Machine Learning for Soil and Crop Management
Professor Somsubhra Chakraborty
Agricultural and Food Engineering Department
Indian Institute of Technology, Kharagpur
Lecture 39
UAV and ML Applications in Agriculture (Contd.)

(Refer Slide Time: 00:22)

NPTEL ONLINE CERTIFICATION COURSES

Machine Learning for Soil and Crop Management
Prof. Somsubhra Chakraborty
Agricultural and Food Engineering Department
Indian Institute of Technology Kharagpur

Week 8: UAV AND ML APPLICATIONS IN AGRICULTURE

LECTURE 39

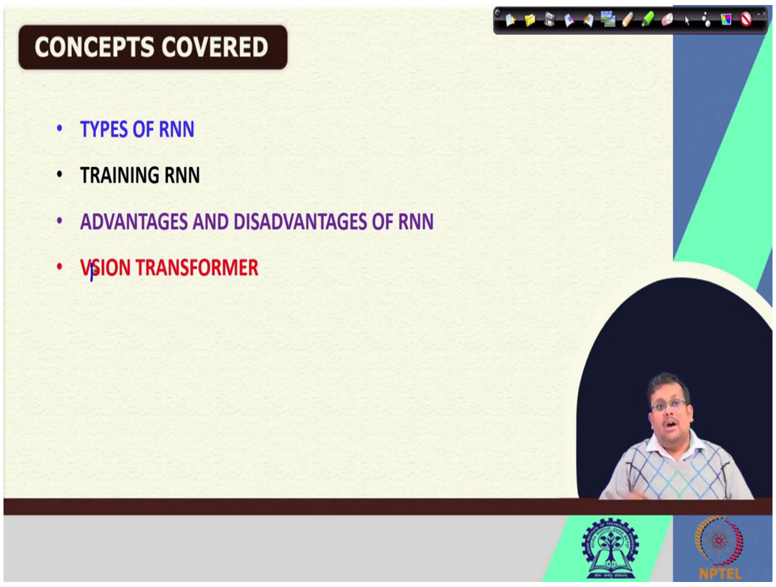
Welcome, friends, to this 4th lecture of week 8 of this NPTEL online certification course of Machine Learning for Soil and Crop Management. In this week, we are discussing different advanced machine learning, advanced deep learning methods like Convolutional Neural Network, their application for crop image processing, and also we have started discussing about the Recurrent Neural Network.

And then, what is the structure of the Recurrent Neural Network, how it works, how it differs from the traditional feedforward artificial neural network, how it calculates the error or adjust the error. Remember that this Recurrent Neural Network is a form of artificial neural network, is a variant of artificial neural network, where they execute the operation in a temporal manner, and the output of one step influence the next step.

So, it has a memory. Based on that memory of the previous output, it can decide on the next step. So, we have also we have already discussed how they can calculate what are the compressed representation as well as unfolded representation of Recurrent Neural Network in our previous lecture.

And we have also seen that what is the difference between Back Propagation Through Time for calculating or adjusting the error in case of Recurrent Neural Network, and how it differs from the and weight adjustment in case of feedforward artificial neural network, we have discussed. And we have seen with an example of idiom, and then we have seen that how this Recurrent Neural Network helps to identify the next output based on the previous outputs.

(Refer Slide Time: 02:50)



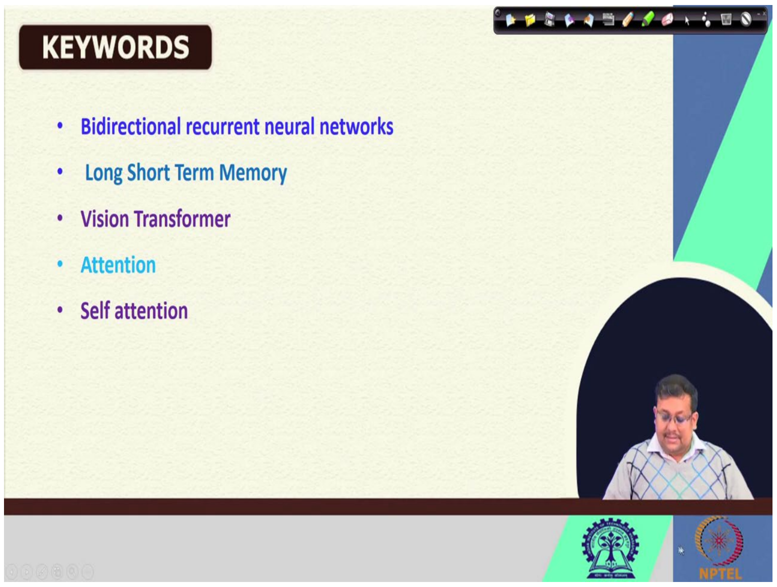
CONCEPTS COVERED

- TYPES OF RNN
- TRAINING RNN
- ADVANTAGES AND DISADVANTAGES OF RNN
- VISION TRANSFORMER

So, let us go, let us start from there and let us discuss more about this Recurrent Neural Network. So, in this lecture I am going to talk about these concepts. First of all, we will be discussing the what are the different types of RNN. We will see one example, and then we will see what are the steps for training and recurring neural network, and then will be will see that what are the advantages and disadvantages of Recurrent Neural Network.

And then we will start a new concept of contemporary deep learning technologies, this is called the Vision Transformer. So, it should be VIT so, vision transformer, we are going to discuss, and what is vision transformer and how it, what is the speciality of the vision transformer and how it differs from Recurrent Neural Network, we are also going to discuss.

(Refer Slide Time: 03:46)

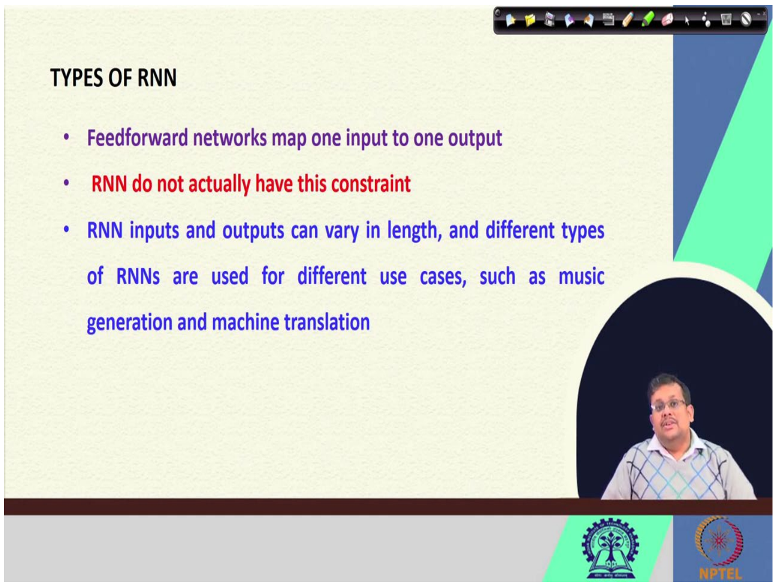


KEYWORDS

- Bidirectional recurrent neural networks
- Long Short Term Memory
- Vision Transformer
- Attention
- Self attention

Now, some of the keywords which we are going to discuss are Bidirectional recurrent neural network, then long term, Long Short Term Memory, then Vision Transformer, Attention, Self attention, So, we are going to discuss all these things.

