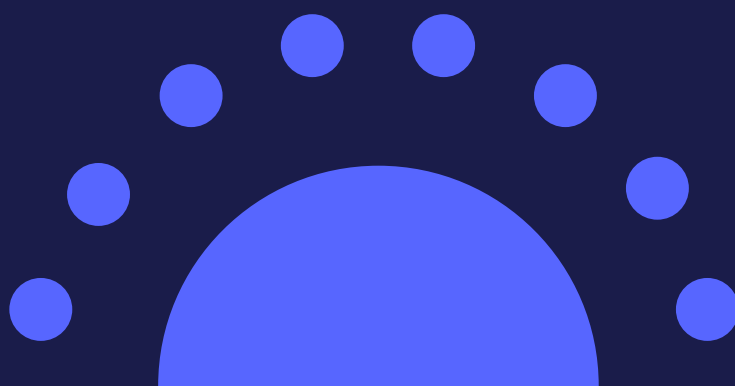




הכשרת אנבידיה AI



על הקורס

סדנה מעמיקה ומעשירה, שנועדה להעניק כלים וידע פורצי דרך בתחום הבינה המלאכותית. הסדנה כוללת חמישה קורסים ייחודיים המתמקדים בתחומים מבוקשים ומתפתחים, ומעניקים הסמכות בינלאומיות מטעם: NVIDIA.

על התוכנית

היכרות עם למידה עמוקה (Getting Started with Deep Learning)
עסקים ברחבי העולם משתמשים בבינה מלאכותית כדי להתמודד עם אתגרים מרכזיים: אנשי רפואה משתמשים בה לאבחון מהיר ומדויק יותר; קמעונאים מציעים חוויות קנייה מותאמות אישית; ויצרני רכב משפרים את הבטיחות והיעילות בתחבורה. ללמידת עומק, שהיא שיטה מתקדמת של בינה מלאכותית המשתמשת ברשתות נוירונים בשכבות, מאפיינת ביצועים מתקדמים במשימות כמו זיהוי אובייקטים, זיהוי דיבור ותרגום שפה. גישה זו מאפשרת למחשבים לזהות דפוסים מורכבים שאינם ניתנים לזיהוי באמצעות תוכנה מסורתית.

מודלים מבוססי ראייה ממוחשבת עם נתונים סינתטיים

(Bootstrapping Computer Vision Models With Synthetic Data)

בקורס זה, נשתמש ב־NVIDIA Omniverse Replicator ובתוסף Omniverse Defect Extension ליצירת נתונים סינתטיים. לאחר מכן, נעבור תהליך של חזרה על מערך הנתונים כדי לאמן רשת נוירונים עמוקה (DNN) לאיתור אובייקטים מטרה (כגון שריטות) בצבעה.

בניית מערכות AI מבוססות לאבטחת סייבר

(Building AI-Based Cybersecurity Pipelines)

שיטות אבטחת סייבר מסורתיות כוללות יצירת חומות הגנה מסביב לתשתיות כדי להגן עליהן מפני פולשים. עם זאת, ככל שחברות מאמצות את תהליך הטרנספורמציה הדיגיטלית שלהן, הן מתמודדות עם ריבוי מכשירים, תוקפים מתוחכמים יותר ורשת נתונים עצומה שיש להגן עליה – דבר שמצריך גישה חלופית. גישה חלופית היא להתייחס לאבטחת סייבר כמערכת מידע נתונים: המטרה היא לזהות ולנטר את כל המשתמשים והפעולות ברשת, כדי שניתן יהיה לזהות אילו עסקאות ואילו פעולות עשויות להיות זדוניות.

על התוכנית

פיתוח מהיר של יישומים עם מודלי שפה גדולים

(Rapid Application Development with Large Language Models – LLMs)

בקורס זה נלמד לפתח יישומים מבוססי LLM על ידי חקר קהילת הקוד הפתוח, הכוללת LLMs מאומנים מראש.

