AI CERTs™

AI+ Architect ™ (5 Days)

Program Detailed Curriculum

Executive Summary

The AI+ Architect certification offers comprehensive training in advanced neural network techniques and architectures. It covers the fundamentals of neural networks, optimization strategies, and specialized architectures for natural language processing (NLP) and computer vision. Participants will learn about model evaluation, performance metrics, and the infrastructure required for AI deployment. The course emphasizes ethical considerations and responsible AI design, alongside exploring cutting-edge generative AI models and research-based AI design methodologies. A capstone project and course review consolidate learning, ensuring participants can apply their skills effectively in real-world scenarios. This certification equips learners with the knowledge and practical experience to excel in AI architecture and development.

Course Prerequisites

- A foundational knowledge on neural networks, including their optimization and architecture for applications.
- Ability to evaluate models using various performance metrics to ensure accuracy and reliability.
- Willingness to know about AI infrastructure and deployment processes to implement and maintain AI systems effectively.

Module 1

Fundamentals of Neural Networks

1.1 Introduction to Neural Networks

- **Basic Concepts of Neural Networks:** This unit discusses basic neural network components, including nodes (neurons), layers, and connections (synapses). It also explores the role of activation functions and how they impact the behavior of the network.
- **Types of Neural Networks:** This unit explores Feedforward Neural Networks (FNNs), focusing on the one-way data flow from input to output. It also discusses Convolutional Neural Networks (CNNs), outlining key concepts like convolutional layers and pooling layers used in image processing.
- **Limitations of Neural Networks:** Neural networks require substantial computational resources and large datasets for effective training. They can suffer from overfitting, making them less generalizable.
- **Applications of Neural Networks:** Neural networks are widely used in image and speech recognition, natural language processing, autonomous vehicles, medical diagnosis, and financial forecasting.

1.2 Neural Network Architecture

- **Architecture Components:** This unit examines essential architecture elements like weights, biases, activation functions, loss functions, and gradients. It also explores common activation functions such as ReLU, Sigmoid, and Tanh, and their roles in non-linear learning.

- **Building Neural Networks:** This unit discusses the process of building a simple neural network, addressing key considerations like layer arrangement, node count, and activation function choices.
- **Common Design Patterns:** This unit highlights popular design patterns for neural networks, such as ResNet, DenseNet, and U-Net. It examines where these patterns are used, with examples like image classification with ResNet or image segmentation with U-Net.

1.3 Hands-on: Implement a Basic Neural Network

- **Task Description:** This unit involves implementing a simple neural network for a basic task, such as image classification on the MNIST dataset or text processing for sentiment analysis.
- **Implementation Steps:** It guides participants through the setup process, explaining how to construct, train, and evaluate a basic neural network. It discusses common evaluation metrics, such as accuracy, loss, and confusion matrix.

Module 2

Neural Network Optimization

2.1 Hyperparameter Tuning

- **Importance of Hyperparameters:** This unit describes the key hyperparameters that influence neural network performance, including learning rate, batch size, number of layers, and number of epochs. It discusses how these impact model performance and training times.
- **Tuning Techniques:** This unit explores different hyperparameter tuning methods, including grid search and random search, highlighting their pros and cons. It also introduces advanced tuning approaches like Bayesian optimization and genetic algorithms.

2.2 Optimization Algorithms

- **Types of Optimization Algorithms:** This unit discusses popular optimization algorithms like Stochastic Gradient Descent (SGD), Adam, and RMSprop, explaining their key characteristics, strengths, and weaknesses.
- **Choosing the Right Algorithm:** This unit explores factors to consider when selecting an optimization algorithm, such as convergence rate, memory requirements, and stability. It provides examples where certain algorithms are preferred, like using Adam for faster convergence or SGD for larger datasets.

2.3 Regularization Techniques

- **Preventing Overfitting:** This unit discusses the concept of overfitting and how it affects neural network generalization. It describes common regularization techniques like dropout, L1/L2 regularization, and early stopping.
- **Other Methods for Model Robustness:** This unit explores additional methods to improve model robustness, such as data augmentation, ensembling, and model checkpoints. It describes practical steps to implement these techniques in a neural network training pipeline.