(Refer Slide Time: 04:05)



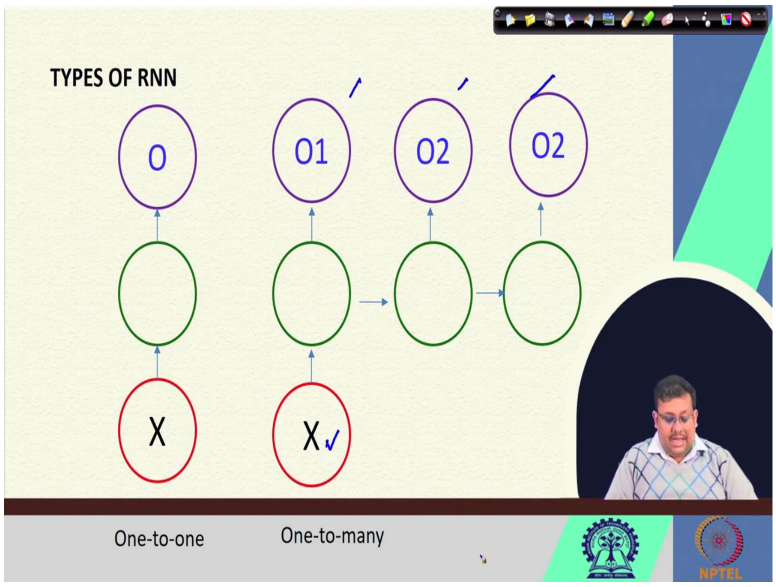
TYPES OF RNN

- Feedforward networks map one input to one output
- RNN do not actually have this constraint
- RNN inputs and outputs can vary in length, and different types of RNNs are used for different use cases, such as music generation and machine translation

So, let us first discuss there are several types of RNN. For example, this Recurrent Neural Network can be, can have different types of variant. One of these, we are going to discuss in upcoming slides, but at this point of time just remember that the feedforward networks generally, the traditional feedforward networks generally map one input to one output. However, the Recurrent Neural Networks do not actually have this constraint of one input to one output.

So, in case of Recurrent Neural Network, inputs and outputs can vary in length and different types of Recurrent Neural Networks are used for different use cases such as music generation and machine translation.

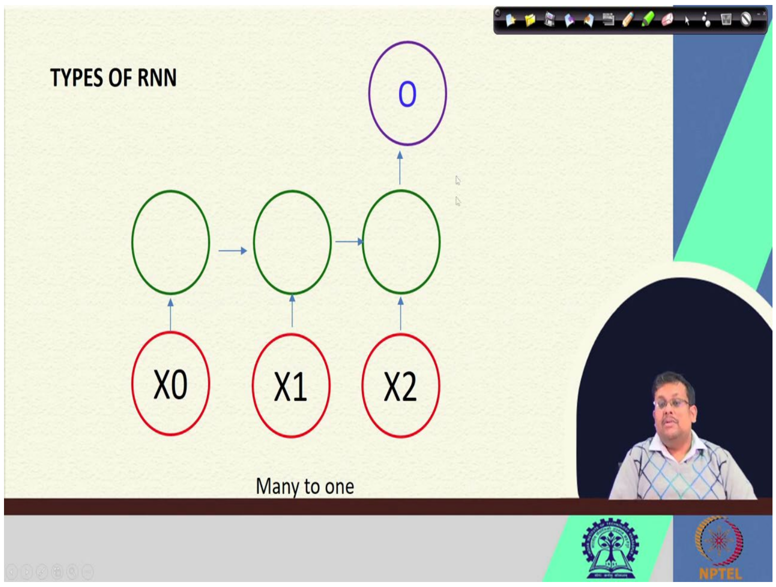
(Refer Slide Time: 05:07)



TYPES OF RNN

So, if you see one example here, the first one is called one-to-one, that means, there is one input, there is an hidden layer, and finally, we are getting the output. It could be one to many also where we can see there is one input, of course, as you can see, and there are multiple hidden layers and multiple output also. So, this is called the one-to-many Recurrent Neural Network.

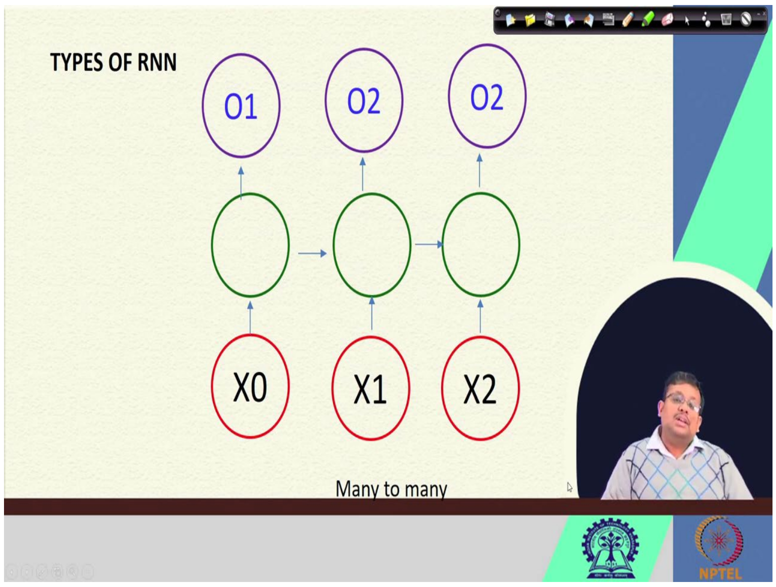
(Refer Slide Time: 05:38)



TYPES OF RNN

Now, there is another variant that is many to one. So, there may be many inputs and many hidden layers but there could be only one output also. So, this is another type of RNN structure.

