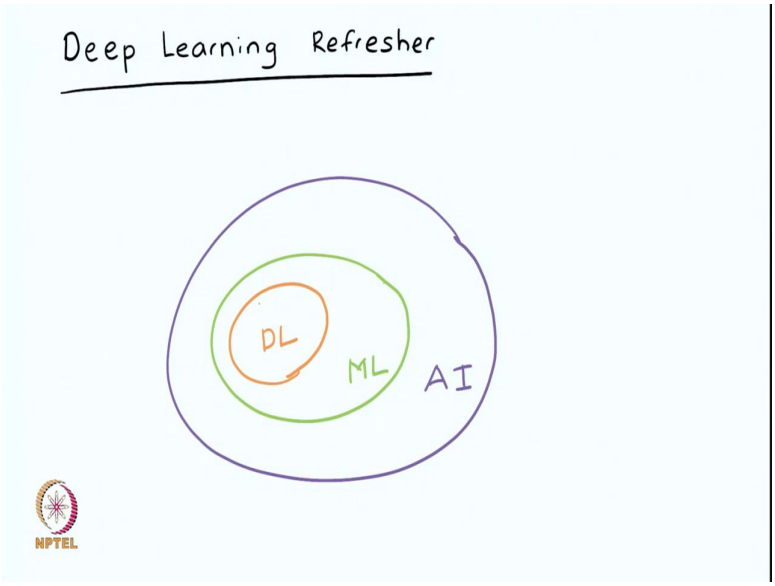
Practical Machine Learning with TensorFlow
Dr. Ashish Tendulkar Google
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

Lecture – 09
Deep Learning Refresher

[FL] Welcome to the next module of our course.

(Refer Slide Time: 00:21)



In this module we will study basic concepts of Deep Learning. AI is the bigger outer circle, within AI we have machine learning and within machine learning we have deep learning techniques.

So, it is very important to understand this perspective DL and ML are not two different things. DL or deep learning is a subset of machine learning, which in turn is a subset of AI techniques. So, within the realm of machine learning, we use deep learning to learn representations from the data automatically. We will see lots of such examples in our course going forward, where we are using deep learning to learn representations of the data automatically.

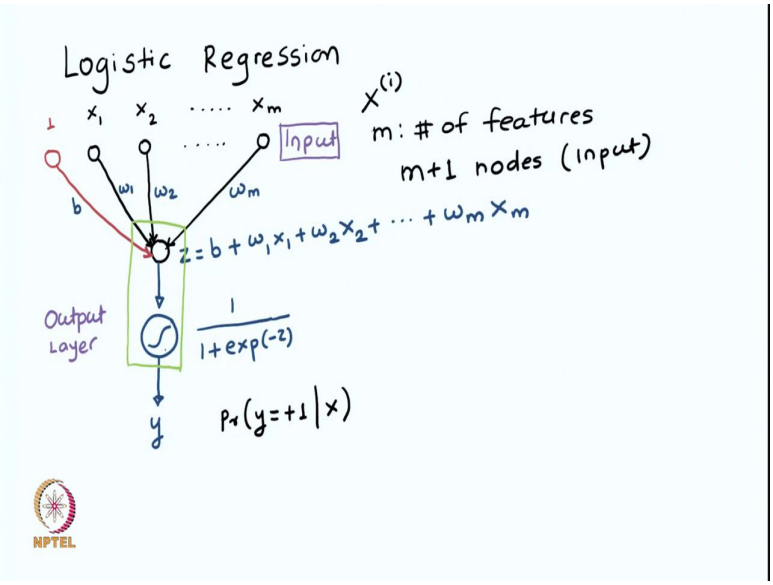
Normally, we use human experts to give us features, but you can see with deep learning, we can automatically learn the representation of the data without any human intervention. Deep learning techniques are not new deep learning techniques are known for quite some time.

However, there has been a tremendous progress in last 7 to 8 years in deep learning, which is mainly propelled by availability of three things. The first thing is data; the large internet companies have collected massive data sets, which are propelling the growth of or the progress in deep learning. Second is specialized hardware to train machine learning algorithms.

