



DEEP
LEARNING
INSTITUTE

LEARNING DEEP LEARNING

Theory and Practice of Neural Networks, Computer Vision,
Natural Language Processing, and Transformers
Using TensorFlow



MAGNUS EKMAN

Learning Deep Learning

Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, Natural Language Processing, and Transformers Using TensorFlow

Table of Contents

Cover

Half Title

Title Page

Copyright Page

Contents

Foreword

Foreword

Preface

Acknowledgments

About the Author

1 THE ROSENBLATT PERCEPTRON

Example of a Two-Input Perceptron

The Perceptron Learning Algorithm

Limitations of the Perceptron

Combining Multiple Perceptrons

Implementing Perceptrons with Linear Algebra

Vector Notation

Dot Product

Extending the Vector to a 2D Matrix

Table of Contents

Matrix-Vector Multiplication

Matrix-Matrix Multiplication

Summary of Vector and Matrix Operations Used for Perceptrons

Dot Product as a Matrix Multiplication

Extending to Multidimensional Tensors

Geometric Interpretation of the Perceptron

Understanding the Bias Term

Concluding Remarks on the Perceptron

2 GRADIENT-BASED LEARNING

Intuitive Explanation of the Perceptron Learning Algorithm

Derivatives and Optimization Problems

Solving a Learning Problem with Gradient Descent

Gradient Descent for Multidimensional Functions

Constants and Variables in a Network

Analytic Explanation of the Perceptron Learning Algorithm

Geometric Description of the Perceptron Learning Algorithm

Revisiting Different Types of Perceptron Plots

Using a Perceptron to Identify Patterns

Concluding Remarks on Gradient-Based Learning

3 SIGMOID NEURONS AND BACKPROPAGATION

Modified Neurons to Enable Gradient Descent for Multilevel Networks

Which Activation Function Should We Use?

Function Composition and the Chain Rule

Using Backpropagation to Compute the Gradient

Forward Pass

Backward Pass

Weight Adjustment

Table of Contents

Backpropagation with Multiple Neurons per Layer
Programming Example: Learning the XOR Function
Network Architectures
Concluding Remarks on Backpropagation

4 FULLY CONNECTED NETWORKS APPLIED TO MULTICLASS CLASSIFICATION

Introduction to Datasets Used When Training Networks
 Exploring the Dataset
 Human Bias in Datasets
 Training Set, Test Set, and Generalization
 Hyperparameter Tuning and Test Set Information Leakage
Training and Inference
Extending the Network and Learning Algorithm to Do Multiclass Classification
Network for Digit Classification
Loss Function for Multiclass Classification
Programming Example: Classifying Handwritten Digits
Mini-Batch Gradient Descent
Concluding Remarks on Multiclass Classification

5 TOWARD DL: FRAMEWORKS AND NETWORK TWEAKS

Programming Example: Moving to a DL Framework
The Problem of Saturated Neurons and Vanishing Gradients
Initialization and Normalization Techniques to Avoid Saturated Neurons
 Weight Initialization
 Input Standardization
 Batch Normalization
Cross-Entropy Loss Function to Mitigate Effect of Saturated Output Neurons
 Computer Implementation of the Cross-Entropy Loss Function

