

Table of Contents

Preface

1

Section 1: TensorFlow 2 and Deep Learning Applied to Computer Vision

Chapter 1: Computer Vision and Neural Networks	11
Technical requirements	11
Computer vision in the wild	12
Introducing computer vision	12
Main tasks and their applications	13
Content recognition	13
Object classification	13
Object identification	14
Object detection and localization	15
Object and instance segmentation	16
Pose estimation	17
Video analysis	17
Instance tracking	18
Action recognition	18
Motion estimation	19
Content-aware image edition	19
Scene reconstruction	20
A brief history of computer vision	20
First steps to initial successes	20
Underestimating the perception task	21
Hand-crafting local features	21
Adding some machine learning on top	23
Rise of deep learning	24
Early attempts and failures	24
Rise and fall of the perceptron	25
Too heavy to scale	25
Reasons for the comeback	25
The internet – the new El Dorado of data science	26
More power than ever	26
Deep learning or the rebranding of artificial neural networks	27
What makes learning deep?	27
Deep learning era	28
Getting started with neural networks	29
Building a neural network	29
Imitating neurons	29
Biological inspiration	29
Mathematical model	30
Implementation	32
Layering neurons together	33

Mathematical model	34
Implementation	35
Applying our network to classification	36
Setting up the task	37
Implementing the network	38
Training a neural network	40
Learning strategies	40
Supervised learning	40
Unsupervised learning	41
Reinforcement learning	41
Teaching time	42
Evaluating the loss	42
Backpropagating the loss	43
Teaching our network to classify	46
Training considerations – underfitting and overfitting	49
Summary	50
Questions	50
Further reading	50
Chapter 2: TensorFlow Basics and Training a Model	51
Technical requirements	51
Getting started with TensorFlow 2 and Keras	52
Introducing TensorFlow	52
TensorFlow's main architecture	52
Introducing Keras	53
A simple computer vision model using Keras	54
Preparing the data	54
Building the model	55
Training the model	56
Model performance	57
TensorFlow 2 and Keras in detail	58
Core concepts	58
Introducing tensors	58
TensorFlow graphs	59
Comparing lazy execution to eager execution	60
Creating graphs in TensorFlow 2	61
Introducing TensorFlow AutoGraph and tf.function	62
Backpropagating errors using the gradient tape	63
Keras models and layers	64
Sequential and functional APIs	65
Callbacks	65
Advanced concepts	66
How tf.function works	66
Variables in TensorFlow 2	67
Distribution strategies	68
Using the Estimator API	69
Available pre-made Estimators	69
Training a custom Estimator	70
The TensorFlow ecosystem	70
TensorBoard	71

TensorFlow Addons and TensorFlow Extended	72
TensorFlow Lite and TensorFlow.js	73
Where to run your model	73
On a local machine	74
On a remote machine	74
On Google Cloud	75
Summary	75
Questions	76
Chapter 3: Modern Neural Networks	77
Technical requirements	77
Discovering convolutional neural networks	77
Neural networks for multidimensional data	78
Problems with fully connected networks	78
An explosive number of parameters	78
A lack of spatial reasoning	78
Introducing CNNs	79
CNN operations	80
Convolutional layers	80
Concept	80
Properties	82
Hyperparameters	83
TensorFlow/Keras methods	85
Pooling layers	89
Concept and hyperparameters	89
TensorFlow/Keras methods	90
Fully connected layers	92
Usage in CNNs	92
TensorFlow/Keras methods	93
Effective receptive field	93
Definitions	94
Formula	95
CNNs with TensorFlow	95
Implementing our first CNN	96
LeNet-5 architecture	96
TensorFlow and Keras implementations	97
Application to MNIST	99
Refining the training process	99
Modern network optimizers	100
Gradient descent challenges	100
Training velocity and trade-off	100
Suboptimal local minima	101
A single hyperparameter for heterogeneous parameters	102
Advanced optimizers	102
Momentum algorithms	102
The Ada family	104
Regularization methods	105
Early stopping	106
L1 and L2 regularization	106
Principles	106

