00:00:01.820 so uh they shifted the entry point on us
00:00:05.339 again today so I expect people will be
00:00:07.080 walking in late
00:00:09.420 um
00:00:10.559 how do you get in that way actually
00:00:21.119 okay
00:00:23.460 so if you go on the left side of that
00:00:26.039 left-hand building then you get down
00:00:28.140 that way
00:00:31.080 I think there's going to be more traffic
00:00:32.520 that way now that we've learned that
00:00:35.640 okay
00:00:40.860 so uh let's do our best because we know
00:00:43.620 people are going to be filtering in with
00:00:45.300 that uh staircase problem
00:00:50.120 convexity optimization yeah this is the
00:00:54.420 basis of machine learning so
00:00:57.719 uh I wanna I wanna get people to quiet
00:01:00.360 down though because it's hard enough to
00:01:03.120 deal with people walking in late let's
00:01:05.760 uh minimize the noise inside the room
00:01:07.860 please
00:01:10.619 okay
00:01:12.659 before we get to convexity and
00:01:14.939 optimization I need to finish up what I
00:01:17.220 didn't do the last time with similarity
00:01:20.880 okay
00:01:22.200 we were talking about the idea of one
00:01:24.780 hot encoding
00:01:27.659 please
00:01:29.520 thanks
00:01:30.780 so uh we're talking about the idea of
00:01:33.119 one hot encoding if there's categorical
00:01:35.759 data
00:01:37.259 and the categorical data is not
00:01:40.979 inherently ordinal
00:01:43.200 right
00:01:44.299 so so what's inherently ordinal
00:01:48.299 anybody ever filled out a survey which
00:01:50.399 is like do you strongly agree or agree
00:01:54.720 or neutral disagree strongly disagree
00:01:58.439 that's categorical
00:02:00.659 but it's ordinal
00:02:02.820 okay because strongly agree is greater
00:02:06.420 than agree
00:02:09.000 so that's called a likert scale a one to
00:02:12.480 five
00:02:13.500 situation and for a likert scale survey
00:02:17.700 response strongly agree to strongly
00:02:19.680 disagree
00:02:21.120 you don't need to one hot and code that
00:02:23.220 you can just replace that likert scale
00:02:26.400 thing with a one to five because there's
00:02:28.860 an inherent order
00:02:30.720 to the categories okay
00:02:33.360 so ordinal data you can ignore this but
00:02:37.080 if your categories are something like
00:02:38.940 red green blue yellow
00:02:41.700 there's no inherent way in which red is
00:02:44.340 greater than blue
00:02:46.019 okay they're just different categories
00:02:48.239 so anything which is inherently
00:02:50.720 non-ordinal needs to be encoded in a one
00:02:54.720 hot encoding because one hot encodings

00:02:58.640 induce orthogonality
00:03:01.800 red and blue don't have any similarity
00:03:04.200 to each other
00:03:05.879 without inducing ordinality
00:03:09.300 okay
00:03:10.860 that's the key
00:03:13.200 okay
00:03:15.540 so if in doubt generally speaking one
00:03:19.620 hot encoding is the right idea
00:03:22.440 it's not going to hurt anything if you
00:03:24.360 take an ordinal data like a likert scale
00:03:26.700 and you one-hot it the only thing that
00:03:29.040 it really hurts is that it makes your
00:03:30.959 data wider right you get more columns in
00:03:35.159 your data set than you really need
00:03:38.879 if in doubt one hot
00:03:42.120 okay
00:03:44.400 so we've talked about dot product as a
00:03:47.640 measure of similarity
00:03:49.560 but the dot product is intimately
00:03:52.319 related to another way to measure
00:03:55.379 similarity that's the cosine similarity
00:03:58.200 cosine similarity has got a DOT product
00:04:01.220 there in the uh I want to use that right
00:04:04.920 it's got a DOT product on the numerator
00:04:08.040 and then it's normalized by the
00:04:10.379 magnitude of the vectors
00:04:12.299 that's actually the cosine of the angle
00:04:14.580 between the two vectors
00:04:16.680 okay
00:04:17.760 so the cosine similarity is different
00:04:20.699 than the dot product
00:04:22.440 it only cares about Direction because
00:04:25.139 it's normalized by the magnitude of the
00:04:28.259 vectors
00:04:29.400 so only a vector's direction matters for
00:04:33.180 a cosine similarity
00:04:34.979 so
00:04:36.979 these two vectors in cosine similarity
00:04:40.580 are exactly identical I'm going to try
00:04:43.680 to figure out how to do this to these
00:04:45.720 two vectors
00:04:47.759 okay the change in length doesn't change
00:04:52.320 the cosine similarity
00:04:55.440 got it
00:04:56.940 only distance sorry only Direction
00:04:59.580 matters not distance
00:05:03.000 so which one's right for you
00:05:05.280 well it depends on what your task is
00:05:09.419 right depends on what the representation
00:05:11.460 of the data is and what makes sense
00:05:13.259 there are definitely machine learning
00:05:15.060 problems we're looking at the similarity
00:05:17.340 between two data points it makes way
00:05:19.680 more sense
00:05:21.060 to use direction to use a DOT product
00:05:24.960 okay because changes in the vector's
00:05:27.600 magnitude matter in that problem
00:05:29.759 so I don't know like um
00:05:33.900 it's a good example if we're talking
00:05:36.720 about patient data again
00:05:40.080 if the blood pressure
00:05:43.680 and the weight of the patient
00:05:47.580 double

