LEC-2 Recurrent Neural Networks

Deep Sequence Modeling

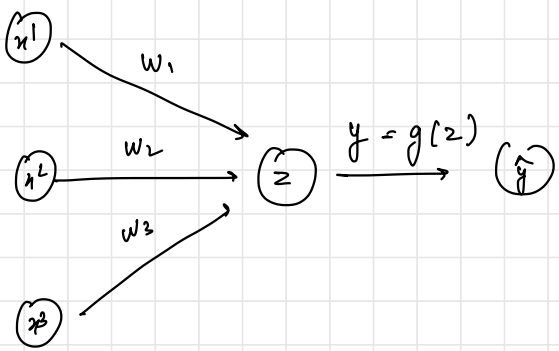
one to one binary classification pass fail

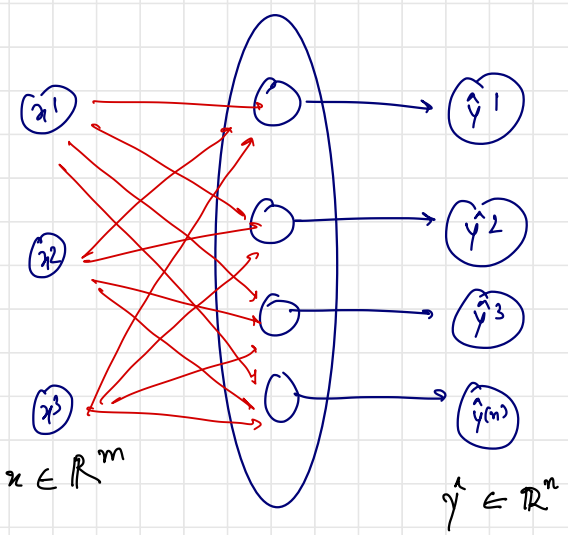
many to one — sentiment Classification

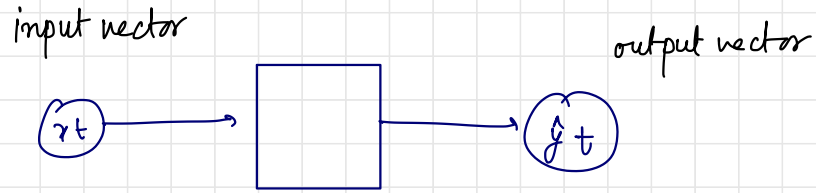
one to many — image captioning

many to many — machine translation Language

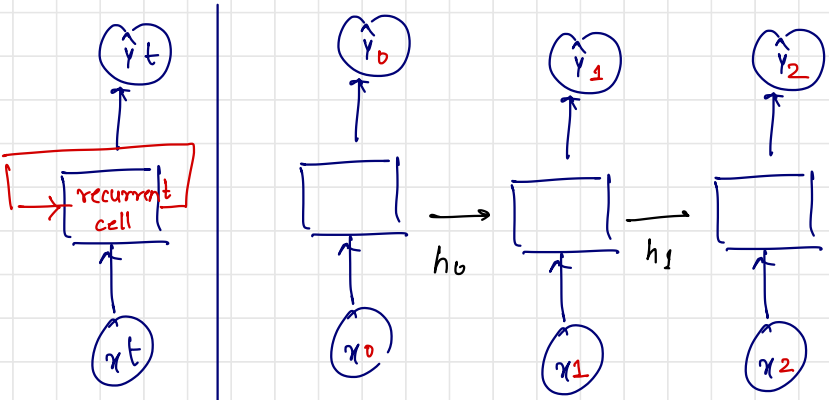
Perceptron







$x_t \in \mathbb{R}^m$ $\quad$ $\hat{y}_t \in \mathbb{R}^n$

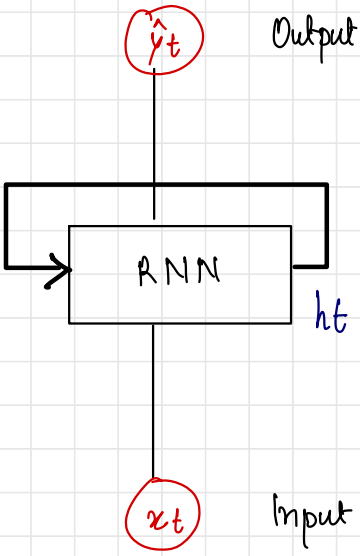


$\hat{y}_t = f(x_t)$ $\quad$ $\hat{y}_t = f(x_t, h_{t-1})$

output $\quad$ input $\quad$ past memory

internal memory $- h_t$

Recurrent Neural Networks (RNN)



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

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

- h_t: cell state
- f_W: function with weight
- x_t: input
- h_{t-1}: previous state

same function & set of parameters are used at every step

Output Vector

$$\hat{y}_t = W_{hy}^T h_t$$

Update Hidden State

$$h_t = \tanh\left(W_{hh}^T h_{t-1} + W_{xh}^T x_t\right)$$

Input Vector, x_t

Weight matrices reused every step

Design Criteria :-

1. Handle Variable length parameters
2. Track long term dependancies
3. Maintain information about order
4. Share parameters across sequence

