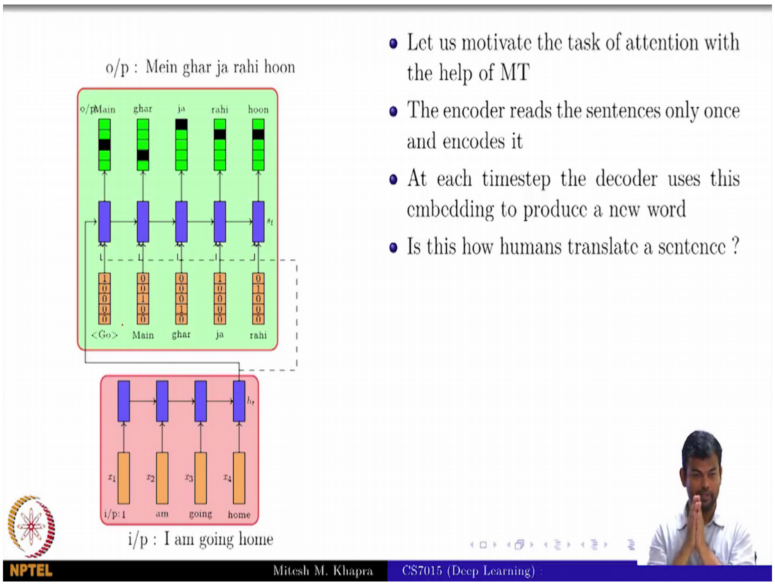
Deep Learning
Prof. Mitesh M. Khapra
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 114
Attention Mechanism

So, let us go on to the next module which is Attention Mechanism.

(Refer Slide Time: 00:14)



- Let us motivate the task of attention with the help of MT
- The encoder reads the sentences only once and encodes it
- At each timestep the decoder uses this embedding to produce a new word
- Is this how humans translate a sentence ?

Mitesh M. Khapra CS7015 (Deep Learning)

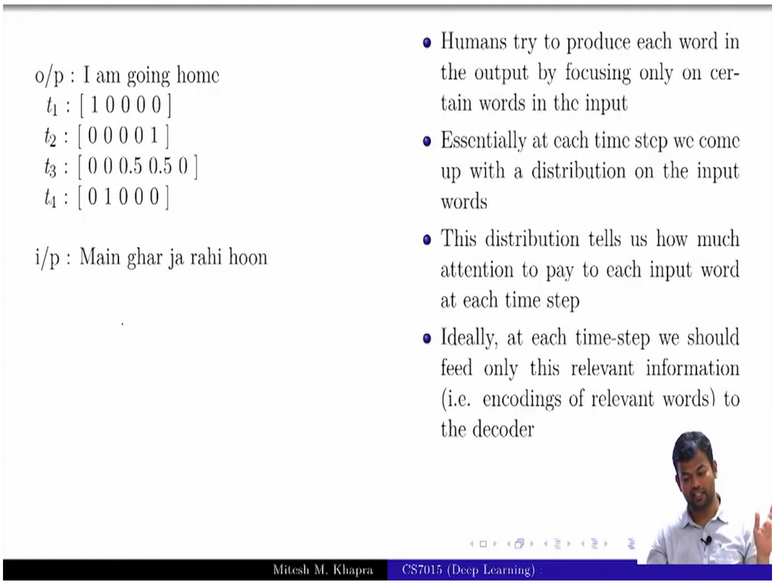
So, let us motivate not the task of attention, let us motivate attention mechanisms with the help of machine translation, ok. So, what is happening in the models that we have seen so far? The current model that we saw for machine translation; by the way all the models that I have shown you so far are wrong or rather incomplete we will complete all of them, right and that is where attention fits in, ok, that was for the camera.

A encoder reads the sentence and its computes the encoding once, right we read the entire sentence and be encoded it. And then we have these two options either the pass the encoding at the 0 time step or pass this encoding at every time step. Is this how humans translate a sentence? What is the human analogy for this? You have read the sentence once done and now we are going to remember this entire thing throughout and then translate. Imagine if you doing this for sentences which have 25 words which is a typical Wikipedia sentence. What is wrong with this? We have read the input ones and we have encoded it, what is likely to happen you will forget something you are going to lose information. Not just that is the entire sentence important that every time step?

Student: (Refer Time: 01:28).

Only certain words are important. You see this conceptually something wrong that we are doing here is saying, ok. I have encoded the sentence and then start decoding from there that is the conceptual error that we are making. So, let us see how humans actually translate it, right.

(Refer Slide Time: 01:39)



o/p : I am going home

t_1 : [1 0 0 0 0]

t_2 : [0 0 0 0 1]

t_3 : [0 0 0.5 0.5 0]

t_4 : [0 1 0 0 0]

i/p : Main ghar ja rahi hoon

- Humans try to produce each word in the output by focusing only on certain words in the input
- Essentially at each time step we come up with a distribution on the input words
- This distribution tells us how much attention to pay to each input word at each time step
- Ideally, at each time-step we should feed only this relevant information (i.e. encodings of relevant words) to the decoder

So, when producing one word in the output suppose my input is the Hindi sentence and I have the output sentence. When I am trying to produce the first word I actually compute this probability distribution which tells me which of the input words that I need to focus on. At this point it is, if I do not know what is the translation for ghar or ja or rahi or hoon, as long as I know the translation for main I am done because that is the word which I first need to produce there, right.

So, I am going to say that at this point I only need to pay attention with the first word in the input and I can ignore everything else. What about the second time step? I just need to focus on the last word. What about the third time step? Is it always going to be that I only need to focus on one word at a time?

Student: No.

What about the third time step?

Student: [FL].