00:05:49.020 that probably changes their health
00:05:50.699 outcomes right
00:05:52.919 it's not just the direction the relative
00:05:55.699 uh how much bigger blood pressure is
00:05:59.759 than somebody's weight okay it's not
00:06:02.880 just that it's probably the Rel the
00:06:04.979 actual magnitudes that matter here
00:06:08.180 whereas uh if we're talking about
00:06:12.660 uh what's a place where the relationship
00:06:15.240 between data features matters more than
00:06:18.479 their magnitudes
00:06:21.180 so gene expression
00:06:24.419 something I work with from time to time
00:06:27.060 so sometimes the thing that's diagnostic
00:06:29.639 is not the absolute level of Gene RNA
00:06:34.319 that's floating around in the system
00:06:35.580 it's the relative levels
00:06:38.699 so when Gene a is expressing twice as
00:06:42.479 much as Gene B
00:06:44.940 that indicates one thing is going on but
00:06:47.639 the thing is is that the measurements of
00:06:49.380 gene expression are kind of wonky if we
00:06:52.319 made the same measurement on the same
00:06:53.880 cells next week
00:06:55.440 the raw numbers might change but their
00:06:59.280 relative relationships wouldn't
00:07:01.740 okay
00:07:03.419 so that's a case where cosine where
00:07:06.600 Direction might be a better choice to
00:07:09.060 measure similarity
00:07:13.620 okay it's kind of you gotta know your
00:07:16.080 problem and what's going on
00:07:18.780 okay
00:07:20.220 so looking at the math remembering what
00:07:23.099 a cosine is pop quiz
00:07:27.180 if there's a cosine similarity of zero
00:07:30.120 are the vectors the most the least
00:07:35.220 similar
00:07:36.300 I'm going to tell you uncertainty is not
00:07:37.800 in there that's a decoy
00:07:40.139 what do we think
00:07:42.660 say it
00:07:46.380 sorry did somebody say it
00:07:48.599 zero is
00:07:50.460 cosine of zero does it happen when
00:07:52.740 vectors are parallel or when they're
00:07:54.900 orthogonal
00:07:58.319 when they're orthogonal
00:08:03.120 that's the least similar
00:08:05.759 actually sorry I apologize not
00:08:07.800 orthogonal isn't one is one they're 180
00:08:10.740 away from each other
00:08:12.300 even I don't know
00:08:16.560 is it orthogonal okay
00:08:19.740 yeah so anyway that wouldn't make sense
00:08:21.300 the least similar is when they're zero
00:08:23.280 the most similar because it's a cosine
00:08:26.580 when they're parallel it's going to be
00:08:28.440 one
00:08:30.960 okay so yeah
00:08:40.559 yeah it would be negative correct yeah
00:08:43.140 so that would be negative one
00:08:47.040 okay
00:08:48.540 so uh feature representation we can

00:08:52.560 revisit our one hot encoded animals
00:08:55.680 and we can in the past we were doing Dot
00:08:58.440 product what if we do cosine similarity
00:09:02.220 in cosine similarity magnitude doesn't
00:09:05.640 play any role it's just you know the
00:09:09.000 direction
00:09:10.080 so in cosine similarity if we have
00:09:13.080 Sparrow bat difference being roughly the
00:09:16.200 same as the Chipmunk bat difference
00:09:19.140 okay
00:09:21.000 but once again Sparrow and chipmunk are
00:09:23.399 the most different from each other just
00:09:25.500 like before
00:09:31.080 okay uh this is where I regret not
00:09:34.740 getting this far last time because I
00:09:37.080 know there was a lecture quiz question
00:09:38.760 on this did people have problems with
00:09:41.820 this idea on the lecture quiz the last
00:09:43.920 problem
00:09:47.160 so feature scaling is going to affect
00:09:51.300 both dot product and cosine similarity
00:09:54.420 measurements
00:09:56.399 if we take one vector element just one
00:10:00.839 of the features
00:10:02.279 and we multiply it by a large number
00:10:05.399 what are we effectively doing
00:10:07.800 okay so like just imagine that our
00:10:09.839 Vector space for the animals is in 2D
00:10:12.000 it's not it's in higher dimensionality
00:10:14.060 but if you know this is our chipmunk
00:10:17.580 vector and this is our
00:10:20.899 bat vector
00:10:24.839 like this
00:10:27.920 and they're similar but not that similar
00:10:30.959 right
00:10:32.940 well if I make one of the vectors A
00:10:36.120 couple of orders of magnitude bigger
00:10:39.060 I stretch out the part of the vector
00:10:42.300 that is related to the weights by going
00:10:46.140 from what had been in kilograms to gram
00:10:49.440 representation so a thousand times
00:10:51.420 bigger
00:10:52.620 I'm stretching out one of these
00:10:54.720 Dimensions right
00:10:57.180 So what had been
00:11:00.360 say you know
00:11:02.100 if we if we just assume that like this
00:11:05.279 direction
00:11:06.360 in the vector space is weight
00:11:11.519 what happens when we stretch out the
00:11:14.399 magnitude by the that's Direction by
00:11:17.640 three orders of magnitude
00:11:19.380 well we get the same
00:11:21.660 you know X component in chipmunk and bat
00:11:25.860 but we're stretching the crap out of the
00:11:28.079 Y component so this Vector now becomes
00:11:32.940 like that
00:11:34.260 and the bat Vector now becomes
00:11:37.740 like that
00:11:39.240 so high I can't even draw my arrows
00:11:42.060 right
00:11:43.440 so
00:11:44.839 the angle between them
00:11:47.820 is now not so different