So, there are hardware like GPUs and TPUs, which are used for rapid training of neural network models. And, third is algorithmic improvements, which are helping us to train really deep models.

What has deep learning achieved? So, far deep learning has already achieved human level, performance in the computer vision task. So, deep learning is able to recognize objects from images as good as a human being. So, let us understand the concepts behind deep learning, and as you study them you will realize that these concepts are simple concepts built on the basics of machine learning that we studied in the previous module.

(Refer Slide Time: 03:15)



So, we studied a technique called logistic regression. In logistic regression, we had several features as an input. So, let us say there are m features as input; we will draw a node corresponding to each of the features. So, there are *m* features there are *m* nodes.

And, we use to tool we use to first perform and we have a special unit called bias unit which is 1. So, each of these inputs are fed to this particular unit, which does a linear combination of parameters and the weights. So, there are weights corresponding to each of the connection.

So, this particular unit performs linear combination. So, $b + w_1x_1 + w_2x_2$ plus and so on to w_mx_m. After performing the linear combination, we have another unit. So, let us call this linear combination at z and we take this z give z to this particular unit, which is a sigmoid activation unit.

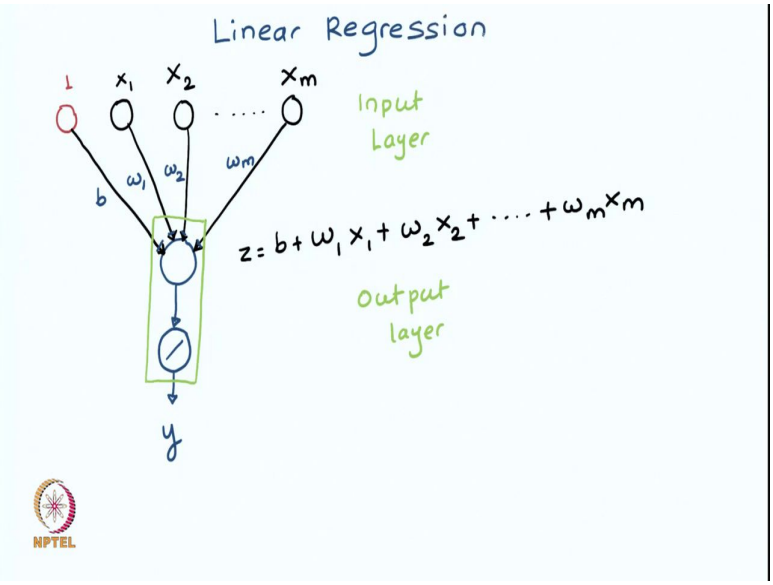
So, what it does is it applies; it passes z through the logistic function, and we get a number between 0 and 1. And, we interpret this y as a probability of the given example belonging to a particular to a positive class. We interpret this y as the probability of an example belonging to a positive class.

So, let us try to understand this more clearly so, what happens is what we pass here is a 1 single example features of a single example let us say this is *i*th example, which we denote by $x^{(i)}$. So, this is *i*th example we pass the features of the *i*th example through this. And, if we know all the weights what happens here we get a linear combination and followed by a sigmoid activation that gives us a number between 0 and 1, which is interpreted as a probability of the positive class.

We are representing logistic regression here in the style of neural network. So, it is important to note that a lot of people get confused about these nodes. So, there are nodes, there are exactly *m* plus 1 nodes. So, *m* is the number of features, and we have *m* + 1 nodes in the input.

So, this is an input layer and then the input layer is connected to a unit, which is called a hidden unit. A hidden unit has two components. The first component performs linear combination followed by the second component, non-linear activation. In case of logistic regression we use sigmoid as a non-linear activation.

