Lecture	Topic
1	Matrix multiplication four different ways: Collection of inner products, Sum of Outerproducts, Collection Matrix-Vector Products & Collection of Vector-Matrix products. Setting up systems of equations as a matrix-vector product. The "dot" or elementwise function application operator. Expressing summations and weighted summation representation of integrals as quadratic forms e.g. sum_{ij} a_{ij} w_i w_j = w' A w where A is a discretization of a function in 2-D and w_i and w_j are integration weights *Companion reading: "Block matrix methods" by Tim Davis* *http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.6.5182* *In HW have them write familiar convolution as a matrix-vector product.* Lab/Computational Component: Learn2Classify codex Setting up a system of equations for linear classification. Setting up system of systems for non-linear classification with a single neuron and dot operator. Intro to gradient descent, stochastic gradient descent and accelerated version (Nesterov). Learning to recognize 2 digits and their own handwriting using code built by them.
2	More on matrix multiplication and deep networks as being compositions of dot operator and matrix-matrix multiplication. Permutation matrices and their properties. Lab/Computational Component: Learn2ClassifyMany codex Setting up a system of equations for linear multi-class classification. Setting up system of systems for non-linear multiclass classification with a single neuron and dot operator. Definition of linear separability vs non-linear separability. Introduction to idea that there are many separating hyperplaces and some problems (e.g. bull's eye data set) are intrinsically NOT linearly separable but ARE separable using neural network activation functions. Important to see how class encoding vector interacts with activation function -- +/-1 encoding is compabitle with tanh but not sigmoid. Compute probability of classification as a function of number of classes for handwriting. See that it goes down quite a bit.
3	The Singular Value Decompsotion – the full, economy and truncated form. Linear dependence & independence, subspaces, span, nullspace, dimensionality, rank, and basis. Orthogonal matrices and their properties. Spotting low-rank matrices and their SVD. Four fundamental subspaces and their bases using SVD. Lab/Computational Component: SVD 2 Background Subtract codex How to induce low rank by reshaping matrix. Spotting low rank matrices. Truncated SVD. Information conveyed in each of the singular vectors.
4	Orthogonal bases vectors and obtaining coordinates of a point with respect to an orthogonal basis. Matrix vector products interpreted using SVD. How the "U2" and "V2" in the SVD are random and why that is ok – the QR decomposition (codex) and Lab/Computational Component: Learn2CompleteMatrix & Learn2CompressAndDenoise

	codexes How linear dependence can be taken advantage of to fill in missing entries of a low-rank matrix. Applications to image in-painting (MIT logo) SVD to compress an image vs SVD to denoise low-rank matrix plus noise. Difference between approximation error and denoising error. Random matrix theory insights on singular value spectrum perturbation theory and why matrix completion works (= operator norm of the equivalent random delta matrix is small with very high probability – students work through Latala's theorem and have to show numerically that theorem is true).
5	Systems of equations, their geometry and when exact solutions are possible. The principle of least-squares. Deriving the solution using just the SVD and making contact with Moore-Penrose pseudoinverse. Spotting when solution is unique versus when it is not. Min norm formulation for latter case to get uniqueness. When is error 0? Nearest subspace classification Lab/Computational Component: SVD2Classify codex Nearest subspace based classification and handwriting recognition using it. Physical interpretation of eigen-images and how digits are "mixed" eigen-images. How probability of correct classification depends on the rank. What happens when we use full rank? Show that we get 95% accuracy whereas using earlier methods we got 80%.
6	Understanding the linear transformation of a vector via y = Ax via the SVD viewpoints. Express A = USV' and then y = Ax = sequence of three linear operations. Interpret each operation geometrically and using svdshow utility. Now state and solve the beamforming problem of maximizing norm(Ax) subject to norm(x)=1. Motivate with wireless and array processing applications. Express U1 and V1 as a manifold optimization problem. PCA via SVD. Uncorrelated matrices and variables. Show how by changing cost function from variance to kurtosis we go for PCA to ICA. Lab/Computational Component: Learn2Project codex Obtain the U's and V's via a manifpold optimization problem (using pymanopt). Then understand properties of it and when the answers can be random (= identity covariance). Describe PCA factorization. Develop ICA as a change of cost function from variance to kurtosis. Unmixing of images, sounds and signals via ICA.
7	Mid-term Exam
8	Why does ICA work? Proof of separability and importance of kurtosis as a discriminating function. Properties of cumulants and how additivity of cumulants is important property needed for why ICA works. Free component analysis – random matrix theory cumulants. Lab/Computational Component: ICA codex Proving ICA and seeing the failure modes. Patch ICA on patches of image revealing dictionary elements. Faster native Julia code for ICA. ICA factorization and ICA on the background subtraction video revealing additional structure that PCA cannot.
9	More ways fitting of equations – regularized least squares. L1, softmax and logistic

	regression. Projected least squares. Convex optimization 101 – what are convex sets, how to setup projections and fast algorithms.Support vector machines (as discriminant analysis with modified cost function). Multi-layer perceptron, Deep networks. Classification using above. When theoretical guarantees are possible. Connections to statistical theory (MAP, etc) – why they aren't for neural networks. Lab/Computational Component: Learn2Fit Codex Designing softmax regression from scratch – by taking derivative and programming. Same for other techniques. Evaluating performance of all again MNIST and Fashion-MNIST datasets.
10	Procrustes analysis – solution via SVD and modification for setting where there is scaling and translaton. Lab/Computational Component: Learn2Align codex
11	Multidimensional scaling – distances to measurements. Solution as an "eig" problem. Formulation as a semi-definite programming optimization problem. Metric MDS vs non-metric MDS – how changing notion of distance changes embedding and can make previously indistinguishable clusters separate. Connections to unsupervised learning and cluster analysis. Lab/Computational Component: Learn2Embed and Learn2Cluster codex Learn2Cluster introduces Kmeans – students write up their own code. Learn2Embed has them program MDS in incarnations taught and see how changing distance matrix from Google map data (road distance vs lat/long computed distances) affects the embedding (especially when cities are separated by a lake because road has to go around lake!)
12	Graph Partitioning and Spectral Clustering. Partitioning via modularity maximization and modularity matrix vs using laplacian matrix. Lab/Computational Component: Learn2Partition codex: Modularity maximization via spectral and SDP method. Application to MLB league data. Multi-class partitioning. Spectral clustering. What happens when graph is very sparse? Non-backtracking matrix 101 and other methods for very sparse data
13	Decision Theory 1010 and Robust Statistics. Flase positive vs false negatives. Naive bayes estimator. Influence function and role of outliers. Outlier detection.Robust least squares – robust penalty functions. Lab/Computational Component: Learn2DetectOutliers codex How median is more robust measure of centrality than mean. Heavy tailed distributes and how estimators behave differently. How extreme sparsity can corrupt methods (especially spectral methods). Numerical schemes for robust regression. Influence functions

14	1D and 2D Convolution and some properties. Edge detection. Blurring/deblurring. Setting up systems of equations. Lab/Computational Component: LearnCNN codex Convolutional neural networks for image detection, classification and in-filling.
	Other (optional) codexes for students wanting to do more : Dictionary learning based denoising Generative adversarial networks Boltzmann machines and autoencoders Other application codexes Style transfer