00:11:50.160 right
00:11:52.740 and and Dot products even more so
00:11:54.839 because dot products are firmly just
00:11:57.600 directly connected to the magnitude of
00:11:59.399 the vectors
00:12:00.839 so if you stretch One Direction you
00:12:04.380 remove the influence of all the other
00:12:06.180 directions in the vector
00:12:08.339 right you make them less influential
00:12:11.640 so this is a very typical machine
00:12:13.740 learning problem
00:12:15.180 you've got a bunch of variables one of
00:12:17.820 those variables on you know a housing
00:12:20.880 data set is number of bedrooms
00:12:23.760 it's going to be a number less than 10
00:12:25.500 right
00:12:27.000 one of the variables is the housing
00:12:29.160 price in dollars
00:12:30.720 San Diego County that's going to be in
00:12:32.640 millions
00:12:34.620 guess which one of those two variables
00:12:36.620 overrides the other
00:12:40.260 okay big variables have undue influence
00:12:44.519 over small variables
00:12:47.519 so we very very typically try to
00:12:50.579 normalize each variable
00:12:52.980 it depends on the algorithm some
00:12:54.839 algorithms don't work like this
00:12:57.300 okay to give you an example decision
00:12:59.459 trees which we will talk about later in
00:13:01.740 the quarter they don't care when they're
00:13:04.139 dealing with housing price they they
00:13:07.079 don't even look at number of bedrooms
00:13:09.839 okay so there's no comparison between
00:13:12.420 them you don't have to normalize for a
00:13:15.480 decision tree
00:13:16.860 but
00:13:18.240 algorithms that work through linear
00:13:20.459 algebra like support Vector machines
00:13:23.160 linear regressions this stuff that we'll
00:13:25.740 talk about
00:13:27.120 when you have one feature which is a a
00:13:31.019 million bigger than another feature you
00:13:34.680 always normalize each feature so what
00:13:37.019 we're going to do is we're going to
00:13:39.060 normalize chipmunk bat and Sparrow's
00:13:41.820 weight
00:13:42.600 we're going to replace this with I don't
00:13:46.019 know like a number between 0 and 1 where
00:13:49.260 1 is the biggest weight that's in the
00:13:51.060 data set and zero is the smallest
00:13:54.079 or maybe we z-score it
00:13:57.360 right so we z-score the entire column
00:14:00.120 and replace the column with the z-scored
00:14:03.839 version of that data
00:14:05.820 for those of you that don't remember
00:14:06.839 what a z-score is it's replacing the
00:14:10.079 number with how many standard deviations
00:14:12.779 away from the mean of the data are you
00:14:16.500 okay
00:14:20.040 then everybody every feature is on the
00:14:23.100 same rough size scale and each feature
00:14:25.560 can be equally influential in the answer
00:14:30.240 questions

00:14:36.120 okay
00:14:37.740 so relative scaling is important we
00:14:41.579 usually normalize to get rid of the
00:14:44.459 influence of that especially if it's a
00:14:46.500 linear algebra kind of algorithm
00:14:50.339 so we've got two ways to measure
00:14:52.199 similarity
00:14:53.820 dot product cares inherently about
00:14:56.040 Vector length as well as Direction
00:14:58.040 cosine similarity cares only about that
00:15:00.540 direction
00:15:01.800 they've got different Min and Max values
00:15:05.880 but their mins are always zero
00:15:09.500 both methods are changed by scaling
00:15:13.980 dot products more sensitive than cosine
00:15:16.260 but cosine can be overwhelmed by scaling
00:15:19.139 as well
00:15:20.279 and you probably want to use one hot in
00:15:23.699 any kind of vector situation that's
00:15:25.639 non-ordinal categorical
00:15:31.199 that's all I got to say about
00:15:32.279 representation and similarity anybody
00:15:34.440 got any questions on that stuff
00:15:38.459 okay
00:15:40.139 um
00:15:41.880 actually
00:15:43.639 briefly let uh
00:15:47.220 trying to think about my timing in here
00:15:51.480 let me just say this
00:15:53.579 okay I'm not going to actually do the
00:15:55.920 class exercise that I had prepared but
00:16:00.300 I want you to note that I had prepared
00:16:02.940 oh f
00:16:12.360 I had prepared an exercise for yesterday
00:16:15.120 but we were just way too slow
00:16:17.899 and if it will allow me to actually
00:16:21.420 connect to the internet no it won't this
00:16:24.240 is lovely
00:16:26.760 um
00:16:31.440 so there's a notebooks repo
00:16:34.139 here cogs18
00:16:37.880 you can see that there's some in-class
00:16:41.040 exercises that we might do if I'm not so
00:16:44.220 damn slow on the lecturing
00:16:46.620 you should definitely
00:16:48.600 if you haven't already
00:16:50.339 play with the vector similarity notebook
00:16:53.399 just because it gives you an opportunity
00:16:54.899 to both do some numpy
00:16:59.279 and form vectors
00:17:01.199 and actually play with scaling yourself
00:17:03.300 so there's a little you know fill in the
00:17:05.579 blanks kind of
00:17:07.260 thing and you can just explore
00:17:10.140 and learn to use numpy if you haven't
00:17:12.299 already learned to use numpy
00:17:14.819 all right
00:17:20.220 but in the meantime let's Boogie so
00:17:23.520 machine learning is really just picking
00:17:25.980 an appropriate model and optimizing some
00:17:29.460 function
00:17:30.540 so when we talk about optimization you
00:17:33.900 will see people say things like loss
00:17:36.059 function cost function objective

