Artificial Intelligence for Economics

Prof. Adway Mitra

Artificial Intelligence

Indian Institute of Technology Kharagpur

Week – 03

Lecture - 15

Lecture 15 : Deep Learning for Time Series Forecasting

Hello everyone, welcome to this NPTEL course on artificial intelligence for economics. I am Adway Mitra an assistant professor in Indian Institute of Technology Kharagpur and I am going to talk today about the applications of neural network especially deep learning for the specific purpose of time series forecasting. So, in the our previous few lectures we have been discussing with with the topic of learning from data we have discussed unsupervised learning and then supervised learning. Under supervised learning we have covered decision trees, linear classifiers and linear regression and then after that we came to the topic of non-linear classification or non-linear regression using neural networks. and we saw that like we can have deep neural networks by adding lots of these hidden layers to it. So, like a neural network is essentially used for as a function approximator which function now we have already discussed when we are doing any supervised learning task we need a function which maps from the feature space to the label space.

Now, in case of linear classifier that function is assumed to be a linear function, but that is often not good enough for many real world applications. So, in that case we need a non-linear function and one good way of representing any arbitrary non-linear function is the neural network. In fact, if we can have a very deep neural network with like a very large number of hidden layers which are basically intermediate steps of computations, we can represent any arbitrarily complicated non-linear function. So,today we are going to discuss a few more concepts related to that we will first start with neural feature extraction using convolutional neural network, then we will go to what is known as a recurrent neural networks and their applications in time series forecasting.

We will also discuss some challenges of neural network and their potential solutions. So, coming to the task of neural feature extraction now in general it is often challenging to find suitable feature representations for complex observations such as images or time series. Remember that we like for any neural for any supervised learning task we have an like our assumption is that we have a feature vector using which we do the classification that is we make a transformation from the feature space to the label space. But the question is from where will this feature space come from. So, suppose we like if we are one way is manual estimation of features that is let us say we all have already decided an expert has already decided which all features to look for.

So, whenever a new example comes. there should be some automatic way of calculating all those features and those features are fed into the algorithm for classification or regression whatever it is. But in general it is very difficult to or in or in many cases especially if the data is something complex like an image it is difficult to for us to decide what kind of features we will like consider. So, like we of course, like these images come or these data comes in huge volumes I mean there are typically millions of examples. So, it is not possible for a human to individually set and notice like what all features or to I mean the manually estimate various features which may be useful for that particular task.

So, we have to find some proxy some feature which will serve as proxy for those like for those things which a human would normally look for in an image. Now suppose you are you want to classify the image as like whether it belongs it is an indoor image or an outdoor image. So, if a human being is asked to do this they will look for certain kinds of objects in the image which are typically associated with indoor settings and also look for images or certain kinds of objects which are associated with outdoor settings. If the first set of objects are present, but not second set then we say that it is probably an indoor image and otherwise we may call it as an outdoor image. But this is possible because we are able to find those objects as humans, but if we are giving the image to a computer it will generally not find it easy to identify any of the objects.

In fact, object detection is itself a challenging task I mean it is a research question by itself. Then how do we construct the feature if we if the computer cannot find the important objects in an image, then how does it build a feature vector for the image for which the classification can be done. So, the traditional approach was to apply various mathematical filters that may highlight some important aspects of the image such as the edges in the image, corner points of the image, the texture in if it if we are talking about a time series instead of an image a time series may have peaks and troughs. So, identifying those peaks and troughs and things like that. So, basically like on the raw data we would apply certain kinds of mathematical operations which are known as filters and thosein the hope that the those mathematical operations will give high value if certain kinds oflocal local structures are present in the input and it will give a low value otherwise.

Now, whether the those these features are interpretable or not that is whether the filter actually gives a high value only when a particular structure which is of interest to us whether like whether that happens or not is a different question. So, often these are like very approximate approaches that is like we will that is for in a given circumstance we never know whether a particular filter will be able to detect a particular kind of structure we are interested in or not, but still we the in the hope that if we throw in a large number of filters that is an ensemble of filters some of them will be useful. So, that is the task which we like that is the approach which we used to use earlier for dealing with these kinds of complex data such as images or time series. But finding these kinds of suitable filters was a very tedious and non-intuitive task that is we never knew what kind of filters we should be using. What filter is essentially a mathematical operation which you carry out on the input.

