RNNS, Transformers & Attention

SEQENTIAL DATA

Sequence Modelling Applications

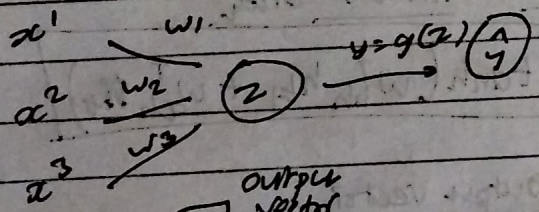
One to one	Many to one	one to many
1) Binary Classification	2) Sentiment Analysis	3) Image captioning

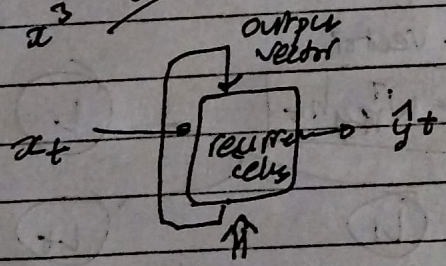
Many to many

4) Machine Translation

Neurons with Recurrence

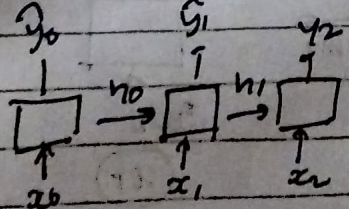
Perceptron Revisited





Handling Individual time steps

$$\hat{y}_t = f(x_t)$$ ✗

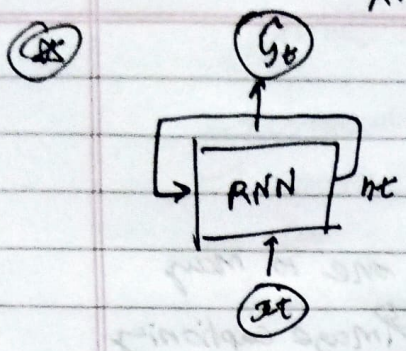


These can't relate to one another

$$\hat{y}_t = f(x_t, h_{t-1})$$ ✓

output input past memory

RNNs

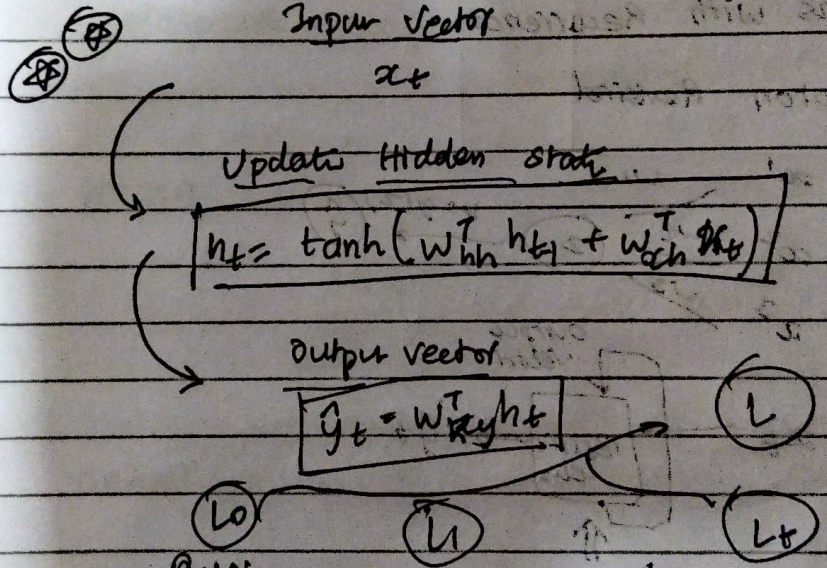


Apply a recurrence relation at every time step to process a sequence

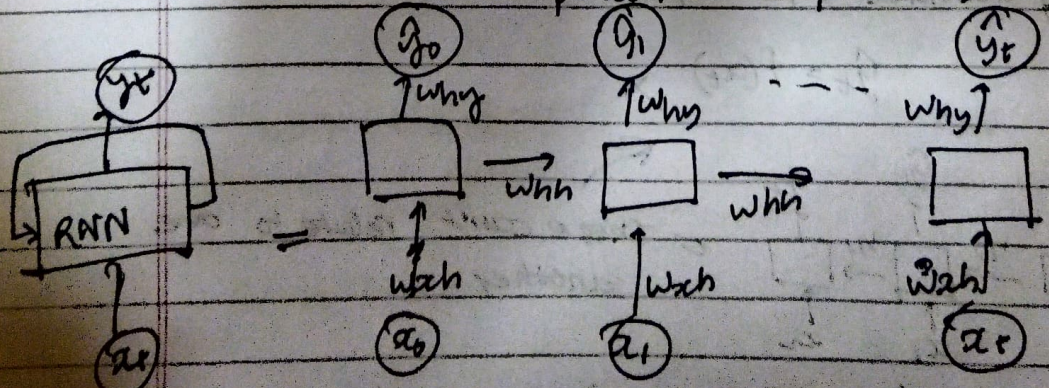
$$h_t = f_W(x_t, h_{t-1})$$

h_t: cell state; h_{t-1}: input old state; f_W: function with weights W

Same function & same set of parameters are used at every time step



RNNs: computational Graph across time



tf.keras.layers.SimpleRNN(rnn_units)

RNNs for

Design criteria - Sequence modelling

1. Handle variable-length sequences
2. Track long-term dependencies
3. Maintain ~~item~~ information about order
4. Share parameters across the sequence

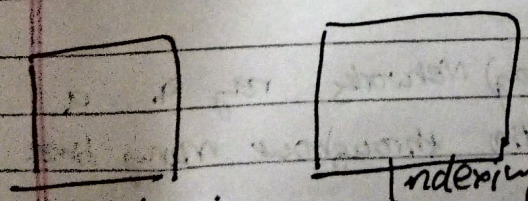
Predict the Next word