00:17:37.860 function Fitness function we all mean
00:17:40.559 the same thing it's some function we're
00:17:43.260 trying to optimize
00:17:45.120 okay different people just use different
00:17:47.160 jargon
00:17:49.380 um
00:17:50.039 so what does that look like
00:17:52.260 what is optimization the idea is that we
00:17:55.919 have some loss we tend to use the word
00:17:58.559 loss in machine learning that we're
00:18:01.260 trying to minimize where you can think
00:18:03.179 of loss as just
00:18:04.919 how bad are our mistakes right
00:18:08.400 we're trying to predict the right
00:18:10.679 answers to the label training data
00:18:15.240 so we make some predictions those
00:18:17.580 predictions
00:18:19.440 right if you followed from our
00:18:23.720 pre-lecture those predictions are
00:18:27.120 just running our model forward that's
00:18:29.160 this term
00:18:30.900 the truth the actual results we should
00:18:33.840 have gotten are why
00:18:36.240 okay so the difference between them is
00:18:39.120 just going to be a vector
00:18:40.980 that has
00:18:42.720 you know how bad our predictions were
00:18:45.900 right for every data point in the vector
00:18:48.900 how bad were our predictions if this
00:18:51.360 Vector is largely close to zeros then
00:18:54.240 we're doing great and if it's far away
00:18:56.520 from that we're doing poorly
00:18:58.679 so uh somebody had on campus wire called
00:19:01.919 me out on my bad notation in the
00:19:03.780 pre-video thank you very much for doing
00:19:05.640 that
00:19:06.480 right the thing is is that it's not just
00:19:09.299 like this raw difference because of
00:19:11.400 course that's a vector
00:19:13.320 that Vector goes through some kind of
00:19:15.419 function
00:19:16.620 and it produces
00:19:18.539 a loss surface so this is a scalar
00:19:21.299 number that comes out
00:19:24.120 so that scalar number might be something
00:19:26.160 like the average across these Vector
00:19:28.200 differences or it might be the sum of
00:19:31.559 those differences
00:19:33.539 so for things like ordinarily squares
00:19:35.460 regression it's a sum
00:19:37.500 it's the sum of squared errors that's
00:19:39.539 what OLS regression does
00:19:43.440 so the vertical here represents how bad
00:19:48.360 we're doing the higher the surface the
00:19:51.299 worse it is down here where the surface
00:19:54.179 touches zero that's perfect
00:19:56.520 okay and the surface is defined in terms
00:19:59.940 of the weight Vector so when we change
00:20:02.220 the weights of the parameters you know
00:20:04.620 if we're over here for our weights
00:20:06.419 really big w0 really small W1 we're
00:20:10.140 doing terribly
00:20:11.700 and when our weight values are over here
00:20:14.760 kind of a middle zone for both w0 and W1

00:20:18.240 we're doing great
00:20:20.039 okay and then we're doing terribly again
00:20:22.440 over here
00:20:24.059 okay got that so the vertical is how bad
00:20:27.179 we're doing that's the loss
00:20:30.299 for those of you that wait how many
00:20:32.340 people here can read a topo map when you
00:20:34.260 go hiking
00:20:36.720 not as many as I could hope for okay so
00:20:39.480 this is a topo map right these curves
00:20:43.440 here
00:20:44.700 they're lines of equal height
00:20:47.520 okay so if you imagine this is the
00:20:50.039 Mountaintop or the the bottom of the
00:20:52.200 valley either way all you know from this
00:20:54.900 view which is literally like a top down
00:20:57.419 of this right it's a top down of this
00:20:59.940 where this circle here around the top of
00:21:03.360 the surface is this circle
00:21:06.059 and this circle here in the middle
00:21:08.820 is that Circle okay
00:21:12.299 so uh when you're moving across lines
00:21:15.660 the closer together the lines the
00:21:18.299 steeper the slope
00:21:20.100 and the further apart the lines the
00:21:22.679 shallower the slope okay that's how you
00:21:24.780 read a topple map when you're going
00:21:26.520 hiking and that's what this is this is a
00:21:28.500 toppo 2d representation of this
00:21:32.340 okay
00:21:33.840 so
00:21:34.980 the way we actually do things in
00:21:37.799 optimization is that we know we want
00:21:40.320 this to be smaller so what we want to do
00:21:42.900 is we want to roll down this hill
00:21:47.039 we want to roll from wherever we're at
00:21:49.620 in the direction of lower towards the
00:21:53.460 minimum so if we're starting off at some
00:21:56.100 w0
00:21:57.780 how do we know how to go downhill well
00:22:00.539 this is a surface and the surface has a
00:22:04.260 tangent line that's derived that's the
00:22:06.480 derivative right every function you've
00:22:08.760 got its derivative is the tangent line
00:22:11.820 well
00:22:13.559 in Vector world when we're doing
00:22:16.740 derivatives of vectors this is
00:22:20.159 the derivative of our Vector with
00:22:22.380 respect to w
00:22:24.179 this gradient
00:22:25.740 and it is by definition the gradient
00:22:29.120 points uphill
00:22:32.280 so if we wish to go downhill
00:22:36.780 what we need to do
00:22:39.059 is move in the negative direction of the
00:22:42.960 gradient so instead I'm going to use a
00:22:46.020 blue arrow we're going to go that way
00:22:49.620 okay we're going to take a little step
00:22:52.320 towards lower
00:22:54.120 and
00:22:55.620 then we'll recalculate the gradient
00:22:58.980 and take another little step in that
00:23:01.320 direction
00:23:02.520 and so on and so forth until we are

00:23:05.580 happy with our answer
00:23:07.500 okay
00:23:09.179 yeah
00:23:16.020 you're a couple of uh slides early but
00:23:18.960 we will talk about that
00:23:22.799 all right
00:23:24.659 so
00:23:26.159 my slides transitions are bothering me
00:23:28.320 now we've got to introduce some
00:23:30.960 optimization math notation right because
00:23:34.260 how do we know we hit the minimum well
00:23:37.679 we need a notation to describe that
00:23:39.960 so if we have a optimization problem we
00:23:44.580 need to Define we're used to the idea of
00:23:46.860 Max right so the maximum net worth of
00:23:51.539 any person on the planet is something
00:23:53.880 like 181 billion dollars
00:23:58.140 but
00:23:59.820 who is that person
00:24:02.400 that is the notation ARG Max the ARG Max
00:24:06.120 of net worth is the parameter choice
00:24:10.020 that maximizes okay so the maximum value
00:24:14.400 is 181 billion but the choice of person
00:24:17.940 who maximizes the net worth function
00:24:20.640 that's Jeff Bezos
00:24:24.000 optimization problems what we're
00:24:26.460 interested in
00:24:28.200 is ARG Max
00:24:29.940 we want to know what parameter Choice
00:24:32.400 gave us the maximum
00:24:34.679 okay that's our answer that's the best
00:24:37.440 answer
00:24:38.820 make sense so max is the the value of
00:24:42.539 the function but we want to know what
00:24:44.700 parameter Choice gave us that max value
00:24:48.840 okay so my bad slide transitions are
00:24:51.720 still bad
00:24:53.280 so likewise ARG Max has its
00:24:57.900 best friend argman
00:25:00.299 same deal it's the parameter choice that
00:25:03.179 minimizes a function because in machine
00:25:06.000 learning we almost always couch the
00:25:08.159 problem as a minimization problem this
00:25:10.919 is the one we're going to use okay
00:25:13.919 what we're looking for is we're looking
00:25:15.840 for the w
00:25:17.580 that minimizes the function loss of w
00:25:23.400 all caught up
00:25:26.940 okay so if our loss function looks like
00:25:30.360 some Wiggly surface
00:25:32.820 what we're saying is
00:25:34.679 you know I mean if our if we only have
00:25:36.960 one single parameter one scale RW for
00:25:40.020 our model because it's a really lame
00:25:41.460 model
00:25:42.240 you can just draw the Lost surface like
00:25:44.760 this in 2D
00:25:46.679 more generally
00:25:48.659 right it isn't more generally W is a big
00:25:51.659 vector
00:25:53.220 okay you can't really visualize that
00:25:56.820 we showed you a two Vector a vector two
00:25:59.820 space weight but you know obviously it's
00:26:03.000 your most of your problems are going to

