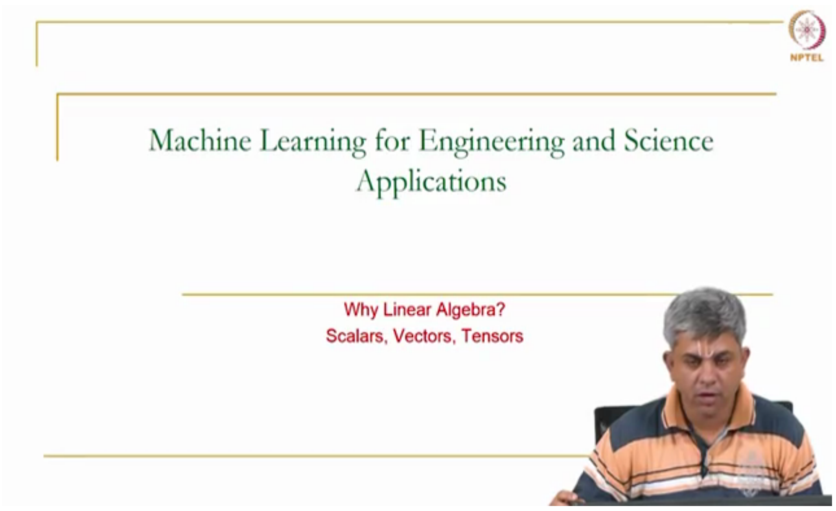
Machine Learning for Engineering and Science Applications
Professor Dr. Balaji Srinivasan
Department of Mechanical Engineering
Indian Institute of Technology, Madras
Why Linear Algebra? Scalars, Vectors, Tensors

(Refer Slide Time: 0:15)

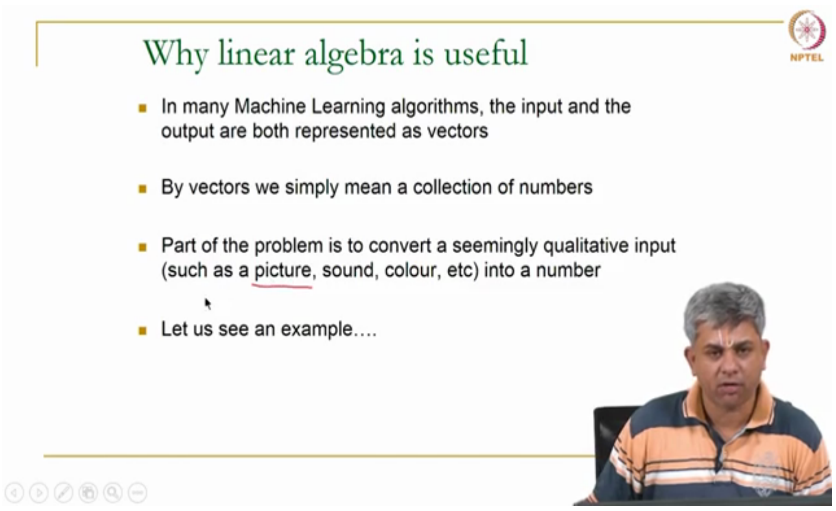


Machine Learning for Engineering and Science Applications

Why Linear Algebra?
Scalars, Vectors, Tensors

In this video, we will be beginning our mathematical excursion, we will start looking at the rudiments of the linear algebra which is needed for this course, as I said earlier linear algebra is a vast vast vast subject we are going to only look at very tiny pieces of linear algebra. In this particular video, I am going to look at just two simple things, why is it that we require linear algebra and also the basics of what scalars, vectors and tensors mean?

