COMP809 Data Mining and Machine Learning

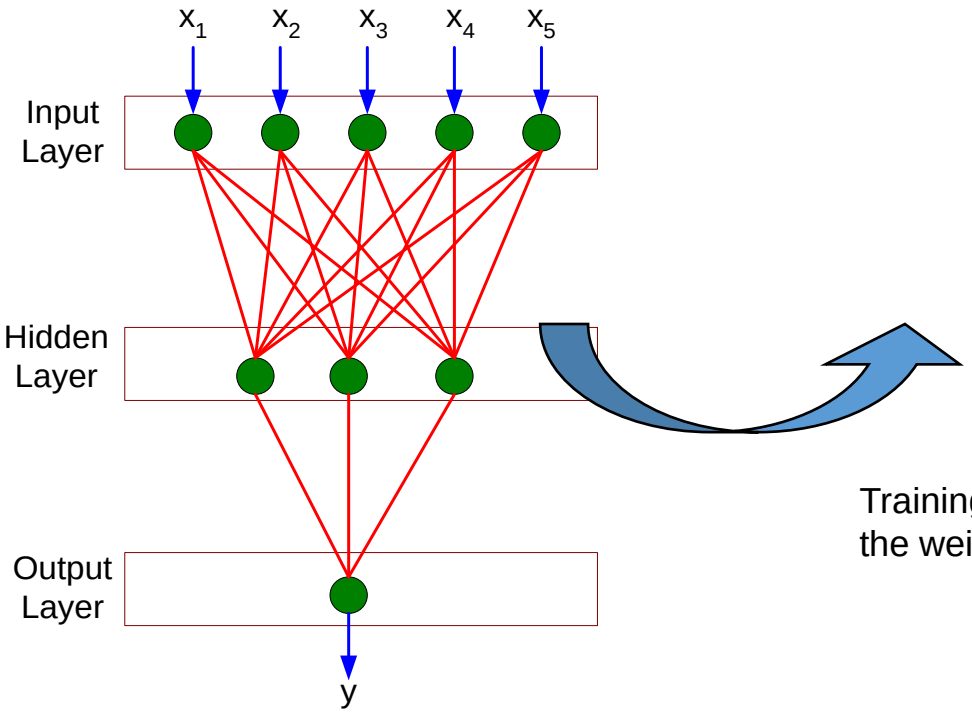
LECTURER: DR AKBAR GHOBAKHLOU

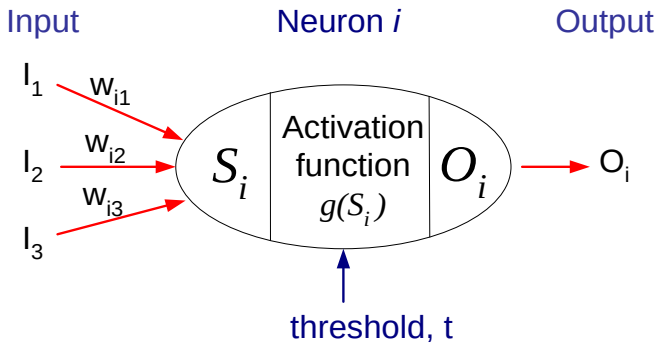
SCHOOL OF ENGINEERING, COMPUTER AND MATHEMATICAL SCIENCES

RNN and LSTM

Neural Networks for Numeric and Time Series Prediction

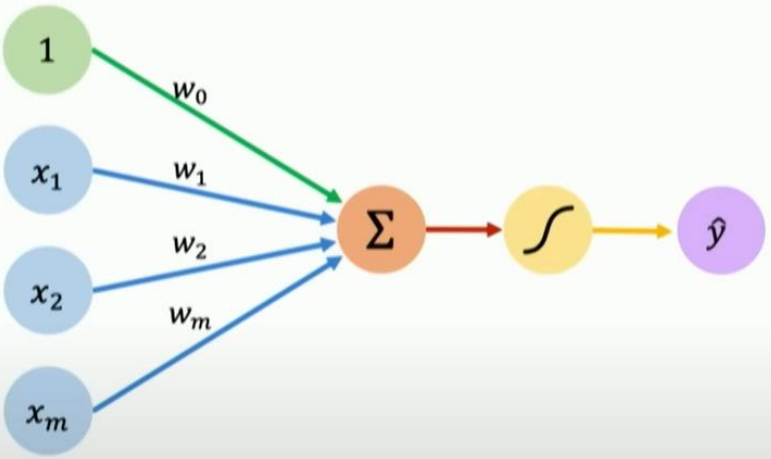
General Structure of ANN





Training ANN means learning the weights of the neurons

The Perceptron: Forward propagation



Activation Functions

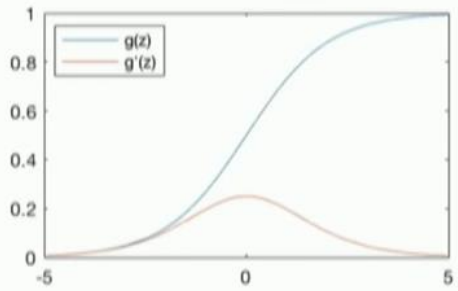
$$\hat{y} = g(w_0 + \boldsymbol{X}^T \boldsymbol{W})$$

- Example: sigmoid function

$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Common Activation Functions

