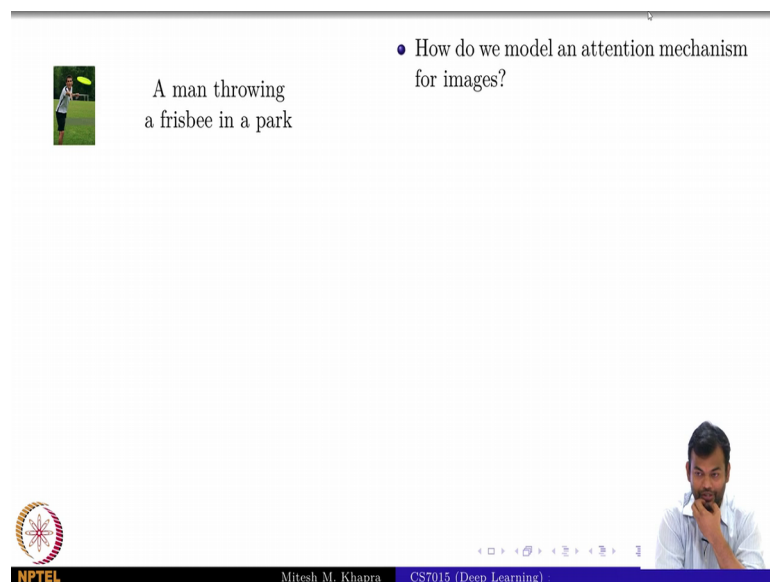


**Deep Learning**  
**Prof. Mitesh M. Khapra**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture - 116**  
**Attention over images**

And so now, in this lecture we will go on to the next module which is talking about Attention over images. So, let us first motivate why is it so different and what could be done there right.

(Refer Slide Time: 00:23)



A man throwing a frisbee in a park

- How do we model an attention mechanism for images?

NPTEL Mitesh M. Khapra CS7015 (Deep Learning)

So, the question is how do we model an attention mechanism for images?

(Refer Slide Time: 00:27)

The diagram illustrates an attention mechanism. At the top, an encoder processes the Hindi input sequence: "main", "ghar", "ja", "raha", "hoon". Each word is converted into a vector (green blocks). These vectors are fed into an encoder (blue blocks) to produce hidden states  $h_1, h_2, h_3, h_4$ . At the bottom, an attention mechanism (red box) takes these hidden states and weights them ( $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ ) to focus on the corresponding English words: "I", "am", "going", "home". The attention weights are shown as orange blocks. The output of the attention mechanism is a weighted sum of the encoder states, which is then used to generate the output sequence.

- How do we model an attention mechanism for images?
- In the case of text we have a representation for every location (time step) of the input sequence

NPTEL Mitesh M. Khapra CS7015 (Deep Learning)

So, in the case of text we have a representation for every location of the input sequence right. So, every location in the input sequence in the case of text was a word, and then you are looking at this problem of transliteration every location was a character. And whether it is a character or a word for everything it was discrete.

So, we could just know that this is the important time step  $t$ , and then we know that along all the inputs at different time steps you want to pay attention to certain time steps right. So, that the definition they was very straightforward right.

(Refer Slide Time: 00:58)

The diagram shows an encoder architecture. An image of a person playing tennis is processed by a CNN (green box). The output of the CNN is fed into an encoder (blue box) to produce a hidden state  $h_0$ .

- How do we model an attention mechanism for images?
- In the case of text we have a representation for every location (time step) of the input sequence
- But for images we typically use representation from one of the fully connected layers
- This representation does not contain any location information
- So then what is the input to the attention mechanism?

NPTEL Mitesh M. Khapra CS7015 (Deep Learning)

Now for images what do we do? So, for images we typically take the representation from a CNN right, it could be FC 7, or any of the convolution layers or max pooling layers right. Now, there is no concept of time step there right because the entire image is given to you at one go.

So, now how do you decide where to pay attention to? But if you think about it does makes sense at least the motivation is very clear. So, for example, for this figure, if I am trying to generate the description as man throwing a Frisbee in a garden or in a park or something like that, right.

So, when I am generating a word man, I would want to focus only on the man and not focus on any other part of the image. Similarly, when I am generating the word Frisbee I would like to focus on this. May be when I am saying throwing I would like to focus on his hand action or something like that. And in a park I would like to focus on the background and so on. So, it does make sense that each word in the description is complete covering from coming from a different space in the image, or different position in the image.

But the representation that we use say the FC 7 representation that is not contain any location information it just a flat, and vector that we had. So, now how do we do this? How do we get attention on locations? This is a motivation and the problem here, and motivation is straight forward. The problem is that we are using FC 7 representation is just a flat vector, remember that was the fully connected vector and does not have any location based encoding.