(Refer Slide Time: 0:48)



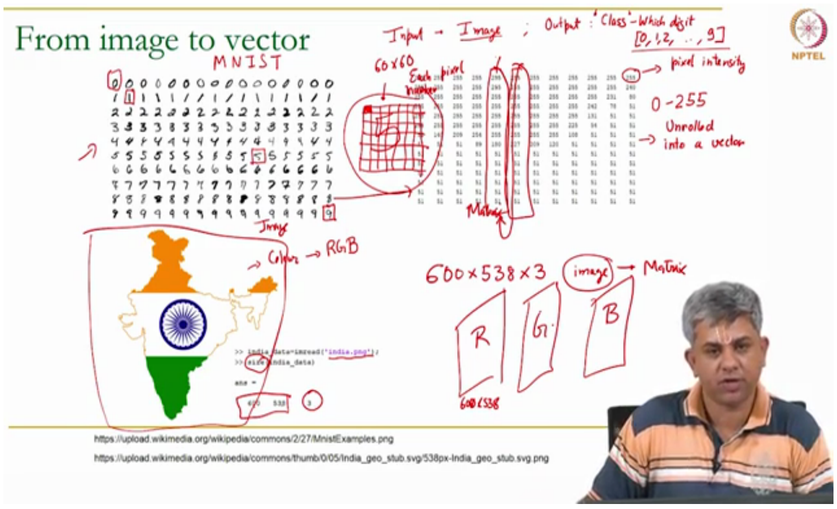
Why linear algebra is useful

- In many Machine Learning algorithms, the input and the output are both represented as vectors
- By vectors we simply mean a collection of numbers
- Part of the problem is to convert a seemingly qualitative input (such as a picture, sound, colour, etc) into a number
- Let us see an example....

Now, why is linear algebra useful in the context of machine learning, as I had said earlier, in many machine learning algorithms or in fact in most machine learning algorithms the input and output are both represented as vectors. By vectors we simply mean a collection of numbers. Now, part of the problem, is in machine learning as I said earlier is to convert what seems to be a qualitative input, for example a picture, a sound, you know colours, even sometimes smells, something of that sort into a number, because machines only understand numbers and our algorithms work only on numbers, they do not work on qualitative inputs, they work on quantitative inputs, okay.

So, let us see how we can do this? So, what I am going to show (in the next video) in the next slide is how we can actually convert what looks like a qualitative input, for example, I am going to look at a picture and let us see, how we can convert it into a number.

(Refer Slide Time: 1:48)


From image to vector
MNIST
Input → Image ; Output : Class - Which digit [0, 1, 2, ..., 9]
60×60
Each pixel number
pixel intensity
0-255
Unrolled into a vector
Matrix
Colors → RGB
600×538×3
image → Matrix
R
G
B
600×538
https://upload.wikimedia.org/wikipedia/commons/2/27/MnistExamples.png
https://upload.wikimedia.org/wikipedia/commons/thumb/0/05/India_geo_stub.svg/538px-India_geo_stub.svg.png
NPTEL

So, how do you take an image and turn it into a vector? Here is an example, you can see on your screens, whole bunch of 0s and 1s, these are hand written digits this database is called, the MNIST database, we will look at this in detail when we come to convolutional neural networks, but what it is, is just a bunch of images, what is shown here is actually a collection of images of 0s, 1s and up till 9.

So, the machine learning task that people usually deal with when it comes to MNIST database, is to look at an image let us say this, and identify which, out of those, 10 digits it is, okay. So, the input here, is an image and the output here, is what we will call a class or actually which digit this is, so you have 10 possibilities 0 through 9. So, now the question is how do you represent the input as a vector, or as a series of numbers, and how do you express the output as a series of numbers, the output is kind of obvious because it goes from 0 through 9, you can at least think of a single number coming out as an output that number is going to be either 0, 1, 2, or up till 9, but what we do with the input, okay.

So, the input usually looks like this, okay, something of this sort, okay. Now, we know, that this input, this image, that even you are seeing on your screens is actually dependent on the resolution of your screen and the way it is actually represented in the computer is actually through a series of pixels, so the machine has a whole bunch of pixels, let us say, in this case we have a 60×60 grid of pixels, so each of these pixels has a single value, associated with it, each pixel has a number, associated with it, typically this is what the numbers look like.

