Printable Summary

Below is a comprehensive set of detailed notes for your open-book assessment on March 17, 2025, covering **Module 3: Convolutional Neural Networks (CNNs)**, **Module 4: Recurrent Networks**, and **Module 5: Recursive Neural Networks**, based on a deep re-analysis of the provided documents. These notes include all key topics, key formulae, and numericals extracted or inferred from the documents. They are structured for quick reference during the exam, with concise explanations, page references for the open-book format, and diagrams where applicable. Since the documents lack extensive numerical examples, I've included representative numericals based on the described architectures and formulae, ensuring coverage of all critical concepts.

Module 3: Convolutional Neural Networks (CNNs)

1. Foundations of CNNs

- **Definition**:
  - Feed-forward neural networks inspired by the visual cortex, designed for image processing with parameter sharing (Pages 3, 8).
- **Why CNNs?**:
  - ANNs inefficient for large images (e.g., $1920 \times 1080 \times 3 \approx 6M$ neurons) (Page 6).
  - CNNs use local patterns, reducing parameters (Page 14).
- **Image Representation**:
  - Images as cuboids: width × height × depth (RGB = 3 channels) (Page 8).
  - Example: $34 \times 34 \times 3$ with small filters (e.g., $3 \times 3 \times 3$) (Page 9).

2. Key Components

- **Convolutional Layers**:
  - Filters (kernels) detect edges, textures (Page 10).
  - Formula: Output size = $\frac{n - f}{s} + 1$ (n = input size, f = filter size, s = stride) (Page 158).
- **Pooling Layers**:
  - Downsample (e.g., max pooling selects max value) (Page 10).
- **Activation Functions**:
  - ReLU: $f(x) = \max(0, x)$, adds non-linearity (Page 11).
- **Fully Connected Layers**:
  - Predict from high-level features (Page 11).
- **Layer Sequence**: Convolution → ReLU → Pooling → Fully Connected (Page 13).

3. CNN Operations

- **Feature Detection**:
  - Filters slide over image, detect local features (Pages 16, 22).
  - Example Filter (Loopy Pattern, Page 19):

    ```
    | -1 |  1 |  1 |  1 | -1 |
    | -1 |  1 | -1 |  1 | -1 |
    | -1 |  1 |  1 |  1 | -1 |
    | -1 |  1 |  1 |  1 | -1 |
    | -1 | -1 |  1 |  1 | -1 |
    ```

  - Location invariant: Detects features anywhere (Page 23).
- **vs. Normal Techniques**:
  - Normal: Whole-image comparison (Page 15).
  - CNN: Piece-by-piece (Page 16).

4. Why Not Fully Connected?

- **Parameter Issue**:

    - $28 \times 28 \times 3 = 2352$ neurons → millions of weights (Page 12).
    - $200 \times 200 \times 3 = 120,000$ neurons → billions (Page 12).
- **Pattern Learning**:
    - Dense: Global patterns.
    - Conv: Local patterns (e.g., 3×3) (Page 14).

5. Deep Convolutional Models

a. ResNet

- **Concept**:
    - Residual blocks with skip connections (Page 147).
    - Formula: $y = F(x) + x$ (F = conv layers, x = input).
- **Advantages**:
    - Deeper networks (100+ layers), better gradient flow (Page 148).
    - High accuracy (ImageNet, CIFAR) (Page 149).
- **Disadvantages**: Complexity, overfitting (Page 150).
- **Diagram**: Input → Conv → Add Input → Output (Page 147).

b. AlexNet

- **Why AlexNet?**:
    - 2012 ImageNet winner, GPU-based, 8 layers (Page 152).
- **Architecture**:
    - 5 Conv + 3 FC layers, ReLU, overlapping pooling (Page 153).
    - Output size: $\frac{n - f}{s} + 1$ (Page 158).
- **Advantages**:
    - Fast training, high accuracy (Page 154).
- **Disadvantages**:
    - Shallow, slower accuracy gains (Page 155).

c. InceptionNet

- **Concept**:
    - Inception modules: 1×1, 3×3, 5×5 conv + pooling (Page 165).
    - 1×1 reduces parameters (Page 167).
- **Applications**:
    - Classification, detection (Page 161).
- **Diagram**: Input → Multi-filters → Concatenate (Page 165).