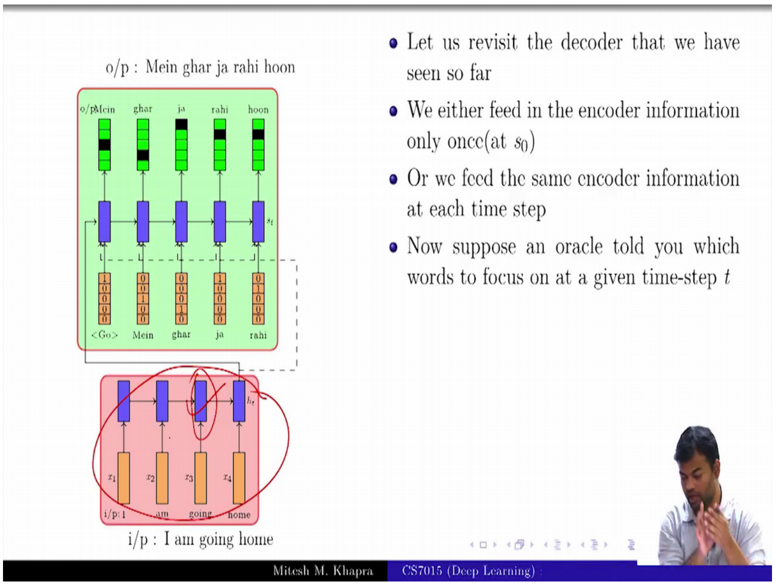
I am sorry I am assuming everyone understands Hindi, but I think that is this is small sentence I can assume that. What will you focus on?

Student: Ja rahi.

Ja and rahi, right. You want to focus on both these things and not an anything else and what about the next one? Hoon, right; so, just on ghar and not an anything else is this what the model encoder decoder model is doing? What is it doing actually? The every time step is focusing on the entire sentence because that is the encoding that your feeding to every time step, that is the problem that we need to correct. We need to learn to pay attention to certain important parts of the sentence. Is the setup clear to all of you? Is the motivation fine? Not your motivation layers; is the motivation for this fine or not, ok.

The distribution actually tells us how much attention to pay to each input word at each time step, and ideally at each time step we should face pay attention to only certain words in the input.

(Refer Slide Time: 03:22)



- Let us revisit the decoder that we have seen so far
- We either feed in the encoder information only once(at s_0)
- Or we feed the same encoder information at each time step
- Now suppose an oracle told you which words to focus on at a given time-step t

Mitesh M. Khapra CS7015 (Deep Learning)

So, let us revisit the decoder that we have seen so far. This is what the decoder looks like. In fact, I also have the encoder there. Now suppose, sorry. So, currently what we are doing is we are either feeding s 0 at the I mean we have either feeding the input embedding or the encoder embedding at time step 0 or at every time step; the suppose there was an oracle which told us exactly which are the words important at time step t, right. So, in our example at time step 3 suppose it told us that the word going is important, actually we need to flip the input and output here also, right. But you can still understand, right.

So, I am saying at time step 3 certain words are important and suppose a oracle actually told us that these are the words which are important. What would you do? Assume that you have already run the encoder, what will you do now? And say someone told you that only this word is important word why weighted I am just saying binary weigths, right only this word is important. What would you do ideally?

Student: (Refer Time: 04:24).

Just feed this blue vector to the decoder and do not feed everything else, does not make sense. Suppose I told you that two words were important send those two words but how concatenate, but now at certain time steps 4 words will be important, and you cannot concatenate 4 words, right because then the dimensions will change. So, what do you do?

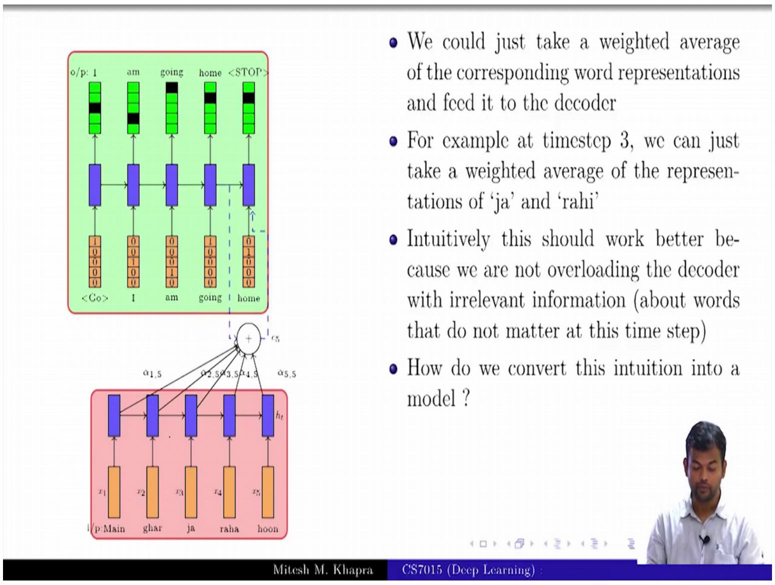
Student: (Refer Time: 04:47).

A weighted.

Student: (Refer Time: 04:49).

Weighted some of the important inputs is that make sense. At time stamp 3 we saw that ja was 0.5 important and rahi was 0.5 important, just a weighted combination of those two blue vectors and feed that to the decoder. So, you are not changing the dimensions at each time stamp because the blue vectors have the same dimension, I am just taking a weighted combination of those I am going to give you the same dimension; does that make sense, ok.

(Refer Slide Time: 05:16)



- We could just take a weighted average of the corresponding word representations and feed it to the decoder
- For example at timestep 3, we can just take a weighted average of the representations of 'ja' and 'rahi'
- Intuitively this should work better because we are not overloading the decoder with irrelevant information (about words that do not matter at this time step)
- How do we convert this intuition into a model ?