So, what you are seeing on your screen is some numbers that vary between 0 to 255, each of these corresponds to a pixel this is typically you can call it the pixel intensity, okay. So if you have a grayscale or let us say a black and white or grayscale image, what it will give you is something like 0 for a completely black pixel and it will ramp up to 255 for a completely white pixel, okay.

So, by giving a value between 0 and 255 and giving various values for various pixels you can actually reconstruct an image, this is the way the image is represented in a computer. So, this is a natural way of converting your image, this is the image, or this is actually a series of images, but you can represent it as a matrix, the matrix further if you wish, you can unroll it into a vector, what do I mean by unroll? Suppose, I take let us say, this was the first column, this is the second column, then I take this first column and have, 1, 2, 3, 4, up till 10 values, then after that I take this put it at the bottom, okay.

So, this way the whole of the matrix is unrolled into one single vector and usually we do, do that in machine learning algorithms because, it is actually easier to handle vectors, rather than matrices, but, it depends on which kind of algorithm you are using. So, you have a non-uniqueness in representation, you can take an image of this sort and either represent it, as a simple matrix of numbers or as a vector of numbers.

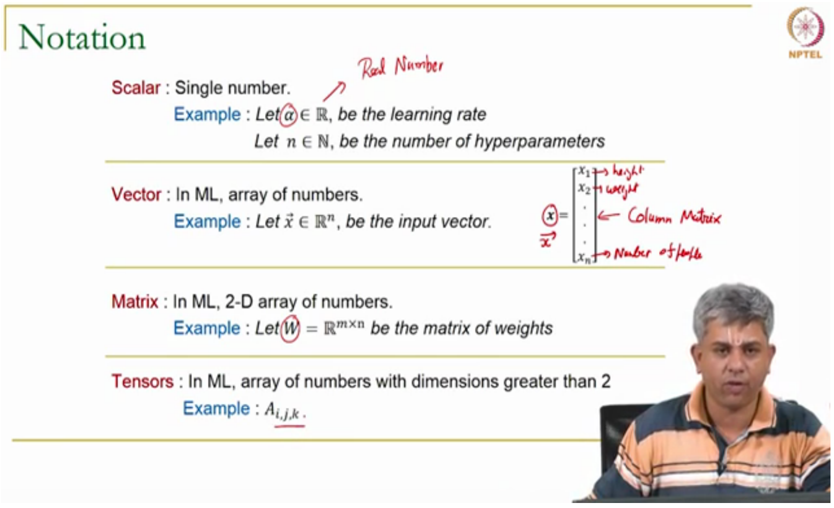
The more important point, here is to see that what looks like a number to us, or what looks like an image to us, can be actually converted into a full vector, okay. To give you, another example, you can see something like this, this is an image of India, this is a colour image, okay, as against the previous one this was a black and white image and this one is a colour image. Now, for a colour image you do not need just, you need not only a single set of intensities, but it is usually represented in, RGB, that is red green and blue.

So, you actually will have 3 sets of matrices, one of which will show you the red intensity of this image, one of which will show you the green intensity, and one of which will show you the blue intensity, and all these, put together give us the impression of a single image with varying colours, okay. We will look at this in greater detail, when we come to convolutional neural networks, but, for now you can see on your screen. You know, I have just a small MATLAB script. I took this image, this India image, and I found out what the size of this image was, you know, this kind of representation, what is the size.

What it shows is, the size or at least the, 2D size of this box has 600×538 pixels, but then there are 3 such layers that it has, so what we have is a, $600 \times 538 \times 3$ image, basically this is a matrix. So, this is a matrix of dimension 3, and in the first dimension there are 600 entries, second dimension 538, and the third dimension you have 3 entries, okay, all this put together essentially it is a stack, it is a stack of 3 images each of which are, 600×538, okay so you can have, RGB.

