Week 3

## Classification

In this and the next few videos, I want

to start to talk about classification problems, where the variable y that

you want to predict is valued. We'll develop an algorithm

called logistic regression, which is one of the most popular and most

widely used learning algorithms today. Here are some examples of

classification problems. Earlier we talked about

email spam classification as an example of a classification problem. Another example would be

classifying online transactions. So if you have a website

that sells stuff and if you want to know if a particular

transaction is fraudulent or not, whether someone is using a stolen credit

card or has stolen the user's password. There's another classification problem. And earlier we also talked about

the example of classifying tumors as cancerous,

malignant or as benign tumors. In all of these problems the variable that

we're trying to predict is a variable y that we can think of as taking

on two values either zero or one, either spam or not spam, fraudulent or not

fraudulent, related malignant or benign. Another name for the class that we denote

with zero is the negative class, and another name for the class that we

denote with one is the positive class. So zero we denote as the benign tumor,

and one, positive class we denote

a malignant tumor. The assignment of the two classes,

spam not spam and so on. The assignment of the two classes to

positive and negative to zero and one is somewhat arbitrary and it doesn't really matter but often there

is this intuition that a negative class is conveying the absence of something

like the absence of a malignant tumor. Whereas one the positive class is

conveying the presence of something that we may be looking for, but

the definition of which is negative and which is positive is somewhat arbitrary

and it doesn't matter that much. For now we're going to start with

classification problems with just two classes zero and one. Later one we'll talk about multi

class problems as well where therefore y may take on four values zero,

one, two, and three. This is called a multiclass

classification problem. But for the next few videos, let's

start with the two class or the binary classification problem and we'll worry

about the multiclass setting later. So how do we develop

a classification algorithm? Here's an example of a training set for

a classification task for classifying a tumor as malignant or

benign. And notice that malignancy takes on

only two values, zero or no, one or yes. So one thing we could do

given this training set is to apply the algorithm

that we already know. Linear regression to this data set and just try to fit the straight

line to the data. So if you take this training set and

fill a straight line to it, maybe you get a hypothesis

that looks like that, right. So that's my hypothesis. H(x) equals theta transpose x. If you want to make predictions one thing

you could try doing is then threshold the classifier outputs at

0.5 that is at a vertical axis value 0.5 and if the hypothesis outputs a value that is greater than

equal to 0.5 you can take y = 1. If it's less than 0.5 you can take y=0. Let's see what happens if we do that. So 0.5 and so

that's where the threshold is and that's using linear regression this way. Everything to the right of this

point we will end up predicting as the positive cross. Because the output values is greater than

0.5 on the vertical axis and everything to the left of that point we will end

up predicting as a negative value. In this particular example, it looks like linear regression is

actually doing something reasonable. Even though this is a classification

toss we're interested in. But now let's try changing

the problem a bit. Let me extend out the horizontal

access a little bit and let's say we got one more training

example way out there on the right. Notice that that additional

training example, this one out here, it doesn't

actually change anything, right. Looking at the training set it's pretty

clear what a good hypothesis is. Is that well everything to

the right of somewhere around here, to the right of this we

should predict this positive. Everything to the left we should probably

predict as negative because from this training set, it looks like all the tumors

larger than a certain value around here are malignant, and all the tumors smaller

than that are not malignant, at least for this training set. But once we've added that extra example

over here, if you now run linear regression, you instead get

a straight line fit to the data. That might maybe look like this. And if you know threshold

hypothesis at 0.5, you end up with a threshold

that's around here, so that everything to the right of this

point you predict as positive and everything to the left of that

point you predict as negative. And this seems a pretty bad thing for

linear regression to have done, right, because you know these are our positive

examples, these are our negative examples. It's pretty clear we really should be

separating the two somewhere around there, but somehow by adding one example

way out here to the right, this example really isn't

giving us any new information. I mean, there should be no surprise

to the learning algorithm. That the example way out here

turns out to be malignant. But somehow having that example out

there caused linear regression to change its straight-line fit to the data

from this magenta line out here to this blue line over here, and

caused it to give us a worse hypothesis. So, applying linear regression

to a classification problem often isn't a great idea. In the first example, before I

added this extra training example, previously linear regression was just

getting lucky and it got us a hypothesis that worked well for that particular

example, but usually applying linear regression to a data set, you might

get lucky but often it isn't a good idea. So I wouldn't use linear regression for

classification problems. Here's one other funny thing about

what would happen if we were to use linear regression for

a classification problem. For classification we know

that y is either zero or one. But if you are using linear

regression where the hypothesis can output values that are much

larger than one or less than zero, even if all of your training examples

have labels y equals zero or one. And it seems kind of

strange that even though we know that the labels should be zero,

one it seems kind of strange if the algorithm can output values much

larger than one or much smaller than zero. So what we'll do in the next few videos

is develop an algorithm called logistic regression, which has

the property that the output, the predictions of logistic regression

are always between zero and one, and doesn't become bigger than one or

become less than zero. And by the way,

logistic regression is, and we will use it as a classification

algorithm, is some, maybe sometimes confusing that the term

regression appears in this name even though logistic regression is

actually a classification algorithm. But that's just a name it was given for

historical reasons. So don't be confused by that logistic

regression is actually a classification algorithm that we apply to settings

where the label y is discrete value, when it's either zero or one. So hopefully you now know why,

if you have a classification problem, using linear regression isn't a good idea. In the next video, we'll start working out the details

of the logistic regression algorithm.

Summary:

To attempt classification, one method is to use linear regression and map all predictions greater than 0.5 as a 1 and all less than 0.5 as a 0. However, this method doesn't work well because classification is not actually a linear function.

