* Different types of norms
* We’ll be trying to minimize squared loss and generally just call it regression
* As lambda increases, the training loss increases and the test loss falls to a certain point before rising later on
* Choice of lambda depends on the training size
* If lambda is too large you are underfitting
* This idea also works with other losses like hinge loss, log loss, etc.
* The loss function for regularization is still readily decomposable
* Only consider models of a class that fit within a region – norm constrained models
* The L2 norm is decreasing as lambda increases
* Heat map on slide 15 is the loss function
* C is constraining so you must be in some region
* If you can’t move towards the minimum of the loss function, you must make a step along the boundary – effectively projecting the gradient onto the boundary of the allowed model region
* Eventually the gradient of the loss is aligned with the gradient of the constraint boundary (may not be the same magnitude)
* Claim – solving the lagrangian solves the norm constrained training problem
* Hallucinating data points – try and take the 2 lambda w term on slide 20 and put it into the summation
* Introduce artificial points and they give the same gradient
* Multi-task learning – you have two training sets that are fairly similar
  + 1 option is just to train them separately, but both models may have high error because the training set is small
  + Option 2 – train them together, and have it so that the weight vectors w and v are close to each other (multi-task regularization)
* L1 – sum of the absolute values of the entries
* L1 function is not differentialable everywhere
* We use a subgradient
* Subgradient is all the slopes that lie beneath the curve
* For L1, the subgradient is the set from [-1, +1]
* At a corner, there’s a continuous range from the subgradient
* One of those subgradients aligns with the gradient, so we’re at optimality
* This implies many of the solutions are at the corners since the corners allow a wide range of gradients
* Many of the solutions are at the corners
* A vector is sparse if we have low l0 norm
* But L0 norm is not continuous
* L1 induces sparsity because solutions are at corners
* Sparsity is important because of computational and memory efficiency; can exploit the fact that w is sparse
* Stochastic gradient descent has a hard time hitting zero
* Lasso is often not as accurate in many cases
* Instead, people will use lasso to identify the dimensions that matter and then rerun only using those dimensions that didn’t go to zero
* Lasso makes it easier to see what’s happening and which dimensions are important