00:26:04.980 be a hundred variables so your vector
00:26:07.380 space is going to be hundreds and then
00:26:09.179 you add the loss as another Direction
00:26:12.960 but the concept Remains the Same what
00:26:16.799 we're looking for is this point w star
00:26:22.440 this W star is the lowest it can goest
00:26:27.179 okay
00:26:29.400 that's the ARG men the W that Returns
00:26:32.760 the lowest value of loss
00:26:38.820 now every minimization problem can be
00:26:41.460 flipped by a negative sign and turned
00:26:43.380 into a maximization problem
00:26:46.740 so other fields that do optimization
00:26:50.700 can sometimes called be doing hill
00:26:53.460 climbing instead of loss minimization
00:26:56.340 same damn math
00:26:58.020 right literally just put a negative sign
00:26:59.940 and change things from Min to Max
00:27:02.460 okay but same kind of answers
00:27:05.460 so again you can flip things around by
00:27:09.179 adding a negative sign you can make it a
00:27:11.880 minimization problem by taking a
00:27:14.340 maximization problem and putting a
00:27:15.900 negative sign on the front of it
00:27:21.000 okay
00:27:22.559 generally speaking we're super lazy
00:27:25.620 and we want to make our math as easy as
00:27:28.140 we can make it
00:27:30.000 now the very easiest math and
00:27:32.159 optimization is when problems are in a
00:27:35.940 form which is called a convex function
00:27:38.520 we're going to talk more about that here
00:27:40.919 but let's just start by looking at tools
00:27:43.919 we might use
00:27:45.600 to make our math easier
00:27:49.500 so
00:27:50.760 one of every machine learning neophytes
00:27:54.600 least favorite things is when we talk
00:27:57.059 about monotonic functions monotonic
00:27:59.940 functions what are they
00:28:02.940 anybody anybody happen to know a good
00:28:05.340 definition before I flip over to one
00:28:09.179 ever heard of a monotonic function
00:28:11.340 yeah
00:28:14.460 no good guess but not it
00:28:18.059 yeah
00:28:21.000 always increasing always be increasing
00:28:24.059 you
00:28:24.900 that is a monotonically increasing
00:28:27.120 function or a monotonically decreasing
00:28:29.400 function so
00:28:31.559 the graph here is a big clue a linear
00:28:33.840 function is monotonic
00:28:36.179 as you go from left to right this
00:28:38.880 positive sloped linear function is
00:28:41.100 always increasing
00:28:42.659 right
00:28:45.480 so if it was a negative sloped function
00:28:47.700 it would be monotonically decreasing
00:28:50.340 it never bends back on itself and does
00:28:52.860 that that's not monotonic
00:28:57.179 okay there are other kinds of monotonic
00:28:59.940 functions out there
00:29:01.980 I don't know what my slide transitions

00:29:03.779 suck okay I mean
00:29:06.179 loads of them exponentials are another
00:29:09.120 class they're always increasing it's
00:29:12.360 just that they really blow up fast
00:29:18.779 um natural logs also always increasing
00:29:22.640 but they're asymptotically
00:29:26.520 approaching some value right even though
00:29:30.059 it looks like it's getting flat a
00:29:32.399 natural log is not ever reaching true
00:29:34.620 flatness it's always very very very
00:29:37.260 slowly increasing
00:29:39.059 as we go to Infinity
00:29:41.100 okay these are all monotonic functions
00:29:44.520 now why am I telling you about these
00:29:47.940 things well it turns out monotonic
00:29:50.399 functions are a useful tool for
00:29:52.500 transformation
00:29:55.320 so when we do optimization problems we
00:29:57.419 can transform them into a better form
00:29:59.039 with a monotonic and we'll show you that
00:30:00.899 in a second but in the meantime let's
00:30:03.720 play a game
00:30:05.460 so
00:30:07.260 um hey where'd my thing go
00:30:10.559 that's really useful
00:30:12.539 oh
00:30:15.720 okay now I remembered what this light is
00:30:19.559 ugh slide transitions so bad
00:30:23.100 so if we have a monotonic
00:30:27.059 function f of x that's monotonically
00:30:29.580 increasing
00:30:30.779 is negative f of x
00:30:33.059 monotonically decreasing
00:30:36.380 yes as I already flipped the answer for
00:30:39.480 you a second ago all right
00:30:43.799 what about if we have a monotonically
00:30:47.480 increasing function f
00:30:52.200 if I take the natural log of that
00:30:56.419 monotonic function
00:31:01.380 is it still monotonically increasing
00:31:06.299 the nods are the only thing I'm seeing I
00:31:09.299 saw one thumbs up I didn't see anybody
00:31:11.220 go no Jason no
00:31:13.320 so yes
00:31:15.120 okay
00:31:16.799 it's easier to see if you think about
00:31:19.020 not natural logs and F but like think of
00:31:22.440 like multiplying two linear functions
00:31:24.960 together
00:31:27.419 okay they're both monotonically
00:31:29.520 increasing
00:31:30.840 the multiplication of each element at
00:31:34.260 every Point here would also be
00:31:36.960 monotonically increasing
00:31:40.679 right
00:31:44.039 so if we have
00:31:47.039 um
00:31:48.120 two monotonically increasing functions
00:31:50.340 and we add them together
00:31:53.220 monotonically increasing
00:31:55.919 yeah
00:31:58.500 okay
00:32:00.059 so
00:32:01.200 what about

