

Deep Learning

Mithesh M. Khapra

Department of Computer Science and Engineering

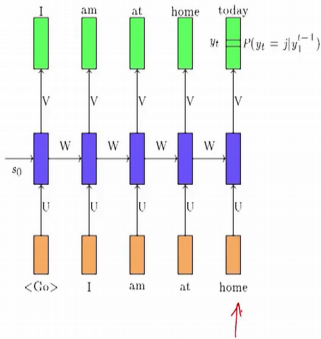
Indian Institute of Technology, Madras

Lecture – 15

Encoder Decoder Models, Attention and Mechanism

And in this lecture, we are going to talk about Encoder Decoder Models and Attention Mechanism. So, this is a very interesting lecture at least interesting to me because, this is where we put all these pieces that we have learnt so far right. We have learnt 3 types of networks; feed-forward networks, recurrent neural networks and convolutional neural networks and we have seen independent applications of each of these word to vec and image classification and so on. Now today, what we are going to see is how do we do different combinations of these networks and come up with a wide range of applications like apply them to a wide range of applications ok. So, let me start by an introduction to encoder decoder models and then we do various applications of encoder decoder models.

(Refer Slide Time: 00:53)



- We will start by revisiting the problem of language modeling
- Informally, given ' $t - i$ ' words we are interested in predicting the t^{th} word
- More formally, given y_1, y_2, \dots, y_{t-1} we want to find
$$y^* = \operatorname{argmax} P(y_t | y_1, y_2, \dots, y_{t-1})$$
- Let us see how we model $P(y_t | y_1, y_2, \dots, y_{t-1})$ using a RNN
- We will refer to $P(y_t | y_1, y_2, \dots, y_{t-1})$ by shorthand notation: $P(y_t | y_1^{t-1})$

3/59 3/59

Mithesh M. Khapra CS7015 (Deep Learning)

So, what we are going to do is we will start by revisiting the problem of language modeling. So, the problem of language modeling was that you are given some $t - 1$ words or characters and you want to predict the t th word or character right, this is like auto complete in short right, whenever we are typing something, you have type 4 words, you want to predict the fifth word. Or you have typed 4 characters and you want to predict the fifth character ok.

So, more formally this is what we are interested in. How many of you get this equation this expression? So, we are given a sequence of $t - 1$ words and you want to find out, what the value of y would be at time step t and we want to find out that value with maximizes this property. That is what this argmax equation means and now we will try to see, how to model this using a RNN.

So, let us see. We are going to start with go that is that, we want to start generating a sentence and then we will produce the first word which is I and what is it that we are predicting at this point, what is the network supposed to predict, what is the output supposed to predict actually?

It is supposed to predict a dash over the vocabulary, a broadly distribution over the vocabulary right. So, this is what is happening. We will of course come back to this on the next few slides, but you have say words W_1, W_2 up to W_V in your vocabulary at every time step, you want to find a distribution over these words and then pick the word, which had the maximum probability at that time step right; that is exactly, what this quantity is that is what we want the RNN in to model and then we want to keep doing this till, we reach the end of the sentence ok. So, that is the language modeling problem and as we had made a case for it earlier, the word produce the time step t depends on a few previous words. How does a recurrent neural network ensure that? At any time step, I am going to give it only 1 word as the input.

So, how does that ensures that it depends on all the previous words also? Through the recurrent connections and the gate and sorry it is not the gate the state t fine.

(Refer Slide Time: 03:03)

- We are interested in

$$P(y_t = j | y_1, y_2, \dots, y_{t-1})$$
 where $j \in \mathcal{V}$ and \mathcal{V} is the set of all vocabulary words
- Using an RNN we compute this as

$$P(y_t = j | y_1^{t-1}) = \text{softmax}(Vs_t + c)_j$$
- In other words we compute

$$P(y_t = j | y_1^{t-1}) = P(y_t = j | s_t) = \text{softmax}(Vs_t + c)_j$$
- Notice that the recurrent connections ensure that s_t has information about y_1^{t-1}

Mitesh M. Khapra CS7015 (Deep Learning) 4/50 4/50

So, we will see this of course, in more detail and we will write down the model equations and what is happening. So, we are interested in this quantity, which is the probability of the word at the time at the t th time step, where this j belongs to vocabulary V and see a vocabulary of 10 k words or 20 k words for English it is actually much higher, but say you are considering only 10 k to 20 k words, then we want to predict a distribution over this vocabulary. So, using an RNN what are you going to do at the output layer is the following is this correct? How many if you understand this equation? Not many why? What does this equation compute, first of all softmax means?

