

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

Lecture - 105
Backpropagation through time

So, that was Recurrent Neural Networks. Now, whenever we propose a network what do we do next? Training, right. So, what we will look at it back propagation through time. This is not the title of a fiction and movie or anything; this is an algorithm that we will see.

(Refer Slide Time: 00:27)

- Before proceeding let us look at the dimensions of the parameters carefully

$x_i \in \mathbb{R}^n$ (n-dimensional input)
 $s_i \in \mathbb{R}^d$ (d-dimensional state)
 $y_i \in \mathbb{R}^k$ (say k classes)
 $U \in \mathbb{R}^{n \times d}$
 $V \in \mathbb{R}^{d \times k}$
 $W \in \mathbb{R}^{d \times d}$

Mitesh M. Khapra CS7015 (Deep Learning) /41

So, before we proceed, right let us look at the dimension of the parameters that we have and I expect you to tell me the dimensions. So, I will define some things for you which are very hard. So, x_i belongs to \mathbb{R}^n , so let us be clear about that; s_i belongs to \mathbb{R}^d that means, the s_i is a d dimensional vector and y_i belongs to \mathbb{R}^k which has k classes, ok.

So, now what is U ? What is V ? D cross k , is it d cross k I am asking Soham? Now I mean we have written it as d cross k . And W is?

Student: (Refer Time: 01:05).

D cross t sure; everyone sure, right. So, these are the dimensions. Why am I talking about these dimensions? Whenever we talk about gradients what we talk about partial derivatives or gradient or something we need to know what is the size of the parameter with respect to which we are taking the gradient because that is what the size of the gradient matrix is going to be, right. That is why I am asking you to focus on this.

(Refer Slide Time: 01:31)

• How do we train this network ?
(Ans: using backpropagation)

• Let us understand this with a concrete example

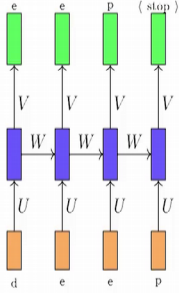
Mitesh M. Khurana CS7015 (Deep Learning) 2/41

Now, how do we train this network? Title of the module.

Student: Backpropagation.

Backpropagation, ok. How? Why do I have a module if I am only going to tell you about backpropagation? Do you see any problem with this? Why cannot you just apply the standard backpropagation (Refer Time: 01:47)? So, we will try to understand this with the help of a concrete example and we will go back to our example of predicting characters, ok.

(Refer Slide Time: 01:51)



- Suppose we consider our task of auto-completion (predicting the next character)
- For simplicity we assume that there are only 4 characters in our vocabulary (d,e,p, <stop>)
- At each timestep we want to predict one of these 4 characters
- What is a suitable output function for this task ? (**softmax**)
- What is a suitable loss function for this task ?

Mitesh M. Khapra CS7015 (Deep Learning) 23/41

So, this is the auto completion task and for simplicity we will assume that English has only these 3 characters d, e, p and then I stop to indicate that the world has been completed, ok. This is what you are going to consider that my vocabulary size is just 4 that means, I can only predict one of these k 4 classes, k is equal to 4, ok.

And at each time say I want to predict one of these things. What is the suitable output function for this task? Can everyone say with probability 99.9 percent?

Student: Soft max.

Soft max, ok. What is the suitable loss function for this task? Small pleasures in life that is all I get, ok.

(Refer Slide Time: 02:34)

Predicted True Predicted True Predicted True Predicted True
 d 0.2 0 0.2 0 0.2 0 0.2 0
 e 0.7 1 0.7 1 0.7 1 0.7 1
 p 0.1 0 0.1 0 0.1 0 0.1 0
 stop 0.1 0 0.1 0 0.1 0 0.1 0

The diagram shows a recurrent neural network with four time steps. Each time step has an input (orange box), a hidden state (blue box), and an output (green/red boxes). The output is a probability distribution over the alphabet 'd', 'e', 'p', 'stop'. The true distribution is shown in red, and the predicted distribution is shown in green. The true distribution is [0, 1, 0, 0] for all steps. The predicted distribution is [0.2, 0.7, 0.1, 0] for all steps. The network is initialized with random weights U, V, and W.

- Suppose we initialize U, V, W randomly and the network predicts the probabilities as shown
- And the true probabilities are as shown
- We need to answer two questions
- What is the total loss made by the model ?
- How do we backpropagate this loss and update the parameters ($\theta = \{U, V, W\}$) of the network ?

