# Contents

0.1 Overview of the book . . . . . . . . . . 11

**1 Introduction [1] 15**

1.1 Teaching a computer to distinguish cats from dogs . . . . . . . . . . 15

1.1.1 The pipeline of a typical machine learning problem . . . . . . . . . . 20

1.2 Predictive learning problems . . . . . . . . . . 21

1.2.1 Regression . . . . . . . . . . 21

1.2.2 Classification . . . . . . . . . . 24

1.3 Feature design . . . . . . . . . . 27

1.4 Numerical optimization . . . . . . . . . . 31

1.5 Summary . . . . . . . . . . 33

**I The basics 35**

**2 Fundamentals of numerical optimization [2] 38**

2.1 Calculus defined optimality . . . . . . . . . . 38

2.1.1 Taylor series approximations . . . . . . . . . . 38

2.1.2 The first order condition for optimality . . . . . . . . . . 40

2.1.3 The convenience of convexity . . . . . . . . . . 42

[1] This document is part of a book currently under development titled "Machine Learning Refined" (Cambridge University Press, late 2016) by Jeremy Watt, Reza Borhani, and Aggelos Katsaggelos. Please do not distribute. Feedback regarding any errors, comments on substance and style, recommendations, etc. is greatly appreciated! Contact: jermwatt@gmail.edu

[2] This document is part of a book currently under development titled "Machine Learning Refined" (Cambridge University Press, late 2016) by Jeremy Watt, Reza Borhani, and Aggelos Katsaggelos. Please do not distribute. Feedback regarding any errors, comments on substance and style, recommendations, etc. is greatly appreciated! Contact: jermwatt@gmail.edu

2.2 Numerical methods for optimization . . . . . . . . . . . . . . . . . . . . . 44
2.2.1 The big picture . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
2.2.2 Stopping condition . . . . . . . . . . . . . . . . . . . . . . . . . 45
2.2.3 Gradient descent . . . . . . . . . . . . . . . . . . . . . . . . . . 47
2.2.4 Newton's method . . . . . . . . . . . . . . . . . . . . . . . . . 50
2.3 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
2.4 Exercises . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

**3 Regression[3] 66**
3.1 The basics of linear regression . . . . . . . . . . . . . . . . . . . . . . 66
3.1.1 Notation and modeling . . . . . . . . . . . . . . . . . . . . . . . 67
3.1.2 The Least Squares cost function for linear regression . . . . . . . . . 68
3.1.3 Minimization of the Least Squares cost function . . . . . . . . . . . . 70
3.1.4 The efficacy of a learned model . . . . . . . . . . . . . . . . . . . 72
3.1.5 Predicting the value of new input data . . . . . . . . . . . . . . . . 72
3.2 Knowledge-driven feature design for regression . . . . . . . . . . . . . . 73
3.2.1 General conclusions . . . . . . . . . . . . . . . . . . . . . . . . 77
3.3 Nonlinear regression and $\ell_2$ regularization . . . . . . . . . . . . . . . . . 78
3.3.1 Logistic regression . . . . . . . . . . . . . . . . . . . . . . . . . 79
3.3.2 Non-convex cost functions and $\ell_2$ regularization . . . . . . . . . . . 81
3.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
3.5 Chapter exercises . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85

**4 Classification [4] 97**
4.1 The perceptron cost functions . . . . . . . . . . . . . . . . . . . . . . 97
4.1.1 The basic perceptron model . . . . . . . . . . . . . . . . . . . . . 98
4.1.2 The softmax cost function . . . . . . . . . . . . . . . . . . . . . . 100
4.1.3 The margin perceptron . . . . . . . . . . . . . . . . . . . . . . . 103

[3] This document is part of a book currently under development titled "Machine Learning Refined" (Cambridge University Press, late 2016) by Jeremy Watt, Reza Borhani, and Aggelos Katsaggelos. Please do not distribute. Feedback regarding any errors, comments on substance and style, recommendations, etc. is greatly appreciated! Contact: jermwatt@gmail.edu

[4] This document is part of a book currently under development titled "Machine Learning Refined" (Cambridge University Press, late 2016) by Jeremy Watt, Reza Borhani, and Aggelos Katsaggelos. Please do not distribute. Feedback regarding any errors, comments on substance and style, recommendations, etc. is greatly appreciated! Contact: jermwatt@gmail.edu

