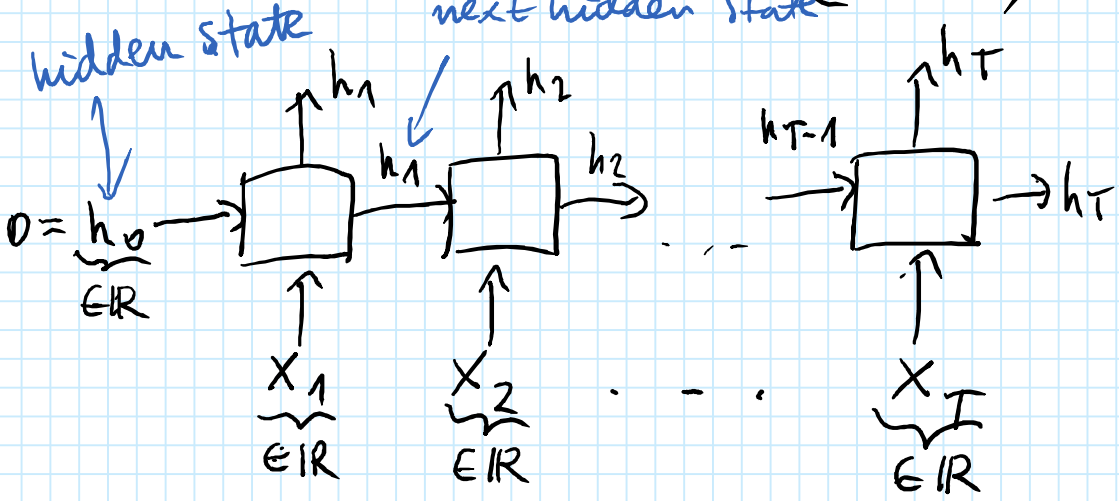
Machine Learning 9.12.2022

Recurrent Neural Network (RNN)



