

Real-World Applications of Machine Learning

HIGHLIGHTS

- Introduction
 - ML in Healthcare
 - Natural Language Processing
 - Autonomous Vehicles
 - Recommender Systems
-

Introduction

Up until now, we have explored various technical approaches to machine learning (ML). In this context, we have seen different machine learning models, training methods, and data preparation techniques. We have also seen small-scale applications of machine learning used for solving specific types of problems in various domains. Now, in this chapter, we will turn our attention to the broader use cases of machine learning as they are applied in a variety of domains and industries. Although machine learning is being increasingly applied in several domains, this chapter will focus on the domains that are seeing the greatest potential benefits and research activity from ML. In this chapter, we will focus on ML applications in four domains: (i) ML in healthcare, (ii) Natural Language Processing (NLP), (iii) autonomous vehicles, and (iv) recommender systems. In each domain, we will discuss the state-of-the-art as well as current limitations and challenges faced by each specific domain in the context of applying machine learning to it.

ML in Healthcare

The healthcare domain has been transforming into a field that produces vast amount of data from a range of sources such as imaging, lab results, and electronic health records. The need to conclude a patient's health, paired with the vast amounts of produced data lead to the natural application of machine learning in healthcare. Although healthcare is a vast field, making the potential applications of ML in healthcare equally vast, we will focus on one of the key uses of ML in healthcare: early disease detection.

The most prominent clinical application of machine learning has been in the early detection and (human-aided) diagnosis of diseases by localizing and classifying abnormalities on medical images, particularly in early stages when they are difficult for humans (including trained professionals) to detect with the naked eye. Such medical imaging scans include, but are not limited to, magnetic resonance imaging (MRI), computed tomography (CT) scans, and X-rays. In this capacity, convolutional neural networks (CNNs) have been utilized to automatically segment cartilage and assess osteoarthritis risk in MRIs of the knee, outperforming manual non-ML approaches. Additionally, deep learning has been applied for segmenting multiple sclerosis lesions in 3D MRI scans as well as for differentially diagnosing benign and malignant breast nodules from ultrasound images. CNNs also matched the performance of dermatologists in classifying biopsy-proven clinical images of various skin cancers, including keratinocyte carcinomas and malignant melanomas, across a vast dataset of 130,000 images. This underscores the transformative potential of deep learning in enhancing medical image analysis and diagnostic accuracy.

In addition, several studies have applied deep learning to predict diseases from a patient's clinical status shown in their electronic health records (EHRs). For example, CNNs have been used to predict congestive heart failure and chronic obstructive pulmonary disease, demonstrating significant advantages over baseline methods. Recurrent neural networks (RNNs) have been used

on word embeddings to infer current illness states and predict future outcomes, incorporating medical interventions for dynamic predictions. RNNs have also been used for predicting diagnoses and medications based on patient history, showcasing higher recall than non-deep learning-based baselines. Deep learning was also applied to model continuous time signals, such as laboratory results, for automatic phenotype identification. Additionally, natural language models (which will be discussed in more detail below) were employed to learn embedded representations of medical concepts from EHRs, demonstrating applications in predicting suicide risk and de-identifying clinical notes. Notably, the prediction of unplanned patient readmissions after discharge has received attention, with one study showing that CNNs can accurately predict readmissions within 6 months. The use of predictive models such as CNNs help with early intervention, where such readmissions can be avoided when detected early enough.

Despite these exciting advancements, ML incurs several limitations in their applications to healthcare. Its primary challenge revolves around trust and accuracy. Current state-of-the-art ML models in healthcare still make mistakes and, in the context of healthcare, mistakes made by an ML model can lead to incorrect diagnoses and a host of other problems. This challenge largely stems from the lack of medical data volume compared to other domains. Health data's heterogeneity, temporality, and complexity present hurdles for training effective models, leading to less generalizable models. Existing deep learning models often assume static inputs, neglecting the evolving nature of diseases over time. Moreover, the interpretability of deep learning models in healthcare is crucial for gaining medical professionals' trust. Despite these challenges, opportunities for future research include feature enrichment by capturing diverse patient data sources, federated inference to maintain data privacy across clinical institutions so that data does not have to be physically transmitted and addressing model privacy concerns through differential

privacy standards. Incorporating expert knowledge, temporal modeling, and developing interpretable modeling methods are essential directions for enhancing the effectiveness and trustworthiness of ML in healthcare.

Natural Language Processing

The field of natural language processing (NLP) is primarily devoted to increasing computers' understanding of human language in both spoken and written forms. Digital voice assistants on mobile phones as well as large language models (LLMs) are both prominent examples of ML-powered NLP.

