FINTECH 540 - Machine Learning for Fintech

Fall Semester 2023

Third Lecture

Basics Concepts for Machine Learning

Duke

PRATT SCHOOL *of* ENGINEERING

Question 1

Considering the content of Deep Learning - Chapter 5, which of the following statements is true regarding the relationship between deep learning and machine learning?

A) Deep learning is a subset of machine learning that emphasizes using neural networks.

B) Deep learning and machine learning are interchangeable terms with no distinct differences.

C) Machine learning is a subset of deep learning focused on statistical analysis.

D) Deep learning is an older field of study compared to machine learning.

Considering the content of Deep Learning - Chapter 5, which of the following statements is true regarding the relationship between deep learning and machine learning?

A) Deep learning is a subset of machine learning that emphasizes using neural networks.

B) Deep learning and machine learning are interchangeable terms with no distinct differences.

C) Machine learning is a subset of deep learning focused on statistical analysis.

D) Deep learning is an older field of study compared to machine learning.

Correct Answer: A

Question 2

Considering Mitchell's definition of machine learning, which of the following scenarios best illustrates a computer program learning according to the criteria of experience (E), tasks (T), and performance measure (P)?

A) A program improves its chess-playing skills (T) by playing more games (E) and increasing its win rate (P).

B) A weather forecasting program improves its prediction accuracy (P) over time by analyzing more climate data (E) and refining its prediction algorithms for weather forecasting (T).

C) A data sorting program (T) that consistently sorts data accurately, but does not improve its sorting speed (P) over time (E).

D) A program designed to identify fraudulent transactions (T) but only improves its performance based on increased computational power, not experience (E).

Question 2

Considering Mitchell's definition of machine learning, which of the following scenarios best illustrates a computer program learning according to the criteria of experience (E), tasks (T), and performance measure (P)?

A) A program improves its chess-playing skills (T) by playing more games (E) and increasing its win rate (P).

B) A weather forecasting program improves its prediction accuracy (P) over time by analyzing more climate data (E) and refining its prediction algorithms for weather forecasting (T).

C) A data sorting program (T) that consistently sorts data accurately, but does not improve its sorting speed (P) over time (E).

D) A program designed to identify fraudulent transactions (T) but only improves its performance based on increased computational power, not experience (E).

Correct Answers: A, B

Question 3

What is the role of the data generating process in machine learning?

A) It helps in generating synthetic data for training models

B) It generates a probability distribution over datasets used in training and testing

C) It is used to create visual representations of data

D) It is used to detect anomalies in data

Question 3

What is the role of the data generating process in machine learning?

A) It helps in generating synthetic data for training models

B) It generates a probability distribution over datasets used in training and testing

C) It is used to create visual representations of data

D) It is used to detect anomalies in data

Correct Answer: B

Question 4

What do the readings suggest about the capacity of a machine learning model?

A) It should be chosen to avoid both underfitting and overfitting

B) It should always be high to capture complex patterns in data

C) It should always be low to avoid complexity

D) There are specific rules to find a good tradeoff for model complexity

Question 4

What do the readings suggest about the capacity of a machine learning model?

A) It should be chosen to avoid both underfitting and overfitting

B) It should always be high to capture complex patterns in data

C) It should always be low to avoid complexity

D) There are specific rules to find a good tradeoff for model complexity

Correct Answer: A

Today we will deal with various steps to come up with a machine learning model

- **Training** a model.
- **Underfitting-Overfitting** of a model.
- **Bias-Variance** tradeoff.

Remember the ML paradigms? Regardless of the approach, they all require a *training* process. Here, we'll focus on the supervised paradigm.

The training process

Remember the ML paradigms? Regardless of the approach, they all require a *training* process. Here, we'll focus on the supervised paradigm.

Training is the standard term in machine learning to indicate the *fitting of a model*, i.e., the search for the optimal model's parameters.

The training process

Remember the ML paradigms? Regardless of the approach, they all require a *training* process. Here, we'll focus on the supervised paradigm.