4.1.4 Differentiable approximations to the margin perceptron . . . . . . . 105
4.1.5 The accuracy of a learned classifier . . . . . . . . . . . . . . . . . . . . 107
4.1.6 Predicting the value of new input data . . . . . . . . . . . . . . . . . . 108
4.1.7 Which cost function produces the best results? . . . . . . . . . . . . 108
4.1.8 The connection between the perceptron and counting costs . . . . . 111
4.2 The logistic regression perspective on the softmax cost . . . . . . . . . . . 112
4.2.1 Step functions and classification . . . . . . . . . . . . . . . . . . . . . 113
4.2.2 Convex logistic regression . . . . . . . . . . . . . . . . . . . . . . . . 113
4.3 The support vector machine perspective on the margin perceptron . . . . . 117
4.3.1 A quest for the hyperplane with maximum margin . . . . . . . . . . . 117
4.3.2 The hard-margin SVM problem . . . . . . . . . . . . . . . . . . . . . . 119
4.3.3 The soft-margin SVM problem . . . . . . . . . . . . . . . . . . . . . . 119
4.3.4 Support Vector Machines and logistic regression . . . . . . . . . . . . 121
4.4 Multiclass classification . . . . . . . . . . . . . . . . . . . . . . . . . . . . 121
4.4.1 One-versus-All multiclass classification . . . . . . . . . . . . . . . . . 122
4.4.2 Multiclass softmax classification . . . . . . . . . . . . . . . . . . . . . 127
4.4.3 The accuracy of a learned multiclass classifier . . . . . . . . . . . . . 130
4.4.4 Which multiclass classification scheme works best? . . . . . . . . . . 131
4.5 Knowledge-driven feature design for classification . . . . . . . . . . . . . . 131
4.5.1 General conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . 134
4.6 Histogram features for real data types . . . . . . . . . . . . . . . . . . . . 135
4.6.1 Histogram features for text data . . . . . . . . . . . . . . . . . . . . . 136
4.6.2 Histogram features for image data . . . . . . . . . . . . . . . . . . . . 139
4.6.3 Histogram features for audio data . . . . . . . . . . . . . . . . . . . . 143
4.7 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 145
4.8 Chapter exercises . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 146

**II Automatic feature design 158**

**5 Automatic feature design for regression [5] 161**
5.1 Automatic feature design for the ideal regression scenario . . . . . . . . . . 162

[5] This document is part of a book currently under development titled “Machine Learning Refined” (Cambridge University Press, late 2016) by Jeremy Watt, Reza Borhani, and Aggelos Katsaggelos. Please do not distribute. Feedback regarding any errors, comments on substance and style, recommendations, etc. is greatly appreciated! Contact: jermwatt@gmail.edu

5.1.1 Vector approximation . . . . . 163
5.1.2 From vectors to continuous functions . . . . . 163
5.1.3 Continuous function approximation . . . . . 165
5.1.4 Common bases for continuous function approximation . . . . . 166
5.1.5 Recovering weights . . . . . 171
5.1.6 Graphical representation of a neural network* . . . . . 172
5.2 Automatic feature design for the real regression scenario . . . . . 172
5.2.1 Approximation of discretized continuous functions . . . . . 173
5.2.2 The real regression scenario . . . . . 174
5.3 Cross-validation for regression . . . . . 178
5.3.1 Diagnosing the problem of overfitting/underfitting . . . . . 181
5.3.2 Hold out cross-validation . . . . . 181
5.3.3 Hold out calculations . . . . . 183
5.3.4 *k*-fold cross-validation . . . . . 185
5.4 Which basis works best? . . . . . 188
5.4.1 Understanding of the phenomenon underlying the data . . . . . 189
5.4.2 Practical considerations . . . . . 189
5.4.3 When the choice of basis is arbitrary . . . . . 189
5.5 Summary . . . . . 191
5.6 Chapter exercises . . . . . 191
5.7 Notes on continuous function approximation* . . . . . 199