(Refer Slide Time: 08:47)

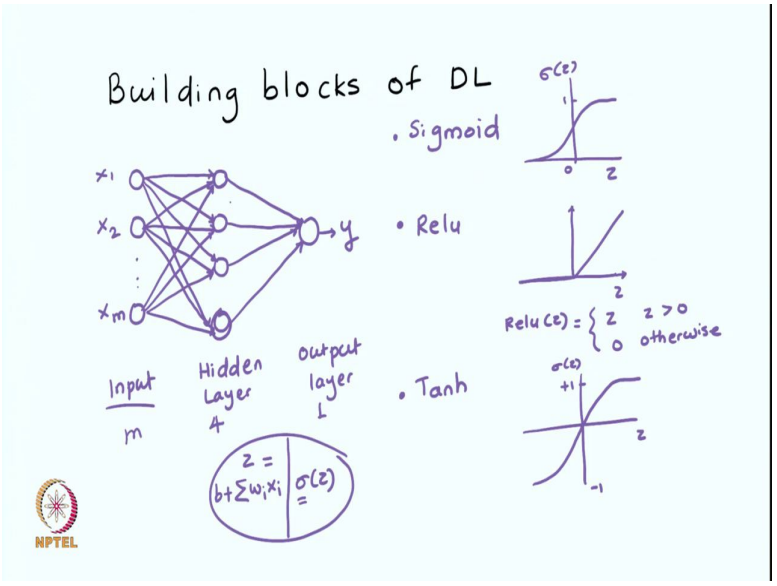


So, let us say again we have input data or each example represented with *m* features and we have a bias unit, then there is a linear combinator or a unit that does linear combination. So, we connect all these inputs in this particular unit. And, this unit calculates a linear combination of features and their weights.

So, there are weights on the connections. And, this particular unit computes a linear combination of the features and their coefficients or weights.

You pass this *z* through a linear activation function; in the case of a linear activation function. So, the value of *z* is essentially pass through this and we get y which is a real number and it is an output of a regression.

(Refer Slide Time: 11:47)



So, what we will do is we will take so, there is this input layer and we will stack multiple units that perform linear combination followed by activation, and we connect each of the input unit to these units in the second layer. And, then let us say we have one more unit where all these the output of this intermediate unit is connected. So, let us say this is these are all inputs x_1 , x_2 all the way up to x_m.

I am not explicitly showing the bias on each of the units there is additional there is an additional parameter here which is a bias parameter. So, we get so, what happens is that in each of the units a linear combination of the input and the weight what happens followed by a non-linear activation. And, there are few non-linear activations that we can use one is we have already seen it is a sigmoid activation.

In case of sigmoid activation a real number is squashed between 0 to 1 or from sigmoid, there is a another activation called ReLU, which made deep neural network training possible. So, what ReLU does is for anything less than 0 for any negative number we ReLU written 0 and the positive numbers are written as it is. So, the ReLU of z is equal to z if z is greater than 0 and it is 0 otherwise.

Third activation function that can be used is a Tanh activation function. So, Tanh activation function is has actual returns a number between plus -1 and +1. So, this nonlinear activations help us to learn complex models through neural networks.

If we use a linear activation over here instead of this nonlinear activations, what happens is that we simply get a linear combination and we are not able to get the complex model that we might need in certain cases. So, the first layer is called input layer the second layer is so, this particular layer is called as hidden layer, and the final layer is called output layer.

So, if you want to have, if you are solving the binary classification problem we generally use a sigmoid activation function. If we want a linear, if you are solving a regression problem then we use a linear activation function in the output. If you are trying to solve a multiclass classification problem, there will be multiple units in the output layer and we will be using either sigmoid or soft max as an activation function.

So, each unit has two parts, one is the linear combination and second is non-linear activation. So, each of the unit operates that way the activation function changes depending from problem to problem. And, it is really the discretion of designer to select the activation function.

