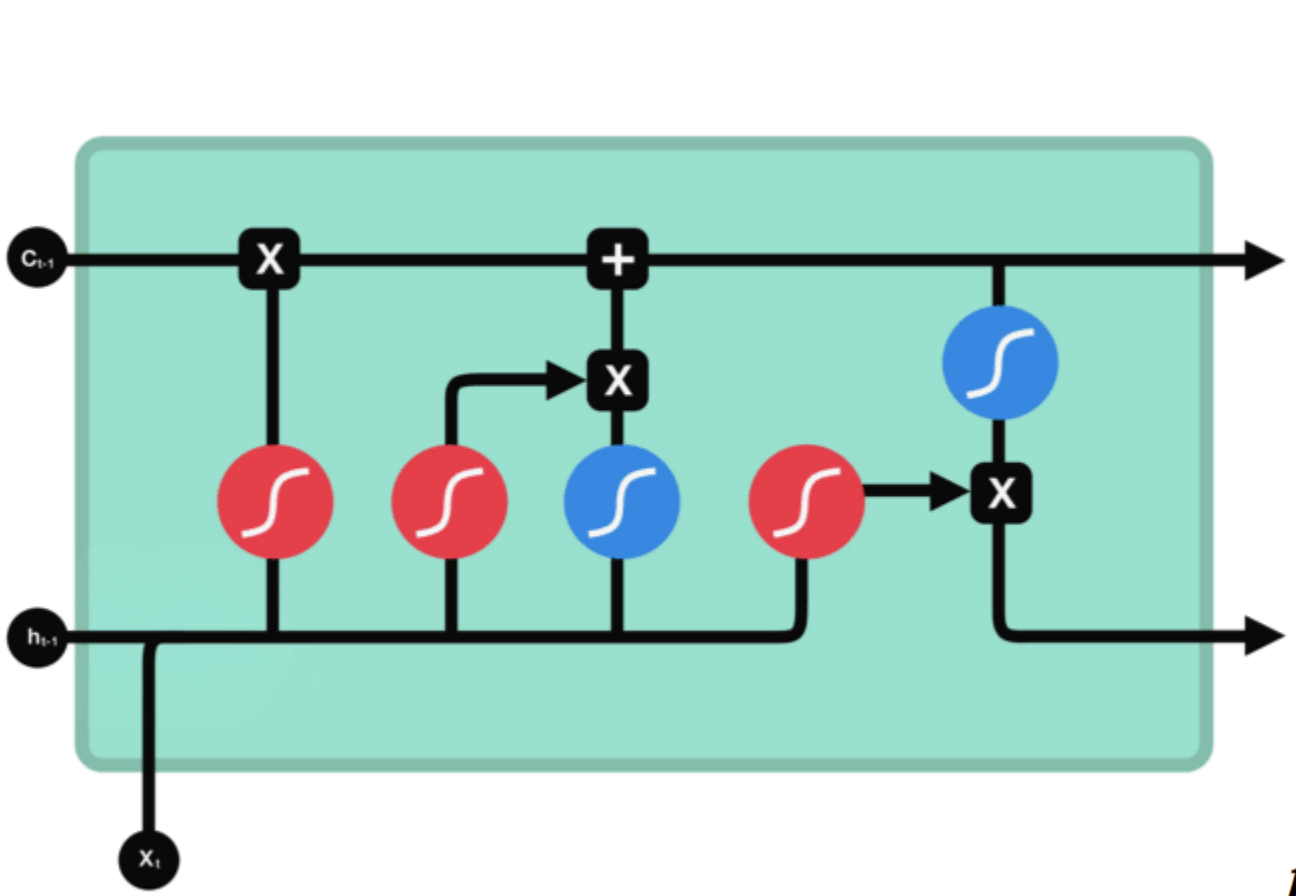
LSTM

Forget gate



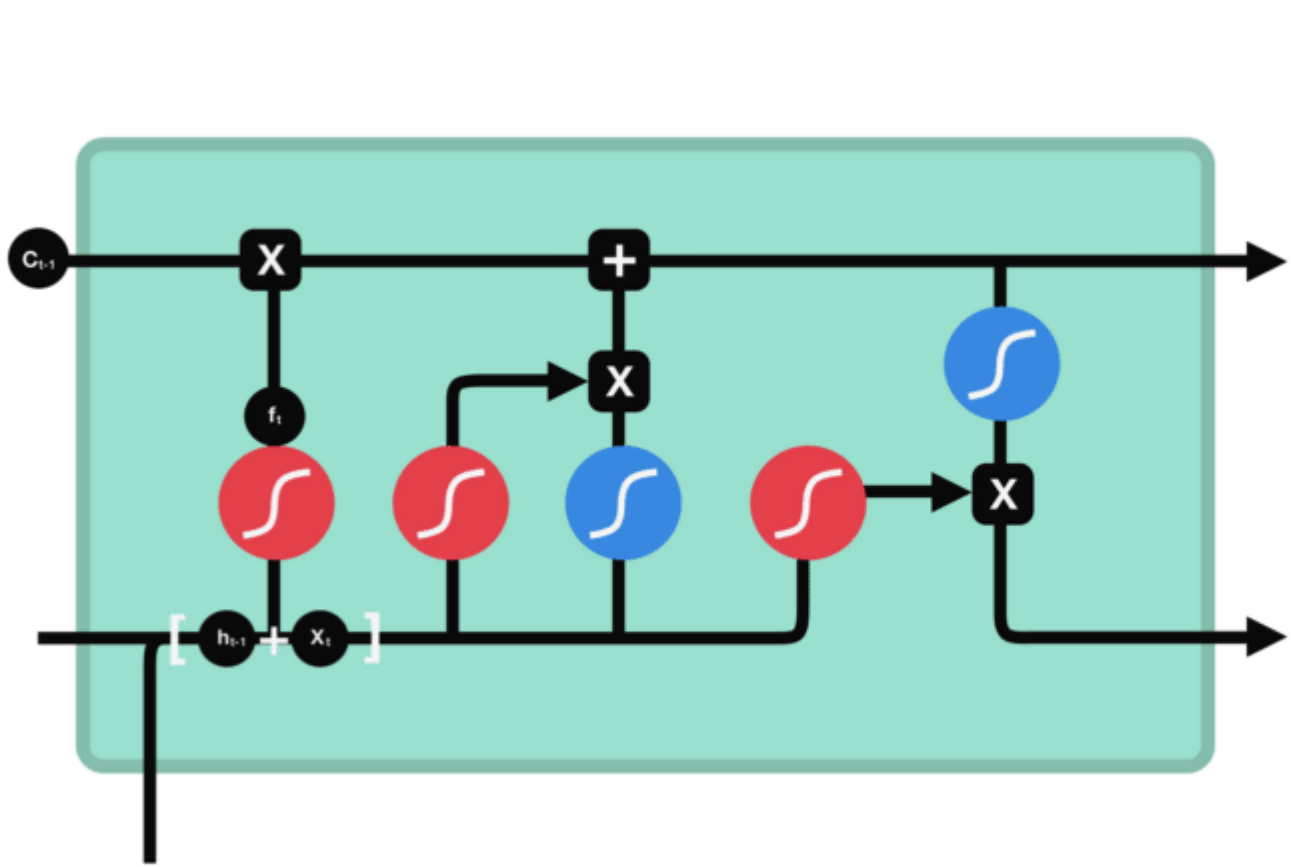
c_{t-1} previous cell state

f_t forget gate output

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_{xf}\boldsymbol{x}^{(t)} + \boldsymbol{W}_{hf}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_f\right)$$