The majority of NLP algorithms are based on transformers, which are a class of neural network architectures like CNNs and RNNs. Recently, generative pre-trained transformers (GPT) have received a lot of attention due to their impressive ability to learn from large volumes of text and then generate unique responses based on various prompts. This has been most prolific with OpenAI's ChatGPT, which is a large language model (LLM) designed to understand and generate human-like text using patterns and information obtained during training consisting of various pieces of text on the internet. At its core, LLMs, including ChatGPT, are machine learning models trained according to the same pipeline we have seen throughout this text. For example, when GPT-3.5 was prompted with "How do you work?," it responded with the following:

I am a machine learning model called GPT-3.5, developed by OpenAI. My architecture is known as the "Generative Pre-trained Transformer 3.5." Here's a simplified explanation of how I work:

*1. **Pre-training:** I've been trained on a diverse range of internet text up until my knowledge cutoff date in January 2022. This pre-training involves predicting what comes next in a sentence, given the context of the previous words. This*

process helps me learn grammar, facts, reasoning abilities, and some level of world knowledge.

2. **Architecture:** I am based on a transformer architecture, which is a type of neural network architecture. Transformers are known for their ability to process and generate human-like text by attending to different parts of a sequence of words.

3. **Fine-tuning:** While I'm not fine-tuned to specific tasks or datasets, my general architecture can be adapted for different applications by fine-tuning on specific data. However, in public usage, I don't have access to information about specific individuals unless it has been shared with me in the course of our conversation.

4. **Natural Language Understanding:** I can understand and generate human-like text, making me versatile in various language-related tasks, such as answering questions, generating creative content, translating languages, and more.

5. **Limitations:** Despite my capabilities, I have limitations. I might generate incorrect or nonsensical answers, be sensitive to the input phrasing, and lack a true understanding of context. I don't have personal experiences, emotions, or consciousness. My responses are generated based on patterns learned during training.

In summary, I work by using patterns and information learned during my training to generate human-like text based on the input I receive. My purpose is to assist and provide information to the best of my ability.

As we can see from its own response above, LLMs are trained using the same ML pipeline that we have seen multiple times throughout this text. The specifics of the training, of course, differ

between models but the overarching approach, in terms of architecture selection, data collection, model training, etc., is the same as it is for any ML model.

Autonomous Vehicles

The integration of machine learning (ML) into autonomous vehicles is a highly researched area as it has potential for a wide range of civilian and military applications. Beyond the familiar realm of self-driving cars, ML algorithms are instrumental in enabling various autonomous vehicles, including unmanned aerial systems (UAS), maritime vessels, and delivery robots, to perceive their surroundings, make decisions, and navigate dynamically. Here, we will discuss applications and limitations of ML in autonomous vehicles in a variety of fields.

The most prevalent application of ML in autonomous vehicles is for self-driving cars. Such vehicles employ an array of sensors, cameras, and lidar to perceive their environment, with ML algorithms processing the data collected from these sensors in real-time. Deep learning models are usually trained offline on a collection of real-world and simulated data. During deployment, models can also be actively trained. The changes incurred from online training could be local to a particular vehicle's software, or it could be reflected in all vehicles through a software update.

UASs is becoming a more prominent application for leveraging the autonomy provided by ML algorithms for various tasks including obstacle avoidance, navigation, and route planning. As these vehicles traverse dynamic environments, ML algorithms analyze sensor data to adapt to changes, avoid collisions, and optimize their flight paths. For example, in agriculture, UASs equipped with ML algorithms can assess crop health, identify pests, and optimize irrigation based on real-time data by flying around farms, taking pictures with an onboard camera, and processing them locally on the UAS with results being sent directly to appropriate personnel. ML algorithms for such

applications work very similarly as they do for self-driving cars. Specifically, data is collected and deep learning models are first trained offline. Then, models are deployed on the UAS for online processing of data.

Autonomous ships and underwater vehicles also employ ML for navigation, collision avoidance, and path planning. Equipped with sensors such as sonar and radar, these vessels utilize deep learning algorithms to navigate the complexities of maritime environments in a similar fashion to their usage in other autonomous vehicles. Maritime applications are still an emerging area, but they have promise for both civilian and military purposes.

