

Contents

| | |
|---|-----------|
| Preface | xiii |
| Acknowledgments | xix |
| 1 Introduction | 1 |
| 1.1 Problems with rule based approaches | 2 |
| 1.2 The structure and design of neural networks | 4 |
| 1.2.1 A simple neural network for template matching | 4 |
| 1.2.2 Definition of neural networks | 11 |
| 1.2.3 Design philosophy | 17 |
| 1.3 Properties that networks should achieve | 20 |
| 1.4 Overview of the book | 35 |
| 2 Highlights of Adaptive Resonance Theory | 39 |
| 2.1 Spatial mechanisms | 40 |
| 2.1.1 Short term memory equations | 40 |
| 2.1.2 Unbiased pattern storage in STM | 43 |
| 2.1.3 Outstar learning | 53 |
| 2.1.4 Instar learning | 56 |
| 2.1.5 Creating a field of instars | 59 |
| 2.1.6 Stabilizing instar learning by using feedback | 64 |
| 2.1.7 Generalizations | 68 |
| 2.1.8 Proof of convergence | 69 |
| 2.2 Using spatial mechanisms to code temporal patterns | 70 |
| 2.2.1 Storing sequences with decreasing activation | 70 |
| 2.2.2 The LTM invariance principle | 73 |
| 2.2.3 Using rehearsal to delete classified items | 77 |
| 2.2.4 Guidelines for performing context sensitive recognition | 79 |
| 2.2.5 Masking fields | 83 |
| 2.2.6 Computer simulations of the masking field | 85 |
| 2.3 Extensions to masking fields | 93 |

| | | |
|----------|---|-----|
| 3 | Classifying Spatial Patterns | 99 |
| 3.1 | Practical reasons for achieving several properties | 103 |
| 3.1.1 | The formation of stable category codes | 104 |
| 3.1.2 | Real-time operation | 107 |
| 3.1.3 | Learning and classifying embedded patterns | 108 |
| 3.2 | Normalizing the output of $F^{(1)}$ | 111 |
| 3.2.1 | Normalizing by length | 111 |
| 3.2.2 | Normalizing at each $F^{(2)}$ cell | 113 |
| 3.3 | The structure of the $F^{(2)}$ cell assembly | 115 |
| 3.4 | STM activation equations after learning | 118 |
| 3.5 | Overview of learning | 124 |
| 3.5.1 | Learning isolated and embedded patterns | 124 |
| 3.5.2 | Forming stable category codes | 126 |
| 3.6 | The excitatory LTM learning equation | 130 |
| 3.7 | Analysis of the excitatory LTM learning equation | 132 |
| 3.7.1 | Simplifications made in the analysis | 133 |
| 3.7.2 | Competition between weights at the same cell | 135 |
| 3.7.3 | Competition between weights at different cells | 138 |
| 3.7.4 | Analyzing different choices for m and n | 139 |
| 3.8 | Determining which inputs are in an $F^{(2)}$ cell's classified pattern | 142 |
| 3.9 | General approach to learning inhibitory connections | 145 |
| 3.10 | Multiplexing two values on each output signal | 147 |
| 3.11 | Regulating inhibitory learning to allow the network to form stable category codes | 150 |
| 3.12 | Freezing inhibitory weights to prevent oscillation | 153 |
| 3.13 | Learning cell sizes | 154 |
| 3.14 | Modifications to improve learning | 155 |
| 3.14.1 | Restricting weight decay | 155 |
| 3.14.2 | Limiting inhibition to uncommitted cells | 157 |
| 3.14.3 | Resetting uncommitted cells with low activity | 159 |
| 3.14.4 | Using r_{ji} to modulate the learning rate | 162 |