Training is the standard term in machine learning to indicate the *fitting of a model*, i.e., the search for the optimal model's parameters.

Training a model consists in providing the algorithm with the data to learn from and repeatedly updating the set of parameters of the algorithm based on the information contained in such data.

The training process

Remember the ML paradigms? Regardless of the approach, they all require a *training* process. Here, we'll focus on the supervised paradigm.

Training is the standard term in machine learning to indicate the *fitting of a model*, i.e., the search for the optimal model's parameters.

Training a model consists in providing the algorithm with the data to learn from and repeatedly updating the set of parameters of the algorithm based on the information contained in such data.

Hence, the training procedure differs depending on the machine learning paradigm, i.e., the kind of dataset experienced.

The Training Process (II)

Model training is the first step in all machine learning approaches. It produces a trained model that can be evaluated using chosen performance metrics and eventually deployed for production.

The Training Process (II)

Model training is the first step in all machine learning approaches. It produces a trained model that can be evaluated using chosen performance metrics and eventually deployed for production.

In a supervised model, we aim to learn the mapping $f(\mathrm{x}; \theta) = y$, where θ represents the model parameters.

The Training Process (II)

Model training is the first step in all machine learning approaches. It produces a trained model that can be evaluated using chosen performance metrics and eventually deployed for production.

In a supervised model, we aim to learn the mapping $f(\mathrm{x}; \theta) = y$, where θ represents the model parameters.

The **training process** is all about finding the best set of θ for the model.

Training a model is not just about fitting: we need to **generalize**. Do you know what that means?

Training a model is not just about fitting: we need to **generalize**. Do you know what that means?

Generalizing means preparing the algorithm to perform well *even* over unseen examples, not just on the data used for training. This is a **CRUCIAL**: *if a machine learning model does not generalize, it is useless.*

Training a model is not just about fitting: we need to **generalize**. Do you know what that means?

Generalizing means preparing the algorithm to perform well *even* over unseen examples, not just on the data used for training. This is a **CRUCIAL**: *if a machine learning model does not generalize, it is useless.*

How do I teach my model to generalize?

Training a model is not just about fitting: we need to **generalize**. Do you know what that means?

Generalizing means preparing the algorithm to perform well *even* over unseen examples, not just on the data used for training. This is a **CRUCIAL**: *if a machine learning model does not generalize, it is useless.*

How do I teach my model to generalize?

We need to organize the available data in a proper way...

A dataset is generally divided into two parts: the **training** set and the **test** set.

A dataset is generally divided into two parts: the **training** set and the **test** set.

An error metric is selected to measure the difference between the predicted output and the actual value.

A dataset is generally divided into two parts: the **training** set and the **test** set.

An error metric is selected to measure the difference between the predicted output and the actual value.

While this error computed on the training set is essential to evaluate the progress of the learning, the discrepancy over the test set is ultimately more important in machine learning because it represents the **generalization error**.

A dataset is generally divided into two parts: the **training** set and the **test** set.

An error metric is selected to measure the difference between the predicted output and the actual value.

While this error computed on the training set is essential to evaluate the progress of the learning, the discrepancy over the test set is ultimately more important in machine learning because it represents the **generalization error**.

The model does not use the data points included in the test set $\longrightarrow$ a good performance on this set of examples, i.e., a low test error, signals a **good generalization**.

ML theory assumes the data to be generated by the same data-generating process (DPG). What does this imply?

ML theory assumes the data to be generated by the same data-generating process (DPG). What does this imply?

Each data point is IID (independent and identically distributed) because it's drawn from the same underlying distribution $p(\mathbf{x}, y)$.

ML theory assumes the data to be generated by the same data-generating process (DPG). What does this imply?

Each data point is IID (independent and identically distributed) because it's drawn from the same underlying distribution $p(\mathbf{x}, y)$.

If we draw both the training and the test set from the same DPG, we would expect to have the same error (same value for $J(\theta)$).

ML theory assumes the data to be generated by the same data-generating process (DPG). What does this imply?

Each data point is IID (independent and identically distributed) because it's drawn from the same underlying distribution $p(\mathbf{x}, y)$.