בניית סוכני RAG עם מודלי שפה

(Building RAG Agents with LLMs)

סוכנים המופעלים על ידי מודלים שפתיים גדולים (LLMs) הוכיחו יכולות אחזור מרשימות לשימוש בכלים, בחינת מסמכים ותכנון גישות פעולה. קורס זה ילמד כיצד לפרוס מערכת סוכנים בפועל, עם גמישות להרחיב את המערכת כך שתתאים לדרישות של משתמשים ולקוחות.

קהל יעד

הסדנה מתאימה למגוון משתתפים ותספק ידע, כלים, וניסיון מעשי בעולמות הבינה המלאכותית, הראייה הממוחשבת, הסייבר, ופיתוח יישומים מבוססי מודלים מתקדמים.

היקף הקורס

40 שעות

בסיום ההכשרה תינתן תעודת הסמכה בינלאומית של אינבידה

תנאי סף

- פייתון כללי, בדגש על סינטקס, פונקציות בסיסיות, מערכים ומילונים.
- דוקרים ברמת היכרות של הכלי
- מבחן כניסה

Day 1 ► Fundamentals of Deep Learning

Course Details

Duration: 09:00 – 17:00

Level: Technical – Beginner

Subject: Deep Learning

Language: English

Course Prerequisites: An understanding of fundamental programming concepts in Python 3, such as functions, loops, dictionaries, and arrays; familiarity with Pandas data structures; and an understanding of how to compute a regression line.

Suggested materials to satisfy prerequisites: Python Beginner's Guide.

Technologies: PyTorch, Pandas

Assessment Type: Skills-based coding assessments evaluate students' ability to train a deep learning model to high accuracy.

Certificate: Upon successful completion of the assessment, participants will receive an NVIDIA DLI certificate to recognize their subject matter competency and support professional career growth.

Hardware Requirements: Desktop or laptop computer capable of running the latest version of Chrome or Firefox. Each participant will be provided with dedicated access to a fully configured, GPU-accelerated server in the cloud.

About this Course

Businesses worldwide are using artificial intelligence to solve their greatest challenges. Healthcare professionals use AI to enable more accurate, faster diagnoses in patients. Retail businesses use it to offer personalized customer shopping experiences. Automakers use it to make personal vehicles, shared mobility, and delivery services safer and more efficient. Deep learning is a powerful AI approach that uses multi-layered artificial neural networks to deliver state-of-the-art accuracy in tasks such as object detection, speech recognition, and language translation. Using deep learning, computers can learn and recognize patterns from data that are considered too complex or subtle for expert-written software

Learning Objectives

- ▶ Learn the fundamental techniques and tools required to train a deep learning model
- ▶ Gain experience with common deep learning data types and model architectures
- ▶ Enhance datasets through data augmentation to improve model accuracy
- ▶ Leverage transfer learning between models to achieve efficient results with less data and computation
- ▶ Build confidence to take on your own project with a modern deep learning framework

Topics Covered

- ▶ PyTorch
- ▶ Convolutional Neural Networks (CNNS)
- ▶ Data Augmentation
- ▶ Transfer Learning
- ▶ Natural Language Processing

Course Outline

The below is a suggested timeline for the course. Please work with the instructor to find the best timeline for your session.

Introduction

- Meet the instructor.
- Create an account at courses.nvidia.com/join

The Mechanics of Deep Learning

Explore the fundamental mechanics and tools involved in successfully training deep neural networks:

- Train your first computer vision model to learn the process of training.
- Introduce convolutional neural networks to improve accuracy of predictions in vision applications.
- Apply data augmentation to enhance a dataset and improve model generalization.

Course Outline

Pre-trained Models and Large Language Models

Leverage pre-trained models to solve deep learning challenges quickly. Train recurrent neural networks on sequential data:

- Integrate a pre-trained image classification model to create an automatic doggy door.
- Leverage transfer learning to create a personalized doggy door that only lets in your dog.
- Use a Large Language Model (LLM) to answer questions based on provided text.

Final Project: Object Classification

Apply computer vision to create a model that distinguishes between fresh and rotten fruit:

- Create and train a model that interprets color images.
- Build a data generator to make the most out of small datasets.
- Improve training speed by combining transfer learning and feature extraction.