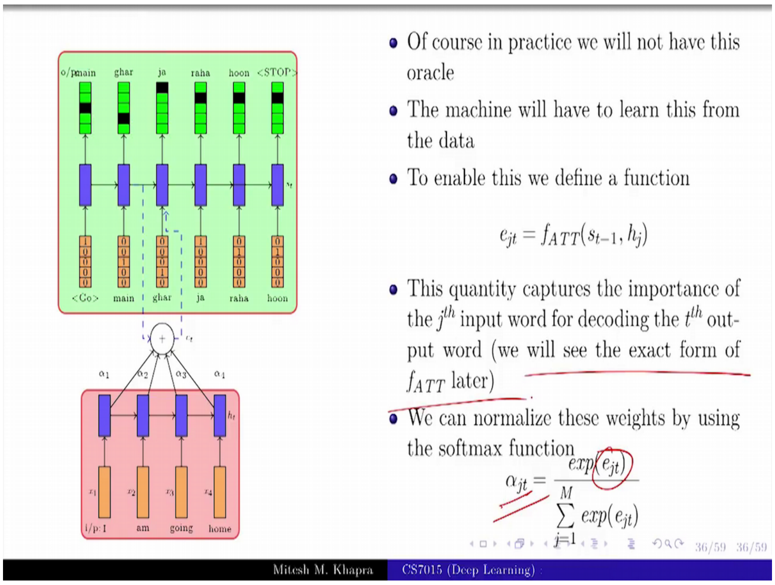
Mitesh M. Khapra CS7015 (Deep Learning)

So, in fact what I am saying is that, I could just take a weighted combination of all the blue vectors that I have at the encoder and the weights of this weighted combination, right now I am assuming that someone oracle has given me is that, ok. If I had his weights does this makes more sense then having the vanilla encoder decoder model, everyone agrees with that, ok.

Now, the question of course, us who is going to give us these weights, we will come back to that later, but at least given the weights this make sense. So, at every time step they just going to focus on the words which are actually important, just take a weighted combination of those words and we will just feed that to the decoder. And intuitively this should work better because unlike before where we were overloading the decoder with the entire sentence remember 25 words, 30 words, entire sentence was being passed to the decoder now you are just overloading it with the amount of information that it actually needs to produce that particular word. Hence, intuitively this should work better, right, ok.

Now, how do you convert this intuition into a model?

(Refer Slide Time: 06:19)



- Of course in practice we will not have this oracle
- The machine will have to learn this from the data
- To enable this we define a function

$$e_{jt} = f_{ATT}(s_{t-1}, h_j)$$

- This quantity captures the importance of the j^{th} input word for decoding the t^{th} output word (we will see the exact form of f_{ATT} later)
- We can normalize these weights by using the softmax function

$$\alpha_{jt} = \frac{exp(e_{jt})}{\sum_{j=1}^{M} exp(e_{jt})}$$

Mitesh M. Khapra CS7015 (Deep Learning) 36/59

In practice of course, there is no angel who is going to coming give as these weights is no oracle. The machine will have to learn this from the data. Whenever you need to learn something you need to.

Student: (Refer Time: 06:34).

Introduce parameters. So, I am going to now introduce a parametric form for the from the figure which thing for those of we cannot see these are alpha 1, alpha 2 and so on. So, now, from the figure we are going to introduce a parametric form for.

Student: Alphas.

For the alphas, ok. So, I am going to introduce I will come to alpha. But and what you think this weight should depend on. What I am trying to say is that at the tth time step of the decoder, so this is e j t at the tth time step of the decoder I want to find out how important is the jth word in the input. That is exactly what I am interested in at every time step of the decoder of all the input words I want to see which of them is the most important, right. So, this is the quantity that I am interested is, how important is the jth input word at the tth time step. This should depend on what? What should it be a function of?

For one it should depend on what that word is, right the other is should depend on what has happened in the decoder so far. What is the decoder produce? So, what is the input and what is the decoder state at so far, right. So, as the decoder has already decoded the word ghar or home, it does not need to look back at home, right. That is why need to know what is the state of the decoder. What captures the state of the decoder at time step t? H t. And what is the state of all the words that we have? It is captured by what? The h j's, right, this is h 1, h 2, h 3, h 4. Does that make sense? How many of you have fine at this point? Please raise your hands high above, ok. How many of you have questions, please ask specific questions if you have a question.

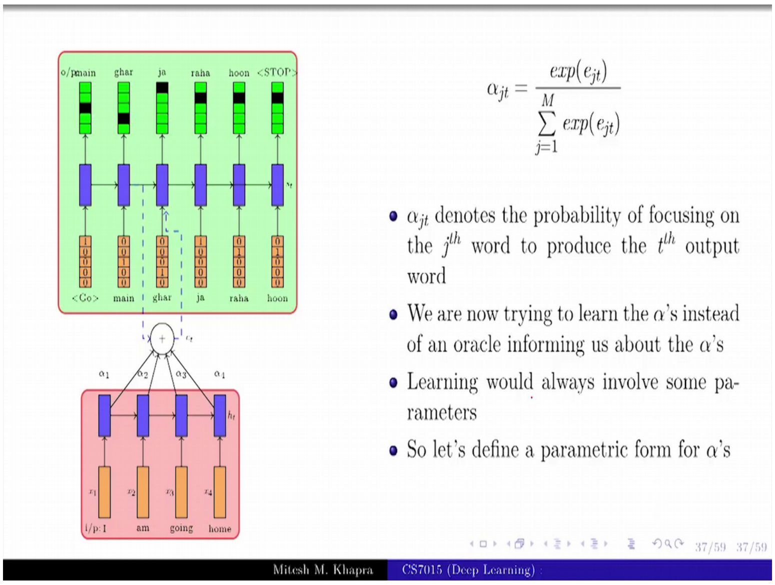
All I am saying is a couple of things one is at every time step instead of the oracle giving me these weights I want to learn these weights. Whenever I want to learn something I have to introduce a parametric form and then I learn those parameters, ok. Now, what is the quantity that I am interested in? I am interested in this for all the input words, for the jth input word I am interested in knowing how important it is for the tth time step. There are several ways I could write this function, I am saying that the two things that are important is one what is this jth word which is captured by h j right and what is the state of the decoder up to this time step which is captured by s t minus 1.

You could think of various other equations. At this point I am fine if you by the intuition that this quantity should indeed depend on these two terms it should depend on what has happened in the decoder so far and what is my current word actually look like. How many of your fine with that please raise your hands up and high? Ok. Now also the other thing that I want is that across all the input words this should actually sum to 1, right. I just want a weighted combination I do not want arbitrary weights it just like taking a probability distribution over what which word is important by how much. So, if I have this e j t how will I convert it to a probability distribution?