2.4 Hands-on: Hyperparameter Tuning and Optimization

- **Task Description:** This unit involves experimenting with hyperparameter tuning and optimization algorithms to improve a basic neural network's performance.
- **Practical Activities:** It includes hands-on exercises focused on tuning key hyperparameters (e.g., learning rate, batch size) and trying different optimization algorithms.
- **Evaluation and Analysis:** It guides participants in evaluating the impact of tuning and optimization on network performance and suggests strategies for further improvement.

Module 3

Neural Network Architectures for NLP

3.1 Key NLP Concepts

- **NLP Fundamentals:** This unit discusses natural language processing (NLP) basics, focusing on typical tasks such as text classification, sentiment analysis, named entity recognition, and language translation.
- **Tokenization and Embedding:** This unit describes tokenization techniques, like word-based and subword-based tokenization, and how they prepare text data for processing in neural networks. It explores word embeddings, discussing methods like Word2Vec, GloVe, and FastText. The unit also highlights how embeddings represent words in dense vector forms, allowing AI models to understand semantic relationships.

3.2 NLP-Specific Architectures

- **RNNs and LSTMs:** This unit explores Recurrent Neural Networks (RNNs) and their applications in NLP tasks. It examines how RNNs handle sequential data and maintain context across sequences. It discusses Long Short-Term Memory (LSTMs) and Gated Recurrent Units (GRUs), explaining how these architectures address the vanishing gradient problem in RNNs and improve learning over long sequences.
- **Transformer-Based Architectures:** This unit delves into Transformer-based models, explaining their attention mechanisms and architectural features. It discusses models like BERT, GPT, and TransformerXL, emphasizing their impact on NLP tasks.

3.3 Hands-on: Implementing an NLP Model

- **Task Description:** This unit involves implementing a Transformer-based NLP model for a simple text-based task, such as text classification or sentiment analysis.
- **Implementation Steps:** It guides participants through setting up the environment, tokenization, and building a basic NLP model. It discusses how to train the model, evaluate its performance, and understand key metrics in NLP.
- **Practical Exercises:** Participants build and train the model, experimenting with hyperparameters and applying different Transformer-based architectures.

Module 4

Neural Network Architectures for Computer Vision

4.1 Key Computer Vision Concepts

- **Computer Vision Fundamentals:** This unit introduces common computer vision tasks, like image classification, object detection, and image segmentation. It explains why neural networks are well-suited for these tasks due to their ability to learn complex patterns from visual data.
- **Convolutional Neural Networks (CNNs):** This unit discusses Convolutional Neural Networks (CNNs), highlighting key components such as convolutional layers, pooling layers, and fully connected layers. It describes how these components work together to process and understand images.

4.2 Computer Vision-Specific Architectures

- **Specialized Architectures for Computer Vision:** This unit examines popular computer vision architectures like ResNet, DenseNet, and MobileNet. It discusses their unique features and explains how they achieve high performance in various tasks.
- **Techniques for Object Detection and Image Segmentation:** This unit discusses methods for object detection, including algorithms like Faster R-CNN, YOLO, and SSD. It explains how these methods detect and localize objects within images. It also explores techniques for image segmentation, including U-Net and Mask R-CNN, discussing how these architectures segment images into distinct regions.

4.3 Hands-on: Building a Computer Vision Model

- **Task Description:** This unit involves building a simple CNN for an image classification task, such as recognizing handwritten digits (MNIST) or classifying images in CIFAR-10.
- **Implementation Steps:** It guides participants through setting up the environment, building the CNN, and training it on a computer vision dataset. It discusses evaluation techniques for computer vision, like accuracy and confusion matrices.
- **Additional Exercises:** Participants explore object detection and image segmentation techniques, using popular architectures like YOLO or Mask R-CNN.

Module 5

Model Evaluation and Performance Metrics

5.1 Model Evaluation Techniques

- **Evaluation Metrics for AI Models:** This unit describes common evaluation metrics for AI models, including accuracy, precision, recall, and F1-score. It discusses when to use each metric, depending on the context and type of task.
- **Cross-Validation and Model Selection:** This unit explains the importance of cross-validation in ensuring a model's robustness and avoiding overfitting. It discusses various cross-validation techniques, like k-fold and stratified cross-validation. It explores strategies for model selection, including the use of validation sets and the concept of model ensembles.

