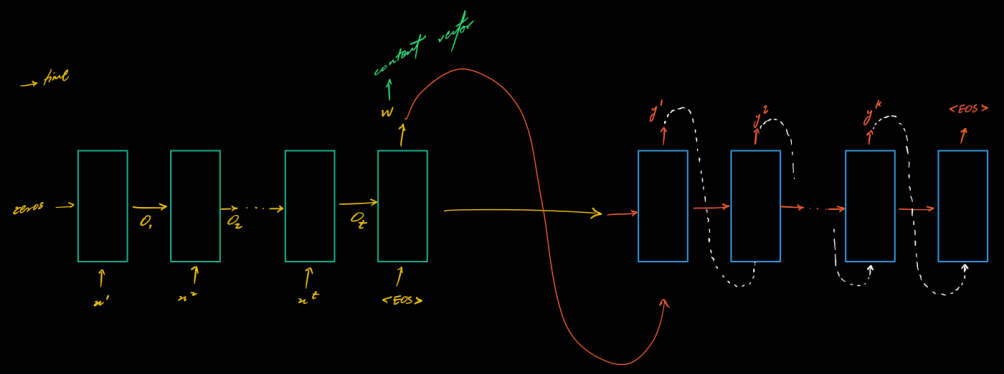
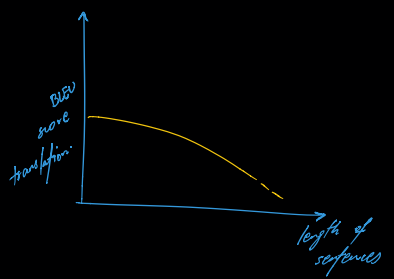


one small context vector is not enough to capture complete essence of a long sentence.



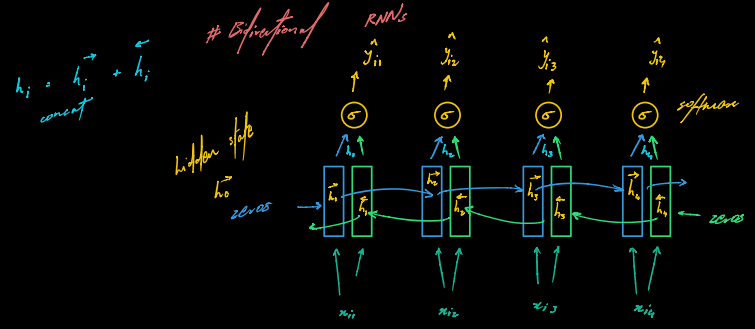
what humans do:

① read some words -- translate

② shift attention to next few words -- translate.

This is NOT what seq2seq does

That's why it doesn't scale up.



It doesn't matter what these boxes are, they can be RNN, LSTM, GRU, etc whatever works

① Forward pass as usual.

② pass stuff backwards and collect predictions for that too

③ Softmax over both predictions.

④ Get better than only doing forward pass.

In attention models we mostly never use vanilla RNNs

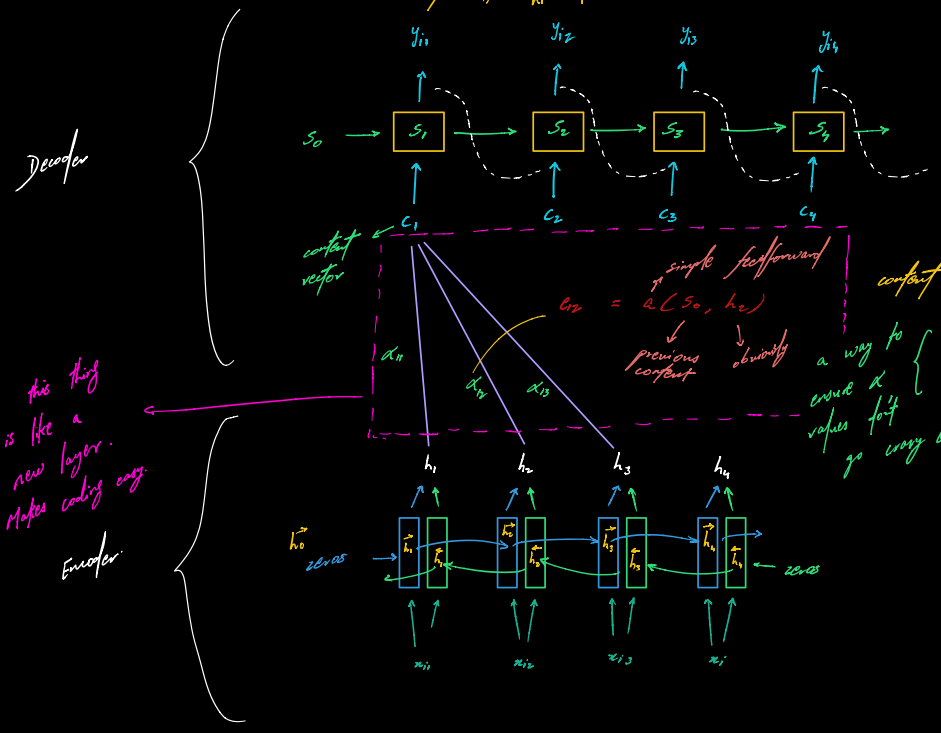
↳ why would we?