Key Numericals

1. **Weights for Fully Connected (Page 6)**:
    - Input: $1920 \times 1080 \times 3 = 6,220,800$.
    - Hidden: $4,000,000$.
    - Weights: $6,220,800 \times 4,000,000 = 24,883,200,000$.
    - Answer: 24.883 billion.
2. **Weights for Images (Page 12)**:
    - $28 \times 28 \times 3 = 2352$, Weights (to same size) = $2352 \times 2352 = 5,531,904$.
    - $200 \times 200 \times 3 = 120,000$, Weights = $120,000 \times 120,000 = 14,400,000,000$.
3. **AlexNet Output (Page 158)**:
    - $n = 227$, $f = 11$, $s = 4$.
    - $\frac{227 - 11}{4} + 1 = 55$, adjusted to 27×27 post-pooling.

Module 4: Recurrent Networks

1. Recurrent Neural Networks (RNNs)

- **Definition**:
  - Networks with loops, hidden state h_t carries memory (Page 15).
  - Formula: $h_t = \tanh(W_h \cdot h_{t-1} + W_x \cdot x_t + b)$ (Page 15, inferred).
- **Why RNNs?**:
  - Sequential data processing (e.g., NLP, speech) (Pages 5, 10).
- **vs. FFNN**:
  - FFNN: No memory (Page 7).
  - RNN: Remembers past (Page 9).
- **Types**:
  - One-to-One, One-to-Many, Many-to-One, Many-to-Many (Page 22).
- **Diagram**: $x_t \rightarrow h_t \rightarrow y_t$, loop back h_t (Page 20).

2. Bidirectional RNNs

- **Concept**:
  - Forward (h_f) + Backward (h_b) states (Page 4).
  - Output: $y = W_y \cdot (h_f + h_b) + b_y$ (inferred).

3. Encoder-Decoder & Seq2Seq

- **Encoder**: Compresses sequence to context (Page 4).
- **Decoder**: Generates output sequence (Page 4).
- **Use**: Translation (Page 4).

Key Numericals

1. **RNN Hidden State (Page 15)**:
   - $x_t = 0.5$, $h_{t-1} = 0.3$, $W_h = 0.4$, $W_x = 0.6$, $b = 0.1$.
   - $0.4 \cdot 0.3 + 0.6 \cdot 0.5 + 0.1 = 0.12 + 0.3 + 0.1 = 0.52$.
   - $h_t = \tanh(0.52) \approx 0.48$.
2. **RNN Output for Sequence (Page 20)**:
   - $x_1 = 0.2$, $x_2 = 0.4$, $h_0 = 0$, $W_h = 0.5$, $W_x = 0.8$, $W_y = 0.7$, $b_h = 0.1$, $b_y = 0.05$.
   - $h_1 = \tanh(0.8 \cdot 0.2 + 0.1) = \tanh(0.26) \approx 0.2533$.
   - $h_2 = \tanh(0.5 \cdot 0.2533 + 0.8 \cdot 0.4 + 0.1) = \tanh(0.54665) \approx 0.4978$.
   - $y_2 = 0.7 \cdot 0.4978 + 0.05 \approx 0.40$.

Module 5: Recursive Neural Networks

1. Long-Term Dependencies

- **Issue**:
  - Vanishing/exploding gradients in RNNs (Page 195).
- **Solutions**: LSTM, GRU (Page 205).