24/41

Mitesh M. Khapra CS7015 (Deep Learning)

Suppose we initialize U, V, W randomly and networks predicts the following probabilities, ok. So, let us understand what is happening. I fed it d as the input I have just started training. So, my U, W and V are all some randomly initialized weight matrices, right now, and so it has predicted this as my probability distribution, this is the predictions that I have got from the network.

And I also know what is the true probability distribution. What is the true probability distribution for the first time step? 0 1 0 0 and so on, right you can see it. Second times that is also 0 1 0 0; third is 0 0 1 0 and the last one should have been 0 0. So, given the situation and before I talk about learning algorithms, what is the first thing that I need to define? Objective function, right. So, what is the objective function here? How many errors do I have? I mean I can make my errors at 4 places, whether I making an error or not is the separate case but I can have 4 loss functions.

So, then these are the two questions that I am interested in. What is the total loss made by the model and how do we back propagate this loss and update the parameters of the model as usual I am ignoring the biases which is W, U and V . So, we can answer these two questions then we are done, right. If you can do this then we are done.

(Refer Slide Time: 03:53)

The diagram illustrates a recurrent neural network (RNN) unrolled over 4 time steps. Each time step t has an input U , a hidden state V , and an output y_t . The loss at each step is $\mathcal{L}_t(\theta)$. The total loss is the sum of $\mathcal{L}_t(\theta)$ for $t=1$ to T .

The diagram shows the flow of information and the calculation of loss at each step. The inputs are d, e, e, e . The hidden states are V . The outputs are $d, e, p, stop$. The loss at each step is $\mathcal{L}_1(\theta), \mathcal{L}_2(\theta), \mathcal{L}_3(\theta), \mathcal{L}_4(\theta)$.

The loss at each step is calculated as $\mathcal{L}_t(\theta) = -\log(y_{tc})$, where y_{tc} is the predicted probability of true character at time-step i .

The total loss is simply the sum of the loss over all time-steps:

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

For backpropagation we need to compute the gradients w.r.t. W, U, V .

Let us see how to do that

Mitesh M. Khapra CS7015 (Deep Learning) 25/41

So, the total loss, what is the total loss actually? Take a guess, sum of all the loss, right good. So, just going to be the sum of the loss over the times steps that you are I mean very logical and what else would it be. And we know that the loss at every time step is, so this is the loss at time step t hence y_t . And what is c actually? The true class at time step t , right. So, it is would be e at first time step, e at second time step, then p and then stop, ok, so that what c is.

So, this is we all comfortable with is this is the cross into p loss and I am going to sum at over all the t time setup that I have. Now, for back propagation what we need is we need to be able to compute the gradient of this loss function with respect to W, U, V .

(Refer Slide Time: 04:48)

Let us consider $\frac{\partial \mathcal{L}(\theta)}{\partial V}$ (V is a matrix so ideally we should write $\nabla_v \mathcal{L}(\theta)$)

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

- Each term in the summation is simply the derivative of the loss w.r.t. the weights in the output layer
- We have already seen how to do this when we studied backpropagation

Mitesh M. Khapra CS7015 (Deep Learning)

If I give you your formula for the gradient the rest is straight forward you will just apply gradient as well, ok. So, let us look at each of these parameters. We will look at the easy one first which is V . So, what is the derivative of the lost function with respect to V ? Have you ever done this in life?

Student: Yes.

Yes.

When?

Student: (Refer Time: 04:55).

Now, I am asking the date, ok. So, you have done this when you doing backpropagation. This is the gradient of the loss function with respect to the weights in the output layer and we know how to do that, right. That is very straightforward and there is no complication there. And you will see what I mean by complication later on.

So, all I need to do is take this loss function and compute its gradient with respect to V , it is very simple chain rule which I can update there, apply there and I can compute it separately for all these guys and I can just sum it up, right. So, this is the easy part. This is very straight forward. So, where one parameter we are all set, we know how to do that, right. We can just add up all these gradients, the some lose notation here this is actually

an addition of 4 matrices, right. Each of this I hope is a matrix, is that a matrix or a scalar or a vector or a tensor.

Student: Matrix.

Matrix, so, we have already seen how to do this back propagation. And this is a smallest chain possible in the back propagation and we have enough confidence in doing this.