(Refer Slide Time: 05:55)

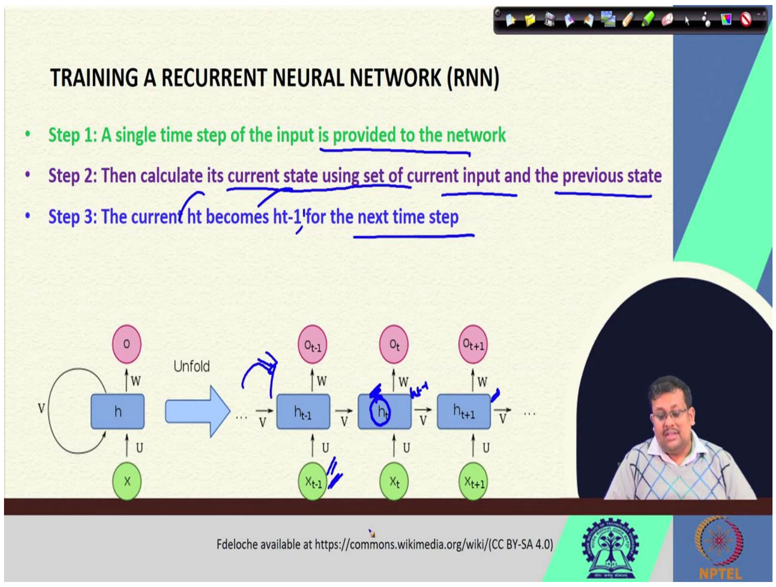


TYPES OF RNN

Many to many

And another structure of RNN is many to many. That means, here, you can see there are many inputs and many hidden layers and many outputs. So, here, you can see that there are different types of structures starting from one-to-one, one-to-many, many to one and many to many. So, these are some of the variants of Recurrent Neural Network, and Recurrent Neural Network is a widely accepted as well as popular method nowadays for deep learning. So, there are different types of application for Recurrent Neural Network.

(Refer Slide Time: 06:40)



TRAINING A RECURRENT NEURAL NETWORK (RNN)

- Step 1: A single time step of the input is provided to the network
- Step 2: Then calculate its current state using set of current input and the previous state
- Step 3: The current ht becomes ht-1 for the next time step

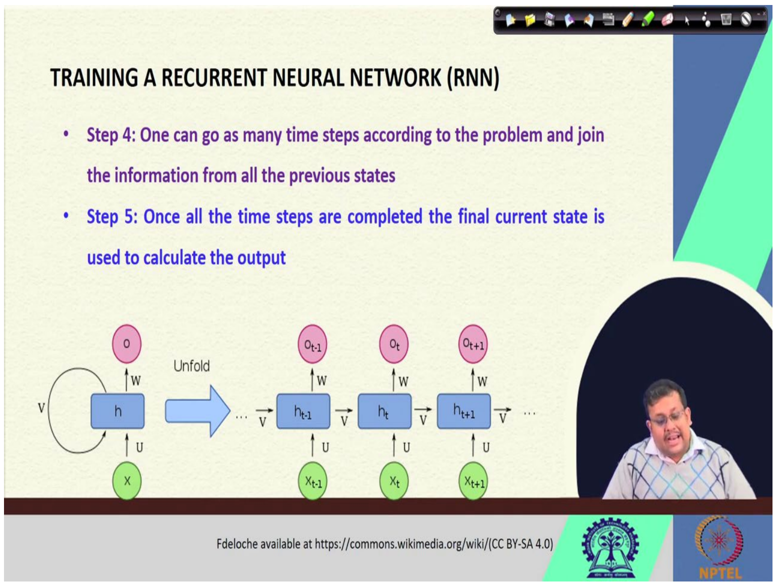
Fdeloche available at https://commons.wikimedia.org/wiki/(CC BY-SA 4.0)

Now, if we see what are the steps for training a Recurrent Neural Network. Now, we have a couple of steps. So, let us see what are those steps. So, first of all, in the step one what we will do, a single time steps of the input is provided to the network. So, here, we are providing a single time step of the input in the network. Then it calculates its current state using the set of current input and the previous state.

So, let us consider, that it calculate the current state using set of current inputs and the previous steps. So, from the previous steps it calculates and then the, suppose this current state becomes, the current h t becomes h t minus 1 for the next time step. So, of course, this current step becomes this step. Suppose this is denoted by this h t and this is h t plus 1. So, of course, this current state becomes h t minus 1 for this next time step because here h t plus 1, so, the previous steps.

So, again, in the first step, we are a single time step, the input is provided to the network and then it calculates its current state using the set of current input and the previous state, and the current h t becomes h t minus 1 for the next time steps because it is temporal changes. As you can see, these hidden layer are temporarily changing from one layer to another layer.

(Refer Slide Time: 08:32)



TRAINING A RECURRENT NEURAL NETWORK (RNN)

- Step 4: One can go as many time steps according to the problem and join the information from all the previous states
- Step 5: Once all the time steps are completed the final current state is used to calculate the output

Fdeloche available at https://commons.wikimedia.org/wiki/(CC BY-SA 4.0)

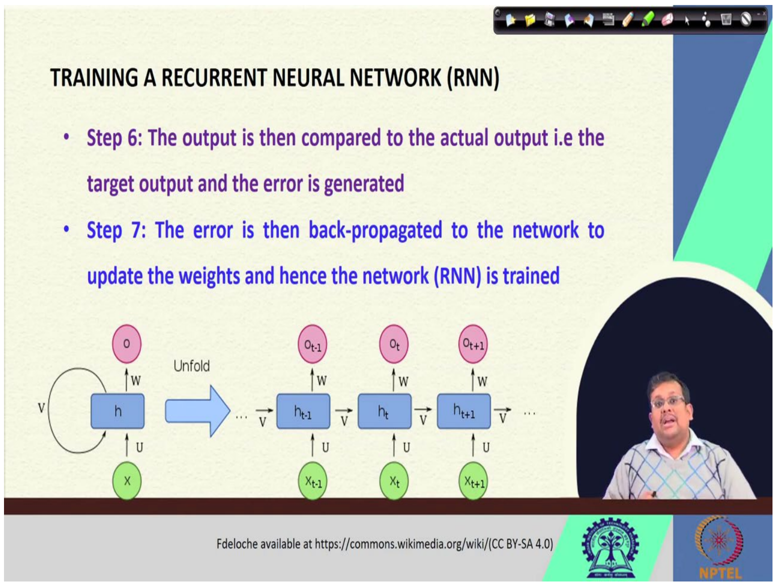
Now, we can see in the next step one can go as many as time steps according to the problem and join the information from all the previous steps. So, suppose, in the previous example which we were discussing earlier, this idiom, that is, “sit on the fence”, suppose from the first three layers we are getting the output of “sit on the”. So, of course, using these previous outputs this RNN can decide, and also with the given input, this RNN can decide the next output as the “fence”.

So, that means one can go as many as time steps according to the problem and join the information from all the previous steps. Once all the time steps are completed the final current state is used to calculate the output. So, in our case when each of these four layers will complete their output, then of course, the final current state is used to calculate the output. So, in our case the final output will be that particular idiom, that is, “sit on the fence”.

So, you can see here there are five steps. Again I am telling you, summarizing, first, we are putting an input, we are including an input in an individual single time step in the network, and then the network will calculate the current state based on the input as well as the output from the previous steps. Now, the current step becomes h t minus 1 for the next time step, and then it will go on and on until the problem is solved.

And finally, join the information from, as it moves in one direction, it will join the information it gathers from the previous steps because it has a memory. And finally, once all the time states are completed the final current state is used to calculate the output. So, this is how this RNN basically works.

