Introduction to Neural Networks

DATA 607 — Session 6 — 11/03/2019

What is a neural network?

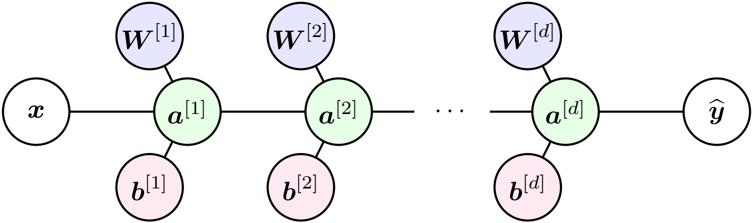
A neural network consists of **neurons** or **units**. Neurons have **inputs**, **outputs**, and **activations**. Connections between these neurons have **weights**. Neurons are organized into **layers**. **Hidden layers** are sandwiched between an **input layer** and an **output layer**.

Linear regression, logistic regression, and perceptron classification are all neural networks, degenerate in the sense that they have no hidden layers.

More formally, a neural network is a function

$$N : \mathbb{R}^p \longrightarrow \mathbb{R}^q$$

constructed in a particular way.



depth of network (number of layers):	d
number of neurons in layer ℓ:	p_ℓ
activation (output) of neuron k in layer ℓ:	$a_k^{[\ell]}$
bias of neuron k in layer ℓ:	$b_k^{[\ell]}$
weight of the connection between neuron j in layer ℓ and neuron i in layer $\ell+1$:	$w_{ij}^{[\ell]}$
activation function in layer ℓ:	h

$$a_i^{[\ell+1]} = h\left(z_i^{[\ell+1]}\right), \quad \text{where} \quad z_i^{[\ell+1]} = b_i^{[\ell]} + \sum_{j=1}^{p_\ell} w_{ij}^{[\ell]} a_j^{[\ell]}$$

Vectorize:

$$\boldsymbol{z}^{[\ell]} = \begin{bmatrix} z_1^{[\ell]} \\ \vdots \\ z_{p_\ell}^{[\ell]} \end{bmatrix} \in \mathbb{R}^{p_\ell}, \quad \boldsymbol{a}^{[\ell]} = \begin{bmatrix} a_1^{[\ell]} \\ \vdots \\ a_{p_\ell}^{[\ell]} \end{bmatrix} \in \mathbb{R}^{p_\ell},$$

$$\boldsymbol{b}^{[\ell]} = \begin{bmatrix} b_1^{[\ell]} \\ \vdots \\ b_{p_\ell}^{[\ell]} \end{bmatrix} \in \mathbb{R}^{p_\ell}, \quad \boldsymbol{W}^{[\ell]} = \begin{bmatrix} w_{11}^{[\ell]} & \cdots & w_{1p_\ell}^{[\ell]} \\ \vdots & \ddots & \vdots \\ w_{p_{\ell+1}1}^{[\ell]} & \cdots & w_{1p_\ell}^{[\ell]} \end{bmatrix} \in \mathbb{R}^{p_{\ell+1} \times p_\ell}$$

Apply h componentwise:

$$\boldsymbol{a}_i^{[\ell+1]} = h\left(\boldsymbol{z}^{[\ell+1]}\right) = h\left(\boldsymbol{b}^{[\ell]} + \boldsymbol{W}^{[\ell]}\boldsymbol{a}^{[\ell]}\right)$$

Useful intermediate quantity:

$$\boldsymbol{b}^{[\ell]} + \boldsymbol{W}^{[\ell]}\boldsymbol{a}^{[\ell]}$$

The process of computing $\widehat{\boldsymbol{y}}$ from $\boldsymbol{x}$, given $\boldsymbol{b}^{[\ell]}$ and $\boldsymbol{W}^{[\ell]}$, $\ell = 1, \ldots, d$, is called **forward propagation** of data.

A **loss function**, $L(\widehat{y}, y)$, asseses a penalty based on the error in approximating $\boldsymbol{y}$ by $\widehat{\boldsymbol{y}}$.

The total loss associated to a training set

$$T = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$$

is called the **cost** of T:

$$C(T) = \sum_{i=1}^{n} L(\widehat{\boldsymbol{y}}_i, \boldsymbol{y}_i)$$

We adjust the variables $\boldsymbol{b}^{[\ell]}$ and $\boldsymbol{W}^{[\ell]}$ based on the derivatives

$$\frac{\partial C}{\partial b_i^{[\ell]}} \quad \text{and} \quad \frac{\partial C}{\partial w_{ij}^{[\ell]}}$$