Course Name: Machine Learning and Deep learning - Fundamentals and Applications

Professor Name: Prof. M. K. Bhuyan

Department Name: Electronics and Electrical Engineering

Institute Name: Indian Institute of Technology, Guwahati

Week-12

Lecture-43

Welcome to NPTEL online course on machine learning and deep learning fundamentals and applications. In my last class, I explained the architecture of unet and also I have highlighted one application of the unet that is the image segmentation. So for image segmentation, I can employ the unet architecture. So in this architecture, there are two paths. The first path is the contraction path. In the contraction path, the feature information is increased and the spatial information is reduced.

And in the expansion phase, the spatial information is increased and also the information of the contraction phase that is concatenated. So during the contraction phase, we can do the down sampling that is the repeated convolution and the pooling operation can be done. And during the expansion phase, the up sampling is done that is the up convolution. This is the structure of the unet.

Today, I am going to discuss the concept of another two popular deep architectures. So one is the autoencoder and another one is the RNN that is the recurrent neural network. So briefly, I will introduce these two architectures, autoencoder and the recurrent neural network. So let me begin this class. So in my last class in the discussion of the unet, I have explained the concept of the encoder and decoder.

So in this case, you can see I have the input image and during the encoding, I am getting the encoded representation that is the representation of the input image. So I can say it is the compact representation of the input image. During the decoding, the image is reconstructed from the information, the information is nothing but the encoded representation. So from the compressed information, the decoder reconstruct the original image. So it may not be same, but it tries to reconstruct the original image as far as possible.

So this is the encoder decoder structure. And based on this, I explained the concept of the unet architecture. So here I have shown the encoder decoder in semantic

segmentation that is the pixel based segmentation. So you can see already I have explained the architecture of the unet. It has two branches.

So during the contraction, we are doing the down sampling. So if you see this phase, so this phase is nothing but the contraction phase and this phase is nothing but the expansion phase. So during the contraction, the convolution and the down sampling operations, these are done to consider or to get the feature information, but the spatial information is reduced. And during the expansion phase, we are doing the up sampling that is the up convolution, so that I can increase the spatial information and also the feature information that is the high resolution feature information that is concatenated in this case. So how to do the concatenation that you can see here.

So here you can see this feature information is concatenated in this expansion phase. That is the structure of the unet architecture that can be explained in terms of encoder and decoder architecture. Now let us discuss the concept of the autoencoder. So autoencoders are artificial neural networks and they are capable of learning efficient representation of the input data. So that is the one important point, the efficient representation of input data and that is nothing but the coding and it is done without any supervision.

And the training set is unlabeled, so that is why it is unsupervised technique and this coding typically have a much lower dimensionality than the input data. So that is why the autoencoders can be employed for dimensionality reduction. So like the PCA, the PCA is employed for dimensionality reduction. In this case also autoencoders can be employed for dimensionality reduction. So during the encoding phase or I can say during the coding phase I am getting the efficient representation of the input data.

That is the compressed representation of the input data and from the compressed representation of the input data I can reconstruct the original input. So this process is very similar to PCA. So in the PCA from the input data I get the compressed representation of the input data and from the compressed representation I can reconstruct the original information. So that is the concept of the PCA. So this autoencoder principle and the PCA principle they are very similar.

So the concept is the efficient data representation. So the structure of the autoencoder it has two parts. So one is the encoding and another one is the decoding. So one is the encoder another one is the decoder. So encoder or the recognition network that converts the input to an internal representation.

So during the encoding the internal representation of the original data is obtained. That is the efficient data representation. After this during the decoding I can say it is the

generative network. So it tries to reconstruct the original data the original input. So that is the function of the decoding.

So this is the composition of the autoencoder. So one is the encoder and another one is the decoder. So what is actually the efficient data representation. So here I am giving one example. So showing two sequences.

So you can see it is the first sequence. So this is the first sequence. So two sequences we are considering. Now the question is which sequence is easy to memorize. So if I see these two sequences it looks like this the first sequence is easy to remember because it is a short sequence as compared to the second sequence.