So, what kind of mathematical operations it should be. Now, signal processing engineers they had a set of like mathematical operations which they frequently used for these kinds of tasks. But like as we move from simpler to more complex tasks these features were or these filters were often becoming inadequate. So, we needed more features like this. So, at this time neural networks came for to our rescue in neural networks can be used to extract a collection of what is known as neural feature maps.

So, these neural features so, like you will remember that in the last lecture itself we were talking about these intermediate values or inter with the which are calculated in the hidden layers of a neural network. So, these intermediate layer values like these we can call as neural features. So, like at like as we said earlier also at each layer at each intermediate or hidden layer of the neural network some calculations are being done. and like the results stored in every hidden layer can be considered to be some kind of a transformation of the input which can be arbitrarily complex. So, we can consider those as some representations of the input and though the hidden layers can be considered as some replacement of the kinds of filters which we had been looking for earlier.

The only difference is that we are not having to specify what kind of linear I mean what kind of filters they are. We just have to build the neural network and just specify the activation functions at the different level layers. So, as we know the activation functions represent some kind of nonlinearity. So, like if we throw in lots of layers with a variety of activation functions in we associated with them like we can hope that we can get a large collection of these kinds of features which like which are something like the filter outputs except that we are not having to specify the filters by themselves. So, like the we are in at each layer of the neural network we are doing essentially the same task, but with some difference that difference comes in one through the change of edge weights and second through the application of the of these activation functions which can be different in the different layers.

So, like these intermediate layers they give us a collection of representations of the input and so those can be considered as the representations of the features. So, these are known as the this task is known as the like the feature neural feature extraction and the collection of these representations which we get from the intermediate layers of the neural network they known as the feature map. Now, we like one such operation which we must talk about is the operation called convolution. So, convolution is a very important concept in signal processing and when we are using these filters to extract features from images or time series, convolution was an important attribute is a was a important characteristic of all those filters. So, now what is convolution? Convolution is basically like we have where we are look the task of looking for a pattern in the data that is.

So, let us say like we have an image. So, as you know an image is represented in the computer by a matrix which where the different entries of the matrix they represent the pixel values. That is low value of pixel means the low value of the matrix means a dark pixel and high value of that matrix element means a bright pixel. So, now suppose I am looking for a particular kind of structure I mean a small structure in the image. So, we can represent that structure by a small matrix like this like this one.

So, this is typically called as a mask. So, as you can see the different values present in are present in the mask. So, now what we are what we want is we are presented with a large matrix and we are interested to see whether this particular pattern is present anywhere within that large matrix or not. So, what we do is that we just slide this small matrix this 3 cross 3 matrix which is also known as the filter or the kernel over the entire matrix and see wherever we can get a good match with it. Now, how do we get a good match for that we carry out some function which is like very similar to the function which is the kind of calculations which is done at each neuron of a neural network.

So, that is called as convolution. So, what what is this convolution? So, like As I said like I have taken this 4 cross 4 image I could have taken a much larger image also, but for just to show I am taking 4 cross 4. So, as I said I am going to slide this mask above it. So, I will initially put this mask over these 9 elements and I will do some calculations. then I will slide it and put it like this will cover the next set of three cross three elements after that I will slide down and put it here and cover these set of nine elements and again I will slide to the right side and I will put cover these nine elements and so on and so forth So, when I am putting the mask on some of these like elements what am I doing I am carrying out an element by element multiplication and then adding up those results.

So, this 0 is multiplied by 1, this 0 is again multiplied by 0, this 0 is multiplied by 0 and so on and so forth. So, like so you can consider that these are the inputs and these are the edge weights. So, just like in a neural network all the inputs are multiplied by the edge weights and then the products are added up. So, here also we are doing the same thing. So, if you like you can just check the calculations.

So, here 0 is multiplied by 1 then 0, 0 is multiplied by 0 that is again 0, 0 is multiplied by

0 that is of course 0, then again 0 here we multiply 1 and 2 to get value of 2. again 0 is multiplied by 1 0 is multiplied by 0 0 is multiplied by 1 and again 2 is multiplied by minus 1. So, we there are only 2 nonzero values here 1 is 2 and the other is minus 2 if we add them up of course, we get 0. So, right now the mask is placed over these the in the over these 9 pixels. So, and the result of the operation which we carried out was 0.