5.2 Improving Model Performance

- **Addressing Overfitting and Underfitting:** This unit examines common causes of overfitting and underfitting in AI models. It discusses strategies to mitigate these issues, such as data augmentation, dropout, L1/L2 regularization, and early stopping.
- **Techniques for Performance Optimization:** This unit discusses methods to improve model performance and efficiency. It explores techniques like hyperparameter tuning, batch normalization, and the use of more efficient architectures.

5.3 Hands-on: Evaluating and Optimizing AI Models

- **Task Description:** This unit involves evaluating the performance of a given AI model using various metrics. Participants apply optimization techniques to improve its accuracy and robustness.
- **Implementation Steps:** It guides participants through model evaluation, discussing how to interpret key metrics. It demonstrates performance optimization methods, allowing participants to experiment with different strategies to enhance their models.

Module 6

AI Infrastructure and Deployment

6.1 Infrastructure for AI Development

- **Hardware Requirements:** This unit discusses the hardware requirements for AI development, focusing on the importance of GPUs and TPUs in accelerating neural network training. It explores the benefits and trade-offs of different hardware options.
- **Cloud-Based AI Services:** This unit provides an overview of popular cloud-based AI platforms, such as Google Cloud AI, AWS, and Microsoft Azure. It discusses their features, cost structures, and how they facilitate AI development and deployment.

6.2 Deployment Strategies

- **Model Deployment Techniques:** This unit explores various strategies for deploying AI models in production environments. It discusses deployment options like containerization (e.g., Docker), serverless computing, and edge deployment.
- **Monitoring and Maintenance:** This unit examines techniques for monitoring deployed AI models, discussing how to ensure they perform consistently over time. It explores the use of monitoring tools, retraining strategies, and mechanisms for detecting model drift or performance degradation.

6.3 Hands-on: Deploying an AI Model

- **Task Description:** This unit involves deploying a simple AI model on a cloud-based platform and monitoring its performance over time.
- **Implementation Steps:** It guides participants through the deployment process, including setting up a cloud-based environment, deploying the model, and implementing basic monitoring techniques.
- **Practical Exercises:** Participants deploy a model using a cloud platform and experiment with various deployment strategies and monitoring tools.

Module 7

AI Ethics and Responsible AI Design

7.1 Ethical Considerations in AI

- **Bias, Fairness, and Accountability:** This unit discusses the potential for bias in AI models and explores strategies to ensure fairness and accountability. It examines how biases can arise from data collection, model design, or other factors, and the impacts these biases can have on various groups.
- **Explainability and Transparency:** This unit describes the importance of explainable AI and discusses various techniques to improve model transparency. It explains why explainability is critical for building trust and accountability in AI systems.

7.2 Best Practices for Responsible AI Design

- **Ensuring Ethical AI Development:** This unit discusses guidelines for responsible AI development and strategies to mitigate ethical risks. It addresses ethical concerns in AI design, like privacy, data protection, and human impact. It explores principles for responsible AI design, such as fairness, accountability, transparency, and safety. It also discusses frameworks for ethical AI development from organizations like IEEE and AI4People.
- **Case Studies in AI Ethics:** This unit examines real-world case studies illustrating ethical issues in AI development. It explores scenarios where AI systems caused harm or controversy due to biases or lack of transparency, and how these issues were addressed. It discusses lessons learned from these case studies and how they can inform responsible AI design and deployment.

7.3 Hands-on: Analyzing Ethical Considerations in AI

- **Task Description:** This unit involves analyzing a given AI model or system for ethical considerations and proposing solutions to address potential biases or fairness issues.
- **Practical Exercises:** Participants identify potential biases in an AI system, discuss their impact, and propose strategies for mitigating them. This could involve analyzing a pre-trained model, examining its outputs, and suggesting improvements for fairness and transparency.

