Deep Learning Book

6.1 Deep Feedforward Networks

October 9, 2016

Feedforward Neural Networks

- 딥러닝의 전형적 모델. "multilayer perceptrons(MLPs)"라고도 함. 분류기를 예로 들면,
  - 어떠한 함수 $f^*(\boldsymbol{x})$가 있어서 입력 $\boldsymbol{x}$에 대해서 카테고리 y로 매핑을 해준다고 하면,
  - feedforward networks는 $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\theta})$로 매핑을 근사시켜주는 파라미터 $\boldsymbol{\theta}$를 학습하는 것이 목표임.
- feedback 연결은 없다(cf. RNN(recurrent neural networks)).
- 객체인식에 쓰이는 CNN(convolutional neural networks)도 특수한 종류의 feedforward networks이다.
- NLP에 많이 활용되는 RNN의 개념적인 초석.
- $f(\boldsymbol{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\boldsymbol{x})))$ 형식의, 연결 체인으로 된 "네트웍"이다. 이때 체인의 전체 길이를 모델의 "깊이(depth)"라 한다.
- 마지막 층을 "출력층(output layer)"이라 함(위의 예에서, $f^{(3)}$).
- 최종 출력층에서만 실제 관측치인 y에 맞추면 되고, 중간층들에서는 어떤 출력을 내야 하는지 입력값 $\boldsymbol{x}$만으로는 결정할 수 없다. 이 중간층을 "은닉층(hidden layer)"이라 한다.
- 은닉층은 대개 벡터 값을 다룬다. 이 벡터의 차원이 모델의 "너비(width)"를 결정한다.

Feedforward Neural Networks (비선형성)

- 비선형성을 반영하기 위한 딥러닝 진영의 전략은 다음과 같은 모델을 상정한다:

$$y = f(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{w}) = \phi(\boldsymbol{x}; \boldsymbol{\theta})^\top \boldsymbol{w} \tag{1}$$

  - "표현"은 $\phi(\boldsymbol{x}; \boldsymbol{\theta})$로 파라미터화시키고,
  - 최적화 알고리즘을 통해 $\boldsymbol{\theta}$를 찾아낸다 (은닉층 학습...).
  - $\boldsymbol{w}$는 $\phi(\boldsymbol{x})$를 최종 출력으로 연결시킨다.
- 학습은 선형모델과 비슷한 방식으로 진행한다:
  - 옵티마이저 선택, 비용 함수 선정, 출력 형태 결정.
- 은닉층의 도입으로 "활성화 함수(activation function)" 개념이 필요하다.
- 얼마나 많은 층을 쌓을 것인지, 층간의 연결을 어떻게 할 것인지, 각 층에는 얼마나 많은 유닛들이 있어야 하는지 등등도 결정해야 한다.
- 딥러닝 학습에서는 복잡한 함수들의 기울기벡터(gradients)를 구해야 하는데, 이를 효율적으로 해줄 수 있는 "역전파 (back-propagation)" 알고리즘에 대해서도 살펴볼 것이다.

6.1 Example: Learning XOR

- 훈련셋:

$$\mathbb{X} = \{[0,0]^\top, [0,1]^\top, [1,0]^\top, [1,1]^\top\}$$

- 비용 함수: Mean Squared Error(MSE). 이산형태 데이터에는 별로 안 맞지만 계산의 단순성을 위해서...

$$J(\boldsymbol{\theta}) = \frac{1}{4}\sum_{\boldsymbol{x}\in\mathbb{X}}(f^*(\boldsymbol{x}) - f(\boldsymbol{x};\boldsymbol{\theta}))^2 \tag{2}$$

- 선형 모델로 가정해 보면,

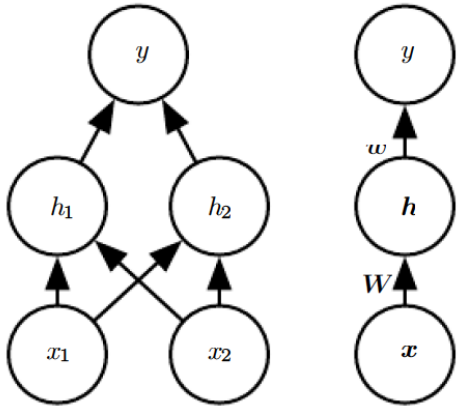
$$f(\boldsymbol{x};\boldsymbol{w},b) = \boldsymbol{x}^\top\boldsymbol{w} + b \tag{3}$$

$$\Rightarrow \boldsymbol{w} = \mathbf{0} \quad \text{and} \quad b = \frac{1}{2} \qquad \text{(FAIL!!)}$$

- 그래서... 다른 방식이 필요함!

6.1 Example: Learning XOR (Cont.)

- 2-유닛 은닉층이 가미된 단순 feedforward network.



- 은닉층 유닛들은 $\boldsymbol{h}$로 표기된다. $f^{(1)}(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c})$의 계산결과이며 두번째 층(=출력층)의 입력값으로 쓰인다.
- 출력층은 선형모델이지만, $\boldsymbol{x}$가 아닌 $\boldsymbol{h}$를 입력으로 받는다. 최종적으로 네트웍을 구성하는 함수는 다음과 같다:

$$\boldsymbol{h} = f^{(1)}(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}) \tag{4}$$

$$y = f^{(2)}(\boldsymbol{h}; \boldsymbol{w}, b) \tag{5}$$

$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = f^{(2)}(f^{(1)}(\boldsymbol{x})) \tag{6}$$

6.1 Example: Learning XOR (Cont.)

- $f^{(1)}$은 어떤 모양이어야 하는가?
  - $f^{(1)}$이 선형이면 전체 모델도 선형화된다.
    $\Rightarrow$ 비선형성을 반영시켜줘야 한다.
  - 아래와 같은 변환함수를 생각하자:

$$\boldsymbol{h} = g(\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}) \tag{7}$$

$$h_i = g(\boldsymbol{x}^\top \boldsymbol{W}_{:,i} + c_i) \tag{8}$$

$$g(z) = \max\{0, z\} \quad \Rightarrow \quad (\text{ReLU}) \tag{9}$$

- 최종 네트웍 수식:

$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b \tag{10}$$

6.1 Example: Learning XOR (Cont.)

- 한 가지 해는 다음과 같다:

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \quad b = 0 \tag{11}$$

- XOR 입력 디자인 행렬 $\boldsymbol{X}$를 가정하고, $\boldsymbol{XW}$를 계산하면:

$$\boldsymbol{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}, \boldsymbol{XW} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix}, ((\boldsymbol{XW})^\top + \boldsymbol{c})^\top = \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}. \tag{12}$$

- 위의 결과에 ReLU g를 적용해서 $(\boldsymbol{h})$를 얻고, 이것에 가중치$\boldsymbol{w}$를 적용해서 최종 결과를 얻는다:

$$\boldsymbol{h} = g(\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}, \boldsymbol{w}^\top \boldsymbol{h} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}. \tag{13}$$