So, I just put it in the center like this. So, this value the 2 cross the 2 by 2 element of the matrix is post is put set to 0. Similarly, after that as I said I will cover these 3 cross 3 values again I will slide it down and so on. Now, you would see that so, you can calculate the values for this position and this position. Now, when I am putting the mask here that is when I am covering these 3 cross 3 values you would see that there is actually a good similarity between the these values and these the values of the mask that is the pattern which we are looking for like the in the this matrix there is actually a pattern like this.

So, here we have 1, here we have 1, here we have 0, we have 0, here we have 1, here we have 0 that is a slight mismatch. Again there are the 0s are matching, the 2s are matching, here is a 2 in the mask we are looking for 1, again minus 1 minus 1. So, there is a descent match between the mask and this 3 cross 3. So, here if we carry out that operation which I just described you will get the value of 10 which is quite large. So, in this case in the previous case we had got the value of 0 which means that there is not much match or similarity between the mask and this 3 cross 3 sub image, but between this 3 cross 3 sub image.

quite strongly resembles the mass. So, we get a high output here. So, basically what we are doing is we are we have the entire input and we are looking for this particular pattern and wherever the pattern is found we are getting a high like high output. So, so the like this kind of operation called the convolution it can be implemented in a neural network with a certain kind of layer known as the convolutional layer. Now this convolutional layer is not any different from the neural the usual hidden layers of the neural network which we had seen earlier with some differences.

The there are sparse connections between the adjacent layer that is to say many of the edge weights are 0 and some of the edges between the two layer they have shared weight. that is the same weight value of the weight is shared across many of the edges which also reduces the number of parameters and this weight vector it has certain repeating structures and the these empty values as I said it is a sparse. So, the outputs can be useful for representation. So, let us just see what it looks like. So, let us say that the input is like this and the weight matrix is here.

So, like that is we input as you can see is 6 dimensional and the output is 4 dimensional. So, we are looking for a 6×4 matrix the weight matrix the in like as I said these 1 2 3 this seems to be repeating across this weight matrix like this. and then the this weight matrix is also sparse that is all of these are zeros. So, we have this repeating structure as well as lots of zeros in the weight matrix. Now you you can convince yourself that if we carry out this matrix multiplication that is if I multiply this this input vector by this weight matrix we will get an output which looks like this 5 0 17 6.

So, now my claim is that the output which we have got like this is actually the result of convoluting this input with this kind of a repeating structure. So, like so like you can actually see what we get like this. So, this repeating structure 1, 2, 3 it is a mask in this case in the previous example here we are considering a 3×3 mask in this case we are considering a 3×1 mask. that is not a problem. So, instead of our input is also a vector instead of a matrix.

So, now if we put the like mask here. So, again we do the element wise multiplication followed by the addition you would see that the result comes out to be 5. Next we as happens in convolution we shift the mask by one stage and we focus on the next 3 elements. and then the further 3 elements finally the last 3 elements and so on and the like you would see that the output we get is like this 5 0 17 6. So, note that this output is exactly the same as the output which we got in this case by doing the this matrix multiplication. So, like this is the calculation which is done in a neural network and it gives us an output and this output is identical to the output the output we get by carrying out the convolution.

So, we have the convolution mask we use that convolution mask to construct our weight matrix. So, the weight matrix as I said has the convolution mask as the repeating structure and the different rows and the other they are just set to 0. So, this is how we implement the operation of convolution in a neural network. So, a neural network which can do this convolution in many stages that is called as a convolutional neural network. Another important operation which is often done by a neural network for the purpose of neural feature extraction is called a pooling operation.

So, pooling operation is simply you like you have the input. So, you divide the input into blocks and from like which may be 2×2 blocks and so on. Now you just retain only one representative value for each of those blocks like here we have like we are dividing this 4 cross 4 into non overlapping blocks of size 2×2 like this these 4 these 4 these 4 and so on. Now, for each of these blocks we are calculating one representative value which is the maximum value here.

So, 1 1 5 6 the maximum value is 6. So, we are storing 6 here 2 4 7 8 the maximum value is 8 and we are storing 8 here 3 2 1 2 the maximum value is 3 I am storing it 1 0 3 4 maximum value is 4. So, I am storing it. So, I could also instead of considering like. So, non overlapping blocks I could also have considered overlapping blocks like this. And instead of considering the maximum value as the representative of a block I could have also chosen the average value or something like that.