This morning I took my car for a walk.

X "deep" → □ → "learning"

Neural networks cannot interpret words

Embedding: transform indexes into vector of fixed size



Backpropagation Through Time (BPTT)

Backpropagation algorithm

1) Take the derivative of loss wrt to each parameter
2) Shift parameters in order to minimize loss

Many values > 1:	many values < 1:
exploding gradients	vanishing gradient
Gradient clipping to scale big gradients	1. Activation function 2. Weight initialization 3. Network Architecture

#1 Using RELU prevents f' from shrinking the gradient when $x > 0$

#2 Initialize weights to identity matrix
Initialize biases to zero

#3 Gated cells

LSTMs (Long Short Term Memory) Networks rely on a gated cell to track information throughout many time steps

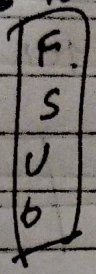
Gated LSTMs
1) Forget
2) Store
3) Update
4) Output

} able to track information throughout many timestamps

LSTMs

Key concepts

1) maintain a cell state
2) use gates to control flow of information



3) BG

3) Backpropagation through time will partially uninterrupted gradient flow

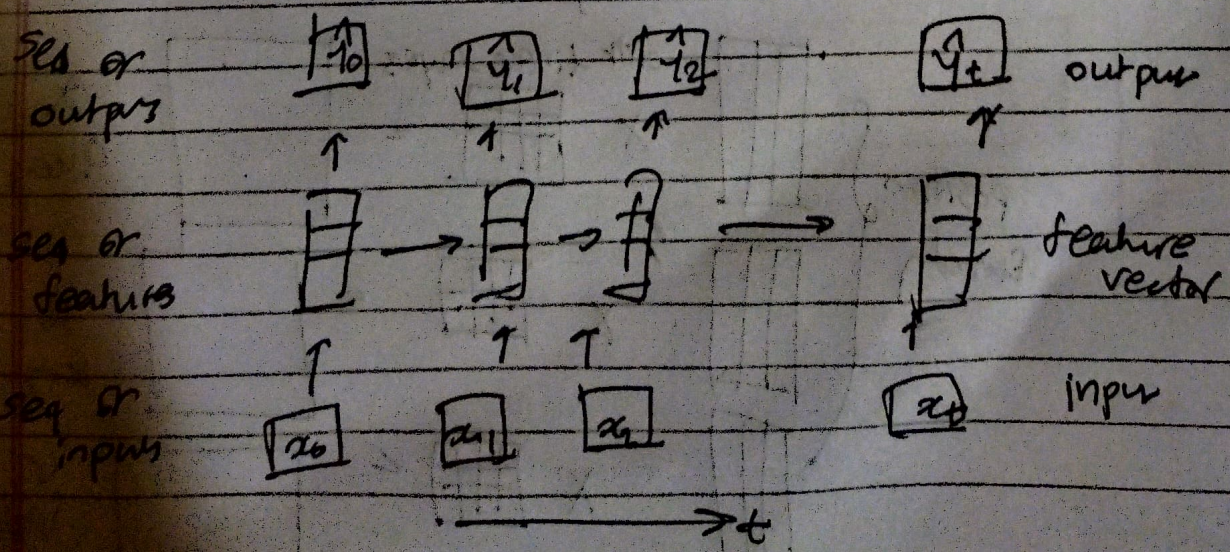
RNN Application & Limitations

1) Music Generation — Input: Sheet music
   Output: next character in sheet music