Source: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Input gate

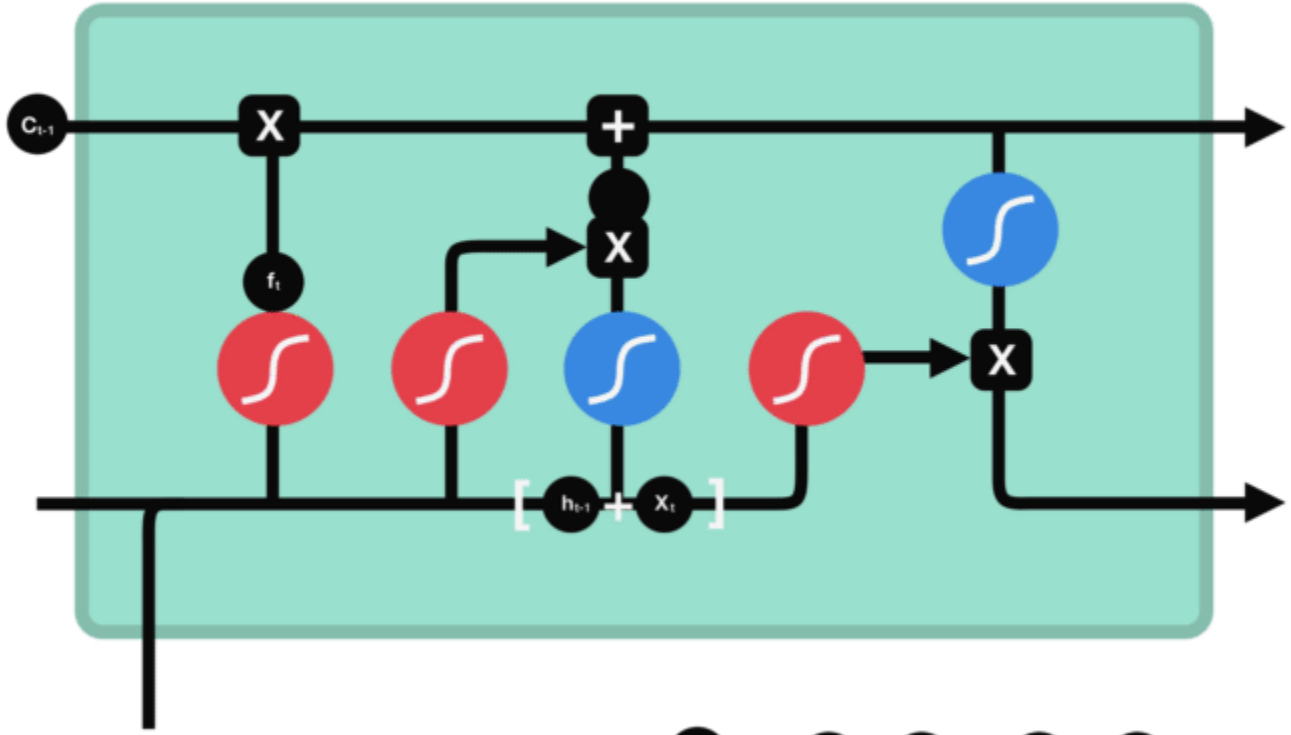


$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_{xi}\boldsymbol{x}^{(t)} + \boldsymbol{W}_{hi}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_i\right)$$

$$\boldsymbol{g}_t = \tanh\left(\boldsymbol{W}_{xg}\boldsymbol{x}^{(t)} + \boldsymbol{W}_{hg}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_g\right)$$