$$h_t = \sigma(W x_t + U h_{t-1} + b) \quad , \sigma - \text{sigmoid}$$

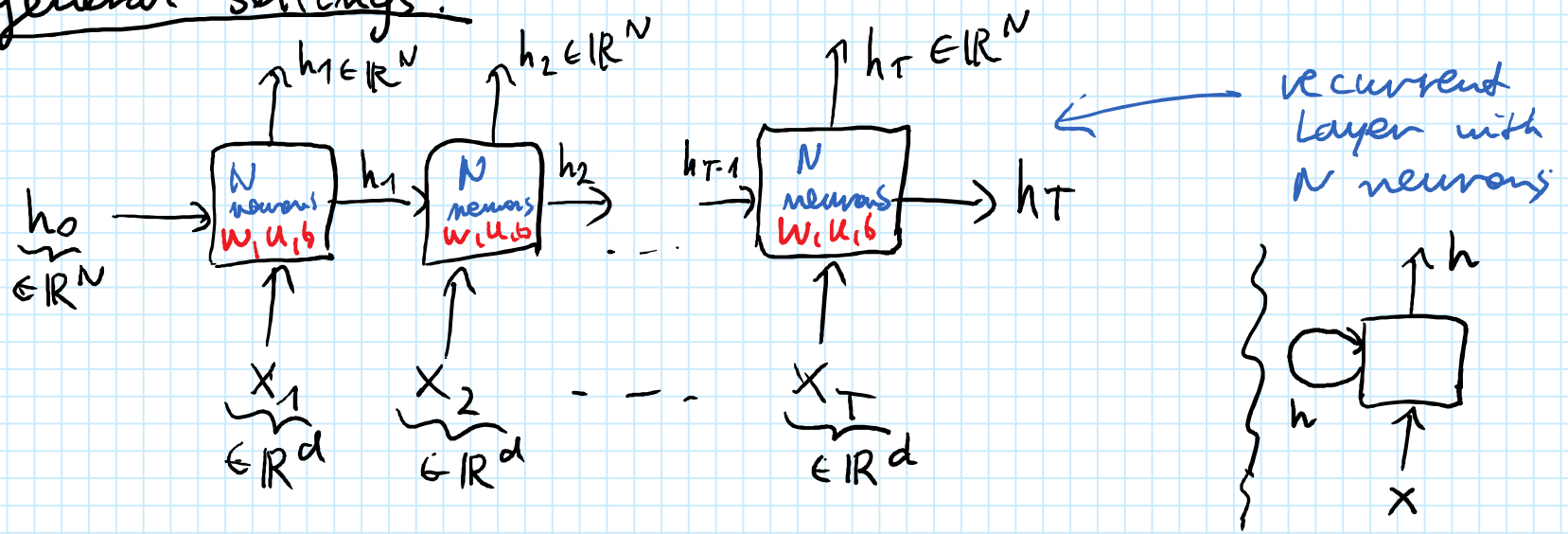
Applications

- Modelling Time Series
- Natural Language Processing (NLP)

Examples

- identyfing sarcastic comments (/s) on Reddit
- identyfying number of stars in movie review
- predicting closing price of stocks

More general settings:



$$\mathbb{R}^N \ni h_t = \sigma\Big(\underbrace{W}_{\in \mathbb{R}^{N\times d}} \underbrace{x_t}_{\in \mathbb{R}^d} + \underbrace{U}_{\in \mathbb{R}^{N\times N}} \underbrace{h_{t-1}}_{\in \mathbb{R}^N} + \underbrace{b}_{\in \mathbb{R}^N} \Big)$$

- σ (sigmoid) is acting on every coordinate
- W, U, b are the same at every time-stamp
- Memory is shared between all neurons

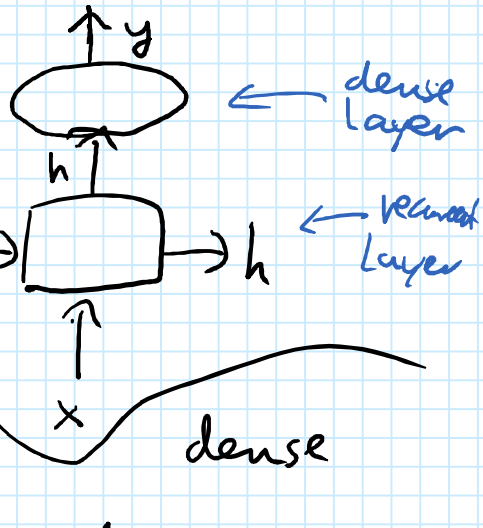
  Number of weights $= Nd + N^2 + N$
- Final output is h_T or $(h_1, h_2, \dots, h_T)$

