COMP 4432 Machine Learning

Lesson 4: Model Training

Agenda

- Training Models
  - Theory and Implementation
  - Requires an understanding an algorithm’s capabilities and hyperparameters
  - When somethings breaks, know and recognize what is at fault

The Process

Objective

Gathered & Prepared Data

Identified Potential Algorithms

Model Training and Selection

Production

The Process

Objective

Gathered & Prepared Data

Identified Potential Algorithms

Model Training and Selection

Production

Training Models

- Easy to execute
- Easy to mishandle
  - Generalizability
- More to it than just "*.fit()*"

Training Models

- Easy to execute
- Easy to mishandle
  - Generalizability
- More to it than just “*.fit()*”
- New tools available to “simplify” training
  - AutoML (Automated Machine Learning)
    - FLAML, AutoGluon, AutoPilot
  - Hyperparameter Tuning Optimization
    - (HyperOpt, OpTuna)

Training Models

- When "*.fit()*" is called, what is happening?
- Finding optimum model parameters by minimizing a cost function

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (h_\theta(x_i) - y_i)^2$$

$$h_\theta(x^{(i)}) = \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \ldots + \theta_f x_f^{(i)} h_\theta(x^{(i)}) = \theta^T x^{(i)}$$

$$J(\theta) = -\frac{1}{n} \sum_{i}^{n} y_i \; log\left(p(x_i)\right) + (1 - y_i) \; log(1 - p(x_i))$$

$$p = \frac{1}{1 + e^{\vec{\beta} \cdot \vec{x}}}$$

$$J(k, t_k) = \frac{m_{\text{left}}}{m} \text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m} \text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} (\hat{y}_{\text{node}} - y^{(i)})^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

Training Models

- When “*.fit()*” is called, what is happening?
- Finding optimum model parameters by minimizing a cost function
  - Theta (Beta) vector in linear (logistic) regression
  - Features and splits in a decision tree

Training Models

- When "*.fit()*" is called, what is happening?
- Finding optimum model parameters by minimizing a cost function
  - Theta (Beta) vector in linear (logistic) regression
  - Features and splits in a decision tree
- Dependent on user-defined hyperparameters
  - Linear Regression
  - Random Forest Regression

Linear Regression (Closed Form)

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

$$h_\theta(x^{(i)}) = \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \ldots + \theta_f x_f^{(i)} h_\theta(x^{(i)}) = \theta^T x^{(i)}$$

Linear Regression (Closed Form)

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

$$h_\theta(x^{(i)}) = \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \ldots + \theta_f x_f^{(i)} h_\theta(x^{(i)}) = \theta^T x^{(i)}$$

$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_f \end{pmatrix} \qquad x^{(i)} = \begin{pmatrix} x_0^{(i)} \\ x_1^{(i)} \\ \vdots \\ x_f^{(i)} \end{pmatrix}$$

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)} \right)^2$$

Linear Regression (Matrix Form)

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\theta^T x^{(i)} - y^{(i)}\right)^2$$

$$X = \begin{pmatrix} x^{(1)T} \\ x^{(2)T} \\ \vdots \\ x^{(n)T} \end{pmatrix} = \begin{pmatrix} x_0^{(1)} & x_1^{(1)} & \dots & x_f^{(1)} \\ x_0^{(2)} & x_1^{(2)} & \dots & x_f^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{(n)} & x_1^{(n)} & \dots & x_f^{(n)} \end{pmatrix} \qquad \theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_f \end{pmatrix}$$

Linear Regression (Matrix Form)

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\theta^T x^{(i)} - y^{(i)}\right)^2$$

$$X = \begin{pmatrix} x^{(1)T} \\ x^{(2)T} \\ \vdots \\ x^{(n)T} \end{pmatrix} = \begin{pmatrix} x_0^{(1)} & x_1^{(1)} & \dots & x_f^{(1)} \\ x_0^{(2)} & x_1^{(2)} & \dots & x_f^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{(n)} & x_1^{(n)} & \dots & x_f^{(n)} \end{pmatrix} \qquad \theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_f \end{pmatrix}$$

$$X\theta = \begin{pmatrix} \theta_0 x_0^{(1)} & + & \theta_1 x_1^{(1)} & + & \dots & + & \theta_f x_f^{(1)} \\ \theta_0 x_0^{(2)} & + & \theta_1 x_1^{(2)} & + & \dots & + & \theta_f x_f^{(2)} \\ \vdots & & \vdots & & \ddots & & \vdots \\ \theta_0 x_0^{(n)} & + & \theta_1 x_1^{(n)} & + & \dots & + & \theta_f x_f^{(n)} \end{pmatrix}$$

Linear Regression (Matrix Form)

