Session 1: LSTMs for Earth Observation Time Series

Understanding Recurrent Neural Networks and Long Short-Term Memory

Stylianos Kotsopoulos

Session Overview

Duration: 1.5 hours (90 minutes) **Type:** Theory Session **Goal:** Master LSTM networks for time series forecasting

You will learn:

- Why time series analysis is critical for EO
- Limitations of standard neural networks for sequences
- LSTM architecture and how it solves vanishing gradients
- EO applications for LSTM-based forecasting
- How LSTM gates control information flow

Prerequisites:

- Understanding of CNNs (Day 3)
- Basic neural network training
- Time series concepts
- Python fundamentals

Materials:

- Theory presentation
- Philippine EO case studies
- Reference materials overview

Part 1: Time Series in EO

Why Time Series Matter

Earth observation data is inherently **temporal** - we observe the same locations repeatedly over time.

Static Analysis (Single Date):

- Land cover classification
- Feature detection
- Snapshot assessments

Time Series Analysis (Multi-Date):

- Vegetation phenology and growth cycles
- Drought onset and recovery
- Crop yield forecasting
- Deforestation detection
- Seasonal pattern analysis
- Climate change trend identification

The Fourth Dimension of EO

Spatial + Temporal Analysis

While CNNs excel at extracting **spatial patterns** from individual satellite images, **LSTMs excel at extracting temporal patterns** from sequences of observations. Together, they form powerful tools for spatiotemporal analysis.

Common EO Time Series

Vegetation Indices:

- **NDVI:** Vegetation health, crop growth, drought stress
- **EVI:** Better for high-biomass areas (tropical forests)
- **SAVI:** Reduces soil background effects

SAR Backscatter:

- **VV, VH polarization:** Flooding, harvest, vegetation changes
- **Coherence:** Surface stability over time

Biophysical Parameters:

- **LAI:** Crop canopy development
- **FPAR:** Productivity indicator
- **LST:** Heat stress, urban heat island

Philippine Seasonal Patterns

Dry Season (Nov-Apr)

- Lower NDVI in rain-fed areas
- Reduced soil moisture
- Increased drought risk (Mindanao)

Wet Season (May-Oct)

- Peak NDVI during growth
- Rice planting seasons
- Flood risk (typhoon-prone)

. . .

Climate Impacts:

- **El Niño:** Prolonged dry conditions, delayed planting, reduced yields
- **La Niña:** Enhanced rainfall, potential flooding, pest outbreaks

---

Mindanao Case Study

> Agricultural Context
>
> **Provinces:** Bukidnon and South Cotabato
> **Major Crops:**
>
> - Corn (maize)
> - Rice
> - Pineapple
> - Coffee
> - Sugarcane
>
> **2015-2016 El Niño Impact:**
>
> - Severe drought causing significant crop losses
> - **Need:** Predict drought 1-3 months ahead for early interventions

---

LSTM Applications for EO

1. Drought Forecasting

- Input: NDVI, rainfall, temperature sequences
- Output: Predicted NDVI 1-3 months ahead
- Benefit: Early warning for agricultural planning

2. Crop Yield Prediction

- Input: In-season NDVI, weather, SAR
- Output: Estimated yield at harvest
- Benefit: Food security planning

3. Flood Risk Assessment

- Input: Precipitation, discharge, soil moisture
- Output: Predicted flood probability
- Benefit: Disaster preparedness

More Applications

4. Land Cover Change Detection

- Input: Multi-temporal optical and SAR
- Output: Change probability, anomaly detection
- Benefit: Deforestation monitoring

5. Phenology Monitoring

- Input: NDVI/EVI time series
- Output: Predicted crop stage, harvest date
- Benefit: Precision agriculture

Part 2: RNN Limitations

Why Standard Networks Fail

Feedforward Networks (including CNNs) assume inputs are **independent**.