The classification problem is just like the regression problem, except that the values we now want to predict take on only a small number of discrete values. For now, we will focus on the **binary classification** **problem** in which y can take on only two values, 0 and 1. (Most of what we say here will also generalize to the multiple-class case.) For instance, if we are trying to build a spam classifier for email, then x^{(i)}*x*(*i*) may be some features of a piece of email, and y may be 1 if it is a piece of spam mail, and 0 otherwise. Hence, y∈{0,1}. 0 is also called the negative class, and 1 the positive class, and they are sometimes also denoted by the symbols “-” and “+.” Given x^{(i)}*x*(*i*), the corresponding y^{(i)}*y*(*i*) is also called the label for the training example.

## Hypothesis and Representation

Let's start talking about

logistic regression. In this video, I'd like to show

you the hypothesis representation. That is, what is the function we're

going to use to represent our hypothesis when we have a classification problem? Earlier, we said that we

would like our classifier to output values that are between 0 and 1. So we'd like to come up with a hypothesis

that satisfies this property, that is, predictions are maybe between 0 and 1. When we were using linear regression,

this was the form of a hypothesis, where h(x) is theta transpose x. For logistic regression,

I'm going to modify this a little bit and make the hypothesis g

of theta transpose x. Where I'm going to define

the function g as follows. G(z), z is a real number, is equal to

one over one plus e to the negative z. This is called the sigmoid function,

or the logistic function, and the term logistic function, that's what gives rise to

the name logistic regression. And by the way,

the terms sigmoid function and logistic function are basically

synonyms and mean the same thing. So the two terms are basically

interchangeable, and either term can be used to

refer to this function g. And if we take these two equations and

put them together, then here's just an alternative way of

writing out the form of my hypothesis. I'm saying that h(x) Is 1 over 1 plus

e to the negative theta transpose x. And all I've do is I've

taken this variable z, z here is a real number, and

plugged in theta transpose x. So I end up with theta transpose

x in place of z there. Lastly, let me show you what

the sigmoid function looks like. We're gonna plot it on this figure here. The sigmoid function, g(z), also called

the logistic function, it looks like this. It starts off near 0 and

then it rises until it crosses 0.5 and the origin, and

then it flattens out again like so. So that's what the sigmoid

function looks like. And you notice that the sigmoid function,

while it asymptotes at one and asymptotes at zero, as a z axis,

the horizontal axis is z. As z goes to minus infinity,

g(z) approaches zero. And as g(z) approaches infinity,

g(z) approaches one. And so because g(z) upwards

values are between zero and one, we also have that h(x)

must be between zero and one. Finally, given this hypothesis

representation, what we need to do, as before,

is fit the parameters theta to our data. So given a training set we

need to a pick a value for the parameters theta and this hypothesis

will then let us make predictions. We'll talk about a learning algorithm

later for fitting the parameters theta, but first let's talk a bit about

the interpretation of this model. Here's how I'm going to interpret

the output of my hypothesis, h(x). When my hypothesis outputs some number,

I am going to treat that number as the estimated probability that y is

equal to one on a new input, example x. Here's what I mean, here's an example. Let's say we're using the tumor

classification example, so we may have a feature vector x,

which is this x zero equals one as always. And then one feature is

the size of the tumor. Suppose I have a patient come in and

they have some tumor size and I feed their feature vector

x into my hypothesis. And suppose my hypothesis

outputs the number 0.7. I'm going to interpret my

hypothesis as follows. I'm gonna say that this

hypothesis is telling me that for a patient with features x,

the probability that y equals 1 is 0.7. In other words, I'm going to

tell my patient that the tumor, sadly, has a 70 percent chance, or

a 0.7 chance of being malignant. To write this out slightly more formally,

or to write this out in math, I'm going to interpret

my hypothesis output as. P of y=1 given x parameterized by theta. So for those of you that are familiar with

probability, this equation may make sense. If you're a little less

familiar with probability, then here's how I read this expression. This is the probability

that y is equal to one. Given x,

given that my patient has features x, so given my patient has a particular tumor

size represented by my features x. And this probability is

parameterized by theta. So I'm basically going to count

on my hypothesis to give me estimates of the probability

that y is equal to 1. Now, since this is a classification task, we know that y must be either 0 or

1, right? Those are the only two values

that y could possibly take on, either in the training set or for new

patients that may walk into my office, or into the doctor's office in the future. So given h(x), we can therefore

compute the probability that y = 0 as well, completely

because y must be either 0 or 1. We know that the probability of y = 0 plus

the probability of y = 1 must add up to 1. This first equation looks

a little bit more complicated. It's basically saying that

probability of y=0 for a particular patient with features x,

and given our parameters theta. Plus the probability of y=1 for

that same patient with features x and given theta parameters

theta must add up to one. If this equation looks

a little bit complicated, feel free to mentally imagine

it without that x and theta. And this is just saying that the product

of y equals zero plus the product of y equals one, must be equal to one. And we know this to be true because y

has to be either zero or one, and so the chance of y equals zero,

plus the chance that y is one. Those two must add up to one. And so if you just take this term and move it to the right hand side,

then you end up with this equation. That says probability that y equals zero

is 1 minus probability of y equals 1, and thus if our hypothesis feature

of x gives us that term. You can therefore quite simply

compute the probability or compute the estimated probability

that y is equal to 0 as well. So, you now know what the hypothesis

representation is for logistic regression and we're seeing

what the mathematical formula is, defining the hypothesis for

logistic regression. In the next video, I'd like to

try to give you better intuition about what the hypothesis

function looks like. And I wanna tell you about something

called the decision boundary. And we'll look at some visualizations

together to try to get a better sense of what this hypothesis function of

logistic regression really looks like.