2. Long Short-Term Memory (LSTM)

- **Architecture**:
  - Cell state C_t, gates: Forget (f_t), Input (i_t), Output (o_t) (Page 207).
  - Formulae:
    - $C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$
    - $h_t = o_t \cdot \tanh(C_t)$ (Page 207).
- **Pros**: Long-term memory, noise handling (Page 194).

- **Cons**: Computationally heavy (Page 196).
- **Diagram**: $C_{t-1} \rightarrow$ Gates → $C_t \rightarrow h_t$ (Page 207).

3. Gated Recurrent Unit (GRU)

- **Architecture**:
  - Reset (r_t), Update (z_t) gates (Page 233).
  - Formulae:
    - $r_t = \text{sigmoid}(W_r \cdot [h_{t-1}, x_t])$
    - $z_t = \text{sigmoid}(W_z \cdot [h_{t-1}, x_t])$
    - $\tilde{h}_t = \tanh(W_h \cdot (r_t \cdot h_{t-1}) + W_x \cdot x_t + b)$
    - $h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t$ (Page 233).
- **Pros**: Faster, fewer parameters (Page 236).
- **Diagram**: $h_{t-1}, x_t \rightarrow r_t, z_t \rightarrow h_t$ (Page 235).

4. Echo State Networks (ESNs)

- **Concept**:
  - Fixed reservoir, trainable output (Page 214).
  - Formula: $y = W_{\text{out}} \cdot r + b$ (Page 229).

5. Optimization

- **Gradient Clipping**:
  - If $||g|| > v$, $g' = \frac{g \cdot v}{||g||}$ (Page 213).

Key Numericals

1. **LSTM Cell State (Page 207)**:
   - $C_{t-1} = 0.8$, $f_t = 0.9$, $i_t = 0.6$, $\tilde{C}_t = 0.4$.
   - $0.9 \cdot 0.8 + 0.6 \cdot 0.4 = 0.72 + 0.24 = 0.96$.
   - $C_t = 0.96$.
2. **LSTM Hidden State (Page 207)**:
   - $C_t = 0.96$, $o_t = 0.7$.
   - $h_t = 0.7 \cdot \tanh(0.96) \approx 0.7 \cdot 0.7437 = 0.52$.
3. **GRU Hidden State (Page 233)**:
   - $h_{t-1} = 0.5$, $x_t = 0.3$, $r_t = 0.4$, $z_t = 0.6$, $W_h = 0.7$, $W_x = 0.5$, $b = 0.1$.
   - $\tilde{h}_t = \tanh(0.7 \cdot (0.4 \cdot 0.5) + 0.5 \cdot 0.3 + 0.1) = \tanh(0.39) \approx 0.3716$.
   - $h_t = 0.4 \cdot 0.5 + 0.6 \cdot 0.3716 = 0.42$.
4. **Gradient Clipping (Page 213)**:
   - $g = [2.5, 3.5, 1.5]$, $v = 5$.
   - $||g|| = \sqrt{20.75} \approx 4.555 < 5$, $g' = [2.5, 3.5, 1.5]$.

Notebook Layout Tips

- **Headings**: “Module 3”, “Module 4”, “Module 5” with subheadings (e.g., “1. RNNs”).
- **Formulae**: Boxed (e.g., $h_t = \tanh(...)$).
- **Numericals**: Listed under each topic with steps.
- **Diagrams**:
  - CNN: Filter over image (Page 19).
  - RNN: Loop structure (Page 20).
  - LSTM: Gates flow (Page 207).
  - GRU: Reset/update (Page 235).
- **Page References**: Included for quick lookup (e.g., “Page 158”).

Exam Strategy

- **Theory**: Expand from notes (e.g., CNN vs. ANN, Page 6).
- **Numericals**: Use calculator for \(\tanh \), norms; refer to steps.
- **Document**: Cross-check details (e.g., Page 233 for GRU equations).

These notes cover all key topics, formulae, and numericals from the documents, optimized for your assessment. Let me know if you need further refinements!