So actually it is not true. So let us move to the next slide. So here again I am considering the same thing. So two sequences we are considering and the question is which is easy to memorize. If I see these two sequences so it seems like this the first sequence is easy to remember because it is shorter than the second one.

But if you minutely see the second sequence so we can find the rules like this. The rules are like this. Even numbers are followed by their half and odd numbers are followed by their triple plus one This is the famous sequence. The name of the sequence is the Hailstone sequence. So by considering these rules I can easily get the second sequence or I can simply memorize the second sequence based on these rules.

But in the first sequence there are no such rules. So if I have these rules for the second sequence I can easily memorize the second sequence. So only with the help of these rules I can get the second sequence. So that means the existence of a particular pattern and that is nothing but the internal representation of data. So that is the concept of the internal representation of data.

So with the help of these rules I can represent the second sequence. But the first sequence cannot be represented by the rules. So let us consider the structure of the autoencoder. So first one is the encoding. So during the encoding we can reduce the dimensionality of the input data.

So that is the concept of the encoding. So we can extract features during the encoding. So after this during the decoding I can reconstruct the original data or original input that is the approximate input I am getting that is the reconstructed input. So for this what I can consider maybe during the encoding I can consider a neural network with a fewer hidden nodes than the input dimensions. This is nothing but the MLP that is the multi-layer perceptron which can be employed for the compression of input data and after this again we can decode that is the decompression.

So this is like this. This is the multi-layer perceptron. So we are showing the input layer the layer L1, layer L2 is the hidden layer and layer L3 that is the output layer. So if I consider this structure so you can see the hidden layer has less number of nodes as compared to the input layer. So the hidden layer can learn the internal representation of the input data and because it has less number of nodes the dimension will be reduced that is the concept of the dimensionality reduction. After this from the compressed representation by considering the second part of the network so second part of the network that is the connections between the hidden layer and the output layer so I can reconstruct the original input.

The perfect reconstruction is not possible but I can reconstruct the original image. If I consider the image so I can reconstruct the original image but the perfect reconstruction may not be possible. So the first phase of the network that is actually the encoder and the second phase of the network that is actually the decoder. So it is a combination of encoding and the decoding. So one is the encoder and another one is the decoder.

So this is the deep autoencoder you can see instead of considering one hidden layer we are considering hidden layers between the input layer and the output layer. So this layer one layer that is the hidden layer that layer is called the bottleneck layer. So first we have to do the encoding so this bottleneck layer that actually learn the internal representation of the input data that is the encoding. So I am getting the compressed or the efficient representation of the input data. After this during the decoding the reconstruction can be done the reconstruction of the original data or the original image that can be done during the decoding.

So each hidden layer up to the bottleneck layer should be able to reconstruct the input from the previous layer. So this is the structure of the deep autoencoder you can see I am showing the combination of encoder and the decoder. So encoder for efficient representation of the input data and the decoder is for the reconstruction of the input data or input image. So this is nothing but the neural network the autoencoder is nothing but the neural network and the learning we can consider the learning is mainly the back propagation learning technique can be employed and this is nothing but the minimizing MSE the mean square error. And it is an unsupervised technique because no class levels are available and the sparse representation of data or features because we have a bottleneck layer and that means we are imposing some constraints on the hidden layer.

So that is why we are having the sparse representation of the data or the features. So hence the autoencoder can learn underlying features or the sparse representation of the data in an unsupervised manner. So that is the concept of the autoencoder. So we are having the sparse representation of the data input data and it is an unsupervised technique.

So you can see these are the examples. So we are showing the natural image patches. So we have shown the input and the target outputs in the first figure and the second figure you can see the feature detectors. So feature are detected by the hidden layers. So the hidden layer weights and corresponding to this the feature detectors I have shown. So different types of feature can be detected by the hidden layer.

So again I am showing another example that is the strokes for the handwritten digits. So input is the handwritten digits and the features will be the strokes. So these are the features. So features will be the strokes. The inputs are handwritten digits and the features are the strokes corresponding to this problem.

