# MACHINE LEARNING APPROACHES TO NUMERIC PREDICTION

Regression and Time Series Analysis

# Linear Regression

## What is Regression?

*“Regression analysis is a statistical tool for the investigation of relationships between variables. Usually, the investigator seeks to ascertain the causal effect of one variable upon another — the effect of a price increase upon demand, for example, or the effect of changes in the money supply upon the inflation rate.” Source: Sykes (1993).*

So, regression:

a) is a set of tools/techniques

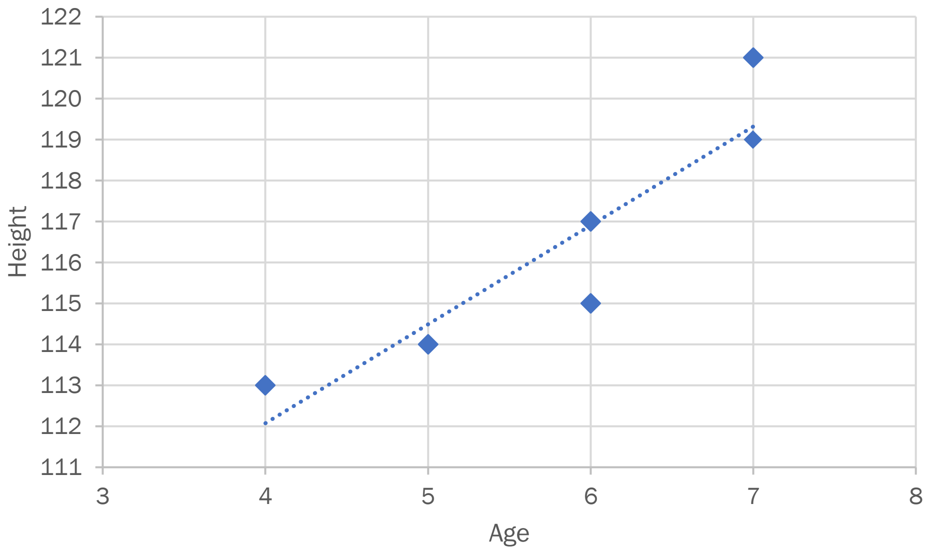
b) regression determines whether there is a relationship between two or more variables

c) regression measures the size/strength of the relationship

# Linear Regression

***Simple linear regression** is used to estimate the relationship between two quantitative variables.*

| Student no | Age | Height |
|---|---|---|
| 1 | 4 | 113 |
| 2 | 7 | 121 |
| 3 | 6 | 117 |
| 4 | 5 | 114 |
| 5 | 6 | 115 |
| 6 | 7 | 119 |
| 7 | 4 | ? |



- Linear regression constructs a linear model that **best fits** a given training dataset
  - By best fit we mean that it minimizes the sum of squares errors

# Regression Fundamentals 1

- Given data with n dimensional variables and 1 target-variable (real number)

  $$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$$

  Where $\mathbf{x} \in \Re^n, y \in \Re$
- The objective: Find a function f that returns the best fit. $f : \Re^n \rightarrow \Re$
- Assume that the relationship between X and y is approximately linear. The model can be represented as (w represents coefficients and b is an intercept)

  $$f(w_1, \ldots, w_n, b) = y = \mathbf{w} \cdot \mathbf{x} + b + \varepsilon$$

# Regression Fundamentals 2

- To find the best fit, we minimize the sum of squared errors
- Which is:

$$\min \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{m} (y_i - (\mathbf{w} \cdot \mathbf{x}_i + b))^2$$

- We estimate w by $\widehat{w}$ by taking the derivative of the above objective function w.r.t. w)

where

$$\hat{\mathbf{w}} = (X^T X)^{-1} X^T Y$$

# Regression Fundamentals 3: Derivation of vector *w*

Our regression expression is:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1K} \\ x_{21} & x_{22} & \cdots & x_{2K} \\ \vdots & \ddots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NK} \end{bmatrix} * \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}$$

$$\epsilon^T \epsilon = \begin{bmatrix} e_1 & e_2 & \cdots & e_N \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} = \sum_{i=1}^N e_i^2$$

$$\epsilon^T \epsilon = (y - Xw)^T (y - Xw)$$

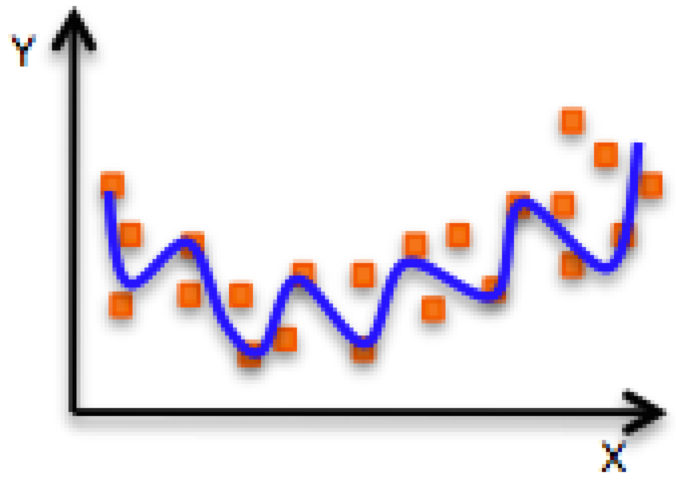
$$\epsilon^T \epsilon = y'y - 2w^T X^T y + w^T X^T X w$$

