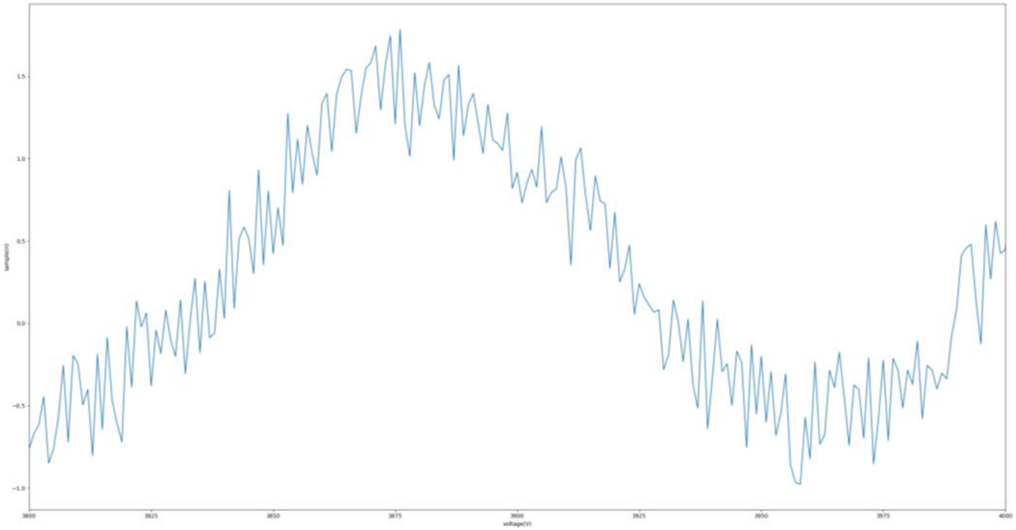
MADEIRA INTERNATIONAL WORKSHOP IN MACHINE LEARNING

2021

WHY RECURRENT NEURAL NETWORKS

WHY RECURRENT NEURAL NETWORKS

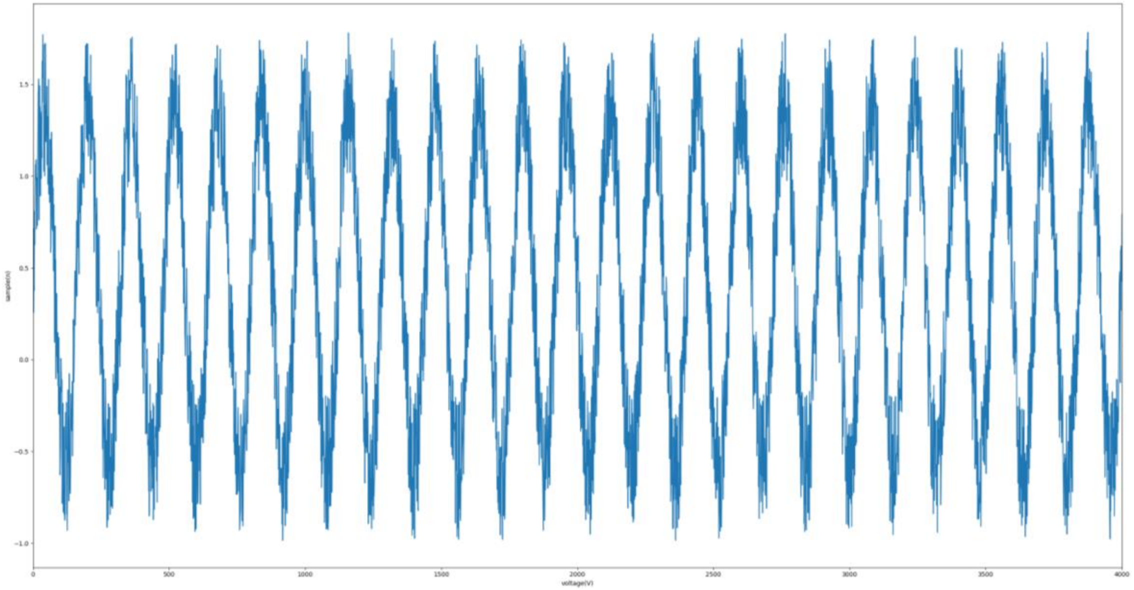
Where is the information?

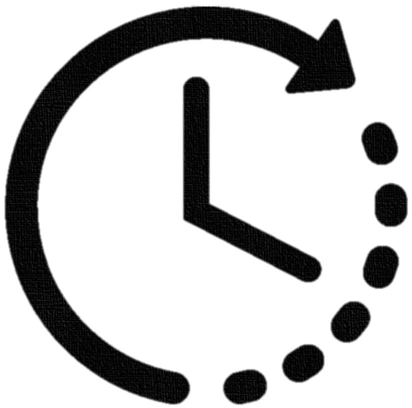


- Variation of the peeks amplitude?
- Frequency of the oscillations?
- Crossings of the trend line?

WHY RECURRENT NEURAL NETWORKS

Where is the information?





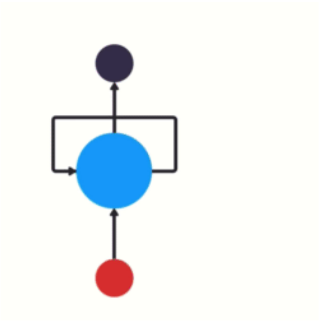
WHY RECURRENT NEURAL NETWORKS

Humans don’t start their thinking from scratch every second

- You don’t throw everything away and start thinking from scratch
- Your thoughts have persistence
- You make use of context and previous knowledge to understand what is coming next

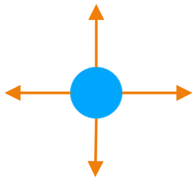
Recurrent Neural Networks (RNN) address this issue

- They are networks with loops, allowing information to persist



WHY RECURRENT NEURAL NETWORKS

Predict the direction of a moving ball:

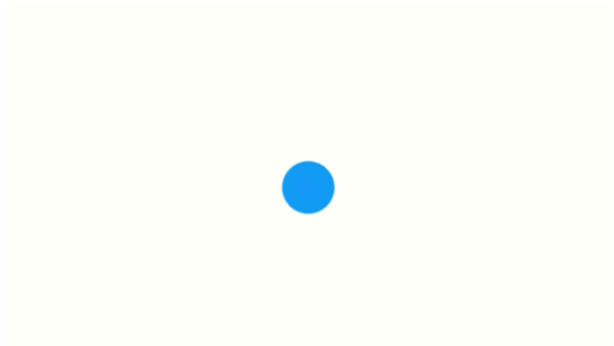


How would you do this by checking only the ball?

- Every guess is purely random without knowledge of where the ball has been
- You don’t have enough data to predict where it’s going

Record snapshots of the ball’s position in succession

- you will have enough information to make a better prediction



WHY RECURRENT NEURAL NETWORKS

RNN is good for processing a sequence data for predictions, but how?

- Make us of the sequence memory

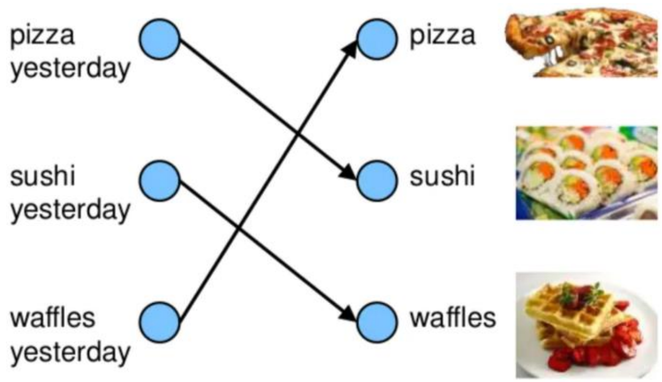
Try to say the alphabet in your head from A to Z

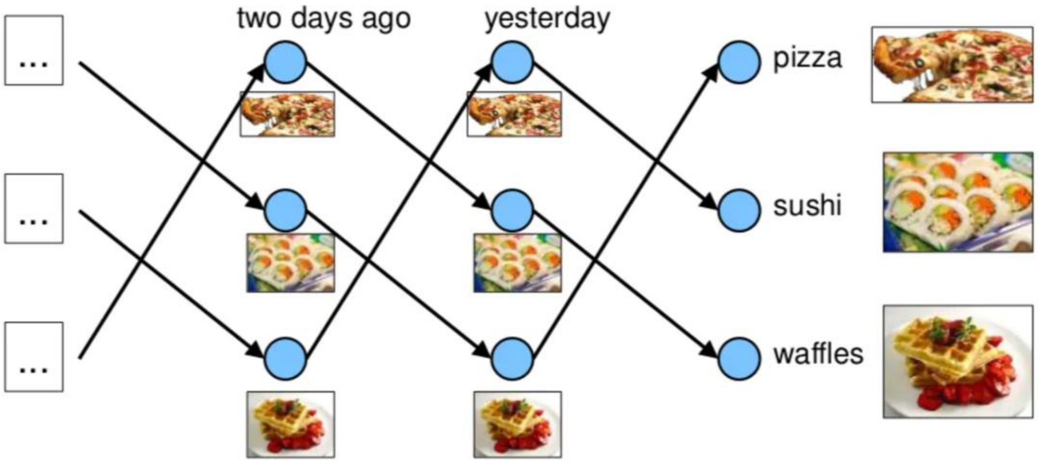
Now try to say from Z to A

- This can be difficult as you learn the alphabet as a sequence and your brain recognizes the sequential patterns

WHY RECURRENT NEURAL NETWORKS

Lunch forecast:





WHAT ARE RECURRENT NEURAL NETWORKS

WHAT ARE RECURRENT NEURAL NETWORKS

The aim of RNNs is to detect dependencies in sequential data

- Find correlations between different points within a sequence

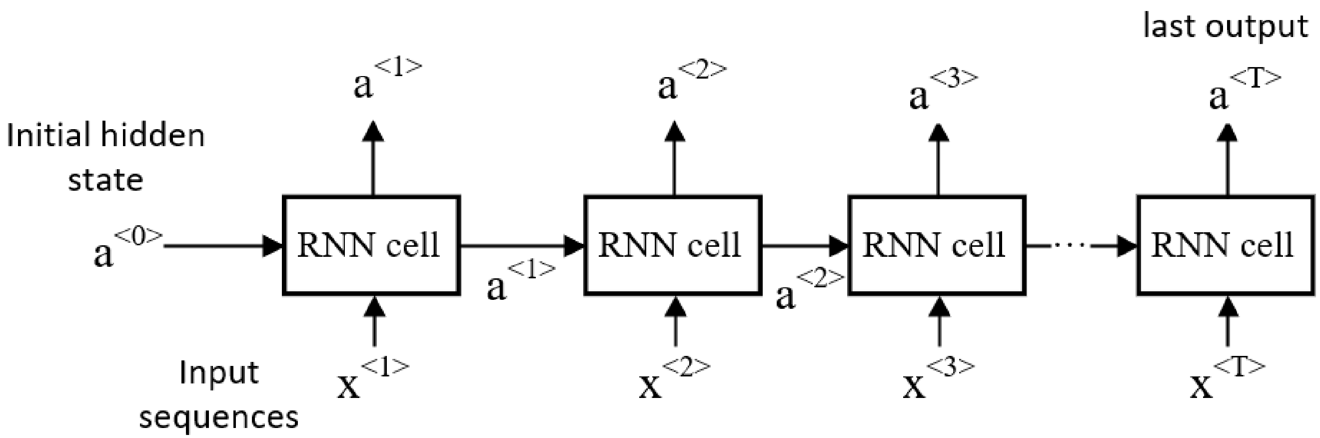
Two kinds of dependencies:

- Short-term dependencies are associated with the recent past
- Long-term dependencies are far away from each other in time

WHAT ARE RECURRENT NEURAL NETWORKS

Key terms:

- An input in a sequence is a time step
- The number of time steps define the sequence length
- Every time step in the sequence has associated a feature vector as input with the values we want to track



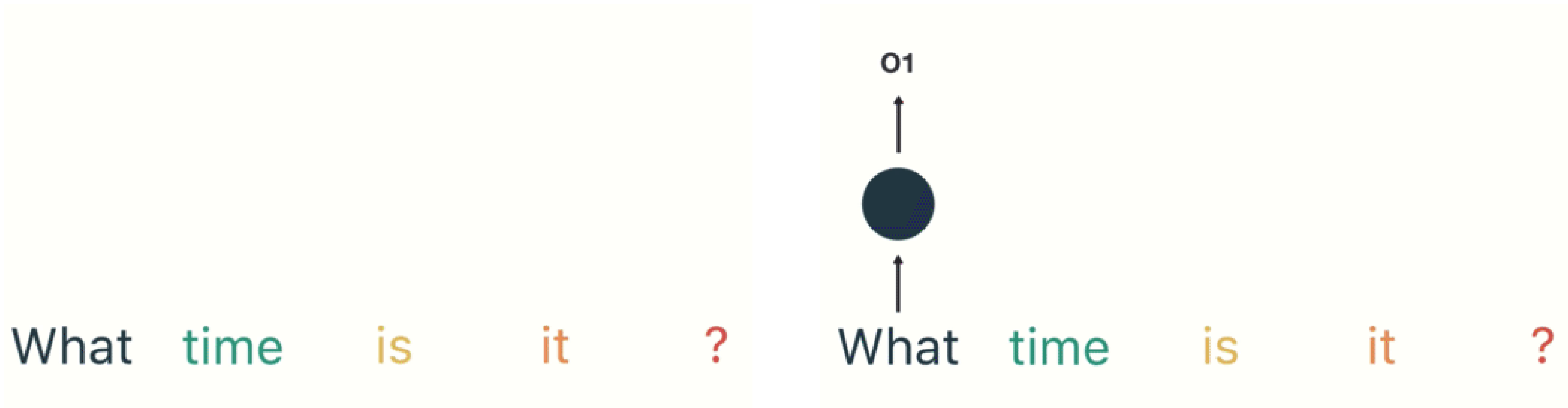
WHAT ARE RECURRENT NEURAL NETWORKS

Example: Classifying intents from users inputs

What time is it?