Predict New Word

"This morning I took my cat for a walk

run, walk	dog, cat
day, sun	happy, sad

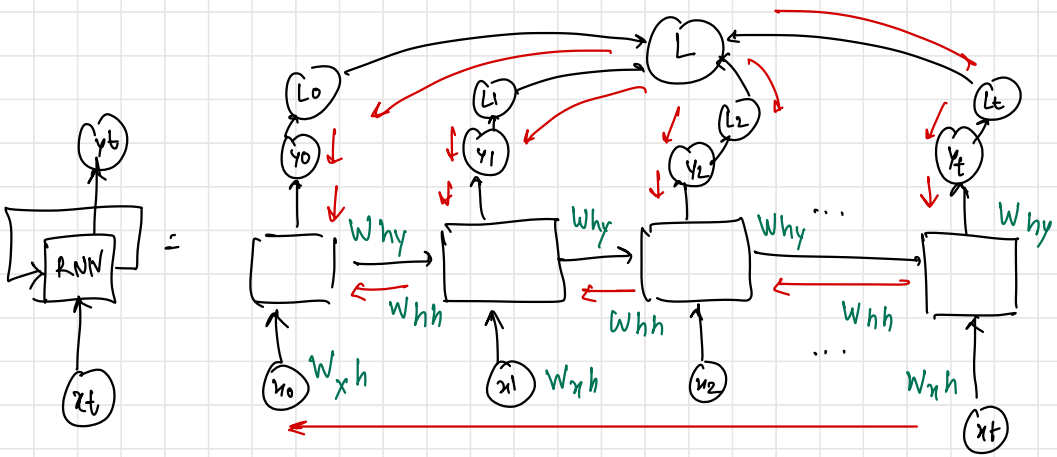
Representing language to a Neural Network
↳ into numerical inputs

↳ encoding → Embedding transform indexes into a vector of fixed size
Vocabulary → Indexing → Embedding

Back propagation Through Time (BPTT)

1. Take the derivative (gradient) of the loss with respect to each parameter.

2. Shift parameters in order to minimize loss



Problems with computing gradient w.r.t h_0 involves many factors of W_{hh} & repeated gradient computation

many values < 1: vanishing gradient | values > 1: exploding gradient

1. Activation function
2. Weight Initialization
3. Network Architecture

Trick #1 Activation function

using ReLU prevents f' from shrinking the gradients when $x > 0$

Trick #2 Parameter Initialization

Init weights to identity matrix, biases to zero

→ helps prevents weights from shrinking to zero

Gated cell — use gates to selectively add or remove info within each recurrent unit

LSTM – Long short term memory network

① forget ② Store ③ Update ④ Output

① maintain self state

② use gates to control flow of information

→ f
→ S
→ U
→ O

③ BPTT with partially uninterrupted gradient flow

Applications
→ music generation
→ sentiment classification

Limitations of ANN:-
① Encoding bottleneck
② Slow, no parallelization
③ Not long memory

Transformers

* Learning Self attention

1. Encode position information — aware encoding
2. Extract query, key, value for search
   ↳ transform by matrix multiplication with linear layers
3. Compute Attention weighting
   Attention score : compute pairwise similarity between each query and key

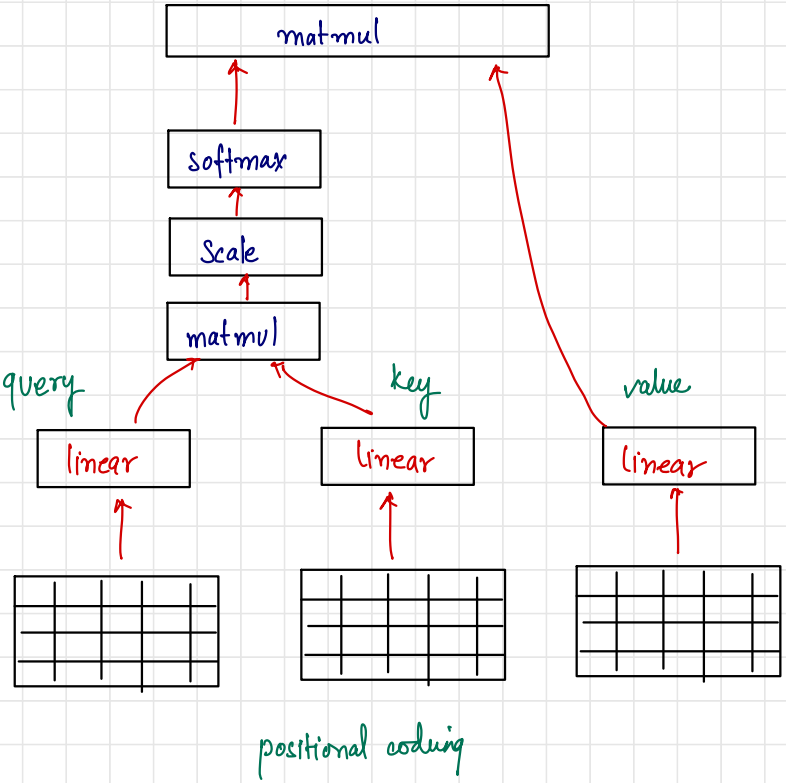
⟹ Dot product

$\frac{Q \times K^T}{\text{scaling}}$ } similarity metric

4. Softmax (squashes value to fall between 0 & 1)

$$\text{softmax}\left(\frac{Q \cdot K^T}{\text{scaling}}\right) \times V = A(Q, K, V)$$ query, key, value

V: value matrix


mat-mul
softmax
Scale
matmul
query
key
value
linear
linear
linear
positional coding