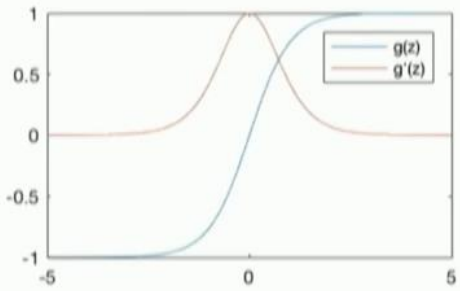
Sigmoid Function



$$g(z) = \frac{1}{1 + e^{-z}}$$

$$g'(z) = g(z)(1 - g(z))$$

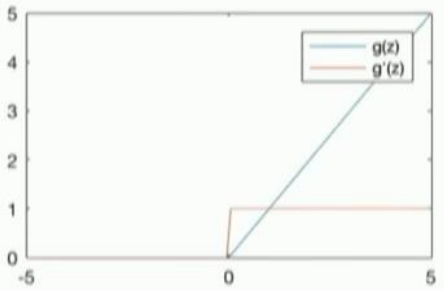
Hyperbolic Tangent



$$g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

$$g'(z) = 1 - g(z)^2$$

Rectified Linear Unit (ReLU)



$$g(z) = \max(0, z)$$

$$g'(z) = \begin{cases} 1, & z > 0 \\ 0, & \text{otherwise} \end{cases}$$

Limitations of Feedforward Neural Networks

- Standard feedforward networks while powerful for many types of problems are unable to cope with patterns that develop over time

- This is due to the fact that feedforward networks process each input independently of the others