$$\frac{\partial(\epsilon^T \epsilon)}{\partial w} = -2X^T y + 2X^T X w = 0$$

$$X^T X w = X^T y$$

$$w = (X^T X)^{-1} X^T y$$

# Guarding against overfitting

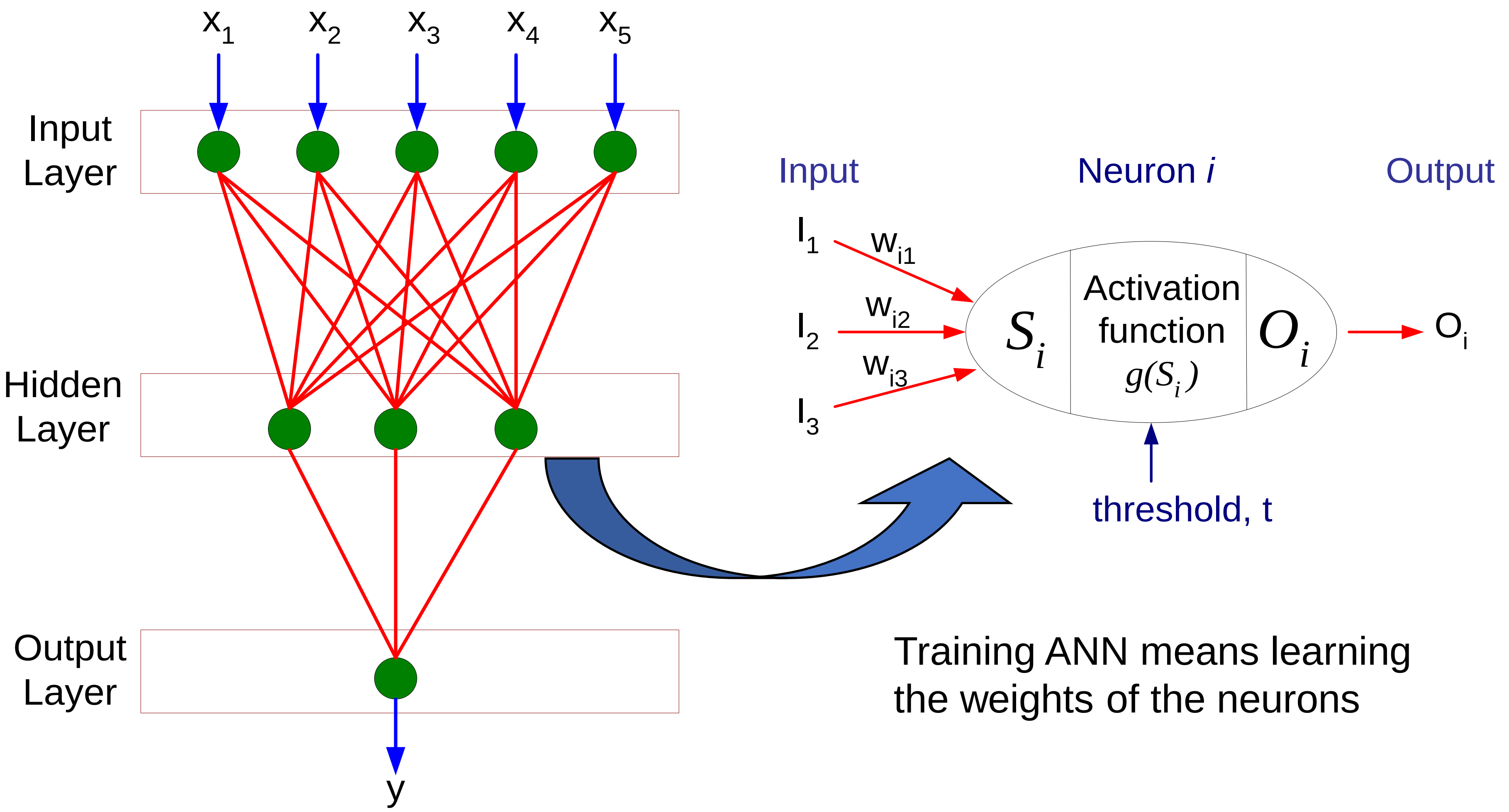


overfitting

- To ovoid over-fitting, a regularization term can be introduced (minimize a magnitude of w)
  - LASSO: $$\min \sum_{i=1}^{m}(y_i - \mathbf{w}\cdot\mathbf{x}_i - b)^2 + C\sum_{j=1}^{n}|w_j|$$
  - Ridge regression: $$\min \sum_{i=1}^{m}(y_i - \mathbf{w}\cdot\mathbf{x}_i - b)^2 + C\sum_{j=1}^{n}|\mathbf{w}_j^2|$$