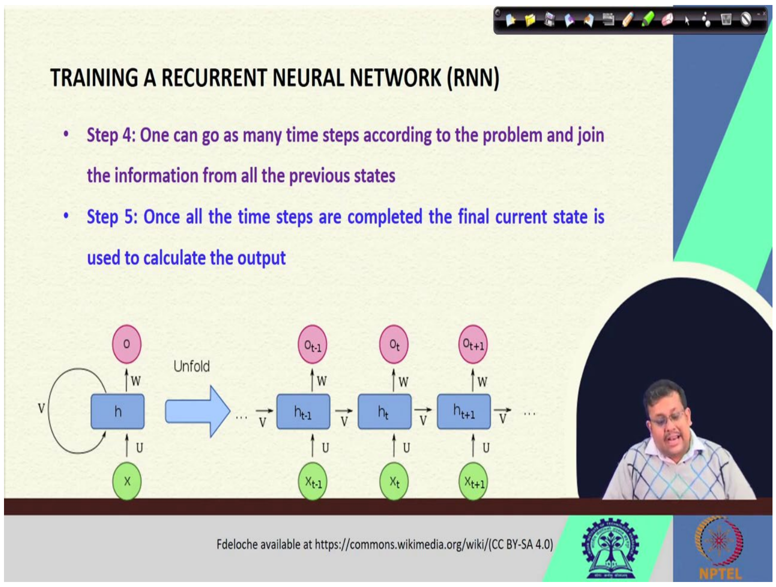
(Refer Slide Time: 11:00)



TRAINING A RECURRENT NEURAL NETWORK (RNN)

- Step 6: The output is then compared to the actual output i.e the target output and the error is generated
- Step 7: The error is then back-propagated to the network to update the weights and hence the network (RNN) is trained

Fdeloche available at https://commons.wikimedia.org/wiki/(CC BY-SA 4.0)



TRAINING A RECURRENT NEURAL NETWORK (RNN)

- Step 4: One can go as many time steps according to the problem and join the information from all the previous states
- Step 5: Once all the time steps are completed the final current state is used to calculate the output

Fdeloche available at https://commons.wikimedia.org/wiki/(CC BY-SA 4.0)

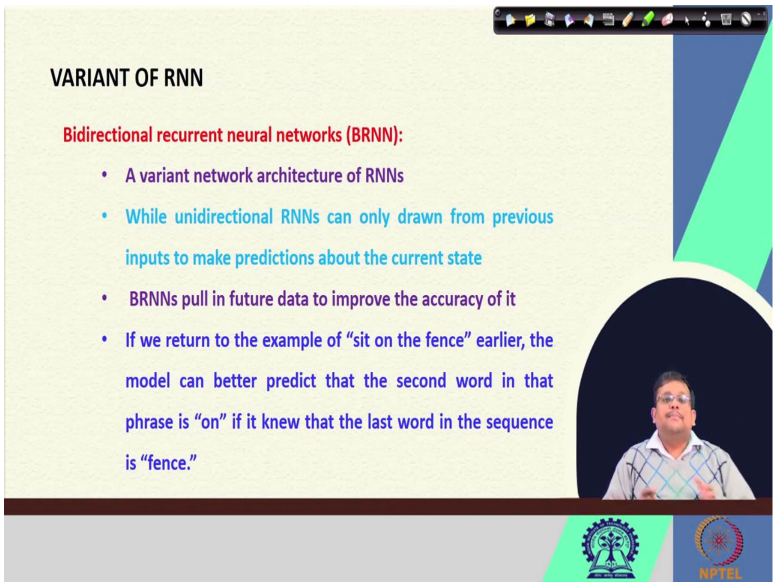
Now, in the next step, the output is then compared with the actual output that is the target output, and the error is generated. And then, the error is then back-propagated to the network to update the weights, and hence the network is trained. So, these seven steps are the steps for training a Recurrent Neural Network. Again I am summarizing, the first step, we are putting an input into a single time step of this network structure.

In the second step, it will calculate in this network, in this layer it will calculate the current state using this current input as well as the output from the previous steps and this current step will become h t minus 1 for the next step. So, that means it will progress temporarily. Now in the next step what will happen you can go as many as steps we can based on our problem.

And while we are moving through this temporarily we are remembering all the information or all the output So, that that those can be helpful for predicting the next output. Finally, once all the time steps are completed, the final current state is used to calculate the output. So, in the next step the output is then compared to the actual output that is the target, and the error is generated.

And in the Step 7, the error is then back-propagated to the network to update the weights, and hence the network is trained. So, this is how these things goes, this is how this Recurrent Neural Network is generated, is trained, actually.

(Refer Slide Time: 13:11)



VARIANT OF RNN

Bidirectional recurrent neural networks (BRNN):

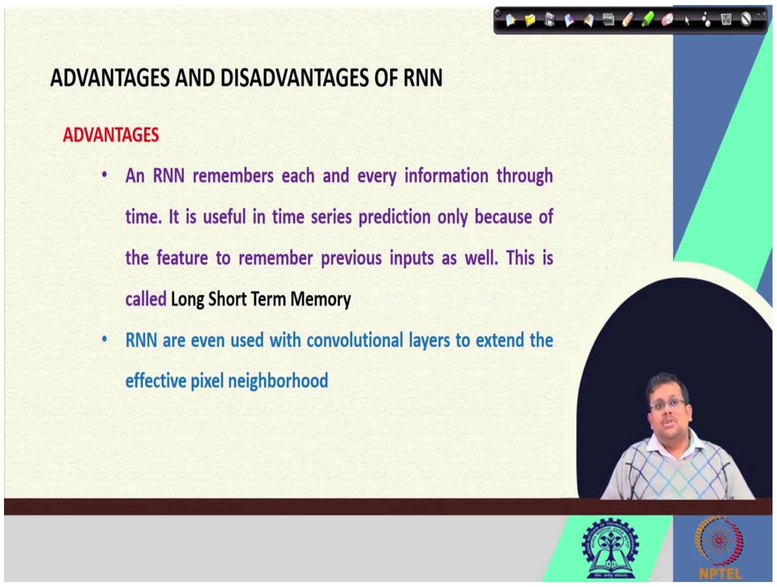
- A variant network architecture of RNNs
- While unidirectional RNNs can only drawn from previous inputs to make predictions about the current state
- BRNNs pull in future data to improve the accuracy of it
- If we return to the example of "sit on the fence" earlier, the model can better predict that the second word in that phrase is "on" if it knew that the last word in the sequence is "fence."

So, the next, let us see is there any variant of Recurrent Neural Network. So, of course, there is a variant of Recurrent Neural Network, we call it Bidirectional recurrent neural network. Now, what is it? So, a variant net, So, it is basically a variant of RNN, and we know that while unidirectional RNN can only drawn from previous inputs to make prediction about the current steps, the BRNN pull in future data to improve the accuracy of it.