  – thus they are incapable of capturing patterns that present in sequences

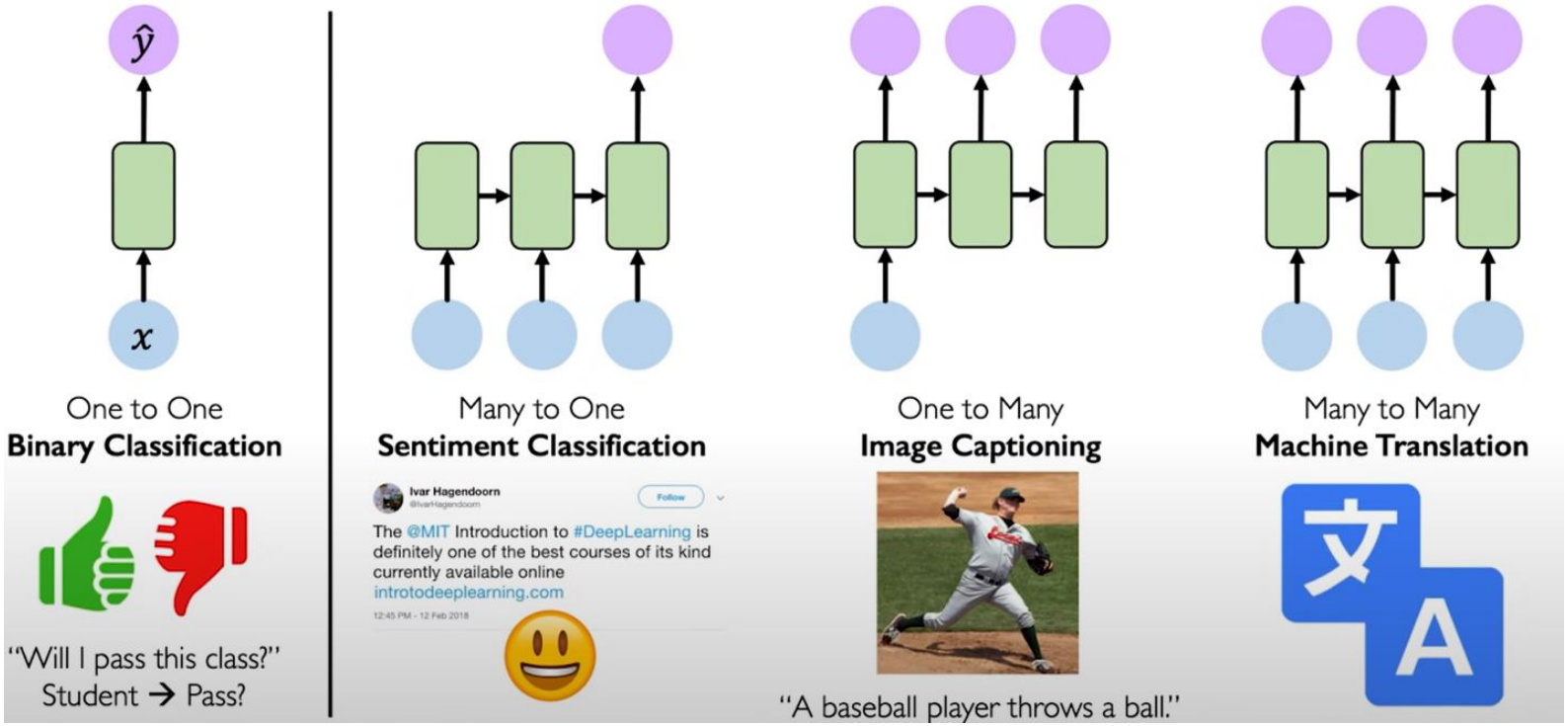
- Such types of applications are common – for example *time series modelling* (*stock price prediction*, *rainfall prediction*, *text mining*, etc.)

Sequences in the wiled

Characters C O M P 8 0 9

Words Machine Learning Approaches to Numeric Prediction

Sequence Modelling Applications



Neural Networks for Time Series Prediction

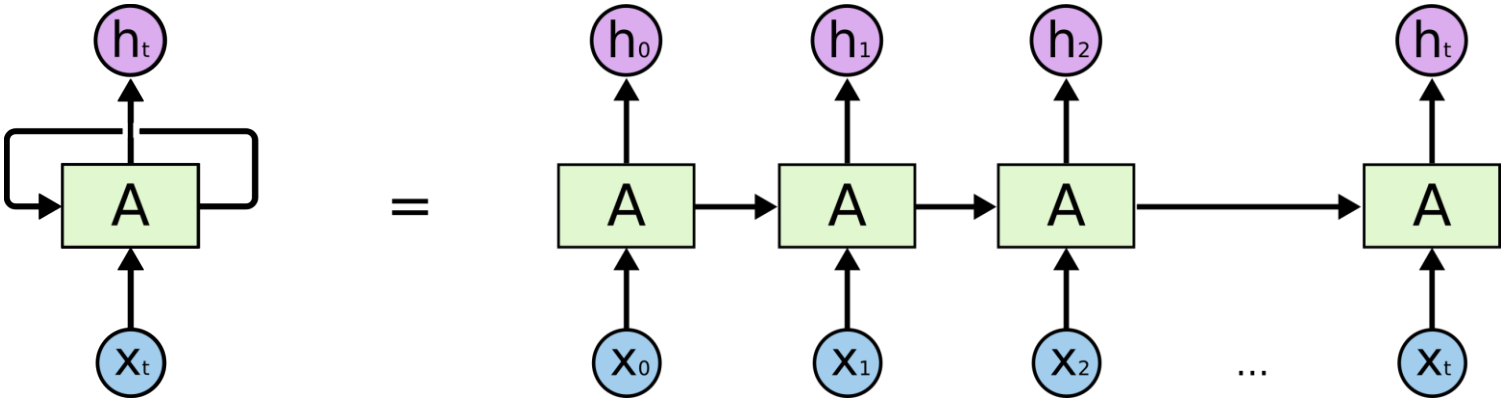
- Standard feedforward neural networks such as the MLP can be used for numeric prediction but cannot capture dependencies over time
- For time series applications hidden nodes in a ANN are modified to contain feedback loops to themselves (in addition to contain forward connections to the next layer)
- Such types of NNs are referred to as Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs)