(Refer Slide Time: 05:56)

The diagram shows a sequence of four loss functions $\mathcal{L}_1(\theta)$ to $\mathcal{L}_4(\theta)$. Each loss function is associated with a 4x2 matrix of predicted and true values. The diagram illustrates the backpropagation of gradients through the weight matrices W .

	$\mathcal{L}_1(\theta)$	$\mathcal{L}_2(\theta)$	$\mathcal{L}_3(\theta)$	$\mathcal{L}_4(\theta)$
	y_1	y_2	y_3	y_4
Predicted	0.2	0.2	0.2	0.2
True	1	1	1	1
d	0.7	0.7	0.7	0.7
e	0.1	0.1	0.1	0.1
p	0.1	0.1	0.1	0.1
stop	0.1	0.1	0.1	0.1

The diagram shows a sequence of four loss functions $\mathcal{L}_1(\theta)$ to $\mathcal{L}_4(\theta)$. Each loss function is associated with a 4x2 matrix of predicted and true values. The diagram illustrates the backpropagation of gradients through the weight matrices W .

- Let us consider the derivative $\frac{\partial \mathcal{L}(\theta)}{\partial W}$

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

- By the chain rule of derivatives we know that $\frac{\partial \mathcal{L}_t(\theta)}{\partial W}$ is obtained by summing gradients along all the paths from $\mathcal{L}_t(\theta)$ to W
- What are the paths connecting $\mathcal{L}_t(\theta)$ to W ?
- Let us see this by considering $\mathcal{L}_4(\theta)$

Mitesh M. Khurana CS7015 (Deep Learning) 27/41

Now, let us considered the derivative of the loss function with respect to W . Just take a minute and see if it is complicated or if it is straight forward to see a lot of W 's in the figure, ok. So, let us see how to do that, right.

So, again the loss with respect to W or the derivative with respect to the loss derivative of the loss with respect to W is going to just be the sum of these 4 or t derivatives. And by changed of derivatives we can just sum the derivative across all the paths which lead from the loss function to W , is that fine, right. Whenever, you want to compute the derivative of the loss function with respect to any parameter a recipes to look at all the paths which go from the loss function to that parameter and some of the gradients across those paths. How many if have fine with this? What are the paths which are actually connecting the loss function to W ?

Student: (Refer Time: 06:51).

There will be t paths, good. So, let us see we will consider L_4 theta, this is the last time step.

(Refer Slide Time: 06:57)

- $\mathcal{L}_1(\theta)$ depends on s_1
- s_1 in turn depends on s_3 and W
- s_3 in turn depends on s_2 and W
- s_2 in turn depends on s_1 and W
- s_1 in turn depends on s_0 and W where s_0 is a constant starting state.

Mitesh M. Khapra CS7015 (Deep Learning)

So, L_4 theta actually depends on s_4 , s_4 depends on what? W and s_3 . s_3 depends on what? W and s_2 , s_2 depends on what? s_1 depends on W and s_0 , always assume there is s_0 . What kind of a network is this? What kind of a function is this? What did I ask to revise? This is not an order derivative. What kind of function is this?

(Refer Slide Time: 07:19)

- What we have here is an ordered network
- In an ordered network each state variable is computed one at a time in a specified order (first s_1 , then s_2 and so on)
- Now we have

$$\frac{\partial \mathcal{L}_1(\theta)}{\partial W} = \frac{\partial \mathcal{L}_1(\theta)}{\partial s_4} \frac{\partial s_4}{\partial W}$$
- We have already seen how to compute $\frac{\partial \mathcal{L}_1(\theta)}{\partial s_4}$ when we studied backprop
- But how do we compute $\frac{\partial s_4}{\partial W}$

Mitesh M. Khapra CS7015 (Deep Learning)

So, we have an ordered network with I will give it to you and it is not be to. Say in an ordered network each state is computed one at a time, right. So, we will first compute s_1 , then we will compute s_2 because s_2 depend on s_1 there is no other way we can compute s_2 , then s_3 , s_4 and then finally, the last function.

So, now, we have the following situation that the derivative of L_4 theta with respect to W can be written using this chain rule which is the derivative with respect to s_4 , and then the derivative of s_4 with respect to W . And that is that looks manageable there is nothing fancy here or is it I see a lot if people that looks manageable, right; everyone is not (Refer Time: 08:01).

Student: (Refer Time: 08:01).