**6 Automatic feature design for classification[6] 201**

6.1 Automatic feature design for the ideal classification scenario . . . . . 201
6.1.1 Approximation of piecewise continuous functions . . . . . 202
6.1.2 The formal definition of an indicator function . . . . . 204
6.1.3 Indicator function approximation . . . . . 204
6.1.4 Recovering weights . . . . . 206
6.2 Automatic feature design for the real classification scenario . . . . . 207
6.2.1 Approximation of discretized indicator functions . . . . . 207

[6] This document is part of a book currently under development titled “Machine Learning Refined” (Cambridge University Press, late 2016) by Jeremy Watt, Reza Borhani, and Aggelos Katsaggelos. Please do not distribute. Feedback regarding any errors, comments on substance and style, recommendations, etc. is greatly appreciated! Contact: jermwatt@gmail.edu

6.2.2 The real classification scenario . . . . . . . . . . . . . . . . . . . . . 208
6.2.3 Classifier accuracy and boundary definition . . . . . . . . . . . . . . 213
6.3 Multiclass classification . . . . . . . . . . . . . . . . . . . . . . . . . . . 215
6.3.1 One versus All multiclass classification . . . . . . . . . . . . . . . . 215
6.3.2 Multiclass softmax classification . . . . . . . . . . . . . . . . . . . . 216
6.4 Cross-validation for classification . . . . . . . . . . . . . . . . . . . . . . 217
6.4.1 Hold out cross-validation . . . . . . . . . . . . . . . . . . . . . . . . 218
6.4.2 Hold out calculations . . . . . . . . . . . . . . . . . . . . . . . . . . 219
6.4.3 k-fold cross validation . . . . . . . . . . . . . . . . . . . . . . . . . . 221
6.4.4 k-fold cross validation for One versus All multiclass classification . . 224
6.5 Which basis works best? . . . . . . . . . . . . . . . . . . . . . . . . . . . 224
6.6 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 226
6.7 Chapter exercises . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 227

**7 Kernels, backpropogation, and regularized cross-validation**[7] **234**
7.1 Fixed feature kernels . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 234
7.1.1 The fundamental theorem of linear algebra . . . . . . . . . . . . . . 235
7.1.2 Kernelizing cost functions . . . . . . . . . . . . . . . . . . . . . . . . 236
7.1.3 The value of kernelization . . . . . . . . . . . . . . . . . . . . . . . . 237
7.1.4 Examples of kernels . . . . . . . . . . . . . . . . . . . . . . . . . . . 239
7.1.5 Kernels as similarity matrices . . . . . . . . . . . . . . . . . . . . . . 241
7.2 The backpropagation algorithm . . . . . . . . . . . . . . . . . . . . . . . . 242
7.2.1 Computing the gradient of a two-layer network cost function . . . . . 243
7.2.2 Three layer neural network gradient calculations . . . . . . . . . . . . 245
7.2.3 Gradient descent with momentum . . . . . . . . . . . . . . . . . . . 246
7.3 Cross-validation via $\ell_2$ regularization . . . . . . . . . . . . . . . . . . . . 248
7.3.1 $\ell_2$ regularization and cross-validation . . . . . . . . . . . . . . . . . . 250
7.3.2 Regularized k-fold cross validation for regression . . . . . . . . . . . . 250
7.3.3 Regularized cross-validation for classification . . . . . . . . . . . . . 252
7.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 252

[7] This document is part of a book currently under development titled "Machine Learning Refined" (Cambridge University Press, late 2016) by Jeremy Watt, Reza Borhani, and Aggelos Katsaggelos. Please do not distribute. Feedback regarding any errors, comments on substance and style, recommendations, etc. is greatly appreciated! Contact: jermwatt@gmail.edu

7.5 Further kernel calculations* . . . . . . 253
7.5.1 Kernelizing various cost functions . . . . . . 253
7.5.2 Fourier kernel calculations - scalar input . . . . . . 255
7.5.3 Fourier kernel calculations - vector input . . . . . . 256