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)} \right)^2$$

$$X = \begin{pmatrix} x^{(1)T} \\ x^{(2)T} \\ \vdots \\ x^{(n)T} \end{pmatrix} = \begin{pmatrix} x_0^{(1)} & x_1^{(1)} & \dots & x_f^{(1)} \\ x_0^{(2)} & x_1^{(2)} & \dots & x_f^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{(n)} & x_1^{(n)} & \dots & x_f^{(n)} \end{pmatrix} \qquad \theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_f \end{pmatrix}$$

$$X\theta = \begin{pmatrix} \theta_0 x_0^{(1)} & + & \theta_1 x_1^{(1)} & + & \dots & + & \theta_f x_f^{(1)} \\ \theta_0 x_0^{(2)} & + & \theta_1 x_1^{(2)} & + & \dots & + & \theta_f x_f^{(2)} \\ \vdots & & \vdots & & \ddots & & \vdots \\ \theta_0 x_0^{(n)} & + & \theta_1 x_1^{(n)} & + & \dots & + & \theta_f x_f^{(n)} \end{pmatrix}$$

$$J(\theta) = \frac{1}{n} (X\theta - y)^T (X\theta - y)$$

Linear Regression (Matrix Form)

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)} \right)^2$$

$$J(\theta) = \frac{1}{n} (X\theta - y)^T (X\theta - y)$$

$$J(\theta) = \theta^T X^T X\theta - 2(X\theta)^T y + y^T y$$

Linear Regression (Matrix Form)

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)} \right)^2$$

$$J(\theta) = \frac{1}{n} (X\theta - y)^T (X\theta - y)$$

$$J(\theta) = \theta^T X^T X \theta - 2(X\theta)^T y + y^T y$$

$$\frac{\partial J}{\partial \theta} = 2X^T X \theta - 2X^T y = 0$$

$$X^T X \theta = X^T y$$

$$\theta = (X^T X)^{-1} X^T y$$

Linear Regression

- Closed form solution is computationally demanding as the number of rows increases
  - Specifically, the inverse

$$\theta = (X^T X)^{-1} X^T y$$

- Gradient Descent offers a solution to large n

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{2}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)} \right) x_j^{(i)}$$

Gradient Descent

- Start with random values in θ
- Identify the direction that offers the largest reduction in error with respect to the parameters we are trying to find θ
- Update the values of θ and try again

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)} \right)^2$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{2}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)} \right) x_j^{(i)}$$

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

Gradient Descent

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)} \right)^2$$

$$\begin{aligned}
\theta^T x^{(i)} &= \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \ldots + \theta_f x_f^{(i)} \\
\theta^T x^{(i)} - y^{(i)} &= \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \ldots + \theta_f x_f^{(i)} - y^{(i)}
\end{aligned}$$

Gradient Descent

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\theta^T x^{(i)} - y^{(i)}\right)^2$$

$$\begin{aligned}
\theta^T x^{(i)} &= \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \ldots + \theta_f x_f^{(i)} \\
\theta^T x^{(i)} - y^{(i)} &= \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \ldots + \theta_f x_f^{(i)} - y^{(i)}
\end{aligned}$$

$$\frac{\partial\left(\theta^T x^{(i)} - y^{(i)}\right)^2}{\partial\theta_j} = \frac{\partial\left(\theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \ldots + \theta_f x_f^{(i)} - y^{(i)}\right)^2}{\partial\theta_j}$$

Gradient Descent

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\theta^T x^{(i)} - y^{(i)}\right)^2$$

$$\begin{aligned}
\theta^T x^{(i)} &= \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \ldots + \theta_f x_f^{(i)} \\
\theta^T x^{(i)} - y^{(i)} &= \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \ldots + \theta_f x_f^{(i)} - y^{(i)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial\left(\theta^T x^{(i)} - y^{(i)}\right)^2}{\partial\theta_j} &= \frac{\partial\left(\theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \ldots + \theta_f x_f^{(i)} - y^{(i)}\right)^2}{\partial\theta_j} \\
&= 2\left(\theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \ldots + \theta_f x_f^{(i)} - y^{(i)}\right) x_j^{(i)}
\end{aligned}$$

$$\frac{\partial J(\theta)}{\partial\theta_j} = \frac{2}{n}\sum_{i=1}^{n}\left(\theta^T x^{(i)} - y^{(i)}\right) x_j^{(i)}$$

