deep Learning Project III

Where are you at with respect to the following?

### 1 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

# Assessing

## Your Deep Learning Project III

Where are you at with respect to the following?

### 1 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

# Assessing

## Your Deep Learning Project III

Where are you at with respect to the following?

### 1 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

# Assessing

## Your Deep Learning Project III

Where are you at with respect to the following?

### 1 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

# Assessing

## Your Deep Learning Project III

Where are you at with respect to the following?

### 1 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



# Assessing

## Your Deep Learning Project III

Where are you at with respect to the following?

### 1 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

# Assessing

## Your Deep Learning Project III

Where are you at with respect to the following?

### 1 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

# Assessing

## Your Deep Learning Project III

Where are you at with respect to the following?

### 1 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

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# Leading DL Libraries

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

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

Review

DL Libraries

**Introduction**

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming

Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# TensorFlow Graphs

Review

DL Libraries

Introduction

Symbolic  
Programming

Graphs

Neurons

Fitting Models

Eight Data Points

Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are

Ten Tunings

Up Next

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



Review

DL Libraries

Introduction

Symbolic  
Programming

Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# TensorFlow Graphs

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

Review

DL Libraries

Introduction

Symbolic  
Programming

Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

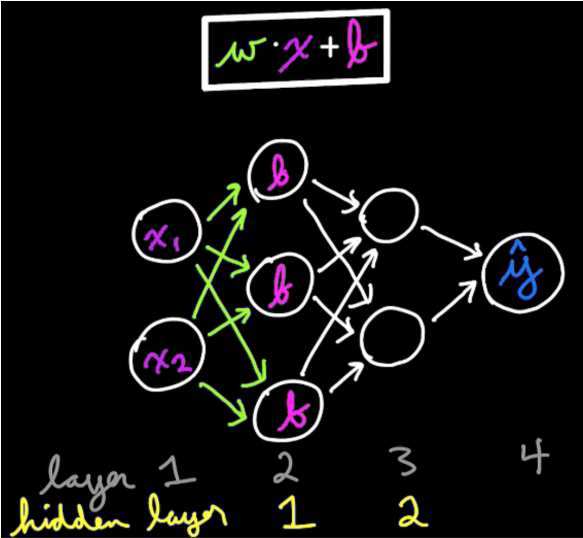
Up Next

# TensorFlow Graphs

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

# A Familiar Equation

$$w \cdot x + b$$

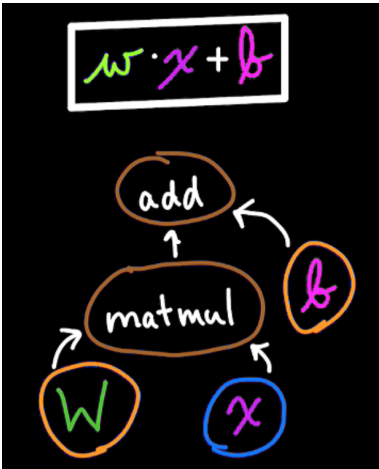


©2018 Pearson, Inc.

- Review
- DL Libraries
- Introduction
  - Symbolic Programming
  - Graphs
  - Neurons
- Fitting Models
  - Eight Data Points
  - Eight Million Points
- Dense Nets
- ConvNets
- Project Improvement
  - Where We Are
  - Ten Tunings
- Up Next

# TensorFlow Graphs

- Review
- DL Libraries
- Introduction
  - Symbolic Programming
  - Graphs
  - Neurons
- Fitting Models
  - Eight Data Points
  - Eight Million Points
- Dense Nets
- ConvNets
- Project Improvement
  - Where We Are
  - Ten Tunings
- Up Next



©2018 Pearson, Inc.

Review

DL Libraries

Introduction

Symbolic  
Programming

Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

Review

DL Libraries

Introduction

Symbolic  
Programming

**Graphs**

Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# TensorFlow Graph Programming

[ *first TensorFlow graphs notebook* ]