So, if for example if the fully connected vector is of size 512 I cannot say that the first 12 or first 24 of these 512 dimensions correspond to this set of pixels, the next 24 corresponds to this set of pixels and so on right, so that is the problem. How do I? What do I attend to? How do I decide where to attend to? Because that is what I am saying that the vector the elements of the vector, or the dimension of the vectors do not have any semantic right that is not that the first dimension corresponds to first location.

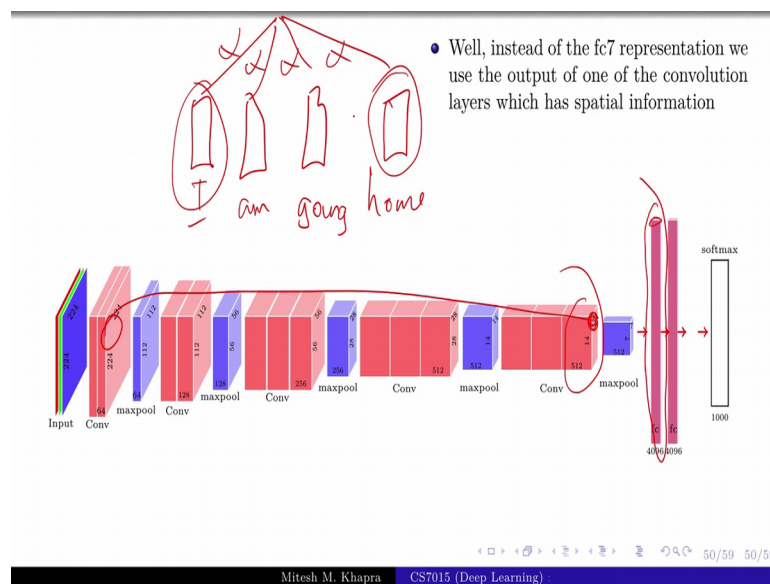
What you want an attention on this? Locations in the image right, but the first dimension in the vector FC 7 vector does not correspond to any specific location in the image you know that was a fully connected vector right. So, it corresponds to everything in the image.

So, what is something simpler than that? Why do I say something simpler than that? Object detection is itself is a in itself is a another convolution neural network which does this and so on right. And we saw this past or seen in past class seen in problems.

Now, let us solve the problem at hand. The problem at hand is that I want I will just rephrase a problem definition, so that the answer becomes obvious. I want a representation which allows to give which allows me to get some location information. No, but the fully connected layer if you back track it was fully connected by definition right, the answer is really straight forward. The problem only arise at the fully connected layer right, because that is fully connected, but what about the outputs on the convolution layers? Do they have position information?

Student: Yeah sir yes.

(Refer Slide Time: 03:56)



We saw that suppose this is VGG whatever it is 16 I guess. And this is what I am saying as a problem because this was fully connected. So, you do not know that this dimension corresponds to location 1, location 2 and so on. But if you look at the convolution layers we know that everything here actually goes back to some location in the image. And if I learn to pay attention to this guy maybe I am paying attention to some equivalent portion in the image does that make sense, right?

Now, can you build on this intuition and tell me I want this as a solution right. So, in the case of words these are the word vectors that I had at every location and I was learning to pay attention to them, let us learning these alphas for each of these. Now what is the corresponding diagram for images? What is each of these box is going to be? So, that we learn the attention weights. What do you mean by pass?

Student: (Refer Time: 04:49).

No. So, maybe I am not understanding your answer. What is the equivalent of this, this box which I have highlighted there, between the image and some attention weight? So, you are directly going to operate on the image. However, attention weights are never given to us no one will mark that man is this, Frisbee is this, and park is this, or whatever, there is no supervision, same size as a convolution what. Now what is the size of a convolution?

Let us take the last one right. So, what do you mean by size? You mean 512, or you mean 14, or you mean the other 14 that is the size of the output 512 cross 14 cross 14. So, what do you mean by the size? 512 or 14, or 14 or 512 cross 14 cross 14. Channel does not make sense because channels capture I mean you do not want to focus on the red part, or the green part, or the blue part in some cases you might that is correct ok.

Student: (Refer Time: 05:39).

These are all partially correct answers going in the right direction. Let just think a bit more and see. So, probably between these two they gave some part of the answer. Now can you think of it? So, you want a representation for the image first of all of us are clear that we do not want to work with the raw image right that is all of us are clear with that. The second part is we are going to work with the convolution neural network. We want to pick up a representation for the convolution neural network which gives us location information right.

And we agree that the fully connected layer does not give us the convolution layers give this ok. Now, I am asking you to focus on one of the convolution layers which is 512 cross 14 cross 14. So, let us see from there how we will be try to get these attentions ok? So, the output of the 5th convolution layer or 5 c this is I think this whole thing is 5 and

this is 5 a, 5 b and 5 c right. These this is how the code or the general architecture is numbered.