| | | |
|----------|--|-----|
| 3.15 | Simulations | 166 |
| 3.16 | Properties not achieved by SONNET 1 | 177 |
| 3.16.1 | Combining inhibitory signals multiplicatively | 180 |
| 3.16.2 | Using a separate vigilance for each link | 181 |
| 3.16.3 | Improving the calculation for I_i^x | 183 |
| 4 | Classifying Temporal Patterns | 187 |
| 4.1 | Constraints on the LTM invariance principle | 191 |
| 4.2 | Implementing the LTM invariance principle with an on-center off-surround circuit | 194 |
| 4.3 | Modifying the feedback weights | 198 |
| 4.4 | Resetting classified portions of the input pattern | 201 |
| 4.5 | Resetting $F^{(1)}$ cells when the field saturates | 204 |
| 4.6 | Simulations | 209 |
| 5 | Multilayer Networks and the Use of Attention | 217 |
| 5.1 | Combining previous circuits to create homologous fields | 219 |
| 5.2 | Using feedback to prevent categories from forming at inappropriate times | 222 |
| 5.3 | Using attention to allow the network to be event driven | 223 |
| 5.4 | Classifying items presented at different rhythms | 227 |
| 5.5 | Representing different dimensions of input patterns in different fields | 232 |
| 5.6 | Eliminating the lockstep operation of the network | 235 |
| 5.7 | Adding an attentional reset mechanism to SONNET | 237 |
| 5.8 | Modifying the network for recurrent operation | 238 |
| 6 | Representing Synonyms | 241 |
| 6.1 | Using multiple representations for each item | 242 |
| 6.2 | Forcing multiple nodes to learn each pattern | 245 |
| 6.3 | Current networks do not represent synonyms well | 248 |

| | | |
|----------|---|-----|
| 6.4 | The use of presynaptic inhibition allows synonyms to be properly handled | 252 |
| 6.5 | Summary of the first segment in the chapter and a preview of the second segment | 255 |
| 6.6 | Learning presynaptic inhibition | 256 |
| 6.7 | Single trial learning of synonymous representations | 257 |
| 6.8 | Manner by which synonyms become associated | 265 |
| 6.9 | Multiple links from each $F^{(1)}$ node to each $F^{(2)}$ node | 267 |
| 6.10 | Summary of the second segment in the chapter and a preview of the third segment | 272 |
| 6.11 | Presynaptic inhibition for intercell competition | 274 |
| 6.12 | Inhibitory weights in the presynaptic connections | 279 |
| 6.13 | Presynaptic inhibition in the feedback links | 280 |
| 6.14 | Summary of the third segment in the chapter and a preview of the chapter's final sections | 284 |
| 6.15 | Distortion insensitive recognition | 284 |
| 6.16 | Classification of spatial patterns | 287 |
| 6.17 | Creating distributed representations | 289 |
| 6.17.1 | Hardware needed for local representations | 291 |
| 6.17.2 | Reducing hardware requirements | 292 |
| 6.17.3 | STM binding of distributed classifications | 295 |
| 6.17.4 | LTM binding of distributed classifications | 300 |
| 6.17.5 | Learning distributed representations | 304 |
| 6.18 | Reducing the number of connections required | 306 |
| 6.18.1 | Reducing wasted lateral connections | 307 |
| 6.18.2 | Transmitting information via shared links | 311 |
| 7 | Specific Architectures That Use Presynaptic Inhibition | 317 |
| 7.1 | First step in implementing SONNET 2 | 317 |
| 7.1.1 | Minimal SONNET 2 architecture | 318 |
| 7.1.2 | The operation of the network after learning | 322 |
| 7.1.3 | New formulation for the function I_i^x | 326 |
| 7.1.4 | Using the network to achieve learning | 329 |

| | | |
|----------|---|-----|
| 7.2 | Translation and size invariant recognition | 334 |
| 7.2.1 | Architecture to center objects | 336 |
| 7.2.2 | Extensions for multiple input dimensions and multiple feature types | 343 |
| 7.2.3 | Achieving size invariant recognition | 348 |
| 7.2.4 | Extensions to allow the centering of objects surrounded by extraneous information | 353 |
| 7.2.5 | Extensions to allow the simultaneous classification of multiple objects | 355 |
| 8 | Conclusion | 361 |
| A | Feedforward Circuits for Normalization and Noise Suppression | 375 |
| B | Network Equations Used in the Simulations of Chapter 3 | 381 |
| C | Network Equations Used in the Simulations of Chapter 4 | 387 |
| D | Glossary | 395 |
| | Bibliography | 399 |
| | Index | 407 |