* Margin is typically distance from hyperplane to the nearest point
* Margin is a measure of the stability
* There are multiple identical solutions because you can scale w – w is in the numerator and denominator
  + One solution is to fix the denominator and say the norm of w is 1
  + This is still challenging to optimize!
  + Instead we often enforce that the numerator is fixed to 1
* Hard margin support vector machine
  + There will always be an “i” that achieves the minimization because if the closest points didn’t exist then w would be rescaled
* Removing points not on the margin will not affect the solution
* Performing leave one out cross validation would give the same model almost every time except for if you point you remove is the one on the margin
* Two potential mistakes this formulation could make but we will lump them into the same thing
  + We will use a soft margin
  + We will allow some set of data points to violate the margin
  + A “slack” variable in the optimization
* If C is set large then the minimization problem will care about the second term – the one with C
* The slack variable is some large positive number (? This is probably wrong)
* If C is very high, we are intolerant of margin violations
* If C is very small, then we don’t really care as much about margin violations
* Anything that sits on or violates the margin is a support vector
* Only the equalities matter – you want the slack variable as small as possible so it’s exactly the distance of the margin violation
* The slack variable is part of the minimization problem
* C MUST be non-negative; will almost always be positive because 0 will lead to degeneracies
* The SVM is optimizing the hinge loss function
* Loss increases linearly as the score of the function moves away from +1
* The loss function is equal to the slack variable
* Hinge loss can also be a surrogate to 0/1 loss – it has some nice properties
  + Square loss says a positive value if the target is 1 is wrong, which is odd
  + Hinge loss says that it’s correct
  + The penalty scales linearly
  + Square loss is much more sensitive to outliers than hinge loss
* The hinge loss has better behavior than 0 1 loss but it’s not smooth
* There is a point of non-differentiability which makes it harder to optimize
* Can use sub-gradient descent
* Conditional probability of an event y given (x, w, b) is what logistic regression gives
  + Let’s us model not just classification but the probability
* Each data point in S is sampled independently so we model the probabilities independently of each other – to maximize the joint probability we maximize the product of all of them
* The SVM is often better at classification
  + But SVM is not well calibrated to be used as a probability
* Loss function for the logistic regression is log loss
* There’s a family of loss functions that are a combination of hinge and log loss
* Trivial neural net with one neuron looks like a logistic regression
  + May apply nonlinear transforms that aren’t sigmoids
* Deep neural networks are just have many layers
* Deep neural nets are sometimes easier to train than fat ones
* Training neural nets is done using SGD and backpropagation, which is just chain rule
* A different loss function could have the diagonals of a cost matrix treated differently
* One simple way is to just weight one misclassification much greater than the other