So, this guy has 14 cross 14 locations right, the 512 cross 14 cross 14 output. So, it is 512 channels, but the number of locations is 14 cross, 14. And we have seen that each of these 14 cross 14 locations corresponds to certain portion in the image right.

(Refer Slide Time: 06:59)

- Well, instead of the fc7 representation we use the output of one of the convolution layers which has spatial information
- For example the output of the 5<sup>th</sup> convolutional layer of VGGNet is a  $14 \times 14 \times 512$  size feature map
- We could think of this as 196 locations (each having a 512 dimensional representation)

Now, for each of these 14 cross 14 locations; that means, I have 196 such locations. And for each of this how many dimensional representation do I have? I want everyone to say this.

Student: 512.

512. Because we are reading it from the figure? No. What you are taking is the one taking this pixel, can you see what I am highlighting? I am taking it across the depth right. So, that is why 512 dimensional representation of one pixel in your output volume. And how many such pixels do you have?

Student: 512.

196. And each of these pixels corresponds to some real location in your image; that means, it has space information right ok. So, now these are the 196 locations that you have. Now this looks very much similar to that diagram that we had for words. So, now

you can think about that you have 196 items in your sequence. And now what will you do? We will learn to pay?

Student: (Refer Time: 07:49).

So what would that look like?

(Refer Slide Time: 07:52)

$\alpha_{tj} = V \tanh(Ws_{t-1} + U h_j + b_m)$

- Well, instead of the fc7 representation we use the output of one of the convolution layers which has spatial information
- For example the output of the 5<sup>th</sup> convolutional layer of VGGNet is a 14×14×512 size feature map
- We could think of this as 196 locations (each having a 512 dimensional representation)
- The model will then learn an attention over these locations (which in turn correspond to actual locations in the images)

$\alpha_{tj} = f(s_{t-1}, h_j)$

NPTEL Mitesh M. Khapra CS7015 (Deep Learning) 50/59

So, at every time step we will have an equation for alphas right. Let us try to write an abstract equation right first of all give me the indices of alpha. What does alpha compute? The importance of the of the j-th location at the p-th time step fine is that ok? What should this be a function of second part is kind of obvious. So, let us call these as h 1 to h 196, so these are the j's or the t's j's or the t's?

Student: j's.

j's. So, what would the second parameter be?

Student: h j.

h j. And what is the first parameter be? What is the decoder in this case? We are trying to generate a caption; that means we are trying to generate a sequence; that means, what will be the decoder be?

Student: RNN.

RNN. So what should I depend on  $s_t$  or  $s_{t-1}$ ?

Student:  $s_{t-1}$ .

$s_{t-1}$ ,  $s_t$  is the current thing right that we do not know yet. So, it will depend on  $s_{t-1}$ , comma  $h_j$ . Of course, you can make it depended on several other things also, but at the minimum you will see these two things right because you are trying to understand the importance of these guys, so these better participate in the function. And you are trying to compute the importance at current time step. So, you better know what has happened till time step  $t-1$ , it is not very different from the attention equation that we had written in fact, it is a same actually.

And what was one form of this attention that we had seen? Does anyone remember that? We had carefully analysed the parameters and the dimensions of that form. What should the output of this function be scalar, vector, matrix? Scalar right.

So, what is the form that we had seen for this function?  $V^T \tanh$  of something. You should get comfortable in writing these equations right because that is what you will do if you are proposing your models and so on right. So, you will see in the previous model this depended on the following two quantities I think it should depend on four more quantities. So, I will write a new equation. Just think about it there are 2 inputs  $s_{t-1}$  and  $h_j$ . What will be doing with each of these inputs? Do a introduce some.

Student: Parameters.

Parameters. So, what will you do?

Student:  $W$  into  $s_{t-1}$ .

$W$  into  $s_{t-1}$  plus.

Student: Plus  $U$ .

$V$  into sorry  $U$  into.

Student:  $h_j$ .

Plus some bias right. So, you just comfortable with this right I mean that is all I mean whatever we have seen. So, first of all remember that the attention is a feed forward



neural network. The moment I tell you that you should know that it will have a linear transformation followed by a non-linearity right. So, that should have been very clear that it would have a linear transformation followed by a non-linearity ok.

And this is the non-linearity and then you have this other constraint that you want the output to be a scalar. That is why you had this guy which was a vector multiplied by a vector which gives you a scalar. Everyone is comfortable with this, right? So, we see at the attention over images is not very different from attention over sequences. It is more or less the same once you figure out what is the correct representation to you so that you get the space information after that it is straightforward right ok. So, that ends the module.