Even though you have done the assignment everyone is not; even though you have revise the assignment everyone is not (Refer Time: 08:06). So, this part we have already seen. This is not the tricky part. L_4 theta by s_4 is straight forward because it only depends on this V and so its fine that part we have seen. This is same as computing the gradient of the loss function with respect to the hidden layer. But now let look at the derivative of s_4 with respect to W .

(Refer Slide Time: 08:28)

- Recall that

$$s_4 = \sigma(Ws_3 + b)$$
- In such an ordered network, we can't compute $\frac{\partial s_4}{\partial W}$ by simply treating s_3 as a constant (because it also depends on W)
- In such networks the total derivative $\frac{\partial s_4}{\partial W}$ has two parts
 - Explicit** : $\frac{\partial^+ s_4}{\partial W}$, treating all other inputs as constant
 - Implicit** : Summing over all indirect paths from s_4 to W
- Let us see how to do this

30/41

What is s_4 actually? $\sigma(W s_3 + b)$. So, now, if I want to compute $\frac{\partial s_4}{\partial W}$ by let me just remove the sigma, right I mean we can always get back the nonlinearity. So, I want to compute $\frac{\partial s_4}{\partial W}$. So, it will just be s_3 , s_3 again.

Student: Depend on W .

Depend on W , right. So, that is the problem with an ordered network. In such an ordered network you cannot compute the gradient of a s_4 with respect to W assuming that s_3 is a constant, s_3 is not a constant its again a function of W and W is the parameter with respect to a computing the derivative, right. That is the problem here.

So, in such networks the total derivative has two parts, what are these two parts? How many if you have revise this? What are the two parts called? Explicit and where at least your language model should be fine at explicit, and what else can it be think on at least have that much smartness, either you do not read its fine. So, that is going to be explicit and implicit. What do we do in the explicit case? If you can read the slide we treat all the other inputs as constant, right. An implicit is summing over all the indirect paths from s_4 to W . So, let us actually try to derive this whole thing, right.

(Refer Slide Time: 09:38)

$$\begin{aligned} \frac{\partial s_4}{\partial W} &= \underbrace{\frac{\partial^+ s_4}{\partial W}}_{\text{explicit}} + \underbrace{\frac{\partial s_4}{\partial s_3} \frac{\partial s_3}{\partial W}}_{\text{implicit}} \\ &= \frac{\partial^+ s_4}{\partial W} + \frac{\partial s_4}{\partial s_3} \left[\underbrace{\frac{\partial^+ s_3}{\partial W}}_{\text{explicit}} + \underbrace{\frac{\partial s_3}{\partial s_2} \frac{\partial s_2}{\partial W}}_{\text{implicit}} \right] \\ &= \frac{\partial^+ s_4}{\partial W} + \frac{\partial s_4}{\partial s_3} \frac{\partial^+ s_3}{\partial W} + \frac{\partial s_4}{\partial s_3} \frac{\partial s_3}{\partial s_2} \left[\frac{\partial^+ s_2}{\partial W} + \frac{\partial s_2}{\partial s_1} \frac{\partial s_1}{\partial W} \right] \\ &= \frac{\partial^+ s_4}{\partial W} + \frac{\partial s_4}{\partial s_3} \frac{\partial^+ s_3}{\partial W} + \frac{\partial s_4}{\partial s_3} \frac{\partial s_3}{\partial s_2} \frac{\partial^+ s_2}{\partial W} + \frac{\partial s_4}{\partial s_3} \frac{\partial s_3}{\partial s_2} \frac{\partial s_2}{\partial s_1} \left[\frac{\partial^+ s_1}{\partial W} \right] \end{aligned}$$

For simplicity we will short-circuit some of the paths

$$\frac{\partial s_4}{\partial W} = \frac{\partial s_4}{\partial s_4} \frac{\partial^+ s_4}{\partial W} + \frac{\partial s_4}{\partial s_3} \frac{\partial^+ s_3}{\partial W} + \frac{\partial s_4}{\partial s_2} \frac{\partial^+ s_2}{\partial W} + \frac{\partial s_4}{\partial s_1} \frac{\partial^+ s_1}{\partial W} = \sum_{k=1}^4 \frac{\partial s_4}{\partial s_k} \frac{\partial^+ s_k}{\partial W}$$

So, this is what the total derivative looks like. All of you are comfortable with this, right. I mean this is all we have done this in the assignments I will not go into the theory and

all that. You should be comfortable if you have not revise, you have to be blamed sorry for that, but I cannot go into the details of that but I still derive the whole thing.

