LINEAR CLASSIFIERS

Linear Machines: Regression with Mean Square

Linear Regression, Mean Square Loss:

- decision rule: $y = W'X$
- loss function: $L(W, y^i, X^i) = \frac{1}{2}(y^i - W'X^i)^2$
- gradient of loss: $\frac{\partial L(W, y^i, X^i)}{\partial W}' = -(y^i - W(t)'X^i)X^i$
- update rule: $W(t+1) = W(t) + \eta(t)(y^i - W(t)'X^i)X^i$
- direct solution: solve linear system $[\sum_{i=1}^{P} X^i X^{i\prime}]W = \sum_{i=1}^{P} y^i X^i$

Linear Machines: Perceptron

Perceptron:

- decision rule: $y = F(W'X)$ (F is the threshold function)
- loss function: $L(W, y^i, X^i) = (F(W'X^i) - y^i)W'X^i$
- gradient of loss: $\frac{\partial L(W,y^i,X^i)}{\partial W}' = -(y^i - F(W(t)'X^i))X^i$
- update rule: $W(t+1) = W(t) + \eta(t)(y^i - F(W(t)'X^i))X^i$
- direct solution: find W such that $-y^i F(W'X^i) < 0 \quad \forall i$

Linear Machines: Logistic Regression

Logistic Regression, Negative Log-Likelihood Loss function:

- decision rule: $y = F(W'X)$, with $F(a) = \tanh(a) = \frac{1-\exp(a)}{1+\exp(a)}$ (sigmoid function).
- loss function: $L(W, y^i, X^i) = 2\log(1 + \exp(-y^i W' X^i))$
- gradient of loss: $\frac{\partial L(W,y^i,X^i)}{\partial W}' = -\left(Y^i - F(W'X)\right) X^i$
- update rule: $W(t+1) = W(t) + \eta(t)(y^i - F(W(t)'X^i))X^i$

General Gradient-Based Supervised Learning Machine

Neural Nets, and many other models:

- decision rule: $y = F(W, X)$, where F is some function, and W some parameter vector.
- loss function: $L(W, y^i, X^i) = D(y^i, F(W, X^i))$, where $D(y, f)$ measures the "discrepancy" between A and B.
- gradient of loss: $\frac{\partial L(W,y^i,X^i)}{\partial W}' = \frac{\partial D(y^i,f}{\partial f}\frac{\partial F(W,X^i)}{\partial W}$
- update rule: $W(t+1) = W(t) - \eta(t)\frac{\partial D(y^i,f}{\partial f}\frac{\partial F(W,X^i)}{\partial W}$

Three Questions:

- What architecture $F(W, X)$.
- What loss Function $L(W, y^i, X^i)$.
- What optimization method.