Gradient Descent

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{2}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)}\right) x_j^{(i)}$$

$$\frac{\partial J(\theta)}{\partial \theta_{j=j'}} = \begin{array}{l} (\theta_0 x_0^{(1)} + \theta_1 x_1^{(1)} + \ldots + \theta_f x_f^{(1)} - y^{(1)}) x_{j'}^{(1)} \\ + \\ (\theta_0 x_0^{(2)} + \theta_1 x_1^{(2)} + \ldots + \theta_f x_f^{(2)} - y^{(2)}) x_{j'}^{(2)} \\ + \\ \ldots \\ + \\ (\theta_0 x_0^{(n)} + \theta_1 x_1^{(n)} + \ldots + \theta_f x_f^{(n)} - y^{(n)}) x_{j'}^{(n)} \end{array}$$

Gradient Descent

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{n} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

$$\frac{\partial J(\theta)}{\partial \theta_{j=j'}} = \begin{array}{l} (\theta_0 x_0^{(1)} + \theta_1 x_1^{(1)} + \ldots + \theta_f x_f^{(1)} - y^{(1)}) x_{j'}^{(1)} \\ + \\ (\theta_0 x_0^{(2)} + \theta_1 x_1^{(2)} + \ldots + \theta_f x_f^{(2)} - y^{(2)}) x_{j'}^{(2)} \\ + \\ \ldots \\ + \\ (\theta_0 x_0^{(n)} + \theta_1 x_1^{(n)} + \ldots + \theta_f x_f^{(n)} - y^{(n)}) x_{j'}^{(n)} \end{array}$$

$$X^T = \begin{pmatrix} x_0^{(1)} & x_0^{(2)} & \ldots & x_0^{(n)} \\ x_1^{(1)} & x_1^{(2)} & \ldots & x_1^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ x_f^{(1)} & x_f^{(2)} & \ldots & x_f^{(n)} \end{pmatrix} \qquad X\theta - y = \begin{pmatrix} \theta_0 x_0^{(1)} & + & \theta_1 x_1^{(1)} & + & \ldots & + & \theta_f x_f^{(1)} & - & y^{(1)} \\ \theta_0 x_0^{(2)} & + & \theta_1 x_1^{(2)} & + & \ldots & + & \theta_f x_f^{(2)} & - & y^{(2)} \\ \vdots & & \vdots & & \ddots & & \vdots & & \vdots \\ \theta_0 x_0^{(n)} & + & \theta_1 x_1^{(n)} & + & \ldots & + & \theta_f x_f^{(n)} & - & y^{(n)} \end{pmatrix}$$

Gradient Descent

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{n}\sum_{i=1}^{n}\left(h_\theta(x^{(i)}) - y^{(i)}\right) x_j^{(i)}$$

$$\frac{\partial J(\theta)}{\partial \theta_{j=j'}} = \begin{array}{l} (\theta_0 x_0^{(1)} + \theta_1 x_1^{(1)} + \ldots + \theta_f x_f^{(1)} - y^{(1)}) x_{j'}^{(1)} \\ + \\ (\theta_0 x_0^{(2)} + \theta_1 x_1^{(2)} + \ldots + \theta_f x_f^{(2)} - y^{(2)}) x_{j'}^{(2)} \\ + \\ \ldots \\ + \\ (\theta_0 x_0^{(n)} + \theta_1 x_1^{(n)} + \ldots + \theta_f x_f^{(n)} - y^{(n)}) x_{j'}^{(n)} \end{array}$$

$$X^T = \begin{pmatrix} x_0^{(1)} & x_0^{(2)} & \ldots & x_0^{(n)} \\ x_1^{(1)} & x_1^{(2)} & \ldots & x_1^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ x_f^{(1)} & x_f^{(2)} & \ldots & x_f^{(n)} \end{pmatrix} \qquad X\theta - y = \begin{pmatrix} \theta_0 x_0^{(1)} & + & \theta_1 x_1^{(1)} & + & \ldots & + & \theta_f x_f^{(1)} & - & y^{(1)} \\ \theta_0 x_0^{(2)} & + & \theta_1 x_1^{(2)} & + & \ldots & + & \theta_f x_f^{(2)} & - & y^{(2)} \\ \vdots & & \vdots & & \ddots & & \vdots & & \vdots \\ \theta_0 x_0^{(n)} & + & \theta_1 x_1^{(n)} & + & \ldots & + & \theta_f x_f^{(n)} & - & y^{(n)} \end{pmatrix}$$

$$\nabla J(\theta) = X^T (X\theta - y)$$

$$\theta \leftarrow \theta - \eta \, \nabla J(\theta)$$

Regularization

- Regularization introduces a penalty for large values of theta
- Capable of eliminating features to address overfitting

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{2}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)} \right) x_j^{(i)}$$

Regularization

- Regularization introduces a penalty for large values of theta
- Capable of eliminating features to address overfitting

