How to train the Neural Web

Witold Warchoł

1 Input Data Files Required

1.1 Text Processing Data

- `vocabulary.txt` - Word vocabulary for text processing and embedding generation. Each line contains one word or token.
- `custom_embeddings.txt` - Pre-trained word embeddings. Format: word followed by space-separated float values representing the embedding vector.
- Text input string - Training text data provided as string constant or loaded from file. Example: "Apple, banana, cherry, date, and elderberry are fruits."

1.2 System State Files

- `memory_system.dat` - Binary file containing previous memory system state (optional for fresh training)
- `hierarchical_memory.dat` - Hierarchical memory structure data (optional)
- `system_parameters.dat` - Optimization parameters and performance metrics from previous runs
- `metacontroller.dat` - Meta-controller state for adaptive learning
- `motivation.dat` - Intrinsic motivation system parameters
- `performance_metrics.dat` - Network performance tracking data
- `reflection_params.dat` - Self-reflection system parameters
- `identity_system.dat` - Self-identity framework state
- `knowledge_filter.dat` - Knowledge categorization system
- `metacognition.dat` - Metacognitive processing parameters
- `meta_learning.dat` - Meta-learning state and strategies

2 User-Defined Training Parameters

2.1 Network Training Configuration

- Initial learning rate (typically 0.01-0.1)
- Number of training steps/epochs
- Batch size for processing optimization
- Target output vectors for supervised learning components
- Performance thresholds for adaptation triggers

2.2 Memory System Configuration

- Memory decay rates for temporal forgetting
- Importance thresholds for memory consolidation
- Similarity thresholds for memory clustering
- Capacity limits for each memory tier (short/medium/long-term)

2.3 Ethical Framework Parameters

- Ethical principle definitions and weights
- Decision evaluation criteria
- Moral compass confidence thresholds
- Ethical constraint enforcement levels

3 Training Data Generation

3.1 Task Prompts

The system requires task-specific prompts and verification criteria:

- Task descriptions for each training step
- Verification instructions with expected outcomes
- Confidence scoring mechanisms
- Reasoning validation frameworks

3.2 Target Generation

- Define target output generation strategy
- Specify error calculation methods
- Set performance evaluation metrics
- Configure adaptive target adjustment mechanisms

4 Social and Emotional Training Data

4.1 Social Interaction Scenarios

- Behavioral patterns for person modeling
- Negotiation contexts and outcome preferences
- Empathy development scenarios
- Social feedback examples

4.2 Emotional State Definitions

- Emotional trigger patterns
- Regulation strategies and thresholds
- Cognitive-emotional integration parameters
- Emotional memory importance weighting

5 Imagination and Creativity Parameters

5.1 Scenario Generation

- Divergence factor ranges for creative exploration
- Plausibility evaluation criteria
- Scenario blending coefficients
- Creativity-coherence balance parameters

5.2 Problem-Solving Configuration

- Error thresholds triggering imagination activation
- Solution influence weights
- Exploration vs exploitation balance
- Creative solution evaluation metrics

6 Training Pipeline Configuration

6.1 Initialization Strategy

1. Load existing system states or initialize with default values
2. Configure dynamic parameter adaptation rates
3. Set up performance tracking and optimization schedules
4. Initialize ethical and social processing frameworks

6.2 Training Loop Requirements

1. Define input tensor generation from text and memory
2. Configure predictive coding and error computation
3. Set adaptation trigger frequencies (memory consolidation, parameter optimization, system reflection)
4. Establish performance evaluation intervals

6.3 Web Search Integration

- Search query generation strategies
- Result filtering and relevance scoring
- Knowledge integration confidence thresholds
- External information validation criteria

7 Performance Optimization Requirements

7.1 Dynamic Parameter Tuning

- Learning rate adaptation schedules
- Batch size optimization criteria
- Network plasticity adjustment parameters
- Noise tolerance configuration

7.2 System Health Monitoring

- Error rate thresholds for intervention
- Memory usage optimization targets
- Identity consistency verification schedules
- Security validation parameters

8 Output and Evaluation Configuration

8.1 Performance Metrics Definition

- Loss function selection and weighting
- Convergence criteria and stopping conditions
- Progress visualization and reporting intervals
- System introspection and self-questioning protocols

8.2 State Persistence

- Save intervals for system components
- Backup and recovery strategies
- State validation and consistency checking
- Performance history tracking and analysis