# Neural Networks for Numeric and Time Series Prediction

# General Structure of ANN



Training ANN means learning the weights of the neurons

# 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

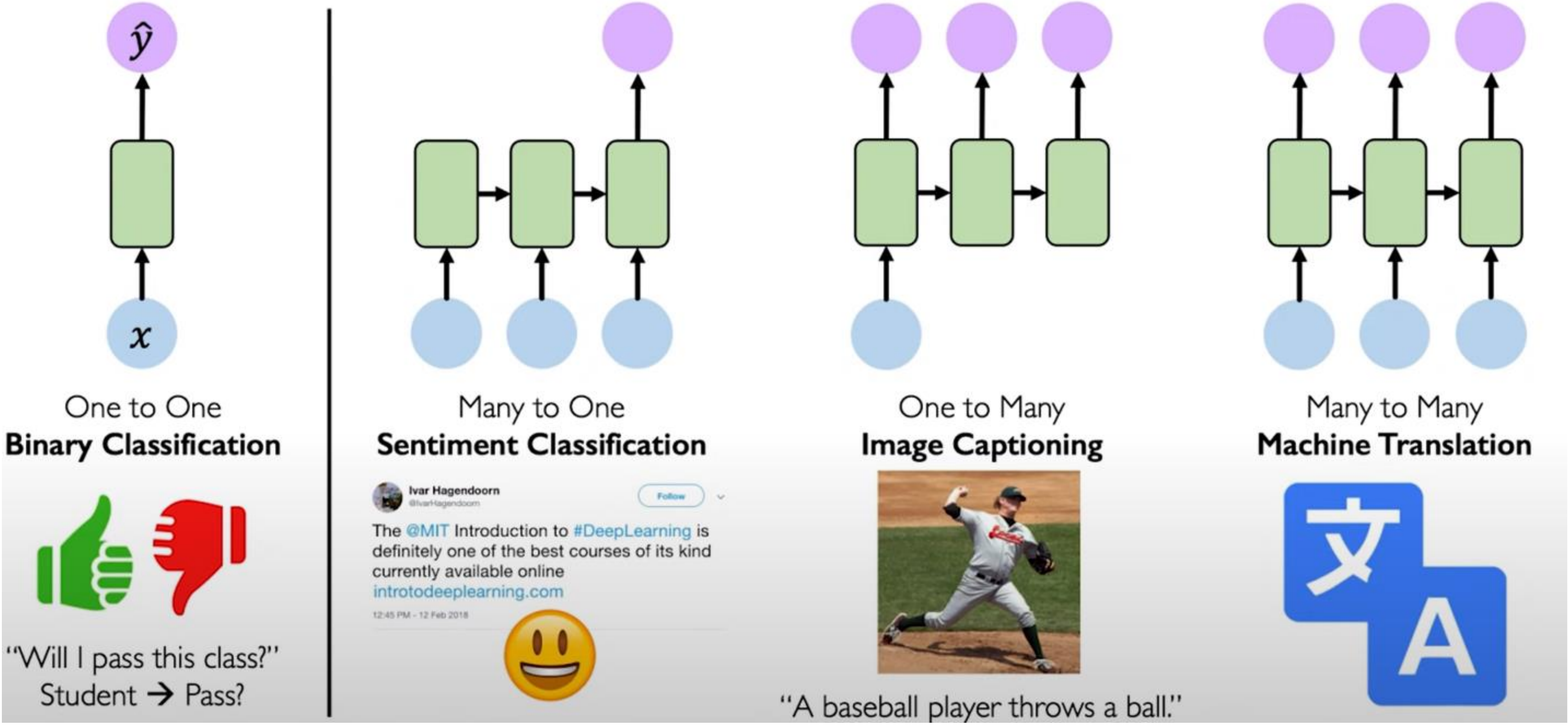
  – *thus they are incapable of capturing patterns that present in sequences*
- Such types of applications are common – for example *time series modelling* (*stock price prediction*, *rainfall prediction*, *text mining*, etc.)

# Sequences in the wiled

Characters C O M P 8 0 9

Words Machine Learning Approaches to Numeric Prediction

# Sequence Modelling Applications



One to One
**Binary Classification**

"Will I pass this class?"
Student → Pass?

Many to One
**Sentiment Classification**

Ivar Hagendoorn
@IvarHagendoorn
Follow

The @MIT Introduction to #DeepLearning is definitely one of the best courses of its kind currently available online
introtodeeplearning.com

12:45 PM - 12 Feb 2018

One to Many
**Image Captioning**

"A baseball player throws a ball."