So, like basically that is this is known as the pooling operation. So, the you have a large input you reduce it to a much smaller input by reduce breaking up the input into small small blocks and represent the only representative represent each block by only one representative value. So, our typical deep neural network it has many convolutional layer and usually each convolutional layer is followed by a pooling layer like this. So, you have the input you have like the first hidden layer is a convolutional layer which basically means that the this weight vector w 1 it is arrange it has that structure which we mentioned that is it is sparse and it has the repeating structure where the repeating structure is a particular convolutional kernel. So, the result we get in the first hidden layer is the result of applying this convolution kernel W a particular convolution kernel on the input layer.

The results is of course, a new transformed version of the input. now on that we apply the task of pooling and which means we are like again dividing this result into certain blocks and from each block we are storing only a representative value then after that we again apply the process of convolution and get some result and again apply the pooling So, this whole thing repeats in many rounds and then finally, we have a like a normal fully connected layer which means the this weight matrix W that is dense it is not a sparse not does it have any repeating structure and then we get the final output. So, this is what a typical neural network looks like. So, a modern neural network may have some hundreds of hidden layers like this and then in each hidden layer of course, there is some activation function also we have which we have not mentioned. So, like the we are talking about these neural features.

So, at like the values at each of these intermediate layers $h_1, h_2, h_3, \ldots, h_l$ each of these we can consider as some kind of a representation of the input some very complex representation of the input. So, like all of these together they consist of what is known as a feature map for the input. Now, I can use any of these feature like here only the last I mean the results of the final layer are being used for the actual classification task by this neural like thing. So, you can say that still here we are basically computing the representation of the input and like what we have got here is the actual feature vector of the input this is like the x which we are talking about this was the raw input and we have calculated the feature vector. Now, we are calculating the $f(x)$ and that $f(x)$ is the implemented in this case by this kind of alike a layer of a neural network I can have more layers also if that is needed.

So, like you can we can say that till this place the feature representation is taking place and finally, we have got a feature vector based on which we are then doing theactual classification or regression. So, this now this idea can be used for time series data also which frequently arises in the domain of economics. Now, what is the time series data? Time series data is basically a sequence of observations where each observation is marked with a time stamp. These observations can be either scalar or vector valued. like so here like this is a typical example of a time series in economics so here three economic quantities inflation output and volatility like they have been measured in a particular economic unit let us say in a country or a state or something like they are measured at different time points so these are different years 1990 91 92 every year these inflation the average inflation output and volatility in a country have been measured and they have been plotted like this.

So, for each of these quantities we are we have a sequence of observations and each observation corresponds to a time point namely these years. So, these are like we have three time series like this or we can say that it is a single vector valued time series that is for each of the times points we have a like a three dimensional vector of observations. We can look up upon it as either three time series or a single vector valued time series. Now, typically a time series has two three components one is the slow moving component or trend, one is a fast moving component or periodic component and the last is the random component.

So, like if you look at this blue plot for example. So, you would see that tip on a like at a very large scale or at a very coarse scale we see that in the initial stages it was typically have this quantity the was having high values we, but as time progresses it gradually seems to be coming down. So, in case of this rate time series typically it was having high values earlier later it has low values and then towards the end it again has high values. So, these are like these broad characteristics these are called the trends. Now, again apart from that there are like the cyclical the fast moving components or periodic components. So, we see that there are some rises and falls here and things like that.

And then there are some other components which are like purely random that is from one year to another there might be some changes which are not part of any either this period that long term trend nor is it part of the short term period, but it is simply a local effect. So, that is called as a random. So, it may have local peaks and troughs. So, like when suppose I want to do some kind of operation based on time series. So, for that we will need a represent a mathematical representation of the time series a feature.

So, how do we get a feature of the time series? So, like one is of course, using the convolutional neural network as we already discussed. So, by the way what are the kinds of operations which we may want to do on this time series. One is of course, the short term forecasting that is like just based on the previous few observations of the time series predict what will be the value of the time series at the next time point. Another is the long term forecasting apart from that I may want to classify the entire time series that is like you have an entire time series as input and you want to like label you put some kind of label on it.

So, that is the time series classification. and similarly we can use the clustering also we can let let us say we have time series of those economic quantities from different countries and we want to see which are the countries which are having similar character economic characteristics. So, in that case each country is represented by a time series like that. So, I can try I may to identify which countries are behaving in a similar way I may want to do a clustering based on the time series. So, for that again we need a distance measure between the time series. So, whatever it is we have to build some kind of a representation of the time series which with the help of suitable features which can be the kind of neural features that we were talking about.

