SVM

① 画点，找 support vector

② （最近点且 $y_i(w^T x_i + w_0) = 1$） or 肉眼看（和 hyper plane）

- support vector $\quad y_i(w^T x_i + w_0) = 1 \quad \lambda_i \neq 0$
- non-support $\quad y_i \quad > 1 \quad \lambda_i = 0$

hyper plane : $w^T x + w_0 = 0$

$\lambda_i = 0 \rightarrow$ correct classified

$\lambda_i \neq 0 \rightarrow$
- on margin
- misclassified

K-means:

① center 1, center 2.
（class1, class2）

② 算 input x 到两 center 距离.

③ 按近的分类.

③ 分好类后，重新算 center，

RBF

① hidden ~~layer~~ unit 个数. n_H

② pick n_H 个 x 作为 C_j

③ $\rho_{max} = max\{\|C_j - C_j\|\}$

④ $\sigma_j = \dfrac{\rho_{max}}{\sqrt{2 \cdot n_H}}$

⑤ $y_1(x_i) = e^{-\frac{\|x_i - C_1\|^2}{2\sigma_1^2}}$ $\quad i = 1, 2, 3, 4$

$y_2(x_i) = e^{-}$ turn x into y domain

~~[illegible]~~

$Z = WY$

$W = ZY^{+}$

perceptron $a = (w_0, w)$

2.

- Sequential: ① $g(x) = a^t x$ $a = [w_0, w_1, w_2]$

  x just aug. [1 1 1 1 +1 +1]

  ~~② if x_k misclassified (~~

  ② assign label ($g(x) \geq 0$ .. $y_k = 1$; $g(x) < 0$ .. -1)

  ③ if $y_k \neq t_k$ (target label t_k) (misclassified)

  → update $a \leftarrow a + \eta \boxed{t_k}^{(w)} x_k$

  ④ until all samples in correct class

- sequential

  x aug + normal [1 1 1 -1 -1] $-x_1$ $-x_1$ $-x_2$ $-x_2$

  ① $g(x) = a^t x$

  ② if $g(x) < 0$ → x_k is misclassified.

  ③ → update $a \leftarrow a + \eta x_k$

- batch : aug + normal

  ① $g(x) = a^t x$

  ② if $g(x) < 0$ → x_k misclass

  sum_misclassified $= \sum x_k$

  ③ update $a \leftarrow a + \eta \sum x_k$

  epoch 2:

- batch aug :

  ① $g(x) = a^t x$

  ② assign label. ($g(x) \geq 0$ $y_k = 1$; $g(x) < 0$ $y_k = -1$)

  ③ if $y_k \neq t_k$ (misclassified)

  $\sum t_k \cdot x_k$

  ④ $a \leftarrow a + \eta \sum t_k \cdot x_k$

~~MSE~~

- multiclass perceptron:

  ① ~~j = arg max~~

  $g_i(x_k) = a_i^t x_k$ ← augment [1 1 1 1]

  ② choose max $g_i(x_k)$

  $y_k = \arg\max_i g_i$

  ③ if $y_k \neq t_k$ mis

  $a_{right} = a_{right} + \eta \cdot x_k$

  $a_{wrong} = a_{wrong} - \eta x_k$

MSE:

- Widrow-Hoff (LMS)

  $a \leftarrow a + \eta (b_k - a^t \cdot x_k) x_k$ ← aug + normal

  ← weight

  until no change

  margin

3. Linear Threshold Unit.

- Delta learning. $a = [-\theta, w_1, w_2]$

  ~~$w \leftarrow w + \eta (t_k - x_k)$~~

  seq: $w \leftarrow w + \eta [t_k - H(w x_k)] x_k^t$ ← Aug.

  target label

  $\begin{cases} 1 & >0 \\ 0 & \end{cases}$

  batch: $w \leftarrow w + \eta \sum [t_k - H(w x_k)] x_k^t$

  delta/error

Lecture 2.

→ Discriminant Function:

- { Linear Discriminant ~
- Generalised Linear ···

— Perceptron learning.

← learning driven by misclassified

← by gradient descent

← multiclass learning.

— MSE learning.

← learning driven by all data

← by gradient descend (widrow-Hoff).

Gradient descent: { batch / sequential

△ $g(x) = a^t y$ $\begin{cases} a = [w_0, w]^t \\ y = [1, \overset{(x_1,x_2)}{x}]^t \end{cases}$ boundary.