Student: Probability distribution.

Probability distribution what does it take as input at every time step? The state right, what does it do with the state? A linear transformation right and then a bias ok. So, what is this quantity, scalar vector matrix, vector of size?

Students: (Refer Time: 04:00).

The (Refer Time: 04:01). What is the g th element of the that?

Students: Probability of the g th word.

The probability of the g th word right. So, I just have to explain it in that many words, everyone gets it, now, everyone gets it? If you do not get it, you will not understand the rest of the lecture. I am very serious everyone gets it?

So in other words, what we do this entire y_1 to y_{t-1} , which we were conditioning on we are just using s_t as a surrogate for that and that is fair because, s_t has actually captured all the previous information that we had, now just using s_t as a state which captures everything that happened so far.

So, that is actually how we are modeling this and the recurrent connections ensures that s_t captures everything which has happened so far.

(Refer Slide Time: 04:43)

- **Data:** All sentences from any large corpus (say wikipedia)
- **Model:**

$$s_t = \sigma(Ws_{t-1} + Ux_t + b)$$

$$P(y_t = j | y_1^{t-1}) = \text{softmax}(Vs_t + c)_j$$
- **Parameters:** U, V, W, b, c
- **Loss:**

$$\mathcal{L}(\theta) = \sum_{t=1}^T \mathcal{L}_t(\theta)$$

$$\mathcal{L}_t(\theta) = -\log P(y_t = \ell_t | y_1^{t-1})$$
- **Algorithm:** Backpropagation Through Time (BPTT)

where ℓ_t is the true word at time step t

5/50 5/59

So now, let us look at the 5 things that we have in a typical supervised machine learning set up which are those? Data, model.

Students: Parameters.

Parameters.

Students: Objective function.

Objective function.

Student: Cross (Refer Time: 04:54).

Very good, no someone said objective function and then loss function, learning algorithm right ok. So, what's the model here?

Students: (Refer Time: 05:01).

You know, what you are trying to model which is a property distribution, what is the actual? So, here y is the probability distribution and your x is the input given to you, can you tell me what's and we have already set always said in this course that, whatever be the y whatever the be the x , we are interested in this function x sorry function f and we should be actually expressively, we able to write this function. So, what is the function here? Can you actually write down the set of equation? Just think of what the output is, how you are going to reach the output given this network? What is y_t going to be, what's the equation for y_t ?

And then try to go back all the way back to x_t . So, y_t depends on something that something might depend on x_t . So, how do you go all the way back? Right, that is the thing, which I expect you to do how many of you get it now? Please raise your hands ok. So, let us see at every time step, what am I interested in predicting?

Students: Probability distribution.

A probability distribution; that means, I will have to compute which function?

Students: Softmax.

Softmax. So, the green vector is what I am going to focus on. So, what's the equation for the green vector? Is this fine? Now what does this contain apart from the parameters s_t ? How do I get s_t ? Is it fine? You can write, now you have written this output y as a function of x because, x appears here or other y_t as a function of x_t is not it is straight forward right once, I show you the answer is should be how many of you get it now? Please raise your hands high up above what are the parameters? B and c right.

So, these are the parameters, what's the objective function? Cross entropy or dash of cross entropies.

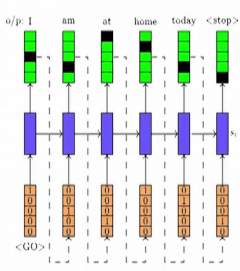
Students: Sum of cross entropies.

Sum of cross entropies right. So, the loss is going to be over, all the time steps at every time step is the cross entropy loss right, everyone gets this? Ok, what's the learning algorithm? Back propagation.