So to summarize autoencoder you can see the first one is the sparse representation of the input data. So that is why it is helpful for recognition and also it is complete because we can recreate the input that is the reconstruction. And also you can see already I told you this is an unsupervised technique. So more layers can be introduced that means their receptive fields increased and that is why this autoencoder can recognize more complex objects. So this is the summary of the autoencoder and in the figure I have shown the structure of the autoencoder.

The first figure you can see I have shown the input image that is the RGB image and we have the bottleneck layer and in the bottleneck layer I will get the latent space representation that is nothing but the compressed representation of the input image that is I can say the efficient representation of the input image that is the latent space representation. And after this from this information I can reconstruct the original image. The reconstruction may not be perfect but it is possible to reconstruct the original image from the latent space representation. So the principle is very similar to PCA the principal component analysis and in the second figure I have shown the reconstruction of the image. The input image has noises you can see this is the noise that is the specular reflection and we are considering the autoencoder.

So first part is the encoding part and the second part is the decoding part and we have a bottleneck layer. So during this encoding we are having the latent space representation and after this during the decoding I can reconstruct the original image that is the output image from the latent space representation. So if you see this example this noise is not important that is actually not the relevant information so only the important information I am extracting from the input image and accordingly I am reconstructing the image. So that is why in the output image the noise is not available. So we are having the efficient representation of the input image so that is why so I can get rid of the noise in the output image.

So that is why the autoencoder can be employed for noise removal. So this is one application of the autoencoder the noise removal or the noise filtering with the help of the autoencoder. So the same thing here I am showing the noisy input image and we are considering the encoding and decoding structure that is the autoencoder. So after the encoder I am getting the compressed representation of the input image. In the input image you can see the noises are available but we are having the efficient representation of the input image.

So that is why so the noise can be removed or noise can be deleted during the reconstruction by the decoder. So we are only considering the important information during the latent space representation. So in the network architecture in this figure I have shown you can see the input layer is available and we have the bottleneck layer that is the hidden layer and after this for the decoding I have the connections between the bottleneck layer and the output layer. So the first part is the encoding part and the second part is the decoding part. So why use autoencoders because the first application is dimensionality reduction.

So if I want to reduce the dimension of the input vector then simply I can consider the encoder I need not consider the decoder part. So with the help of the encoding part I can reduce the dimensionality of the input feature vector and autoencoders act as powerful feature detectors. So we can extract features from the input image or from the input data and already I told you this is the unsupervised technique and they are capable of randomly generating new data that looks very similar to the training data. So that is why I can consider autoencoder as a generative model.

So these are the important points of the autoencoders. So in this example I have shown one that noisy image because the autoencoder can be employed as a generative model and you can generate the new faces. The input is the noisy image and since we can extract important information from the input image that is nothing but the latent space representation or the efficient representation of the input image. So from this information I can generate new faces. So that is actually the decoding part. So encoding part is for the dimensionality reduction that is for the efficient representation of the input data and a decoding part is for the reconstruction.

So autoencoders looks at the input and converts them to an efficient internal representation. So that is for the encoding and then split out something that looks very close to the inputs that is nothing but the decoding and that is actually the reconstruction. So I am showing one example for the efficient data representations. So the example is suppose I am showing a chess board and you can see the positions of the pieces in the chess board. So any ordinary person he cannot remember the position of the pieces but one expert chess player he can memorize the positions of the pieces in the chess board because he knows the internal representation of the positions in the chess board.

So because of this he can easily memorize the positions of the pieces in the chess board but the ordinary person he cannot do this. So that is actually the internal representation of the input data. So in the figure you can see I have shown the inputs and we can get the internal representation of the data with the help of the encoding. So encoder I have shown that is nothing but the internal representation of the data and the decoder it can reconstruct the original image. So output I will be getting so that will be almost equal to the input that is the perfect reconstruction is not possible but we get the approximate reconstruction like the PCA the principal component analysis.

So you can see it is a simple structure we have the encoder and the decoder that is the autoencoder. So the composition of the autoencoder that is actually a multi-layer perceptron except that the number of neurons in the output layer must be equal to the number of inputs. So that is the condition. So in this case you can see I have three input nodes x1, x2, x3 and three output nodes x1 prime, x2 prime and x3 prime.

