all right in this short lecture i want to introduce the adaptations of machine learning we need to to do simple classification so in uh simple classification um or just classification we're taking a data item we thought of it as a vector x and we're assigning it a label um in the simplest case that label is just a yes no you know yes it's in the class or no it's not what we've seen so far allows us to assign an arbitrary real number and i want to review that to set the stage for talking about what we do to make the label be a yes or a no so in our basic setting we have a predictor that we called f and it takes in some data item x except we've learned that the data item x can be pre-processed to extract features right so we'll have some features of the data item features v and the f is going to be a predictor is going to output a label so i'm going to call that label f of x f of x and it's so far been a real number right let me call that the label that got assigned and what we've seen is that we're parameterizing f by a vector of weights of the same dimensionality as the feature vector and the predictor we've been working with thus assigns the label that's simply the dot product okay and we've learned that we can train a predictor like this by changing the weights using gradient descent to minimize the loss so how good is one of these predictors is measured by or really how bad is measured by the loss we want to minimize the loss and we can do that by going down the gradient so what is the loss and what we've seen is that we're going to have some training data i'll put that here so n items of training data that each one suggests that some input x i should be labeled with a label y i will define the loss per training data point or per data point and of course it's the loss for a particular particular predictor since our predictor is determined once we pick the weights really the weights and the features we'll imagine we pick the features and fix them before we get into defining loss so all we need to think of as varying is the weight vector right and so the the loss is going to be the prediction the distance the in something some measure of how how much penalty we want for the predictor not predicting y i so what does the predictor predict it predicts w dotted with fee and the input was x i that's the prediction and the prediction is wrong or not by measuring how far it is from y i we could take the absolute value of that that would be one kind of loss but we following our previous uh discussion we square it we're minimizing the squared loss so this is the squared loss and we can compute the gradient of this function we have done so and select w by gradient descent so that should be all very familiar everything i just said is a summary of our paradigm so what happens in this paradigm when all the y eyes are basically yes or no so we'll presume that we're going to use two real values one for yes and one for no turns out we're going to use plus one and minus one but that's not the only choice here's one and zero but basically there'll be a yes value and a no value for the y eyes if we just run this we're going to experience loss whenever our prediction isn't equal to plus 1 or minus one and in fact one of those as well depending if y i is plus one we'll experience loss if we predicted minus one so one thing to consider is possibility that this would select manage to select a predictor that always predicts plus one that will be possible if one of the features is a constant and we can set that feature's weight so it multiplies to plus one and all the other weights zero we'll be able to always predict plus one and we'll get zero loss on all the plus ones but we'll get lots you know measurable loss two squared uh um on all the minus ones um and likewise we could have a predictor that predicts always minus one and we get all the plus ones wrong chances are those are the gonna do the best um compared to anything else which is any anything else that isn't constant plus one or minus one it's going to have some direction where it's growing well it could be a constant in between there then it's going to get both of them wrong that might be the best but any anything that's not making a constant prediction is going to have a direction in which it's growing and as features travel in that direction the prediction will grow this dot product as the features travel in the direction that makes the dot product grow larger and larger feature values are going to give larger and larger and larger answers further and further away from plus one and minus one and experience lots of loss in that direction so just generally this paradigm doesn't seem very well set up to capture this kind of data and in fact it's not nobody would use this to learn this to learn a classification predictor so what we're going to do instead for classification is we're going to change both the predictor so it's not always producing a real number so i'm sorry it is always producing a real number so it's not producing arbitrary real numbers it's always going to produce a plus 1 or minus 1. change the predictor so it's at least making a stab that could be correct um to produce only plus ones and minus ones and then we'll change the notion of loss because we're not really going to want to only notice whether we got them right or wrong we want something we can smoothly adjust so we're going to need something to some measure of how sure we are that it's plus one or minus one if we only just have the predictor saying plus one or minus one it'll be hard to define loss in a way that has a meaningful gradient because as we change the weights maybe none of the predictions change small changes in weights right epsilon changes in the weights probably won't change any of the predictions and the gradient in the loss will be zero if we only look at what's right and wrong so we're really going to want the loss to have available to it some something from inside the predictor that indicates how confident it is in its predictor and you'll see that it's very relatively easy for us to get all of this so that's the essentially the entire topic today is how do we get how do we adjust the predictor and how do we adjust the loss so it has a useful gradient and then we just apply gradient descent so so this is a basic predictor for classification would be to say well okay go ahead and compute this number w dot phi of x is it's called the score of x and we're going to think of that as our confidence in a plus one prediction and note that it can be negative if it's negative we're opposite of confident in a plus one prediction we start to become confident in a minus one prediction then f of v of x so f of a that item can be given by the sign of this confidence okay so the sign of the score is our prediction and that is going to be an element of minus one plus one now sometimes the right answer is minus one and sometimes the right answer is plus one so sometimes we want the score to be large and sometimes we want the score to be small in order to unify those two concepts and talk about different training data points in a sort of unified framework we introduce a concept we call the margin which is how correct we are that's intuitively how correct we are which is how big is the score in the right direction so the margin for x i y i the margin well it's going to be the score if y is plus 1 and it's going to be the negative of the score if y is minus 1 so it's y i times the score of x i okay so what we want is a positive margin we have a positive margin we'll predict it correctly and the more positive it is intuitively the weights have adjusted so they're not even close to getting this one