Examples

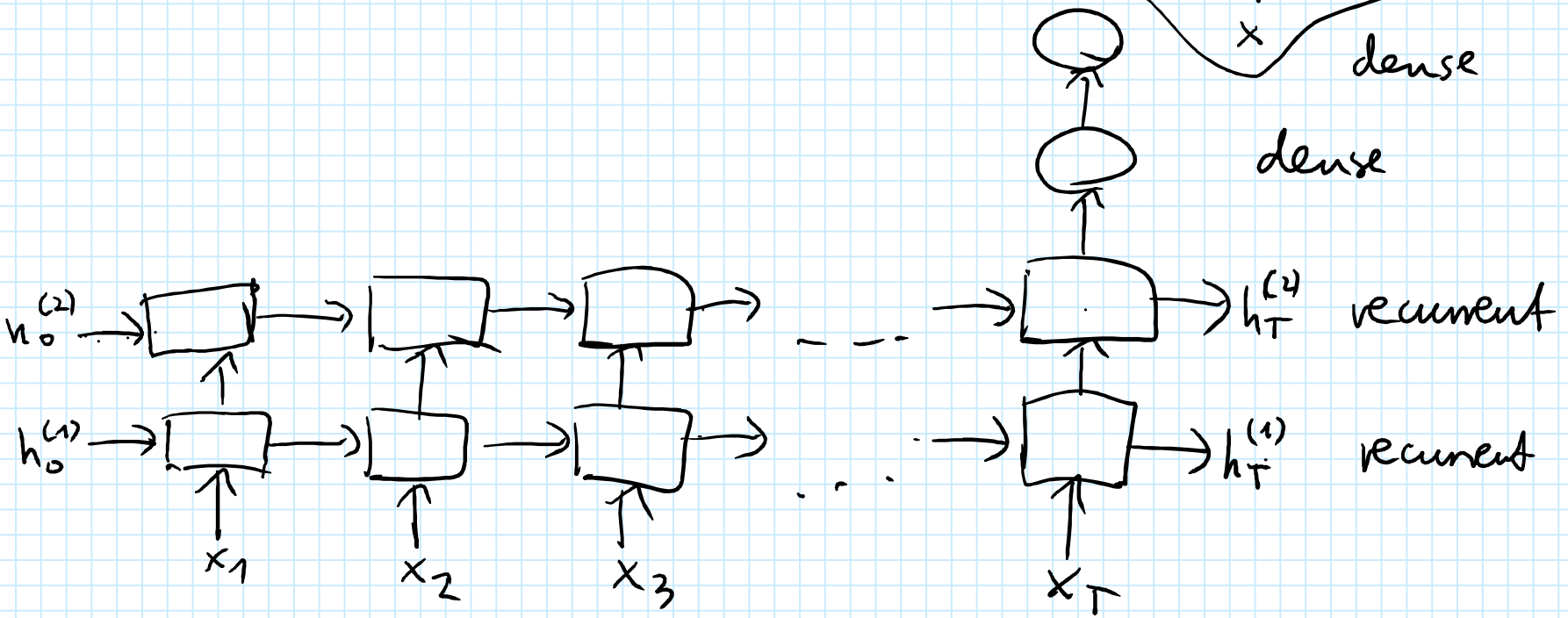
1) Elman Network

$$h_t = \sigma\left(W_h x_t + U_h h_{t-1} + b_h\right)$$

$$y_t = \sigma_y\left(W_h h_t + b_y\right)$$



2)



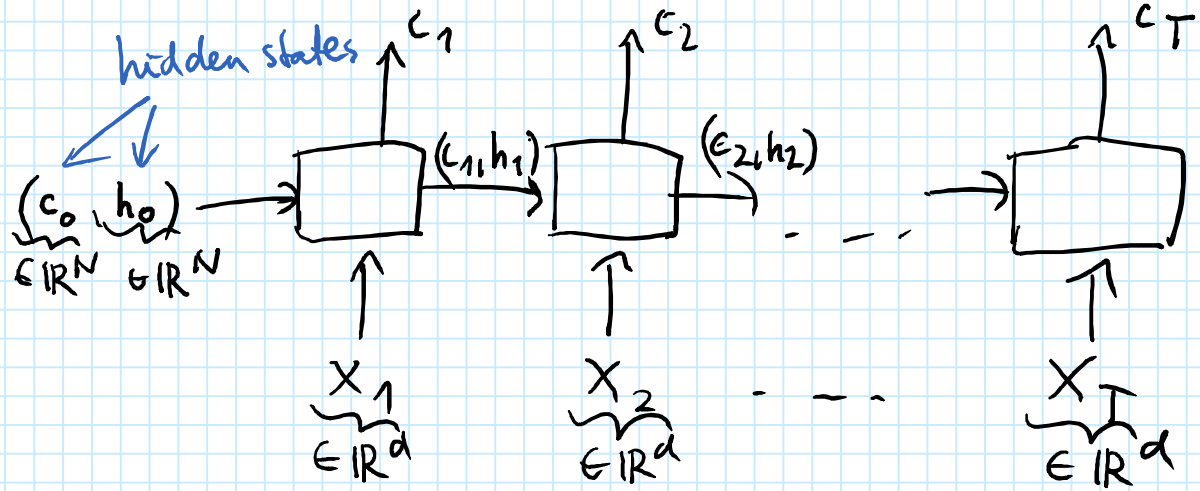
PROBLEM: Vanishing Gradients!

If T is large, then distance from x_1 to x_T is very long and the network is very deep in fact!

SOLUTION LSTM!