and everything is bidirectional. no vanilla gradient fucking stuff

when we have LSTMs & GRUs

Reference Paper & notation:

Neural Machine Translation:
- Yoshua Bengio et. al.



y_{i1} will give different to different h_i values of input. Not all h_i values are taken, it depends on T_x

① context vector is a α_{ij} weighted average of all encoder outputs.

② Weights are for each decoder cell.

$c_2 = a(s_0, h_2)$ → simple feedforward
previous context, obviously

context vector $(c_1) = \sum_{i=1}^{T_x} h_i \alpha_{1i}$ — hyper-parameter

a way to ensure α values don't go crazy & stay positive: $\left\{ \sum_{i=1}^{T_x} \alpha_{ii} = 1 \;\&\; \alpha_{ij} \geq 0 \right.$

this thing is like a new layer. Makes coding easy.

Why bidirectional encoder?
↳ output word could depend on words before & after it

Why unidirectional decoder?
→ word just generating a sequence of output + we have attention. So no need for bi-directional.

$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$

This will keep it ≥ 0 & sum $= 1$
This is like softmax

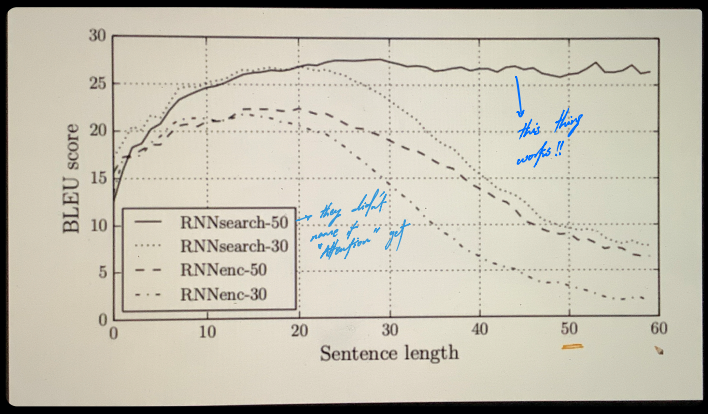
$e_{ij} = a(s_{i-1}, h_j)$

they didn't know what f^n to use. All models are f^n approximators. So let's put a (NN) there. Let the model decide it ...
↓
simple 1/2 layer NN

Make architecture differentiable & let it converge

↓

given large dataset, this thing will be cool → doesn't matter how hard it is or complex.

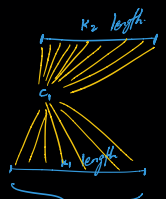


Drawback

length of output

Time Complexity = $O(K_1 \cdot K_2)$

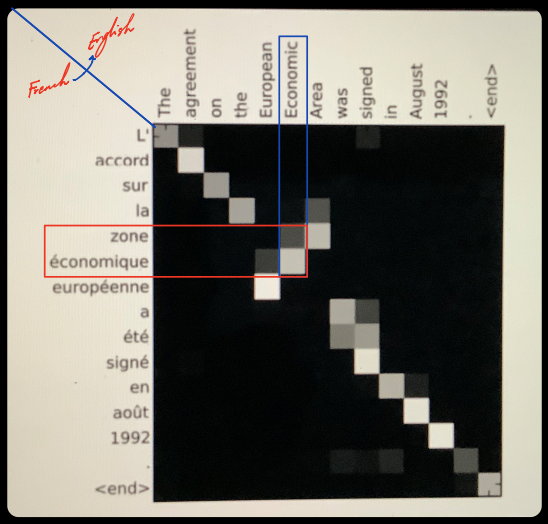
↓ length of input



T_{ex} → if T_{ex} is large enough → c_1 will depend on all K_1 inputs & K_2 if depend on c_1, then $K_1 \cdot K_2$

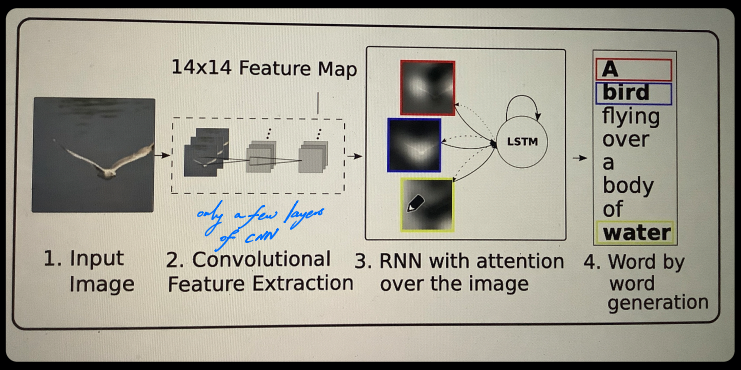
#Visualizing α_{ij}'s → helps debug the NN

$\alpha_{ij} \rightarrow 1$
color → white



#understanding what the funk is happening inside the black box.