So, if we return to the example of this “sit on the fence” idiom, which we discussed earlier, the model can better predict that the second word in that phrase is “on”, if it knew that the last word in the sequence is fence. So, here, you can see not only the model is gaining the information or memory from the previous step, but also they are also getting the information from the last word in the sequence, that is, “fence”. So, using these two both information, it can predict the output, that is, “the”.

So, the so, this is called the bidirectional, because here, the information flow is from both the direction, from the normal direction and also from the backward also. So, that is why we are getting, we are calling it is a Bidirectional recurrent neural network.

(Refer Slide Time: 15:02)



ADVANTAGES AND DISADVANTAGES OF RNN

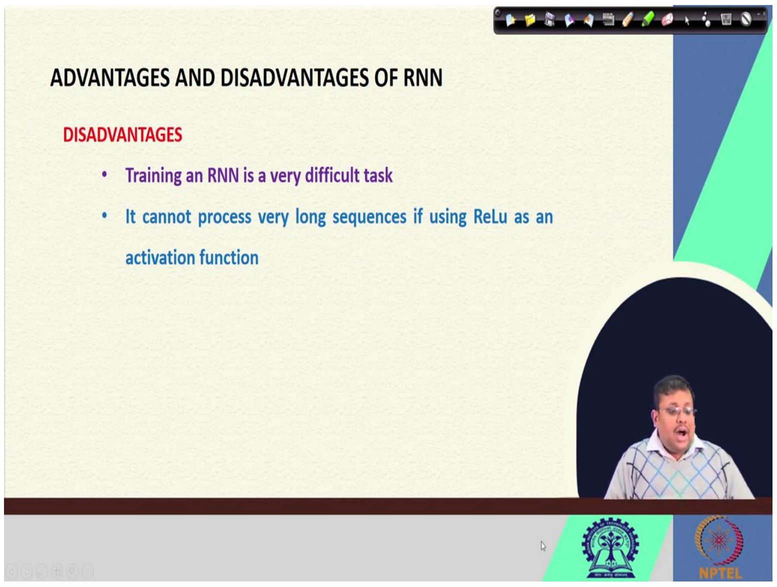
ADVANTAGES

- An RNN remembers each and every information through time. It is useful in time series prediction only because of the feature to remember previous inputs as well. This is called Long Short Term Memory
- RNN are even used with convolutional layers to extend the effective pixel neighborhood

Now, what are the advantages of Recurrent Neural Network? Remember that the advantage of an RNN members remembers each and every information through time and it is useful in time series prediction only because of the feature to remember previous inputs as well. This is called Long Short Term Memory. So, this is a very important another variant, that is called Long Short Term Memory.

And remember, RNN are even used with convolutional layers to extend the effective pixel neighborhood. Remember, in case of our CNN, we use the image process, we use for image processing, So, to improve the pixel, to augment the pixel neighborhood we also use this Recurrent Neural Network.

(Refer Slide Time: 15:55)



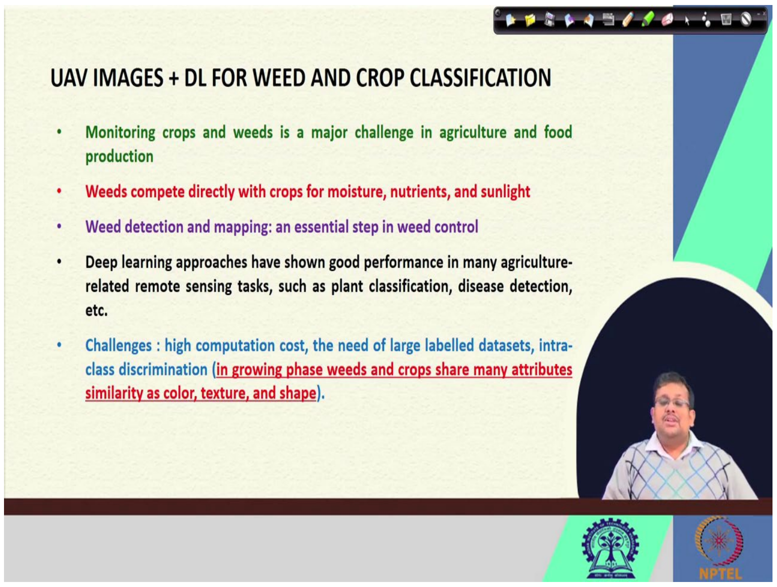
ADVANTAGES AND DISADVANTAGES OF RNN

DISADVANTAGES

- Training an RNN is a very difficult task
- It cannot process very long sequences if using ReLu as an activation function

What are the disadvantages? Of course, just like anything, it has also some disadvantages. First of all, training and RNN is a very difficult task. And secondly, it cannot process very long sequences if using this ReLu as an activation function. We have discussed about RelU in our previous lectures. So, these two are the disadvantages of the Recurrent Neural Network.

(Refer Slide Time: 16:21)



UAV IMAGES + DL FOR WEED AND CROP CLASSIFICATION

- Monitoring crops and weeds is a major challenge in agriculture and food production
- Weeds compete directly with crops for moisture, nutrients, and sunlight
- Weed detection and mapping: an essential step in weed control
- Deep learning approaches have shown good performance in many agriculture-related remote sensing tasks, such as plant classification, disease detection, etc.
- Challenges : high computation cost, the need of large labelled datasets, intra-class discrimination (in growing phase weeds and crops share many attributes similarity as color, texture, and shape).

So, guys, we have completed discussing about the Recurrent Neural Network. Now, we are going to actually enter into these machine learning based UAV image processing for crop application. And here, we will be focusing on crop classification. Now, we know all that monitoring crops and weeds is a major challenge in agriculture and food production.

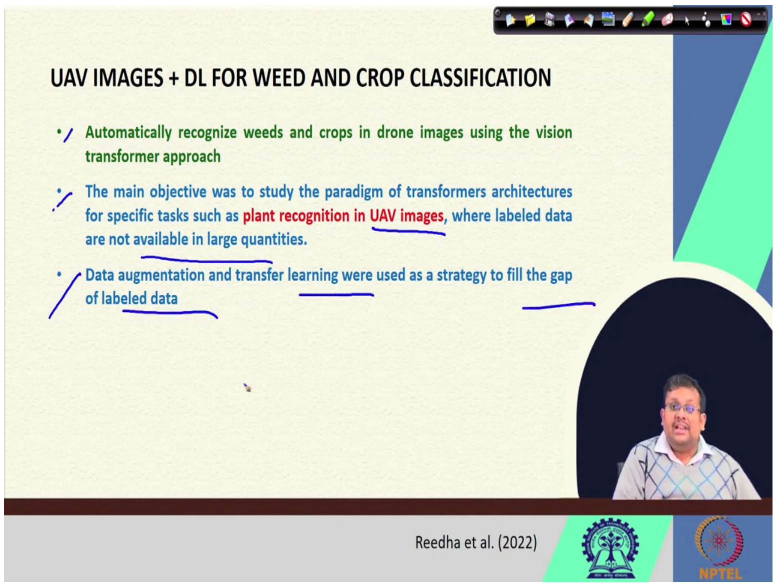
And weeds, generally, compete directly with crops for moisture, nutrients and sunlight. So, weed detection and mapping of course is an essential step in weed control. And deep learning approaches have shown good performance in many agriculture related remote sensing tasks, such as plant classification, disease detection, et cetera.