Long Short-Term Memory (LSTM)

Reference: Hochreiter, Schmidhuber 1997



$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right)$$
$$i_t = \sigma_g\left(W_i x_t + U_i h_{t-1} + b_i\right)$$
$$o_t = \sigma_g\left(W_o x_t + U_o h_{t-1} + b_o\right)$$
$$\tilde{c}_t = \sigma_c\left(W_c x_t + U_c h_{t-1} + b_c\right)$$

(W_f, W_i, W_o, W_c: $N \times d$; x_t: d; U_f, U_i, U_o, U_c: $N \times N$; h_{t-1}: N; b_f, b_i, b_o, b_c: N)

σ_g - sigmoid, σ_c - tanh

f_t - forget gate

i_t - input gate

o_t - output gate

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$
$$h_t = o_t \odot \sigma_c(c_t)$$

Number of weights $= 4(Nd + N^2 + N)$

GRU (Gated Recurrent Unit)

- less parameters than in LSTM
- very similar results

Number of weight $= 3(Nd + N^2 + N)$

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z)$$ update gate

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r)$$ reset gate

$$\hat{h}_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h)$$ candidate activation

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

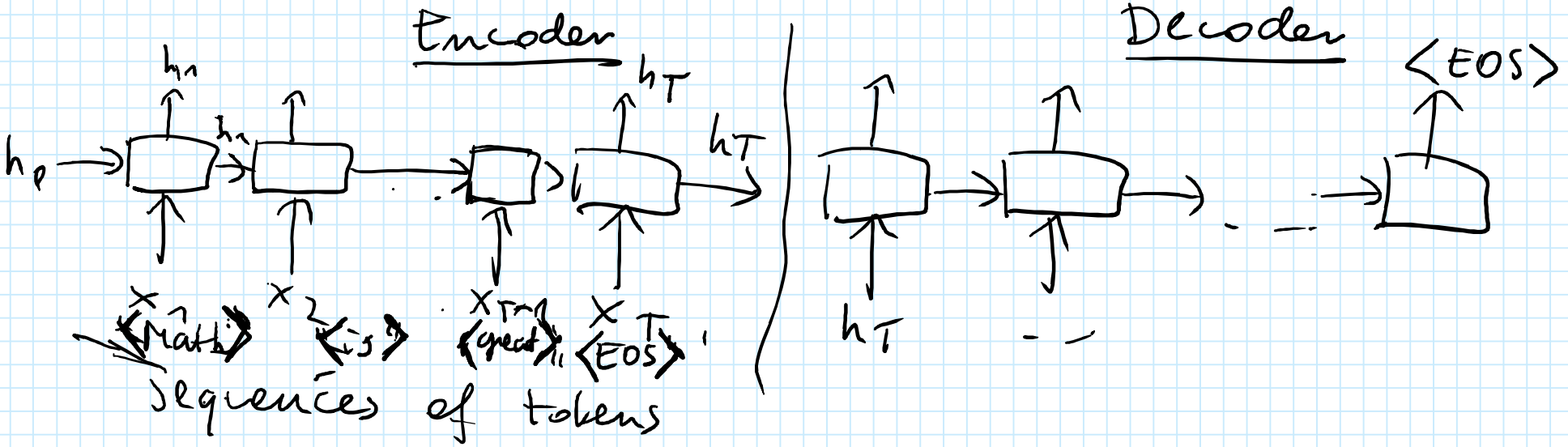
σ_g - sigmoid

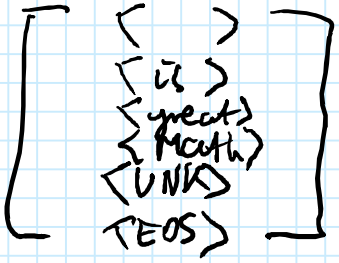
σ_h - tanh

$z_t \approx 1$ on some coordinate, then $h_t \simeq \hat{h}_t$, so we choose new candidate ($\hat{h}$) as h

$z_t \approx 0$ on some coordinate, then $h_t \simeq h_{t-1}$ so we choose „old" h

Encoder - Decoder





<UNK> - unknown word
<EOS> - end of sentence

Keras

- RepeatVector
- return_sequences = TRUE