Student: Softmax.

Softmax. So, I will just compute the alpha j's as a softmax of e j t, e j's is that fine; did not get this, ok. Now, we have still not seen what the exact form of attention is, of the f attention function is.

(Refer Slide Time: 10:03)



$$\alpha_{jt} = \frac{exp(e_{jt})}{\sum_{j=1}^{M} exp(e_{jt})}$$

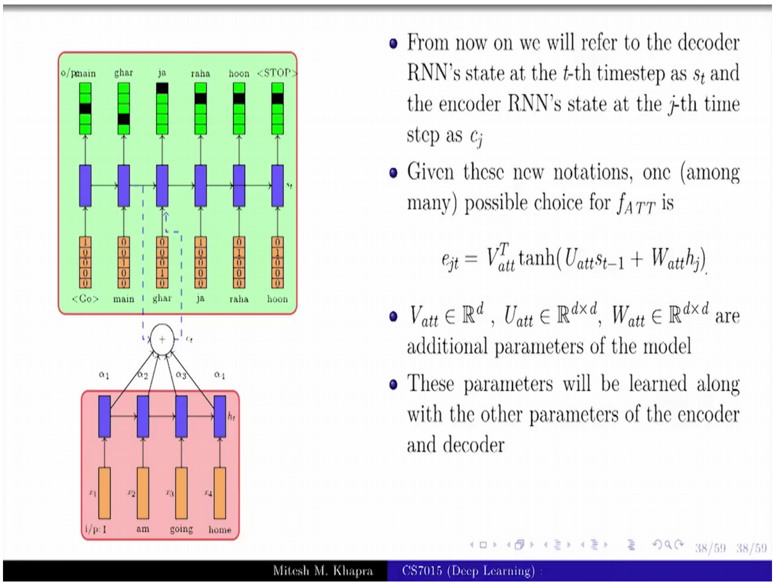
- α_{jt} denotes the probability of focusing on the j^{th} word to produce the t^{th} output word
- We are now trying to learn the α's instead of an oracle informing us about the α's
- Learning would always involve some parameters
- So let's define a parametric form for α's

37/59 37/59

Mitesh M. Khapra CS7015 (Deep Learning) :

So, this is what the equation for the alpha j t is that we had an alpha j actually denotes the probability of focusing on the jth word at the tth time step, ok. Now, we are now trying to learn these alphas instead of an oracle telling us what these alphas are. So, learning is always going to involve some parameters. So, let us define a parametric form for alphas and just a couple of notations.

(Refer Slide Time: 10:25)



- From now on we will refer to the decoder RNN's state at the t-th timestep as s_t and the encoder RNN's state at the j-th time step as c_j
- Given these new notations, one (among many) possible choice for f_{ATT} is

$$e_{jt} = V_{att}^{T} \tanh(U_{att}s_{t-1} + W_{att}h_j)$$

- $V_{att} \in \mathbb{R}^d$, $U_{att} \in \mathbb{R}^{d\times d}$, $W_{att} \in \mathbb{R}^{d\times d}$ are additional parameters of the model
- These parameters will be learned along with the other parameters of the encoder and decoder

38/59 38/59

Mitesh M. Khapra CS7015 (Deep Learning) :

So, from now on we would not change this we are going to refer to the decoder state as s t and the encoder state as s h j, ok. So, these blue vectors are s s and these blue vectors are h s, ok.

Given these new notations one among many many possible choices for f attention is the following. I wanted it to depend on the current decoder state I am making a dependent on the current decoder state but I am also adding a parameter in front of it, right. I also wanted to make a dependent on h j I am making a dependent on that I am also adding a parameter here.

Why do I need this parameter? What is the dimension of this? Let us assume this is also what is the dimension of this. Remember after multiplying with u attention and after multiplying with w attention the two vectors should be addable, is that fine. Something cross d what about this same thing cross d, good, ok. So, let us call that same thing as d 1. What is this output then? The tanh output is vector scalar matrix vector of size.

Student: (Refer Time: 11:38).

You said matrix or scalar. It is.

Ask R raise to d 1. What is the quantity on the left hand side? Vector? Scalar? Matrix? Vector; even though it has two indices it is a vector. What is this quantity capturing? At time step t what is the importance of the jth input that is a I will keep asking till everyone replies that is a?

Student: Scalar.

Scalar, ok. Now, you have scalar equal to something multiplied by R raised to d 1. So, why do you need this something?

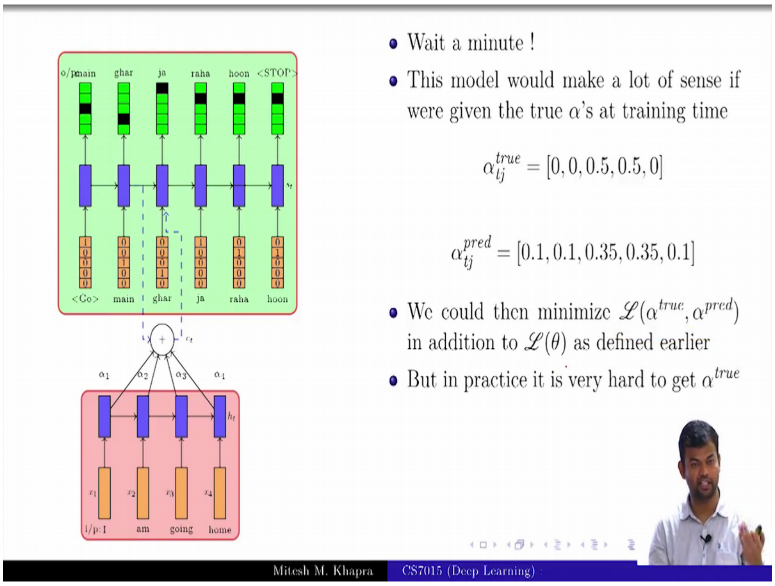
Student: (Refer Time: 12:16).

