Lesson 07
Notes on Chapter 4

CSC357 Advanced Topics—Machine Learning

23 January 2020

- so far, looked at ML models like black boxes
- can do a lot without knowing much about how models work!
  - predict price of housing
  - recognize a hand-drawn numeral
  - distinguish spam from other e-mail
- more knowledge will help us choose. . .
  - right model
  - right training algorithm
  - right values for hyperparameters (tune model)
- more knowledge will help us. . .
  - debug
  - understand / analyze errors
- need to know more to build and train neural networks
- start with Linear Regression (one of simplest models)
- 2 ways to train Linear Regression models
  - "closed form" — directly compute coefficients
  - iteratively — successively better approximations (Gradient Descent)
- both methods should yield same results
- training means finding values for model parameters that minimize the cost function over the training set
- 3 variants of Gradient Descent now and again in later chapters when we study neural networks

  - Batch GD
  - Mini-batch GD
  - Stochastic GD

- Polynomial Regression
  - works when relationships are non-linear
  - more complex than Linear Regression
  - more parameters
  - more prone to overfitting
    * use learning curves to detect overfitting
    * use regularization techniques to reduce risks of overfitting

- 2 more models for classification
  - Logistic Regression
  - Softmax Regression

- what kind of math do we need?
  - vectors and matrices
    * products
    * transposes
    * inverses
  - calculus
    * derivatives (rates of change of a function of one variable)
    * partial derivatives (rates of change of a function of several variables)

- dot product of 2 vectors

$$\begin{aligned}
\vec{u} &= (u_0, u_1, u_2, \ldots, u_n) \\
\vec{v} &= (v_0, v_1, v_2, \ldots, v_n) \\
\vec{u} \cdot \vec{v} &= u_0 v_0 + u_1 v_1 + u_2 v_2 + \ldots + u_n v_n \\
\vec{u} \cdot \vec{v} &= \vec{v} \cdot \vec{u} \quad \text{(commutative operation)}
\end{aligned}$$

- column vectors, transposes, and dot products

$$
\begin{aligned}
\vec{u} &= \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} \\
\vec{v} &= \begin{bmatrix} 6 \\ 7 \\ 8 \end{bmatrix} \\
\vec{u}^T \vec{v} &= \begin{bmatrix} 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 6 \\ 7 \\ 8 \end{bmatrix} \\
&= 2 \cdot 6 + 3 \cdot 7 + 4 \cdot 8 \\
&= 65
\end{aligned}
$$

- if $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are $n \times n$ matrices and $\mathbf{AB} = \mathbf{C}$ then ...
  - each element of $\mathbf{C}$ is the dot product of a row in $\mathbf{A}$ with a column in $\mathbf{B}$
  - let $c_{i,j}$ be the element in the i^{th} row and the j^{th} column in $\mathbf{C}$
  - let $a_{i,k}$ be the element in the i^{th} row and the k^{th} column in $\mathbf{A}$
  - let $b_{k,j}$ be the element in the k^{th} row and the j^{th} column in $\mathbf{B}$
  - then...

$$
c_{i,j} = a_{i,0} b_{0,j} + a_{i,1} b_{1,j} + \ldots + a_{i,n-1} b_{n-1,j}
$$

- transpose of a matrix
  - rows become columns
  - columns become rows
  - elements reflected across line drawn from upper left corner to lower right corner of matrix

$$
\begin{bmatrix} a & b \\ c & d \end{bmatrix}^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix}
$$

- identity matrix $\mathbf{I}$
  - all elements of $\mathbf{I}$ are zeroes except for those on the main diagonal (row index = column index)
  - all elements on main diagonal are ones
  - for any matrix $\mathbf{A}$... $\mathbf{AI} = \mathbf{IA} = \mathbf{A}$

- some (but not all) matrices have inverses

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

- a simple regression model of life satisfaction—

$$life\_satisfaction = \theta_0 + \theta_1 \cdot GDP\_per\_capita$$

- a linear function of the input feature GDP_per_capita
- θ_0 and θ_1 are models parameters
- linear model makes a prediction by computing weighted sum of the input features, plus a constant
- constant is the bias term (also called the intercept term)

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- In this equation:
  - $\hat{y}$ is the predicted value.
  - n is the number of features.
  - x_i is the i^{th} feature value.
  - θ_j is the j^{th} model parameter (including the bias term θ_0 and the feature weights $\theta_1, \theta_2, \cdots, \theta_n$).
- more concisely written using a vectorized form

$$\begin{aligned} \hat{y} &= h_\theta(\mathbf{x}) \\ &= \theta \cdot \mathbf{x} \end{aligned}$$

- In this equation:
  - θ is the models *parameter vector*, containing the bias term θ_0 and the feature weights θ_1 to θ_n.
  - $\mathbf{x}$ is the instances *feature vector*, containing x_0 to x_n, with x_0 always equal to 1.
  - $\theta \cdot \mathbf{x}$ is the dot product of the vectors θ and $\mathbf{x}$, which is of course equal to $\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$
  - h_θ is the hypothesis function, using the model parameters θ.