Source: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Cell State



C_{t-1} previous cell state

f_t forget gate output

i_t input gate output

$\tilde{C}_t$ candidate

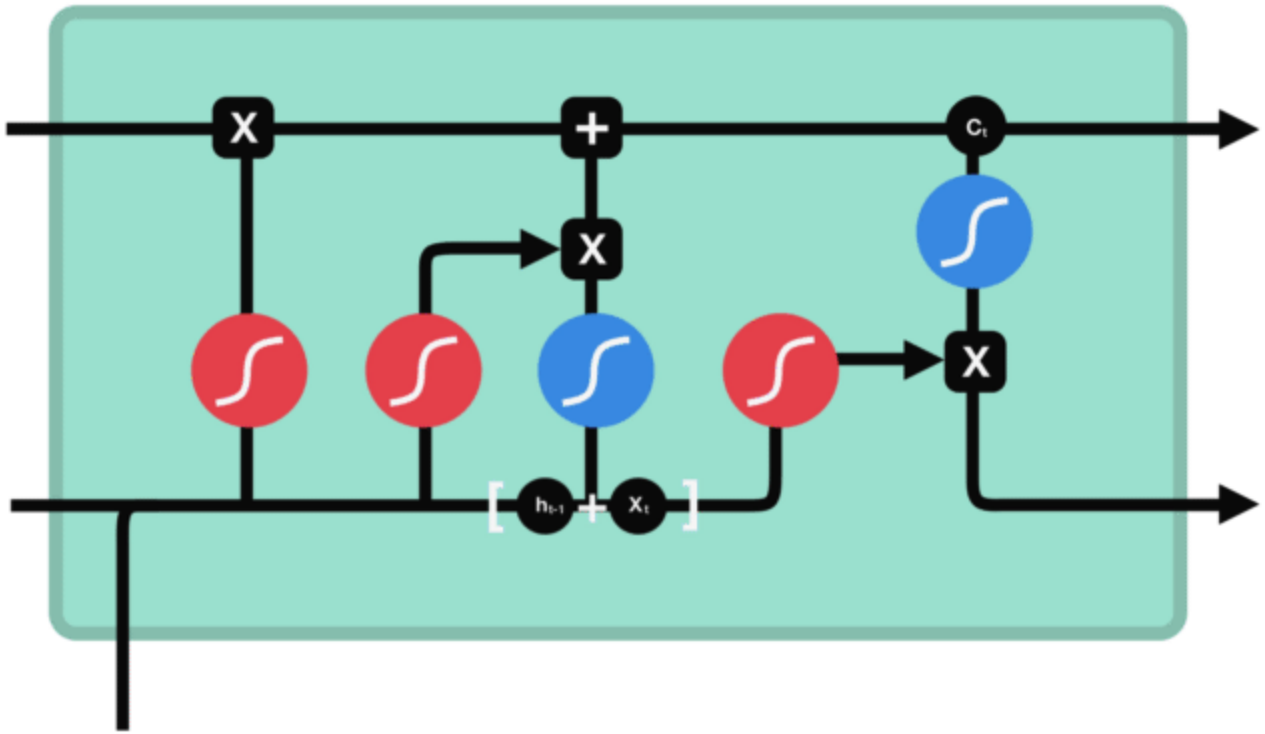
C_t new cell state

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

$$\boldsymbol{C}^{(t)} = \left(\boldsymbol{C}^{(t-1)} \odot \boldsymbol{f}_t\right) \oplus \left(\boldsymbol{i}_t \odot \boldsymbol{g}_t\right)$$

Source: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Output Gate



C_{t-1} previous cell state

f_t forget gate output

i_t input gate output

$\tilde{C}_t$ candidate

C_t new cell state

o_t output gate output

h_t hidden state

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_{xo}\boldsymbol{x}^{(t)} + \boldsymbol{W}_{ho}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_o\right)$$

$$\boldsymbol{h}^{(t)} = \boldsymbol{o}_t \odot \tanh\left(\boldsymbol{C}^{(t)}\right)$$

Source: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Training texts

I am from Germany and I eat lots of Bratwurst.

God bless the queen.

Life is like a box of choccolate.

- Token_index contains 500 000 words
- Length of sequence: T = 10

Question 1: what would be the dimension of the training data containing the three sentences above using One-hot-encoding?

Question 2: how many rows of zeros for each text?

Question 3: what happens if a word in the text is not contained in the token index?

Training texts

I am from Germany and I eat lots of Bratwurst.

God bless the queen.

Life is like a box of choccolate.

- Token_index contains 500 000 words
- Length of sequence: T = 10

A1: (3, 10, 500 000), samples x sequence length x length word index (one-hot-encoded)

Question 1: what would be the dimension of the training data containing the three sentences above using One-hot-encoding?

Question 2: how many rows of zeros for each text?

Question 3: what happens if a word in the text is not contained in the token index?