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j} \qquad \frac{\partial J(\theta)}{\partial \theta_j} = \frac{2}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)} \right) x_j^{(i)}$$

$$J(\theta) = \frac{1}{n} \left[\sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)} \right)^2 + \lambda \sum_{j=1}^{f} \theta_j^2 \right]$$

$$\theta_j \leftarrow \theta_j \left(1 - \alpha \frac{\lambda}{n} \right) - \frac{\alpha}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)} \right) x_j^{(i)}$$

Regularization

- Worked Example

Polynomial Regression

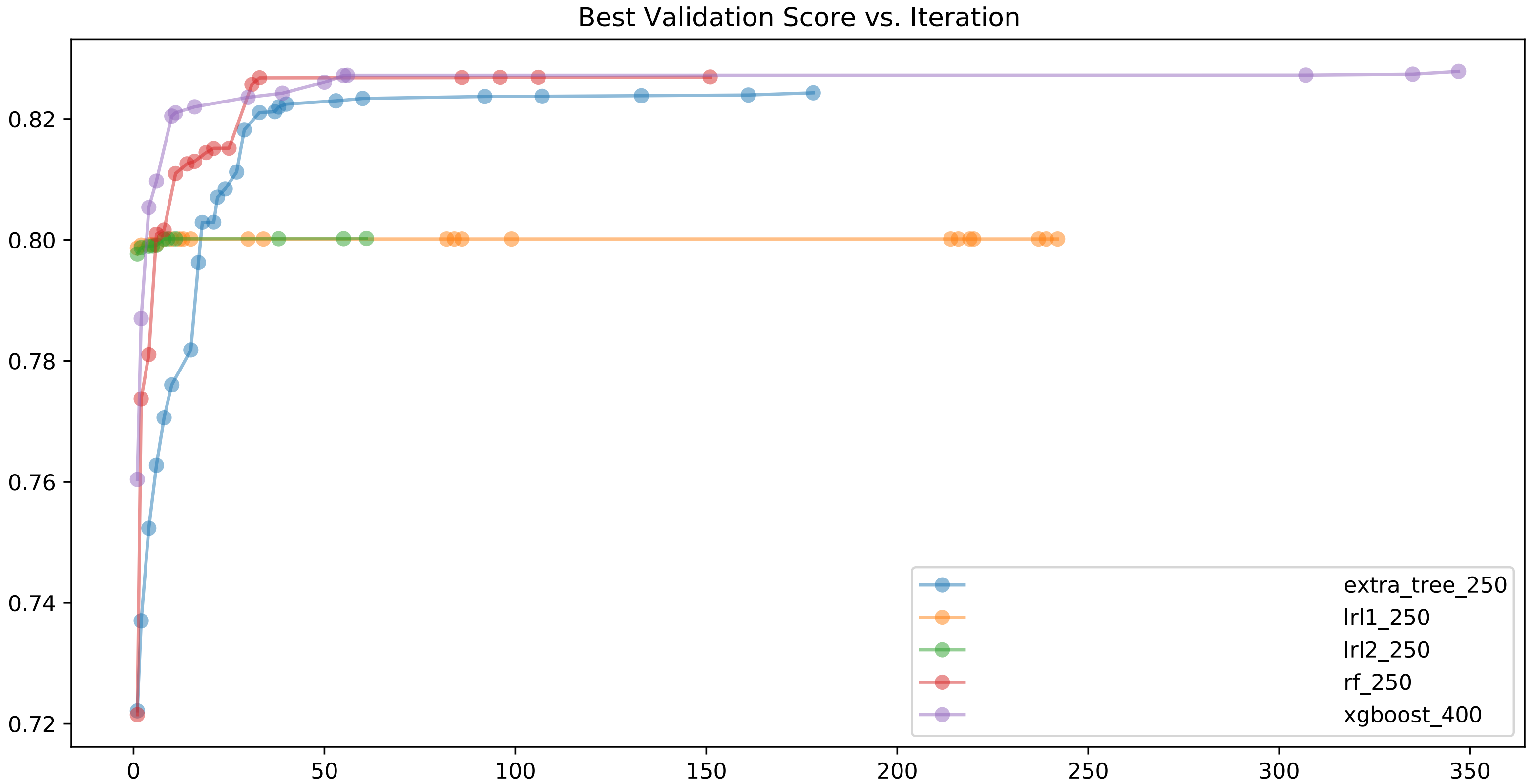
$$f(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_j x^j$$

- Typically, we don't know j beforehand
- sklearn.linear_model.LinearRegression still works

- Worked Example

Learning Curves

- Example from FLAML



Early Stopping

- Definition in XGBoost (early_stopping_rounds):
  Validation metric needs to improve at least once in
  every **early_stopping_rounds** rounds to continue training.

Early Stopping

- Definition in XGBoost (early_stopping_rounds):
  Validation metric needs to improve at least once in
  every **early_stopping_rounds** rounds to continue training.