Table of Contents

Different Activation Functions to Avoid Vanishing Gradient in Hidden Layers

Variations on Gradient Descent to Improve Learning

Experiment: Tweaking Network and Learning Parameters

Hyperparameter Tuning and Cross-Validation

Using a Validation Set to Avoid Overfitting

Cross-Validation to Improve Use of Training Data

Concluding Remarks on the Path Toward Deep Learning

6 FULLY CONNECTED NETWORKS APPLIED TO REGRESSION

Output Units

Logistic Unit for Binary Classification

Softmax Unit for Multiclass Classification

Linear Unit for Regression

The Boston Housing Dataset

Programming Example: Predicting House Prices with a DNN

Improving Generalization with Regularization

Experiment: Deeper and Regularized Models for House Price Prediction

Concluding Remarks on Output Units and Regression Problems

7 CONVOLUTIONAL NEURAL NETWORKS APPLIED TO IMAGE CLASSIFICATION

The CIFAR-10 Dataset

Characteristics and Building Blocks for Convolutional Layers

Combining Feature Maps into a Convolutional Layer

Combining Convolutional and Fully Connected Layers into a Network

Effects of Sparse Connections and Weight Sharing

Programming Example: Image Classification with a Convolutional Network

Table of Contents

Concluding Remarks on Convolutional Networks

8 DEEPER CNNs AND PRETRAINED MODELS

VGGNet

GoogLeNet

ResNet

Programming Example: Use a Pretrained ResNet Implementation

Transfer Learning

Backpropagation for CNN and Pooling

Data Augmentation as a Regularization Technique

Mistakes Made by CNNs

Reducing Parameters with Depthwise Separable Convolutions

Striking the Right Network Design Balance with EfficientNet

Concluding Remarks on Deeper CNNs

9 PREDICTING TIME SEQUENCES WITH RECURRENT NEURAL NETWORKS

Limitations of Feedforward Networks

Recurrent Neural Networks

Mathematical Representation of a Recurrent Layer

Combining Layers into an RNN

Alternative View of RNN and Unrolling in Time

Backpropagation Through Time

Programming Example: Forecasting Book Sales

Standardize Data and Create Training Examples

Creating a Simple RNN

Comparison with a Network Without Recurrence

Extending the Example to Multiple Input Variables

Dataset Considerations for RNNs

Table of Contents

Concluding Remarks on RNNs

10 LONG SHORT-TERM MEMORY

Keeping Gradients Healthy

Introduction to LSTM

LSTM Activation Functions

Creating a Network of LSTM Cells

Alternative View of LSTM

Related Topics: Highway Networks and Skip Connections

Concluding Remarks on LSTM

11 TEXT AUTOCOMPLETION WITH LSTM AND BEAM SEARCH

Encoding Text

Longer-Term Prediction and Autoregressive Models

Beam Search

Programming Example: Using LSTM for Text Autocompletion

Bidirectional RNNs

Different Combinations of Input and Output Sequences

Concluding Remarks on Text Autocompletion with LSTM

12 NEURAL LANGUAGE MODELS AND WORD EMBEDDINGS

Introduction to Language Models and Their Use Cases

Examples of Different Language Models

n-Gram Model

Skip-Gram Model

Neural Language Model

Benefit of Word Embeddings and Insight into How They Work

Word Embeddings Created by Neural Language Models

Programming Example: Neural Language Model and Resulting Embeddings

King - Man + Woman! = Queen

Table of Contents

King - Man + Woman ! = Queen

Language Models, Word Embeddings, and Human Biases

Related Topic: Sentiment Analysis of Text

Bag-of-Words and Bag-of-N-Grams

Similarity Metrics

Combining BoW and DL

Concluding Remarks on Language Models and Word Embeddings

13 WORD EMBEDDINGS FROM word2vec AND GloVe

Using word2vec to Create Word Embeddings Without a Language Model

Reducing Computational Complexity Compared to a Language Model

Continuous Bag-of-Words Model

Continuous Skip-Gram Model

Optimized Continuous Skip-Gram Model to Further Reduce Computational Complexity

Additional Thoughts on word2vec

word2vec in Matrix Form

Wrapping Up word2vec

Programming Example: Exploring Properties of GloVe Embeddings

Concluding Remarks on word2vec and GloVe

14 SEQUENCE-TO-SEQUENCE NETWORKS AND NATURAL LANGUAGE TRANSLATION

Encoder-Decoder Model for Sequence- to-Sequence Learning

Introduction to the Keras Functional API

Programming Example: Neural Machine Translation

Experimental Results

Properties of the Intermediate Representation