Final Project: Object Classification

- Discuss advanced neural network architectures and recent areas of research where students can further improve their skills.

Course Outline

Final Project: Object Classification Final Review

- Integrate a pre-trained image classification model to create an automatic doggy door.
- Leverage transfer learning to create a personalized doggy door that only lets in your dog.
- Use a Large Language Model (LLM) to answer questions based on provided text.

Day 2 ▶ Adding New Knowledge to LLMs

Course Details

Duration: 09:00 –17:00

Level: Technical – Intermediate

Subject :Generative AI/LLM

Language: English

Course Prerequisites

Familiarity with Python programming and Jupyter notebooks.

Comfort reading JSON, YAML, notebook outputs, and model responses.

Basic understanding of LLM prompting and API-shaped model calls.

No prior experience with NeMo AutoModel, NeMo Data Designer, NeMo Curator, or NeMo RL is required.

Tools, libraries, and services used

- ▶ Python, NVIDIA NeMo AutoModel, NVIDIA NeMo Data Designer, NVIDIA NeMo Curator, NVIDIA NeMo RL, vLLM, PEFT, Docker, JupyterLab

About this Course

Large Language Models (LLMs) are increasingly capable, but production workflows often require more than general capability. Teams need models that follow local taxonomies, return reliable schemas, respect domain tradeoffs, and improve on a task that can be measured before deployment.

In this full-day hands-on workshop, you will design and run an end-to-end model-customization pipeline for a domain-specific triage workflow. You will begin by building an evaluation that defines what "better" means for the task, then measure a stock 8B open-weight model, test the ceiling of prompting, generate targeted synthetic training data, train a LoRA adapter with supervised fine-tuning, and use Direct Preference Optimization (DPO) to improve response quality after task behavior is mostly in place.

The workshop is grounded in Meridian Response, a fictional humanitarian logistics organization that needs to classify incoming flood-response field reports and route them to the right team. Meridian serves as a running case study for the pipeline decisions you are learning to make. The final capstone transfers the same decision sequence to Atlas Support, a customer-support triage domain, so you can practice applying the workflow beyond the original case study.

Learning Objectives

By participating in this workshop, participants will be equipped to:

- ▶ Design an evaluation that captures task behavior, measurable success, and important failure modes before training.
- ▶ Use evaluation evidence to decide whether prompting, synthetic data generation, supervised fine-tuning, or preference optimization is the right next move.
- ▶ Generate targeted synthetic data for a domain task using a teacher model and NVIDIA NeMo Data Designer.
- ▶ Inspect and curate generated data using quality checks and NVIDIA NeMo Curator.
- ▶ Train a task-focused LoRA adapter with supervised fine-tuning using NVIDIA NeMo AutoModel.
- ▶ Serve and evaluate model stages with vLLM, comparing base, prompted, SFT, and DPO behavior against task metrics.
- ▶ Build and assess preference pairs for Direct Preference Optimization using NVIDIA NeMo RL.
- ▶ Reason about model-customization tradeoffs and communicate evidence-based decisions to teammates and coding agents.

Topics Covered

In service of teaching how to add knowledge and task behavior to LLM-based systems, this workshop will cover the following topics and technologies:

- ▶ Evaluation-first model customization
- ▶ Task metrics, baseline measurement, and in-context learning ceiling tests
- ▶ Synthetic data generation with NVIDIA NeMo Data Designer and a Nemotron teacher model
- ▶ Data inspection, curation, and deduplication with NVIDIA NeMo Curator
- ▶ Supervised Fine-Tuning (SFT) with LoRA adapters using NVIDIA NeMo AutoModel and PEFT
- ▶ Direct Preference Optimization (DPO) with NVIDIA NeMo RL
- ▶ Model serving and evaluation with vLLM
- ▶ Decision frameworks for choosing the smallest technique that supplies the missing signal

Course Outline

Orientation

- Understand the workshop goal, lab environment, and JupyterLab workflow.
- Meet Meridian Response, the running case study used to ground the pipeline decisions.
- Review the core tools used in the workshop: NVIDIA NeMo AutoModel, NeMo Data Designer, NeMo Curator, NeMo RL, vLLM, LoRA/PEFT, and Jupyter notebooks.