TensorFlow and Keras implementations	107
Dropout	110
Definition	110
TensorFlow and Keras methods	111
Batch normalization	111
Definition	111
TensorFlow and Keras methods	112
Summary	112
Questions	113
Further reading	113
Section 2: State-of-the-Art Solutions for Classic Recognition Problems	
Chapter 4: Influential Classification Tools	117
Technical requirements	117
Understanding advanced CNN architectures	118
VGG – a standard CNN architecture	118
Overview of the VGG architecture	118
Motivation	118
Architecture	119
Contributions – standardizing CNN architectures	120
Replacing large convolutions with multiple smaller ones	120
Increasing the depth of the feature maps	121
Augmenting data with scale jittering	121
Replacing fully connected layers with convolutions	122
Implementations in TensorFlow and Keras	123
The TensorFlow model	123
The Keras model	123
GoogLeNet and the inception module	124
Overview of the GoogLeNet architecture	124
Motivation	124
Architecture	125
Contributions – popularizing larger blocks and bottlenecks	127
Capturing various details with inception modules	127
Using 1 x 1 convolutions as bottlenecks	128
Pooling instead of fully connecting	129
Fighting vanishing gradient with intermediary losses	129
Implementations in TensorFlow and Keras	130
Inception module with the Keras Functional API	130
TensorFlow model and TensorFlow Hub	132
The Keras model	133
ResNet – the residual network	134
Overview of the ResNet architecture	135
Motivation	135
Architecture	136
Contributions – forwarding the information more deeply	137
Estimating a residual function instead of a mapping	137
Going ultra-deep	138
Implementations in TensorFlow and Keras	138
Residual blocks with the Keras Functional API	139

The TensorFlow model and TensorFlow Hub	139
The Keras model	140
Leveraging transfer learning	140
Overview	141
Definition	141
Human inspiration	141
Motivation	142
Transferring CNN knowledge	144
Use cases	144
Similar tasks with limited training data	145
Similar tasks with abundant training data	145
Dissimilar tasks with abundant training data	146
Dissimilar tasks with limited training data	146
Transfer learning with TensorFlow and Keras	146
Model surgery	147
Removing layers	147
Grafting layers	148
Selective training	148
Restoring pretrained parameters	148
Freezing layers	149
Summary	149
Questions	150
Further reading	150
Chapter 5: Object Detection Models	151
Technical requirements	151
Introducing object detection	151
Background	152
Applications	152
Brief history	152
Evaluating the performance of a model	153
Precision and recall	154
Precision-recall curve	154
Average precision and mean average precision	156
Average precision threshold	156
A fast object detection algorithm – YOLO	157
Introducing YOLO	158
Strengths and limitations of YOLO	158
YOLO's main concepts	158
Inferring with YOLO	160
The YOLO backbone	160
YOLO's layers output	161
Introducing anchor boxes	162
How YOLO refines anchor boxes	164
Post-processing the boxes	165
NMS	167
YOLO inference summarized	168
Training YOLO	169
How the YOLO backbone is trained	170
YOLO loss	170

Bounding box loss	171
Object confidence loss	172
Classification loss	173
Full YOLO loss	173
Training techniques	173
Faster R-CNN – a powerful object detection model	174
Faster R-CNN's general architecture	175
Stage 1 – Region proposals	175
Stage 2 – Classification	177
Faster R-CNN architecture	177
RoI pooling	178
Training Faster R-CNN	180
Training the RPN	180
The RPN loss	181
Fast R-CNN loss	182
Training regimen	182
TensorFlow Object Detection API	183
Using a pretrained model	183
Training on a custom dataset	183
Summary	184
Questions	184
Further reading	184
Chapter 6: Enhancing and Segmenting Images	185
Technical requirements	185
Transforming images with encoders-decoders	186
Introduction to encoders-decoders	186
Encoding and decoding	186
Auto-encoding	188
Purpose	189
Basic example – image denoising	191
Simplistic fully connected AE	191
Application to image denoising	191
Convolutional encoders-decoders	192
Unpooling, transposing, and dilating	192
Transposed convolution (deconvolution)	192
Unpooling	195
Upsampling and resizing	196
Dilated/atrous convolution	197
Example architectures – FCN and U-Net	199
Fully convolutional networks	199
U-Net	201
Intermediary example – image super-resolution	202
FCN implementation	202
Application to upscaling images	203
Understanding semantic segmentation	203
Object segmentation with encoders-decoders	204
Overview	204
Decoding as label maps	204
Training with segmentation losses and metrics	206

