Knight Foundation School of Computing and Information Sciences

Course Title: Introduction to Deep Learning **Date:** 6/2/2024

Course Number: CAI 4203

Number of Credits: 3

Subject Area: Artificial Intelligence
Catalog Description: This course introduces the fundamentals of deep learning, covering feed forward neural networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and other advanced models.
Textbooks: " Introduction to Deep Learning" (2018) Sandro Skansi ISBN: 978-3-319-73004-2
References:" Introduction to Deep Learning" (2018) Sandro Skansi ISBN: 978-3-319-73004-2
Prerequisites Courses: (STA 3033 or STA 2023 or STA 2122 or STA 4322) and (COP 3465 or COP 3530)
Corequisite Courses: None

Type: Core Course for BS in Data Science; Elective for CS (Applications) and CY Majors.

Prerequisites Topics:

1. Basic statistics and probability concepts.
2. Data structures.
3. Programming languages.

Course Outcomes:

1. Identify the role of deep learning within the fields of AI and machine learning.
2. Describe the basic components of a neural network and how they are used to model complex relationships between input and output data.
3. Implement and train basic neural networks using popular deep learning frameworks such as TensorFlow.
4. Analyze and optimize deep neural network design and hyperparameters to enhance performance.
5. Solve a variety of real-world problems by applying deep learning techniques.
6. Evaluate the principles underlying advanced neural network architectures like convolutional neural networks (CNNs) and recurrent neural networks (RNNs).
7. Critically assess current research trends and developments in deep learning, and assess their implications for the future of artificial intelligence.
8. Collaborate with peers on hands-on projects and assignments to develop practical experience with deep learning concepts and techniques.

Knight Foundation School of Computing and Information Sciences
CAI 4203 Introduction to Deep Learning

Association between Student Outcomes and Course Outcomes

BS in Computing: Student Outcomes	Course Outcomes
1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.	5, 6, 7, 8
2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.	5, 6, 7, 8
3) Communicate effectively in a variety of professional contexts.	8
4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	8
5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.	8

Program Specific Student Outcomes

6) Apply theory, techniques, and tools throughout the data science lifecycle and employ the resulting knowledge to satisfy stakeholders' needs. [DS]	5, 6, 7, 8
6) Apply computer science theory and software development fundamentals to produce computing-based solutions. [CS]	1, 2, 3, 4
6) Apply security principles and practices to maintain operations in the presence of risks and threats. [CY]	8

Assessment Plan for the Course and how Data in the Course are used to assess Student Outcomes

Student and Instructor Course Outcome Surveys are administered at the conclusion of each offering, and are evaluated as described in the School's Assessment Plan: https://abet.cis.fiu.edu/

Knight Foundation School of Computing and Information Sciences
CAI 4203 Introduction to Deep Learning

Outline

Topic	No. of Lecture Hours	Course Outcomes

• Introduction to Supervised Learning & classification
  ○ Introduction to AI and machine learning
  ○ Supervised learning methods
  Classification
  Regression
	3	1, 2

• Multilayer Perceptron (MLP)
  ○ The Perceptron
  ○ Feedforward neural network
  ○ Backpropagation
  ○ Activation functions
  ○ Design and Train an MLP
	9	1, 2, 3

• Deep learning
  ○ What is deep learning?
  ○ History & Applications
  ○ Introduction to deep learning in Python and Keras
	3	1, 2, 3, 7

• Optimization techniques
  ○ Stochastic gradient decent
  ○ Learning rate schedules
  ○ Regularization
  ○ Overfitting avoidance methods
	6	2, 3, 4

• Convolutional Neural Network (CNN)
  ○ Motivation & advantages
  ○ Convolutional & Pooling layers
  ○ Design & train a CNN for image analysis
	6	5, 6, 7, 8

• Autoencoder ○ Motivation & application	3	3, 5, 8

• Recurrent Neural Network (RNN)
  ○ Motivation & Advantage
  ○ Basic RNN & their limitation
  ○ LSTM networks
	3	5, 6, 7, 8

• Advanced topics
  ○ Transfer learning
  ○ Ensemble learning
  ○ Shortcut learning
  ○ Generative adversarial networks & Deep fake
  ○ Deep reinforcement learning
  ○ Transformers
	6	5, 6, 7, 8

Knight Foundation School of Computing and Information Sciences
CAI 4203 Introduction to Deep Learning

Performance Measures for Evaluation

All assignments are assigned through the Canvas course site.