Okay, so the take away point from this particular slide is that you can take any image and you can turn it into a vector or a matrix of numbers, so please do remember this. So we will be dealing with such vectors throughout the course.

(Refer Slide Time: 8:40)



Notation

Scalar : Single number.

Example : Let $\boxed{\alpha} \in \mathbb{R}$, be the learning rate

Let $n \in \mathbb{N}$, be the number of hyperparameters

Real Number

Vector : In ML, array of numbers.

Example : Let $\vec{x} \in \mathbb{R}^n$, be the input vector.

$$\underset{\vec{x}}{\boxed{x}} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \begin{array}{l} \rightarrow \text{height} \\ \rightarrow \text{weight} \\ \leftarrow \text{Column Matrix} \\ \rightarrow \text{Number of [illegible]} \end{array}$$

Matrix : In ML, 2-D array of numbers.

Example : Let $\boxed{W} = \mathbb{R}^{m \times n}$ be the matrix of weights

Tensors : In ML, array of numbers with dimensions greater than 2

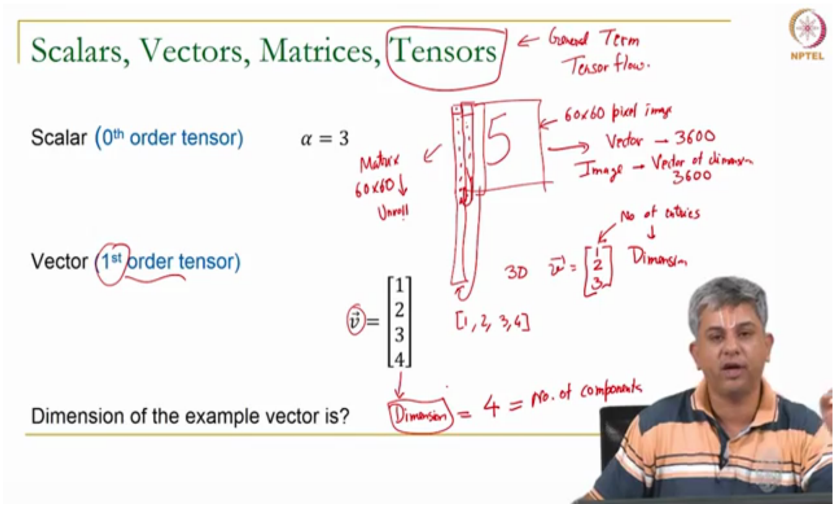
Example : $\underline{A_{i,j,k}}$.

So, let us now look at some simple notation, if you are familiar with the notation for matrices scalars, etc., you can skip this slide very easily, okay. A scalar is, you know, a single number we typically use small Greek letters for scalars, okay. $\mathbb{R}$, here is real numbers as you might know. So, an example is let us say α, is the learning rate, let us say, n is the number of hyperparameters, learning rate and hyperparameters, are things that we will come up later on in the algorithms that we use for machine learning.

A vector in machine learning is simply an array of numbers, okay. Now, typically in physics or even in hard core mathematics, vectors have very specific meanings, we are not looking at that we are looking at any, even if it is an unconnected series of numbers, for example, x_1 could be height, x_2 could be weight, x_n could be number of people. So, you can put together, any number of things together, all of those put together as long as you concatenate it into a column, we call it a vector, $\mathbf{x}$. So, please do remember this is a bold letter, bold small letters, we will typically use this for vectors or sometimes we might even use something like $\vec{x}$.

So, we use this kind of notation interchangeably, either we, use this or, we use the column matrix representation, for vectors, vectors will be the quantities that we are dealing with most often. Of course, you have the next level, it is a matrix, this again means that, $\mathbf{W}$ is a $m \times n$, matrix and it is simply a 2-D array of numbers. In general we use the term tensors for anything, which is a series of numbers with number of entries, or number of dimensions greater than 2, okay. So, in that case you will denote it by a subscript $A_{i,j,k}$, okay.

