Gated Recurrent Units

Course: Machine Learning / Deep Learning
Submission Deadline: 16/05/2025

Part A: Theoretical Questions

1. **Conceptual Understanding**
    - (a) Explain the key limitation of a traditional Feedforward Neural Network (FNN) in handling sequential data. How does an RNN address this?
    - (b) Compare and contrast the roles of the **update gate** and **reset gate** in a Gated Recurrent Unit (GRU).
2. **Mathematical Formulation**
    - Derive the equations for the hidden state (h_t) in a **vanilla RNN** and the candidate activation ($\tilde{h}_t$) in a **GRU**. Highlight how gates influence the flow of information.
3. **Short Answer (3 Marks)**
    - Why do LSTM networks include a "forget gate" while GRUs do not? Explain with an example where forgetting past information might improve model performance.

Part B: Practical Implementation

1. **Code Implementation**
    - Implement a **GRU-based model** using PyTorch/TensorFlow to predict the next word in a sentence (language modeling). Use the following steps:
        - Load a toy text dataset (e.g., Shakespeare excerpts).
        - Preprocess the text (tokenization, sequencing).
        - Define a GRU layer with **hidden_size=128**.
        - Train the model and report training/validation loss curves.

Submission:

- Jupyter Notebook with commented code.
- Brief explanation of how the GRU's gates affect predictions.

Experimental Analysis

- (a) Replace the GRU layer with a vanilla RNN and compare the training dynamics (e.g., loss convergence, gradient behavior).
- (b) Visualize the hidden state activations (using PCA/t-SNE) for both models and discuss which captures longer dependencies better.

Submission Guidelines:

- **Theory:** Submit a PDF with typed answers (handwritten derivations allowed for equations).
- **Code:** Share a GitHub repo link or .ipynb file with outputs.
- **Evaluation Criteria:**
  - Clarity of explanations (theory).
  - Correct implementation and analysis (practical).
  - Justification of observations.