Many to Many
**Machine Translation**

# Neural Networks for Time Series Prediction

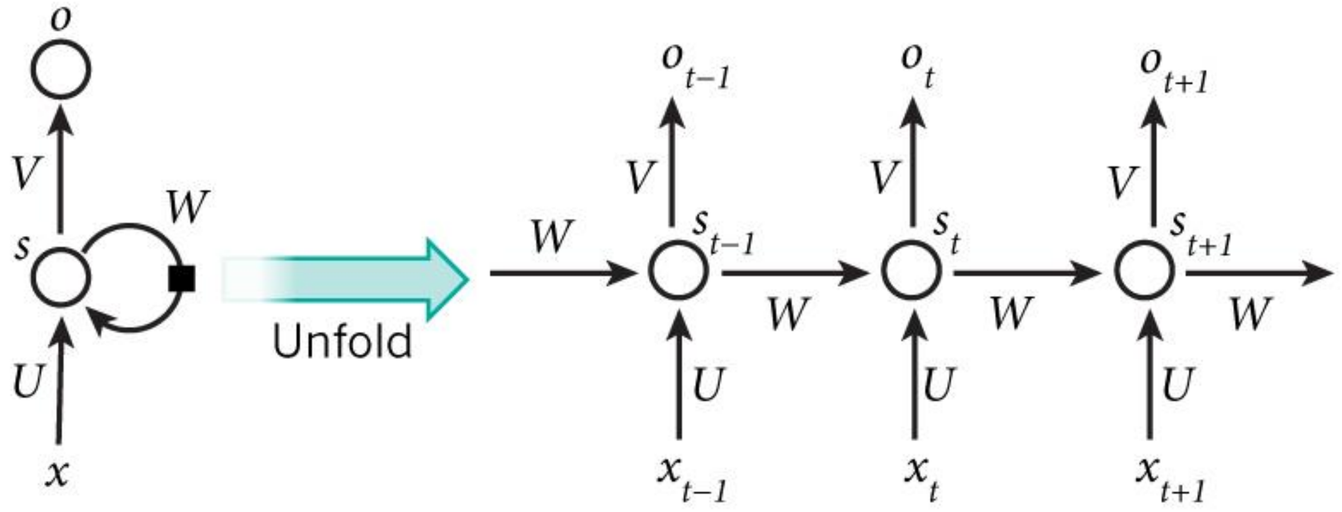
- Standard feedforward neural networks such as the MLP can be used for numeric prediction but cannot capture dependencies over time
- For time series applications hidden nodes in a ANN are modified to contain feedback loops to themselves (in addition to contain forward connections to the next layer)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



x:input; o:output;
s:hidden state;
U, V and W are
weight parameters

$s_t = f(Ux_t + Ws_{t-1})$ $o_t = softmax(Vs_t)$ where *f* is usually the tanh (Hyperbolic) function.

# 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 U, V and W.
- 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.*

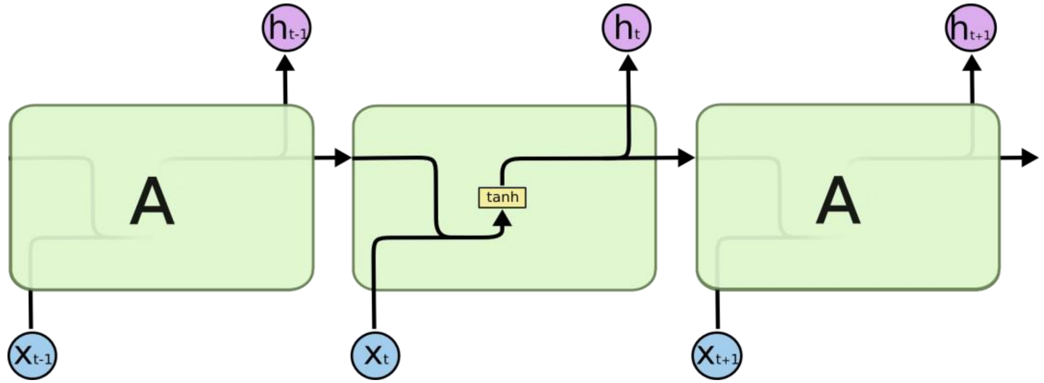
# Issues with simple RNNs

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

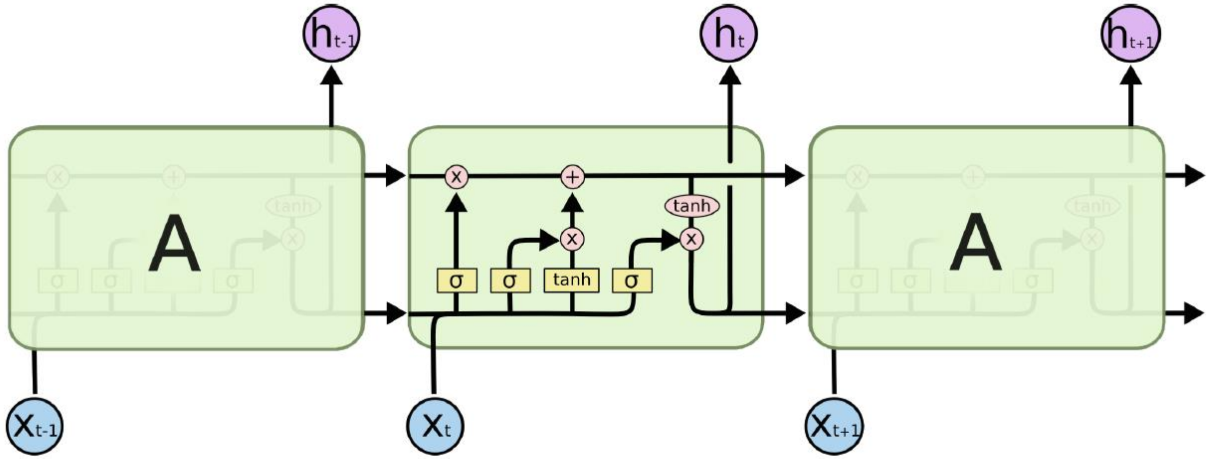
# Long Short Term Memory (LSTM)