However, there are some challenges, when we use the deep learning methods. These challenges are high computation cost and the need for large labelled dataset because you need to have a robust data set for getting higher accuracy. Intra-class discrimination is also there.

That means, in growing phase weed, sometime we can see, we can get confused between weeds and crops because in the growing phase weeds and crops share many attributes similar, such as color their texture and the shape. So, it is very difficult to classify them particularly during the growing phase.

So, these are some of the traditional challenges and traditional problems we face while use the traditional deep learning methods in the, for weed detection in the field.

(Refer Slide Time: 18:18)



UAV IMAGES + DL FOR WEED AND CROP CLASSIFICATION

- Automatically recognize weeds and crops in drone images using the vision transformer approach
- The main objective was to study the paradigm of transformers architectures for specific tasks such as plant recognition in UAV images, where labeled data are not available in large quantities.
- Data augmentation and transfer learning were used as a strategy to fill the gap of labeled data

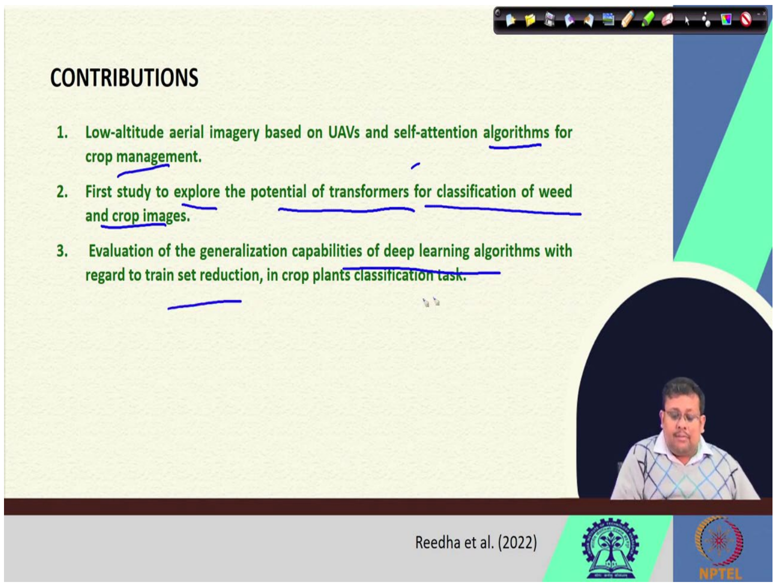
Reedha et al. (2022)

Now, these deep learning methods can automatically recognize weeds and crops in drone images using the Vision Transformer approach. So, there is a new research which is published in just this year in 2022, and in this paper, I will share this paper with you, you see that they have addressed some important aspects. First of all, they have recognized the weeds and crops in drone images using the Vision Transformer Approach.

And the main objective was to study the paradigm of transformers architecture for specific tasks such as plant recognition in UAV images where labeled data are not available in large quantities, and thirdly data augmentation and transfer learning were used as a strategy to fill the gaps of the labeled data. We have already defined the transfer learning.

Transfer learning means when while training a dataset, while training a dataset using a deep learning method you get some knowledge and that knowledge is being used for classifying or for executing another operation. That is called transfer learning. Now, in this paper, they have addressed these three novel aspects. And we can see that these are the three novel contributions in this paper.

(Refer Slide Time: 19:45)



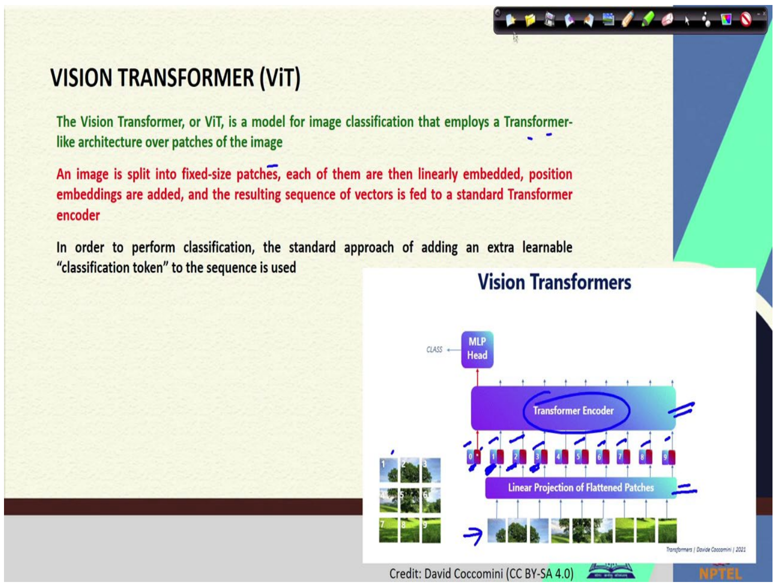
CONTRIBUTIONS

1. Low-altitude aerial imagery based on UAVs and self-attention algorithms for crop management.
2. First study to explore the potential of transformers for classification of weed and crop images.
3. Evaluation of the generalization capabilities of deep learning algorithms with regard to train set reduction, in crop plants classification task.

Reedha et al. (2022)

First of all, they used the low altitude aerial imagery based on UAVs and self-attention algorithms from crop management. Secondly, it is the first study to explore the potential of transformers for classification of weed and crop images. And thirdly, they evaluated the generalization capabilities of deep learning algorithms with regard to train set reduction in crop plants classification task.

(Refer Slide Time: 20:20)



VISION TRANSFORMER (ViT)

The Vision Transformer, or ViT, is a model for image classification that employs a Transformer-like architecture over patches of the image

An image is split into fixed-size patches, each of them are then linearly embedded, position embeddings are added, and the resulting sequence of vectors is fed to a standard Transformer encoder

In order to perform classification, the standard approach of adding an extra learnable "classification token" to the sequence is used

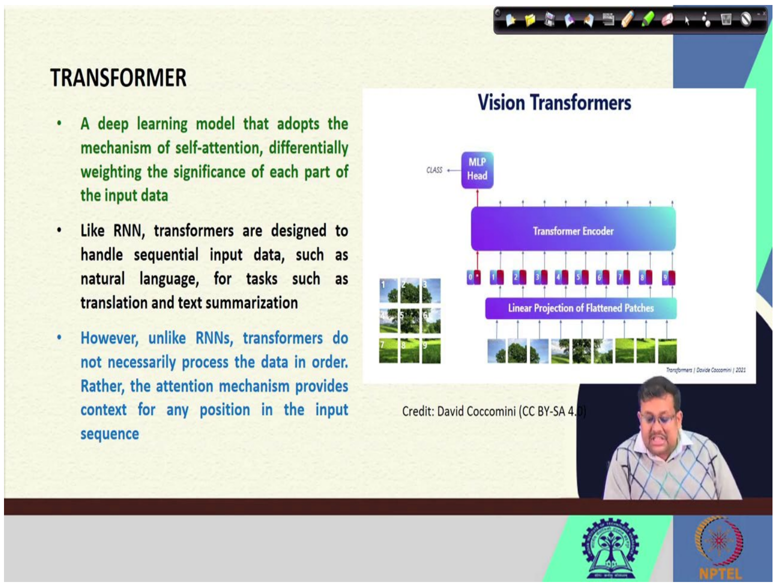
Credit: David Coccomini (CC BY-SA 4.0)