00:32:02.520 two monotonically increasing functions
00:32:06.360 and take one and subtract it from the
00:32:09.240 other
00:32:16.500 I saw one truck
00:32:20.340 a lot of no responses
00:32:23.760 you got a yes
00:32:25.559 the answer is it depends
00:32:28.500 who's going up faster
00:32:31.440 right
00:32:32.580 if f is going up faster than G yes if G
00:32:37.200 is going up faster than F no
00:32:42.360 okay so monotonic trans yeah
00:32:51.840 are we so
00:32:54.899 um I believe no but also I'm not a
00:32:57.779 mathematician
00:33:00.480 so if they were two flat lines I don't
00:33:02.880 think they're monotonic
00:33:05.520 okay
00:33:07.440 so the definition of monotonic is for
00:33:09.840 all possible pairs of points when point
00:33:14.519 one is to the right of point two then
00:33:17.940 the function of point one has got to be
00:33:20.340 bigger than the point function of point
00:33:22.019 two
00:33:23.399 so
00:33:25.080 if we take a monotonic transformation
00:33:29.220 of any function we don't change where
00:33:33.419 the optimum is
00:33:36.419 okay so if you have a loss function
00:33:39.600 and you run the loss function through a
00:33:41.880 monotonic function you don't change the
00:33:44.820 location of the best answer
00:33:48.000 I mean you change the the loss right you
00:33:51.600 run a whatever the vertical is you
00:33:53.519 change the Max and you change the min
00:33:55.860 but you don't change the shape
00:33:59.519 okay think of it this way if I have
00:34:04.860 a loss function like that
00:34:07.200 and I run it through
00:34:10.080 well let me use a different color
00:34:12.239 okay I run it through a monotonic
00:34:14.280 function
00:34:15.359 what I get out
00:34:17.280 is maybe this
00:34:23.639 okay
00:34:27.960 but the shape is the same and where the
00:34:32.339 absolute minimum is is the same
00:34:37.020 the values have changed but not the
00:34:39.239 shape to the point where the minimum has
00:34:42.839 changed location
00:34:46.139 got it kinda
00:34:48.899 so these kinds of monotonic
00:34:51.060 Transformations are how we make math
00:34:52.918 easier for ourselves
00:34:54.899 we can take a loss function which is not
00:34:57.060 lovely and run it through a monotonic
00:34:59.280 function to make it easier to deal with
00:35:07.260 okay
00:35:08.339 ah slide transitions so if for instance
00:35:12.839 we have yeah okay so I just did the
00:35:14.460 drawing right so if we take a linear
00:35:17.460 function
00:35:18.480 G sub V and
00:35:21.780 apply it to a monotonic function we just

00:35:26.040 get this new linear function of the loss
00:35:28.560 function and stuff stays the same
00:35:32.520 okay one of these that we use a lot
00:35:35.700 is the natural log the natural log like
00:35:38.520 we said is monotonically increasing and
00:35:41.160 the reason we use it is because it has
00:35:44.040 some lovely properties
00:35:46.619 so
00:35:48.180 let me
00:35:49.740 um
00:35:50.760 yeah so just a reminder that we don't
00:35:52.560 change the argument or ARG Max by
00:35:54.480 applying this
00:35:57.480 so here's an example
00:36:01.260 logistic regression is something that
00:36:03.240 we're going to talk about in much
00:36:04.440 greater detail I think in week three
00:36:08.460 here's the rough sketch of what we're
00:36:10.859 trying to accomplish
00:36:13.079 so for a bunch of data points we want to
00:36:16.920 maximize the probability of correct
00:36:19.560 classification of each data point so you
00:36:22.320 should read this as the probability of
00:36:25.740 getting of saying that a data point is
00:36:29.099 class y
00:36:30.859 given an input X and a particular weight
00:36:34.560 vector
00:36:35.940 okay
00:36:36.960 so this is our classifier X and sorry W
00:36:41.460 is our classifier and we take a data
00:36:44.339 point x and we run it through the
00:36:45.780 classifier and the classifier says I
00:36:49.380 think with probability this that it's
00:36:52.680 bird class or I think with probability
00:36:55.859 that
00:36:57.240 that it's diabetes or whatever we're
00:37:00.240 classified okay
00:37:02.099 so we want to maximize the probability
00:37:05.460 of being correct
00:37:08.280 so we want to maximize the probability
00:37:12.240 of every data point and because we're
00:37:15.599 just going to assume that every data
00:37:17.400 point getting it right on the
00:37:19.740 classification is statistically
00:37:22.079 independent with every other data point
00:37:26.099 okay so when you roll dice the
00:37:28.740 probability of getting two dice to both
00:37:30.960 turn up Snake Eyes both turn up ones is
00:37:35.220 1 6 times 1 6.
00:37:39.480 when when probabilities are independent
00:37:42.480 they multiply together so that's why
00:37:45.300 this damn thing is a product
00:37:47.220 okay
00:37:48.780 so we hate products products suck
00:37:52.260 products suck computationally
00:37:54.720 so if we want to do things quicker
00:38:00.119 let's turn it into a sum
00:38:02.339 how do we do that well the natural log
00:38:04.680 of a product
00:38:07.560 you can pull the sum from outside of the
00:38:10.800 natural log and put it in front and
00:38:13.380 that's equivalent to the natural log of
00:38:15.660 the product natural log of the product
00:38:17.400 is the same as the sum of natural logs