Training texts

I am from Germany and I eat lots of Bratwurst.

God bless the queen.

Life is like a box of choccolate.

- Token_index contains 500 000 words
- Length of sequence: T = 10

A1: (3, 10, 500 000), samples x sequence length x length word index (one-hot-encoded)

Question 1: what would be the dimension of the training data containing the three sentences above using One-hot-encoding?

Question 2: how many rows of zeros for each text?

A2: 0 for first text, 6 for second, 3 for last

Question 3: what happens if a word in the text is not contained in the token index?

Training texts

I am from Germany and I eat lots of Bratwurst.

God bless the queen.

Life is like a box of choccolate.

- Token_index contains 500 000 words
- Length of sequence: T = 10

A1: (3, 10, 500 000), samples x sequence length x length word index (one-hot-encoded)

Question 1: what would be the dimension of the training data containing the three sentences above using One-hot-encoding?

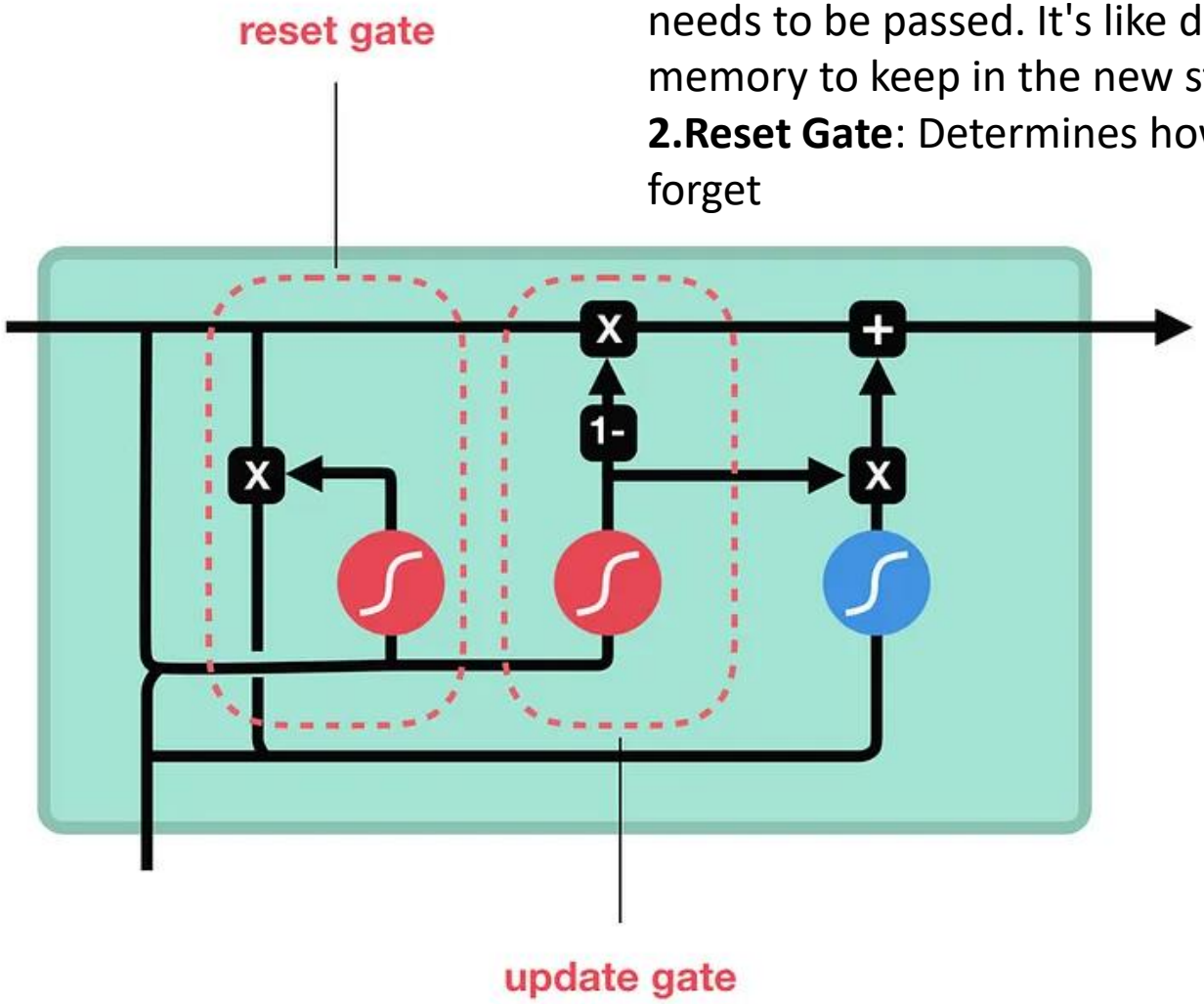
Question 2: how many rows of zeros for each text?