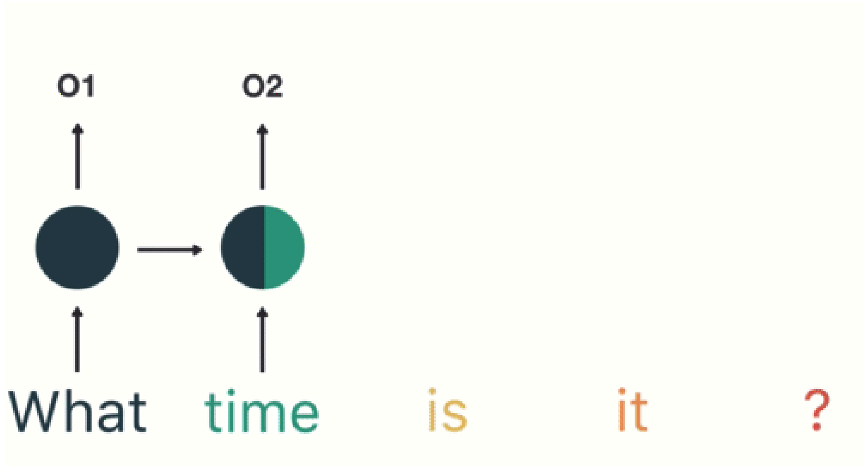
WHAT ARE RECURRENT NEURAL NETWORKS

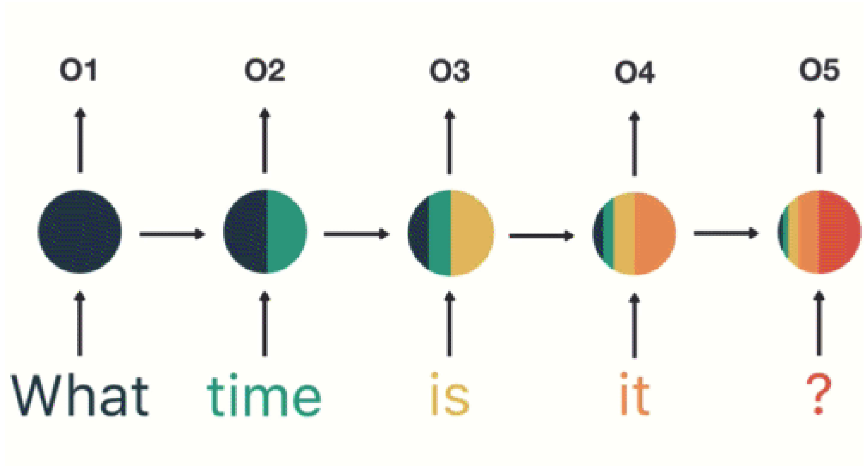
Example: Classifying intents from users inputs



WHAT ARE RECURRENT NEURAL NETWORKS

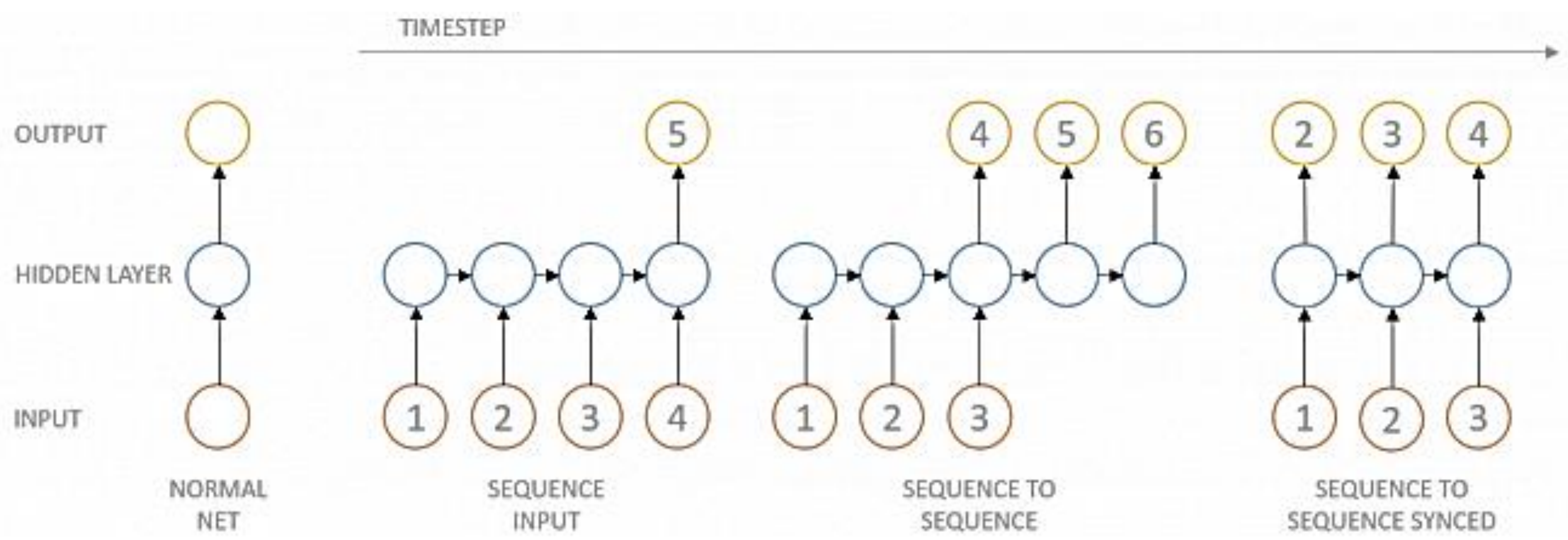
Example: Classifying intents from users inputs





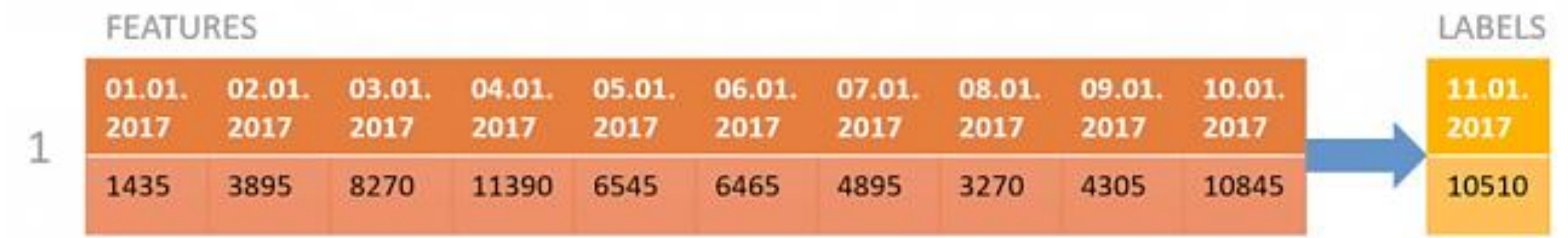
WHAT ARE RECURRENT NEURAL NETWORKS

Sequence prediction problems



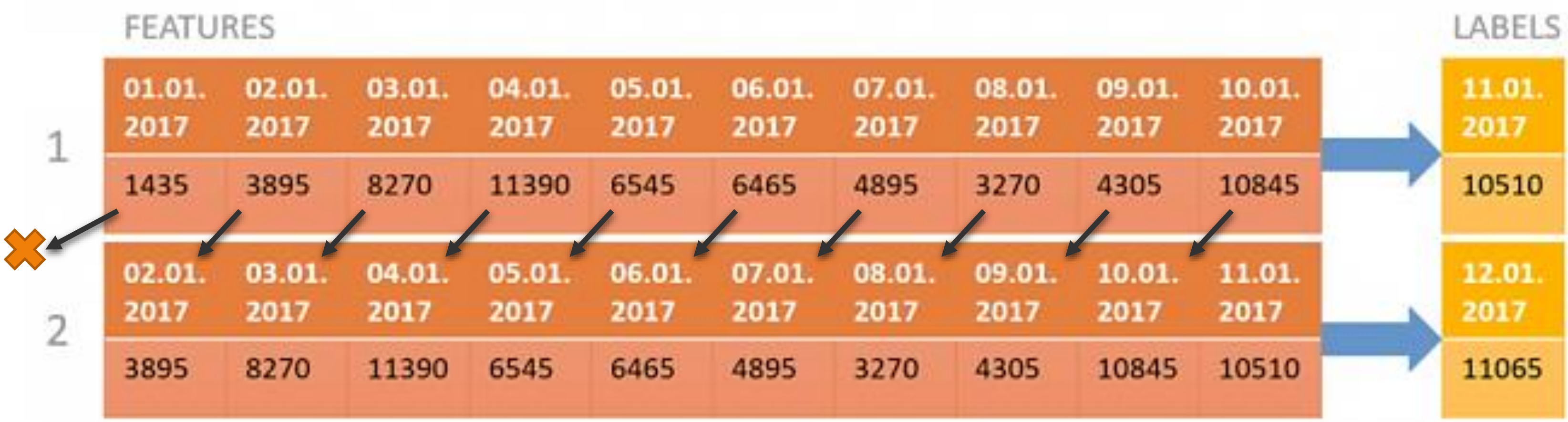
WHAT ARE RECURRENT NEURAL NETWORKS

Sequence creation



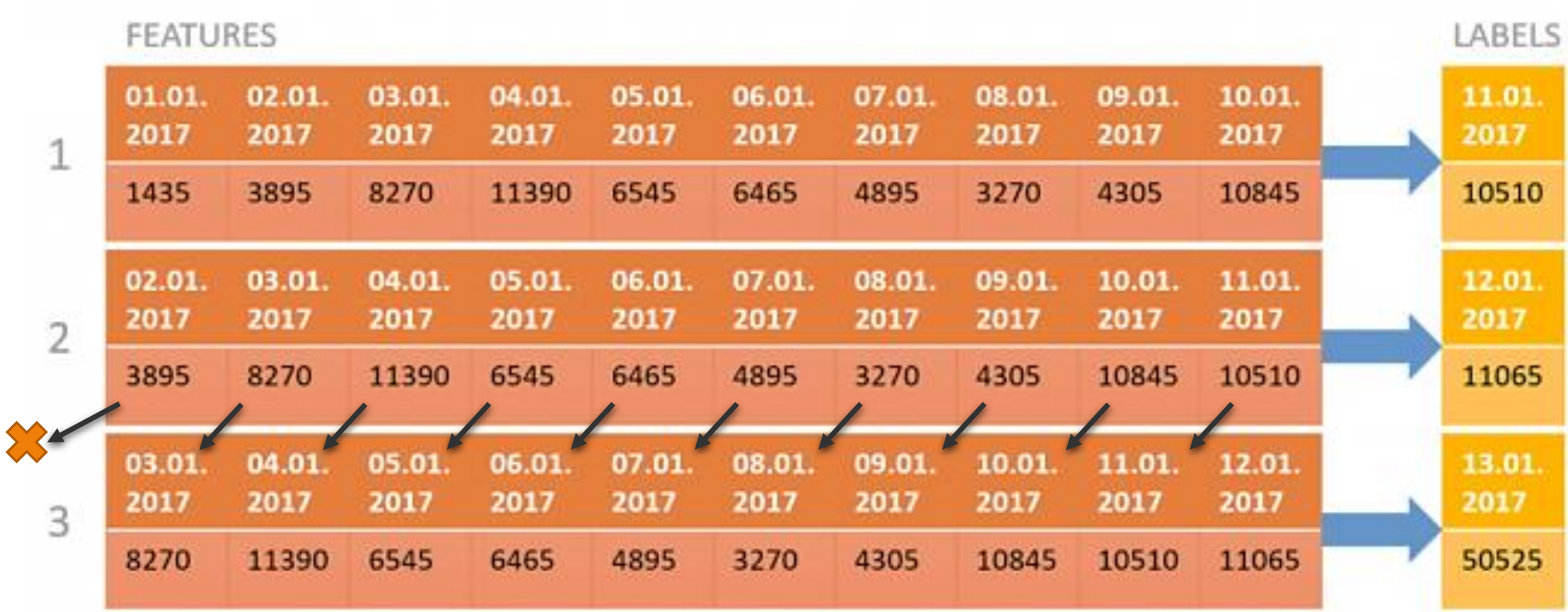
WHAT ARE RECURRENT NEURAL NETWORKS

Sequence creation



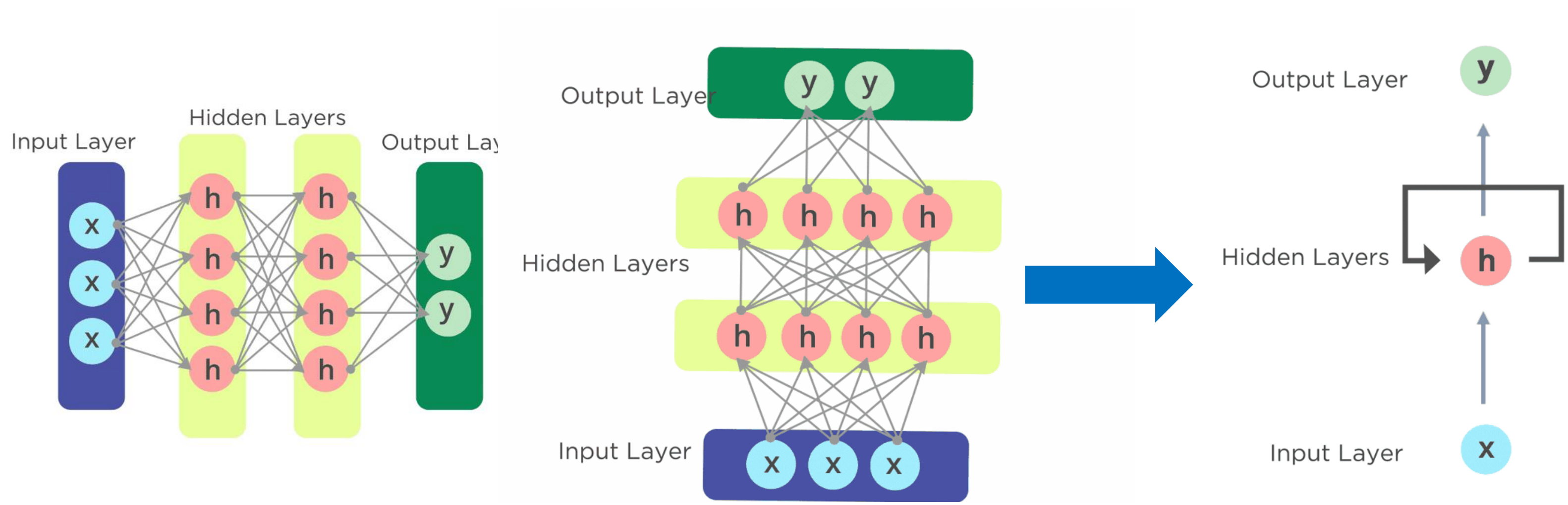
WHAT ARE RECURRENT NEURAL NETWORKS

Sequence creation



WHAT ARE RECURRENT NEURAL NETWORKS

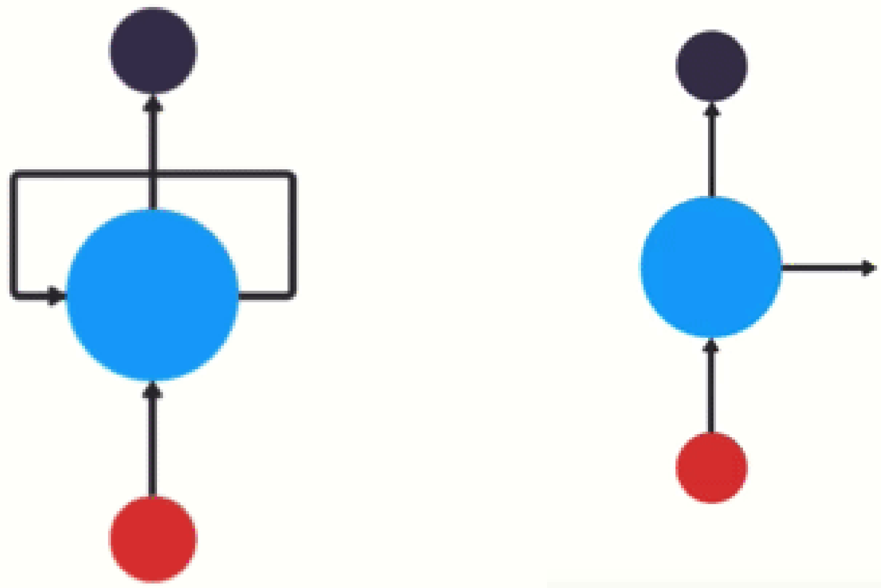
RNN structure:



Key idea: RNNs have an “internal state” that is updated as a sequence is processed

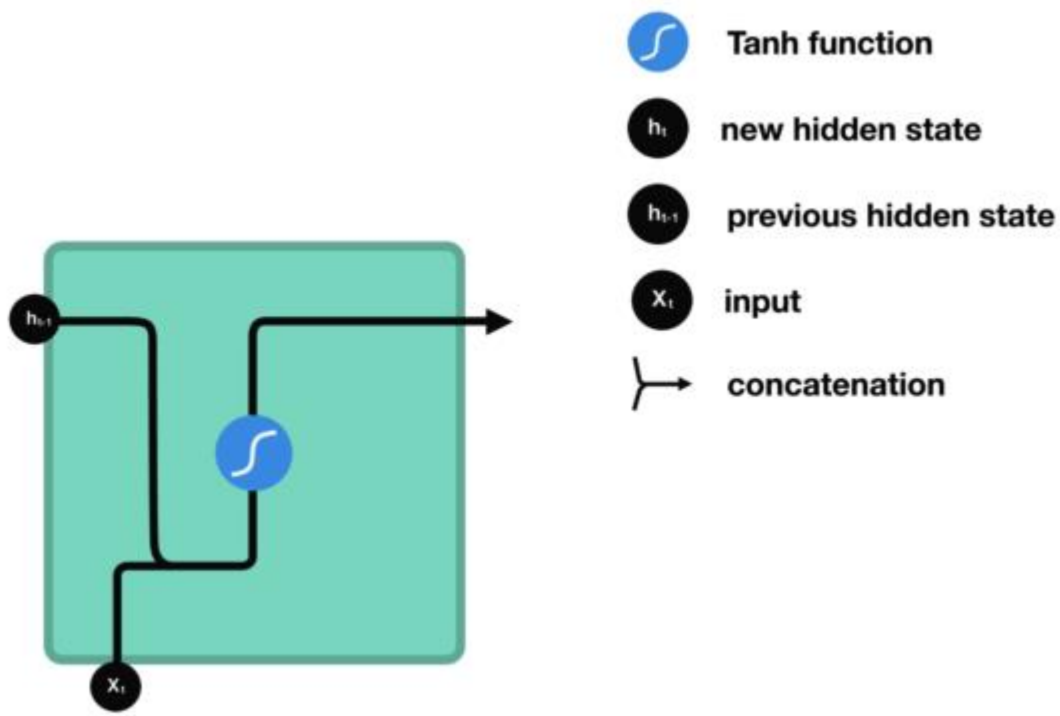
WHAT ARE RECURRENT NEURAL NETWORKS

RNN structure:



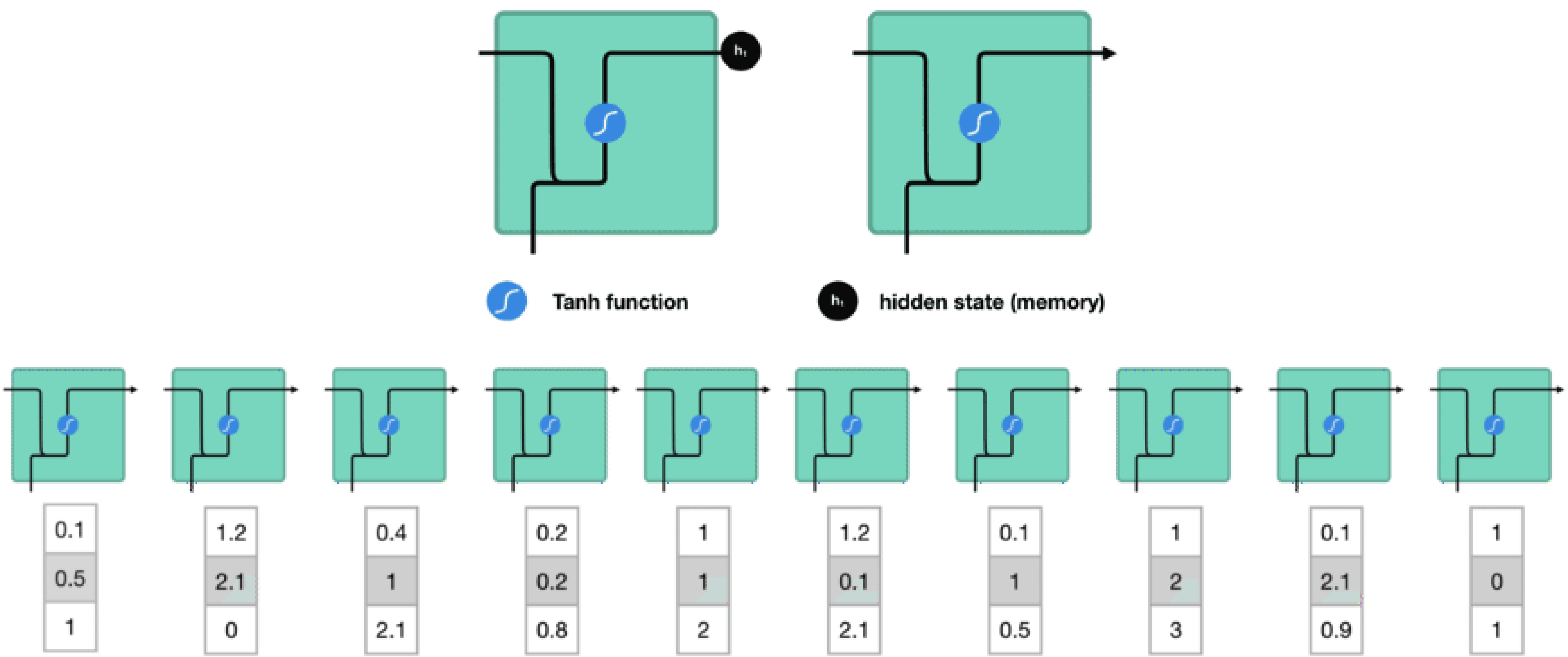
WHAT ARE RECURRENT NEURAL NETWORKS

RNN structure:



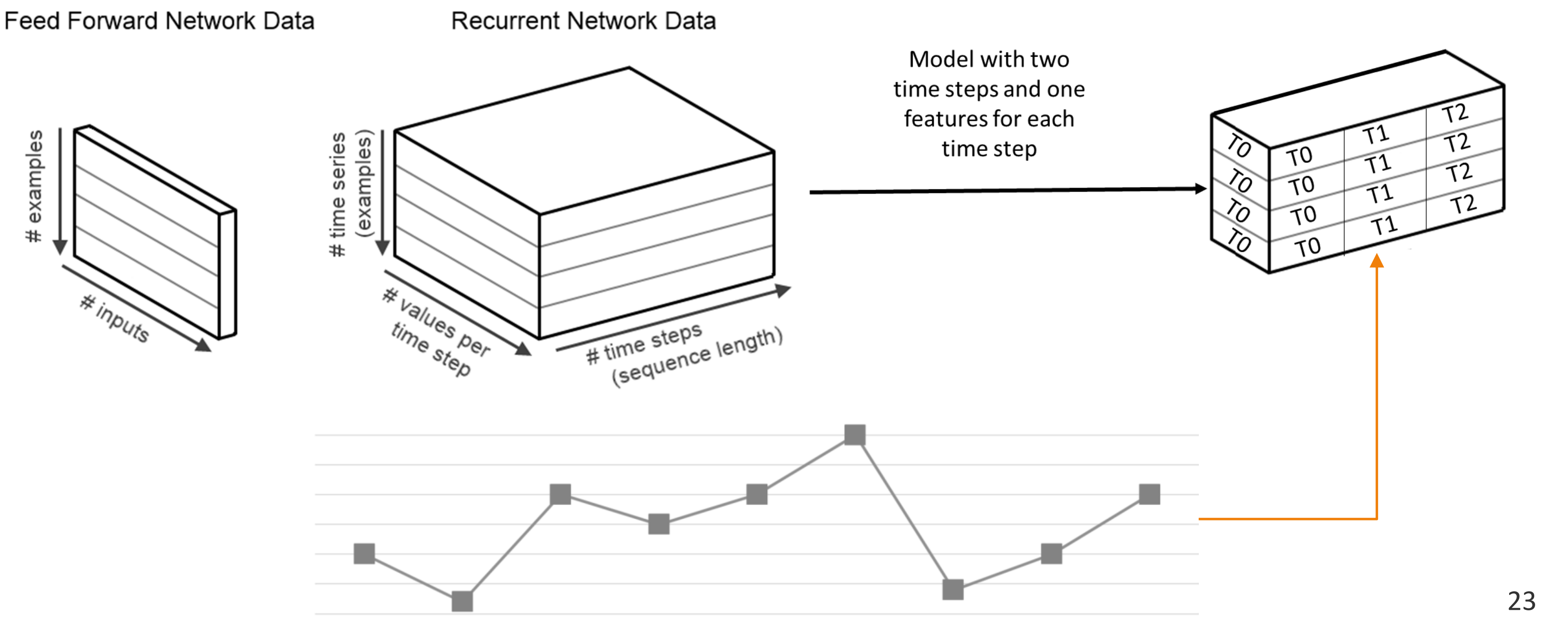
WHAT ARE RECURRENT NEURAL NETWORKS

RNN structure:



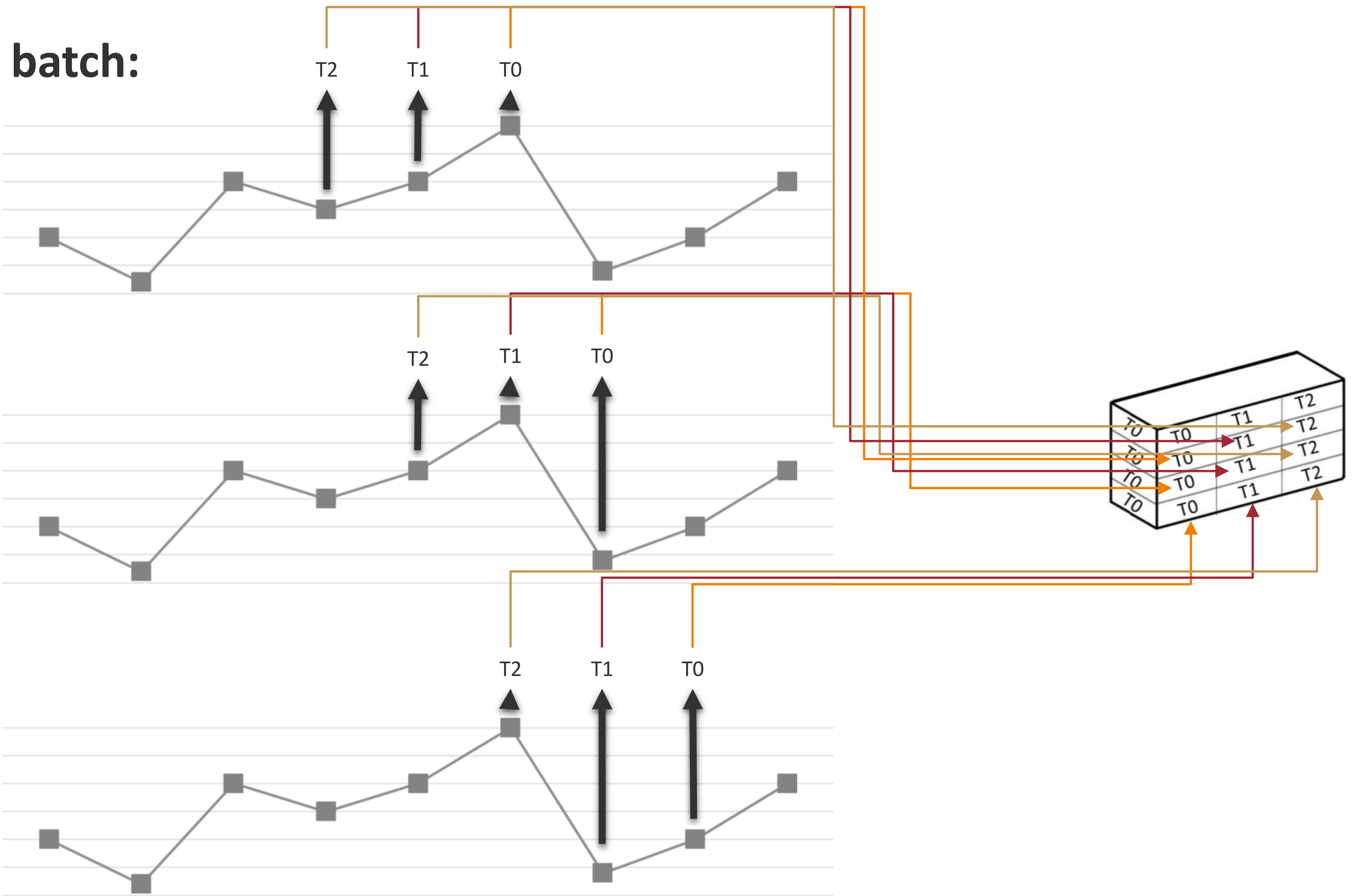
WHAT ARE RECURRENT NEURAL NETWORKS

RNN batch:



WHAT ARE RECURRENT NEURAL NETWORKS

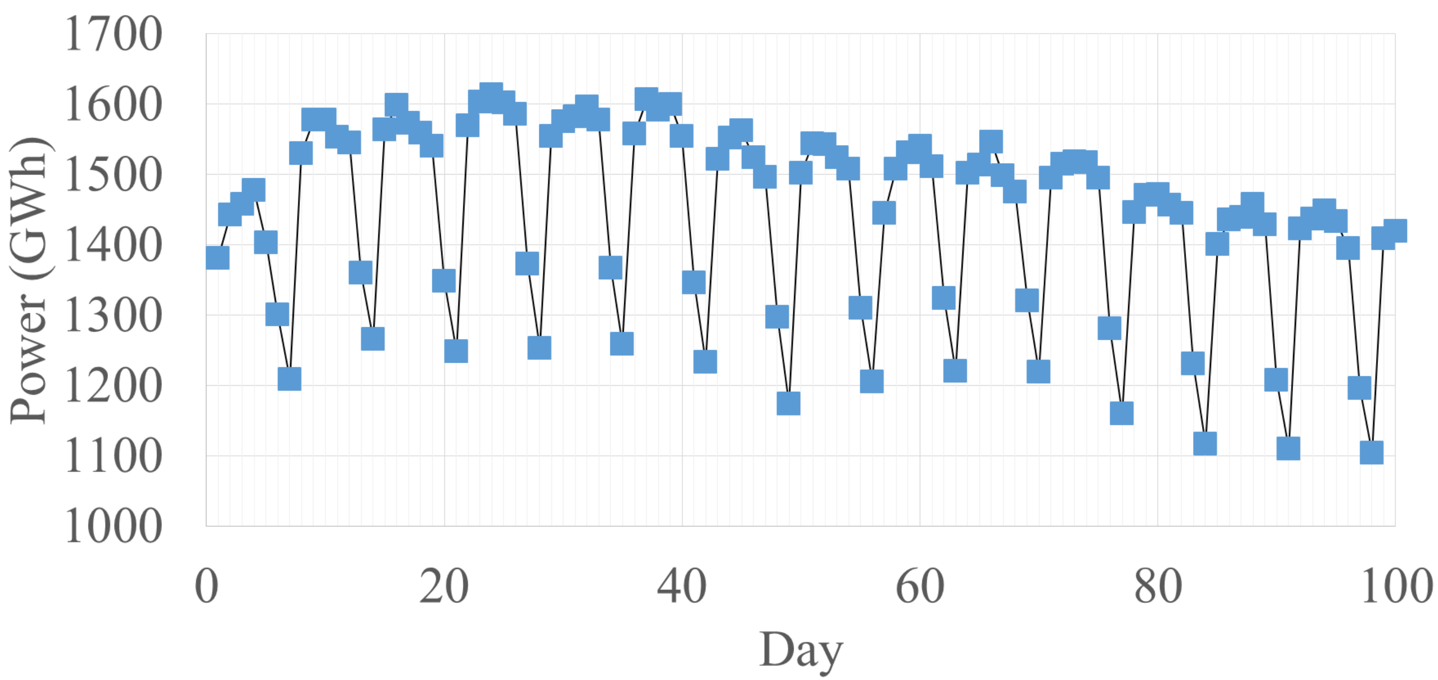
RNN batch:



WHAT ARE RECURRENT NEURAL NETWORKS

RNN Colab example: RNN – Energy

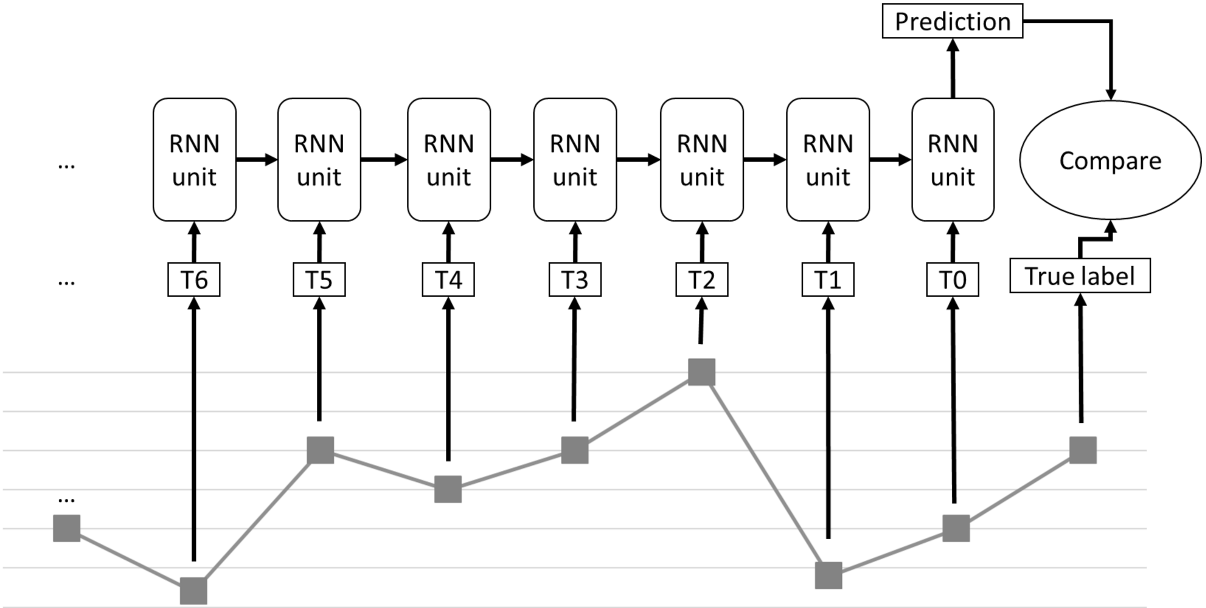
- Forecast the electricity consumption (in GWh) in Germany



WHAT ARE RECURRENT NEURAL NETWORKS

RNN Colab example: RNN – Energy

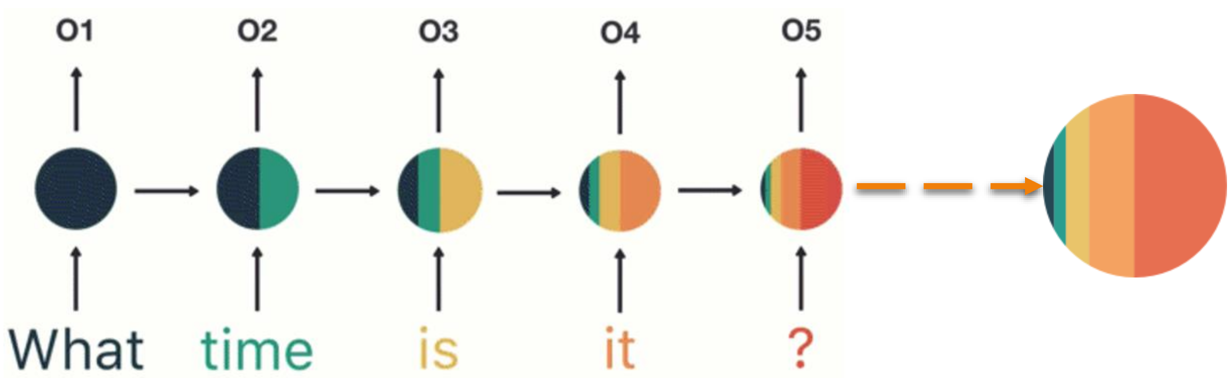
- Use 33 time steps (33 days)
- Estimate the power consumption for next day



WHEN THE RNN BREAKS

WHEN THE RNN BREAKS

As the RNN processes more steps, it has troubles retaining information from previous steps



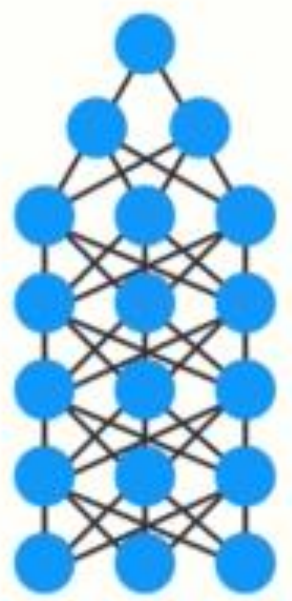
Final hidden state of the RNN

- Information from the words “what” and “time” is nearly extinct at the final time step
- This short-term memory problem is caused by the vanishing gradient during back-propagation

WHEN THE RNN BREAKS

Training with back-propagation:

- Forward pass to make a prediction
- Compares the prediction to the ground truth
- Estimate the error
- Uses the error value to do back propagation, calculating the gradients for each neuron in the network



WHEN THE RNN BREAKS

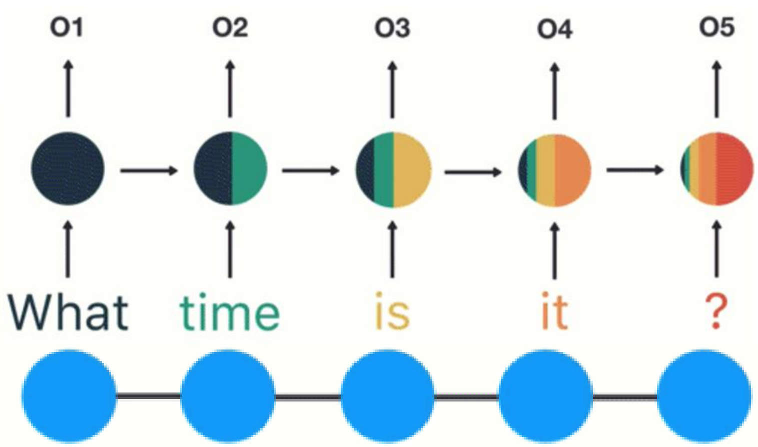
Vanishing gradient:

- Gradient allows the network to learn by adjusting the weights
- The higher the gradient, the higher the adjustments
- Each neuron estimates it’s gradient with respect to the gradient of the layer before it
- If the layers before have small adjustments, then adjustments to the current layer will be even smaller
- Gradients exponentially shrink as it back propagates

WHEN THE RNN BREAKS

Vanishing gradient:

- Think of each time step of the RNN as a layer
- Use back-propagation through time to train
- The gradient values will exponentially shrink as it propagates through each time step



LONG SHORT-TERM MEMORY

LONG SHORT-TERM MEMORY

Intuition:

- Read a review to decide if you want to buy a cereal
- Determine if someone thought it was good or bad

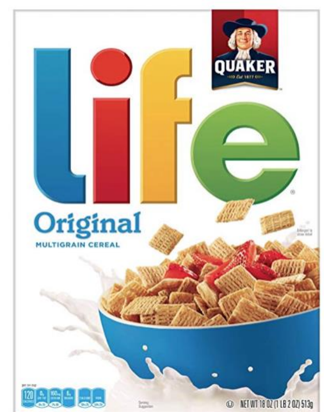
Customers Review 2,491

Thanos

September 2018

Verified Purchase

Amazing! This box of cereal gave me a perfectly balanced breakfast, as all things should be. I only ate half of it but will definitely be buying again!



A Box of Cereal
$3.99

LONG SHORT-TERM MEMORY

Intuition:

- Your brain will only remember the important keywords such as“amazing” and “perfectly balanced breakfast”
- The irrelevant words will be ignored

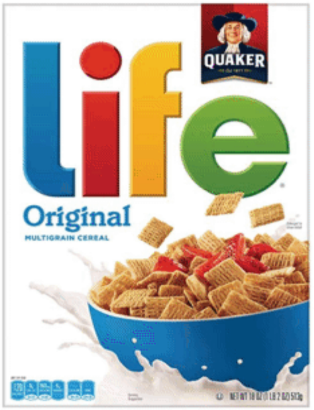
Customers Review 2,491

Thanos

September 2018

Verified Purchase

Amazing! This box of cereal gave me a perfectly balanced breakfast, as all things should be. I only ate half of it but will definitely be buying again!



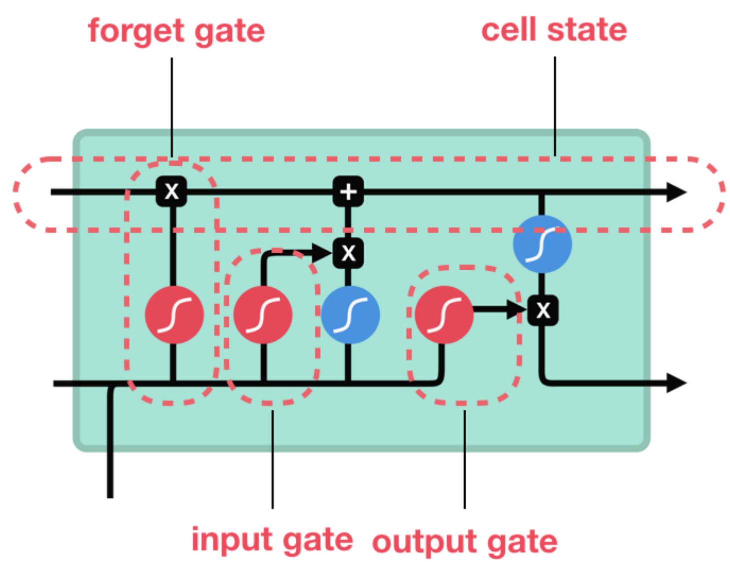
A Box of Cereal

$3.99

LONG SHORT-TERM MEMORY

How to address the problem:

- The Long Short-Term Memory (LSTM) keeps only relevant information to make predictions
- Use gate mechanism to learn long-term dependencies
- These gates are trained to identify what information should be added or removed to the hidden state



LONG SHORT-TERM MEMORY

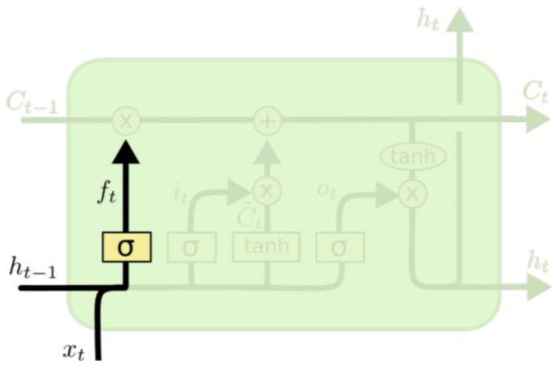
How LSTM works:

- The LSTM is a combination of gates and a cell state
- The cell state acts as the network's memory and transfers information across the sequence chain
- Information from all time steps can reach the output cell, reducing the short-term memory effects

LONG SHORT-TERM MEMORY

How LSTM works:

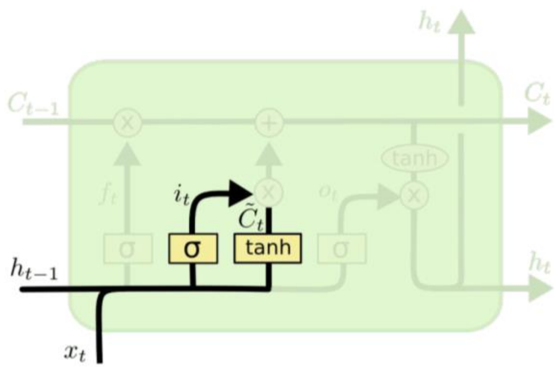
- Forget gate decides what information should be kept or thrown away
- The information from the previous hidden state and current input is transformed by the sigmoid (0 to 1)
- Values closer to 1 means to keep while closer to 0 is to forget



LONG SHORT-TERM MEMORY

How LSTM works:

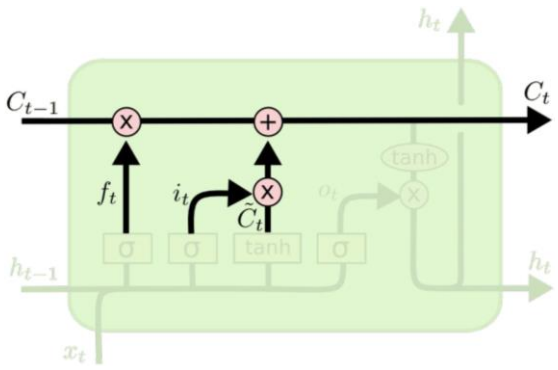
- Input gate allows to update the cell state, according to the output of the sigmoid function (0 to 1)
- If 0 then is irrelevant (skipping the time step) while 1 is very important
- The information from the previous hidden state and current input is multiplied by the sigmoid output to update the cell state



LONG SHORT-TERM MEMORY

How LSTM works:

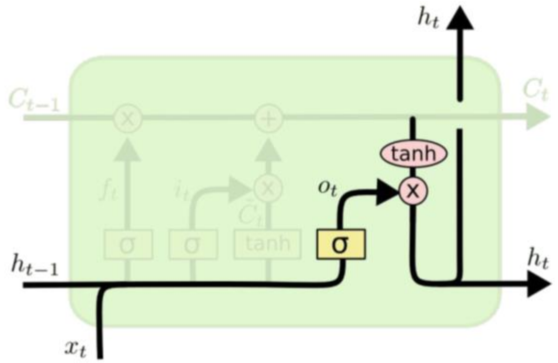
- The previous cell state is multiplied by the forget gate's output
- Then the input gate's output is added, producing the new cell state



LONG SHORT-TERM MEMORY

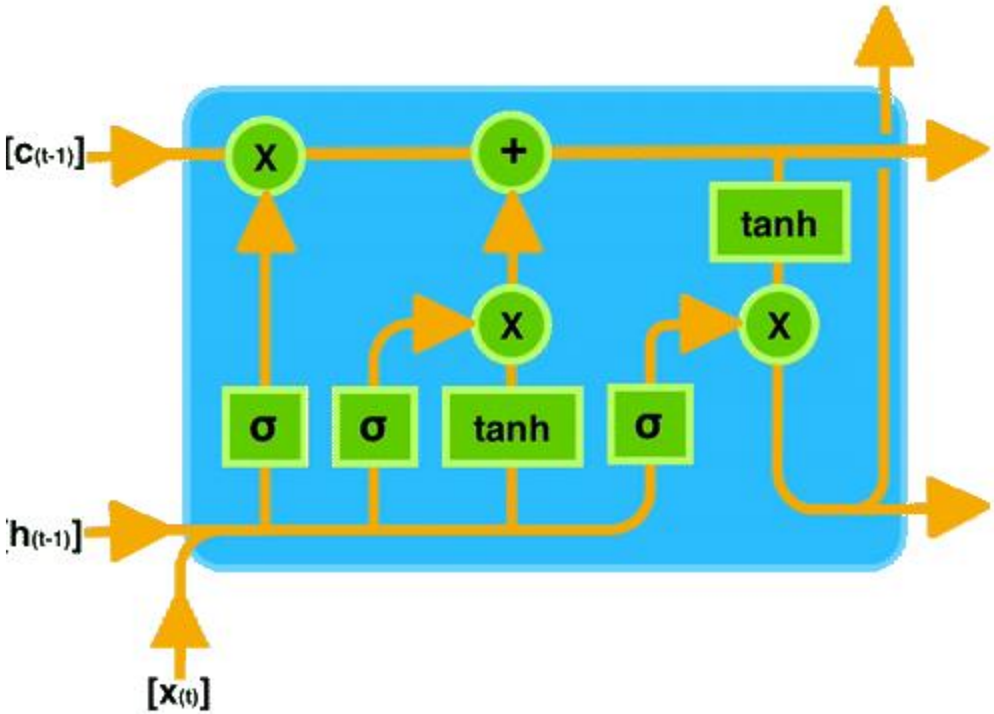
How LSTM works:

- The output gate selected the relevant information to be used as the next hidden.
- This decision is taken according to the output of the sigmoid function
- The output is the hidden state of the current cell



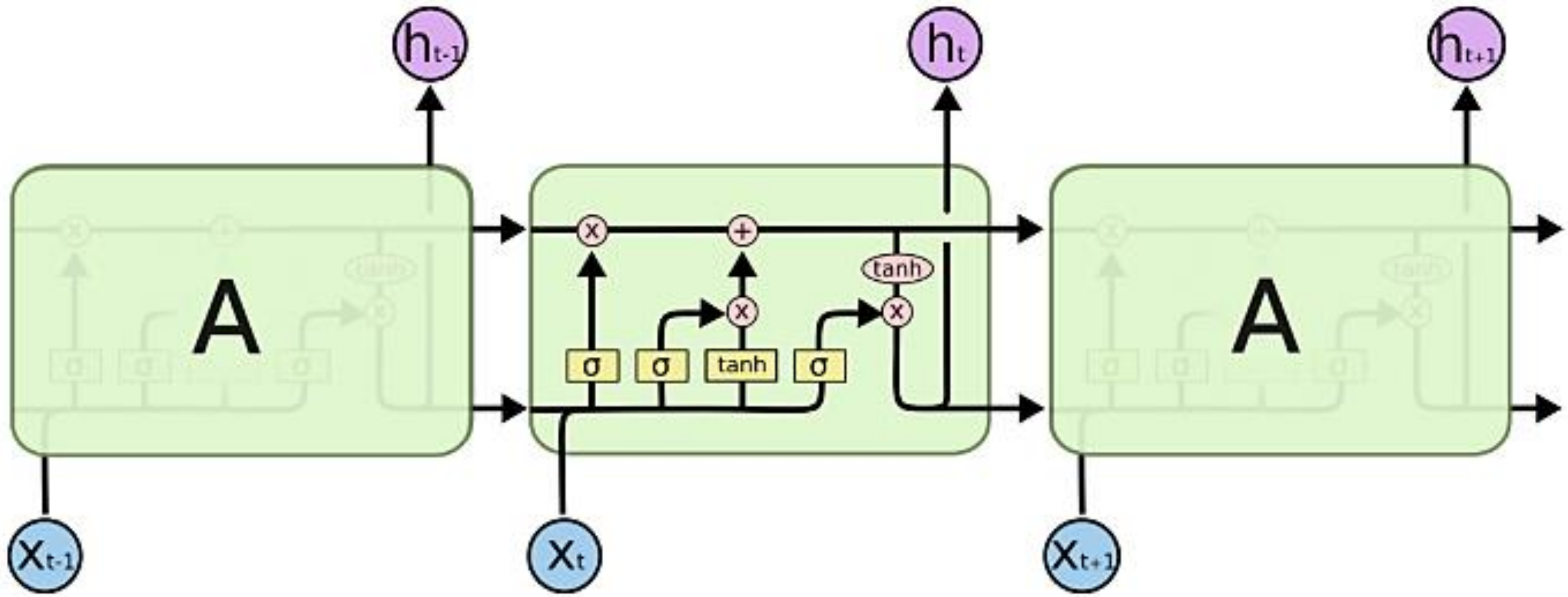
LONG SHORT-TERM MEMORY

How LSTM works:



LONG SHORT-TERM MEMORY

How LSTM works:



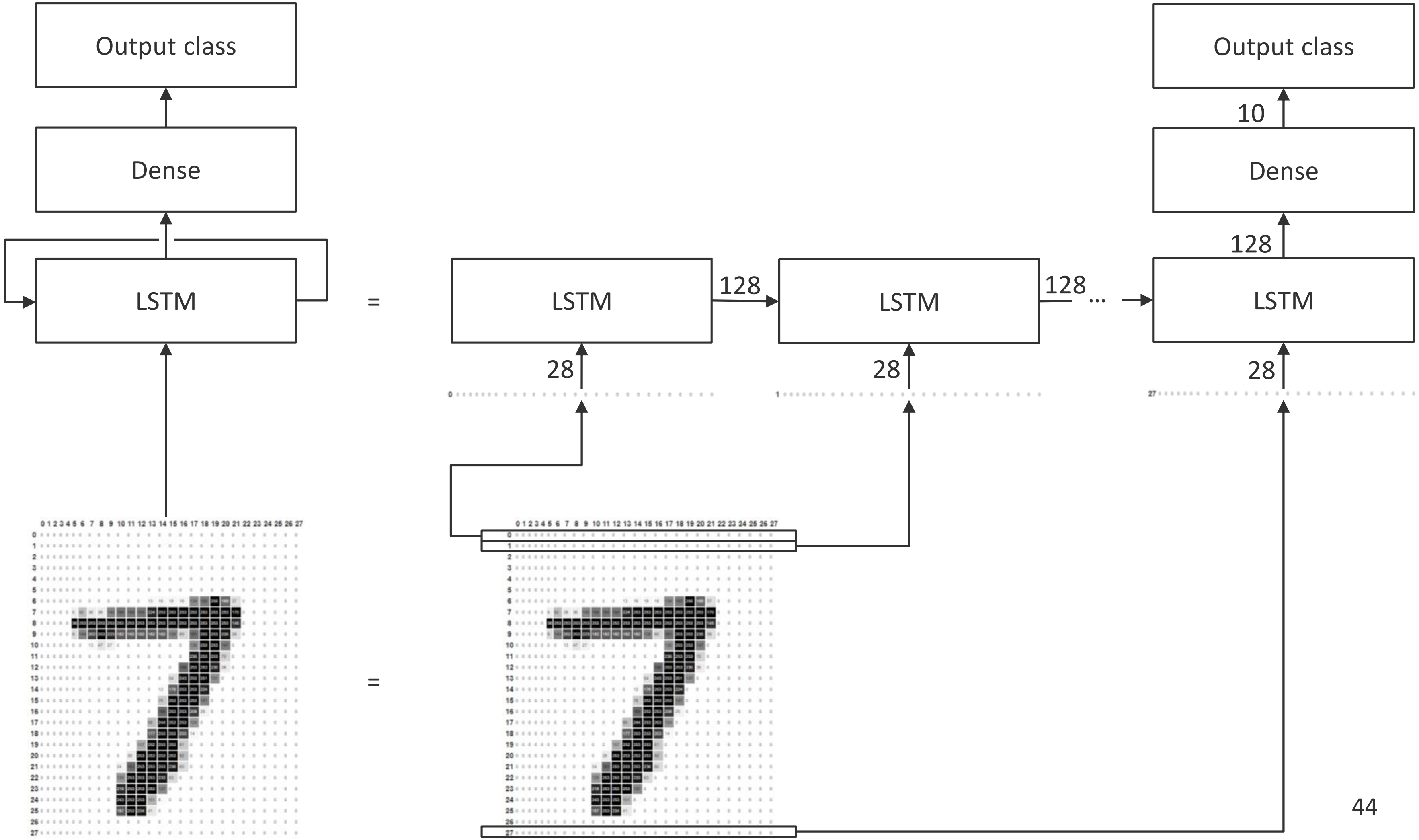
LONG SHORT-TERM MEMORY

LSTM Colab example: LSTM – MNIST

- Handwritten dataset
- 70000 images
- All are 28x28
- 784 pixels in total

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	13	18	18	18	134	150	255	169	37	0	0	0	0	0	0
7	0	0	0	0	0	5	92	36	36	140	154	154	154	224	253	253	253	253	253	253	253	170	0	0	0	0	0	0
8	0	0	0	0	0	36	253	253	253	253	253	253	253	253	253	253	253	253	253	253	253	145	0	0	0	0	0	0
9	0	0	0	0	0	9	144	202	253	223	182	182	182	182	182	138	65	161	253	253	235	38	0	0	0	0	0	0
10	0	0	0	0	0	0	0	13	47	27	0	0	0	0	0	0	0	130	253	253	147	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	235	253	253	70	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166	253	253	230	36	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	54	243	253	251	131	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	176	253	253	224	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	78	253	253	253	141	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	168	253	253	208	25	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	95	244	253	253	124	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	177	253	253	203	14	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	2	107	252	253	253	81	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	38	253	253	253	253	92	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	24	163	253	253	253	236	60	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	125	253	253	253	233	63	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	218	253	253	253	137	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	243	253	253	101	8	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	187	253	234	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LONG SHORT-TERM MEMORY



LONG SHORT-TERM MEMORY

LSTM advantages:

- Are usually the most accurate among the RNN
- As the best when the problem involves longer sequences

LSTM issues:

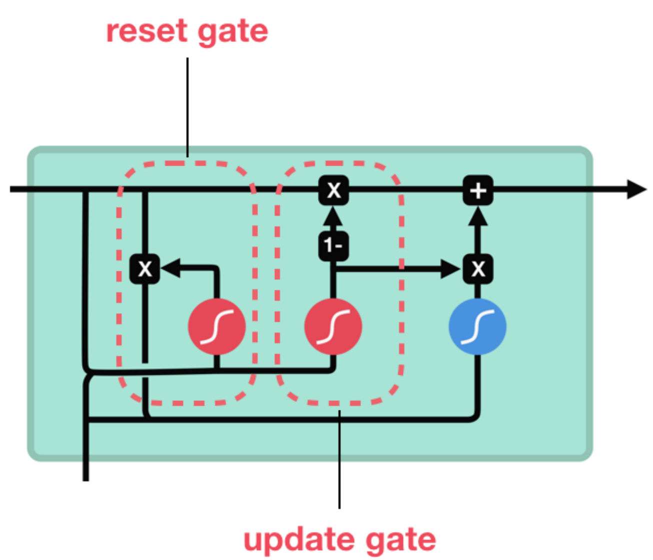
- Are slow to train
- As the complexity of the problem increases, it also increases the amount of data required to properly train
- Requires hardware with large memory

GATED RECURRENT UNITS

GATED RECURRENT UNITS

Gated Recurrent Units (GRU) as alternatives to the LSTM:

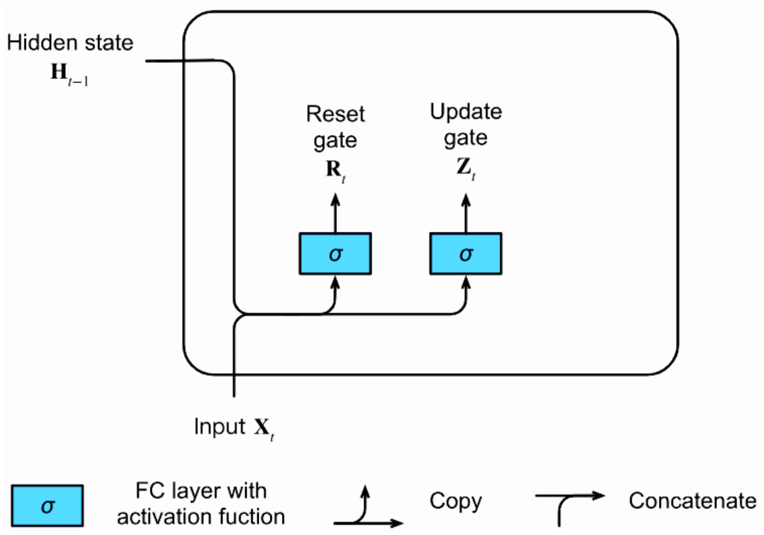
- Are less complex
  - Use less training parameters
  - Use less memory
  - Execute faster and train faster
- Useful when the accuracy is not very critical or when the sequences are short



GATED RECURRENT UNITS

How GRU works:

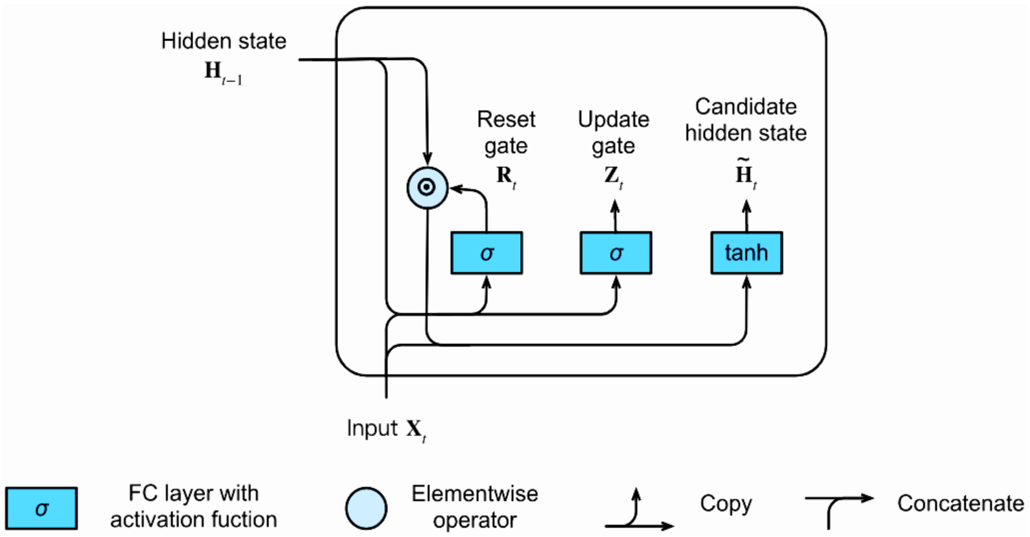
- The update and reset gates are a combination of LSTM forget and input gates
- Reset gate control how much of the previous state will be remembered
- Update gate control how much of the new state is copied from the previous state



GATED RECURRENT UNITS

How GRU works:

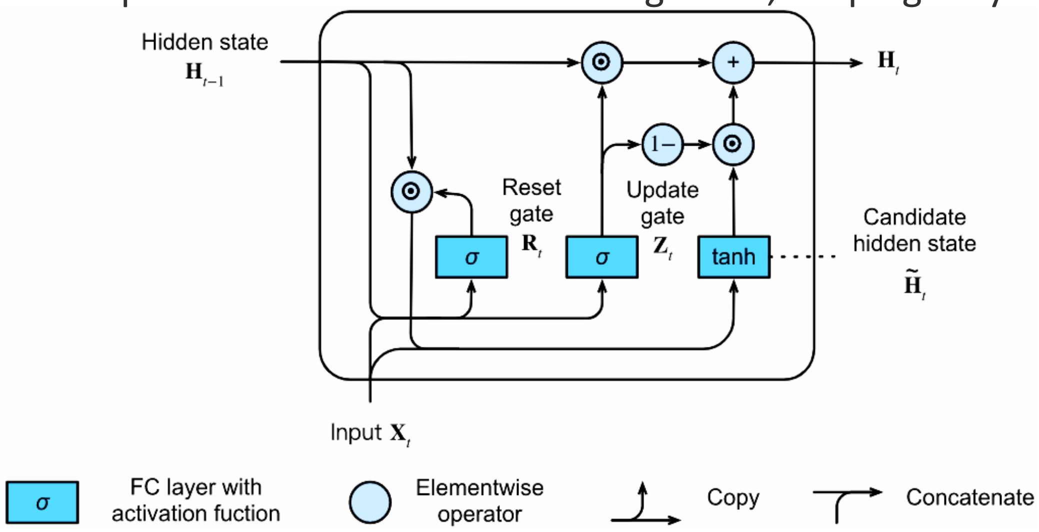
- The new candidate tor the hidden state is determined by the reset gate
  - For the extreme cases
    - When the reset gate is 1 then we have the standard RNN and when it is 0 we have the standard fully connected layer



GATED RECURRENT UNITS

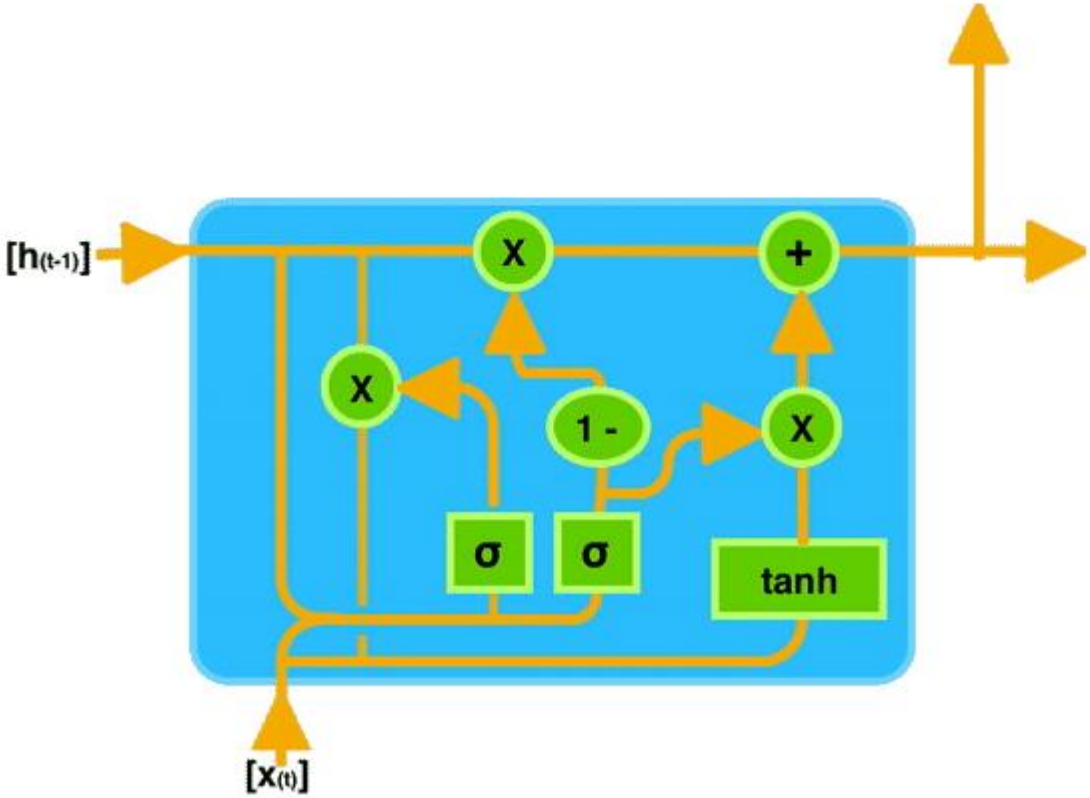
How GRU works:

- The update gate defines how much should the hidden state incorporate the contributions from the candidate hidden state
  - For the extreme cases
    - when the update gate is 1 then all new contributions are ignored, skipping the current time step, and when it is 0 the previous contributions are all ignored, keeping only the new information



GATED RECURRENT UNITS

How GRU works:

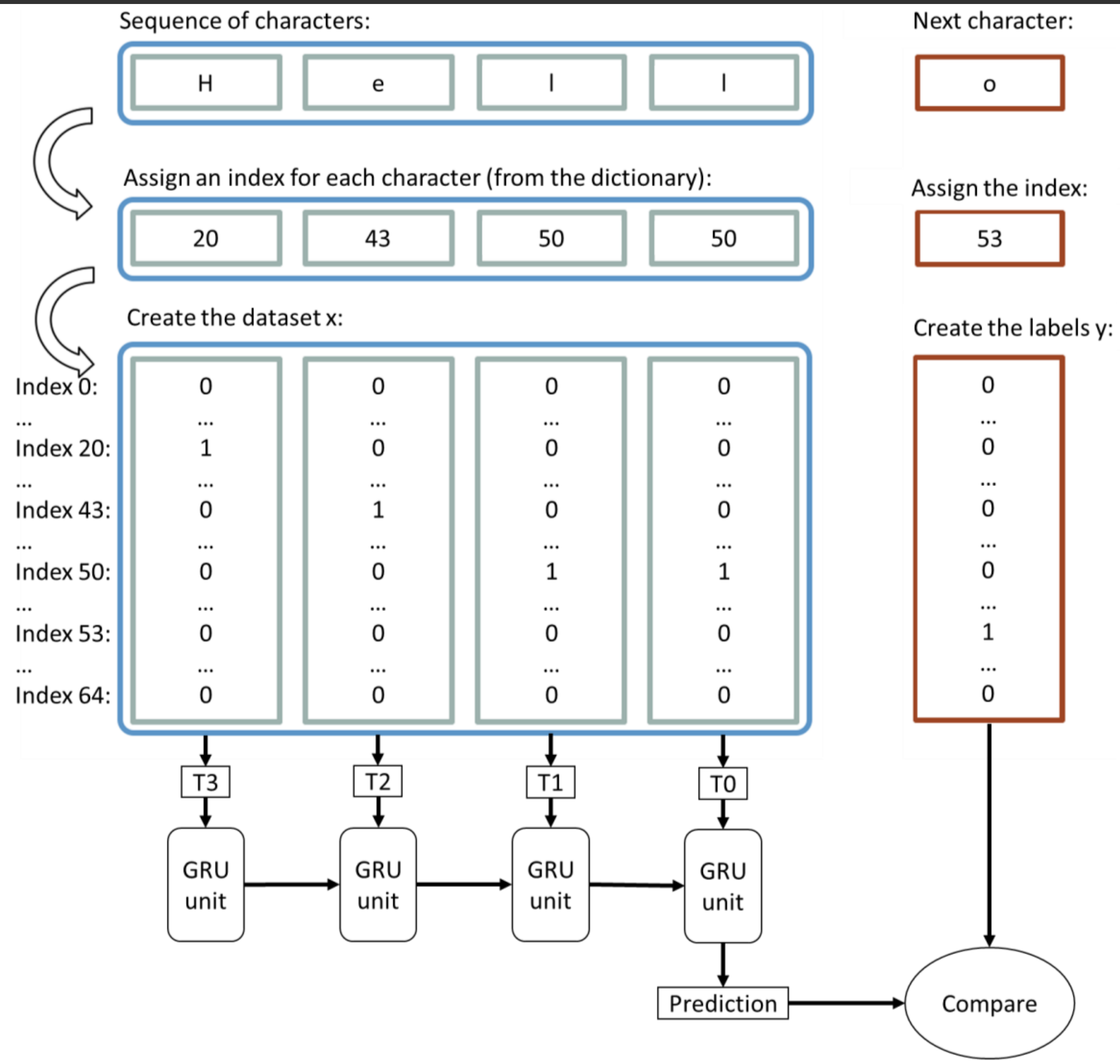


GATED RECURRENT UNITS

GRU Colab example: GRU – Words

- All of Shakespeare's plays, characters, lines, and acts
- Total of 1115394 characters where 65 are different
- All unique characters: \n, , !, $, &, ', ,, -, ., 3, :, ;, ?, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z

