

Lab: Time Series & Recurrent Neural Networks

ACTL3143 & ACTL5111 Deep Learning for Actuaries

Part 1: Time series as ordered data

Exercise 1: Tabular data versus time-series data

In a standard supervised-learning problem, we often write observations as

$$\{(x_i, y_i)\}_{i=1}^n.$$

In a time-series problem, observations arrive in order:

$$\{(x_t, y_t)\}_{t=1}^T.$$

1. What is the difference between the index i and the time index t ?
 2. Why does the order of observations matter in a time-series problem?
 3. Why might the independence assumption be less reasonable for time-series data?
-

Exercise 2: Examples of sequence problems

For each of the following applications, identify the input sequence and the prediction target.

1. Forecasting future claim payments.
2. Forecasting mortality rates over calendar years.
3. Forecasting daily stock returns.
4. Forecasting tomorrow's temperature.
5. Predicting the next word in a sentence.

For each example, also decide whether the sequence is naturally indexed by time or by position in a sequence.

Exercise 3: Prediction target and information available at time t

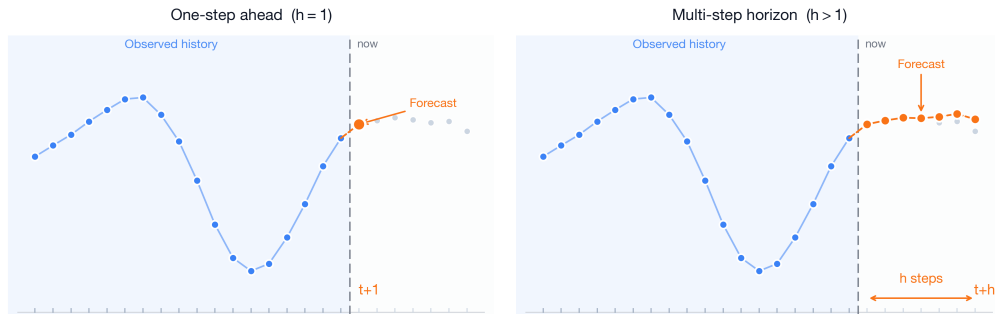


Figure 1: Forecasting problem: using information available at time t to predict future values.

Suppose that at each time point t , we observe predictors x_t and a response y_t .

A common forecasting problem is to predict y_{t+1} using information available up to time t :

$$(x_1, y_1), (x_2, y_2), \dots, (x_t, y_t) \longrightarrow y_{t+1}.$$

1. What information is available when predicting y_{t+1} ?
2. If x_{t+1} is only observed at time $t + 1$, should it be used to predict y_{t+1} at time t ?
3. In a real forecasting problem, why is it important to define the prediction time clearly?

Exercise 4: Trend, seasonality and noise

Consider a daily temperature series.

1. What is meant by trend, seasonality and noise?
2. Which of these components would you expect to be important for weather data?
3. Which of these components would you expect to be important for daily stock returns?
4. If we use only the previous 7 days of temperature to predict tomorrow's temperature, can the model easily learn annual seasonality?
5. What extra information could we give to an RNN to help it learn seasonal patterns?

6. Why might sine and cosine transformations of the day of year be useful seasonal features?
-

Part 2: Classical time-series models

Before using RNNs, it is useful to understand classical time-series models and simple baselines.

An autoregressive model uses previous values of the series to predict the next value. For example,

$$y_t = \alpha + \phi_1 y_{t-1} + \epsilon_t.$$

ARIMA models extend this idea by combining autoregression, differencing, and moving-average errors.

Answer the following questions.

1. In the autoregressive model above, what does y_{t-1} represent?
 2. What does the coefficient ϕ_1 measure?
 3. Why might differencing be useful when the series has a trend?
 4. What is a persistence forecast?
 5. What is a seasonal naive forecast?
 6. Why should an RNN be compared against simple baselines such as persistence or seasonal naive forecasts?
 7. Why might ARIMA be a strong method for a single short time series?
 8. Why might RNNs be more attractive when we have many related sequences or many predictors?
-

Part 3: Simple recurrent neural networks

Exercise 5: RNN intuition

A SimpleRNN processes a sequence one step at a time.

At time t , the hidden state is updated using the current input and the previous hidden state:

$$h_t = f(W_x x_t + W_h h_{t-1} + b).$$

1. What is x_t , h_{t-1} and h_t ?
2. Why does the hidden state act like a memory?
3. What role does the activation function f play?