- Order is important
- Variable length
- Used for sequential data
- Each item is processed in context
- Used for audio/music

A Basic RNN

- In a basic RNN each hidden node contains a feedback loop to itself.
- The loop iterates over different time steps enabling the network to learn temporal patterns.



An unrolled recurrent neural network.

Learning a RNN

- Backpropagation is once again the learning mechanism used to compute weights
- In this case, backpropagation over time is used to learn the weight vectors for each time step.
- Two major issues arise with the basic RNN:
  1. Exploding gradients
  2. Vanishing gradients
- Error gradients accumulate during a weight update and can result in very large gradients.
- The explosion occurs through exponential growth by repeatedly multiplying gradients through the network layers that have values larger than 1.0.
- In the extreme case, the values of weights can become so large as to overflow and result in NaN values, thus crippling the network.

Vanishing Gradients

- Vanishing gradient is the opposite problem, it occurs when the gradients at weight update steps are smaller than 1.0 and the network is deep.
- With a vanishing gradient weight updates do not occur; this prevents learning from taking place.
- A special type of RNN called the Long Short term Memory (LSTM) was developed by *Hochreiter & Schmidhuber* that resolves the problem of vanishing gradients.

*Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". Neural Computation. **9** (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276.*

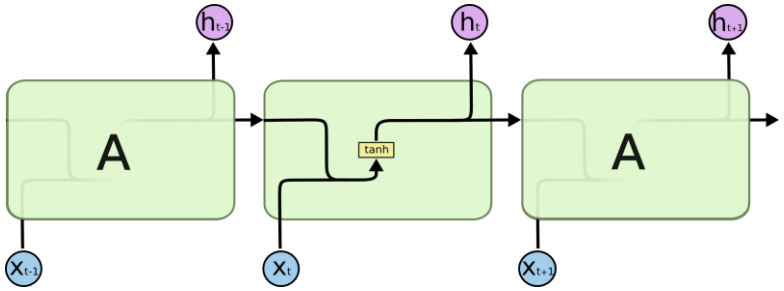
Issues with simple RNNs

- No long-term memory
- Network can’t use info from the distant past
- Can’t learn patterns with long dependencies

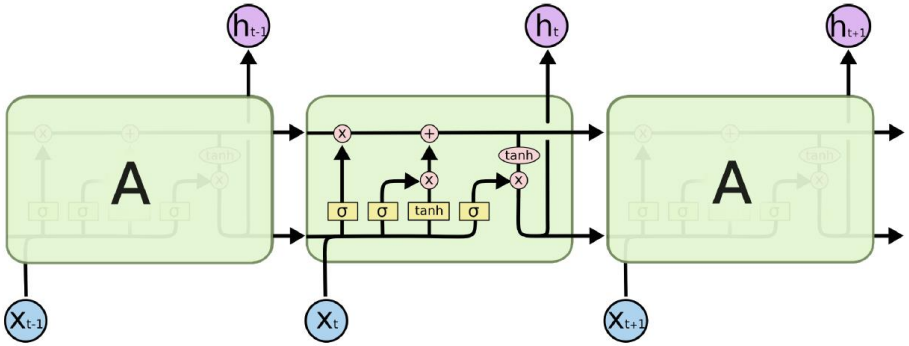
Long Short Term Memory (LSTM)