- 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**



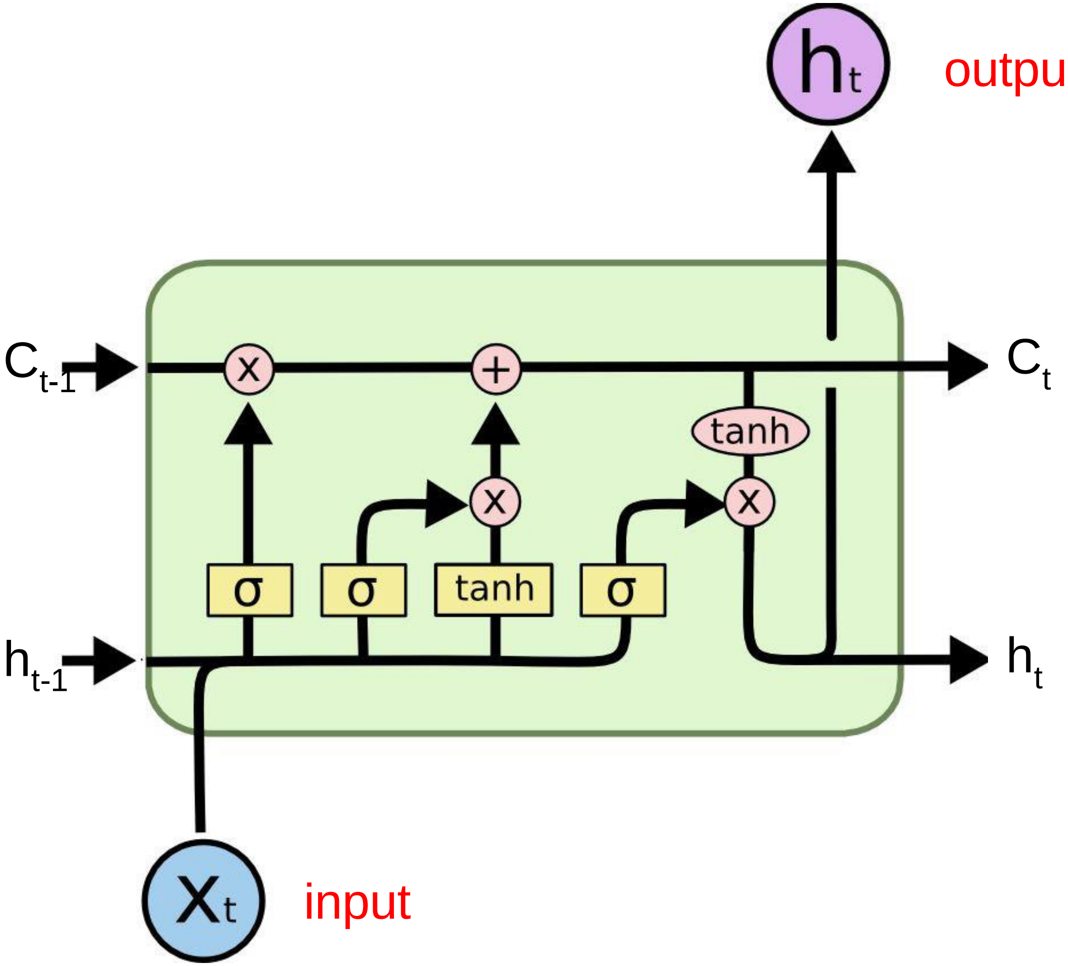
**The repeating module in an LSTM contains four interacting layers.**

taken from” Understanding LSTM Networks” [http://colah.github.io/posts/2015-08-Understanding-LSTMs/](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