Evaluation First

- Design an evaluation instrument before generating data or training.
- Measure the base 8B model on Meridian's classification task.
- Run an in-context learning ceiling test to see how much prompting can improve the model.
- Compare local model behavior with larger API models and decide what evidence should guide training.

Synthetic Data Generation and Curation

- Use the eval to identify the training signal the model is missing.
- Select and use a teacher model for synthetic data generation.
- Build targeted records with NVIDIA NeMo Data Designer.
- Inspect, filter, deduplicate, and curate records with NVIDIA NeMo Curator.

Course Outline

Supervised Fine-Tuning

- Learn how supervised training teaches a model task behavior from examples.
- Understand why LoRA is useful for efficient domain adaptation.
- Configure and run a one-epoch SFT adapter training job with NVIDIA NeMo AutoModel.
- Serve the trained adapter with vLLM and evaluate whether it improves the held-out task eval.

Direct Preference Optimization

- Identify the response-quality gap that remains after SFT.
- Design a style rubric and build chosen/rejected preference pairs.
- Evaluate preference-pair quality before training.
- Run and inspect a DPO workflow with NVIDIA NeMo RL, then evaluate how preference optimization changed the model.

Capstone

- Apply the same customization sequence to Atlas Support, a new customer-support triage domain.
- Build and evaluate a baseline task eval.
- Generate synthetic data and train an SFT adapter for the new task.
- Use assessment gates to confirm that each stage is complete and evidence-based.



Course Outline

Take Home

- Review the decision playbook for model customization.
- Download reference notebooks, source material, and take-home assets.
- Connect the workshop workflow to future domain-specific customization projects.

Day 3 Rapid Application Development with Large Language Models (LLMs)

Course Details

Duration: 09:00 – 17:00

Level: Technical – Beginner

Subject: Generative AI/LLM

Language: English

Course Prerequisites:

Introductory deep learning, with comfort with PyTorch and transfer learning preferred. Content covered by DLI's Getting Started with Deep Learning or Fundamentals of Deep Learning courses, or similar experience is sufficient. Intermediate Python experience, including object-oriented programming and libraries. Content covered by Python Tutorial (w3schools.com) or similar experience is sufficient.

Tools, libraries, frameworks used: Python, PyTorch, HuggingFace, Transformers, LangChain, and LangGraph

About this Course

Recent advancements in both the techniques and accessibility of large language models (LLMs) have opened up unprecedented opportunities to help businesses streamline their operations, decrease expenses, and increase productivity at scale.

About this Course

Additionally, enterprises can use LLM-powered apps to provide innovative and improved services to clients or strengthen customer relationships. For example, enterprises could provide customer support via AI companions or use sentiment analysis apps to extract valuable customer insights. In this course you will gain a strong understanding and practical knowledge of LLM application development by exploring the open-sourced ecosystem including pretrained LLMs, enabling you to get started quickly in developing LLM-based applications.

Learning Objectives

By participating in this workshop, you will:

- Find, pull in, and experiment with the HuggingFace model repository and Transformers API.
- Use encoder models for tasks like semantic analysis, embedding, question-answering, and zero-shot classification.
- Work with conditioned decoder-style models to take in and generate interesting data formats, styles, and modalities.
- Kickstart and guide generative AI solutions for safe, effective, and scalable natural data tasks.
- Explore the use of LangChain and LangGraph for orchestrating data pipelines and environment-enabled agents.

Topics Covered

The workshop covers large language models from beginning to end, starting with fundamentals of transformers, progression into foundational large language models, and finishing in model/agent orchestration. Each of these sections is designed to equip participants with the knowledge and skills necessary to progress further in developing useful LLM-powered applications.

Course Outline

The table below is a suggested timeline for the course. Please coordinate with the instructor for the best timeline for your session.