- Special type of RNN
- Can learn long-term patterns
- Detects patterns with 100 steps
- Struggles with 100s/1000s of steps

Standard RNN



The repeating module in a standard RNN contains a single layer



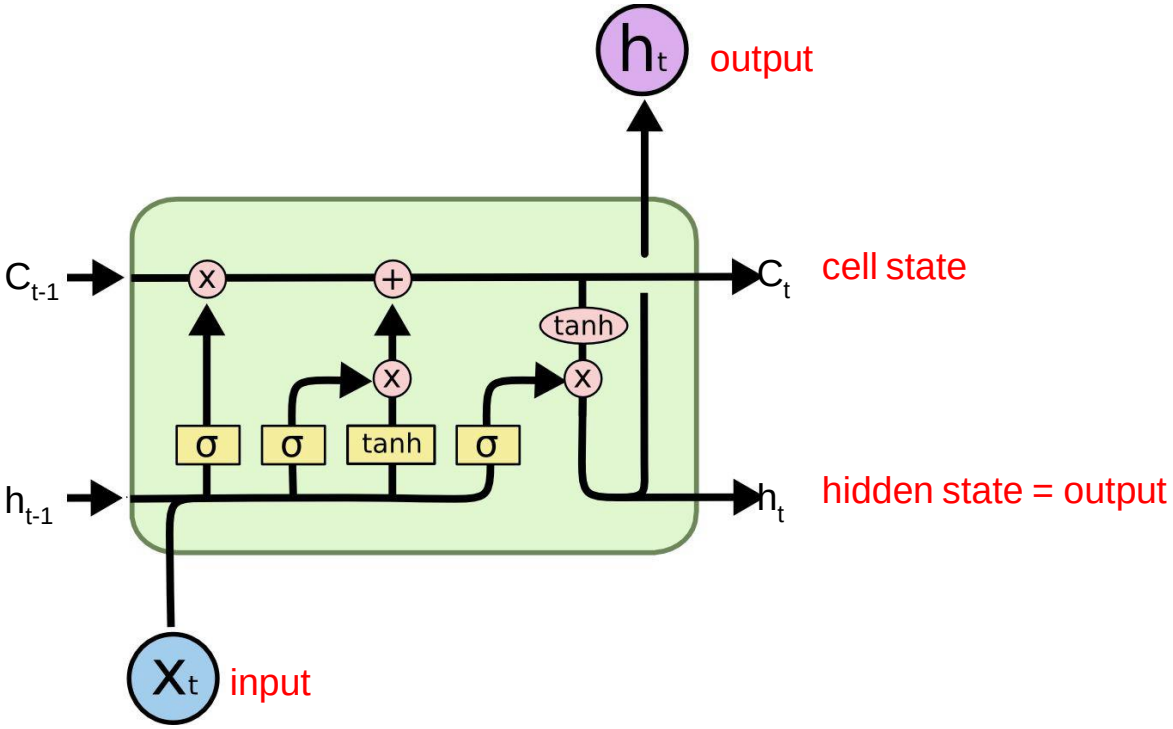
The repeating module in an LSTM contains four interacting layers.

taken from" Understanding LSTM Networks" http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM cell

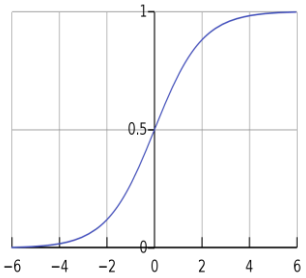
- Contains a simple RNN cell
- Second state vector = cell state = long-term memory
- Forget gate
- Input gate
- Output gate
- Gates work as filters