If we draw both the training and the test set from the same DPG, we would expect to have the same error (same value for $J(\theta)$).

However, this is valid only if we fix the model parameters and then evaluate the algorithm over the training and the test set. Instead, we optimize the training set and use the test set just for evaluation purposes.

ML theory assumes the data to be generated by the same data-generating process (DPG). What does this imply?

Each data point is IID (independent and identically distributed) because it's drawn from the same underlying distribution $p(\mathbf{x}, y)$.

If we draw both the training and the test set from the same DPG, we would expect to have the same error (same value for $J(\theta)$).

However, this is valid only if we fix the model parameters and then evaluate the algorithm over the training and the test set. Instead, we optimize the training set and use the test set just for evaluation purposes.

To summarize: *an ML algorithm aims to reduce the training error while minimizing the test and training error gap.*

Usually, this process leads to two possible outcomes:

- **High training error** results in **underfitting**: the model cannot capture the information in the data.

Usually, this process leads to two possible outcomes:

- **High training error** results in **underfitting**: the model cannot capture the information in the data.

- **Low training error**, the significant gap between the test and the training error: the model **overfits** the training data, but it cannot generalize to unseen examples.

Usually, this process leads to two possible outcomes:

- **High training error** results in **underfitting**: the model cannot capture the information in the data.

- **Low training error**, the significant gap between the test and the training error: the model **overfits** the training data, but it cannot generalize to unseen examples.

One way to control the **overfitting-underfitting** tradeoff is to change the model capacity, i.e., the possibility to choose the function to represent the model and to fit the data from a broad set of functions. In simple terms, it is the **expressiveness** of a model.

The **model capacity** changes through the magnitude of the hypothesis space $\mathcal{H}$, the set of possible functions that could solve the ML problem.

The **model capacity** changes through the magnitude of the hypothesis space $\mathcal{H}$, the set of possible functions that could solve the ML problem.

Let's make an example: in a linear model, the set $\mathcal{H}$ includes all the possible linear equations that can fit the data.

The **model capacity** changes through the magnitude of the hypothesis space $\mathcal{H}$, the set of possible functions that could solve the ML problem.

Let's make an example: in a linear model, the set $\mathcal{H}$ includes all the possible linear equations that can fit the data.

Therefore, a linear model with one parameter has less expressiveness than a linear model with two parameters. A larger number of parameters implies a **more extensive hypothesis** space $\longrightarrow$, a more expressive way to represent the function of the data.

Overfitting-Underfitting (III)

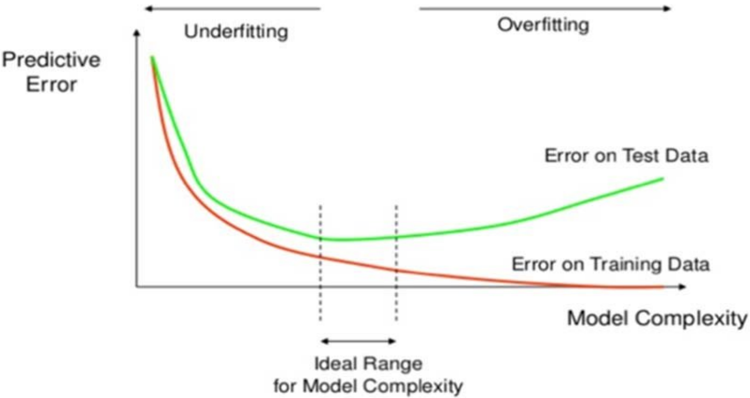
The **model capacity** changes through the magnitude of the hypothesis space $\mathcal{H}$, the set of possible functions that could solve the ML problem.

Let's make an example: in a linear model, the set $\mathcal{H}$ includes all the possible linear equations that can fit the data.

Therefore, a linear model with one parameter has less expressiveness than a linear model with two parameters. A larger number of parameters implies a **more extensive hypothesis** space $\longrightarrow$, a more expressive way to represent the function of the data.

We will see how this applies to more complex algorithms, such as regression trees and neural networks.