Problem with Sequential Data:

- Each input depends on previous inputs
- Context matters: Today's NDVI depends on past weeks
- Fixed input size challenge

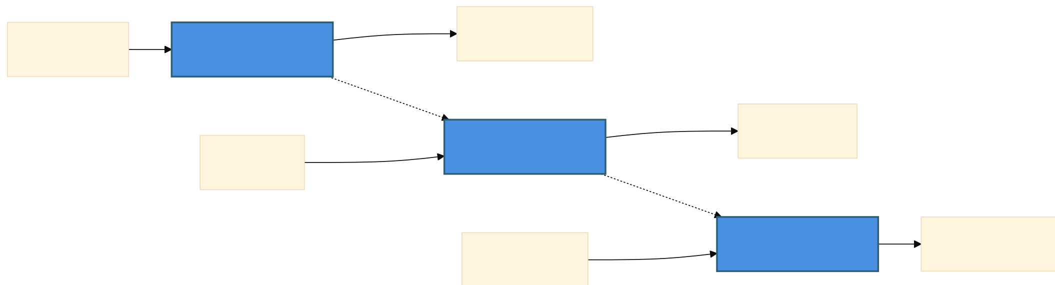
Example: Predicting next month's vegetation health

- Feedforward: Each month treated independently (no memory)
- But we know: If NDVI declining for 3 months → drought worsening

Solution: Networks with **memory** of previous inputs

Recurrent Neural Networks

Key Idea: Add **feedback loop** to remember previous inputs



How RNNs Work

At each time step `t`:

1. Receive input `x_t` (e.g., current month's NDVI)
2. Combine with previous hidden state `h_{t-1}` (memory)
3. Compute new hidden state `h_t`
4. Produce output `y_t`
5. Pass `h_t` to next time step

Mathematical Formulation:

$$h_t = \tanh(W_{hh} \cdot h_{t-1} + W_{xh} \cdot x_t + b_h)$$

$$y_t = W_{hy} \cdot h_t + b_y$$

RNN Advantages

- Handles variable-length sequences
- Maintains memory across time steps
- Shares weights across time (parameter efficiency)
- Designed for temporal dependencies

The Vanishing Gradient Problem

Critical Flaw of Standard RNNs:

When training on long sequences (e.g., 24 months), gradients become extremely small during backpropagation.

Why This Happens:

Gradients multiply repeatedly:

$$\frac{\partial L}{\partial h_1} = \frac{\partial L}{\partial h_T} \cdot \frac{\partial h_T}{\partial h_{T-1}} \cdot ... \cdot \frac{\partial h_2}{\partial h_1}$$

If each derivative < 1, product shrinks exponentially

Gradient Decay Consequences

Impact:

- **Vanishing gradients:** Cannot learn long-term dependencies (e.g., drought from 6 months ago)
- **Exploding gradients:** Less common, but gradients can grow exponentially

EO Application Impact:

Predicting August drought based on:

- July data: RNN learns easily
- April-June: Partially learned
- January-March: **Lost due to vanishing gradients**

But January-March dry season conditions are **critical** for August prediction!

Mini-Challenge: Gradient Decay

Calculate Gradient Decay

Task: How many time steps for gradient of 0.9 to shrink below 0.01?
Formula: $0.9^n < 0.01$
Answer: $n = \frac{\log(0.01)}{\log(0.9)} \approx 44$ steps
Meaning:

- Standard RNN: Learn only ~44 recent steps
- Monthly data: Less than 4 years
- 10-day composites: Less than 15 months

This is why LSTMs are essential!

Part 3: LSTM Architecture

What is an LSTM?

Long Short-Term Memory networks solve the vanishing gradient problem

Key Innovation:

Replace simple hidden state with a **memory cell** controlled by learnable **gates**

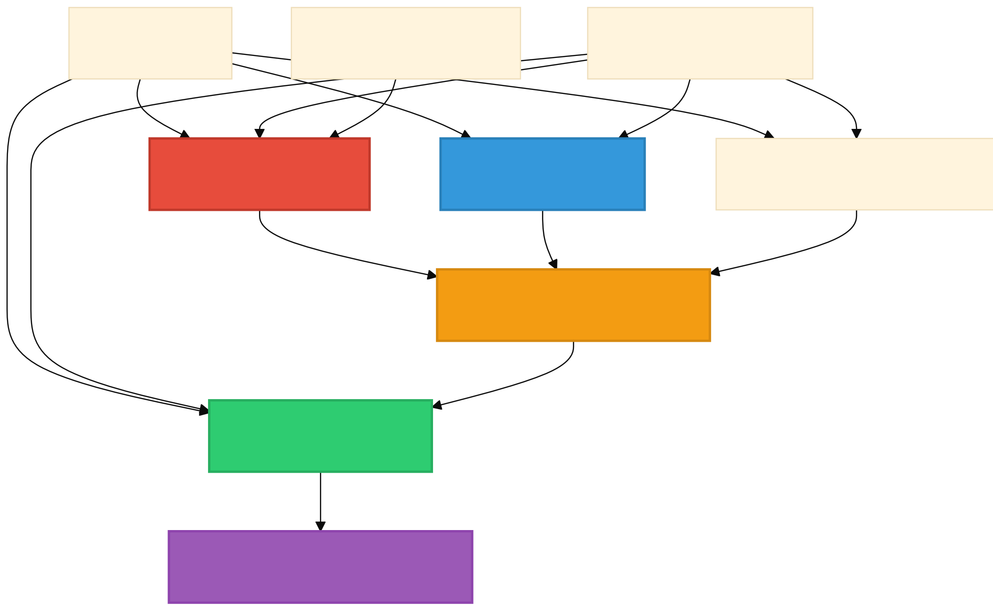
LSTM Advantages:

- Learn long-term dependencies (100+ time steps)
- Selective memory: Remember important, forget irrelevant
- Gradient flow preserved through time

LSTM Cell Structure

Components:

- **Cell State (C__t):** Long-term memory "conveyor belt"
- **Hidden State (h__t):** Short-term memory and output
- **Three Gates:** Control information flow



The Three Gates

1. Forget Gate

- Purpose: Decide what to discard from cell state
- Question: "Should I forget old information?"
- Example: Dry season ended → forget drought patterns

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Output: 0 (forget) to 1 (keep)

Input Gate

2. Input Gate

- Purpose: Decide what new information to add
- Question: "What new information should I remember?"
- Example: Wet season started → remember rainfall pattern

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

- `i_t`: How much to add (0 to 1)
- `C_tilde_t`: Candidate values

Output Gate

3. Output Gate

- Purpose: Decide what to output from cell state
- Question: "What should I output this time step?"
- Example: Output drought risk based on accumulated evidence

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

Cell State Update

Step 1: Forget old information

$$C_t = f_t \cdot C_{t-1}$$

Step 2: Add new information

$$C_t = C_t + i_t \cdot \tilde{C}_t$$

Combined:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

Why LSTMs Solve Vanishing Gradients

Key Insight: Cell state acts as **gradient highway**

Standard RNN:

- Gradient multiplied by weight matrix at each step → decay

LSTM:

- Gradient flows through cell state with **element-wise operations**
- Gate values learned, can be close to 1
- Minimal gradient decay

Result:

- Learn dependencies over 100+ time steps
- Remember events from months ago
- Forget irrelevant fluctuations

Think-Through Discussion

Drought Monitoring Example

Question: What might the gates do for Mindanao drought?
Forget Gate:

- Discard normal seasonal fluctuations
- Remove short-term weather noise

Input Gate:

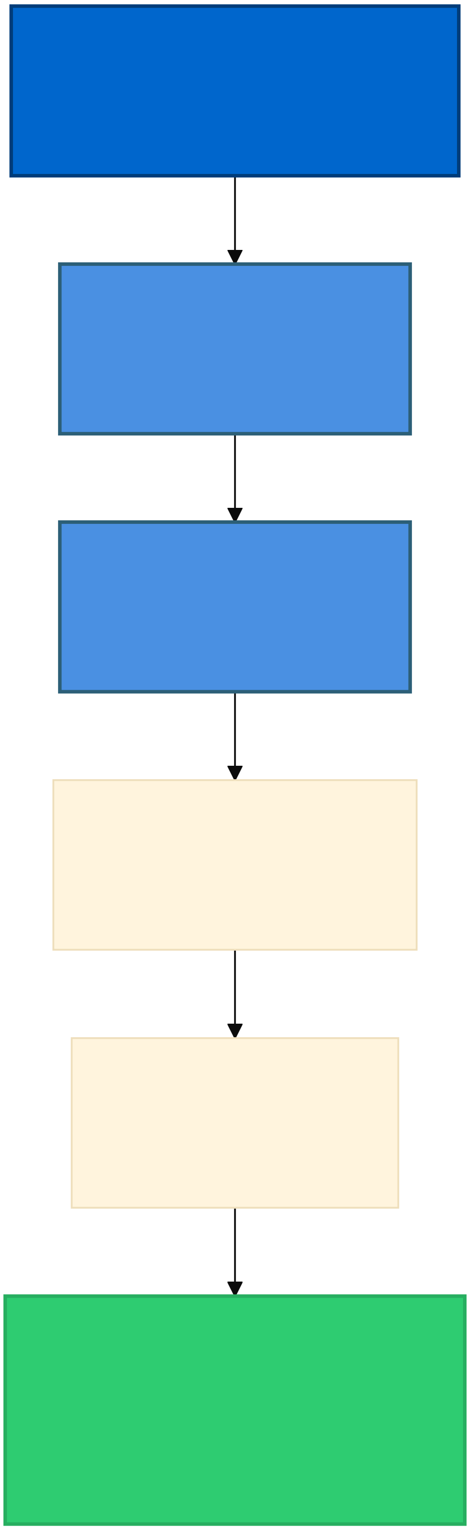
- Preserve El Niño indicators
- Remember declining NDVI trend
- Store anomalous patterns