So, let, before we go to the details of this paper and their methodology, we will first discuss a very important topic that is called Vision Transformer. So, Vision Transform is also known as ViT. So, this vision transformer or ViT is a model for image classification that employs a Transformer like architecture over patches of the images. So, what happens, this is suppose an image and this image is split, first split into fixed size patches. So, you can see here, this image has been divided into nine patches.

So, each of them are linearly embedded. So, here, you can see each of them are linearly embedded, 1, 2, 3, 4, 5, 6, 7, 8, 9. And this position embedding are added, and the resulting sequence of vectors and then fit to a standard transformer encoder. So, this is a standard linear projection of the flattened patches, and you can see they are being coded here like 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. So, once these image patches are flattened, we are adding some labels and then they are fed to the standard transformer encoder.

Now, in order to perform this classification, the standard approach of adding an extra learnable classification token, you can see here, they are basically applying the classification, for classification they are applying this labels like 0, 1, 2, 3, and these, after we are labeling them, we are basically channeling them into this transformer encoder. Now the question comes, So, this is basically, and finally we are getting here, the output here, as the class.

(Refer Slide Time: 22:12)



TRANSFORMER

- A deep learning model that adopts the mechanism of self-attention, differentially weighting the significance of each part of the input data
- Like RNN, transformers are designed to handle sequential input data, such as natural language, for tasks such as translation and text summarization
- However, unlike RNNs, transformers do not necessarily process the data in order. Rather, the attention mechanism provides context for any position in the input sequence

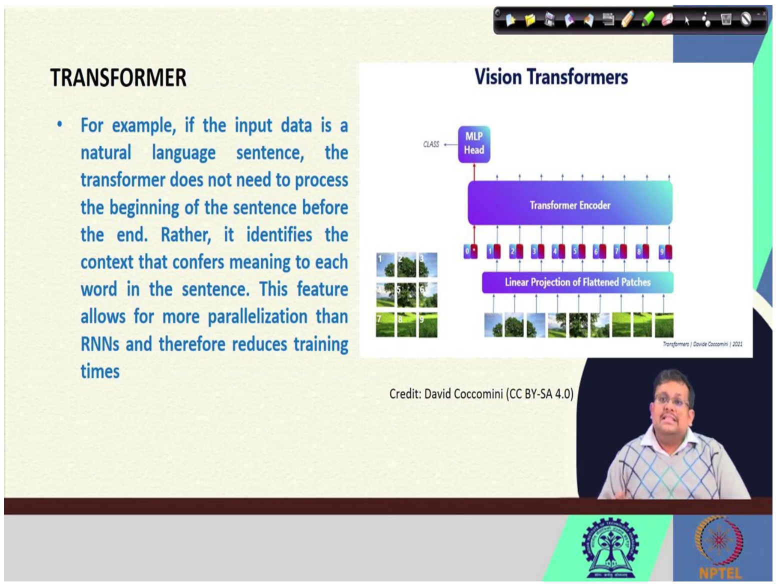
Credit: David Coccomini (CC BY-SA 4.0)

Now, the question is, what is this Transformer? Transformer is a deep learning model that adapts the mechanism of self-attention, differentially weighting the significance of each part of the input data. So, here, you can see, like RNN, the transformers are designed to handle the sequential input data. So, here you can see, we are sequentially putting the patches of the images here.

So, like RNN, these transformers are also designed to handle the sequential data such as neural natural language for tasks such as translation or text summarization. However, unlike this Recurrent Neural Network, transformers do not necessarily process the data in order. Rather, the attention mechanism provides context for any position in the input sequencer.

So, now remember, in case of Recurrent Neural Network, these events are happening recurrently. That means, they are following a certain temporal steps. However, in case of transformer, this mechanism is not happening necessarily in the order. Rather, they can attain this, this mechanism can happen from any position of the input sequence.

(Refer Slide Time: 23:37)



TRANSFORMER

- For example, if the input data is a natural language sentence, the transformer does not need to process the beginning of the sentence before the end. Rather, it identifies the context that confers meaning to each word in the sentence. This feature allows for more parallelization than RNNs and therefore reduces training times

Vision Transformers

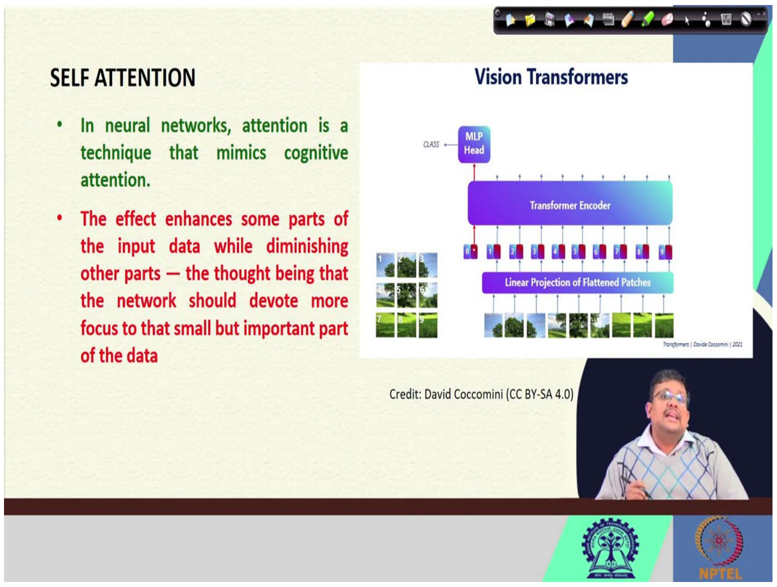
CLASS
MLP Head
Transformer Encoder
Linear Projection of Flattened Patches
Transformers | Davide Coccomini | 2021

Credit: David Coccomini (CC BY-SA 4.0)

Now, for example, if the input data is a natural language sentence, the transformer does not need to process the beginning of the sentence before, and the end. Rather, it identifies the context that converts meaning to each word in the sentence, and this feature allows for more parallelization than RNN, and therefore reduces training time. So, in case of RNN, we need to know the previous output to get the information or to guess the next word.

However, in case of transformer architecture, it identifies the context that converts the meaning to each word in the sentence from any word. So, it is not dependent on the beginning of the sentence before, there is no need to process the beginning of the sentence before the end. It can take any sequence of the data.