Overfitting-Underfitting relationship (IV)



Training the parameters of a model is an estimation of the set of actual parameters θ, which are **unknown**.

Training the parameters of a model is an estimation of the set of actual parameters θ, which are **unknown**.
Assume you have a dataset $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ of IID M data points, drawn from an unknown DGP. A **parameter estimator** is the set $\hat{\theta}$ that describes a valid mapping $f(\mathbf{x}; \hat{\theta}) + \epsilon = y$ for each point.

Training the parameters of a model is an estimation of the set of actual parameters θ, which are **unknown**.
Assume you have a dataset $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ of IID M data points, drawn from an unknown DGP. A **parameter estimator** is the set $\hat{\theta}$ that describes a valid mapping $f(\mathbf{x}; \hat{\theta}) + \epsilon = y$ for each point.
ML solves an approximation problem of unknown functions.

Training the parameters of a model is an estimation of the set of actual parameters θ, which are **unknown**.
Assume you have a dataset $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ of IID M data points, drawn from an unknown DGP. A **parameter estimator** is the set $\hat{\theta}$ that describes a valid mapping $f(\mathbf{x}; \hat{\theta}) + \epsilon = y$ for each point.
ML solves an approximation problem of unknown functions.

How do we evaluate the goodness of an estimator?

We should look at two important properties: **bias** and **variance**.

- The bias for an estimator $\hat{\theta}_N$ is defined as:

$$\text{Bias}(f(\mathbf{x};\hat{\theta})) = \mathbb{E}(f(\mathbf{x};\hat{\theta})) - f(\mathbf{x};\theta)$$

where the expectation is computed over the set of training samples of length M.

An estimator is **unbiased** if $\mathbb{E}(f(\mathbf{x};\hat{\theta})) = f(\mathbf{x};\theta)$ or **asymptotically unbiased** if $\lim_{M\to\infty} \mathbb{E}[f(\mathbf{x};\hat{\theta})] = f(\mathbf{x};\theta)$.

The bias reflects how well the estimator approximates the actual value on average.

- The variance of an estimator is simply the variance of the estimated function

$$\mathrm{Var}(f(\mathbf{x};\hat{\theta})) = \mathbb{E}\left[(f(\mathbf{x};\hat{\theta}) - \mathbb{E}[f(\mathbf{x};\theta)])^2\right]$$

The variance measures how much one would expect the estimator to vary if one computes it over another sample of the data from the same DGP.

Since the number of data is finite, the variance measures how much one would expect the estimate to vary when independently sampling the dataset multiple times from the same DGP.

Bias and **Variance** represent two different sources of error in the estimate

Remember that *bias measures the expected deviation from the actual value to be estimated. Conversely, variance calculates the deviation from the expected estimation, likely to occur if one changes the training examples.*

Bias and **Variance** represent two different sources of error in the estimate

Remember that *bias measures the expected deviation from the actual value to be estimated. Conversely, variance calculates the deviation from the expected estimation, likely to occur if one changes the training examples.*

What would be the ideal case for our estimator?

Bias and **Variance** represent two different sources of error in the estimate

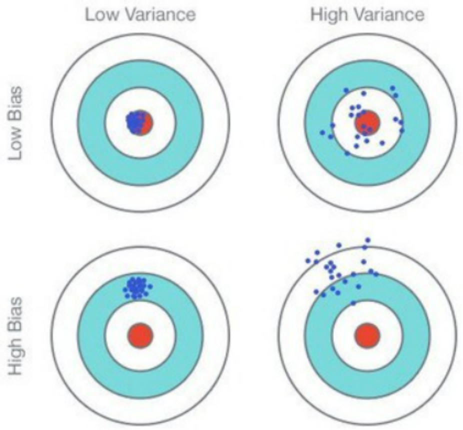
Remember that *bias measures the expected deviation from the actual value to be estimated. Conversely, variance calculates the deviation from the expected estimation, likely to occur if one changes the training examples.*

What would be the ideal case for our estimator?

Low bias and low variance are not always achievable.