Currently, the ubiquitous deployment of ML on fleets of autonomous vehicles is limited by two main factors. First, the collected data is often not representative of the environments the autonomous vehicle will encounter. For example, consider the problem of deploying an ML system for a fleet of ships for navigation. The objective of the model is to perform object detection by processing images from the vessel to avoid collisions. In order to do this, data would be collected from existing surface vessels to create a training set. However, if data were only collected during the day when the weather was clear, the model would have poor performance during the night and/or in snowy weather. One solution to this problem could be to collect data in these conditions. However, we are restricted to data collection on the days the vessels are actually in motion. This would ultimately lead to a non-representative dataset. Second, transmitting data collected on multiple participating autonomous agents may not be feasible due to security risks. For example, if navigation data is being actively collected, it may be risky to transmit that data in real-time. One method, known as federated learning (FL), has emerged in an attempt to remedy this problem. FL allows for ML models to be trained in a distributed manner and only requires model parameters to

be transmitted rather than data itself. Although FL is a promising solution towards privacy-preserving ML, it is still susceptible to malicious attacks in which the transmitted parameters can be intercepted and reverse-engineered to recover the training data. Current research is aimed at directly addressing these types of limitations in an effort to move towards the deployment of robust and secure ML systems in autonomous vehicles.

Recommender Systems

Recommendation systems are a type of information-filtering system that provides suggestions for items that are most pertinent to a particular user. For example, new connection requests recommended on social media platforms, merchandise recommended for users based on their previous purchases on online shopping websites, and titles suggested for users to watch provided by streaming services are all examples of ML-powered recommendation systems. User engagement and content consumption have experienced transformative advancements in personalization due to the integration of machine learning (ML) into recommendation systems. These systems, designed to provide user-specific suggestions, leverage ML algorithms to analyze user behavior and preferences. Here, we will explore the intricacies of how ML is harnessed in recommendation systems. Let's begin by examining the different types of ML-based recommendations systems.

In the realm of recommendation systems, collaborative filtering stands out as a prominent approach. This technique relies on users' interactions with available items as the foundation for generating recommendations. It can be implemented in two forms: user-based collaborative filtering, which compares a user's preferences with those of similar users, and item-based

collaborative filtering, which suggests items based on their similarity to those previously liked by the user. This is often used by streaming services to recommend certain titles and videos to a user.

Content-based filtering, another type of ML-based recommendation system, takes a different route by recommending items based on the features intrinsic to those items. NLP techniques often play a crucial role in extracting relevant features from textual data. For example, music may be recommended based on the titles that are typically listened to.

Matrix factorization, a more nuanced technique, involves breaking down user-item interaction matrices into labels that represent underlying characteristics. Linear algebra approaches such as Singular Value Decomposition (SVD) and Alternating Least Squares (ALS) are commonly used in matrix factorization. Matrix factorization is often used to recommend products based on user purchase history and preferences.

Training an ML model for recommendation systems commences with data collection and preprocessing. These systems are usually trained on extensive user data, including preferences, interactions, and contextual information. Rigorous preprocessing is required to clean and transform data into a suitable format for training.

Feature engineering plays a pivotal role in refining the model's performance. Depending on the recommendation system type, relevant data attributes are selected and transformed. For collaborative filtering, this may involve user preferences, while content-based systems might focus on item features. The features used to train recommendation systems are usually unique to the context for which the system is being developed.

Model training, a critical phase, can take the form of supervised or unsupervised learning approaches, depending on the recommendation system's nature. Algorithms such as k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), and neural networks are commonly employed. Deep learning models have also gained traction for their ability to capture intricate patterns in user behavior.

Despite the advancements facilitated by machine learning in recommendation systems, certain limitations persist. The cold start problem poses a considerable challenge, particularly when dealing with new users or items lacking sufficient interaction data. The cold start problem refers to systems not having any user data prior to their usage of the platform. Data sparsity further complicates the accuracy of recommendations, especially when users have limited interactions with items. Matrix factorization techniques may exacerbate this issue as they rely on complete user-item interaction matrices. Overfitting, a common concern in machine learning, can lead to overly specific recommendations tailored to individual user preferences, resulting in poor generalization to new users or items. Mitigating strategies such as regularization techniques and careful model tuning are essential to address this challenge. Thus, despite the exciting strides made in ML-based recommendation systems, several challenges still present themselves and are the focus of current research in the field.

Closing Remarks

Throughout this text, we have seen an introduction to current machine learning approaches. This has included data collection and processing, appropriate methods for data splitting, various types of machine learning using a wide range of models, model evaluation metrics, deep learning approaches, and applications of machine learning in different real-world contexts. Several of the

concepts that are commonly used in machine learning are not new and were developed several decades ago (e.g., the idea of artificial neural networks was developed in the mid-1900s). However, the development of more capable computers paired with the plethora of data that has been produced by digital computers and sensors over the recent decades has accelerated the advancements of ML. Although ML has matured quite a bit, as we saw throughout this text, it still faces several challenges that must be addressed before ML can be widely adopted. The next few decades are expected to be more towards secure, reliable, and robust machine learning.