PAGE
DATE

*(Course 5) Sequence models
(week 1)

→ why sequence models

- models like RNNs have transformed speech recognition NLP & others
- is speech recognition we give input X (wwww) → "Hi" (y)
- Music generation is another example with seq data. The input maybe (empty set, single int. refering to genre) output is sequence
- sentiment classification input phrase → classify it
- DNA Seq analysis from given seq. label which part is protein
- Machine translation moi → Me
- Video activity recognition input (seq. of frames) → running
- Name entity recognition omar is.... → omar.
  it can also be used find people's names, companies names times, locations, countries names, currency names.

→ Notation

- X: omar ($x^{<1>}$) and ($x^{<2>}$) Ahmed ($x^{<3>}$) are g--- T_x = length
  y: 1 ($y^{<1>}$) 0 ($y^{<2>}$) 1 0 0 --- T_y = length
- to represent a word in sentence first we come up with a vocabulary (or) dictionary. [omar, and, ahmed] → 1, 2, 3, 4

then use one-hot representations to represent each word

DATE

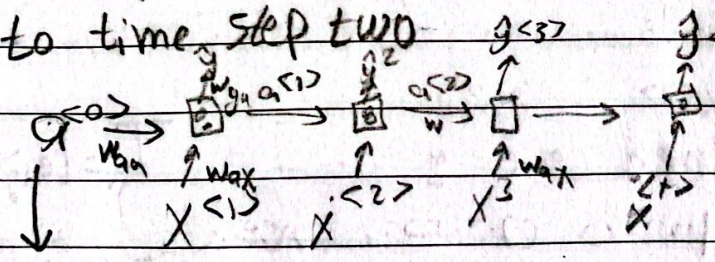
Recurrent Neural Network model.

- We will talk about how you can build models, built NN to learn the mapping from X → y → Bec RNN handle seq. data & have memory that capture dependencies bet future feat
- why not standard NN.
  ① Doesn't lean features across different position of text
  ② inputs, outputs can be different lengths

- in RNN if we are reading a sentence from left to right the first word we will read is $X^{<1>}$ then feed it to NN then the NN try to predict if it is a person's name or not

$$X^{<1>} \longrightarrow \boxed{\quad} \longrightarrow \hat{y}^{<1>}$$

- what RNN does is when it goes on the read the second word in the sentence Instead of just predicting y^2 using X^2 only it also gets to input some information from that time step one so in particular, de activation value from time step one is passed on to time step two



vector of zeros

- one weakness of RNN is that it uses the info that is earlier in the seq. to make a prediction, but not use info later in the sequence

- $a^{<1>} = g(W_{aa}\, a^{<0>} + W_{ax} X^{<1>} + b_a)$ ← Tanh/Relu
  $y^{<1>} = g(W_{ya}\, a^{<1>} + b_y)$ ← sigmoid/Softmax

- $a^{<t>} = g(W_{aa}\, a^{<t-1>} + W_{ax} X^{<t>} + b_a)$
  $\hat{y}^{<t>} = g(W_{ya}\, a^{<t>} + b_y)$

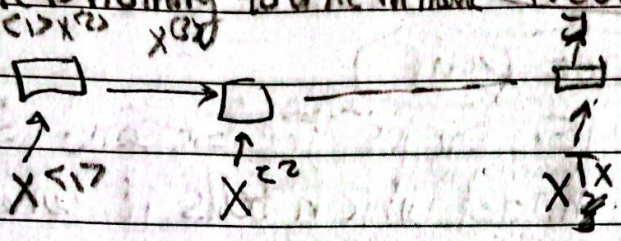
PAGE
DATE

→ Backpropagation Through time

→ Different types of RNN

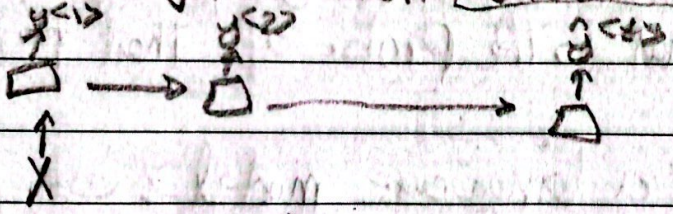
- Sentiment Classification: y = text
  - y: num from 1-5 or 0/1
  - text: sequence

ex: There is nothing to like in this movie (+ve or -ve review)



many to one

- Music generation (one to many)