output

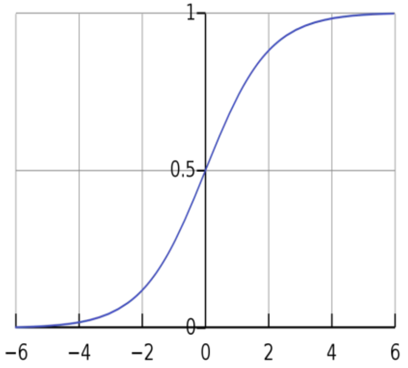
cell state

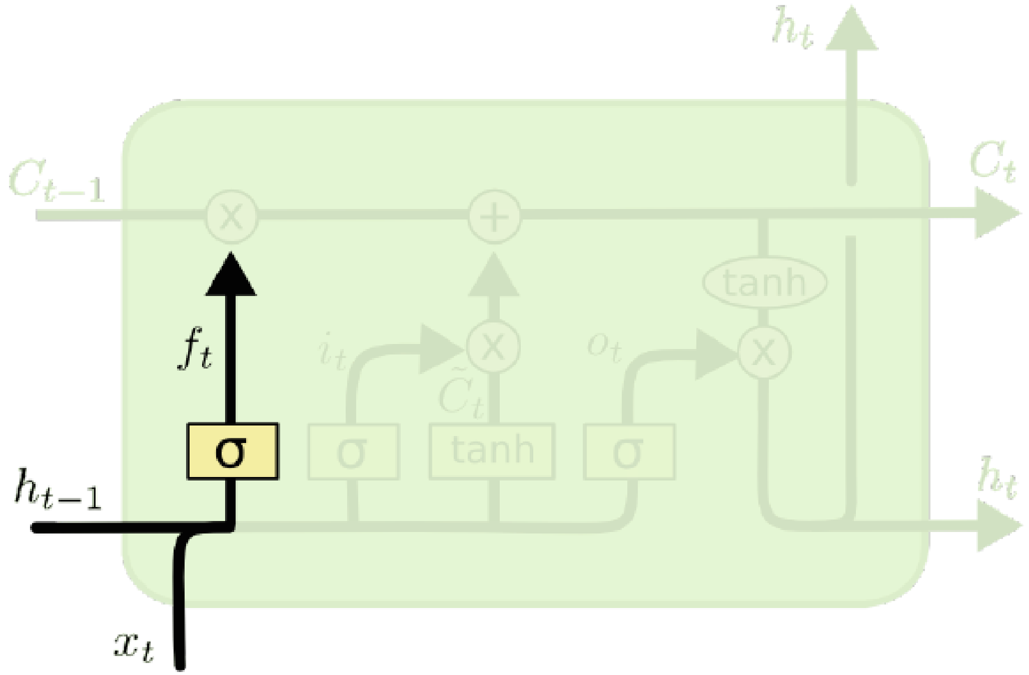
hidden state = output

input

# Forget Gate

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

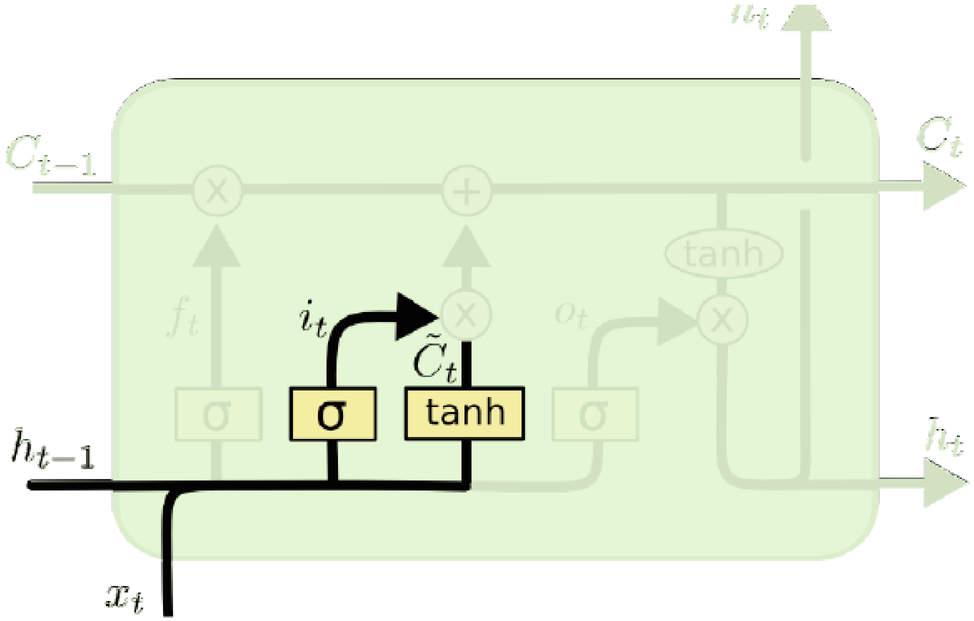




$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

# Input Gate

- Should we input current data or not?