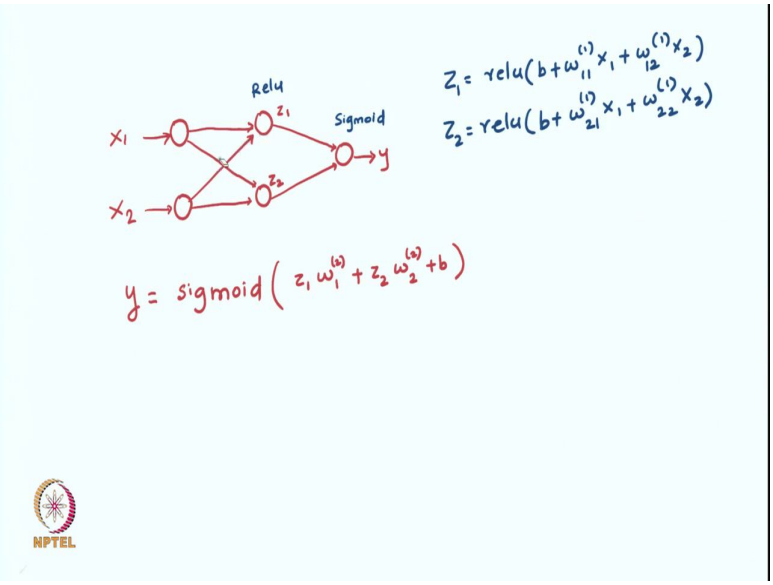
So, these are the basic building blocks of neural networks. So, there is an input layer, there can be one or multiple hidden layers, and there are output layers. So, in this particular case we have a neural network with one hidden layer and the hidden layer has got 4 units.

The input layer has got m inputs there are 4 units in the hidden layer and there is 1 unit in the output layer. We can also have multiple hidden layers; if we include multiple hidden layers naturally the number of parameters will increase. How many parameters are there in this network the number of parameters are equal to the number of edges or the number of connects that we see in the network plus the bias. So, it is very easy in the case of neural network to get complex models.

All that you have to do is we have to increase the number of layers or we can increase the number of units in the single layer. It is generally advisable to add more layers rather than increasing units in a single layer.

Notice that a neural network learns complex functions by breaking it down into simple functions. Let us see how it does it.

(Refer Slide Time: 17:49)



Let us take the toy neural network with two inputs; one in hidden layer with 2 units and one output layer. So, if we and let us say we are trying to solve a classification problem.

$$y = sigmoid(z_1 w_1 + z_2 w_2) + b$$

where z_1 and z_2 are outputs of the hidden units, w_1 and w_2 are weights corresponding to z_1 and z_2 and b is the bias term. z_1 and z_2 are calculated as:

$$z_1 = relu(x_1 w_{11} + x_2 w_{12}) + b$$

$$z_2 = relu(x_1 w_{2\,1} + x_2 w_{22}) + b$$

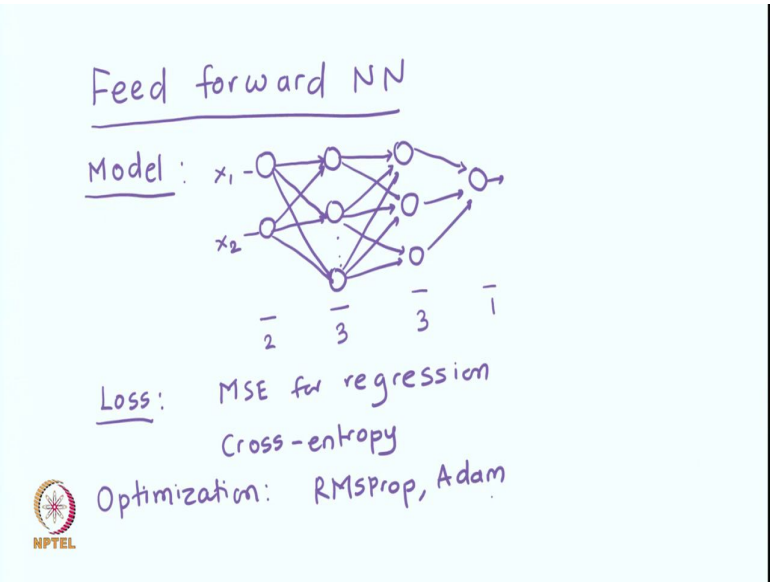
So, you can see that this particular function this is a complex function. And, it is broken down into smaller functions which are determined so, this complex function is broken down into simpler functions and that is the power of neural networks.