Students: True time.

True time, fine. So, that is what it is going to be right everyone is clear? So, you can see that we have written the final output as a function of the input right and this is end to end trainable; that means, the gradients can flow modulo this vanishing exploding gradient problem and we have a way of handling that, we can replace RNS by LSTMs that is all right . So, that is what it is now this, just make sure you understand this properly. So, that we are going to do various instantiations of this for different problems ok.

(Refer Slide Time: 07:31)



- What is the input at each time step?
- It is simply the word that we predicted at the previous time step
- In general

$$s_t = RNN(s_{t-1}, x_t)$$
- Let j be the index of the word which has been assigned the max probability at time step $t - 1$

$$x_t = e(v_j)$$
- x_t is essentially a one-hot vector ($e(v_j)$) representing the j^{th} word in the vocabulary
- In practice, instead of one hot representation we use a pre-trained word embedding of the j^{th} word

Mitesh M. Khapra CS7015 (Deep Learning) 6/59 6/59

Now here is 1 question, we all smartly wrote this x_t , but why is the input at every time step? When I am predicting home, the input was at, but how did I get at? That is what I dash at the previous time step, predicted at the previous time step right.

So, this is what the input looks like. So, at time step 1 I predicted I as the output at the next time step, I am going to feed that as the input, does it make sense? So, just see if you are doing auto complete, you would select that I am fine with the word I at this time step. So, it is going to take that as the input and then try to predict the next word that is

what exactly is happening here and now you are predicted am at the next time step you are going to feed am as the input and continue this chain throughout ok.

So, the input at every time step is going to be the word that you have predicted the previous time step and I am just going to represent it by a 1 hot vector right. It is the index of the j th word only that could be hot everything else would be 0 and all of you are fine with this no. So, at training time this is the inference time at training time, we will have the real inputs no, that is at inference time at training time, we will just use that through because, training time you know what the inputs are right, you know the true sentence you have the Wikipedia sentence right and you know what the 2 sentence is going to be.

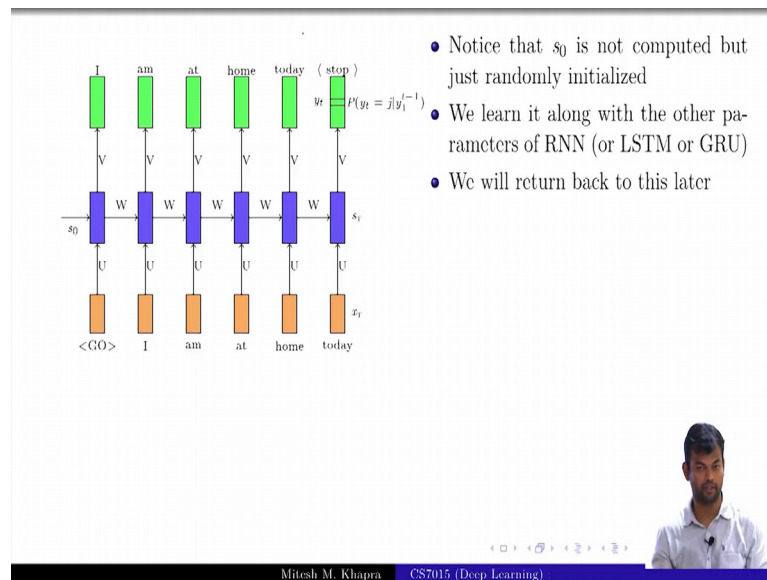
What I am talking about, how will you generated at test time at training time, you know all these things right. No about training time, how will you do that? You will know what the next input is right. So now, ok. So, I said that the input is going to be a 1 hot vector. is everyone fine with that? 1 hot vectors are ok, what else could you use?

Students: Word representation.

The word representation for that right. So, assume that you have already done the word to vec assignment and you have completed all the word representations and you have them with you now instead of feeding the one hot representation of the input, you can just feed the word representation of it does that make sense? One hot representation is just one of the many representations possible for the world. So, why just do that, you could do s v d, you could do one word vec or whatever you want right. So, that is in practice, what we will feed is the word to vec representation.