(Refer Slide Time: 10:58)



Scalars, Vectors, Matrices, Tensors

← General Term
Tensor flow.

Scalar (0th order tensor) $\alpha = 3$

Vector (1st order tensor)

$\vec{v} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$

Dimension of the example vector is? Dimension $= 4 =$ No. of components

60x60 pixel image
Vector → 3600
Image → Vector of dimension 3600
Matrix 60x60
Unroll
[1, 2, 3, 4]
3D $\vec{v} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$
No of entries
Dimension
NPTEL

So, let us look at this is some more detail, once again the same thing, all of these, whether it is a scalar, vector, matrix or tensor, all of these are effectively still the general term here is, tensor, okay, and it is because of this, use of tensors or multi-dimensional matrices that Google calls its package for machine learning as, tensor flow, which we will look at a little bit later in a couple of weeks.

So, now, scalar is nothing but a 0th order tensor, a vector is called a first order tensor. Now, I would like you to be a little bit careful, what is the dimension of this vector? Now suppose, you have a vector in 3 dimensional space, we will, typically, denote it with let us say, it is a location you would denote it with three numbers. So the number of entries, is what we call the dimension, okay.

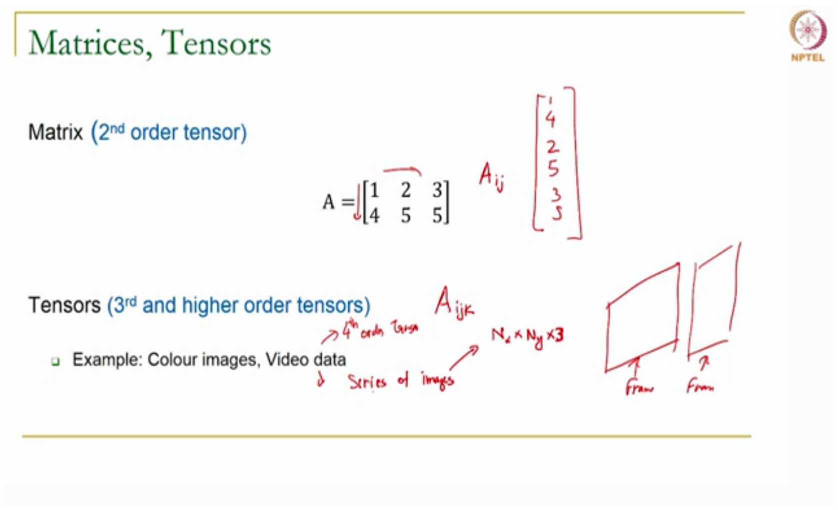
So, in this case, the dimension is 4, however the order of the tensor is 1, what I mean by the order of a tensor is you simply have one single column, if you had a column and a row then, it will be a second order tensor as we will see shortly, but notice, that the number of dimensions of this vector is equal to the number of entries, or number of components, that it has, okay, so this has 4 components, you could denote it by, $[1,\ 2,\ 3,\ 4]$, for example.

Now, if we go back and think about our image example, remember this example that we just had let us say, this is a 60×60 pixel image, and if I turn this into a vector, the way I would do it, is first I will turn it into a matrix, the matrix will be a 60×60 matrix, at this point it has both a row and a column, then I could unroll it, how would I unroll it? As I said you have first column, you have a series of numbers here, you have second column, you have another series of numbers here, you take this series of numbers, put it at the bottom, third column put it here, so on and so forth then you unroll it, the size of the vector is going to be 3600.

So, the point is that, the image, a 60×60 image can be written as a vector of dimension 3600. So, this is a huge number of dimensions, if we think, about it in terms of, the number of dimensions that we usually deal with in physics, in physics you are typically dealing with 3 dimensions, okay, so length, breadth, height, okay, x_1, x_2, x_3, xyz, coordinates, okay. So an image, a 60×60 image can be thought of, as a vector which has 3600 coordinates, so 3600 coordinates, each pixel, can be thought of as a coordinate.