So, what is the dimension of that going to be?

Student: R raised to d 1.

R raised to d 1. So, that is the dot product. So, you see why we have these parameters, ok. So, what we have done is made it dependent on s t 1 and h j, and also made sure that the output is a scalar that is what these 3 parameters are doing, ok. And these parameters will be learned along with all the other parameters of the encoder and the decoder.

(Refer Slide Time: 12:39)



- Wait a minute !
- This model would make a lot of sense if were given the true α's at training time

$$\alpha_{tj}^{true} = [0, 0, 0.5, 0.5, 0]$$

$$\alpha_{tj}^{pred} = [0.1, 0.1, 0.35, 0.35, 0.1]$$

- We could then minimize $\mathscr{L}(\alpha^{true}, \alpha^{pred})$ in addition to $\mathscr{L}(\theta)$ as defined earlier
- But in practice it is very hard to get α^{true}

Mitesh M. Khapra CS7015 (Deep Learning)

So, now this is all fine. You would actually someone had given me the true alpha j's and I had predicted this alpha j's to that fancy equation which I just showed you and then I added a dash function, softmax that is the [laughter] safest choice in this course. I want to learn something. So, what do I what should I add?

Student: Loss function.

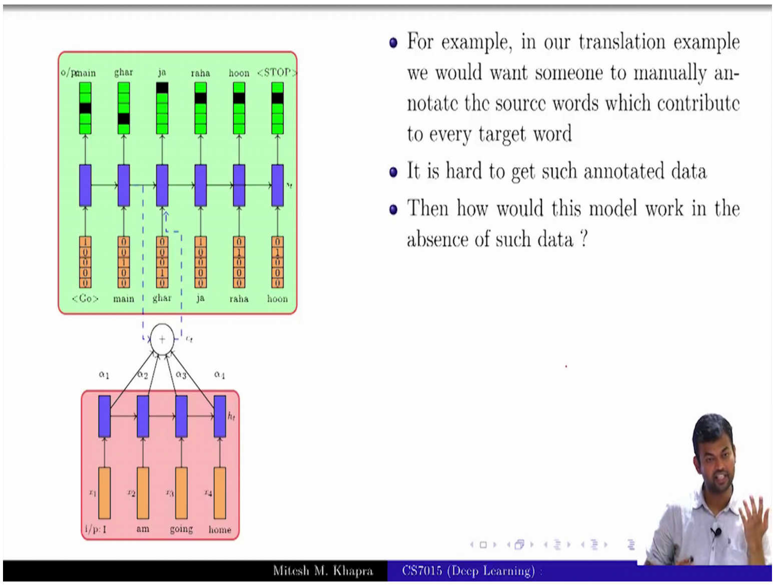
Loss function what would the loss function be?

Student: (Refer Time: 13:04).

Say a squared error loss; and then I want to adjust the parameters to minimize this squared error loss. Then all of this makes a lot of sense, right because then you can imagine that your U attention, W attention and V attention will get tuned in a way that the predicted weights are very close to the true weights this we all understand.

Given an objective function we understand that the weights will get adjusted so that you are there to the objective function. But the whole premise was that we do not have the true alphas because in the case of translation no one is going to tell you that the kth word can come came from the jth word or the set of j words, do you all agree with that. So, if we had the true alphas this makes a lot of sense because then we could have added a loss function which takes the loss of alpha true with respect to alpha prediction. And then an addition to our lost theta, which was the sum of the cross entropy errors and then we could have jointly minimize this and we could have hope that the attention parameters would have been learnt accordingly.

(Refer Slide Time: 14:06)



- For example, in our translation example we would want someone to manually annotate the source words which contribute to every target word
- It is hard to get such annotated data
- Then how would this model work in the absence of such data ?

In practice we will not have this. In our translation example we would want someone to manually annotate for every word in the output which is the set of input word from is this which it came is not going to be possible this, we cannot collect so much annotated data. So, what do we do? Why should this model then work? They does not have any supervision why should this model work in the absence of such data.

How many of you get the meaning of the question? How many of you see the problem please raise your hands. We are not given the true alphas and that is what a problem is then why should this work better. This works better because this is a better dash, better dash choice language model better dash choice; what is the possibility is there, better modelling choice. Why? Ok, I give you the answer. This was better because this is a better modelling choice. Why so? So, I will given you anology and the reinforcement learning fans will cringe but they can just go out.

So, suppose I trying to learn a bicycle how to ride a bicycle, that is why I said they will cringe I already see some of you can see as if you guy have a copyright on bicycles, ok.

So, suppose you trying to learn a bicycle, and for some reason you in your infinite wisdom, you decide that you can learn how to ride it without holding the handle, and you start trying to do it. It is conceivable that you know few years or decades you will actually know how to ride the bicycle, right even if you are not holding the handle, right people do that and people can ride it before without that.

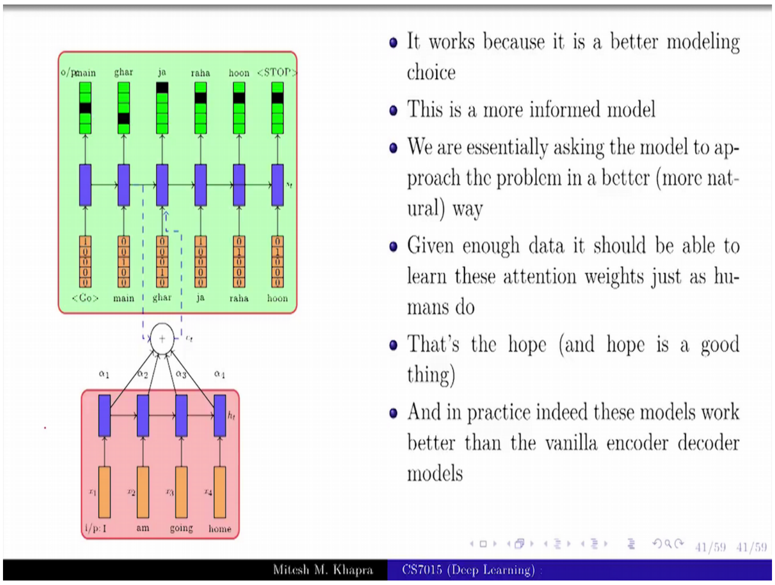
Now, the only thing that I do is I come and tell you that instead of doing this why not you try to hold the handle and then try to ride the bicycle, right that is all I am telling you. I am not giving you any other supervision I have just given you a better model. I have said that instead of just trying to adjust the parameters with respect your feet and the pedal and your back position why not you also introduce this additional parameters where you are holding the bicycle which your hands and now try to figure out what kind of weights you need to put on your left hand, right hand and so on.