$g(x) = w_0 + w_1x_1 + w_2x_2$

△ generalised Linear Discriminant Function:

$g(x) = w_0 + w_1x_1 + w_2x_2 + \sum w_{ij} x_i x_j$

$= a^t y$ $\begin{cases} a^t = [w_0, w_1 \cdots w_{11}, w_{12} \cdots] \\ y^t = [1, x_1, x_2, x_1x_1, x_1x_2 \cdots] \end{cases}$

y_k correctly classified: $a^t y_k > 0$ & y_k labelled ω_1

$a^t y_k < 0$ & y_k labelled ω_2

sample Normalisation.

① $y \leftarrow -y \quad \forall y \in \omega_2$.

②. y_k is correctly classified $\leftrightarrow$ $a^t y_k > 0$.

margin. b

$a^t y_k > b$

- Gradient descent:

cost: $J(a)$ $\qquad \eta$: learning rate, step size.

$a \leftarrow a - \eta \nabla J(a)$

- perceptron learning ← multiple input, one output

$J_p(a) = \sum_{y \in misclass} (-a^t y)$ $\qquad -a^t y \downarrow$ 离 boundary 越远.

△ Batch perceptron

$\nabla J_p(a) = \sum_{y \in misclass} (-y)$

$a \leftarrow a - \eta \sum_{y \in mis} \omega (-y)$

△ Sequential perceptron

For each sample y_k:

if y_k misclassified:

$a \leftarrow a - \eta(-y_k)\omega_k$

sample normalisation
$a \leftarrow a - \eta \sum_{y \in mis} (-y)$
$a \leftarrow a - \eta(-y_k)$

• ▷ multiclass perceptron

η.

initialise a_c for each class

→ for each sample (y_k, w_k)

classify : $c' = \arg\max g_c(x)$

if y_k is misclassified $c' \neq \omega_k$:

$a_{\omega_k} \leftarrow a_\omega + \eta y_k$

$a_{c'} \leftarrow a_{c'} - \eta y_k$

∘ until a not change

margin b.

$Ya = b$

↓ ↓ weight

aug + normalised

MSE : All sample used.

$J_s(a) = \|Ya - b\|^2$ $\quad J_s(a) = (a^t y - b_k)^2$

• $\boxed{\text{Widrow-Hoff (LMS)}}$:

$\nabla J_s(a) = 2Y^t(Ya - b)$

$a \leftarrow a - \eta Y^t(Ya - b)$ batch

sequential; $a \leftarrow a - \eta(a^t y_k - b_k) y_k$

∘ until

K-NN : k nearest neighbour

① new sample.

② existing samples (with label)

③ compute distance between new sample & existing ones.

④ choose first k labels ,

lecture 3.

- Linear Threshold Units.
  - → Delta learning
- Competitive network
- Negative Feedback Network
- Autoencoder Network.

△ Delta learning:

- initialise w, η.
- For each sample (x_k, t_k)

  update weight.

  $$w \leftarrow w + \eta[t_k - H(w x_k)] x_k^t$$

  until all sample correctly classified.

△ Negative feed back

Activation

- Ini. y_0, w.

  $$e = x - w^T y$$

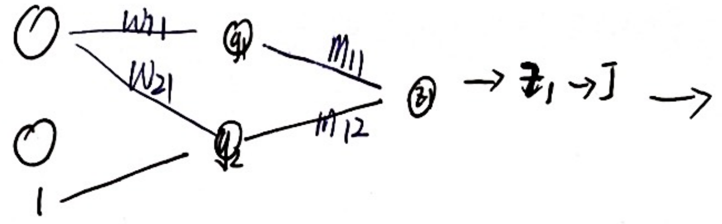
  $$y = y + \alpha W e$$

learning: $w \leftarrow w + \beta y e^T$

y. output

x. input

back propagation



$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \quad m = [m_{11} \quad m_{12}]$$

$$z = m\,(w \cdot x + w_0) + m_0$$

$$\Delta w_{20} = -\eta \frac{\partial J}{\partial w_{20}}$$

$$= -\eta \frac{\partial J}{\partial z_1} \frac{\partial z_1}{\partial net_{z_1}} \frac{\partial net_{z_1}}{\partial y_2} \frac{\partial y_2}{\partial net_{y_2}} \frac{\partial net_{y_2}}{\partial w_{20}}$$

$$w_{20} \leftarrow w_{20} + \Delta w_{20}$$