So, neural network essentially learns a very complex function by breaking it down into small into simpler functions, and then combining the output from the simpler functions to gradually learn more and more complex functions. It is important to understand how do we set up

number of hidden layers, how many units do we have in each of the hidden layer and what kind of activation functions we should be using. These are part of configurations of neural network architecture.

Number of units and number of hidden layers are specified as configurable parameters as part of neural network architecture. In this course we will be studying some of the popular architectures for solving problems in image recognition as well as text generation, along with the feed for a neural network that we have seen so far in the course.

(Refer Slide Time: 21:55)



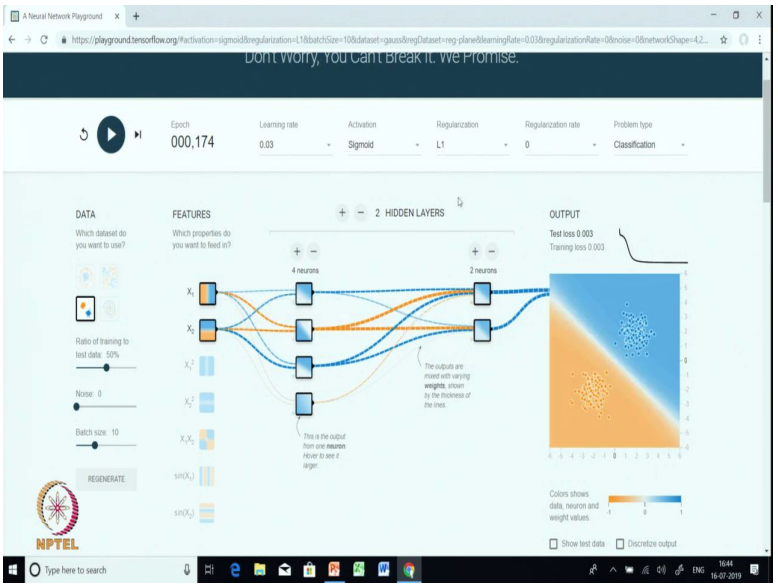
So, the first architecture that we will study is a feed forward neural network. So, feed forward neural network. So, what are the machine learning components for feed forward neural network? So, let us so, there are 4 components in any machine learning model 1 is the model. So, for a feed forward neural network we have an architecture, where each unit from the previous layer is connected to every other unit in the next layer.

And, we use multiple hidden units and we use multiple hidden layers. So, this is a classic feed for a neural network, it has got 2 hidden layers. In each of the hidden layer, there are there are currently 3 units that we are using and there is 1 output layer and in input layer there are 2 parameters that you are passing. So, these are parameters x_1 and x_2.

So, what is the loss function? If, you are solving the regression problem we use mean squared error as a loss for regression. And, if you are solving, if you are using feed for a neural network to solve a classification problem then we use cross entropy loss. And, what are the optimization algorithm that we use here?

We can use gradient descent, but more popular algorithm for neural networks are RMS prop or Adam. And, these algorithms extend some of the ideas that we learnt in stochastic gradient descent. So, they use the concept of momentum and make sure that the neural network is not stuck in a local minima. We will look at how to visualize the neural networks and training of neural networks.

(Refer Slide Time: 24:35)