wrong because the margin is going to change continuously as the weights change so if it's far away from zero it's far it's far away from changing from right to raw or wrong to right we want it to be far away from zero but on the positive side so this gives us a sort of training goal other than getting it right we want to get the margin driven up at least a bit away from zero that said right now we don't have a loss function that's designed for this predictor we can use this loss but this loss is going to assign um all kinds of penalties even when we're getting the prediction right right this boss doesn't even have the sign of the score built into it all right so that would be our our next topic which is how to get the loss set the loss function so that training this will get try to get the answers right and not only get the answers right but ideally get the answers right confidently so drive the margin up away from zero so first let's consider a natural loss function for classification and that would be one that's called zero one loss we get a loss of one if we get it wrong so the loss on a particular x i that should be labeled y i with given weights will be equal to zero if the score the margin rather is greater than zero so if our correctness is bigger than zero and one otherwise now the thing to notice about this is as we change the weights the margin changes smoothly and if it isn't close to zero it isn't going to change from one of these cases to the other so the gradient of this loss for a particular training point really for all the training points if we make epsilon changes to the weights it's going to be zero none of the training points are going to change their loss because none of the margins are going to cross 0 and a small epsilon change to the weights if one of them does cross 0 the gradient will be ill-defined because an epsilon change in the weights will cause a very non-epsilon change in the loss so the upshot of this is the gradient nearly always serial zero and the rest of the time it's not defined so not useful for gradient descent so let's look at what that plot what that last function looks like all right so here the important elements of this plot and this should be a slide credit to the stanford course that i've credited a number of other times [Music] this lot this plot of the loss functions the the x-axis is this complicated new concept the margin it's not complicated once you get page in the concepts though concepts are to learn a real number that we can get from any input so we have an input like an image and the real number is going to be how sure am i that that's a cat okay that's this part but we're going to change the sign of this according to whether it's supposed to be a cat if i'm really sure it's a cat and it's supposed to be a cat this will be positive and if i'm really sure it's a cat but it's not labeled the cat said i'm really wrong so this is how right we are not just how sure we are it's a cat all right so this is how right we are and if we're positively correct we give no loss this red and if we're negatively correct that means we're wrong we get loss of 1. this is sort of like grading a multiple choice quiz right you get a penalty if you get the wrong answer and you don't get a penalty if you get the right answer that's the red so the red is 0 1 loss okay and it has no gradient as you can see if we take the gradient of this um loss as a function of margin uh it it's zero everywhere and that means that the gradient with respect to any components of the margin in other words the waves is going to be zero so we need a gradient that drives us in this positive direction right we want to adjust the weights to get the margin up above zero and maybe even away from zero and the classic thing to do um classic in one sense at least classic in the machine learning literature is this green which is called hinge loss which is used in these older not terribly old but quite a bit older than about you know maybe your age anyways decades-old idea of support vector machines this you don't need to know so i just put the acronym there but hinge loss does is what is minimized by the support vector machine machine learning technique that met a lot of success around the turn of the century and let's see what the hinge loss is doing it's saying that we really would like to get the margin above zero all the way up to one now this is a particular somewhat arbitrary choice some constant above zero that we want to get it to get it uh in this case up to one the margin once up to one we don't really care what you do with it so the gradient is zero here but but any data point any training point that is scoring um so that its margin is close to zero but it's still above zero it's being gotten right but we're not happy we're still penalizing until you get it up to one this is essentially to try to make a very clear boundary around the decision on the other hand if your margin is negative and you're getting it wrong you're definitely no matter how negative you get the same sort of signal to keep going this way keep traveling to increase the margin adjust the weights to increase the margin that's the gradient i'm talking about right the gradient with respect to the weights says we're going to be reducing the loss if we head so the margin goes this way to the right that's what hinge loss is doing now we're not going to discuss it in this lecture but another option is this logistic loss this yellow and it's more well motivated probabilistically and we will discuss the probabilistic motivations for this logistic loss later which have to do with thinking of the score in terms of the probability that the data item is in the in in the class and you can just for reference since it's on the slide you can see that if we were to minimize that it would give us the same sort of signal as hinge loss but it would never quite give up like no matter how big the margin we get some reward for increasing the margin even further okay so what remains is to say what is this function hinge loss we can see it in the plot so that's pretty good i'm going to define it for you and then we'll be pretty much done with introducing basic classification so hinge loss so what is the hinge loss well if the margin is good then it's just zero right margin greater than one if the margin is not that good then it's got a slope of minus one in terms of the changing margin and and when the margin is zero if the loss is one just reading it from that plot so that one minus margin is this this graph right where we have intercepts of one and if this is the margin and this is the loss then this this part right here is the part that's plotted there and when the margin is bigger than one we get zero so and that's the hinge loss now um this gives us the desired gradient so we now have a predictor that predicts plus one or minus one we have a notion of loss that we can compute per training data point and we can sum it over all the training data points or we can do just do one data point at a time now we can deploy our mechanisms for adjusting the weights which were gradient descent or stochastic gradient descent just the same as we did for linear regression but we're now using a different predictor and a different loss function and we will learn a class a weight vector that minimizes this loss function and gets a lot of the data points correct so this pretty much completely defines the basic technique that you can use with regard to ignoring such choices as how do we degrade the learning rate how do we choose the features that we already had those issues in linear regression as well we've now adapted that setting to straightforward yes no prediction