So three input nodes and the three output nodes. So autoencoder is nothing but the multi-layer perceptron except that the number of neurons in the output layer must be equal to the number of inputs. And you can see that there is just one hidden layer composed of two neurons. In this case I have shown one hidden layer that is the hidden layer it has two neurons. So that is the encoder and output layer composed of three neurons that is the decoder. So this is nothing but the multi-layer perceptron but slightly it is different because the number of input nodes should be equal to number of output nodes.

And in this case the encoder and decoder concept is considered that is for the internal representation of the input data and also for the reconstruction of the input data. So the outputs are the reconstructions since the autoencoder tries to reconstruct the inputs. And for this we can consider the cost function. The cost function may be the reconstruction loss cost function that we can consider. So the reconstruction loss function may be considered for the reconstruction and based on this I can train the autoencoder.

So because of the internal representation has a lower dimensionality because the dimension is reduced the autoencoder is said to be under complete. So it is under complete system and it is forced to learn the most important features in the input data and the drop the unimportant ones. So that is nothing but the internal representation of the data that is the efficient representation of the input data. So I will be getting the most important features and drop the unimportant ones.

So that is the fundamental concept of the autoencoder. So variance of autoencoder so there are different variants of autoencoder that means there are different architectures of

autoencoder. They are very popular but I am not going to discuss about these architectures of autoencoder. So one is the sparse autoencoder, stack autoencoder, variational autoencoder, the popular is denoising autoencoder that can be used for the denoising operation and the contractive autoencoder. So these are the variants of autoencoder.

So this is about the autoencoder. Next I am discussing the concept of the recurrent neural network. So briefly I will explain the concept of the recurrent neural network and also one or two applications of the RNN. So recurrent neural networks takes the previous output or the hidden state as input. The composite input at time t has some historical information about the happenings at that time.

So T < t. So that means we are considering the temporal information in the recurrent neural network because we are considering the past information. So past information is also considered in the recurrent neural network and that is why this can be employed for the temporal pattern recognition. So here you can see simple feed forward network I have shown. So at t = 1 the time is equal to 1. Input is x1 and after this I have the hidden layer and after this the output is the y1.

So this is the simple feed forward network. So we are not considering any past information in this feed forward network but in case of the recurrent neural network we are considering the past information. So let us see the structure of the simple RNN. So here I have shown three networks. So at t = 1 the input is x1 and we have the hidden node h1 and we are getting the output y1 but the input h0 is also considered that is the past input the past value it is considered. If I consider t = 2 that means the time t is equal to 2 the input is x2 and also the input from the previous hidden layer that is also considered.

So this input is also considered. So one is the x2 another one is the input from the previous step that is also considered and corresponding to this I am getting the output the output is y2. And similarly at t = 3 the input is x3 and corresponding to this network here you can see again the input is taken from the previous step. So that means we are having the temporal information in this structure that is the structure of the simple RNN recurrent neural network. So this is a popular RNN cell. So input at the time t is the x_t and w is the connected weight and h_{t-1} that is the output of the previous network that is the past information we are considering h_{t-1} and in this case tan hyperbolic is considered as the activation function.

So I am getting the output, output is the h_t that is nothing but we are considering the hyperbolic tan hyperbolic function as an activation function. So h_{t-1} is considered that is the previous output of the previous layer that is considered along with the current input the current input is the x_t. So you can see the structure of the vanilla RNN that is in the

forward direction. So input is if you see this network input is x1 and the previous output the output is h0 that is also considered and corresponding to this we can get h1 and after this from the h1 considering this activation I am getting the output, output is y1 and that corresponds to the class c1. If you see the second network the second network has the input the input is x2 and also it is getting the input from the previous layer that is the h1.

So it can compute h2 and from h2 we can get the y2 and that is nothing but the corresponding class c2 and if you see the third network so it has the input the input is x3 and also it is getting the input from the previous layer that is nothing but the h2 and corresponding to this if I apply this that activation function tan hyperbolic function like this I am getting the output, output is h3 and from the h3 we can get the output y3 and that is nothing but the c3 that is the corresponding class is the c3 and we are considering the sharing of the weights. So you can see we are considering the shared weights these are the shared weights in the output layer. So this is the structure of the RNN. So you can see here corresponding to a particular network it is getting the input from the previous layer or the previous network.