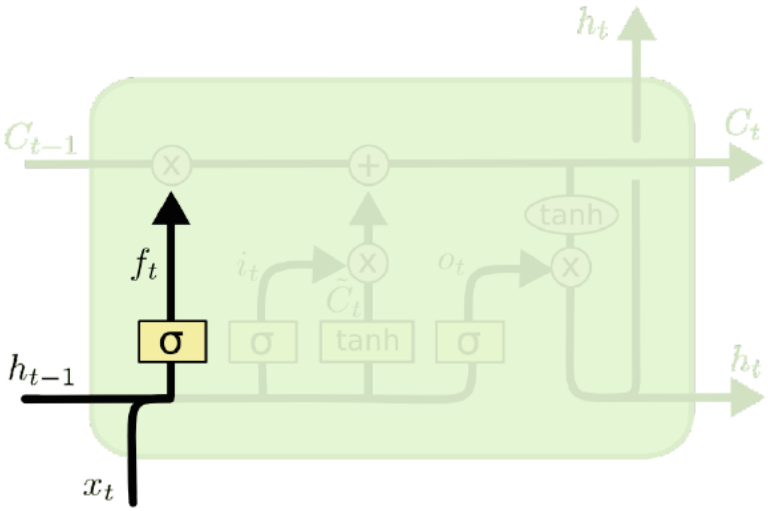
LSTM cell


h_t
output
C_{t-1}
C_t
cell state
tanh
σ
σ
tanh
σ
h_{t-1}
h_t
hidden state = output
X_t
input

Forget Gate

- How much of the past state h_{t-1} should we forget?

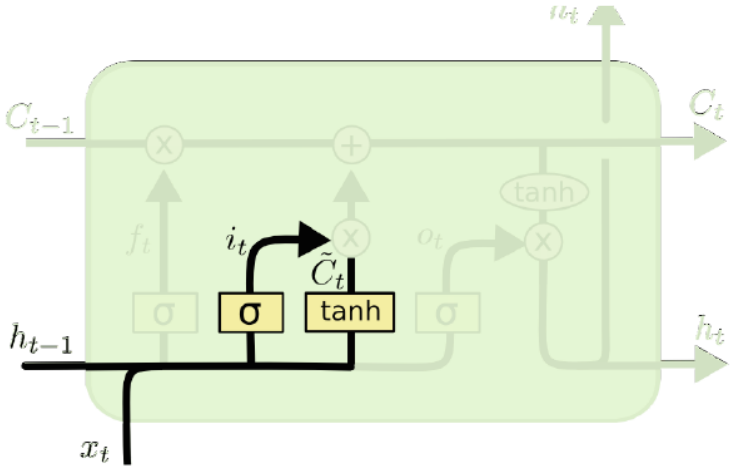




$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Input Gate

- Should we input current data or not?