9 Training Methodology

9.1 Pre-Training Setup

1. Prepare all required input files in correct formats
2. Initialize system parameters with conservative values
3. Set up performance monitoring and logging systems
4. Configure backup and state persistence mechanisms

9.2 Training Phases

9.2.1 Phase 1: Basic Network Stabilization (Steps 1-100)

- Focus on basic neuron state convergence
- Use simple target outputs and low learning rates (0.001-0.01)
- Monitor network stability and prevent oscillations
- Establish baseline memory formation patterns
- Disable complex subsystems (imagination, social processing)

9.2.2 Phase 2: Memory System Integration (Steps 101-300)

- Gradually increase memory system engagement
- Introduce hierarchical memory consolidation
- Begin predictive coding integration
- Monitor memory-neuron interaction stability
- Adjust memory decay and importance parameters

9.2.3 Phase 3: Multi-System Activation (Steps 301-600)

- Enable emotional processing subsystem
- Activate imagination system with low creativity factors
- Introduce basic social modeling components
- Begin ethical framework evaluation
- Monitor inter-system interference patterns

9.2.4 Phase 4: Advanced Integration (Steps 601-1000)

- Full system activation with dynamic adaptation
- Enable web search integration and external knowledge
- Activate creative problem-solving mechanisms
- Implement complete ethical decision-making
- Focus on emergent behavior optimization

9.3 Training Loop Implementation

Each training step follows this sequence:

1. **Input Generation**: Create input tensor from text, memory, and context
2. **Forward Processing**: Update neuron states through network
3. **Memory Operations**: Retrieve relevant memories and update working memory
4. **Subsystem Processing**: Apply emotional, social, ethical, and imaginative processing
5. **Error Computation**: Calculate prediction errors and loss functions
6. **Backpropagation**: Update weights and connection strengths
7. **Parameter Adaptation**: Adjust learning rates and system parameters
8. **State Consolidation**: Update memory systems and save critical states
9. **Performance Evaluation**: Assess progress and trigger optimizations

9.4 Training Monitoring

9.4.1 Critical Metrics to Track

- Loss convergence and stability
- Memory system utilization and effectiveness
- Inter-system communication quality
- Ethical alignment maintenance
- Identity consistency over time
- Creative output quality and relevance

9.4.2 Warning Signs Requiring Intervention

- Oscillating or diverging loss functions
- Memory system overflow or corruption
- Identity consistency failures
- Ethical constraint violations
- Excessive system resource consumption
- Degraded performance after optimization attempts

9.5 Adaptive Training Strategies

9.5.1 Dynamic Parameter Adjustment

- Reduce learning rate when loss plateaus
- Increase exploration when performance stagnates
- Adjust memory consolidation frequency based on utilization
- Modify creativity factors based on problem-solving success
- Scale ethical constraint enforcement with system maturity

9.5.2 Curriculum Learning Approach

- Start with simple, well-defined tasks
- Gradually increase task complexity and ambiguity
- Introduce social scenarios after basic competence
- Add ethical dilemmas only after stable decision-making
- Incorporate creative challenges as final training phase

9.6 Troubleshooting Common Issues

9.6.1 Training Instabilities

- **Oscillating outputs**: Reduce learning rate, increase damping factors
- **Memory overflow**: Adjust consolidation frequency, increase decay rates
- **System conflicts**: Temporarily disable conflicting subsystems, retrain integration
- **Identity drift**: Strengthen identity verification, restore from backup

9.6.2 Performance Degradation

- **Loss plateaus**: Increase learning rate temporarily, add exploration noise
- **Overfitting**: Introduce regularization, reduce model complexity
- **Memory saturation**: Clear low-importance memories, optimize storage
- **Ethical violations**: Strengthen constraint enforcement, retrain ethical framework

9.7 Training Completion Criteria

Training is considered successful when:

1. Loss function converges to acceptable levels
2. Memory system operates efficiently without overflow
3. All subsystems interact harmoniously
4. Ethical decision-making remains consistent
5. Identity system maintains coherence over time
6. Creative outputs demonstrate both novelty and relevance
7. Performance metrics stabilize across evaluation periods

9.8 Post-Training Validation

9.8.1 System Testing Protocol

1. Present novel scenarios requiring multi-system integration

2. Evaluate ethical decision consistency under pressure

3. Test memory recall and consolidation effectiveness

4. Assess creative problem-solving capabilities

5. Verify identity stability across diverse contexts

6. Confirm social interaction appropriateness

Success requires careful tuning of interaction parameters between subsystems and continuous monitoring of emergent behaviors to ensure stable and beneficial learning outcomes.