## III Tools for large scale data 257

**8 Advanced gradient schemes[8] 260**
8.1 Fixed step length rules for gradient descent . . . . . . 260
8.1.1 Gradient descent and simple quadratic surrogates . . . . . . 261
8.1.2 Functions with bounded curvature and optimally conservative step length rules . . . . . . 263
8.1.3 How to use the conservative fixed step length rule . . . . . . 266
8.2 Adaptive step length rules for gradient descent . . . . . . 268
8.2.1 Adaptive step length rule via backtracking line search . . . . . . 268
8.2.2 How to use the adaptive step length rule . . . . . . 269
8.3 Stochastic gradient descent . . . . . . 271
8.3.1 Decomposing the gradient . . . . . . 272
8.3.2 The stochastic gradient descent iteration . . . . . . 273
8.3.3 The value of stochastic gradient descent . . . . . . 274
8.3.4 Step length rules for stochastic gradient descent . . . . . . 275
8.3.5 How to use the stochastic gradient method in practice . . . . . . 277
8.4 Convergence proofs for gradient descent schemes* . . . . . . 277
8.4.1 Convergence of gradient descent with Lipschitz constant fixed step length . . . . . . 278
8.4.2 Convergence of gradient descent with backtracking line search . . . . . . 279
8.4.3 Convergence of the stochastic gradient method . . . . . . 281
8.4.4 Convergence rate of gradient descent for convex functions with fixed step length . . . . . . 283
8.5 Calculation of computable Lipschitz constants* . . . . . . 284

[8] This document is part of a book currently under development titled “Machine Learning Refined” (Cambridge University Press, late 2016) by Jeremy Watt, Reza Borhani, and Aggelos Katsaggelos. Please do not distribute. Feedback regarding any errors, comments on substance and style, recommendations, etc. is greatly appreciated! Contact: jermwatt@gmail.edu

8.6 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 286
8.7 Chapter exercises . . . . . . . . . . . . . . . . . . . . . . . . . 287

**9 Dimension reduction techniques [9] 290**
9.1 Techniques for data dimension reduction . . . . . . . . . . . . . 290
9.1.1 Random subsampling . . . . . . . . . . . . . . . . . . . . 291
9.1.2 K-means clustering . . . . . . . . . . . . . . . . . . . . . 292
9.1.3 Optimization of the K-means problem* . . . . . . . . . . . 296
9.2 Principal Component Analysis . . . . . . . . . . . . . . . . . . 297
9.2.1 Optimization of the PCA problem . . . . . . . . . . . . . 302
9.3 Recommender Systems . . . . . . . . . . . . . . . . . . . . . . . 303
9.3.1 Matrix completion setup . . . . . . . . . . . . . . . . . . . 305
9.3.2 Optimization of the matrix completion model . . . . . . . . 305
9.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 306
9.5 Chapter exercises . . . . . . . . . . . . . . . . . . . . . . . . . 307

**IV Appendices 310**
9.6 Vector operations . . . . . . . . . . . . . . . . . . . . . . . . . 311
9.7 Matrix operations . . . . . . . . . . . . . . . . . . . . . . . . . 313
9.8 Basic definitions . . . . . . . . . . . . . . . . . . . . . . . . . . 316
9.9 Commonly used rules for computing derivatives . . . . . . . . . 317
9.10 Examples of gradient and Hessian calculations . . . . . . . . . . 318
9.11 Fundamental matrix factorizations . . . . . . . . . . . . . . . . 324
9.11.1 The Singular Value Decomposition . . . . . . . . . . . . . 324
9.11.2 Eigenvalue decomposition . . . . . . . . . . . . . . . . . 326
9.11.3 The pseudo-inverse . . . . . . . . . . . . . . . . . . . . . 327
9.12 Formal proofs of fundamental matrix factorizations . . . . . . . 328
9.12.1 Proof of the eigenvalue decomposition . . . . . . . . . . . 328
9.12.2 Proof of the Singular Value Decomposition . . . . . . . . 329
9.12.3 Rayleigh quotient definition and eigenvalues of a real symmetric matrix 330

[9] This document is part of a book currently under development titled “Machine Learning Refined” (Cambridge University Press, late 2016) by Jeremy Watt, Reza Borhani, and Aggelos Katsaggelos. Please do not distribute. Feedback regarding any errors, comments on substance and style, recommendations, etc. is greatly appreciated! Contact: jermwatt@gmail.edu

9.12.4 Proof that the pseudo-inverse is the smallest solution . . . . . . . . . . 332
9.13 Definitions of convexity . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 335
9.13.1 $0^{\text{th}}$ order definition of a convex function . . . . . . . . . . . . . . . . . 335
9.13.2 $1^{\text{st}}$ order definition of a convex function . . . . . . . . . . . . . . . . . 336