I am not giving you any supervision for that that you need to still discover on your own. That you will start riding it you might fall on one side you might fall on the other side. But you will eventually figure out what these weights need to be, right. Because a second model where you hold the handle is a better model than the first model where you are not holding the hand.

In the second model you have additional parameters where you could adjust these parameters. So, that you could learn to drive better that some more natural way more close to human way of learning how to ride a bicycle the same thing is happening here. The second model where you have a way of learning these attention on the weights even though I am not all I am telling you that look maybe if you decided every time step which were to pay attention on you might be able to do better than feeding the information from the entire sentence at every time step.

That is all the information that I am giving you which is very similar to saying just hold the hand. That is not going to teach you how to ride a bicycle, right. You still have to do the extra work of learning these parameters, but now you are given a chance you are giving the model a chance to learn these parameters they are telling it that, this is a better way of modelling it with this you should be able to learn better, right.

(Refer Slide Time: 17:17)



- It works because it is a better modeling choice
- This is a more informed model
- We are essentially asking the model to approach the problem in a better (more natural) way
- Given enough data it should be able to learn these attention weights just as humans do
- That's the hope (and hope is a good thing)
- And in practice indeed these models work better than the vanilla encoder decoder models

41/59 41/59

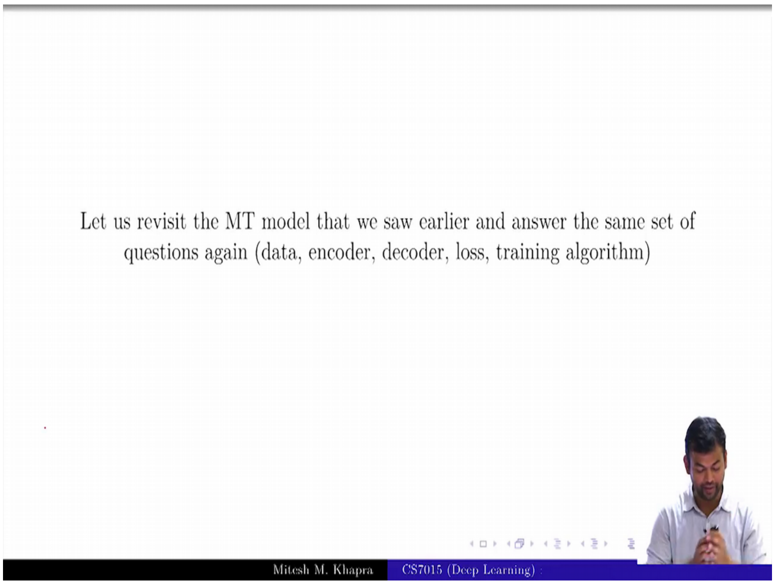
Mitesh M. Khapra CS7015 (Deep Learning)

So, there is a hope of doing better because now the model is actually making a more informed choice, right. It is a more informed way of learning how to do translation by focusing on certain words at every time step.

And now these parameters how will they get adjusted? They will get adjusted because at every at a given time step you produced a wrong output you did that maybe because this parameter was wrong which is the v parameter or maybe because these recurrent connections were wrong or maybe because your attention weights are not proper. So, now, adjust the attention weights and that should given sufficient data it should be able to learn which words to focus on, just as humans learn how to do translation, right.

Even when we are doing learning how to translate or when we learn translating from one language to another we are not given this word by word supervision, right. We just do a lot of translations or read a lot of translations and somehow understand that while translating I need to focus on certain words and at every time step this is the word that I need to focus on. So, given enough words it should be able to learn that at least someone gets the joke good. So, that is the hope and in practice indeed these models work better as compared to vanilla encode, you do not know where the statement comes from.

(Refer Slide Time: 18:34)

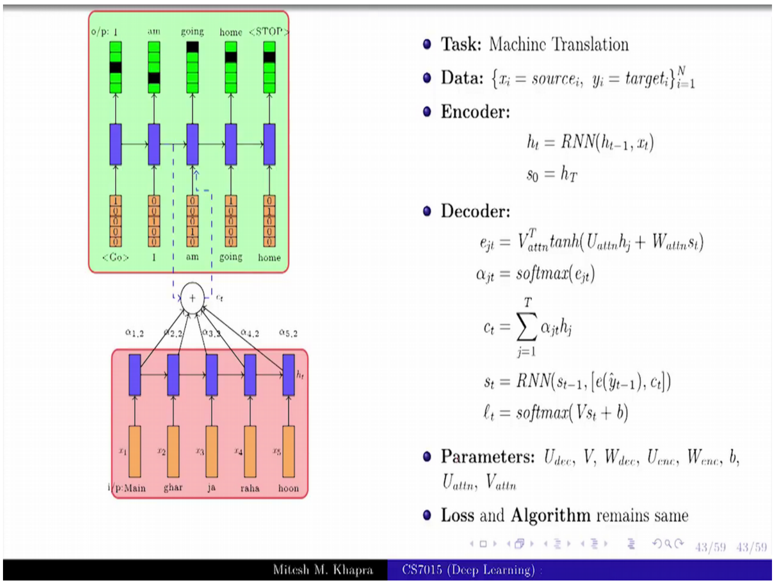


Let us revisit the MT model that we saw earlier and answer the same set of questions again (data, encoder, decoder, loss, training algorithm)

So, now, let us what we will do is, so this entire thing hints on hope only, right that is all that is all I am saying but it does makes sense, right because you have these additional parameters, which you can learn and you can back propagate through them. I will just not stop there will actually prove what happens not proved by demonstrate what happens in practice, right.