Course Introduction	• Overview of workshop topics and schedule. • Introduction to HuggingFace and Transformers. • Discuss how LLMs can enhance enterprise applications.
Transformers and LLMs	• Introduce and motivate the transformer-style architecture from deep learning first principles. • Understand input-output processing with tokenizers, embeddings, and attention mechanisms.
Task-Specific Pipelines	• Profile encoder models for different NLP tasks where they are most useful.

Course Outline

Task-Specific Pipelines

- Investigate the use of lightweight models for natural language embedding, classification, subsetting, and zero-shot prediction.

Seq2Seq with Decoders

- Introduce GPT-style decoder models for sequence generation and autoregressive tasks.
- Apply encoder-decoder architectures for applications like machine translation and few-shot task completion.

Multimodal Architectures

- Integrate different data modalities (text, images, audio) into LLM workflows.
- Explore multimodal models like CLIP for cross-modal learning, visual language models for image question-answering, and diffusion models for text-guided image generation.

Scaling Text Generation

- Explore LLM inference challenges and deployment strategies, including optimized server deployments.
- Incorporate LLMs into interesting applications that can scale to larger repositories and user bases.

Course Outline

Orchestration and Agentics

- Introduce LangChain for LLM orchestration and agentic workflows.
- Investigate use of agentics and tool-calling for integrating natural language with standard applications and data.

Final Assessment

- Build an LLM-based application integrating text generation, multimodal learning, and agentic orchestration.

Review and Wrap-up

- Review key learnings and answer final questions.
- Earn a certificate upon successful completion.
- Complete the workshop survey.

Day 4 Building RAG Agents with LLMs

Course Details

Duration: 09:00 – 17:00

Level: Technical – Intermediate

Subject: Generative AI/LLM

Language: English

Course Prerequisites:

- Introductory deep learning knowledge, with comfort with PyTorch and transfer learning preferred.
- Intermediate Python experience, including object-oriented programming and libraries.

About this Course

Agents powered by large language models (LLMs) are quickly gaining popularity as people are finding new capabilities and opportunities to greatly improve their productivity. An especially powerful recent development has been the popularization of retrieval-based LLM systems that can hold informed conversations by using tools, looking at documents, and planning their approaches. These systems are fun to experiment with and offer unprecedented opportunities to make life easier, but they also require many queries to large deep learning models and need to be implemented efficiently.

About this Course

This course will observe how you can deploy an agent system in practice and scale up your system to meet the demands of users and customers. Along the way, you'll learn advanced LLM orchestration techniques for internal reasoning, dialog management, tooling, and retrieval.

Learning Objectives

This course will observe how you can deploy an agent system in practice and scale up your system to meet the demands of users and customers. Along the way, you'll learn advanced LLM orchestration techniques for internal reasoning, dialog management, tooling, and retrieval.

- ▶ Compose an LLM system that can interact predictably with a user by leveraging internal and external reasoning components.
- ▶ Design a dialog management and document reasoning system that maintains state and coerces information into structured formats.
- ▶ Leverage embedding models for efficient similarity queries for content retrieval and dialog guardrailing.
- ▶ Implement, modularize, and evaluate a RAG agent that can answer questions about the research papers in its dataset without any fine-tuning.

Topics Covered

The workshop includes topics such as LLM Inference Interfaces, Pipeline Design with LangChain, Gradio, and LangServe, Dialog Management with Running States,

Topics Covered

Working with Documents, Embeddings for Semantic Similarity and Guardrailing, and Vector Stores for RAG Agents. Each of these sections is designed to equip participants with the knowledge and skills necessary to develop and deploy advanced LLM systems effectively.

Course Outline

- ▶ Introduction to the workshop and setting up the environment.
- ▶ Exploration of LLM inference interfaces and microservices.
- ▶ Designing LLM pipelines using LangChain, Gradio, and LangServe.
- ▶ Managing dialog states and integrating knowledge extraction.
- ▶ Strategies for working with long-form documents.
- ▶ Utilizing embeddings for semantic similarity and guardrailing.
- ▶ Implementing vector stores for efficient document retrieval.
- ▶ Evaluation, assessment, and certification.