So, let us specifically consider the time task of time series forecasting that is we have the past values of the time series till x_t and we want to predict the next value x_{t+1} or maybe the next few values x_{t+1}, x_{t+2} etcetera. Now, typically a time series exhibits a property known as auto correlation which means that the current values have some correlation with the past values. And this is the property which gives us some hope that if we look at the past values then maybe we will be able to predict the future values also. So, the simplest task is the auto regression that is the next value x_{t+1} is simply expressed as a linear combination of the past k values. So, where k is a particular window length it is a hyper parameter which the modeler can choose and then and then estimate these coefficients a_1, a_2 etcetera.

So, by minimizing a suitable loss function in the same way as that we do the linear regression. So, once we. So, so this is one simple approach, but here the idea is that the dynamics of the time series is available is can be found out from these raw values themselves I mean the observed values themselves. but it might happen that the values of the time series which we are observing is like the I mean the dynamic nature or the coming from some other aspect which we do not observe. So, like for this purpose we imagine something known as latent variables.

So, some variables which cannot be directly measured, but the temporal dynamics they take place at the level of those unobserved values and the observations which we see are just some local realizations of those latent variables. That is those latent variables we have a we can have a time series of the latent variables as well except that we do not observe them. So, the like the temporal dynamics is at the level of this latent variable that is x_{t+1} can be expressed as a like some function linear or non-linear of the past values of that latent variable. And the observation which we get at any time point t is a function of just the latent variable at that particular function. So, if we want to predict x_{t+1} we have to predict what is x_{t+1} will be.

So, which we can predict using the past values of this s and then we can have to make the like prediction of x_{t+1}. So, we need this function f and this function g. So, now how to get this. So, there are multiple approaches for doing this like earlier people used to go for a probabilistic model known as hidden Markov model, but now when neural networks are very popular and powerful we go with something known as recurrent neural networks. So, like here the hidden state dynamics is represented by a relation like this that is we have the past value of the of the hidden state.

the hidden state can be like again a vector or something like that. So, the past value of the hidden state is multiplied by a parameter called w and then the input at that particular time the x which like we can consider is the like the observation. So, here we are considering the observation as the input and we are trying to estimate the or we are trying to consider the dynamics at the level of the hidden values. So, like this is something like the inverse problem that is we like here in this model it seems that it is the hidden component which drives the input, but the hidden component is of course, not known to us only the I mean the observations are known. So, x is known. So, even though x is a function of s here we are considering s as a function of x that is the reverse process and the inverse problem.

and like if we can like understand or if we can model the dynamics of s then hopefully we can estimate the future values of x as well. So, this is how the recurrent the function represented by the new recurrent neural network. So, the this hidden value s its gets updated at every time point as new input new observations x come and like the at every point we may have an output also which is represented in this particular way . So, then when we are trying to train a recurrent neural network the task is essentially to estimate these parameters w and u. So, the train this the trained RNN can then be used to forecast the future values.

So, how will the training take place the basically we have to estimate the parameters w and u. So, again just like in a convolutional neural network or any neural network in general we have a loss function that is like which can be calculated with respect to some output or something else which the like which the neural network should be able to predict at the end. I mean and so that loss function again we calculate the derivative of these w_u etcetera parameters like this I mean we calculate the derivative of the loss function with respect to these parameters and like so like these parameter. So, now the like you can see that the same parameter w it is used at different time points. It is not that we have different values of the parameters for different time points.

So, like to estimate the derivative of the loss function with respect to these parameters we have to do the calculations at every time point not just any one of them. So, this requires the extensive application of the chain rule as we had discussed earlier. So, this the basic idea is the same as back propagation which we had discussed in the last lecture except that this in this case what we are doing is essentially back propagation through time. Now, this estimation of parameters is extremely difficult if the training sequence is long and we come across problems known as vanishing gradients and exploding gradient. So, what are these? So, I am not going into the details of what these are, but intuitively you can understand.

So, if let us say the time series is of length 100 and the final output is obtained after 100 steps. So, the loss function is measured after 100 steps. Now, when I am considering the or calculating the derivative of the loss functions of with respect to this w, I have to like take into account all the these 100 steps here. So, I have to calculate the derivatives at the 99 step, then at the 98 step, then at the 97 step and so on and so forth. And every time like I am basically applying the chain rule of differentiation at every step.