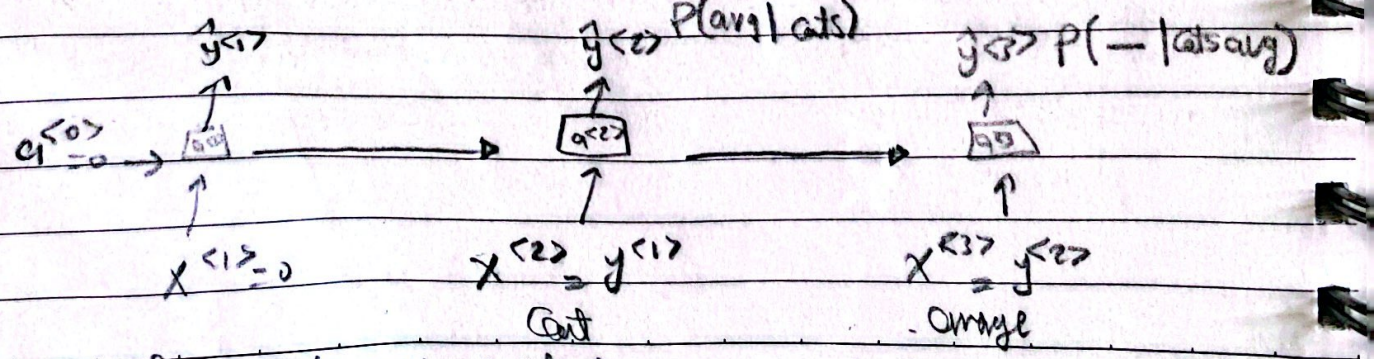
- Machine translation: many to many but input & output could be different length.

it has encoder which take as input the frenchsentence & then the decoder which having read in the sentence & outputs the translation into a different language

PAGE
DATE

→ language model & sequence generation.

- language modeling is one of the most basic & important tasks in n/p. we will talk about building a language model using an RNN
- What is lang. model? lets say we are building a speech recognition system & you hear the sentence
  - The apple & pear salad was delicious
  - not pair
- the way SR sys. picks the second sentence is by using a language model which tells it what is the prob. of either of these two sentences.
- so what lang. model does is given any sentence its job is to tell you what is the prob. of that particular sentence
- How do you build a language model? using RNN
- first need a training set corpus of english text or whatever language
  - in n/p means tons of english sentences
- ex a sentence in your training set as follows:
  Cats Average 15 hours of sleep a day
  ① tokenize the sentence (form a vocab & map each word to one hot vector
  ② build RNN to model the chance of these diff. sequences



- This RNN learns to predict one word at time going from left to right.

PAGE
DATE

→ Sampling novel sequences ✗

- After training a sequence model, one of the ways you can informally get a sence of what is learned is to have a sample novel sequences

- using a character level language model has some pros & cons one is that you don't ever have to worry about unknown word tokens ; in a character level language model is able to assign a sequence like (mau), ex. word if mau is not in the vocabulary for the word level lang. model you just have to assign it the Unknown word token, But the main disadvantage of the character level lang. model is that you end up with much longer seq.

→ Vanishing Gradient with RNNs

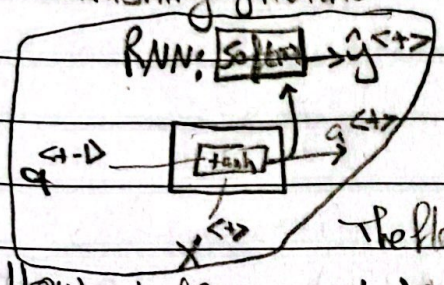
- for ex: the cat, which already ate ----, was full
  "Cats, " " " , were full
  . this is example that language can have long-term dependency where earlier worded can affect what needs to come later .it might be difficult to get a NN to realize that it need to memorize a singular noun or plural noun so as to that later in seq. it can generate either was or were

- if derivitives do explode we apply Gradient clipping

PAGE
DATE

→ Gated recurrent unit (GRU)

- It is modification to RNN hidden layer that makes it much better at capturing long-range connections & helps alon with vanishing gradient.
- The GRU have © → memory cell that provide memory that stores info to remember it over long time



- Γ_u → Gate bet 0-1 usig sigmoid, It Controls The flow of info. Thrugh The network. Updategate (decides How much of prev. memory to Keep & how much to update with new info). Reset gate (decides how much to forget before updating the hidden state with new input)

→ Long Short term memory (LSTM)

- More Powerfull than GRU and allows us to learn long term seq.
- LSTM is type of RNN designed to Solve vanishing gradient and learning over long sequences
- LSTM introduces a memory cell with three gates
  - input gate: decides how much new info to store in memory cell
  - forget gate: " " " info " forget " " "
  - output gate: Controls the output based on the memory "

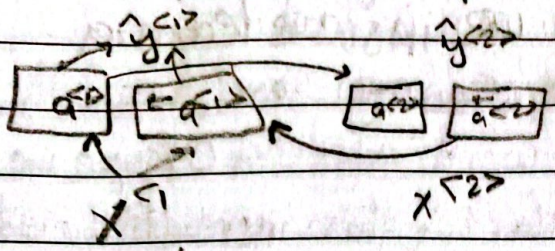
update ←

- memory cell is Core Component that store the LT info; Gates are mechanism that Controls the flow of info to & out the memory cell & The gates used AF (sigmoid) to decide how much the info to allow thrugh

PAGE
DATE

→ Bidirectional RNN

. allows us at the point in time to take info from both earlier & later in sequence



. the disadvantages that we need the entire seq. of data before you can make predictions anywhere.

. In BRNN, one layer processes the input from start to end (forward) & other layer processes the sequence from end to start (back) & at each time step, the outputs from the forward & backward are combined to form the final output.

. GRU, LSTM, SimpleRNN are types of recurrent NN used in language models but they are not considered seq2seq models.