Other Classifiers: SVMs, Neural Nets, Decision Trees, etc.
Input \( (X,y) \subseteq \mathbb{R}^d \times \{-1,+1\} \)
\( x_i \in \mathbb{R}^d \)
\( y_i \in \{-1,+1\} \)

Goal function
\( f : \mathbb{R}^d \rightarrow \{-1,+1\} \)
\( \text{s.t. } g : \mathbb{R}^d \rightarrow \mathbb{R} \)
so
\( g(x_i) = y_i \) or as many \( x_i, y_i \) as possible

\( \text{sign}(g(x_i)) = y_i \)

Classification

Supervised

Cross-Validation

Train \((X,y)\) 

Test

Goal generalize on new data
Non-linear Kernel Classifiers

\[ g(x) = \langle \omega, x \rangle = \sum_{i=1}^{\infty} \alpha_i y_i \langle x_i, x \rangle \]

\[ \omega = (\alpha_1, \alpha_2, \ldots, \alpha_n) \]

mostly \( \alpha_j = 0 \)

Non-0 \( \alpha_j = \text{support vectors} \)

\[ \alpha = (2, 0, 1, 0, 0, 2, 0, 0, 1) \]
The cost function is given by:

\[ h(x) = \sum_{i=1}^{m} \mathbb{I}(y_i \neq g(x_i)) \]

where \( \mathbb{I} \) is the indicator function.

For \( x \in \mathbb{R}^n \), but can restrict to support vectors.

Run gradient descent on

\[ h(x) \]

Find

(a) Perceptron

(b) SGD w/ hinge loss
**KNN Classifiers**

- Uses distance
- No training

\[ f(p) = \{-1, +1\} \]

= Find the k closest points \( x, x_2, ... x_k \in X \) to \( p \).

Then vote for which sign is more prevalent \( \rightarrow \{-1, +1\} \).
Activation Function

\[ \phi: \mathbb{R} \to [0,1] \text{ or } [-1,1], \text{ or } [0,\infty) \]

sigmoid \( \phi(y) = \frac{1}{1+e^{-y}} \in [0,1] \)

ReLU \( \phi(y) = \max(0, y) \in [0,\infty) \)
Parameters

$\alpha_1, \ldots, \alpha_k$ layer $1$

$\alpha_{l1}, \ldots, \alpha_{lm}$ layer $l$ parameters

$h_c(s) = \phi_i(\langle \alpha_3, x \rangle)$

Neural Network

$\Phi_e$ $m$ layers

$\Phi_m$

Sign $-$
Train (Deep) NN with GD.

Ken efficient compute gradient.

\( \approx \frac{1}{2} \times \text{parameters} \)

by layers from last backwards.

Backpropagation:

\[
g(x_i) = \omega^m \phi^m (\omega^{m-1} \phi^{m-1} (\ldots \phi'(\omega' x)(\ldots)))
\]

\( x_i \)

\( \omega_i = \frac{\partial x_i}{\partial y_i} \)

\( h_i = (y_i - \hat{y}_i)^2 \)
Decision Trees

$X \in \mathbb{R}^d$ or $Y \in \mathbb{R}_{+,0}$