Production ML Systems

Static vs. Dynamic Training

Broadly speaking, you can train a model in either of two ways:

- **Static training** (also called **offline training**) means that you train a model only once. You then serve that same trained model for a while.
- **Dynamic training** (also called **online training**) means that you train a model continuously or at least frequently. You usually serve the most recently trained model.

Static vs. Dynamic Inference

Inference is the process of making predictions by applying a trained model to **unlabeled examples**. Broadly speaking, a model can infer predictions in one of two ways:

- **Static inference** (also called **offline inference** or **batch inference**) means the model makes predictions on a bunch of common **unlabeled examples** and then caches those predictions somewhere.
- **Dynamic inference** (also called **online inference** or real-time inference) means that the model only makes predictions on demand, for example, when a client requests a prediction.

When to Transform Data?

Raw data must be feature engineered (transformed). When should you transform data? Broadly speaking, you can perform feature engineering during either of the following two periods:

- *Before* training the model
    - **Advantages**
        - The system transforms raw data only once.
        - The system can analyze the entire dataset to determine the best transformation strategy.

    - **Disadvantages**
        - You must recreate the transformations at prediction time
- *While* training the model
    - **Advantages**
        - You can still use the same raw data files if you change the transformations
        - Youre ensures the same transformations at training and prediction time
    - **Disadvantages**
        - Complicated transforms can increase model latency
        - Transformations occur for each and every batch

Deployment Testing

When deploying, your machine learning (ML) pipeline should run, update, and serve without a problem. If only deploying a model were as easy as pressing a big **Deploy** button. Unfortunately, a full machine learning system requires tests for:

- Validating input data.
- Validating feature engineering.
- Validating the quality of new model versions.
- Validating serving infrastructure.
- Testing integration between pipeline components.

Questions to Ask

- Is each feauture helpful?
- Is your data source reliable?
- Is your model part of a feedback loop?

Completion

You earned the **Machine Learning Crash Course: Production ML systems** badge!

The badge has been added to your profile.