So, everyone gets this what is happening at every time step?

(Refer Slide Time: 09:44)



Now, one more thing, that you need to notice that s_0 ; which is the input at time step 1, the previous; so, s_{1-1} . So, that we do not know what it is. So, we just keep it as a parameter, we say that s_0 is also weight vector and you are going to learn it, along with all the other parameters in the network does not make sense, because you do not know, what s_0 means is a semantics of it is not clear like, what was generated at the zero-th time step, we do not really know right. So, will just make it a learnable parameter and that would be trained along with all the other parameters of the network.

(Refer Slide Time: 10:17)

$$s_t = \sigma(Ux_t + Ws_{t-1} + b) \quad \tilde{s}_t = \sigma(W(o_t \odot s_{t-1}) + Ux_t + b) \quad \tilde{s}_t = \sigma(W h_{t-1} + Ux_t + b)$$

$$s_t = i_t \odot s_{t-1} + (1 - i_t) \odot \tilde{s}_t \quad s_t = f_t \odot s_{t-1} + i_t \odot \tilde{s}_t$$

$$h_t = o_t \odot \sigma(s_t)$$

$$s_t = \text{RNN}(s_{t-1}, x_t) \quad s_t = \text{GRU}(s_{t-1}, x_t) \quad h_t, s_t = \text{LSTM}(h_{t-1}, s_{t-1}, x_t)$$

• Before moving on we will see a compact way of writing the function computed by RNN, GRU and LSTM

So, before we move on what we are going to do is, we are going to see a very compact representations for RNNs, GRUs and LSTMs. So, remember RNN is the following equation RNN is defined by the following equation. The s_t is a recursive function of s_{t-1} and x_t right. So, I am just going to write it as that s_t is equal to RNN of s_{t-1} comma x_t , instead of writing all these parameters and sigma's and all that, I am just going to write it compactly as this, now this is what, what is this? GRU. So, how may going to write it as?

Students: GRU.

GRU of?

Students: s_{t-1} x_t .

s_{t-1} comma x_t .

What is this?

Students: LSTM.

LSTM, how may going to write it? LSTM of when the output of the LSTM is both h_t minus 1 and s_t minus 1 right fine. So, in some sometimes I will just say s_t , sometimes I will say both h_t minus h_t and s_t as per whatever I needed right. So, this is I am not going to write these equations and parameters again, I will just say that LSTM of this, assume that is a function which does this calculation and gives you back ok.

(Refer Slide Time: 11:22)

- So far we have seen how to model the conditional probability distribution $P(y_t | y_1^{t-1})$
- More informally, we have seen how to generate a sentence given previous words
- What if we want to generate a sentence given an image?
- We are now interested in $P(y_t | y_1^{t-1}, I)$ instead of $P(y_t | y_1^{t-1})$ where I is an image
- Notice that $P(y_t | y_1^{t-1}, I)$ is again a conditional distribution

Mitesh M. Khapra CS7015 (Deep Learning) 9/59 9/59

So, far what you have done is we have seen, how to model the conditional probability distribution given the previous t minus 1 words. Now, let me give you a different application right, what if we want to generate a sentence given an image. So, this is what I am interested in doing, I am giving an image and I want to generate a sentence, can we just think of it formally, what is it that you want to do? So, we saw that in this case formally, we were interested in this conditional distribution in this case, what is it that we are formally interest in?

If I were to write it as something formal, what would I write it as? Ok I will give you a hint, what kind of a distribution is this? A conditional distribution right, given the previous sequences previous sequence of words generate the t -th word, now in this situation can you stated an similar words? Given the.

Students: Image.

Image generate the.

Students: Sentence.

Sentence or given the image and the description that are generated so far, because I am going to write the description, 1 word at a time given, the image and the description that have written so far, generate the next word in the mission. So, what kind of conditional distribution is that? p of y t given.

Students: Y 1 to t minus 1.

Y 1 to t minus 1?

Students: Comma.

Comma?

Students: Image.