Exercise 6: Reusing weights across time

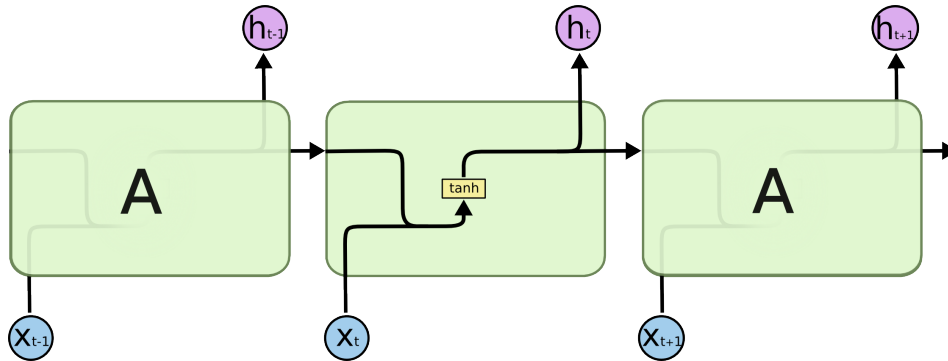


Figure 2: SimpleRNN cell architecture.

In an RNN, the same weight matrices are used at every time step.

1. Why is this different from using a separate dense layer for each time step?
2. Why does weight sharing make sense for sequence data?
3. How does weight sharing reduce the number of parameters?
4. What would happen to the number of parameters if we used separate weights for every time step?

Exercise 7: Manual RNN calculation

Consider a very small RNN with scalar inputs and scalar hidden states:

$$h_t = \tanh(w_x x_t + w_h h_{t-1} + b).$$

Let

$$w_x = 0.5, \quad w_h = 0.2, \quad b = 0, \quad h_0 = 0.$$

For the input sequence

$$x_1 = 2, \quad x_2 = 1,$$

calculate:

1. h_1 ;
2. h_2 .

You may leave your answers in terms of $\tanh(\cdot)$.

Exercise 8: Interpreting RNN input shape

An RNN input array has dimension

$$(1000, 10, 3).$$

Interpret this dimension vector.

1. What does the first dimension, 1000, represent?
2. What does the second dimension, 10, represent?
3. What does the third dimension, 3, represent?
4. How many values are contained in one sequence?
5. If this is daily weather data, give an example of what the 3 features at each time step could be.
6. Why does an RNN require this three-dimensional structure instead of a two-dimensional matrix?

Exercise 9: RNNs for text sequences

RNNs can also be used for natural language processing because text is ordered sequence data.

Consider the sentence:

“The policyholder reported a car accident.”

1. What are the sequence elements in this sentence?
2. Why does word order matter in text?
3. How could the words be converted into numerical inputs for an RNN?
4. If we use an RNN to classify accident reports into severity classes, what is the input sequence and what is the prediction target?

Exercise 10: Many-to-one and many-to-many problems

For each task below, decide whether it is many-to-one or many-to-many.

1. Use the last 20 days of weather variables to predict tomorrow's weather class.
 2. Use a sentence to predict whether a review is positive or negative.
 3. Translate an English sentence into French.
 4. Predict the next value at every time point in a sequence.
 5. Use a patient's history to predict whether they will be hospitalised next month.
-

Part 4: Limitations of SimpleRNN

Exercise 11: Long-term dependence

SimpleRNNs can struggle with long sequences.

1. What does long-term dependence mean?
 2. Why might yesterday's temperature be more useful than the temperature 300 days ago?
 3. Give an example where information from much earlier in the sequence could still matter.
 4. Why might SimpleRNNs have difficulty learning this?
-

Exercise 12: Vanishing gradients

When training an RNN, gradients are propagated backward through time.

1. What does "backward through time" mean?
 2. What is the vanishing-gradient problem?
 3. Why does this make it difficult to learn long-term dependencies?
 4. Why is this problem less severe for short sequences?
-