- Late Work Policy: Late submissions will receive a 10% automatic deduction for every day past the due date. The deduction will continue until 3 days past the due date. The assignment is automatically closed on the third day at midnight. The late work policy is non-negotiable.
- Policy Regarding Contesting a Grade: You will have one week (seven calendar days) following the posting of a grade to contest the grade. If the grade is not contested by 5 pm (Eastern) on the seventh day, then the grade posted will stand as final. If the seventh day falls on a holiday, then you will have until the next business day. For purposes of contesting a grade, an email to the email address listed above with a subject line of CONTESTING MY GRADE and a body with your name, the course, the assignment, and a brief explanation of why you are contesting the grade shall constitute notice of your intention to contest a grade.

Assignment	**Total Points**	**Percentage of Final Grade**
Homework (5)	100 each	15%
Exams (2)	100 each	35%
Projects (3)	100 each	25%
Final Project	100	25%
	TOTAL	100%

Letter Grade Distribution Table

Letter	**Range%**		**Letter**	**Range%**		**Letter**	**Range%**
A	95 or above		B	83 - 86		C	70 - 76
A-	90 - 94		B-	80 - 82		D	60 - 69
B+	87 - 89		C+	77 - 79		F	59 or less

Knight Foundation School of Computing and Information Sciences
CAI 4203 Introduction to Deep Learning

Description of Possible Projects

Project 1: Implementing a Neural Network from Scratch

Criteria	Excellent (100)	Good (80)	Average (60)	Below Average (40)	Poor (20)
Neural Network Implementation (30%)	Code implements a neural network with correct forward and backward passes, weight initialization, and activation functions.	Code implements a neural network with mostly correct forward and backward passes, weight initialization, and activation functions.	Code implements a neural network with some errors in forward and backward passes, weight initialization, or activation functions.	Code implements a neural network with significant errors in forward and backward passes, weight initialization, or activation functions.	Code does not implement a neural network or has major errors.
Training and Convergence (25%)	Network trains successfully and converges to a good solution with clear evidence of correct hyperparameter tuning.	Network trains successfully and converges to a reasonable solution with some hyperparameter tuning.	Network trains but has difficulties converging or suboptimal hyperparameter choices.	Network struggles to train or does not converge, with poor hyperparameter choices.	Network does not train at all.
Performance Metrics (20%)	Achieves excellent performance metrics (e.g., accuracy, loss) on the synthetic dataset with insightful analysis.	Achieves good performance metrics on the synthetic dataset with adequate analysis.	Achieves acceptable performance metrics on the synthetic dataset with limited analysis.	Achieves poor performance metrics on the synthetic dataset with minimal analysis.	Fails to achieve any meaningful performance metrics.
Code Quality and Readability (25%)	Code is well-structured, well-documented, and follows best practices, making it easy to understand.	Code is mostly well-structured and documented but may lack some clarity.	Code is organized but may lack proper documentation and readability.	Code lacks structure, documentation, and readability, making it challenging to understand.	Code is disorganized and entirely lacking documentation

Knight Foundation School of Computing and Information Sciences
CAI 4203 Introduction to Deep Learning