So, when I am trying to let us say the loss function is calculated after 100 steps, but I am trying to estimate the derivative after at the second or third steps. So, like I have like for that I have had to the chain which we have to calculate is really long as a long chain of multiplications from 100s to down to the second. So, when I have that every chain like if the derivatives are small then after I multiply them so many times it will become as good as 0. So, that is the vanishing gradient problem on the other hand if the gradients are large then after I do so many multiplications it may just blow up it can go towards infinity that is the exploding gradient problem. So, which basically means the early parts of the neural ah the observations of the time series they play very little role in estimating the parameters of the neural network which is which can also be said like the early values are forgotten.

So, to get rid of this problem we have a more sophisticated version of the recurrent neural network which is known as the long short term memory. So, here instead of having a single hidden representation we also had an additional variable called the cell state variable it is which we can consider as a long term memory while h is only the short term memory. So, this say this like this long term memory is used to store the early parts of the input and it is overwritten only if there is a forget signal that is if somehow it turns out that the. the earlier parts of the input are no longer important and they may be forgotten. So, where does these outputs signal this forget signal come from it also comes like is generated by operation like this in the neural network.

So, this is like these sets of operations is called as a LSTM cell. So, like it has the cell state and I mean which is the long term memory and the hidden variable which is like the short term memory and like it receives the input and as well. So, like here the input as you can see is coming from x_t you have the short term memory which is the hidden state as well as the cell state which is the long term memory from the previous step. So, here both of these are updated to c_t and h_t. So, in general c_t will be same as c_{t-1} unless forget signal has been applied that is the past values of c_t in the past values of the inputs are simply forgotten and over written and we can also have the output h. So, this is the long short term memory which is you consider to be more useful in representing long sequential data and instead like another way in which the more complex functions can be represented is instead of having one hidden layer you can stack many hidden layers together.

that is like as you as you can see in this way that is the hidden layer now I mean instead of having just one hidden layer we have can have three hidden layers. So, it gives more representative power to the neural networks. So, like in case earlier in case of convolutional neural networks also we were saying that this the intermediate or hidden states they are they are like the neural features which are having a basically representing the input. Here also we want to represent the time series input through various by like identifying various patterns. So, the patterns can be at various scales it can be at like using small windows of the time series or using the larger windows of time series using very large or entire windows of the time series.

So, for different kinds of patterns like this we can like we can have multiple hidden states or multiple LSTM cells which are stacked together. So, like these are the stacked models which can be used for like representing the time series it gives us more representative power. So, these time series forecasting this has many applications in economics as we can easily understand such as the like time series forecasting can be used for the forecasting the GDP of a country or the stock price of a company. or like we like we hear of credit ratings which are often given to certain agencies or firms or economies to say like whether it is like going to make good progress or its going to I mean its values are going to rise or fall.

So, that is a classification problem. So, usually ratings are given by agencies like standard and poor to strong to good to very good or very poor etcetera. then similarly clustering of time series we have already discussed now segments of time like segmentation of time series so let us say that we want to identify like we let us say different economic policies have been applied to a country at different points of time and we want to understand the impacts of those different economic policies So, we can look at the time series of the economic time series of that country with respect to different economic variables like GDP and we can look for segments within it. So, that is segmentation of time series and then we can understand whether a particular segment I mean let us say from year t_1 to t_2 the time series behaved in a certain way from t_2 to t_3 it behaved in a different way t_3 to t_4 yet another way. So, what happened at this time t_2 t_3 why did the behavior of the time series change was there a policy intervention at these points. So, these are the kinds of questions which we may be interested if you are trying to understand the like implications of policy and like.

So,these time series representations are very important in that case. So, to conclude neural networks can be used to create feature representations of complex data like images or time series. Convolutional neural networks simulate the convolution operation which is often used for traditionally for feature extraction. Future values of a time series can be forecasted based on past values because of the property called auto-correlation. Recurrent neural networks are useful for time series modeling and forecasting. However, they suffer from computational problems and may they tend to forget the earlier values in the time series to remedy this problem we have the long short term memory representation.

So, with this we come to the end of this lecture in the coming lectures we will or the coming few lectures we will deal with the topic of causality which is also another very important aspects of machine learning and especially when it is being applied to the domain of economics. So, like we will this start with Granger causality apply where we are trying to understand whether two time series are related to each other or not that is if we can predict one time series based on another so we will discuss that in the next lecture so till then everyone please stay well see you soon bye bye