So, with this attention model in mind let us look back at the encoder decoder model that we had for machine translation integrate the attention mechanism with it and then let us see the end to end equation that we get, ok.

(Refer Slide Time: 19:05)



- **Task:** Machine Translation
- **Data:** $\{x_i = source_i,\ y_i = target_i\}_{i=1}^{N}$
- **Encoder:**

$$h_t = RNN(h_{t-1}, x_t)$$
$$s_0 = h_T$$

- **Decoder:**

$$e_{jt} = V_{attn}^{T} tanh(U_{attn}h_j + W_{attn}s_t)$$
$$\alpha_{jt} = softmax(e_{jt})$$
$$c_t = \sum_{j=1}^{T} \alpha_{jt}h_j$$
$$s_t = RNN(s_{t-1}, [e(\hat{y}_{t-1}), c_t])$$
$$\ell_t = softmax(Vs_t + b)$$

- **Parameters:** U_{dec}, V, W_{dec}, U_{enc}, W_{enc}, b, U_{attn}, V_{attn}
- Loss and **Algorithm** remains same

43/59 43/59

Mitesh M. Khapra CS7015 (Deep Learning) :

So, this is what the diagram looks like. The input and output still remains the same we have just given the source sentence and the target sentence in particular you are not given which words to pay attention to every time step, right that is not given. So, remember that my input is not changing it still the same source sentence and the target sentence. What is the encoder?

Now, try to work out the equations I wanted to write the equation for y t which is going to be some composite function of x where x is a vector, it is a x 1, x 2, x 3 all the words in the input, and somewhere along the line, it also going to have this attention equation. It is going to take a while but at least try to imagine it there is some hints in the diagram itself you could take a look at it I am just asking you to convert the diagram to a set of equations.

So, encoder part is fine, I have computed the representation of each word at time step t. So, this is a contextual representation the word because it is aware of what the neighboring words are, right. Now, what is the decoder going to be? What is the first thing at time step 1 or a time step t in the decoder, what is the first thing that I am going to compute? The dash weights the last time step of the decoder of the encoder, sorry.

What is the first thing that we need to compute a time step t? t attention weights. Speak up please. What is the first thing that you need to compute at every time step? How what kind of a combination I take off the inputs or rather which are the words that I need to focus on from the input. Who tells us this?

Student: Attention weights.

The attention weights. How will you compute the attention weights? Using this fancy equation that we have seen earlier, is this enough I need to convert this to a.

Student: Probability distribution.

Probability distribution, right that is just to ensure that everything is a neat combination. Once I know the attention weights what do I need to feed the decoder? A dash combination of the inputs, a weighted combination. How do I take a weighted combination of the inputs? Summation i is equal to 1 to capital T.

Student: Alpha j t.

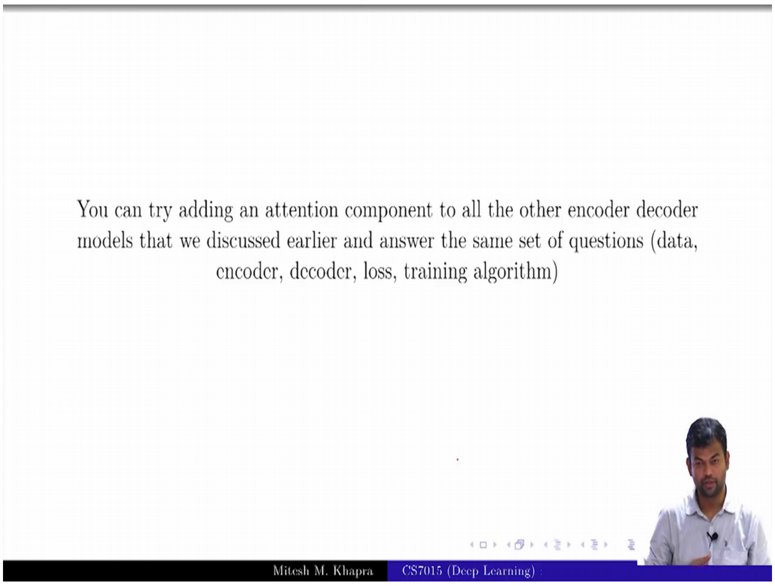
Alpha j t into h j, right, no j; t is the decoder time step j is the input word. So, at the tth time step of the decoder I am taking a weighted combination of all my inputs the index over the inputs is going from j equal to 1 to t. By the way did that answer your question that is what you are asking, right, ok, is that fine, ok.

Now, what next now I want to produce a word at the output. What is the decoder going to be? First thing that I am going to do is; decoder is a dash RNN, ok. What is the input to the RNN every time step? The previous predicted word as well as the weighted combination input that you have given it; does that make sense, ok. And then finally, how do I get the probability distribution is that fine yeah I think this should be a distribution, right l t; does not make sense l t is r max of this, right. And what is the loss function?

Student: Cross entropy.

Cross entropy, there is no change in the loss function, right, loss and the algorithm remains the same. Say seen these set of equations. Now, how many of your confident of going back and modifying all the wrong encoder decoder modules that we have covered in the initial part of lecture; modifying them to add the attention equations in it. How many of you can do that? Please raise your hands, I am not going to ask you just do it so that I feel happy you can do, right. Any questions at this point, very good, ok.