Now, this kind of representation, is extremely useful as we will see throughout the course and I will talk about it briefly at the end of this video also, okay. So, please do remember this idea, you have the order of the tensor, which is simply the way you represent it, you also have the dimension, the dimension simply means, the number of independent components that you have in the vector, okay.

(Refer Slide Time: 14:53)



Matrices, Tensors

Matrix (2nd order tensor)

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 5 \end{bmatrix} \quad A_{ij} \quad \begin{bmatrix} 1 \\ 4 \\ 2 \\ 5 \\ 3 \\ 5 \end{bmatrix}$$

Tensors (3rd and higher order tensors) A_{ijk}

- Example: Colour images, Video data

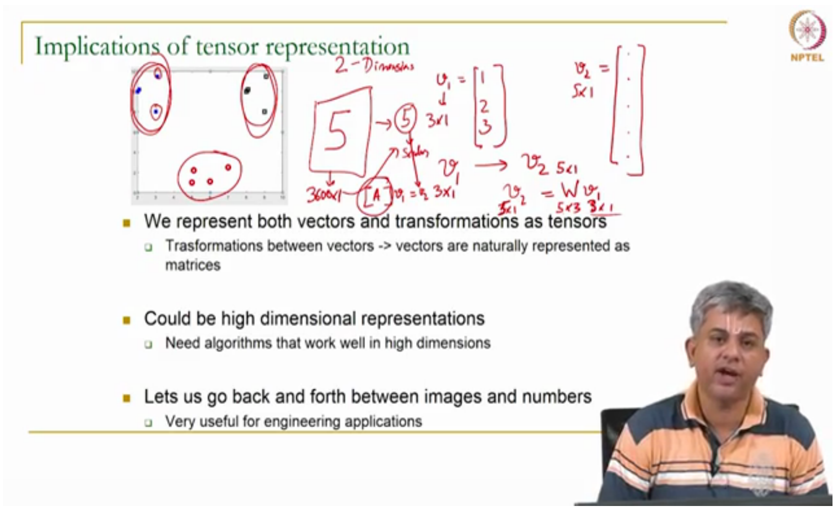
→ 4th order Tensor
→ Series of images
$N_x \times N_y \times 3$
Frame
Frame

We can move on, we have matrices this is a matrix, because it has, both the length and the breath and I would typically call this A_{ij} as a particular entry, if you want to think about dimensions, you can unroll this too, you can unroll this into, $[1\ 4\ 2\ 5\ 3\ 5]^\top$, or alternate ways, then you can think of this as a 6 dimensional vector, okay, you can even think about it this way. Now tensors are, third and higher order tensors effectively, basically you will have as I said earlier, A_{ijk} three sets of components effectively.

Now colour images, as I showed you, the India image, earlier it has naturally, got a tensor representation, okay, number of pixels in each channel multiplied by number of channels, okay. Video data, now you can think of that is interesting, because, now a video data is a series of images, and each image, has you know, $N_x \times N_y \times$ (let us say 3), so, each image is a 3 dimensional tensor, and you can now, think of the video as a series of images, each of these, is a frame.

So, this now becomes, a 4th order tensor, okay. So, colour videos, for example, naturally fall into 4th order tensors, and as usual, as you can imagine, they will have a huge huge huge number dimensions, because each image by itself has so many pixels.

(Refer Slide Time: 16:44)



Implications of tensor representation

- We represent both vectors and transformations as tensors
  - Trasformations between vectors -> vectors are naturally represented as matrices
- Could be high dimensional representations
  - Need algorithms that work well in high dimensions
- Lets us go back and forth between images and numbers
  - Very useful for engineering applications

Implications of this kind of representation, I will go back to something, I showed earlier, this is an example, of let us say pre labelled data, okay. Now obviously, I am showing this figure to you in 2 dimensions, but now you can think. Now, each of these crosses, could mean any number of things, okay. Now, suppose now, if you, you know, relax your imagination and think about it, suppose this was a 3600 dimension image, okay, you have multiple dimensions, and each of these crosses or each of these dots actually represented an image, okay.

