Coding Blocks | Online

Machine Learning Online

Assignment – 2: Encoder-Decoder Models using RNN and LSTM

Part-I: Theoretical Understanding of RNN, LSTM, and Encoder-Decoder

Objective:

To understand the architecture and working of Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and the encoder-decoder structure for sequence-to-sequence tasks.

Assignment Tasks

Task 1: Conceptual Questions

Answer the following questions in 2–4 sentences each:

1. What is the difference between RNN and LSTM?
2. What is the vanishing gradient problem, and how does LSTM solve it?
3. Explain the purpose of the Encoder-Decoder architecture.
4. In a sequence-to-sequence model, what are the roles of the encoder and decoder?
5. How is attention different from a basic encoder-decoder model?

Task 2: Sequence-to-Sequence Data Flow

Draw or describe the **data flow** in an encoder-decoder model using RNN/LSTM.
Clearly label:

- Input sequence
- Hidden states
- Context vector
- Output sequence

Part-II: Implementation of Encoder-Decoder using RNN / LSTM

Objective:

To implement a simple encoder-decoder model for sequence tasks like translation or summarization using TensorFlow/Keras or PyTorch.

Assignment Tasks

Task 3: Data Preparation for Sequence Learning

- Use a toy dataset or download an English-to-French translation dataset (e.g., from http://www.manythings.org/anki/).
- Preprocess the text:
    - Tokenize input and output sequences
    - Pad sequences for batching
    - Prepare `input_tensor`, `target_tensor`

Task 4: Build Encoder and Decoder using LSTM (Keras)

- Define an **encoder model** using `Embedding` + `LSTM`
- Define a **decoder model** using `Embedding` + `LSTM` + `Dense`
- Compile and train the model for at least 10 epochs
- Print the training loss after each epoch

Task 5: Inference and Evaluation

- Implement inference (testing) using a loop:
    - Feed encoder with input sequence
    - Predict one word at a time from the decoder using teacher forcing or greedy search
- Translate 5 test sentences and print both input and output

Part-III: Visualizing and Enhancing Encoder-Decoder

Objective:

To interpret and improve encoder-decoder outputs with basic attention and performance visualization.

Assignment Tasks

Task 6: Add Basic Attention Mechanism (Optional – Bonus)

- Modify your decoder to include attention on encoder outputs
- Visualize attention weights (e.g., with heatmaps)

Task 7: Plotting Loss and Accuracy

- Plot training loss and accuracy curves using `matplotlib`
- Write observations on:
  - Overfitting
  - Underfitting
  - Training stability

Task 8: Model Performance Discussion

Answer:

1. What are the challenges in training sequence-to-sequence models?
2. What does a “bad” translation look like? Why might it happen?
3. How can the model be improved further?

Submission Guidelines

- Push Code on your Github
    - Python notebook
    - A README file with:
        - Model structure
        - Instructions to run the code
        - Sample outputs
- Clearly label each section of the assignment
- Include code comments for clarity

Coding Blocks | Online