(Refer Slide Time: 23:05)



You can try adding an attention component to all the other encoder decoder models that we discussed earlier and answer the same set of questions (data, encoder, decoder, loss, training algorithm)

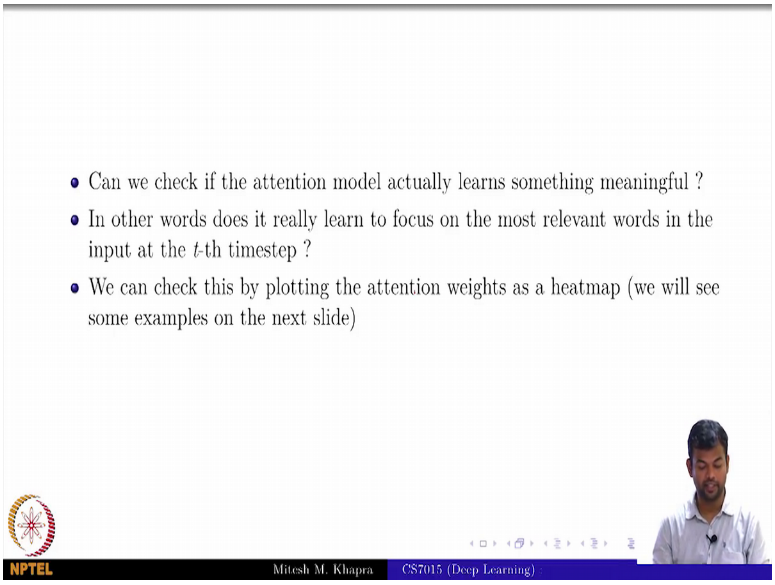
Mitesh M. Khapra CS7015 (Deep Learning)

So, you can go back and try adding attention mechanisms to all the models that we have seen before, right. See how will you compute. So, remember the only purpose of, ok. What kind of a network is the attention network? It is a single feed forward neural network, right, this is just transforming a simple linear transformation of the inputs and in a non-linearity on top of that and then just again one more transformation, right.

It is a simple feed forward neural network. Only these 3 equation somehow need to be fitted in all the other models that we have seen so far, right. This is a very generic framework just as the encoder decoder framework or the very generic framework. The encode attend decode framework is also very generic framework, you can go back and model all the applications that we saw and you can change them change them to at the other case, ok. Try to answer the same set of questions what is the data, what is the encoder, what is the decoder, what is the loss, what is the training function.

And in particular remember that in when you go back represent all the applications that you have done the data is not going to change. No one is going to give us the supervision for the alphas. That is one thing which is not going to change, ok.

(Refer Slide Time: 24:12)



- Can we check if the attention model actually learns something meaningful ?
- In other words does it really learn to focus on the most relevant words in the input at the t-th timestep ?
- We can check this by plotting the attention weights as a heatmap (we will see some examples on the next slide)

So, here is one more thing. So, this probably tie to this question, like how do we be sure that the alphas actually learn something meaningful. Now, what do I mean by this? If I have to convince you that alphas are actually learning something meaningful, and let us take the context of machine translation. What do I need to show you? Suppose the model has generated an output for a given input sentence it has generated a translation. What do I need to show you to convince that it is learn some kind of weights? At every time step what should I show you?

Student: (Refer Time: 24:46).

What does the attention weights look like, right? So, let us see.

(Refer Slide Time: 24:50)

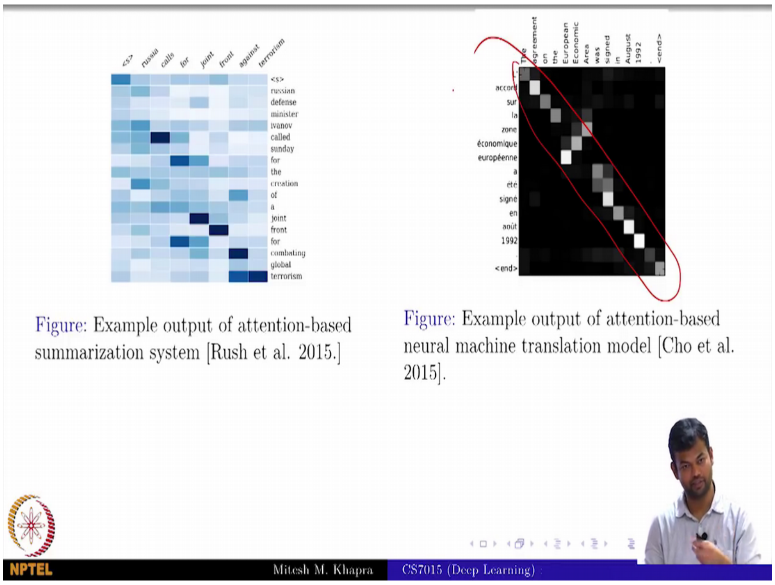


Figure: Example output of attention-based summarization system [Rush et al. 2015.]

Figure: Example output of attention-based neural machine translation model [Cho et al. 2015].

This is a common trick or not a trick actually it is a probably a trick only, but this is the common thing which is used in several papers and that is why I call it a trick because it is a trick to get a paper accepted that you actually show what the attention weights actually look like, right. So, on this is the input document and this is the summary that you want to generate, ok.

And what you see here is that at different time step, so look at the last time step terrorism it paid maximum attention to the word terrorism in the input, right. So, we can draw this matrix suppose you had capital L time steps in the output and capital T time steps in the input. So, you could draw this L cross T time step or T matrix which tells you what was the attention paid to every input word at every output time step. Do you get that?

You see what is matrix is this heat map is essentially a matrix of size L cross T and every cell here tells you how much attention you paid to a particular word at that time step and the darker the cell that means, more the attention that you paid everyone get this, ok. So, what this is saying is that probably see when you wanted to generate Russia, the maximum attention was paid to Russian and maybe some other words also, sometimes it does not work very well but sometimes it does, right.

So, for calls the maximum attention was paid to called, and then similarly for front with the maximum attention was paid to front and so on. You see some meaningful patterns that it is learning here.

And here is another example for machine translation. So, roughly to quickly understand what this figure is, right. So, this is I think English to French, is it French its French or French to English translation which is largely monotonic, right. That means, at the 4th English word you would end up paying attention to the 4th French word, right that means, you are almost doing a word by word translation.

And that is exactly what you see the most of the attention is along the diagonal, right. So, it is learning some meaningful attention weights. As always helpful if you are using if you are using an attention mechanism to plot the sense see if it is actually learning any meaningful attentions or attention weights or not, right. So, that is a common trick which people use.