A2: 0 for first text, 6 for second, 3 for last

Question 3: what happens if a word in the text is not contained in the token index?

A3: The whole row contains zeros

Gated Recurrent Units - GRU



1.Update Gate: Determines how much of the past information needs to be passed. It's like deciding how much of the previous memory to keep in the new state.

2.Reset Gate: Determines how much of the past information to forget

Gated Recurrent Units - GRU

Check CNN-RNN_GRU.ipynb

CNN - RNN

1. Video Analysis and Processing
2. Time-Series Forecasting with Spatial Components
3. Audio and Speech Processing.
4. Medical Image Analysis
5. Sign Language Interpretation.

CNN - RNN

- Video Analysis and Processing

Application: Emotion Recognition in Videos

- Time-Series Forecasting with Spatial Components

Application: Environmental Event Detection

- Audio and Speech Processing

Application: Speech Emotion Recognition

- Medical Image Analysis

Application: Disease Progression Analysis

- Sign Language Interpretation

Application: Sign Language to Text Conversion

Training RNNs on time series data

We have *~420551* timesteps, each with 14 features.

Generator Parameters:

- lookback=1440 (minutes): We look back 1440 timesteps for each sample (24 (hours) * 60 (minutes) =1440).
- delay=144: We are predicting the value 144 (one day) where data was recorded every 10 minutes (24 (hours) * 60 (minutes)//10 =144).
- batch_size=128: Each batch contains 128 samples.
- min_index=0, max_index=200000: We're using the first first 200000 timesteps for training.
- The sequence length is10
- step=6: The period, in timesteps, at which we sample data. We sample every 6th timestep within our lookback window.
- Each sample in a batch consists of 240 time steps. This is calculated as lookback // step = 1440 // 6 = 240.

Batch Examples:

•First Batch:

- **Sample 1:**
  - Input: range(0, 1440, 6) out of 240 indices: [0, 6, 12, 18, 24, 30, 36, 42, 48, 54]
  - Target: Feature from timestep 1584
- **Sample 1:**
  - Input: range(1, 1441, 6) out of 240 indices: [1, 7, 13, 19, 25, 31, 37, 43, 49, 55]
  - Target: Feature from timestep 1585

- **Sample 3:**
  - Input: range(2, 1442, 6) Features from outof 240 indices: [2, 8, 14, 20, 26, 32, 38, 44, 50, 56]
  - Target: Feature from timestep 1586
- **…**
- **Sample 128:**
  - Input: range (127, 1567, 6) Features from timesteps [127, 133, 139, 145, 151, 157, 163, 169, 175, 181]
  - Target: Feature from timestep 1711.

Class Activity - "Exploring Applications of CNN-RNN Architectures"

- **Objective**: To explore and present the real-world applications of combined CNN-RNN models in various domains.
    - Make groups, focusing on one of the following topics:
        - Video Analysis and Processing
        - Time-Series Forecasting
        - Audio and Speech Processing
        - Medical Image Analysis
        - Sign Language Interpretation
    - Each group focusses discusses the
        - Application
        - How the CNN-RNN can be applied
        - Challenges
- **Learning Goals**:
    - Gain insights into practical uses of CNN-RNN models.
    - Encourage critical thinking about the advantages and limitations of deep learning applications.