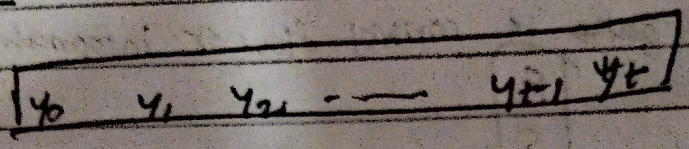
2) Sentiment classification

Limitations of RNNs

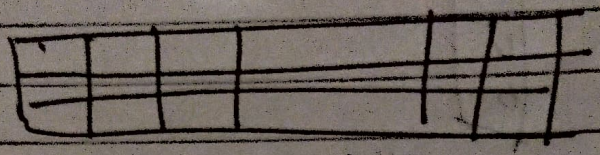
- Encoding bottleneck
- Slow, no parallelisation
- No long memory

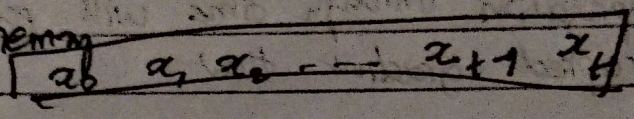


Naive Approach



✓ No recurrence
✗ Not scalable
✗ No order
✗ No long term memory





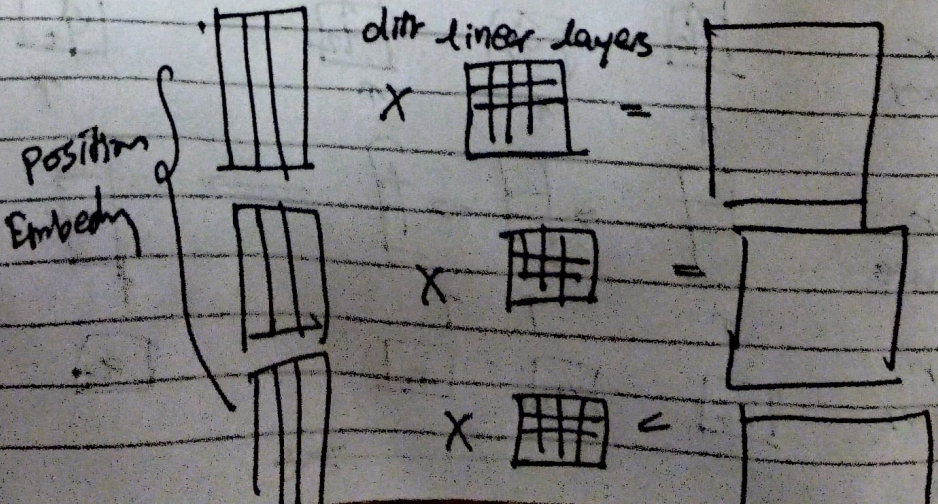
Attention

↳ foundational mechanism of transformer architecture

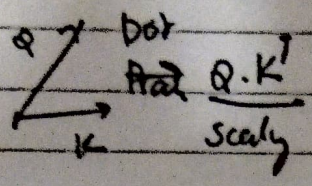
Learning self Attention with Neural Networks

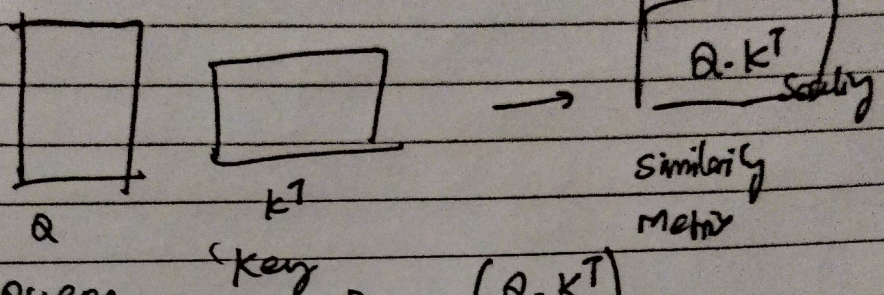
Goal: Identify and attend to most important features in input

1) Encode position information
2) Extract query, key, value for search
3) Compute attention weighting
4) Extract features with high attention



computing pairwise similarity b/w query & key





Query Key

Softmax $\left(\frac{Q \cdot K^T}{\text{scaling}}\right)$

Attention weighting

Softmax $\left(\frac{Q \cdot K^T}{\text{scaly}}\right) \cdot V = A(Q, K, V)$