Part 5: LSTM and GRU

Exercise 13: Motivation for LSTM

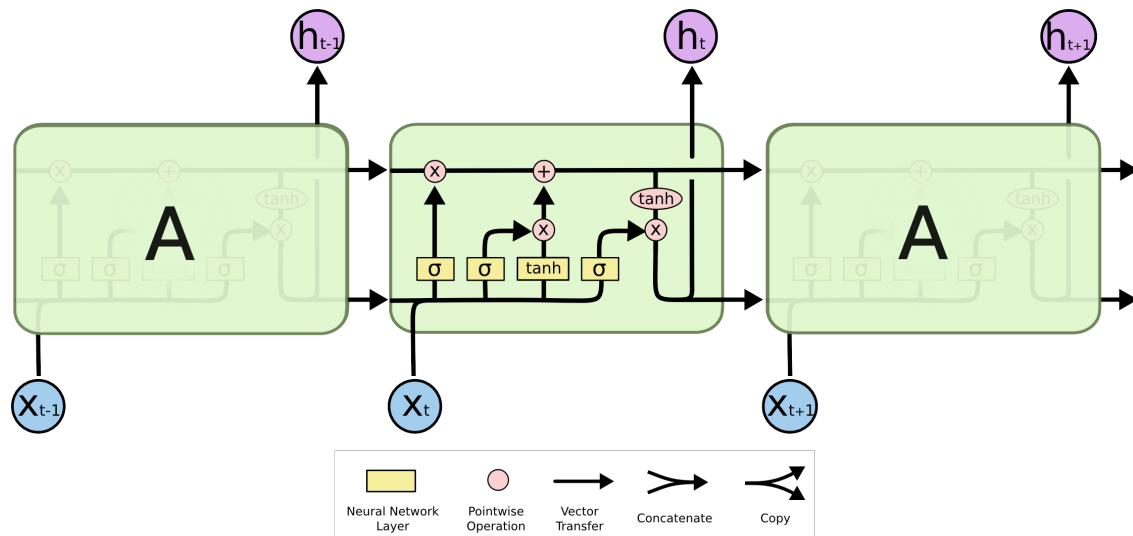
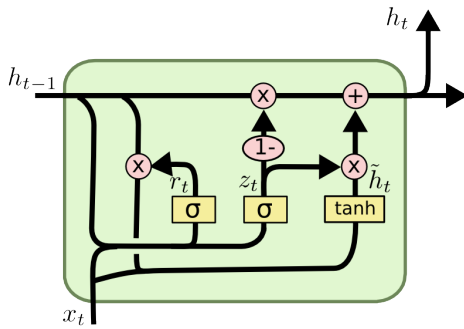


Figure 3: LSTM cell architecture with gating mechanisms.

LSTM models were designed to improve the handling of long-term dependencies.

1. What problem is LSTM trying to solve?
2. What is the purpose of having a cell state?
3. What is the intuitive role of a forget gate?
4. What is the intuitive role of an input gate?
5. What is the intuitive role of an output gate?

Exercise 14: Motivation for GRU



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 4: GRU cell architecture.

GRU is a simpler alternative to LSTM.

1. What problem is GRU also trying to solve?
 2. How is GRU conceptually similar to LSTM?
 3. Why might GRU be faster to train than LSTM?
 4. In practice, why is it useful to compare SimpleRNN, LSTM and GRU rather than assuming one is always best?
-

Exercise 15: Comparing SimpleRNN, LSTM and GRU

Suppose you fit a SimpleRNN, an LSTM and a GRU to the same forecasting problem.

1. Which model has the simplest recurrent structure?
 2. Which models are designed to better handle long-term dependence?
 3. Which model may have more parameters?
 4. What validation metric would you use for a regression forecasting problem?
 5. What validation metric would you use for a classification forecasting problem?
-

Part 6: Application: weather prediction with RNNs

In this section, we use weather prediction as a worked application of RNNs. The goal is to practise the full modelling workflow for sequence data.

Exercise 16: Weather prediction as a sequence-modelling problem

Using the daily weather dataset, build and compare several models for predicting tomorrow's weather class.

Follow the steps below.

1. Load the dataset and inspect the variables.
2. Identify the response variable and the available predictors.
3. Explain why this can be viewed as a sequence-modelling problem.
4. Split the data into training, validation and test sets using chronological order rather than random splitting.
5. Preprocess the predictors:
 - scale the numerical variables;
 - encode the categorical weather variable;
 - make sure preprocessing is fitted using the training data only.
6. Create lagged sequence inputs.
For example, use the previous m days of observations to predict the weather class on the next day.
7. Check the shape of the RNN input array.
Explain what each dimension represents in

(samples, time steps, features).
8. Fit a simple baseline model, such as a majority-class or persistence baseline.
9. Fit a feedforward neural network using flattened lagged inputs.
10. Fit a SimpleRNN model using the sequence input.
11. Fit an LSTM model using the same sequence input.
12. Fit a GRU model using the same sequence input.
13. Compare the models using validation and test accuracy.
14. Plot or inspect the training histories of the neural-network models.
15. Discuss whether the recurrent models improve meaningfully over the simpler baselines.
16. Based on the results, decide which model you would choose for this application and justify your choice.