Day 5 ► Fundamentals of Accelerated Computing with Modern CUDA C++

Course Details

Duration: 09:00 –17:00

Level: Technical – Beginner

Subject: Accelerated Computing

Language: English

Course Prerequisites

Basic C++ competency, including familiarity with lambda expressions, loops, conditional statements, functions, standard algorithms and containers.

No previous knowledge of CUDA programming is assumed

About this Course

This workshop provides a comprehensive introduction to general-purpose GPU programming with CUDA. You'll learn how to write, compile, and run GPU-accelerated code, leverage CUDA core libraries to harness the power of massive parallelism provided by modern GPU accelerators, optimize memory migration between CPU and GPU, and implement your own algorithms. At the end of the workshop, you'll have access to additional resources to create your own GPU-accelerated applications.

Learning Objectives

At the conclusion of the workshop, you'll have an understanding of the fundamental concepts and techniques for accelerating C++ code with CUDA and be able to:

- ▶ Write and compile code that runs on the GPU
- ▶ Optimize memory migration between CPU and GPU
- ▶ Leverage powerful parallel algorithms that simplify adding GPU acceleration to your code
- ▶ Implement your own parallel algorithms by directly programming GPUs with CUDA kernels
- ▶ Utilize concurrent CUDA streams to overlap memory traffic with compute
- ▶ Know where, when, and how to best add CUDA acceleration to existing CPU-only applications

Topics Covered

Technologies: NVIDIA Nsight™ Systems

Certificate: Upon successful completion of the assessment, participants will receive an NVIDIA certificate to recognize their subject matter competency and support professional career growth.

Hardware Requirements: Desktop or laptop computer capable of running the latest version of Chrome or Firefox. Each participant will be provided with dedicated access to a fully configured, GPU-accelerated server in the cloud.

Course Outline

Introduction

- Meet the instructor.
- Create an account at courses.nvidia.com/join

CUDA Made Easy: Accelerating Applications with Parallel Algorithms

- To make your first steps in GPU programming as easy as possible, this lab teaches you how to leverage powerful parallel algorithms that make GPU acceleration of your code as easy as changing a few lines of code. While doing so, you'll learn fundamental concepts such as execution space and memory space, parallelism, heterogeneous computing, and kernel fusion. These concepts will serve as a foundation for your advancement in accelerated computing. By the time you complete this lab, you will be able to: Write, compile, and run GPU code
- Refactor standard algorithms to execute on GPU
- Extend standard algorithms to fit your unique use cases

Break (60 mins)

Course Outline

Unlocking the GPU's Full Potential: Harnessing Asynchrony with CUDA Streams

- In the previous lab, you learned how to use parallel algorithms. However, But the concept of parallelism is not sufficient for accelerating your applications. To fully utilize GPUs, this lab will teach you another fundamental concept: asynchrony. In this lab, you'll learn how and when to leverage asynchrony. You'll use Nsight Systems to distinguish synchronous and asynchronous algorithms and identify performance bottlenecks. By the time you complete this lab, you will be able to: Use CUDA streams to overlap execution and memory transfers
- Use CUDA events for asynchronous dependency management
- Profile CUDA code with NVIDIA Nsight Systems

Break (15 mins)

Course Outline

Implementing New Algorithms with CUDA Kernels

- Previous labs equipped you with necessary understanding of how using standard parallel algorithms can provide both convenient and speed-of-light GPU acceleration. However, sometimes your unique use cases are not covered by accelerated libraries. In this lab, you'll learn the CUDA SIMT programming model to program the GPU directly using CUDA kernels. Besides that, this lab will cover utilities provided by the CUDA ecosystem to facilitate development of custom CUDA kernels. By the time you complete this lab, you will be able to: Write and launch custom CUDA kernels
- Control thread hierarchy
- Leverage shared memory
- Use cooperative algorithms

Final Review

- Review key takeaways and wrap up questions.
- Complete the assessment to earn a certificate.
- Take the workshop survey.