ML Notes

Simon Hulse

April 20, 2022

1 Feed-forward neural networks

An N-layer Neural Network (NN) comprises a vector of activations $\boldsymbol{a}^{(n)}\ \forall n \in \{1, \dots, N\}$ for each layer, where $\boldsymbol{a}^{(1)} \equiv \boldsymbol{x}$ is the vector of inputs, and $\boldsymbol{a}^{(N)}$ is the output of the network. Generating the activations for the next layer is achieved by:

$$\begin{gathered} \boldsymbol{a}^{(n+1)} = f\underbrace{\left(\boldsymbol{W}^{(n+1)}\boldsymbol{a}^{(n)} + \boldsymbol{b}^{(n+1)}\right)}_{\boldsymbol{z}^{(n+1)}},\ \forall n \in \{1, \dots, N-1\}. \\ \implies \boldsymbol{a}^{(N)} = f\left(\boldsymbol{W}^{(N)} f\left(\boldsymbol{W}^{(N-1)} f\left(\cdots f\left(\boldsymbol{W}^{(2)}\boldsymbol{a}^{(1)} + \boldsymbol{b}^{(2)}\right) + \cdots\right) + \boldsymbol{b}^{(N-1)}\right) + \boldsymbol{b}^{(N)}\right) \end{gathered} \tag{1}$$

Given $\boldsymbol{a}^{(n)} \in \mathbb{R}^i$ and $\boldsymbol{a}^{(n+1)} \in \mathbb{R}^j$, $\boldsymbol{W}^{(n+1)} \in \mathbb{R}^{j\times i}$ is a matrix of *weights* and $\boldsymbol{b}^{(n+1)} \in \mathbb{R}^j$ is a vector of *biases*. A non-linear *activation function* $f : \mathbb{R}^j \to \mathbb{R}^j$ is subsequently applied to the affine operation on $\boldsymbol{a}^{(n)}$ to generate the next layer.

1.1 Feed-forward algorithm

1. Input $\boldsymbol{x}$. Set $\boldsymbol{a}^{(1)} \to \boldsymbol{x}$
2. For n in $\{1, 2, \dots, N-1\}$:

$$\begin{aligned} \boldsymbol{z}^{(n+1)} &= \boldsymbol{W}^{(n+1)}\boldsymbol{a}^{(n)} + \boldsymbol{b}^{(n+1)} \\ \boldsymbol{a}^{(n+1)} &= f\left(\boldsymbol{z}^{(n+1)}\right) \end{aligned}$$

3. Output $\boldsymbol{a}^{(N)}$

N.B. The pre-activations $\boldsymbol{z}^{(n+1)}$ will be needed for backpropagation, hence their explicit computation.

1.2 Training: Backpropagation