Project 2: Convolutional Neural Networks (CNNs) for Image Classification

Criteria	**Excellent (100)**	**Good (80)**	**Average (60)**	**Below Average (40)**	**Poor (20)**
CNN Architecture (30%)	Implements a CNN with a well-structured architecture, appropriate layers, and effective use of pre-trained models for transfer learning if applicable.	Implements a CNN with a sound architecture, suitable layers, and some use of pre-trained models for transfer learning if applicable.	Implements a CNN with a basic architecture, limited layer choice, or minimal use of pre-trained models for transfer learning if applicable.	Implements a CNN with a poor architecture, inadequate layer choice, or no use of pre-trained models for transfer learning if applicable.	Does not implement a CNN or has a severely flawed architecture.
Data Preprocessing (25%)	Preprocesses the image data effectively with data augmentation techniques where necessary, leading to improved model performance.	Preprocesses the image data adequately, with some data augmentation, contributing to reasonable model performance.	Performs basic data preprocessing with limited augmentation, resulting in modest model performance.	Performs minimal data preprocessing and lacks data augmentation, leading to subpar model performance.	Neglects data preprocessing and augmentation, severely impacting model performance.
Model Training and Evaluation (30%)	Successfully trains the CNN, achieves high accuracy, and provides detailed evaluation metrics and insightful analysis.	Trains the CNN with reasonable accuracy, offers evaluation metrics, and provides some analysis.	Manages to train the CNN with moderate accuracy, but evaluation metrics and analysis are limited.	Struggles to train the CNN with low accuracy, and evaluation metrics and analysis are lacking.	Fails to train the CNN effectively and does not provide any evaluation or analysis.
Code Quality and Readability (15%)	Code is well-structured, well-documented, follows best practices, and is easy to understand.	Code is mostly well-structured and documented but may lack some clarity.	Code is organized but may lack proper documentation and readability.	Code lacks structure, documentation, and readability, making it challenging to understand.	Code is disorganized and entirely lacking documentation.

Knight Foundation School of Computing and Information Sciences
CAI 4203 Introduction to Deep Learning

Project 3: Sequence-to-Sequence Models for Natural Language Processing

Criteria	Excellent (100)	Good (80)	Average (60)	Below Average (40)	Poor (20)
Model Architecture (30%)	Implements a sequence-to-sequence model (RNN or Transformer) with a well-structured architecture, correct attention mechanisms, and effective embeddings.	Implements a sequence-to-sequence model with a sound architecture, attention mechanisms, and reasonable embeddings.	Implements a sequence-to-sequence model with a basic architecture, limited attention mechanisms, or suboptimal embeddings.	Implements a sequence-to-sequence model with a poor architecture, inadequate attention mechanisms, or incorrect embeddings.	Does not implement a sequence-to-sequence model or has a severely flawed architecture.
Data Preprocessing (25%)	Preprocesses the text data effectively, handles tokenization, padding, and attention masks properly, leading to improved model performance.	Preprocesses the text data adequately, handles tokenization, padding, and attention masks, contributing to reasonable model performance.	Performs basic data preprocessing with some issues in tokenization, padding, or attention masks, resulting in modest model performance.	Performs minimal data preprocessing and struggles with tokenization, padding, or attention masks, leading to subpar model performance.	Neglects data preprocessing, tokenization, padding, or attention masks, severely impacting model performance.
Model Training and Evaluation (30%)	Successfully trains the sequence-to-sequence model, achieves high performance (e.g., BLEU score, ROUGE score), and provides detailed evaluation metrics and insightful analysis.	Trains the sequence-to-sequence model with reasonable performance, offers evaluation metrics, and provides some analysis.	Manages to train the sequence-to-sequence model with moderate performance, but evaluation metrics and analysis are limited.	Struggles to train the sequence-to-sequence model with poor performance, and evaluation metrics and analysis are lacking.	Fails to train the sequence-to-sequence model effectively and does not provide any evaluation or analysis.
Code Quality and Readability (15%)	Code is well-structured, well-documented, follows best practices, and is easy to understand.	Code is mostly well-structured and documented but may lack some clarity.	Code is organized but may lack proper documentation and readability.	Code lacks structure, documentation, and readability, making it challenging to understand.	Code is disorganized and entirely lacking documentation.