Reflection: How would gates behave differently for:

1. Typical seasonal NDVI decline (expected)
2. Anomalous drought event (unexpected)

Part 4: LSTM for EO

LSTM Network Architecture



Components:

- Input: Sequence of observations (12 months NDVI)
- LSTM Layers: Extract temporal patterns (1-2 layers)
- Dense Layers: Map LSTM output to prediction
- Output: Predicted value(s)

Input Data Preparation

Sequence Creation (Sliding Window):

Given monthly NDVI 2015-2021, create training sequences:

Lookback: 12 months **Forecast:** 1 month ahead

Example:

- Seq 1: [Jan 2015, …, Dec 2015] → Predict Jan 2016
- Seq 2: [Feb 2015, …, Jan 2016] → Predict Feb 2016
- Seq 3: [Mar 2015, …, Feb 2016] → Predict Mar 2016

Result: Hundreds/thousands of training sequences from multi-year data

---

Multivariate Inputs

LSTMs use multiple features per time step:

- NDVI (vegetation health)
- Rainfall (water availability)
- Temperature (heat stress)
- Soil moisture
- Previous year same month

Input Shape: `(samples, time_steps, features)`

Example: `(5000, 12, 4)` = 5000 sequences, 12 months, 4 features

Training Process

1. Data Splitting:

- Training: 2015-2019 (80%)
- Validation: 2020 (10%)
- Test: 2021 (10%)

Important: Temporal splits (not random) to avoid leakage

2. Normalization:

- Scale to [0, 1] or standardize (mean=0, std=1)

3. Model Compilation:

- Loss: Mean Squared Error (MSE)
- Optimizer: Adam
- Metrics: RMSE, MAE

Training Details

4. Training:

- Batch size: 32-128
- Epochs: 50-200 (early stopping)
- Monitor validation loss

5. Evaluation:

- Test set predictions vs actual
- Visualize time series
- Calculate error metrics

Parameter	Description	Typical Range

Hyperparameters

Parameter	Description	Typical Range
LSTM units	Hidden state size	32-256 per layer
Layers	LSTM stack depth	1-3
Lookback	Time steps to use	6-24 months
Dropout	Regularization	0.1-0.3
Learning rate	Optimization step	0.0001-0.01
Batch size	Samples per update	32-128

Philippine Applications

Mindanao Drought Monitoring

Objective: Predict drought 1-3 months ahead for Bukidnon & South Cotabato

Data Sources:

- Sentinel-2 NDVI: 2015-2021 (10-day)
- PAGASA rainfall: Monthly accumulation
- PAGASA temperature: Monthly mean
- El Niño index (ONI): NOAA data

Model Setup:

- Input: 12-month sequences (NDVI, rainfall, temp, ONI)
- Output: NDVI prediction 1 month ahead
- Architecture: 2 layers (64, 32 units), dropout 0.2

Expected Results

Training:

- Historical: 2015-2019
- Validation: 2020
- Test: 2021

Performance:

- RMSE < 0.05 on NDVI scale [0-1]
- Early detection 1-3 months in advance
- Correlation with reported crop losses

Operational:

- Monthly predictions as new data arrives
- Alerts to DA, PAGASA, LGUs
- Integration with agricultural advisory systems

Session 2 Lab Preview

> **!** Session 2 Hands-On Lab (Today)
>
> Session 2 (2.5 hours) implements full workflow:
>
> - Download Sentinel-2 NDVI for Mindanao
> - Prepare sequences and training data
> - Build and train LSTM model
> - Evaluate predictions
> - Visualize results
> - Discuss operational deployment