Post-processing with conditional random fields	208
Advanced example – image segmentation for self-driving cars	209
Task presentation	209
Exemplary solution	210
The more difficult case of instance segmentation	210
From object segmentation to instance segmentation	210
Respecting boundaries	210
Post-processing into instance masks	211
From object detection to instance segmentation – Mask R-CNN	212
Applying semantic segmentation to bounding boxes	212
Building an instance segmentation model with Faster-RCNN	213
Summary	214
Questions	215
Further reading	215
Section 3: Advanced Concepts and New Frontiers of Computer Vision	
Chapter 7: Training on Complex and Scarce Datasets	219
Technical requirements	220
Efficient data serving	220
Introducing the TensorFlow Data API	220
Intuition behind the TensorFlow Data API	220
Feeding fast and data-hungry models	220
Inspiration from lazy structures	221
Structure of TensorFlow data pipelines	221
Extract, Transform, Load	222
API interface	223
Setting up input pipelines	224
Extracting (from tensors, text files, TFRecord files, and more)	224
From NumPy and TensorFlow data	224
From files	225
From other inputs (generator, SQL database, range, and others)	225
Transforming the samples (parsing, augmenting, and more)	226
Parsing images and labels	226
Parsing TFRecord files	227
Editing samples	228
Transforming the datasets (shuffling, zipping, parallelizing, and more)	228
Structuring datasets	228
Merging datasets	230
Loading	231
Optimizing and monitoring input pipelines	231
Following best practices for optimization	231
Parallelizing and prefetching	231
Fusing operations	233
Passing options to ensure global properties	234
Monitoring and reusing datasets	235
Aggregating performance statistics	235
Caching and reusing datasets	236
How to deal with data scarcity	237
Augmenting datasets	237

Overview	237
Why augment datasets?	238
Considerations	239
Augmenting images with TensorFlow	241
TensorFlow Image module	241
Example – augmenting images for our autonomous driving application	242
Rendering synthetic datasets	243
Overview	243
Rise of 3D databases	243
Benefits of synthetic data	244
Generating synthetic images from 3D models	245
Rendering from 3D models	246
Post-processing synthetic images	247
Problem – realism gap	248
Leveraging domain adaptation and generative models (VAEs and GANs)	249
Training models to be robust to domain changes	249
Supervised domain adaptation	250
Unsupervised domain adaptation	250
Domain randomization	253
Generating larger or more realistic datasets with VAEs and GANs	254
Discriminative versus generative models	254
VAEs	255
GANs	258
Augmenting datasets with conditional GANs	260
Summary	262
Questions	262
Further reading	262
Chapter 8: Video and Recurrent Neural Networks	263
Technical requirements	263
Introducing RNNs	264
Basic formalism	264
General understanding of RNNs	266
Learning RNN weights	267
Backpropagation through time	268
Truncated backpropagation	269
Long short-term memory cells	270
LSTM general principles	270
LSTM inner workings	271
Classifying videos	273
Applying computer vision to video	273
Classifying videos with an LSTM	274
Extracting features from videos	276
Training the LSTM	281
Defining the model	281
Loading the data	282
Training the model	283
Summary	283
Questions	283
Further reading	284

Chapter 9: Optimizing Models and Deploying on Mobile Devices	285
Technical requirements	285
Optimizing computational and disk footprints	286
Measuring inference speed	286
Measuring latency	287
Using tracing tools to understand computational performance	287
Improving model inference speed	289
Optimizing for hardware	289
Optimizing on CPUs	290
Optimizing on GPUs	290
Optimizing on specialized hardware	290
Optimizing input	291
Optimizing post-processing	291
When the model is still too slow	292
Interpolating and tracking	292
Model distillation	292
Reducing model size	293
Quantization	294
Channel pruning and weight sparsification	294
On-device machine learning	294
Considerations of on-device machine learning	295
Benefits of on-device ML	296
Latency	296
Privacy	296
Cost	296
Limitations of on-device ML	297
Practical on-device computer vision	297
On-device computer vision particularities	297
Generating a SavedModel	298
Generating a frozen graph	298
Importance of preprocessing	299
Example app – recognizing facial expressions	300
Introducing MobileNet	301
Deploying models on-device	301
Running on iOS devices using Core ML	302
Converting from TensorFlow or Keras	302
Loading the model	304
Using the model	304
Running on Android using TensorFlow Lite	305
Converting the model from TensorFlow or Keras	306
Loading the model	306
Using the model	307
Running in the browser using TensorFlow.js	309
Converting the model to the TensorFlow.js format	309
Using the model	310
Running on other devices	311
Summary	312
Questions	312
Migrating from TensorFlow 1 to TensorFlow 2	313

Table of Contents

Assessments	329
Other Books You May Enjoy	339
Index	343