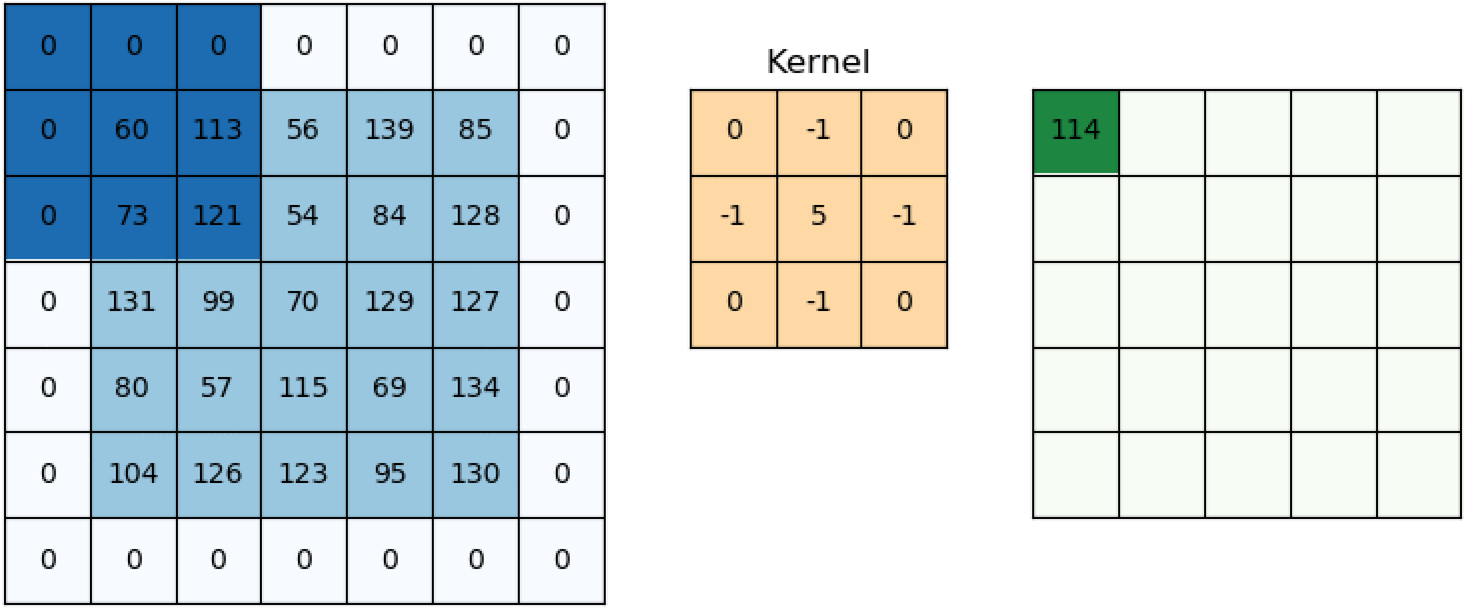
GATED RECURRENT UNITS



CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY

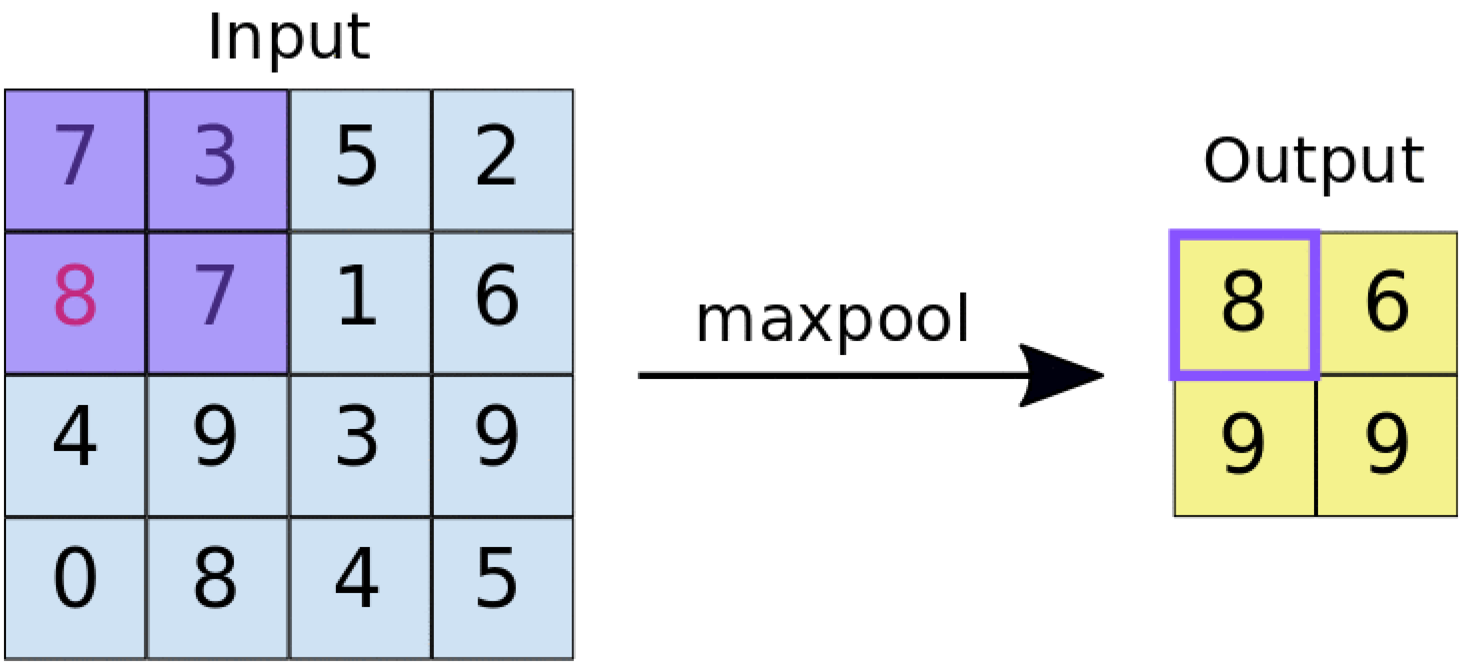
CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY

How Convolutional Neural Networks (CNN) works:



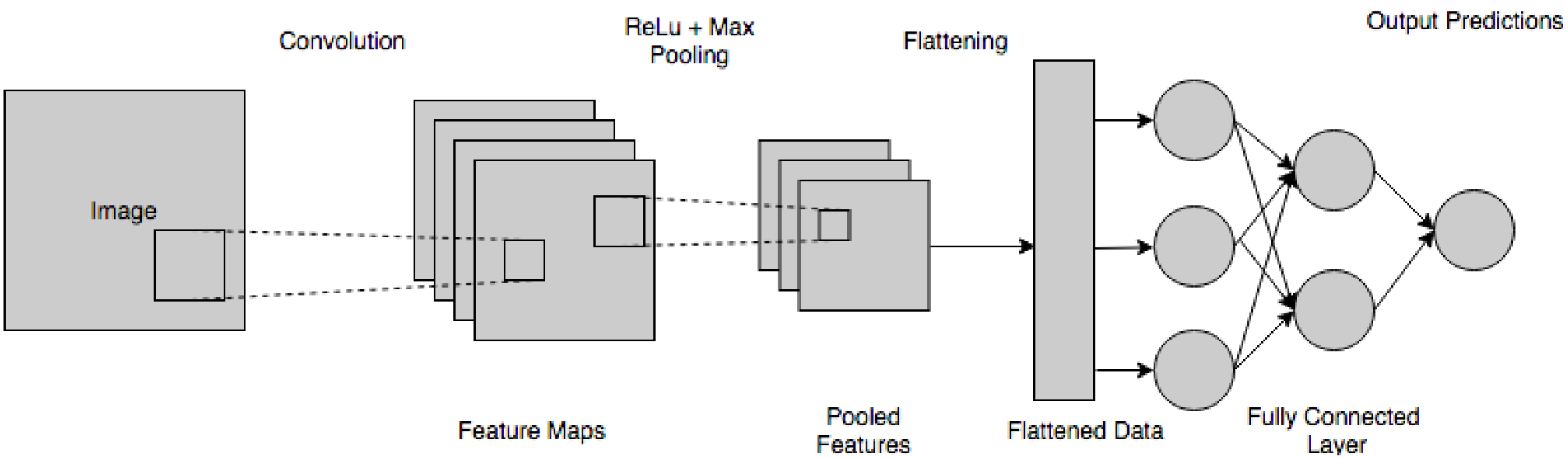
CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY

How Convolutional Neural Networks (CNN) works:



CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY

How Convolutional Neural Networks (CNN) works:



CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY

Advantages:

- Take advantage of local spatial coherence in the input to have a low number of parameters
- Excellent for feature extraction

CNN

Disadvantages:

- Cannot handle sequential data
- Considers only the current input
- Cannot memorize patterns from previous inputs

CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY

Advantages:

- Can process sequential data
- Can learn long dependencies in the data

LSTM

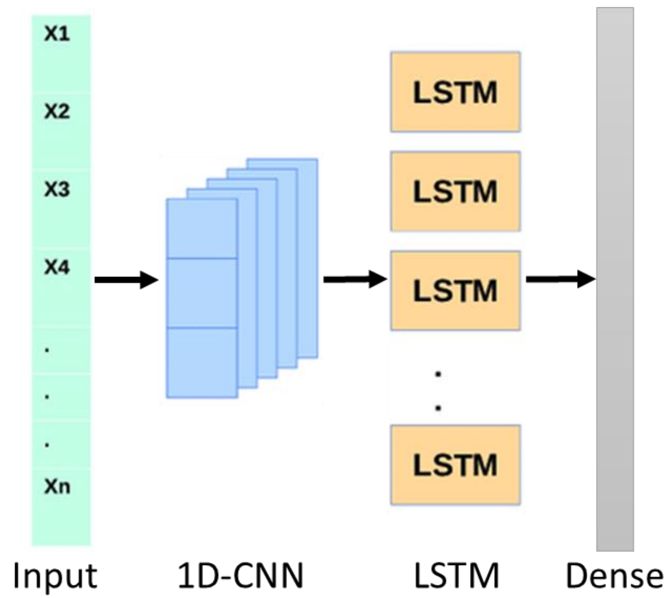
Disadvantages:

- Cannot take advantage of spatial coherence for feature extraction
- Not so good extracting features

CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY

Combine the CNN and the LSTM:

- Use CNN for feature extraction
- Use LSTM to find the patterns in the time series
- Use fully connected (dense) layer to perform the classification/regression



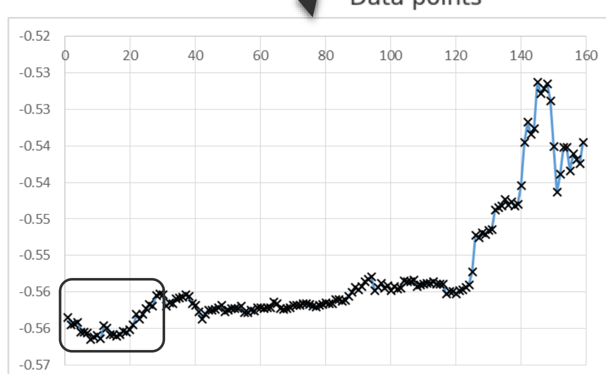
CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY

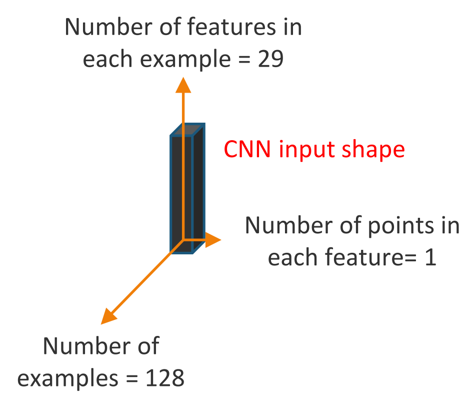
CNN-LSTM Colab example: CNN-LSTM – Bitcoin

- Forecast the next day Bitcoin value based on the values from previous days

CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY

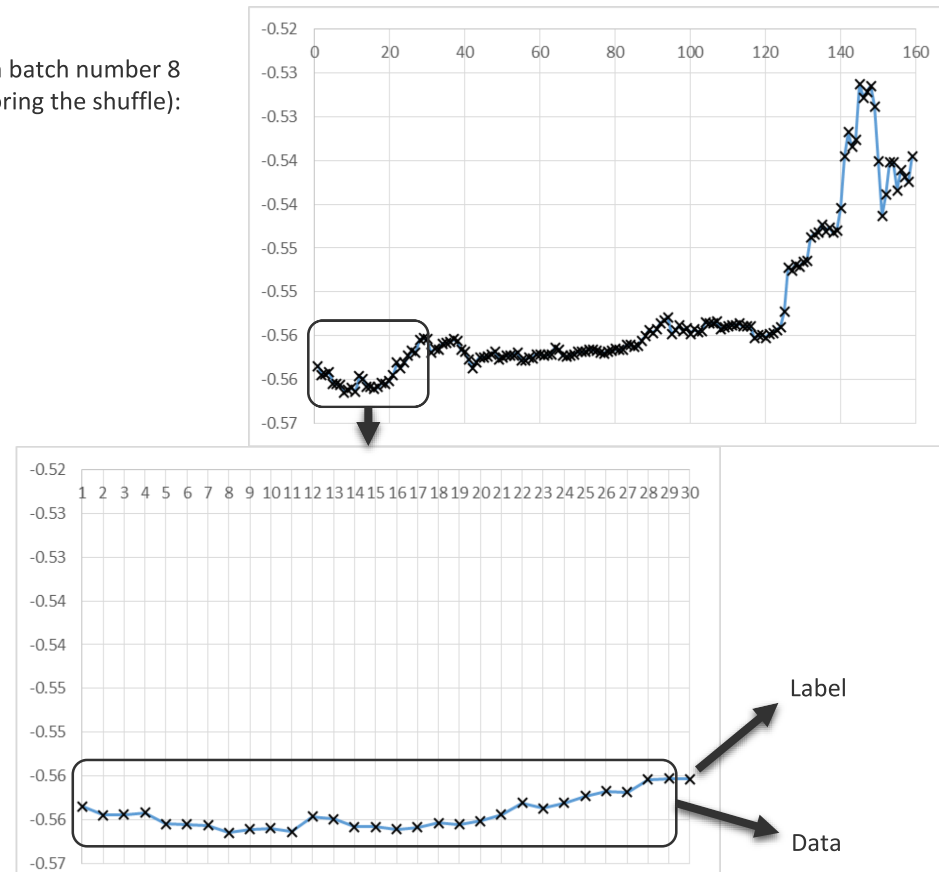
Train batch number 8
(ignoring the shuffle):





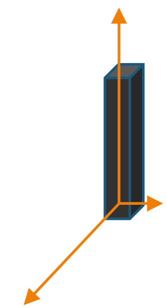
CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY

Train batch number 8
(ignoring the shuffle):



Train example number
1:

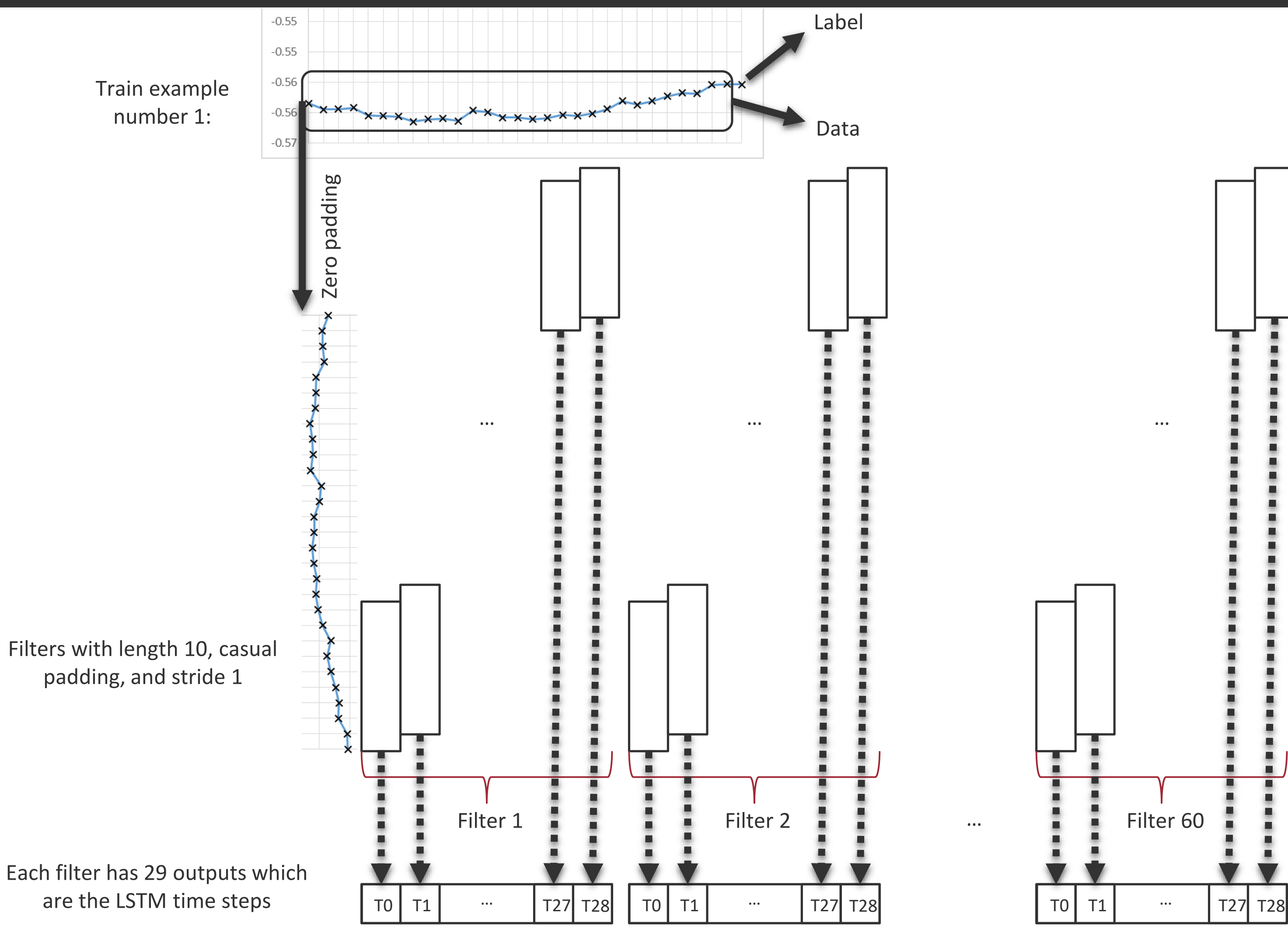
Number of features in each
example = 29



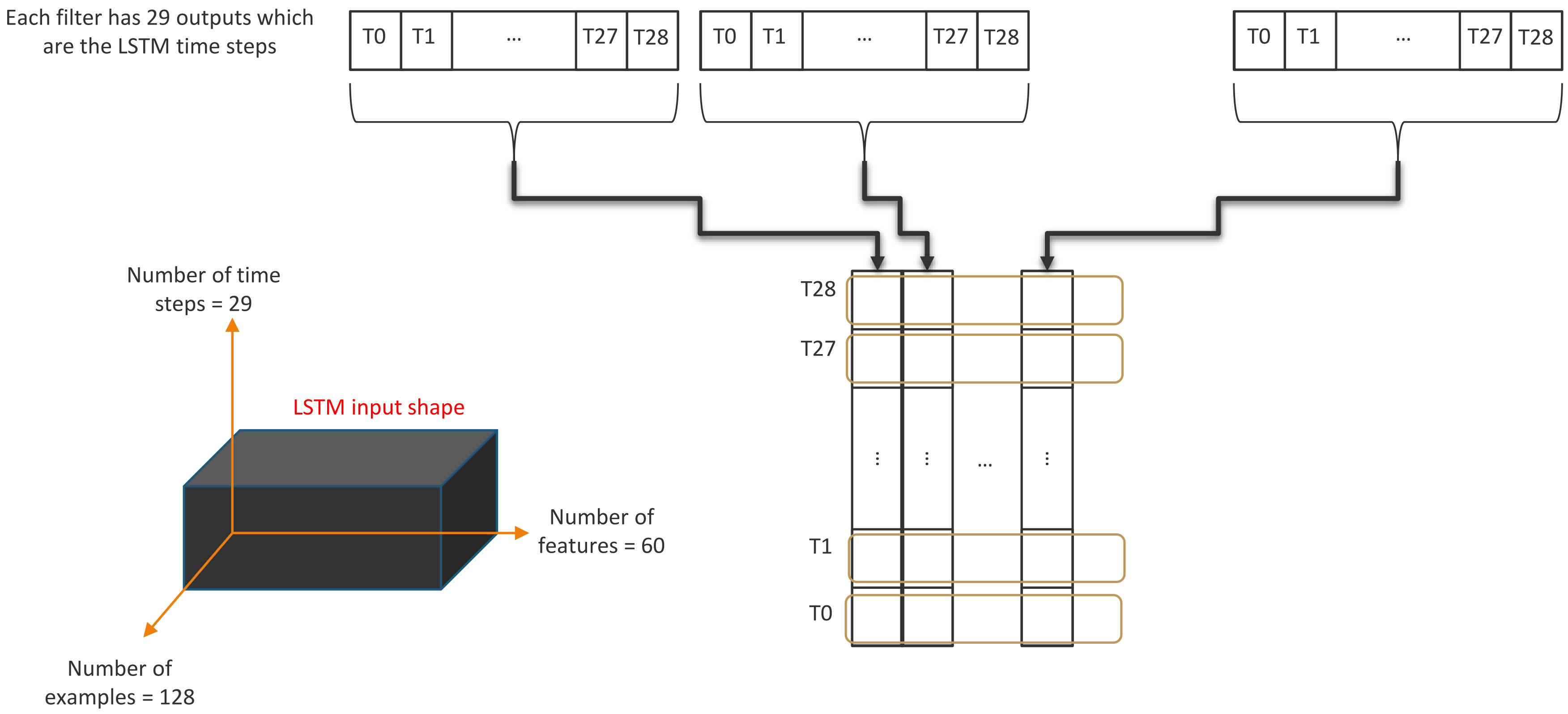
Number of points in each
feature= 1

Number of
examples = 128

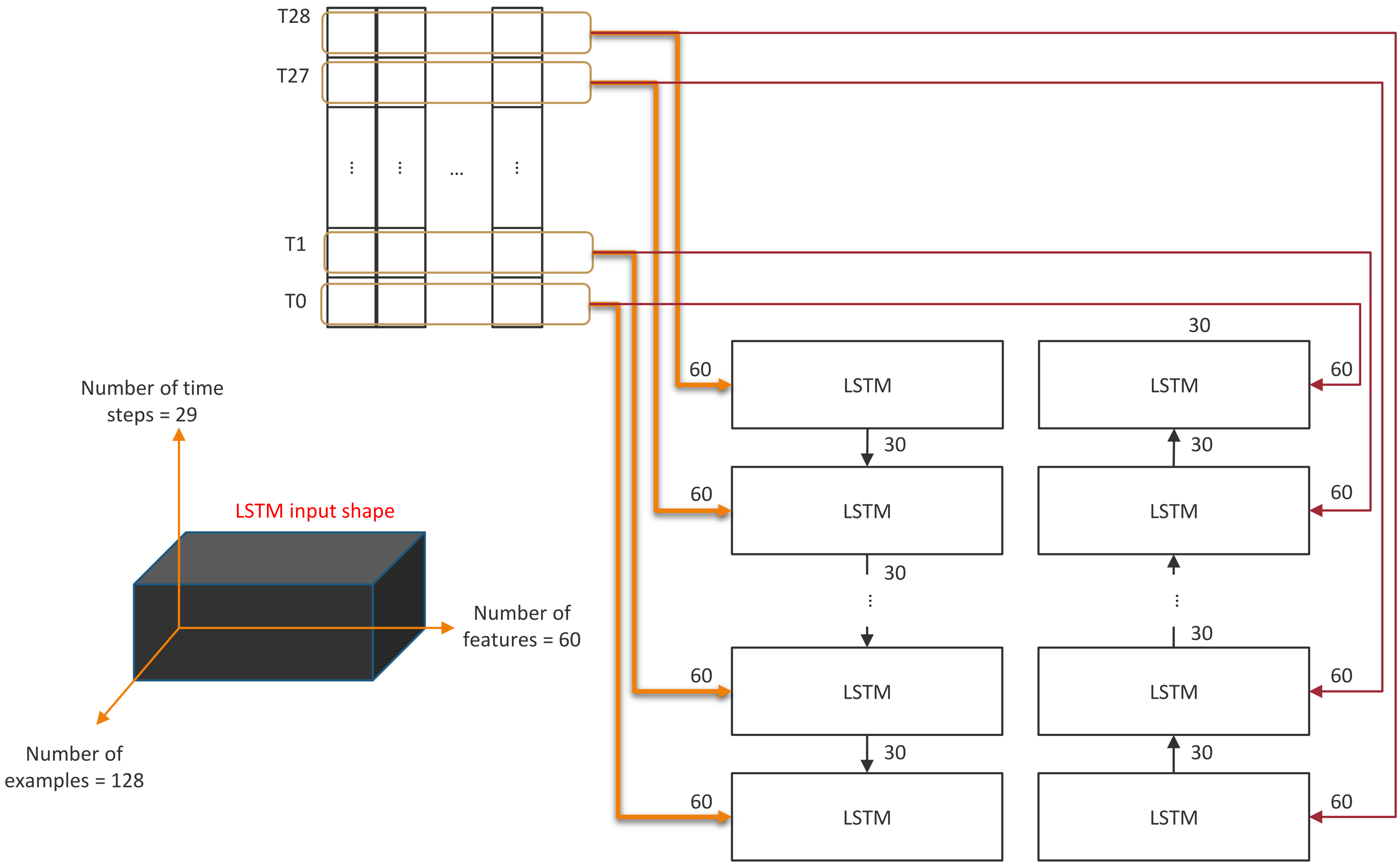
CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY



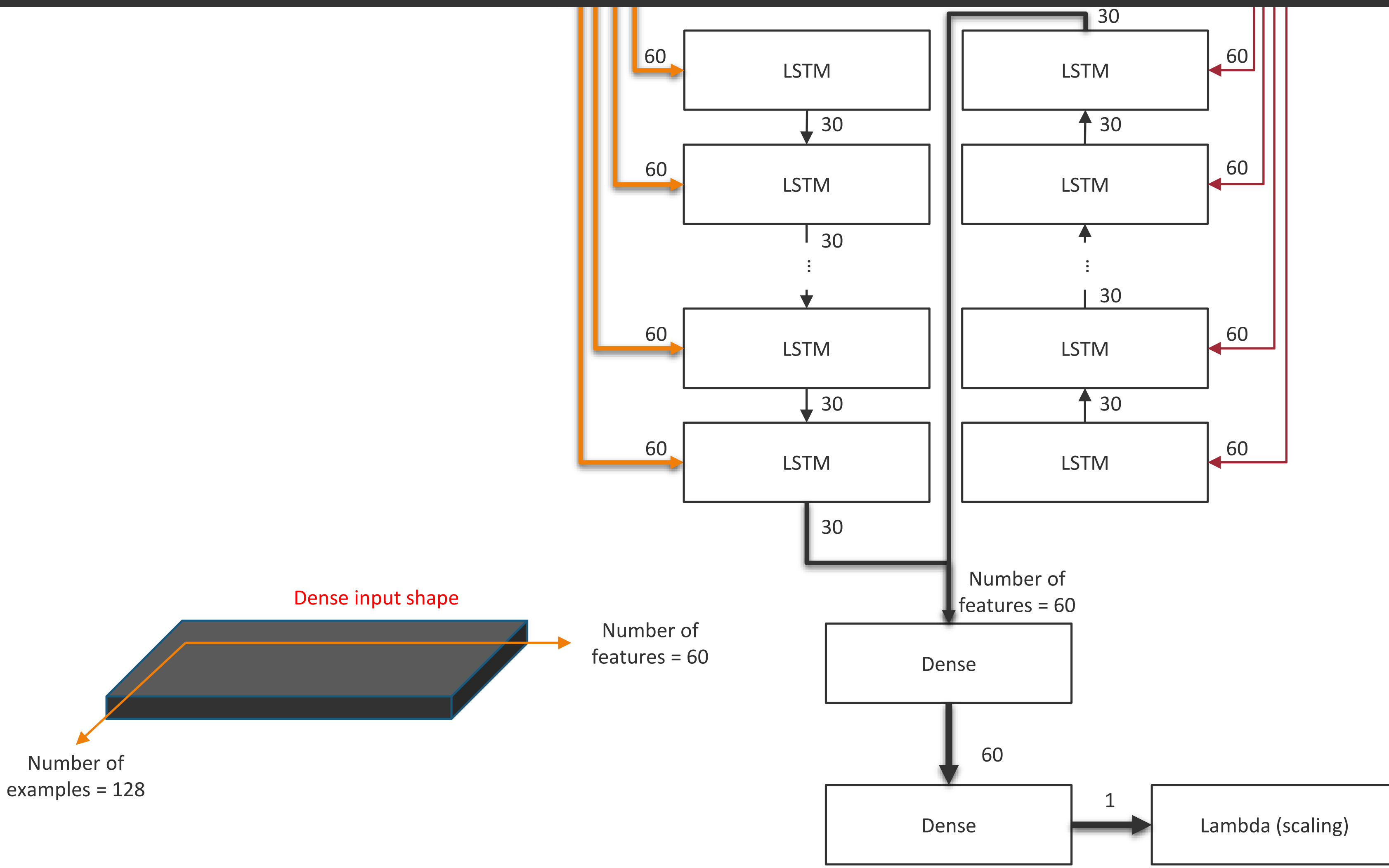
CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY



CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY



CONVOLUTIONAL NEURAL NETWORKS WITH LONG SHORT-TERM MEMORY



WHAT'S NEXT?

WHAT'S NEXT?

Bayesian models

Generative adversarial networks

Self-supervised learning

SOURCES

SOURCES

- https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9
- https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21
- https://www.novatec-gmbh.de/en/blog/recurrent-neural-networks-for-time-series-forecasting/
- http://clipart-library.com/
- https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn
- https://www.dlology.com/blog/how-to-use-return_state-or-return_sequences-in-keras/

SOURCES

- https://colah.github.io/posts/2015-08-Understanding-LSTMs/
- https://towardsdatascience.com/ghost-writing-with-tensorflow-49e77e26978f
- https://hackernoon.com/understanding-architecture-of-lstm-cell-from-scratch-with-code-8da40f0b71f4
- http://imatge-upc.github.io/telecombcn-2016-dlcv/slides/D2L6-recurrent.pdf
- https://d2l.ai/chapter_recurrent-modern/gru.html
- https://www.kdnuggets.com/2020/06/introduction-convolutional-neural-networks.html
- https://mgubaidullin.github.io/deeplearning4j-docs/usingrnns.html