The second midterm (November 21 in lab + take home) covers in-class material days 9-19, labs 5-8, and reading weeks 5-10. For the in-lab portion (short answer questions), you may bring a 1 page (front and back), hand-written "cheat-sheet" (created by *you*), and a calculator, but no other notes or resources. (You shouldn't need a calculator, but it may make things quicker). I have put vocab in blue.

1. Logistic Regression
   - Why don't we use linear regression for classification problems?
   - Logistic function of a linear transformation of $\boldsymbol{X}$ as our model in logistic regression.
   - Logistic regression creates a *linear* decision boundary (visualize for $p = 1, 2$).
   - Idea of a likelihood function and finding the MLE (maximum likelihood estimator).
   - Bernoulli random variable example of MLE calculation.
   - In logistic regression our cost is the negative log likelihood.
   - Intuition behind the cost function.
   - Derivation of SGD for logistic regression, relationship to linear regression.
   - Idea of multi-class logistic regression (not the mathematical details).
2. Evaluation Metrics
   - Confusion matrices as evaluation metrics for classification problems
   - Binary classification problems with "positive" (atypical) and "negative" (typical) results
   - Confusion matrices for the above specific case: FP, TP, FN, TN
   - Precision: fraction of flagged examples that are truly positive (intuition: "purity")
   - Recall/TPR: fraction of true positives that we found (intuition: "completeness")
   - Using the FPR and TPR at different thresholds to create a ROC curve
   - What is a "random guessing" ROC curve and what is the ideal ROC curve?
   - Where you want to be on the ROC curve depends on the application
   - Cross-validation: procedure and goals (hyper-parameter selection, test error distribution)
3. Ensemble Methods
   - Idea of using an ensemble of classifiers (ideally with low bias) to reduce variance
   - To test, let each classifier in the ensemble "vote" (could be weighted or unweighted)
   - Bootstrap: sampling from our data with replacement (usually keeping n the same)
   - Bagging (Bootstrap Aggregation): create a classifier for each bootstrapped training dataset
   - How does averaging the results of many "weak" classifiers reduce the overall error?
   - Ensemble notation and example of reducing the error via bagging!
   - Random Forest classifiers as ensembles of decision stumps (or small-depth trees)
   - What was the idea behind Random Forests? Why might they be better than regular Bagging?
   - AdaBoost: upweight training examples that were classified incorrectly in the previous iter

- AdaBoost details: weighted error, score, update example weights, testing with weighted vote
- Decision Trees with weighted examples (how do we modify the probability calculations?)

4. Support Vector Machines

- Idea and equation of a separating hyperplane (weight vector points toward the + side)
- Perceptron algorithm and derivation of the weight updates; perceptron cost function
- Perceptron weight updates: geometric interpretation and gradient descent interpretation
- Guarantees and limitations of the perceptron algorithm
- Support Vector Machines (SVMs) can find the maximum margin hyperplane
- What are support vectors? What is the geometric (γ) vs. the functional ($\hat{\gamma}$) margin?
- How we used the functional and geometric margins to cast SVMs as an optimization problem
- Motivation and method of Lagrange multipliers, application to SVMs
- High-level steps of transforming the SVM Lagrangian into a problem involving only α values
- What do these α values represent and how can we use them to find $\vec{w}^*$?
- Reformulation of SVMs as maximizing $W(\vec{\alpha})$ uses only inner products between examples
- Idea of a kernel function and how it can replace the dot product (not Gaussian kernel details)
- Incremental SVM optimization algorithm for training
- Soft-margin SVMs and associated optimization problem (not how to solve)

5. Neural Networks

- What is a Neural Network (NN)? Motivation and goals when using them
- High level idea of training using gradient descent on the loss function
- Fully Connected architectures, dimensionality analysis, parameters vs. hyperparameters
- Choice of activation function, pros and cons of sigmoid, tanh, and ReLU
- Softmax function as the activation function for the last layer, cross-entropy loss after that
- Training: how to initialize the weights/biases, what is the point of mini-batches?
- Motivation behind Convolutional Neural Networks (CNNs); application to images
- CNN architectures: idea of 3D volumes, typical steps CONV, RELU, POOL, FC
- CONV layer details: filters computing cross-correlations, slide filter over width and height
- Dimensionality analysis (shapes of filter weights/biases, shapes of input/output)
- Skip: dropout, regularization, any pooling besides 2×2 with stride 2