Image does that make sense everyone gets that So what. So, this is what we want right. So here, now we are interested in this quantity as a post to this quantity does that make sense? Ok and this is again a conditional distribution.

(Refer Slide Time: 13:02)

• Earlier we modeled $P(y_t | y_1^{t-1})$ as

$$P(y_t | y_1^{t-1}) = P(y_t | s_t)$$

• Where s_t was a state capturing all the previous words

• We could now model $P(y_t = j | y_1^{t-1}, I)$ as $P(y_t = j | s_t, f_{c7}(I))$

Mitesh M. Khapra CS7015 (Deep Learning)

So earlier, how did we model this? We just modeled it as the following we said that the whole context of y_1 to y_{t-1} is just contained in that blue vector, which is s_t right. So, remove this variable and replace it by a vector does that make sense? =

Now you have the image also. So, how are you going to model this? So, what are you going to write on the right hand side? Ok let me give you a hint, we all agreed that this is the quantity that, we are interested in right. We also agreed that the following is fine replacing y_1 to y_{t-1} by s_t is fine. Now what about the image? What do you mean by objection in the image? You will supply the words, which are there the object names that man fine that is fair enough well, if want to make it more abstract more neural. So,

what you are saying is that whatever information is contained in the image should be passed here.

Whatever information is contained in the image should be passed here, how do you what's the way that you have learnt of computing the information in the image a dash neural network?

Students: (Refer Time: 14:15).

A?

Students: Convolutional neural network.

Feedforward neural network?

Students: Convolutional neural network.

Convolutional neural network ok. So, but what from a convolutional neural network? How many representations that is a convolutional neural network learn? How many does v g g network learn? v g g 16, the last layer is a softmax layer 15 right. So, which one will you give now? One before the last one, that is called the?

Students: (Refer Time: 14:36).

Dash layer, fully dash layer?

Students: Fully connected layer.

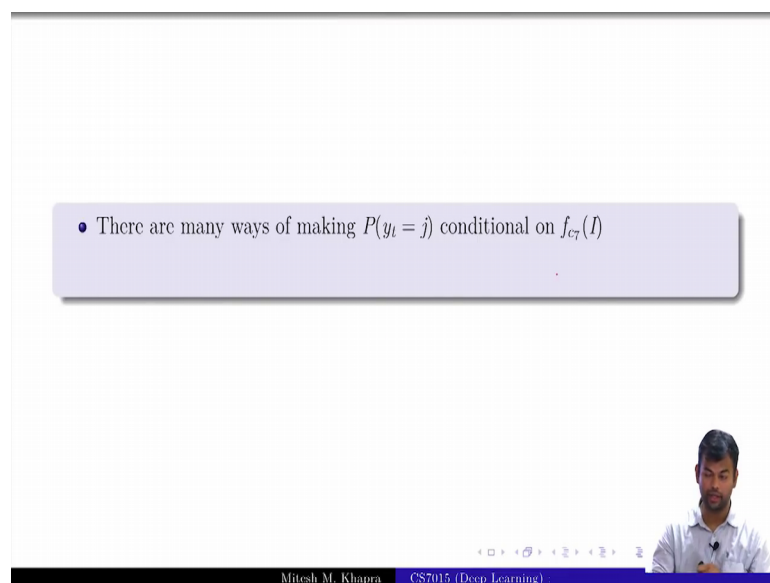
Fully connected layer ok, at least your language moral works fine ok. So, that is the fully connected layer remember that all the layers in the convolutional neural network learn, an abstract representation of the image and as she was trying to say, that this abstract representation contains or at least you believe it contains all the information that is there in the original image, just as s t contains all the information that was there in the sequence y 1 2 t minus 1, this abstract representation that we will get from a CNN, contains all the representation all the information, that is there in the image we all believe that ok.

And we also believe that any of these representation is fine in practice, the convention is to use the fully connected layer that is called as f c 7, the seventh fully connected layer

right and it is 7, because you also start the numbering from the convolution layer 1, 2, 3, 4, 5 and then the seventh layer ok. So, that is what you will take ok. So, does this make sense and it is a very simple extension from what we were doing earlier, this is what I have circles is what we were doing earlier right, where we only had s t. Now, I am saying is just as you believe that s t and codes all the information in the previous sequence.