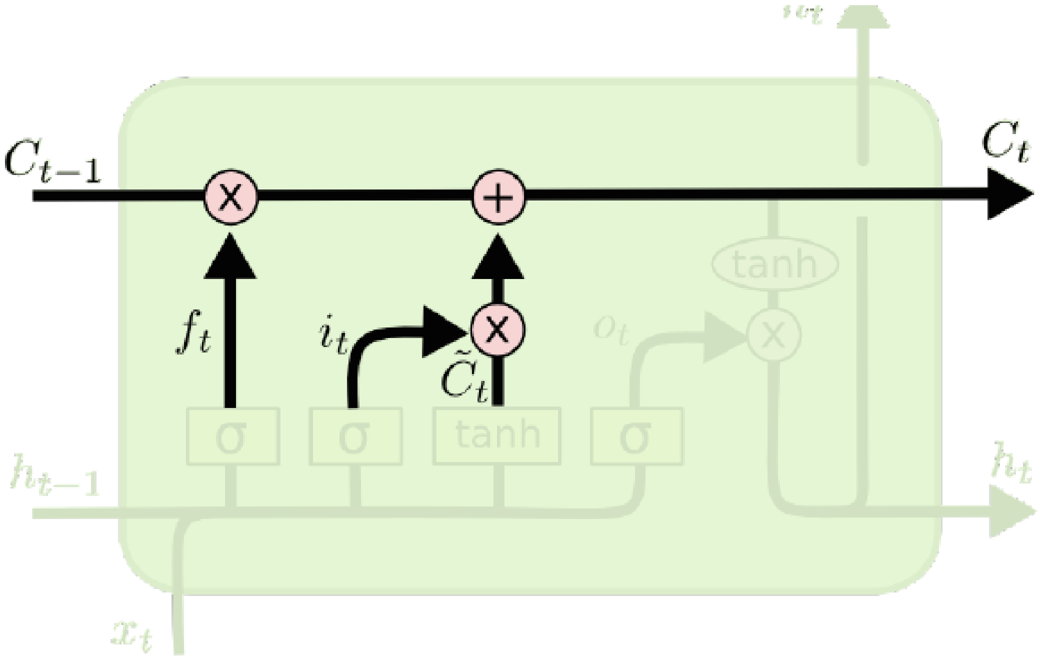


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

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

# Memory Update

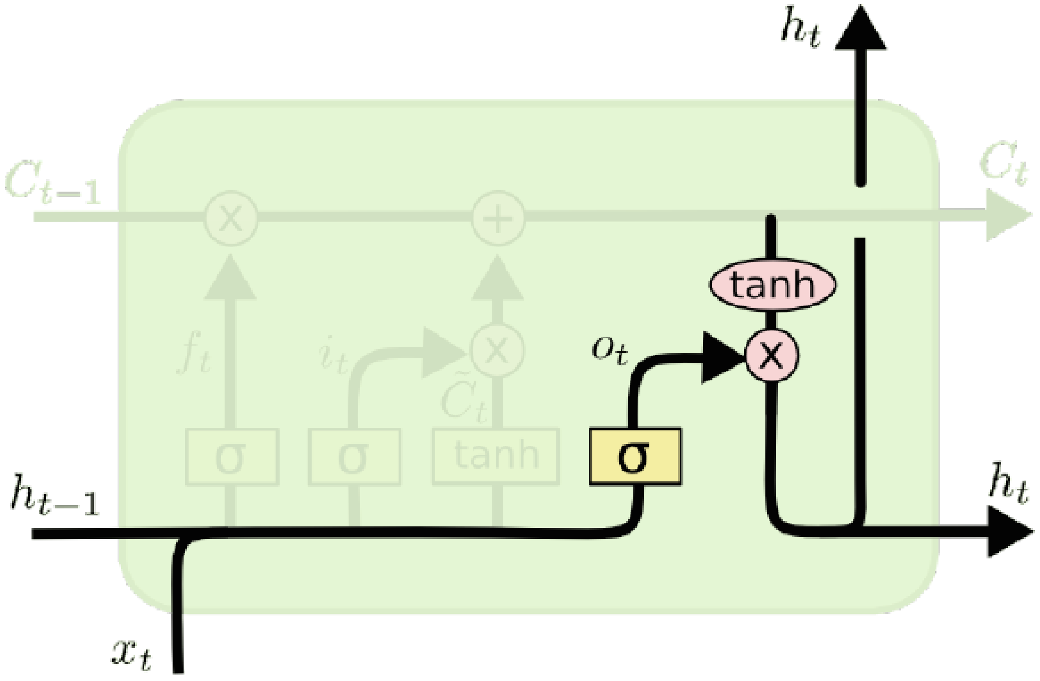
- Now collate what needs to be forgotten and what needs to be remembered