# 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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

Review

DL Libraries

Introduction

Symbolic  
Programming

Graphs

**Neurons**

Fitting Models

Eight Data Points

Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

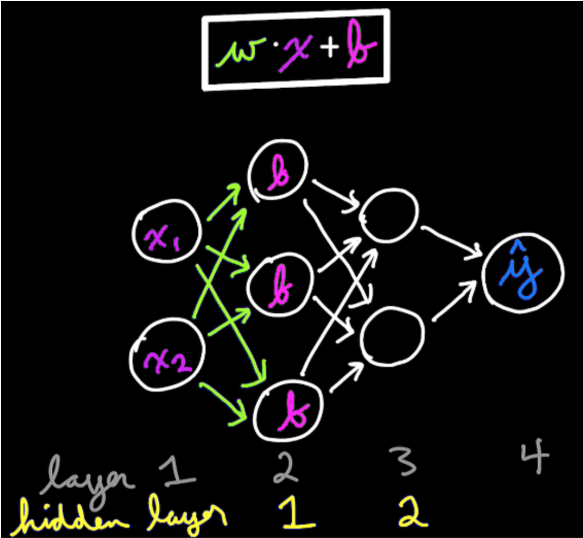
Where We Are

Ten Tunings

Up Next

# A Familiar Equation

$$w \cdot x + b$$



©2018 Pearson, Inc.



# Neurons in TensorFlow

## Programming

Review

DL Libraries

Introduction

Symbolic  
Programming

Graphs

**Neurons**

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

[ *first TensorFlow neurons notebook* ]

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# Fitting Eight Points

[ *point by point intro to TensorFlow notebook* ]

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# Fitting Eight Points

with Tensors

[ *tensor-fied intro to TensorFlow* notebook ]

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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**
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# Fitting Eight Million Points

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
**Eight Million Points**

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

[ *intro to TensorFlow times a million notebook* ]

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets**
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics



Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# Dense Nets

[ *intermediate net in TensorFlow notebook* ]

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

**Dense Nets**

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# Dense Nets

[ *dense net in TensorFlow* notebook ]

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# LeNet-5

LeCun et al. (1998)

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

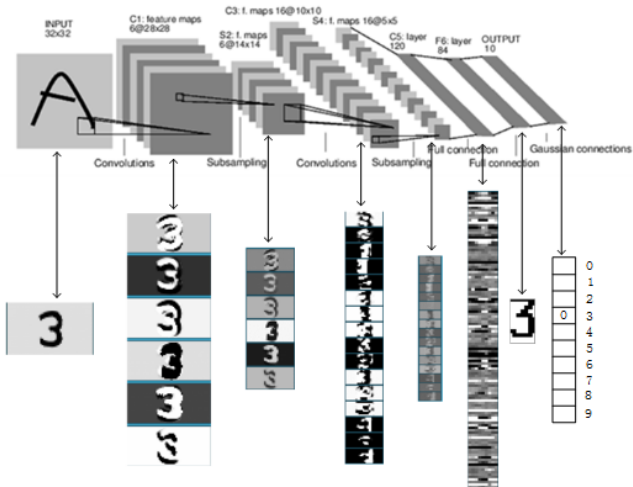
Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next



Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# LeNet-5

LeCun et al. (1998)

[ *LeNet in TensorFlow* notebook ]

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 **Deep Learning Project IV: Improving Where We Are**
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

Review

DL Libraries

Introduction

- Symbolic Programming
- Graphs
- Neurons

Fitting Models

- Eight Data Points
- Eight Million Points

Dense Nets

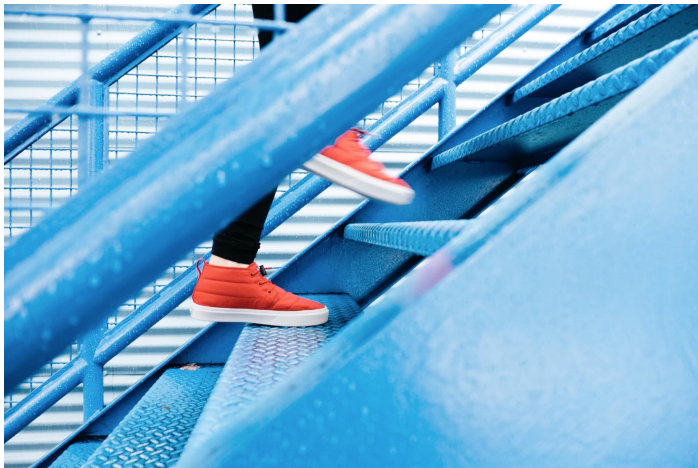
ConvNets

Project Improvement

- Where We Are
- Ten Tunings

Up Next

# Improving Your Deep Learning Project IV





# Improving Your Deep Learning Project IV

## 1 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

## 3 Improving performance & tuning hyperparameters in ten steps...

# Improving Your Deep Learning Project IV

## 1 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

## 3 Improving performance & tuning hyperparameters in ten steps...

# Improving

## Your Deep Learning Project IV

### 1 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

### 3 Improving performance & tuning hyperparameters in ten steps...

# Improving

## Your Deep Learning Project IV

### 1 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

### 3 Improving performance & tuning hyperparameters in ten steps...

# Improving

## Your Deep Learning Project IV

### 1 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

### 3 Improving performance & tuning hyperparameters in ten steps...

# Improving

## Your Deep Learning Project IV

- 1 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
- 3 Improving performance & tuning hyperparameters in ten steps...