Show Attend & Tell. Yoshua Bengio et al. (2015)



only a few layers of CNN

Encoder → CNN
Decoder → same

Divide image into sections
L d-dim vectors for each image region.

They use Vects for the image. FC layer is not used we the 4th layer.
14 × 14 × 512 → after 4th Vects layer

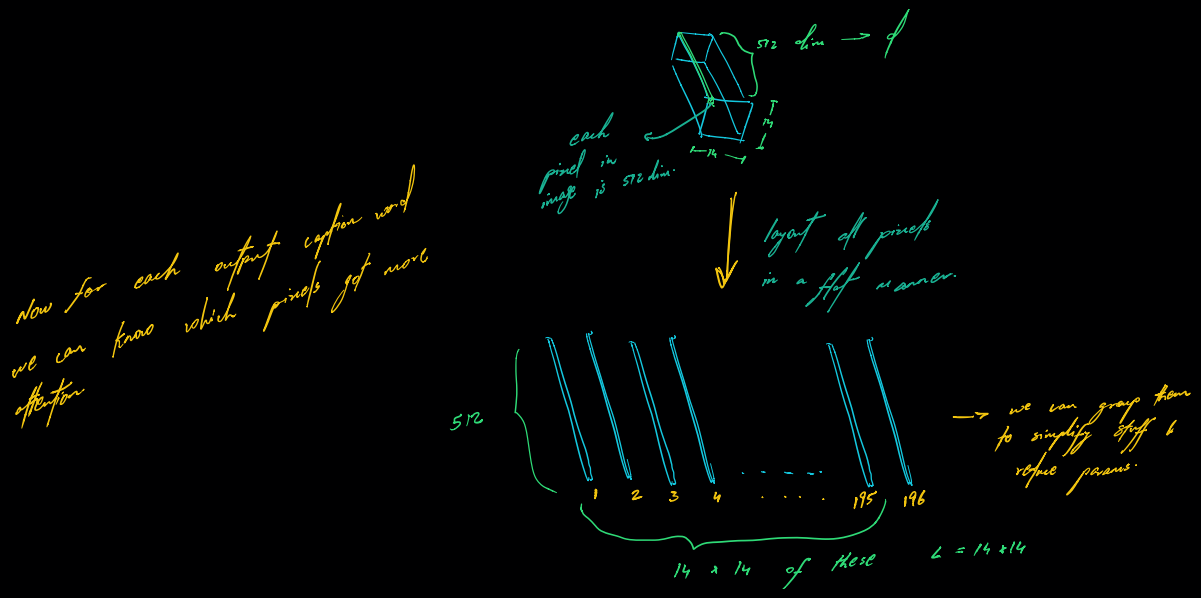
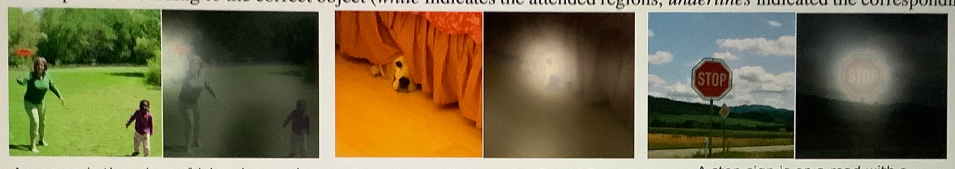


Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding

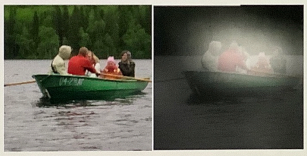


A woman is throwing a frisbee in a park.

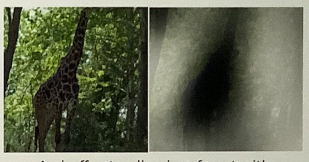
A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.

The model implicitly learns where to look based on the output categorical.

Implementation

In Keras, it's as simple as adding a new layer → just like conv layers.

Blog → better than official docs → more complete & has viz simulation.

Attention in Deep Networks with Keras

By Thushan Ganegedara

Markov Chain, LDA, Topic modelling is old school. Wake up, boomer...

This course covers the most imp. most relevant stuff.

Text, Image & Video is important for Google That's why all the good research in these areas by Google.

I already know answers to the QnA session, so not taking much notes at this point

Tx → just make it very big 50–60 & usually it works fine.