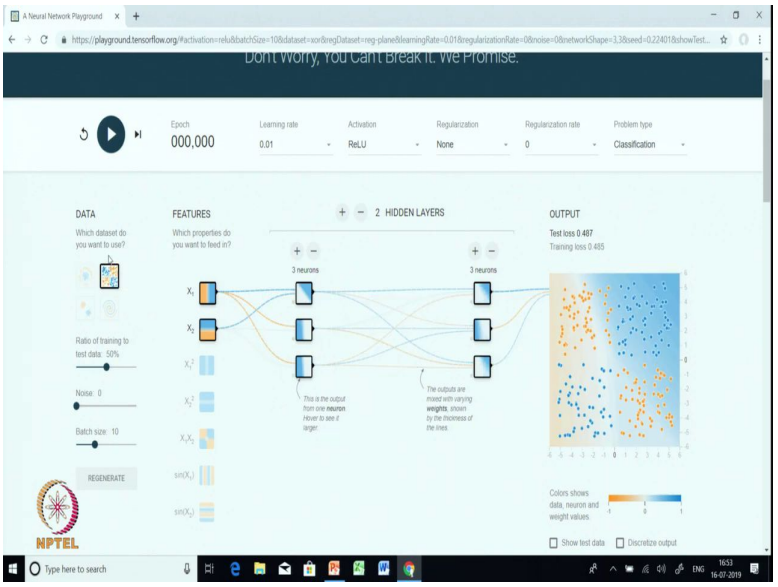
This is a deep playground that is familiar to us. So, we will take the linearly separable data and now what you can do is, we can define a neural network here. So, we have a control over here to add a hidden layer. So, let us say add a hidden layer and here we have a control to add more neurons in a hidden layer or more units in the hidden layer.

So, let us add 4 units. And, let us try to train them all, we use sigmoid activation function because we have a classification problem here and we do not use any regularization to begin with. So, let us use turning rate of 0.03 and we can see that within a few iterations we got almost a perfect classifier. And, you can see the weights on each of these on each of these

connections. If, we apply regularization, let us say we apply L2 regularization, you can see that some of these weights are very small.

So, let us apply a L1 regularization to drive some of the weights of parameters that are not useful to 0. We can see that within 89 iterations we reached quite a quite a small loss.

(Refer Slide Time: 27:01)



Let us try to use a neural network to separate 2 non-linearly separable classes. So, let us add a hidden layer with 2 units. And, let us see whether we are able to separate the 2 classes, we can see that we are not able to separate them perfectly.

So, let us add one more hidden layer with 2 units and retrain the model, looks like training loss is improving, but validation loss is not improving. So, we might be hitting over fitting kind of a situation. So, let us try to add a regularization with a small regularization rate and let us feed on the model ok. Model is still not learning into separate 2 classes.

Let us try to get a separation between 2 classes by adding 2 hidden layers, each with 3 units, let us try to train it and see whether we are able to get a reasonable decision boundary. Yes, you can see that now we are able to separate 2 classes with a complex decision boundary, the important thing to note here is that we achieved we are able to achieve the separation between 2 classes without adding any hand tool features, instead we added hidden layers and units in

each of the hidden layer. And, those hidden layers helped us learn the non-linear decision boundary as we can see over here.

So, you can see that neural network can generate features that can help us to separate non-linear classifiers. Earlier when we are using a simple linear model, we have to hand code the higher order or the interaction features between the between the original input features.

So, that is why deep learning is also used in a representation learning where we take the raw features and we learn some interesting combinations of them to learn complex patterns. So, we can try one more example on this particular data set - XOR dataset.

So, in order to solve this problem let us add 1 more hidden layer with 2 units and try to train the model. So, we have a batch size of 10. So, we are going to use mini batch mini batch approach to optimization. So, we are going to use batch size of 10. And, so, let us train the model with ReLU activation function and learning rate of 0.01.

And, you can see that it has learned some classifier which is separating 2 classes to some extent, but there are some points which are still not separable. So, let us try to increase the complexity of model further and see whether we can separate them.

That is quite interesting as you can see that the classes are separated precisely here. And, if you look at the individual neurons in the first layer neurons are only learning the linear separators. In the second layer neurons are learning regional separators. And, in the third layer they are learning combination of the separators that I learn in the second layer and they learnt how to separate the classes that are having points in the XOR fashion.

So, this is this is again reinforcing the fact that we did not hand code any of the features and just with the help of a neural network. we were able to classify points which have which have a very complex decision boundary.

So, hope this gives you some sense of how these hidden layers or the neural network architecture is helping us to transform the simple features into feature crosses, or complex features and is able to learn a complex decision boundary or complex model.

In other words what is happening is neural network is learning the complex function by composing simpler function that they learn from layer to layer. So, this is the beauty of neural network.