

Radial Basis Function Networks

비록 아주 일반적인 정의라고 보기는 어렵지만, RBFN를 Gaussian kernel을 사용한 neural networks라고 보았을 때, RBFN는 다음과 같이 정의된다.

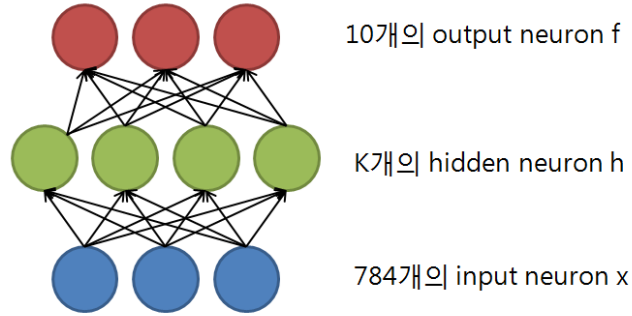


그림 1. RBFN의 확률 그래프 모델

$$x_i \in \mathbb{R}^{784}, h_i \in \mathbb{R}^K, f_i \in \mathbb{R}^{10}$$

$$f_i = (f_{i0}, \dots, f_{i9}), y_i \in \mathbb{R}^{10}$$

$$c_k \in \mathbb{R}^K, \Sigma_k \in \mathbb{R}^{K \times K}$$

$$h_i \propto N(x_i; c_k, \Sigma_k)$$

$$h_i \propto \exp((x_i - c_k)^T \Sigma_k^{-1} (x_i - c_k))$$

$$w_j \in \mathbb{R}^k, w_{j0} \in \mathbb{R}$$

$$f_{ij} = w_j^T h_i + w_{j0}$$

이 때, y_i 에 대하여 학습 데이터 1번의 class가 9인 경우, $y_1 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$ 로 표현되는 것을 가정했다. f_i 는 y_i 를 예측하는 regression된 실수 값이다. 이 때, K는 Gaussian kernel의 개수이다. 이 K는 당연하게도 꼭 10일 필요는 없다. 이 때, k-means로 Gaussian kernel K개를 만들 수 있다. 이 경우, 각 cluster k에 할당되는 instance $\{x^{(k)}\}$ 를 구한 뒤, c_k, Σ_k 를 다음과 같이 구하게 된다.

$$\mu_k = \frac{1}{N_k} \sum_{x_i \in \{x^{(k)}\}} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_{x_i \in \{x^{(k)}\}} (x_i - \mu_k)(x_i - \mu_k)^T$$

w를 얻기 위하여 linear regression을 취하는 방법을 검토해 볼 수 있다. 이 때 RBFN은 Gaussian kernel을 통하여 얻어진 x를 통하여 얻어진 h를 linear regression하여 f를 얻는 방법으로

생각된 것이다. W 를 계산하기 위하여, 전체 학습 데이터 X 를 통하여 얻어진 H 와 학습 데이터의 클래스 Y 에 대해 다음 식을 사용할 수 있다.

$$X \in \mathbb{R}^{N \times 784}, H \in \mathbb{R}^{N \times K}, Y \in \mathbb{R}^{N \times 10}, W \in \mathbb{R}^{K \times 10}$$

$$W = (H^T H)^{-1} H^T Y$$

이 때, $Y = (y_1, y_2, \dots, y_N)^T$ 의 y_i 에 대하여 학습 데이터 1번의 class가 9인 경우, $y_1 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$ 로 표현되는 것을 가정했다. 위 계산의 확장으로 w_j 뿐 아니라 w_{j0} 을 계산할 수 있을 것이다.

그러나 위와 같은 방법으로는 좋은 성능이 나오지 않는 것을 실험을 통해 확인할 수 있을 것이다 (왜 그럴까? 역으로 생각하면, 왜 RBFN이 좋은 성능을 가질 것이란 기대는 좌절되었는가?). 이 때, Gaussian kernel을 정의하는 다른 방식이나, W 를 구할 수 있는 다른 방식을 검토해 볼 수 있을 것이다.

예컨대, Gaussian classifier가 일반적인 RBFN 구현보다 성능이 더 좋음을 확인할 수 있다. 이 경우, K 는 10이며, 특별히 다음 특성을 가진다.

$$\mu_k = \frac{1}{N_k} \sum_{y_{ik}=1} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_{y_{ik}=1} (x_i - \mu_k)(x_i - \mu_k)^T$$

$$h_i = N(x_i; c_k, \Sigma_k)$$

$$h_i = \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} \exp(-\frac{1}{2}(x - c_k)^T \Sigma_k^{-1} (x - c_k))$$

$$W_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}$$

Gaussian classifier보다 더 좋은 성능을 가지는 RBFN을 생각할 수 있을까?