6.036 Official Cheat Sheet

- A (hyper-)plane is a set of points $x \in \mathcal{R}^d$ such that $\theta \cdot x + \theta_0 = 0$. Vector θ is normal to the plane. The signed distance of any point x from the plane is $(\theta \cdot x + \theta_0)/\|\theta\|$. The value of distance is positive on the side where θ points to, and negative on the other side.

- A linear classifier with offset: $h(x;\theta) = \text{sign}(\theta \cdot x + \theta_0)$

- Training error (classification error): $\epsilon_n(h) = \frac{1}{n}\sum_{i=1}^{n}[[y^{(i)} \neq h(x^{(i)})]]$

- Loss functions:

$$z = y(\theta \cdot x + \theta_0) \text{ (agreement)}$$

$$\text{Loss}_{0,1}(z) = [[z \leq 0]]$$
$$\text{Loss}_h(z) = \max\{1 - z, 0\}$$

- SVM: Finds a large margin classifier by minimizing

  $\frac{1}{n}\sum_i \text{Loss}_h(y^{(i)}(\theta \cdot x^{(i)} + \theta_0)) + \frac{\lambda}{2}||\theta||^2$

  which can be done using stochastic gradient descent (Pegasos).

- Linear regression:
  finds the parameters of a linear predictor $\theta \cdot x + \theta_0$ by minimizing

  $\frac{\lambda}{2}||\theta||^2 + \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \theta \cdot x^{(i)} - \theta_0)^2/2$

- Low-rank matrix factorization for collaborative filtering: Minimize

  $J(U,V) = \sum_{(a,i)\in D}(Y_{ai} - [UV^T]_{ai})^2/2 + \frac{\lambda}{2}\sum_{a=1}^{n}\sum_{j=1}^{k}U_{aj}^2 + \frac{\lambda}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}V_{ij}^2$

  Can be solved iteratively by fixing one matrix and using linear regression to find the other.

- Kernels: $K(x,x') = \phi(x) \cdot \phi(x')$

Kernel	form
Linear	$x \cdot x'$
Quadratic	$x \cdot x' + (x \cdot x')^2$
Radial basis	$\exp(-\|\|x - x'\|\|^2/2)$

- Kernel Perceptron (with offset): Cycles through each point t=1,..n and checks if $y^{(t)}(\sum_{i=1}^{n}\alpha_i y^{(i)}[K(x^{(i)}, x^{(t)}) + 1]) \leq 0$. If true, $\alpha_t = \alpha_t + 1$.

- Neural Nets:
  - unit i in layer l evaluates its aggregate input based on the previous layer as $z_i^l = \sum_{j=1}^{m} f(z_j^{l-1})w_{ji}^l + w_{0i}^l$ and its activation as $f(z_i^l)$
  - common activation functions include ReLU ($f(z) = \max\{0, z\}$), tanh, and the identity function
  - backpropagation: $\delta_j^{l-1} = f'(z_j^{l-1})\sum_i w_{ji}^l \delta_i^l$

- RNN equations are given if used

- Good luck! ⌣̈