Q12: When only a small fraction of the features are expected to have non-zero weights, one should use an L0 or L1 penalty to zero out features, but when most of the weights are non-zero, as is the case here, then there is generally little benefit to zeroing out the few non-significant features. The go-to method is then a ridge (L2) penalty. (For linear regression, people often use elastic net, which includes ridge as a special case.)

Would not an L1 penalty be similar in terms of accuracy to a L2 penalty in this case? Yes, it isn't obvious which one would be better; ***We'll give credit for both.***

Q15: The answer can be option (a) Model A or option (d) Model D
A student writes that for L2 regularized (ridge) regression:
If we use $\lambda=1$, $\theta=[0.60, 5333]$, which is option **A.** If we use $\lambda=5$ we get option D**.**
We will give credit for a,d or e on this.

Q16: For a function *k(x1,x2)* to be a kernel function, it must be the case that any kernel matrix K generated using it must be positive semi-definite (PSD). If K is not PSD, the k() is not a kernel function. If K is PSD, we haven't *proved* anything, just suggested that it is likely to be a kernel.

Q18: stepwise and streamwise methods are both greedy search algorithms. When applied to a non-convex problem (e.g. L0-penalized regression), we cannot guarantee which one will be better or worse. It is the case that stepwise is "less greedy", but sometimes streamwise will stumble onto a better solution.

Q23: "A larger stepsize in gradient descent can lead to faster convergence, but lower accuracy." This is true. It can, and often does, lead to faster convergence. (It can also lead to lots of fluctuations and no convergence, but the question does not claim it always leads to faster convergence.) It also can, and often does, lead to lower accuracy, but might happen to bounce you to a higher accuracy solution.

Q25 The hyperplane are the points where ***w*x****=0.* Thus, in a two dimensional classification problem your decision boundary will be a line (one dimensional), in three dimensional space the decision boundary will be a plane (two dimensional), etc. So ***w*** is p-dimensional, the decision boundary ***w****^T****x*** = *0* is p-1 dimensional.

Q27: SVM's have a global optimum (even if it is not separable). The loss function is convex. Neural network's in practice, virtually never find their global minimum. Dropout helps bounce the network out of local minima, but does not come close to finding global optima. (And we mostly don't want global optima, which would be seriously overfit; hence we usually only partially converge the gradient descent.) Neural Nets have a set of global optima (a bunch of different weight vectors, all of which give identical predictions), but we cannot in general find the optimum.

Q28: Error (e.g. from a neural net or a regression model), as used in gradient descent, is a function both of the training data and the current estimate of the model weights. If you use different training data, you'll get a different error. Since Error changes with the training data, so does the gradient (d Error/dw), and gradient descent will follow a different path, and, in general, converge to different weights.

Q33: When fitting a model using Adaboost, training should stop before the training error reaches zero. Apologies, this was not well-explained in class, but it is false. One way to picture this is to image fitting a large margin method like a perceptron to fully separable data. In this case, just getting zero training error is not optimal. You want to keep increasing the margin even after you have perfect classification. Boosting is, of course, not exactly the same, and since it fits a much more complex model than a hyperplane, can overfit. (So people usually stop boosting before complete convergence.) However, boosting tends not to overfit as badly as many methods, and it is often the case that it is valuable (in the sense of giving optimal test error) to keep training even past the point of having zero training error. It can, in effect, still be the case that adding another weak learner to the model in effect increases the margin.

Q34: A student writes "using a smaller dataset will increase the overfitting" This is wrong in the case of weak learners. It will make each of the weak learners more different from the other weak learners. (They each overfit in their own way.) So the combined model (the "ensemble") overfits less, since it combines more diverse weak learners.

Q35: For gradient boosting, we fit in the "pseudo-residuals." However (perhaps confusingly) for the case of regression, these are exactly the (standard) residuals. Also note that in boosting lecture slide 10, it is stated that for squared error the pseudo-residuals are the standard residuals.

Q37 Adaboost will converge to the minimum possible error that one can get on the training error. If the same ***x*** value has two different class labels, this will still not be zero.
Credit given for either answer.