Bias-Variance tradeoff.

Bias-Variance Tradeoff (II)



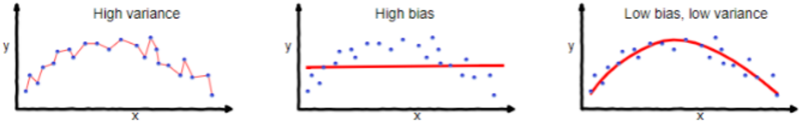
Source: Understanding Bias Variance tradeoff

Bias-Variance Tradeoff (III)

One can show that in estimating $f(\mathbf{x};\hat{\theta})$, the expected mean square error (MSE) on the test set is

$$\mathbb{E}[f(\mathbf{x};\hat{\theta}) - y]^2 = \mathrm{Var}(f(\mathbf{x};\hat{\theta})) + [\mathrm{Bias}(f(\mathbf{x};\hat{\theta}))]^2$$

In general, a model with **many parameters** can obtain estimates with lower bias and high variance. Could you tell why?



The first case shows a lack of generalization, the second offers a lack of fitting. The challenge in ML is to keep both measures low.

Duke
PRATT SCHOOL of ENGINEERING

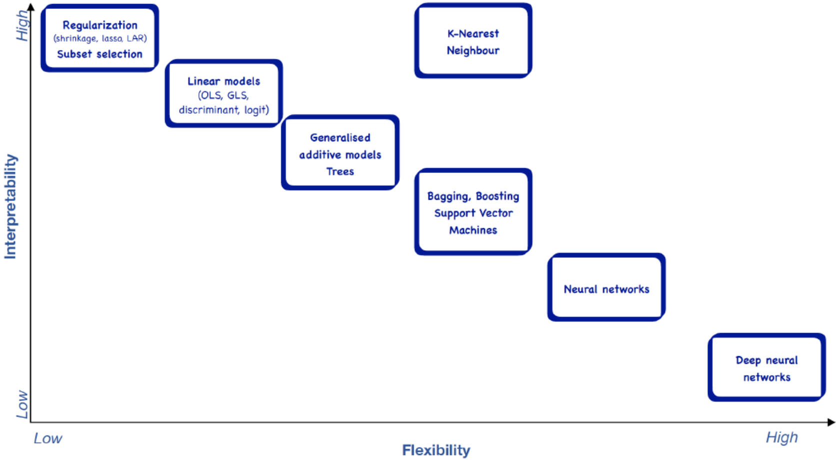
When trading off bias for variance, an aspect to consider is the interpretability of the **machine learning** model.

Being able to **explain** a model is crucial, even more in finance. Look at this hedge fund story.

There are specific frameworks to interpret even complex machine learning models. We are going to use them during the coming classes.

Remember that *the best model is not always the complex one.* Sometimes simple is better.

Flexibility vs. Interpretability (II)



Source: AIFI bootcamp

Do you know the difference between linear regression and machine learning regression?

Linear Regression:

- Fits a linear equation to data.
- Assumes a linear relationship between variables.
- Regression analysis enhances interpretability, used in economics, and finance.

Linear Regression vs. Machine Learning Regression

Linear Regression:

- Fits a linear equation to data.
- Assumes a linear relationship between variables.
- Regression analysis enhances interpretability, used in economics, and finance.

Machine Learning Regression:

- Broad category of techniques to handle complex relationships.
- Suitable for detecting non-linear patterns in high-dimensional data.
- Support Vector, Decision Trees, Random Forest, Neural Network Regression.
- Widely used in healthcare, engineering, NLP, etc.

Linear Regression:

- Fits a linear equation to data.
- Assumes a linear relationship between variables.
- Regression analysis enhances interpretability, used in economics, and finance.

Machine Learning Regression:

- Broad category of techniques to handle complex relationships.
- Suitable for detecting non-linear patterns in high-dimensional data.
- Support Vector, Decision Trees, Random Forest, Neural Network Regression.
- Widely used in healthcare, engineering, NLP, etc.

Machine learning regression captures complex patterns in data

Questions? Comments?