Deep Learning: An Introduction for Applied Mathematicians (Part 1)

Gobind Puniani

MATH 190: Fall 2019

08/29/19

Artificial Neural Networks

- Purpose: Introduce deep learning, which has become popular in recent years
- In artificial neural networks (ANNs), each “neuron” delivers binary output by applying a nonlinear function to input data
    - Sigmoid function is commonly used
    - The function is scaled and translated through “weights” and “biases”
    - Each neuron uses its own weights and biases on real numbers outputted from each neuron in the previous layer
- Many “hidden layers” lie between the input layer and the output layer
    - Characteristic of deep learning

Stochastic Gradient and Back Propagation

- “Steepest descent” / “gradient descent”: iterative process that attempts to converge at vector that minimizes cost function
    - Each layer is represented by a vector
    - Cost function determines the category based on components of final vector
    - Goal: minimize cost function
- “Gradient”: the vector of partial derivatives
- “Learning rate”: the step-size with each iteration
- “Stochastic Gradient”: To save computational time, the gradient of random training point is used in lieu of the mean of all gradients
    - A small sample mean could also be used to preserve accuracy
- “Back propagation”: Computing output layers in a forward pass and then computing the errors in a backward pass

Important Equations

- Sigmoid function: $\frac{1}{1+e^{-x}}$
- Vector of Outputs: $\sigma(Wa+b)$, where W is the matrix with the weights and b is the vector of biases
  - Column: previous layer neurons
  - Row: current layer neurons
- General cost function: Cost = $\frac{1}{N}\Sigma_{i=1}^{N}\frac{1}{2}\|y(x^{(i)})-a^{[L]}(x^{(i)})\|_2^2$
- Updating a vector in gradient descent: $p \to p-\eta\nabla\text{Cost}(p)$

Questions

- What is a “neuron” in the context of computing?
- How many neurons should be in a given layer, and how many layers should there be?