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

# Output Gate

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

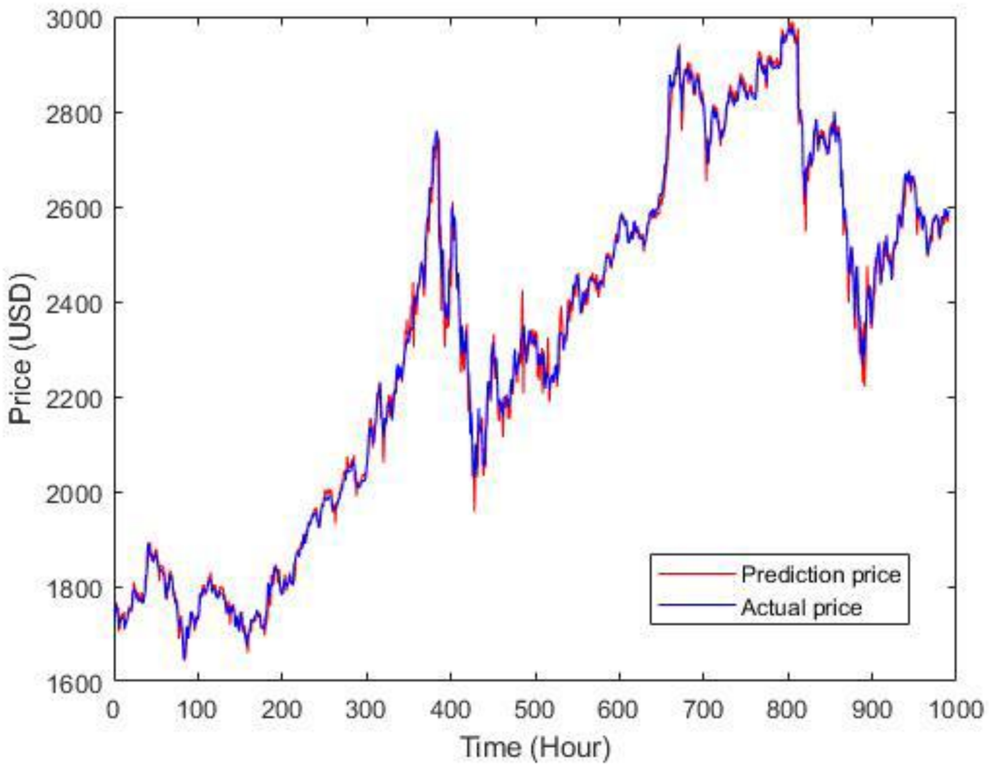


$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

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

# 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

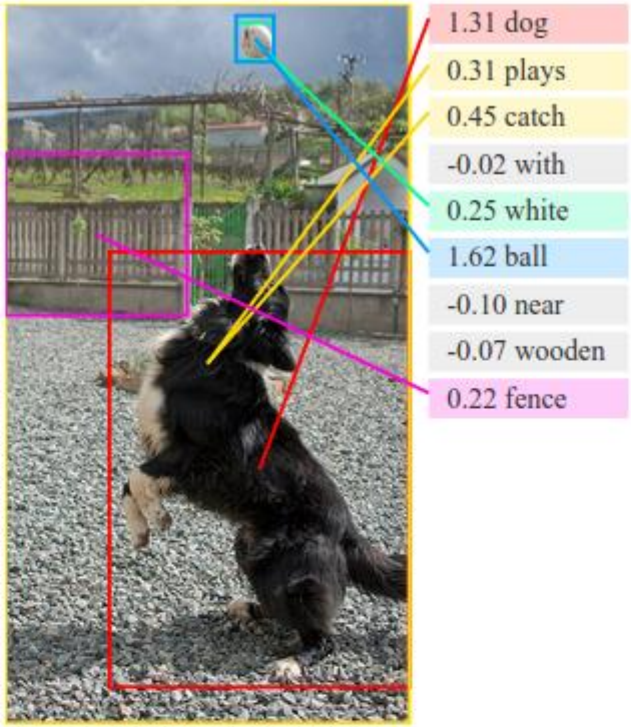
**Actual (French):**
“ Les te´le´phones portables sont ve´ritablement un probl`eme , non seulement parce qu’ ils pourraient e´ventuellement cre´er des interfe´rences avec les instruments de navigation , mais parce que nous savons , d’ apre`s la FCC , qu’ ils pourraient perturber les antennes-relais de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

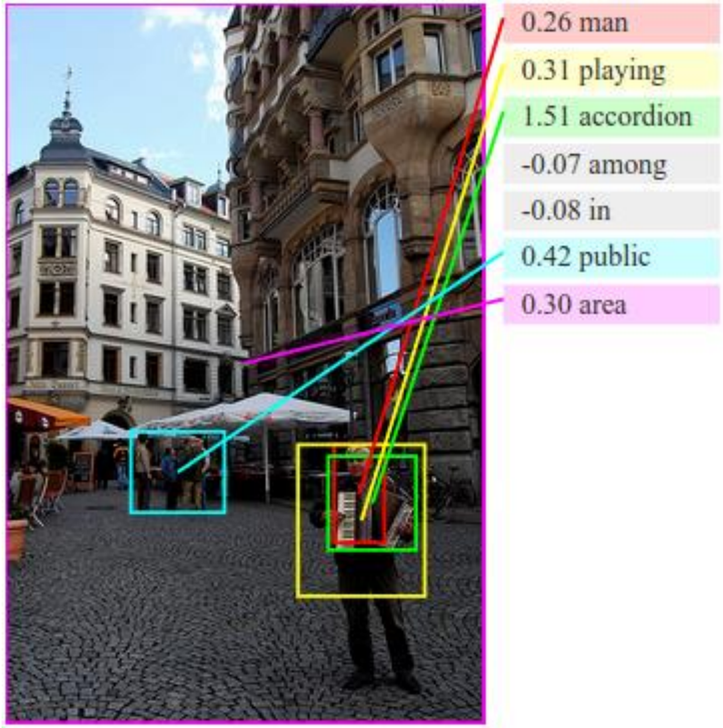
**Model (French):**
“ Les te´le´phones portables sont ve´ritablement un probl`eme , non seulement parce qu’ ils pourraient e´ventuellement cre´er des interfe´rences avec les instruments de navigation , mais parce que nous savons , d’ apre`s la FCC , qu’ ils pourraient perturber les antennes-relais de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

**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 (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...](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...)