I am just asking you to stretch that a bit more and say that f c 7 of the image contains all the information that was there in the image is it fine?

(Refer Slide Time: 16:01)

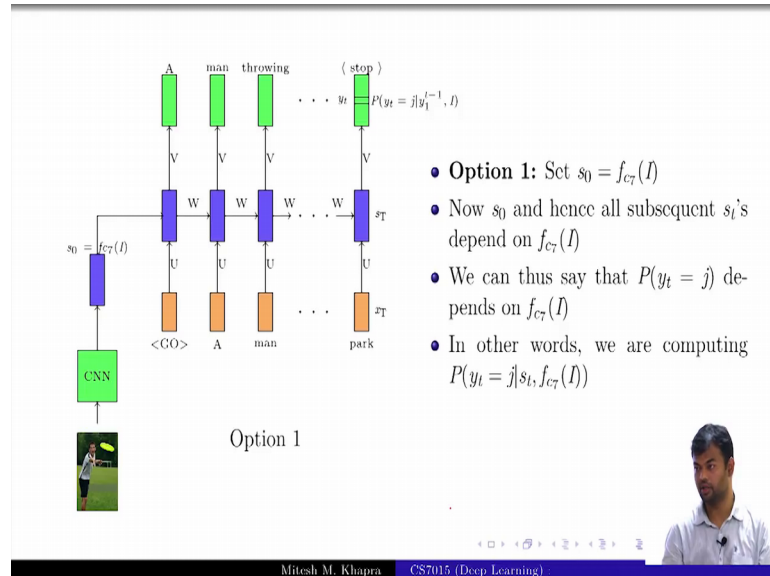


Ok, but still there are some issues and there are other ways of making this condition on f c 7, in particular what you could have done as she was trying to suggest initial is that maybe you have a vocabulary of all the objects that are possible in your image right. So, maybe in your image there is man, woman there is flying desk, frisbee or there is dog, cat and all these things right.

So, you do an object detection first, get out all the object which are there and then make the distribution conditional on these objects right. So, you can say that I will allow for a 10 words to describe the image. So, there word 1 is equal to man, because I have detected the object man in the image word 2 is equal to Frisbee, because I have detected the object Frisbee, in the image that is all that is another way of doing it ok. So, I just want to make it clear, that there are different ways of making the conditional distribution,

conditional on the image itself, we are choosing to make it conditional on f_{c7} of i right that is the neural way of doing it ok.

(Refer Slide Time: 16:59)



So let us see, 2 such options, the first thing that we could do is we could set s_0 to f_{c7} of the image, what is s_0 ? The first thing that was passed to the language model ok. So remember, we had this $\langle GO \rangle$ symbol and we had this s_0 , which was mysterious we did not know how it comes, but now we know it that s_0 could just be the image that is what my starting point is. So, take this image and now start generating the representation generating a description does that make sense? Ok. So, this is what the network looks like.

So, what do you saying is that these things are of dimension d , the CNS output was say of dimension 4096. So, this has to be converted to size d right; that means, what will you, how will you do that? We have a 4096 dimensional vector and you want to convert it to a d dimensional vector w belonging to?

Students: (Refer Time: 17:51).

4096 was d fine in general, any 2 vectors if you want to make them compatible, this is what you will do you will project. So, that they are of the same dimensions like x_0 will be the $\langle GO \rangle$ symbol. So, the is the special word in your vocabulary, which says start generating the sentence right. So, whatever vocabularies you will add 2 special words

right one is go and other is stop. So, whenever you generate stop, you stop generating after that fine what is the other way of. So, here now what happens is. So, this is what is happening technically and that is why that is what I wanted you to understand this now s_1 depends on s_0 what we are interested in is the following that y_t should be conditional on y_1 to t minus 1 comma image ok.

We have made sure that i is s_0 and this quantity is s_t , you have to find now since the first time step depended on the image, all subsequent time steps will depend on the image, is that ok? What is the other way of doing this? What now in this looks slightly inefficient, what's the other option that you could have used? Just feed the image at every time right. So, that is the one constant thing that this is the image, now whatever you have generated so far, considered that, but in the addition to that also consider the image.