# Improving

## Your Deep Learning Project IV

- 1 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
- 3 Improving performance & tuning hyperparameters in ten steps...

# Improving

## Your Deep Learning Project IV

### 1 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

### 3 Improving performance & tuning hyperparameters in ten steps...



# Improving

## Your Deep Learning Project IV

- 1 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
- 3 Improving performance & tuning hyperparameters in ten steps...

# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
**Ten Tunings**

Up Next

# 1. Initialization

Experiment with varying:

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
**Ten Tunings**

Up Next

# 1. Initialization

Experiment with varying:

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# 1. Initialization

Experiment with varying:

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

# 1. Initialization

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

...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
```

```
weight_dict = {
    'W_c1': tf.get_variable('W_c1',
                            [k_conv_1, k_conv_1, 1, n_conv_1], initializer=wt_init),
    'W_c2': tf.get_variable('W_c2',
                            [k_conv_2, k_conv_2, n_conv_1, n_conv_2], initializer=wt_init),
    'W_d1': tf.get_variable('W_d1',
                            [dense_inputs, n_dense], initializer=wt_init),
    'W_out': tf.get_variable('W_out',
                             [n_dense, n_classes], initializer=wt_init)
}
```

## 2. Get Above Chance

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

If your accuracy is below chance, try:

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

## 2. Get Above Chance

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

If your accuracy is below chance, try:

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



## 2. Get Above Chance

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

If your accuracy is below chance, try:

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
**Ten Tunings**

Up Next

## 3. Layers

Experiment with varying:

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
**Ten Tunings**

Up Next

## 3. Layers

Experiment with varying:

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

## 3. Layers

Experiment with varying:

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

## 4. Cost

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up 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)
```

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

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 Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

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

## 5. Avoid Overfitting

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

...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:

```
# max pooling layer:
pool_size = 2
mp_layer_dropout = 0.25

# dense layer:
n_dense = 128
dense_layer_dropout = 0.5
```

```
# convolutional and max-pooling layers:
conv_1 = conv2d(square_x, weights['W_c1'], biases['b_c1'])
conv_2 = conv2d(conv_1, weights['W_c2'], biases['b_c2'])
pool_1 = maxpooling2d(conv_2, mp_size)
pool_1 = tf.nn.dropout(pool_1, 1-mp_dropout)

# dense layer:
flat = tf.reshape(pool_1, [-1, weights['W_d1'].get_shape().as_list()[0]])
dense_1 = dense(flat, weights['W_d1'], biases['b_d1'])
dense_1 = tf.nn.dropout(dense_1, 1-dense_dropout)
```



## 6. Learning Rate

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up 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)
```

# 7. Epochs

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

...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

```
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
```

```
with tf.Session() as session:  
    session.run(initializer_op)  
  
    print("Training for", epochs, "epochs.")  
  
    # loop over epochs:  
    for epoch in range(epochs):
```

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

## 8. Regularization $\lambda$

If using L1 or L2 regularization, consider:

- adjusting  $\lambda$  by orders of magnitude

## 9. Batch Size

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

...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

```
epochs = 20  
batch_size = 128  
display_progress = 40 # after this i  
wt_init = tf.contrib.layers.xavier_
```

```
# loop over all batches of the epoch:  
n_batches = int(mnist.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+1, " of ", n_batches, " in epoch ", epoch+1, ".", sep='')  
  
    batch_x, batch_y = mnist.train.next_batch(batch_size)
```

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

## 10. Automation

For grid search of hyperparameters, consider:

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

## 10. Automation

For grid search of hyperparameters, consider:

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# All together now...

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# All together now...

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



Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# All together now...

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# All together now...

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# All together now...

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# All together now...

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# All together now...

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# All together now...

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# All together now...

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

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

## All together now...

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



# Outline

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

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
- 5 Dense Nets
- 6 Convolutional Nets
- 7 Deep Learning Project IV: Improving Where We Are
  - Ten Hyperparameter-Tuning Steps
- 8 Up Next: Advanced Topics

Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# Generative Adversarial Networks

December 16th



Review

DL Libraries

Introduction

Symbolic  
Programming  
Graphs  
Neurons

Fitting Models

Eight Data Points  
Eight Million Points

Dense Nets

ConvNets

Project  
Improvement

Where We Are  
Ten Tunings

Up Next

# Reinforcement Learning

December 16th

