Section 5: RNNs & LSTMs for Sequential Data

Advanced Time Series Forecasting for Energy Systems

Dr Bibin Wilson & Prof Anand Singh

September 10, 2025

Indian Institute of Technology Bombay

Section Overview

RNN Fundamentals

LSTM Architecture

Advanced RNN Architectures

Energy Load Forecasting Applications

Advanced Techniques and Optimization

RNN Fundamentals

Recurrent Neural Networks: Core Concepts

Recurrence Equation:

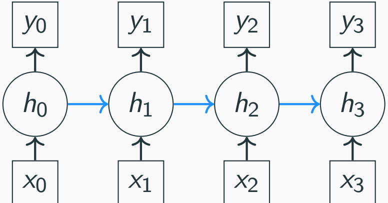
$$\mathbf{h}_t = f(\mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{xh}\mathbf{x}_t + \mathbf{b}_h)$$

$$\mathbf{y}_t = g(\mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_y)$$

Key Properties:

- **Temporal Memory**: Previous states influence curr.
- **Parameter Sharing**: Same weights across time

Unrolled Through Time:



Applications in Energy:

- Load forecasting
- Price prediction
- Demand response
- Fault detection sequences
- Generation scheduling

Challenges:

- Vanishing gradients
- Exploding gradients
- Long-term dependencies
- Computational cost

Backpropagation Through Time (BPTT)

Gradient Flow:

$$\frac{\partial L}{\partial \mathbf{W}} = \sum_{t=1}^{T} \frac{\partial L_t}{\partial \mathbf{W}}$$

Chain Rule Application:

$$\frac{\partial L}{\partial \mathbf{h}_t} = \frac{\partial L}{\partial \mathbf{h}_{t+1}} \cdot \frac{\partial \mathbf{h}_{t+1}}{\partial \mathbf{h}_t} + \frac{\partial L_t}{\partial \mathbf{h}_t}$$

Gradient Problems:

Vanishing: $\prod_i \frac{\partial \mathbf{h}_i}{\partial \mathbf{h}_{i-1}} \to 0$
Exploding: $\prod_i \frac{\partial \mathbf{h}_i}{\partial \mathbf{h}_{i-1}} \to \infty$

Solutions:

- Gradient clipping
- Better initialization
- Gated architectures (LSTM/GRU)
- Skip connections
- Truncated BPTT

Truncated BPTT:

- Limit backprop steps
- Trade-off: memory vs accuracy
- Common: 20-35 steps

LSTM Architecture

Long Short-Term Memory (LSTM)

Gate Equations:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{1}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{2}$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c) \tag{3}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \tag{4}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{5}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{6}$$

Gate Functions:

- **Forget Gate** ($\mathbf{f}_t$): What to discard
- **Input Gate** ($\mathbf{i}_t$): What to store
- **Candidate** ($\tilde{\mathbf{c}}_t$): New information
- **Output Gate** ($\mathbf{o}_t$): What to output

LSTM Deep Dive:

`05_lstm_energy_advanced.ipynb`

Gate visualizations included

GRU: Gated Recurrent Unit

Simplified Architecture:

$$\mathbf{z}_t = \sigma(\mathbf{W}_z[\mathbf{h}_{t-1}, \mathbf{x}_t]) \tag{7}$$
$$\mathbf{r}_t = \sigma(\mathbf{W}_r[\mathbf{h}_{t-1}, \mathbf{x}_t]) \tag{8}$$
$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}[\mathbf{r}_t \odot \mathbf{h}_{t-1}, \mathbf{x}_t]) \tag{9}$$
$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \tag{10}$$

Gates:

- **Update Gate** ($\mathbf{z}_t$): How much to update
- **Reset Gate** ($\mathbf{r}_t$): How much to forget

GRU vs LSTM:

- Fewer parameters (3 vs 4 gates)
- No separate cell state
- Often comparable performance
- Faster training

When to Use:

- **LSTM**: Complex patterns, long sequences
- **GRU**: Limited data, faster training needed

GRU Implementation:

`05_lstm_energy_advanced.ipynb`

Advanced RNN Architectures

Bidirectional RNNs

Architecture:

$$\overrightarrow{\mathbf{h}}_t = f(\overrightarrow{\mathbf{W}}_{hh}\overrightarrow{\mathbf{h}}_{t-1} + \mathbf{W}_{xh}\mathbf{x}_t)$$

$$\overleftarrow{\mathbf{h}}_t = f(\overleftarrow{\mathbf{W}}_{hh}\overleftarrow{\mathbf{h}}_{t+1} + \mathbf{W}_{xh}\mathbf{x}_t)$$

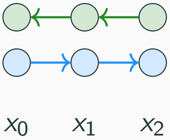
$$\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t]$$

Benefits:

- Context from both directions
- Better feature extraction
- Improved accuracy

Energy Applications:

- Anomaly detection (need full context)
- Data imputation
- Pattern recognition
- NOT for real-time forecasting



Encoder-Decoder Architecture

Sequence-to-Sequence:

- **Encoder**: Compress input sequence
- **Context Vector**: Fixed-size representation
- **Decoder**: Generate output sequence

Mathematical Formulation: Encoder:

$\mathbf{h}_t^{enc} = f_{enc}(\mathbf{x}_t, \mathbf{h}_{t-1}^{enc})$

Context: $\mathbf{c} = g(\mathbf{h}_1^{enc}, ..., \mathbf{h}_T^{enc})$

Decoder: $\mathbf{h}_t^{dec} = f_{dec}(\mathbf{y}_{t-1}, \mathbf{h}_{t-1}^{dec}, \mathbf{c})$

Energy Forecasting:

- Multi-step ahead prediction
- Different input/output lengths
- Weather $\rightarrow$ Load mapping
- Cross-domain translation

Encoder-Decoder:

`05_lstm_energy_advanced.ipynb`

Multi-horizon forecasting

Attention Mechanisms

Attention Score:

$$\alpha_{t,s} = \frac{\exp(e_{t,s})}{\sum_{s'=1}^{S} \exp(e_{t,s'})}$$

where $e_{t,s} = a(\mathbf{h}_{t-1}^{dec}, \mathbf{h}_s^{enc})$

Context Vector:

$$\mathbf{c}_t = \sum_{s=1}^{S} \alpha_{t,s} \mathbf{h}_s^{enc}$$

Types:

- **Bahdanau**: Additive attention
- **Luong**: Multiplicative attention
- **Self-Attention**: Query = Key = Value

Benefits for Time Series:

- Focus on relevant time steps
- Handle long sequences
- Interpretability
- Variable importance weighting

c_t

Visualization:

Energy Load Forecasting Applications

Time Series Data Preparation

Feature Engineering:

- **Temporal**: Hour, day, month, season
- **Lag Features**: t-1, t-24, t-168
- **Rolling Statistics**: Mean, std, min, max
- **Calendar**: Holidays, weekends
- **Weather**: Temperature, humidity
- **Economic**: Industrial indices

Normalization:

- StandardScaler: $z = \frac{x-\mu}{\sigma}$
- MinMaxScaler: $x' = \frac{x-x_{min}}{x_{max}-x_{min}}$
- RobustScaler: Using median/IQR

Sequence Creation:

- Input window: 24-168 hours
- Output horizon: 1-24 hours
- Sliding window approach
- Overlap considerations

Data Splits:

- Train: 70%
- Validation: 15%
- Test: 15%
- **Important**: Temporal order

Data Pipeline:

`05 lstm energy advanced.ipynb`

Multi-Step Forecasting Strategies

1. **Recursive (Single-Step):**
   - Train: One step ahead
   - Inference: Use predictions as input
   - Issue: Error accumulation
2. **Direct (Multi-Output):**
   - Train: All horizons simultaneously
   - Separate model per horizon
   - Issue: No correlation between outputs
3. **MIMO (Seq2Seq):**
   - Train: Sequence to sequence
   - Single model for all horizons
   - Best for related outputs

Hybrid Approaches:

- DirRec: Combine direct and recursive
- Multi-stage: Coarse to fine
- Ensemble: Multiple strategies

Evaluation Metrics:

- MAE: $\frac{1}{n}\sum |y - \hat{y}|$
- RMSE: $\sqrt{\frac{1}{n}\sum (y - \hat{y})^2}$
- MAPE: $\frac{100}{n}\sum \frac{|y-\hat{y}|}{y}$
- R^2: Explained variance

Handling Seasonality and Trends

Decomposition:

$$Y_t = T_t + S_t + R_t$$

- T_t: Trend component
- S_t: Seasonal patterns
- R_t: Residual/Random

Multiple Seasonalities:

- Daily: 24-hour cycle
- Weekly: 7-day pattern
- Annual: Seasonal variations

Neural Approaches:

- Seasonal neurons
- Fourier features
- Wavelet decomposition
- STL decomposition

Hybrid Models:

- SARIMA + LSTM
- Prophet + Neural residuals
- Decomposition + Deep learning

Seasonality Handling:

`05_lstm_energy_advanced.ipynb`

Advanced Techniques and Optimization

Temporal Fusion Transformer (TFT)

Architecture Components:

- **Variable Selection**: Feature importance
- **Gating**: Suppress unnecessary info
- **Static Enrichment**: Context encoding
- **Temporal Processing**: LSTM encoder
- **Multi-Head Attention**: Self-attention
- **Position-wise FFN**: Final processing

Interpretability:

- Variable importance scores
- Attention weights visualization
- Temporal patterns identification

Advantages:

- Handles multiple time series
- Known/unknown variables
- Multi-horizon forecasting
- Uncertainty quantification
- State-of-the-art performance

TFT Implementation:

`05_lstm_energy_advanced.ipynb`

Simplified version included

Training Best Practices

Gradient Management:

- Gradient clipping: $\|\mathbf{g}\| \leq$ threshold
- Gradient normalization
- Adaptive clipping
- Skip connections

Regularization:

- Dropout (standard/variational)
- Weight decay
- Early stopping
- Noise injection
- Zoneout (for RNNs)

Learning Rate Scheduling:

- Warmup phase
- Cosine annealing
- ReduceLROnPlateau
- Cyclical learning rates

Loss Functions:

- MSE: Standard regression
- MAE: Robust to outliers
- Huber: Combination
- Quantile: Uncertainty
- Custom: Peak-aware

Model Optimization and Deployment

Optimization Techniques:

- Teacher forcing ratio decay
- Scheduled sampling
- Curriculum learning
- Transfer learning
- Multi-task learning

Inference Optimization:

- Beam search
- Caching hidden states
- Batch processing
- Model quantization
- ONNX export

Production Deployment:

- Real-time inference pipeline
- Online learning updates
- Model versioning
- A/B testing
- Monitoring & alerts

Performance Metrics:

- Latency (ms/prediction)
- Throughput (predictions/sec)
- Memory usage
- Model size

Uncertainty Quantification

Probabilistic Forecasting:

- Quantile regression
- Prediction intervals
- Monte Carlo dropout
- Deep ensembles
- Bayesian RNNs

Quantile Loss:

$$L_q(y, \hat{y}) = \begin{cases} q(y - \hat{y}) & y \geq \hat{y} \\ (1 - q)(\hat{y} - y) & y < \hat{y} \end{cases}$$

Applications:

- Risk assessment
- Decision making
- Grid stability
- Reserve planning
- Trading strategies

Evaluation:

- Coverage probability
- Interval width
- CRPS (Continuous Ranked Probability Score)
- Pinball loss

Summary: RNNs & LSTMs for Energy Systems

Key Concepts:

- RNN fundamentals
- LSTM/GRU architectures
- Bidirectional processing
- Attention mechanisms
- Encoder-decoder models

Energy Applications:

- Load forecasting
- Multi-step prediction
- Seasonality handling
- Anomaly detection
- Demand response

Advanced Techniques:

- Temporal Fusion Transformer
- Uncertainty quantification
- Online learning
- Production deployment

Complete Implementation:

`05_lstm_energy_advanced.ipynb`

All Notebooks Available:

03_deep_learning_advanced.ipynb

04_cnn_solar_advanced.ipynb

05_lstm_energy_advanced.ipynb