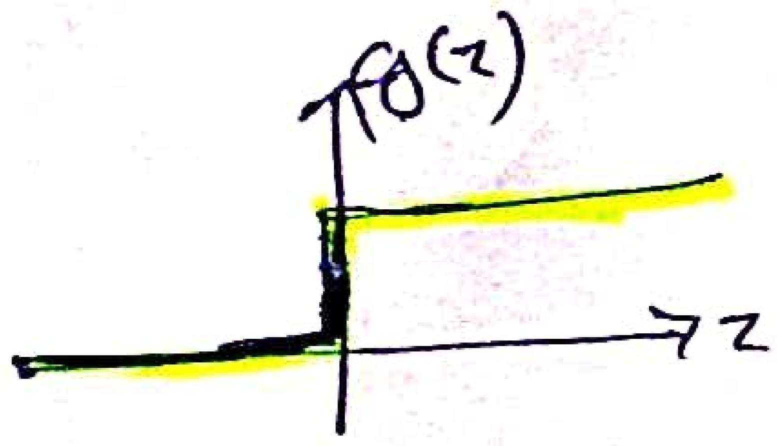
5. Kernelizing the Perceptron

⇒ Let there be a binary classification problem with

$$y \in \{0, 1\}$$

⇒ Perceptron uses hypotheses of the form:

$$h_\theta(x) = g(\theta^T x)$$



$$\text{where } g(z) = \text{Sign}(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

⇒ Here we consider stochastic gradient descent-like implementation of the perceptron algorithm.

$$\theta^{i+1} := \theta^{(i)} + \alpha\left(y^{(i+1)} - h_{\theta^{(i)}}(x^{(i+1)})\, x^{(i+1)}\right)$$

where $\theta^{(i)}$ is value of the parameters after the algorithm has seen i training examples.

$$\theta^0 = 0$$

(a) Let K be a Mercer kernel corresponding to some feature mapping ϕ.

(i) high-dimentional parameter vector can be represent as linear combination of input features $\phi(x)$

$$\theta^{(i)} = \sum_{j=1}^{m} \alpha_j^{(i)} \phi(x) \quad \left\{ \begin{array}{l} \text{This is a valid assumption} \\ \text{due to representation theorem} \end{array} \right\}$$

⇒ Here we only need to store $\alpha_1^{(i)}, \alpha_2^{(i)} \ldots \alpha_m^{(i)}$ m parameter to represent probable infinite dimentional $\theta^{(i)}$.

(ii)

$$h_{\theta^{(i)}}(x^{(i+1)}) = g\left(\theta^{(i)T}\phi(x^{(i+1)})\right)$$

$$= g\left(\left(\sum_{j=1}^{m}\alpha_j^{(i)}\phi(x^{(j)})\right)^T\phi(x^{(i+1)})\right)$$

$$= g\left(\sum_{j=1}^{m}\alpha_j^{(i)}\phi(x^{(j)})^T\phi(x^{(i+1)})\right)$$

$\Rightarrow$ Let $\tilde{K}$ be the kernel matrix corresponding kernel K and data set $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$

$$\tilde{K} \in \mathbb{R}^{m\times m}$$

$$\boxed{h_{\theta^{(i)}}(x^{(i+1)}) = g\left(\underbrace{\tilde{K}(:, i+1)}_{\{i+1^{st}\text{ Column of Kernel matrix}\}}{}^T\alpha^{(i)}\right)}$$

(iii)

$$\underbrace{\theta^{(i+1)}}_{\sum_{j=1}^{m}\alpha_j^{(i+1)}\phi(x^{(j)})} := \underbrace{\theta^{(i)}}_{\sum_{j=1}^{m}\alpha_j^{(i)}\phi(x^{(j)})} + \alpha\left(y^{(i+1)} - h_{\theta^{(i)}}(x^{(i+1)})\right)x^{(i+1)}$$

$\Rightarrow$ Let $X = \begin{bmatrix}\phi(x^{(1)T})\\ \phi(x^{(2)T})\\ \vdots \\ \phi(x^{(m)T})\end{bmatrix}$ be the design matrix.

⇒ Multiplying both side of the equation with design matrix.

$$\underbrace{\sum_{j=1}^{m} \alpha_j^{(i+1)} \times \underbrace{\phi(x^{(j)})}_{\tilde{K}(:,j)}}_{\tilde{K}\alpha^{(i+1)}} = \sum_{j=1}^{m} \alpha_j^{(i)} \times \phi(x^{(j)})$$

$$+ \alpha\left(y^{(i+1)} - g(\tilde{K}(:,i+1)^T \alpha^{(i)}\right) \times \phi(x^{(i+1)})$$

$$\Rightarrow \tilde{K}\alpha^{(i+1)} = \tilde{K}\alpha^{(i)} + \alpha\left(y^{(i+1)} - g(\tilde{K}(:,i+1)^T \alpha^{(i)})\right)\tilde{K}(:,i+1)$$

$$\boxed{\alpha^{(i+1)} = \alpha^{(i)} + \alpha\left(y^{(i+1)} - g(\tilde{K}(:,i+1)^T \alpha^{(i)})\right)\tilde{K}^{-1}\tilde{K}(:,i+1)}$$