Summary Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling

Superiority of Gated Units: Both LSTM and GRU units outperformed traditional tanh units across all tasks, highlighting the effectiveness of gating mechanisms in capturing complex sequence dependencies.

GRU vs. LSTM:

- In polyphonic music modeling, GRUs performed comparably to or slightly better than LSTMs.
- In speech signal modeling, results varied: LSTMs performed better on Ubisoft A, while GRUs outperformed on Ubisoft B.

Convergence Speed: GRUs demonstrated faster convergence in terms of both the number of updates and actual CPU time, particularly in the music modeling tasks.

Example

1. Using Traditional RNN (tanh units)

A basic RNN passes the input through a layer with a tanh activation function.

- Problem: Over time, gradients become **very small (vanishing gradient problem)**.
- Result: The model may **forget the early characters** like 'h' and struggle to learn long-term dependencies.

Output Prediction:
→ May incorrectly predict '?' as something unrelated, like 'a' or 'z'.

---

2. Using LSTM

LSTM has **gates** (input, forget, output) that **control information flow** and help the model **remember long-term patterns**.

- For example, LSTM can **remember that the first letter was 'h'**, even after seeing several steps.
- Helps model structured patterns like "hello" or "world".

Output Prediction:
→ More likely to correctly predict '?' as ' ' or even start a new word.

---

3. Using GRU

GRU is a **simplified version** of LSTM:

- Combines the input and forget gates into a single **update gate**.
- Uses fewer parameters but still captures long-term dependencies.

Output Prediction:
→ Performs **similarly to LSTM**, sometimes faster and more efficient.