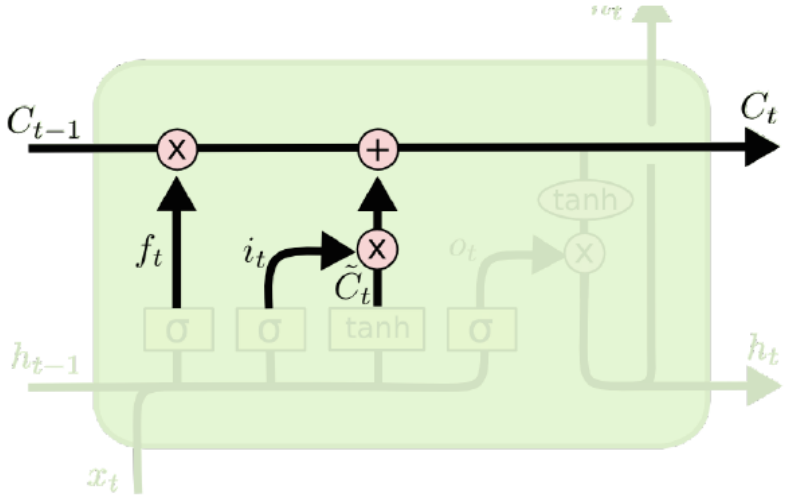


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

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

Memory Update

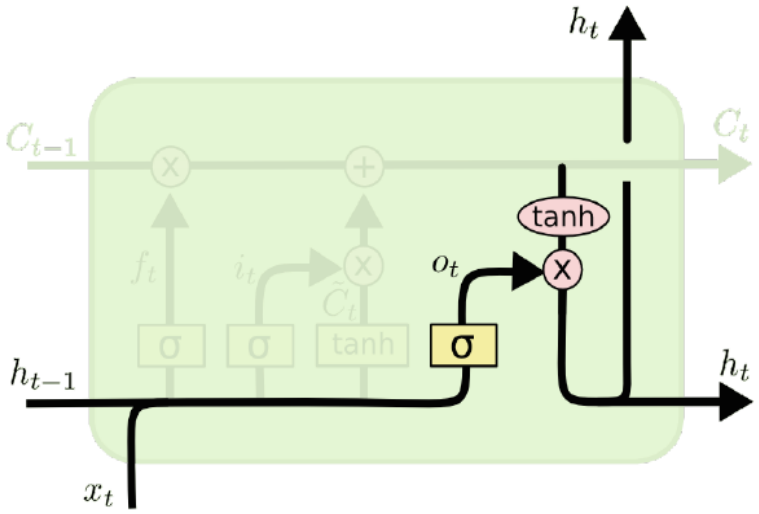
- Now collate what needs to be forgotten and what needs to be remembered



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

Output Gate

- Should we output to next state/layer (h_{t+1})?

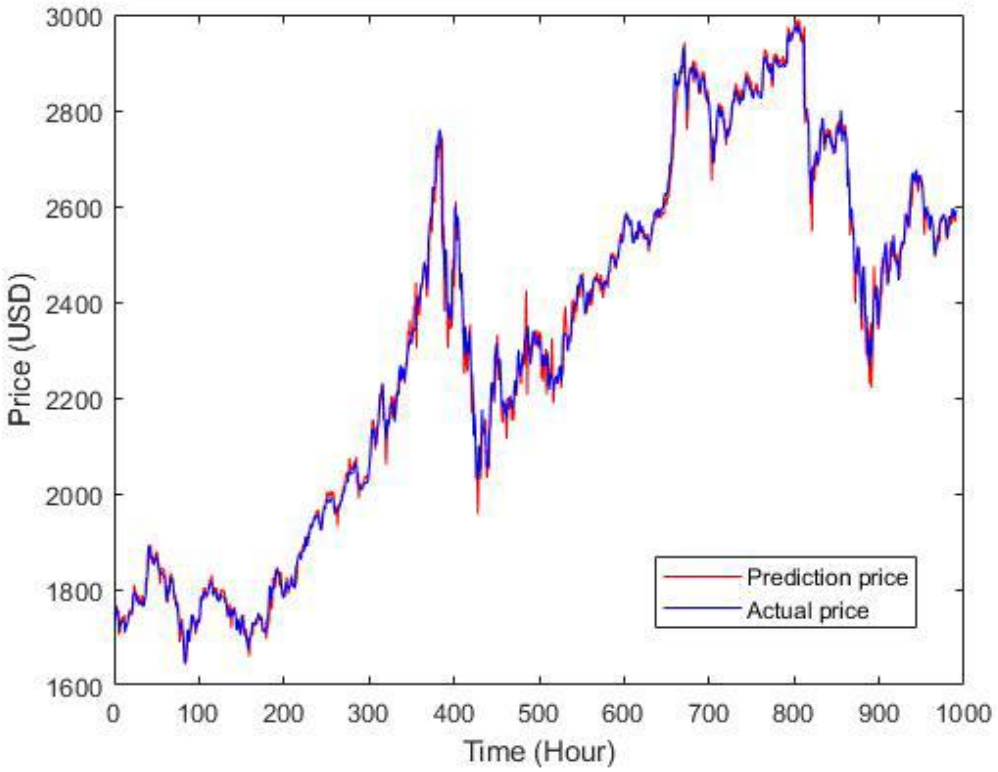


$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

LSTM in Action: Numeric Time Series Prediction

- The most straightforward application of LSTM is in time series prediction – predicting the movement of stock price, bitcoin price, etc.
- Prediction of Bitcoin movement



LSTM in action: Language Translation

- Machine Translation also known as sequence to sequence learning (https://arxiv.org/pdf/1409.3215.pdf)
- From a high level perspective the translation proceeds as follows:

1. An encoder (an LSTM) uses the known input from the source language (say English) to convert each word in the sentence to a numeric vector representation
2. A decoder (another LSTM) converts the encoded numeric vector into a sentence in the target language (say French). This conversion maximises the conditional probability of obtaining the French sentence given the original English sentence

LSTM in action: Google Translate

Actual (French):
“ Les te´le´phones portables sont ve´ritablement un probl`eme , non seulement parce qu’ ils pourraient e´ventuellement cre´er des interfe´rences avec les instruments de navigation , mais parce que nous savons , d’ apre`s la FCC , qu’ ils pourraient perturber les antennes-relais de te´le´phonie mobile s’ ils sont utilise´s a` bord ” , a de´clare´ Rosenker .

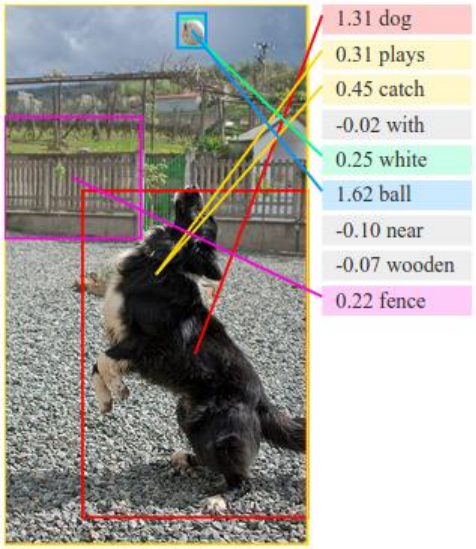
Actual English Translation
"Cellular telephones, which are really a question, not only because they could potentially interfere with navigation devices, but we know, according to the FCC, that they could interfere with cell phone towers when they are in the air, "says UNK.

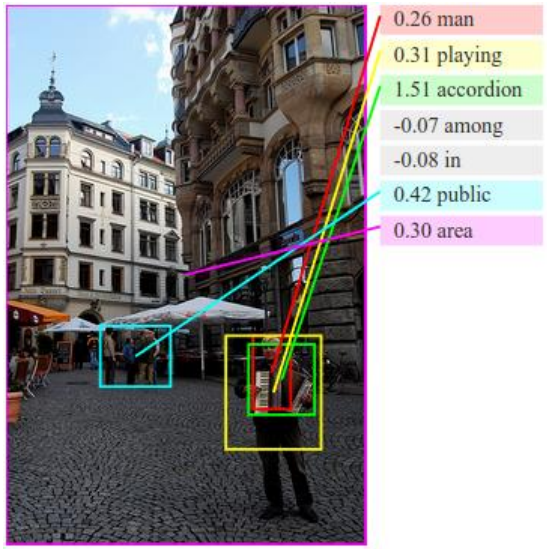
Model (French):
“ Les te´le´phones portables sont ve´ritablement un probl`eme , non seulement parce qu’ ils pourraient e´ventuellement cre´er des interfe´rences avec les instruments de navigation , mais parce que nous savons , d’ apre`s la FCC , qu’ ils pourraient perturber les antennes-relais de te´le´phonie mobile s’ ils sont utilise´s a` bord ” , a de´clare´ Rosenker .

Model (English Translation):
"Mobile phones are really a problem, not only because they could eventually interfere with navigational instruments, but because we know, afterwards. s the FCC, that they could disrupt mobile telephone relay antennas if they are used on board, "said Rosenker

LSTM in action: Image Captioning

- One of the most interesting and useful applications is assigning meaningful text to images
- Uses a combination of CNN (object recognition) and LSTM (sentence generation)
- Image captioning (with and without attention, https://arxiv.org/pdf/1411.4555v...)





Other applications of LSTM networks

- Hand writing generation (http://arxiv.org/pdf/1308.0850v5...)
- Image generation using attention models (https://arxiv.org/pdf/1502.04623...)
- Question answering (http://www.aclweb.org/anthology/...)
- Video to text (https://arxiv.org/pdf/1505.00487...)