Other Philippine Applications

1. Rice Yield Forecasting (Luzon)

- SAR backscatter + NDVI time series
- Yield 1 month before harvest
- DA food security planning

2. Typhoon Impact Prediction (Visayas)

- Pre-typhoon NDVI, rainfall, wind
- Expected NDVI drop (damage proxy)
- Pre-position relief supplies

3. Coral Bleaching (Palawan)

- Sea surface temperature series
- Bleaching risk 2-4 weeks ahead
- DENR early warning for MPAs

4. Urban Growth (Metro Manila)

- Historical built-up area
- Urban expansion locations
- MMDA infrastructure planning

Hands-On Exercise

Build Your First LSTM

Step 1: Data Preparation

- Generate synthetic Mindanao NDVI (2019-2024)
- Visualize seasonal patterns
- Create sliding window sequences

Step 2: Understand Gradients

- Calculate vanishing gradient decay

- Compare RNN vs LSTM
- Visualize gradient flow

Model Building

Step 3: Build and Train

```
model = Sequential([
    LSTM(64, return_sequences=True, input_shape=(12, 1)),
    Dropout(0.2),
    LSTM(32),
    Dropout(0.2),
    Dense(16, activation='relu'),
    Dense(1)
])
```

Step 4: Evaluate

- Compare predictions vs actual
- Calculate RMSE and MAE
- Visualize drought prediction accuracy

Expected Results

- Prediction accuracy: MAE $<$ 0.05 NDVI units
- Drought detection: 80%+ accuracy
- Training time: ~3-5 minutes on CPU

Open Student Notebook to begin!

Key Takeaways

Summary: Time Series

> ❗ Important
>
> **Time Series in EO:**
>
> - Unlocks temporal patterns invisible in single images
> - Philippine agriculture: Strong seasonal cycles
> - Applications: Drought, yield, phenology, change detection
> - Fourth dimension beyond spatial analysis

Summary: RNNs

> ❗ Important
>
> **RNNs and Limitations:**
>
> - Standard networks can't handle sequences
> - RNNs add memory via recurrent connections
> - Vanishing gradient: Can't learn >10 time steps
> - Limits: ~44 steps with 0.9 gradient retention

Summary: LSTM

> ❗ Important
>
> **LSTM Architecture:**
>
> - Three gates (forget, input, output) control flow
> - Cell state = long-term memory "conveyor belt"
> - Gradients flow without decay → 100+ steps
> - Selective memory: Remember important, forget noise

Summary: EO Forecasting

> **!** Important
>
> **LSTM for EO:**
>
> - Input: Sequences via sliding windows
> - Architecture: Stacked LSTM + dense layers
> - Output: Predicted values 1-N steps ahead
> - Training: Temporal splits, MSE loss, Adam
>
> **Philippine Context:**
>
> - Mindanao drought: NDVI 1-3 months ahead
> - Multivariate: NDVI + rainfall + temp + climate
> - Operational: Early warning for agencies

Next Session

> Session 2: Hands-On Lab (Today)
>
> **Session 2** implements full LSTM drought monitoring for Mindanao with real Sentinel-2 data!
> **To Prepare:**
>
> 1. Python environment with TensorFlow
> 2. Review LSTM concepts from today
> 3. Understand sequence preparation
> 4. Think about drought indicators
>
> **Software:** Python 3.8+, TensorFlow 2.x, GEE account

Resources

Research Papers

- Hochreiter & Schmidhuber (1997). "Long Short-Term Memory"

- Gers et al. (2000). “Learning to Forget with LSTM”
- Ndikumana et al. (2018). “Deep RNN for Agricultural Classification”

Tutorials

- Understanding LSTM Networks (colah’s blog)
- Keras LSTM Tutorial
- TensorFlow Time Series

Philippine EO Data

- PAGASA Climate Data
- CoPhil Infrastructure
- Google Earth Engine

Questions & Discussion

Think About:

- What time series problems in your work need LSTM?
- What features beyond NDVI improve drought prediction?
- How far ahead can we realistically forecast?
- What are the limitations and uncertainties?

Contact: skotsopoulos@neuralio.ai

Thank You!

Questions?

See you in Session 2 for the hands-on lab!

Day 4: Time Series Analysis, Emerging Trends, and Sustainable Learning - CoPhil 4-Day Advanced Training on AI/ML for Earth Observation, funded by the European Union under the Global Gateway initiative.