Concluding Remarks on Language Translation

Table of Contents

15 ATTENTION AND THE TRANSFORMER

Rationale Behind Attention

Attention in Sequence-to-Sequence Networks

Computing the Alignment Vector

Mathematical Notation and Variations on the Alignment Vector

Attention in a Deeper Network

Additional Considerations

Alternatives to Recurrent Networks

Self-Attention

Multi-head Attention

The Transformer

Concluding Remarks on the Transformer

16 ONE-TO-MANY NETWORK FOR IMAGE CAPTIONING

Extending the Image Captioning Network with Attention

Programming Example: Attention-Based Image Captioning

Concluding Remarks on Image Captioning

17 MEDLEY OF ADDITIONAL TOPICS

Autoencoders

Use Cases for Autoencoders

Other Aspects of Autoencoders

Programming Example: Autoencoder for Outlier Detection

Multimodal Learning

Taxonomy of Multimodal Learning

Programming Example: Classic Classification with Multimodal Input Data

Multitask Learning

Why to Implement Multitask Learning

How to Implement Multitask Learning

Other Aspects and Variations on the Basic Implementation

Table of Contents

Programming Example: Multiclass Classification and Question Answering with a Single Network

Process for Tuning a Network

When to Collect More Training Data

Neural Architecture Search

Key Components of Neural Architecture Search

Programming Example: Searching for an Architecture for CIFAR-10 Classification

Implications of Neural Architecture Search

Concluding Remarks

18 SUMMARY AND NEXT STEPS

Things You Should Know by Now

Ethical AI and Data Ethics

Problems to Look Out For

Checklist of Questions

Things You Do Not Yet Know

Reinforcement Learning

Variational Autoencoders and Generative Adversarial Networks

Neural Style Transfer

Recommender Systems

Models for Spoken Language

Next Steps

Appendix A: LINEAR REGRESSION AND LINEAR CLASSIFIERS

Linear Regression as a Machine Learning Algorithm

Univariate Linear Regression

Multivariate Linear Regression

Modeling Curvature with a Linear Function

Computing Linear Regression Coefficients

Table of Contents

Classification with Logistic Regression
Classifying XOR with a Linear Classifier
Classification with Support Vector Machines
Evaluation Metrics for a Binary Classifier

Appendix B: OBJECT DETECTION AND SEGMENTATION

Object Detection

R-CNN
Fast R-CNN
Faster R-CNN

Semantic Segmentation

Upsampling Techniques
Deconvolution Network
U-Net

Instance Segmentation with Mask R-CNN

Appendix C: WORD EMBEDDINGS BEYOND word2vec AND GloVe

Wordpieces

FastText

Character-Based Method

ELMo

Related Work

Appendix D: GPT, BERT, AND RoBERTa

GPT

BERT

Masked Language Model Task
Next-Sentence Prediction Task
BERT Input and Output Representations
Applying BERT to NLP Tasks

Table of Contents

RoBERTa

Historical Work Leading Up to GPT and BERT

Other Models Based on the Transformer

Appendix E: NEWTON-RAPHSON VERSUS GRADIENT DESCENT

Newton-Raphson Root-Finding Method

Newton-Raphson Applied to Optimization Problems

Relationship Between Newton-Raphson and Gradient Descent

Appendix F: MATRIX IMPLEMENTATION OF DIGIT CLASSIFICATION NETWORK

Single Matrix

Mini-Batch Implementation

Appendix G: RELATING CONVOLUTIONAL LAYERS TO MATHEMATICAL CONVOLUTION

Appendix H: GATED RECURRENT UNITS

Alternative GRU Implementation

Network Based on the GRU

Appendix I: SETTING UP A DEVELOPMENT ENVIRONMENT

Python

Programming Environment

Jupyter Notebook

Using an Integrated Development Environment

Programming Examples

Supporting Spreadsheet

Datasets

MNIST

Bookstore Sales Data from US Census Bureau

Frankenstein from Project Gutenberg

Table of Contents

GloVe Word Embeddings

Anki Bilingual Sentence Pairs

COCO

Installing a DL Framework

System Installation

Virtual Environment Installation

GPU Acceleration

Docker Container

Using a Cloud Service

TensorFlow Specific Considerations

Key Differences Between PyTorch and TensorFlow

Need to Write Our Own Fit/Training Function

Explicit Moves of Data Between NumPy and PyTorch

Explicit Transfer of Data Between CPU and GPU

Explicitly Distinguishing Between Training and Inference

Sequential versus Functional API

Lack of Compile Function

Recurrent Layers and State Handling

Cross-Entropy Loss

View/Reshape

Appendix J: CHEAT SHEETS

Works Cited

Index