So, then you could think, of this set of images being something, let us say cats, this set of images being something horses, this set of images being something let us say dogs, okay. What you would like is some way in which, similar, images in case you are doing a classification problem, similar, images land up at similar places, this is an implication of our turning everything into a number, is finally you can represent it in a graph, and then you can, start thinking about a classification problem, simply, as if you are doing a graphical partition, okay.

Now, this need not only, be an image, it could be sounds, maybe people who speak, you know if, I have a speak signature, maybe all of my speech signatures, will land up in the same part of the graph and somebody else's speech signature will land up somewhere else, it could be words, if you could somehow, turn every single word into a number, then maybe words with similar meanings or similar implications or close relationships, land up in the same part of the graph, this is actually a profound implication, okay.

So, our point is that, we are going to represent, both vectors as well as transformations as tensors, I will talk about that shortly. A transformation, is something, it is an operation between, or it is a map between a vector to a vector, so let us say you have $\mathbf{v}_1$, is say, $[1\ 2\ 3]^\top$ and it is a 3×1, vector and $\mathbf{v}_2$ is you know, it is a 5×1 vector. Now, somehow if you want to find out a map that goes from $\mathbf{v}_1$ to $\mathbf{v}_2$, what is the most natural way to do it?

So, you can say, something like, $\mathbf{v}_2$ is, so, remember this is 3×1, this is 5×1 and I could choose some matrix, $\mathbf{W}$ and say $\mathbf{v}_2 = \mathbf{W}\mathbf{v}_1$, if this is 3×1, this is 5×1, the natural way to go from one vector to another vector is, to stick a matrix up front or up later and what sort of matrix should this be? 3×5, okay, sorry I think I switched the numbers so in case this is 5×1 and in case this is 3×1, then this simply becomes 5×3, so $(5\times3)\times(3\times1)$, you see, 5×1.

The point is, the natural transformation, or in the natural mapping between one vector and the other vector, is to put a matrix up front, once again, I would like you to think about the image classification problem that we were looking at earlier, I had an image, it was something like this and this image was represented, remember, as a 3600×1, vector, my output, is simply a scalar, okay how do you go from this to this? You put a matrix up front, okay.

So, if you put a 3600×3600, matrix up front, $\mathbf{A}\times\mathbf{v}_1$ is $\mathbf{v}_2$. So the machine learning algorithm, has to somehow figure out this matrix which will take every possible image and then turn it into the right number, okay, obviously what we do is actually, a little bit more sophisticated, then this it is not as simple, but, none the less this gives you an idea of what we are going to do with machine learning, it is to try to find out what is this transformation, which is going to turn one vector into another vector.

As I said earlier, all these images, or all these dots, all these vectors, that we have, could be very high dimensional, even as smaller image has 60×60 essentially has to be represented as a 3600×1, vector which means it is 3600 dimensions. So, the implication is that you will need algorithms, that work very very well in high dimensions, we will see the need optimization algorithms, that work on very high dimensional data.

A very important implication, especially for engineering applications, is that this kind of representation, let us go between images and numbers, okay, every image can be thought of as a series of numbers, but, it also means that a series of numbers can be thought of as an image, okay.

So, we will, actually intelligently use this, later on in some applications, as we come to vision algorithms.

So, in this video what we saw were two important things, one is the idea of turning any kind of qualitative data, you have, into a vector of numbers, in particular, we saw some examples of how to do this with images, the second thing we saw, was simple notations for scalars, vectors and tensors the most important, mathematical idea that, I would like you to take is, that of a dimension of a vector, it simply means the number of components, to uniquely represent any image, you need a large number of components, therefore, an image can also be thought of as a very high dimensional vector, thank you.