(Refer Slide Time: 19:14)

Option 2

- Option 2: Another more explicit way of doing this is to compute

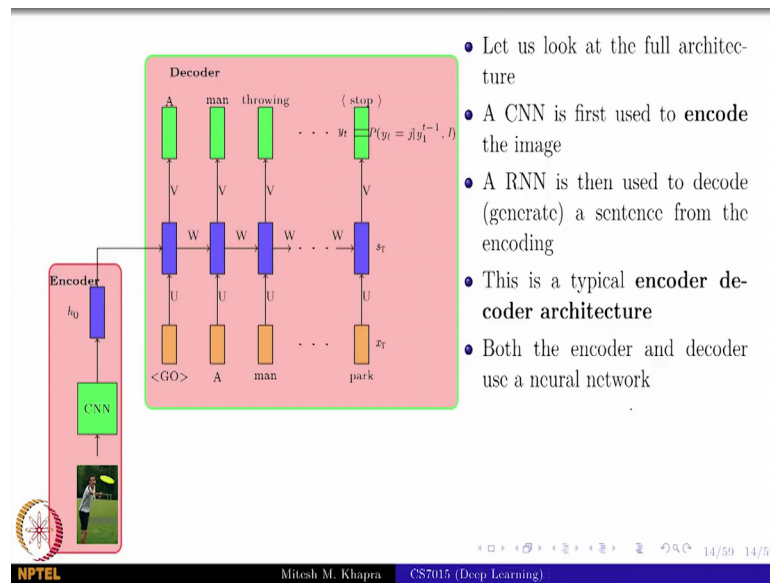
$$s_t = RNN(s_{t-1}, [x_t, f_{c7}(I)])$$
- In other words we are explicitly using $f_{c7}(I)$ to compute s_t and hence $P(y_t = j)$
- You could think of other ways of conditioning $P(y_t = j)$ on f_{c7}

NPTEL
Mitesh M. Khapra CS7015 (Deep Learning)

So, what would the diagram look like just passing, the input to every stage of the decoder ok.

I have already started using terminology, which have not introduced, but I will just introduce it shortly.

(Refer Slide Time: 19:21)



- Let us look at the full architecture
- A CNN is first used to **encode** the image
- A RNN is then used to decode (generate) a sentence from the encoding
- This is a typical **encoder-decoder architecture**
- Both the encoder and decoder use a neural network

So, let us look at what the full architecture looks like, there is something known as the encoder, which takes your input encodes it and gives you a representation right then, you have something known as a decoder because, given this input you want to decode what the output is right. So, remember general terminology would be whatever input is given to you, you want to encode it and whatever is the output that needs to be decoded right it is you could think of it that, this is the image. Now, I am trying to decode the description for there, is that fine? Ok and then you have an RNN, which is used to decode the sentence from this input.

So, such architectures are known as encoder decoder architecture and these are become extremely popular and we will see why they are so popular and why they have led to the popularity of deep learning in general ok. So, everyone understands this diagram, anyone who does not see a problem with this diagram? There is actually, no problem, but I want you, I want you to see beyond the diagram and to look at the equations, what do you mean by that? What do I have here as the input? What is my x ? So, this this looks fine, I have taken 1 box and connected it to another box and everything is fine right, but that is not what I am interested in.

What am I interested in? Can you write the input as a the output as the function of the input in this case, is it possible to do that. So, that is what we need to make sure that we are able to do right. So, we look at various applications suggest lepted criptical here, but

I am going to come back to it. So, I just the emphasis that look beyond the diagram, the diagram looks very nice. I hope it does thanks to the t s, but it does and, but we need to understand, what is the what is the set of equations being conveyed through this diagram right? What is the function that we are trying to learn? We are going to write y as a function of x , are we able to write that function? Because, now we are suddenly thrown in a convolution neural network at some place, we have an recurrent neural network then, we have the feed forward layer at the output, which is the green vectors. So, does all this combined together right.