So that is the concept of the RNN. So note that the weights are shared over time so sharing of the weights over time and essentially copies of the RNN cell are made over time that is mainly the unrolling and the unfolding with different inputs at different time steps. So that is the concept of the temporal information. So RNN can preserve the temporal information. So if I consider the problem of the temporal pattern recognition this RNN can be used. So corresponding to this I am giving one example how RNN can be used for sentiment classification that is nothing but the NLP natural language processing.

So the problem is classify restaurant review. So I have to review the restaurant and in this case the response will be positive or the negative the restaurant is good or the restaurant is not good. So input will be like this multiple words one or more sentences. So inputs will be multiple words or one or more sentences that is in the NLP and output will be the positive response or the negative response that is the positive classification and the negative classification. So that is about the review of the restaurant. So input maybe like this the food was really good that is one input the chicken cross the road because it was uncooked.

So these are inputs and the corresponding to these inputs the response should be positive or the negative that is the review of the restaurant. So here also I have shown one structure. So we are considering this RNN structure the input is the sentence the food quality about the food quality and we are getting the outputs h1, h2, hn. So all the outputs I am getting h1, h2, hn and we are combining this and based on this it is inputted to a linear classifier and we have the output the output will be the positive response or

the negative response that is about the review of the restaurant.

So this is the concept of the sentiment classification with the help of the RNN. After this another application of the RNN that is the image captioning. So what is the image captioning given an image produce a sentence describing the contents that is the concept of the image captioning. So from the image I have to produce a sentence describing the contents of the image. So we can extract image features by the CNN the convolutional neural network and output will be multiple words or maybe we can consider sentence. Corresponding to this example if I have this image the captioning maybe like this the dog is hiding that is the captioning that is the image captioning that is the description of the content of the image.

So for image we can extract image features with the help of the CNN and the output will be the multiple words or maybe one sentence we can consider. So corresponding to this problem I can consider this network input you can see that is the input image and with the help of the CNN I can extract the features and after this we are considering the RNN and you can see here the output for output we are considering the linear classifier. So like this we can consider the that is the output the dog that is the output. So from the input image I am getting the sentence the dog is under the bed.

So like this I can get. So this is the structure of the RNN for image captioning. So RNN outputs these are the examples of the image captioning. These were demonstrated in the conference CVPR 15 this is a computer vision conference and you can see the examples of the image captioning. So like this a person riding a motorcycle on a dirt road. So this is one example of the image captioning two dogs play in the grass.

So these are some outputs corresponding to these inputs. So these are the image captioning examples and these are the scenarios that is the input output scenarios. So if you see the first one we have the single input and a single output that is nothing but the feed forward network. The second one is the single input and the multiple outputs. So this can be considered for image captioning for image captioning this network can be considered.

The third one is the multiple inputs and a single output. So already I have shown this example. So I can consider the problem of the sentiment classification. So for sentiment classification I can consider this structure multiple inputs single output and the multiple multiple multiple inputs and the multiple outputs. So that I can consider for the application of the translation that is the translation of the words and also it can be employed for image captioning. In this class I briefly explain the concept of the autoencoder and also the structure of the RNN the recurrent neural network.

So autoencoder is nothing but the combination of the encoder and the decoder. So during the encoding I am getting the efficient representation of the input data that is nothing but the compressed representation of the input data. I can say it is the latent space representation and during the decoding from the compressed data I can reconstruct the original image or the original information. So that is the concept of the autoencoder. The autoencoder can be employed for dimensionality reduction like the PCA.

So during the encoding I am getting the compressed representation of the input data. So that is why the concept is very similar to PCA. So I can reduce the dimension of the input feature vector with the help of the autoencoder and there are different types of autoencoders like the denoising autoencoder, sparse autoencoder. So these concepts are very important.

And finally I discussed the concept of the RNN. So briefly I explained the concept of the RNN. So it is not possible to discuss all the concepts in this 12 weeks course on machine learning and deep learning fundamentals and applications. So let me stop here today. Thank you.