(Refer Slide Time: 24:41)



SELF ATTENTION

- In neural networks, attention is a technique that mimics cognitive attention.
- The effect enhances some parts of the input data while diminishing other parts — the thought being that the network should devote more focus to that small but important part of the data

Vision Transformers

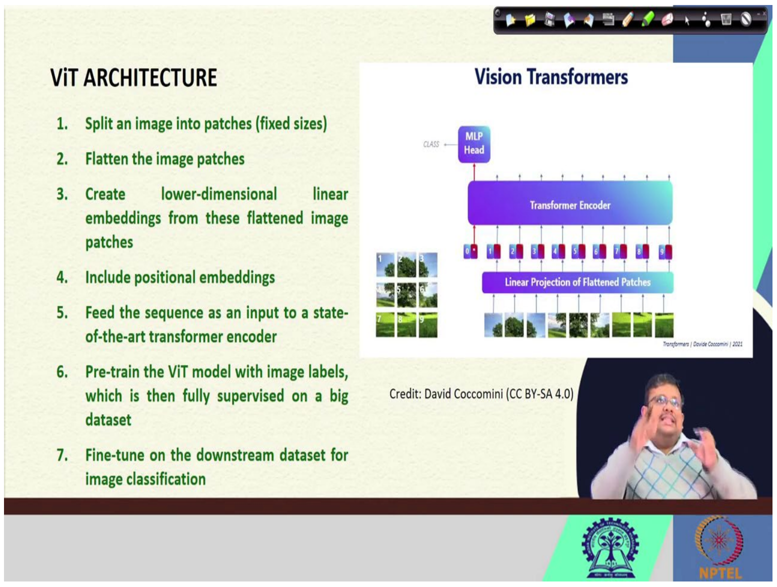
Credit: David Coccomini (CC BY-SA 4.0)

CLASS
MLP Head
Transformer Encoder
Linear Projection of Flattened Patches
Transformers | Davide Coccomini | 2021
NPTEL

Now, in neural networks attention is technic, there is a term called self-attention. I have told you here, that that is called the self-attention here. There is a, in the definition of the transformer, we have seen there is a term called self-attention. What is self-attention? Now, the self-attention is, in neural network, attention is a technique that mimics the cognitive attention.

Now, the effect enhances some parts of the inputs data while diminishing the other parts, the thought being that that the network should devote more focus or more time to the small but important part of the data without wasting the time to other unimportant features. So, this is called self-attention.

(Refer Slide Time: 25:30)



ViT ARCHITECTURE

1. Split an image into patches (fixed sizes)
2. Flatten the image patches
3. Create lower-dimensional linear embeddings from these flattened image patches
4. Include positional embeddings
5. Feed the sequence as an input to a state-of-the-art transformer encoder
6. Pre-train the ViT model with image labels, which is then fully supervised on a big dataset
7. Fine-tune on the downstream dataset for image classification

Vision Transformers

Credit: David Coccomini (CC BY-SA 4.0)

CLASS
MLP Head
Transformer Encoder
Linear Projection of Flattened Patches
Transformers | Davide Coccomini | 2021
NPTEL

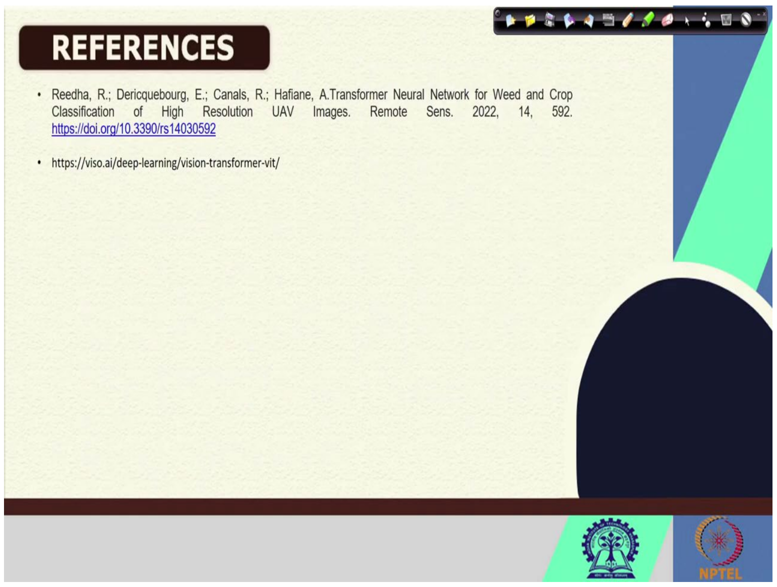
So, if we see the ViT architecture, we are coming back to ViT, Vision Transformer, so, if we come back to Vision Transformer, we can see there are seven steps. So, the first step is splitting an image into patches. So, we can see we are splitting the image into nine patches. Then, we are flattening the image patches. And then, we are, in the third step we are creating the lower dimensional linear embedding from this flattened image patches.

And then we are including the positional embedding. Then, we are feeding the sequence as an input to the state of the art transformer encoder. So, we are inputting this in the transformer encoder. Then pre-train a ViT model within each image levels, which is then fully supervised on a big dataset. And then, fine tune on the downstream dataset for image classification. So, these are the seven steps.

Again, we first split an image into patches of fixed size. Then we flatten those patches, then we create a lower dimensional linear embedding from these flattened images, we code them for, just like 0, 1, 2, 3, and so, on, and then we feed these in the transformer encoder, and in the transformer encoder, this pre-trained transformer encoder, we will provide the output.

And this pre-trained ViT model with image labels because now we have labeled the part of the image which is then fully supervised on a big dataset. Now, since we have already given them the class labels, that is 0, 1, 2, 3 up to here, up to up to 9, so, 1, 2, 3, 4,5, 6, 7, 8, 9, so, here, we have given that the labels. So, using these labeled images, they are basically fully trained and fine tuned, finally, fine tuning happen on the downstream dataset for image classification. So, this is how this ViT architecture works.

(Refer Slide Time: 27:40)



REFERENCES

- Reedha, R.; Dericquebourg, E.; Canals, R.; Hafiane, A.Transformer Neural Network for Weed and Crop Classification of High Resolution UAV Images. Remote Sens. 2022, 14, 592. https://doi.org/10.3390/rs14030592
- https://viso.ai/deep-learning/vision-transformer-vit/

So, guys, this is the reference. I hope that you have learnt something new. We have discussed a very important concept, that is, ViT, Vision Transformer. In the next lecture, we will start from here, and we will see how this group of researchers have used this Vision Transformer approach for predicting the weed using the UAV based images.

So, if you have any doubts, please feel free to write me the questions, and I will be happy to answer your queries. So, I hope that these things are now clear to you. We will meet in our next class, to move from here and discuss this whole experiment, and we will see the application of this UAV based image and machine learning and deep learning in more details. Thank you.