So, this is what it looks like. The plus here indicates that we are going to treat everything else as a constant and just take the derivative with respect to W . And then the implicit part would be this, they are going to sum across all the paths. So, this is a path, ok.

Now, here again we have a total derivative ds_3 by dW . So, what am I going to do for that? Again explicit and implicit; again I have this ds_2 by dW which is again explicit plus implicit, again ds_1 by dW is that fine and then this is finite because s_1 that is depends on s_0 which has no connection to W . So, this is what your entire formula looks like. Now, this, sum slide abuse of notation here because what is each of these actually? Scalar? Vector? Matrix?

Student: (Refer Time: 10:41).

S_4 is?

Student: S_4 actually vector.

Vector. W is?

Student: Matrix.

Matrix. The derivative of a vector with respect to a matrix is?

Student: Tensor.

Tensor. You cannot do this in your head is it; these 3 sentences is one after the other, ok. So, for simplicity what I am going to do is I am going to short circuit some of these paths, right. So, let us I will just tell you what I am going to short circuit. So, I am going to write just for ease of coming up with the generic formula. The first term I am going to write as this and this is fair because this is just one, right. The second term also is fine. The third term I am going to short circuit this path I am just going to write as ds_4 by ds_2 and then ds_2 by dW , and again I am going to short circuit these paths and just write it as ds_4 by ds_1 and then this.

The reason I am doing this then I can write it as a very simple summation, where I have s_4 by s_k , where k goes from 1, 2, 3, 4 and then I just have the explicit derivative of s_k with respect to W . Just (Refer Time: 11:51) this for the minute and not a minute actually just 10 second (Refer Time: 11:54). If you have any problems with this let me know, I will use my standard trick. If you do not understand this you will not understand anything afterwards, no one is falling for that, ok. Everyone is comfortable with this, ok.

So, we have a formula for $\frac{\partial \mathcal{L}_4}{\partial W}$, and we have dealt with the tricky situation where we have these multiple paths in an ordered network and hence we are to split into explicit and implicit derivatives. So, we have done all that (Refer Time: 12:20) math and you have come up with the simplified formula for this, ok. So, finally, this is what we have.

(Refer Slide Time: 12:23)

• Finally we have

$$\frac{\partial \mathcal{L}_4(\theta)}{\partial W} = \frac{\partial \mathcal{L}_4 \theta}{\partial s_4} \frac{\partial s_4}{\partial W}$$

$$\frac{\partial s_4}{\partial W} = \sum_{k=1}^4 \frac{\partial s_4}{\partial s_k} \frac{\partial^+ s_k}{\partial W}$$

$$\therefore \frac{\partial \mathcal{L}_i(\theta)}{\partial W} = \frac{\partial \mathcal{L}_i(\theta)}{\partial s_t} \sum_{k=1}^t \frac{\partial s_t}{\partial s_k} \frac{\partial^+ s_k}{\partial W}$$

• This algorithm is called backpropagation through time (BPTT) as we backpropagate over all previous time steps

NPTEL
Mitesh M. Khapra CS7015 (Deep Learning) 32/41

You noting it down, right?

Student: (Refer Time: 12:28).

[laughter] I do not see you noting it down, ok. So, now, let us look at $\frac{\partial \mathcal{L}_4}{\partial W}$, that is exactly what we have derived on the previous slide and that was a summation of t terms, and for as t is equal to 4, ok. And in general L t by, the this was for L 4 so in general if I want to do L t then it is going to be this which I am replaced by t , and this which I have replaced by this formula. Everyone is fine with this? What were this

means? Everyone is fine with this formula, right. This is generic formula with respect to any time step. The only thing is that on the previous slide we are derived with respect to s_4 , now I have just come up with the generic form, ok.

So, this algorithm is called backpropagation through time because now we have taken care of this ordered network and you have a way of computing this gradient, once you have this gradient your life is simple because now we can just supply the gradient descent update, ok. So, we have dealt with V , we have dealt with W and as the name suggest who will deal with U , you ok, fine. So, you will to find out what it is for U , ok. By its going to be something very similar, and I do not want to do it because that is not I mean going to be something very similar you can do it on your own. But I want to focus on something which is important.