00:38:20.700 good old-fashioned calculus seal
00:38:24.359 okay
00:38:25.500 so what we can do is we can take the
00:38:28.140 monotonic transform of the probabilities
00:38:30.200 by putting a natural log around it turn
00:38:32.880 it into a sum for our computational
00:38:34.859 convenience
00:38:37.380 and also making everything quite clear
00:38:40.380 and lovely and then because we're
00:38:42.240 machine learning instead of maximizing
00:38:44.160 probabilities we're going to minimize a
00:38:45.839 loss function by just flipping a
00:38:47.280 negative sign on top of that
00:38:49.980 okay this don't worry about picking this
00:38:52.800 up right now we're going to cover it in
00:38:55.560 detail in week three
00:38:57.720 I just want to show you we actually do
00:39:00.180 this this is not me just like throwing
00:39:02.520 math at you for the no particular reason
00:39:05.099 okay
00:39:07.200 okay so monotonic functions
00:39:09.960 we use them to make our math life easier
00:39:12.960 anything is monotonically increasing or
00:39:15.900 decreasing can be applied to a loss
00:39:18.839 function without changing the optimum
00:39:21.180 value coming out of it
00:39:26.820 okay questions on that before we
00:39:29.280 carriage return line feed
00:39:34.260 40 okay
00:39:37.079 so in general
00:39:39.480 it's important to think of optimization
00:39:41.960 as something we are trying to get right
00:39:45.660 we have a training set and a test set
00:39:48.720 okay what we really want to do is we
00:39:51.300 want to make our test set to Performance
00:39:53.460 better but we can't optimize directly on
00:39:56.160 it
00:39:57.359 because that would be overfitting right
00:39:59.460 we've talked about that so as a proxy we
00:40:02.460 optimize on the training set
00:40:04.440 let me introduce a new piece of notation
00:40:06.599 for you see this one
00:40:09.300 this is a vector
00:40:11.220 which says Hey I want to minimize the
00:40:15.660 number of wrong predictions I have
00:40:18.420 this Vector is one every data point
00:40:21.720 where our prediction doesn't match the
00:40:26.220 truth
00:40:27.599 if the truth doesn't match the
00:40:29.220 prediction I'm going to fill this Vector
00:40:31.440 up with ones if the TR if the prediction
00:40:34.619 matches the True Value this Vector is a
00:40:37.079 zero
00:40:38.099 so what this does is if we sum up this
00:40:41.700 ones vector
00:40:43.260 and divide by the number of data points
00:40:46.859 this is
00:40:49.619 what
00:40:54.359 yeah
00:40:56.520 yeah
00:41:00.599 it is
00:41:02.040 indeed it is the ratio of right to wrong
00:41:05.460 right
00:41:07.020 so I'm not sorry it's not the ratio of

00:41:09.000 right to wrong it is the fraction of
00:41:11.400 wrong
00:41:13.320 okay
00:41:15.660 so that's called the misclassification
00:41:17.400 error and generally speaking that's our
00:41:20.400 direct concern is minimizing
00:41:22.140 misclassification if we're doing
00:41:23.579 classification
00:41:25.800 so
00:41:28.200 um
00:41:29.099 what we're going to do is realize that
00:41:33.140 convex functions are the single easiest
00:41:36.060 functions we can optimize there is an
00:41:38.940 entire textbook on doing convex
00:41:41.579 optimization that is freely available I
00:41:44.400 when I was in your shoes as an undergrad
00:41:46.859 took an entire semester on convex
00:41:50.339 optimization
00:41:52.079 we don't have that here right
00:41:54.680 probably in the math department there is
00:41:56.940 such a class
00:41:57.900 but if you want to get more into the
00:42:01.260 Weeds on convex optimization this is
00:42:03.359 freely available online it's kind of the
00:42:05.040 gold standard
00:42:06.420 but here's the two slide summary
00:42:09.000 of the entire textbook
00:42:11.339 ready for non-local
00:42:13.859 and local
00:42:15.300 Solutions
00:42:18.480 so
00:42:19.980 in general we don't know the shape of
00:42:23.339 the Lost surface
00:42:25.980 okay the lost service could be super
00:42:28.680 ugly like the drawings that we had
00:42:30.060 earlier
00:42:32.400 okay but somewhere in that lost service
00:42:35.760 lost service
00:42:38.339 there is a point this red point the W
00:42:41.880 asterisk okay
00:42:44.160 this bit right here
00:42:46.800 which is the lowest
00:42:48.839 of all the possible choices nothing is
00:42:51.780 lower than this Global Optimum
00:42:54.960 how should you read this math
00:42:58.020 so for every possible Solution that's in
00:43:02.460 the domain
00:43:03.780 of w domain is like all the possible
00:43:06.599 values that W could take
00:43:09.720 or we could call it the domain of all
00:43:12.420 possible
00:43:13.760 Solutions all possible Minima
00:43:17.819 okay so in that case what we'd be
00:43:19.980 looking at is all the places where the
00:43:24.060 derivative is zero
00:43:26.640 maybe in your math class you call those
00:43:28.500 inflection points
00:43:30.180 okay
00:43:31.920 so it's either every single point
00:43:34.920 that could exist in the W Vector space
00:43:39.420 or it's all the possible solutions that
00:43:44.040 could be the global
00:43:45.900 Optimum
00:43:47.400 okay