Module 8

Generative AI Models

8.1 Overview of Generative AI Models

- **Generative Adversarial Networks (GANs):** This unit discusses the basic architecture of GANs, explaining the roles of the generator and discriminator and how they work together to create realistic synthetic data. It explores various types of GANs, including CycleGAN, StyleGAN, and DCGAN.
- **Transformer-based Models:** This unit introduces the Transformer architecture and its key components, such as multi-head attention and position encoding. It explores the application of Generative Pre-Trained (GPT) models for text generation and natural language processing tasks.

8.2 Generative AI Applications in Various Domains

- **GANs for Visual and Multimedia Artifacts:** This unit discusses use cases for GANs in creating realistic images, videos, and artwork. It explores how GANs are used in industries like fashion, film, and video game design, and the creative possibilities they offer.
- **Transformer-based Models for Text Generation:** This unit explores applications of GPT models in generating textual content from diverse sources, including website articles, press releases, and whitepapers. It discusses the potential for automation and creative uses in content generation.

8.3 Hands-on: Exploring Generative AI Models

- **Building a Simple GAN:** This unit involves constructing a basic GAN from scratch to generate synthetic images from a simple dataset (e.g., MNIST, CIFAR-10). It introduces popular frameworks and libraries for GAN development, such as TensorFlow and PyTorch.
- **Text Generation with GPT:** This unit involves using a pre-trained GPT model to generate text based on given prompts. Participants experiment with text generation to create different types of content, such as stories, articles, or dialogues.
- **Style Transfer and Text-to-Image:** This unit involves implementing a basic style transfer model with GANs, allowing participants to blend artistic styles with image content. It explores popular style transfer techniques and their applications in art and design.

Module 9

Research-Based AI Design

9.1 AI Research Techniques

- **Research Methodologies in AI:** This unit discusses common research methodologies in AI architecture design, including experimental studies, case studies, and theoretical analyses. It explains the process of conducting AI research and how to approach experimentation and hypothesis testing.
- **Interpreting Research Papers:** This unit explores techniques for reading and understanding AI research papers. It discusses how to extract key information from research papers, including the introduction, methodology, results, and discussion sections.

9.2 Cutting-Edge AI Design

- **Exploring Recent AI Research:** This unit explores recent AI research in the context of architecture design. It reviews recent trends and advancements in AI, discussing their implications for AI design and development. It discusses new developments in neural network architectures, optimization techniques, and applications of AI in various domains.

- **Applying Research to AI Design:** This unit discusses how research informs the design and evolution of AI architectures. It explores practical applications of research findings and how to apply them in real-world AI projects. It encourages participants to stay updated with the latest AI research and to apply new concepts and techniques to their AI projects.

9.3 Hands-on: Analyzing AI Research Papers

- **Task Description:** This unit involves analyzing recent research papers on AI architecture design and discussing their findings in a group setting. Participants work in teams to explore the implications of the research and suggest applications for AI design.
- **Practical Exercises:** Participants select a recent AI research paper, analyze its methodology and findings, and present their interpretations and conclusions to the class. The exercise encourages discussion and collaborative analysis of AI research.

Module 10

Capstone Project and Course Review

10.1 Capstone Project Presentation

- **Presentation of Capstone Projects:** This unit involves participants presenting their capstone projects, demonstrating their understanding of AI architecture design and their ability to apply concepts from the course. Participants explain their projects, describe their approach, and discuss the results.

10.2 Course Review and Future Directions

- **Comprehensive Course Review**: This unit reviews key concepts covered throughout the course. It summarizes the critical aspects of AI architecture design, optimization, computer vision, NLP, generative AI, ethics, and research.
- **Exploring Future Directions in AI:** This unit discusses emerging trends and the future of AI architecture design. It explores new directions in AI research, technological advancements, and their potential impact on AI development.

10.3 Hands-on: Capstone Project Development

- **Task Description:** This unit involves participants working on their capstone project, incorporating the knowledge and skills gained throughout the course. It provides an opportunity for participants to apply concepts and techniques from previous modules and receive feedback from peers and instructors.
- **Practical Exercises:** Participants develop their capstone projects, with guidance from instructors, and work on refining their presentations. They receive feedback and suggestions for improvement, allowing them to finalize their projects for presentation.