00:43:48.420 either way it doesn't matter how you
00:43:50.040 couch it but that's how you should read
00:43:51.420 this
00:43:52.500 Omega
00:43:54.000 okay
00:43:55.200 so the domain of possible solutions
00:43:58.319 so the global Optimum is the one which
00:44:00.480 is got the lowest there is nothing lower
00:44:03.420 than it
00:44:05.160 okay it's less than or equal to all
00:44:06.960 other possibilities
00:44:08.880 however
00:44:10.319 there's going to be loads of Minima so
00:44:13.020 these are Minima right
00:44:15.839 these are all Minima
00:44:19.020 and they're all places where the
00:44:20.640 derivative is zero
00:44:23.280 right
00:44:25.920 so there's going to be other ones that
00:44:28.500 are locally optimal so for some
00:44:31.680 neighborhood
00:44:33.119 near this location this is the lowest
00:44:36.960 value that's possible
00:44:39.660 okay
00:44:41.700 so there are local Optima and there is
00:44:44.040 one single Global Optimum
00:44:47.700 generally speaking we would love to find
00:44:49.859 the global but sometimes we're doing
00:44:53.460 good enough to find a local
00:44:56.880 okay
00:44:58.380 so things we often need to do to solve
00:45:00.540 an optimization problem
00:45:02.099 so check to see if a given possible W
00:45:05.640 value is a Minima of any sort
00:45:09.119 okay
00:45:11.099 and
00:45:12.420 calculate the derivative or maybe the
00:45:16.859 second derivative or sometimes even the
00:45:19.560 third derivative
00:45:21.060 in order to take a step towards that
00:45:24.660 Optimum okay just like we were showing
00:45:27.599 in the beginning of class
00:45:31.619 so why because we want to roll downhill
00:45:35.819 right and again the negative gradient
00:45:40.079 the negative first derivative
00:45:42.420 that is the direction of downhill
00:45:45.839 now why do we calculate sometimes the
00:45:48.839 second
00:45:50.160 derivative
00:45:51.420 because that can take us more directly
00:45:53.880 downhill there are places where
00:45:56.099 following the first derivative is not
00:45:58.079 the fastest way to get to the bottom
00:46:00.839 okay we'll talk much more about that at
00:46:03.599 a later date
00:46:04.920 but that's why there is a ton
00:46:08.339 that's why there's a textbook called
00:46:10.319 convex optimization and not just these
00:46:12.480 three slides okay because there's ways
00:46:15.839 to make this whole system work better
00:46:20.040 but this is called gradient descent
00:46:22.200 right running downhill from start
00:46:25.140 someplace in the solution space and roll
00:46:27.839 downhill towards a better answer

00:46:34.079 so convex loss functions however make
00:46:37.560 optimization easy
00:46:39.599 so if something is convex then it there
00:46:44.460 is only
00:46:46.680 one Optimum
00:46:48.960 and it is the global
00:46:52.200 okay
00:46:54.060 so that is the best possible case
00:46:57.599 let's talk about the mathematical
00:46:59.520 definition of convexity for one second
00:47:02.940 okay here are three functions which of
00:47:06.000 them are convex
00:47:07.560 somebody who isn't normally answering
00:47:11.400 yeah
00:47:12.960 a is convex what about B
00:47:16.200 no what about C
00:47:18.180 yes
00:47:19.319 okay
00:47:24.020 cool
00:47:25.880 all right how do we know it's convex
00:47:30.180 all right here's a mathematical
00:47:32.160 definition of convexity but let me give
00:47:34.500 you the intuition because it's not as
00:47:36.420 scary as it looks
00:47:38.339 here's the idea okay pick any two random
00:47:42.300 points on a curve
00:47:44.640 for any pair of points you can pick
00:47:48.060 for all pairs of points you pick
00:47:51.480 this inequality has to be true
00:47:55.020 draw a line between those two points
00:47:59.339 okay
00:48:00.599 so if I'm drawing a line between w0 and
00:48:04.380 W1 straight
00:48:08.579 that line has to be
00:48:11.480 equal to or greater than the height of
00:48:15.720 actually following the Curve
00:48:17.940 so as I move back and forth on this line
00:48:21.300 this line has to be above moving back
00:48:25.560 and forth on the curve for the same x
00:48:27.599 value okay so this point is above this
00:48:31.020 point this point is above this point
00:48:33.060 this point is above that point
00:48:35.640 that means it's convex
00:48:38.640 okay
00:48:39.839 just look at how the math is set up I
00:48:42.480 have some a term the a term is how far
00:48:46.440 back and forth I am from w0 to W1 when a
00:48:50.819 is equal to 1 then all my weight is over
00:48:54.420 here I'm over on this side when a is
00:48:56.819 equal to zero I'm all the way over here
00:48:59.280 on this side
00:49:00.839 okay that's how that works this part
00:49:03.839 this is the line
00:49:06.780 and this part is the loss function going
00:49:09.180 back and forth from w0 to W1 on the loss
00:49:12.119 function itself
00:49:14.339 okay
00:49:16.079 so how could we break that with a
00:49:18.720 non-convex function
00:49:20.280 well here's two points I can pick
00:49:23.400 such that there's a place here
00:49:26.040 where the Curve
00:49:28.140 is above the line
00:49:30.540 if I can pick a pair of points where

00:49:33.119 anywhere where the curve goes above the
00:49:36.000 line in between those two points it's
00:49:38.099 not convex
00:49:40.500 okay
00:49:41.700 so this guy here convex I can pick two
00:49:46.980 points
00:49:49.079 such that
00:49:51.480 the curve and the line are the same
00:49:55.020 anything that's got straight sides
00:49:57.900 is not strictly convex
00:50:01.560 but it is convex strictly convex would
00:50:04.800 be getting rid of that equal sign
00:50:08.520 okay
00:50:09.900 so things that have straight lines in
00:50:12.420 them not strictly convex because you can
00:50:16.500 pick two points on the line and be right
00:50:18.960 on top
00:50:20.339 okay how am I doing for time I've I've
00:50:23.520 lost
00:50:24.660 so we are out of time
00:50:27.780 you can take a look at the little
00:50:29.579 convexity quiz on your own that's in the
00:50:32.339 rest of these slides
00:50:34.079 and I will see you next time
00:50:38.880 have fun