C S 12B: DEEP LEARNING

Foothill College Course Outline of Record

Heading	Value
Effective Term:	Winter 2026
Units:	4.5
Hours:	4 lecture, 2 laboratory per week (72 total per quarter)
Prerequisite:	C S 12A.
Degree & Credit Status:	Degree-Applicable Credit Course
Foothill GE:	Non-GE
Transferable:	CSU/UC
Grade Type:	Letter Grade (Request for Pass/No Pass)
Repeatability:	Not Repeatable

Student Learning Outcomes

- Explain different optimization strategies and identify appropriate methods for specific tasks
- Recognize the potential for bias and analyze the ethical implications of Deep Learning applications in different domains.
- Understand the need for explainability in certain domains and the challenges of interpreting increasingly complex networks
- Use Python packages to train and evaluate deep learning models, including Feedforward, Convolutional, and Recurrent Neural Networks

Description

This course offers an introduction to deep learning theories, principles, and practices. Students will explore neural networks, including perceptrons, gradient descent, and multilayer perceptrons, as well as advanced topics like convolutional neural networks (CNNs), recurrent neural networks (RNNs), generative adversarial networks (GANs), variational autoencoders (VAEs), and attention mechanisms. By the end of the course, students will be proficient in implementing and training neural networks using frameworks like TensorFlow, Keras, scikit-learn, and PyTorch, and will be able to critically evaluate and improve deep learning models.

Course Objectives

The student will be able to:

1. Demonstrate foundational knowledge of machine learning principles
2. Apply selected concepts from linear algebra and calculus to deep learning
3. Create, train, and evaluate a feed-forward network using publicly available packages
4. Identify techniques for optimizing models
5. Create, train, and evaluate a recurrent neural network using publicly available packages
6. Create, train, and evaluate a convolutional neural network using publicly available packages
7. Explain the concept and use of generative models
8. Articulate paths for future exploration of deep learning
9. Recognize deep learning as a tool that can reduce or amplify problems in society

Course Content

1. Machine learning foundations
   a. Training and evaluation
   b. Bias-variance tradeoff
   c. Logistic regression
   d. Perceptrons
2. Concepts from linear algebra and calculus
   a. Vector and matrix operations
   b. Gradient and partial derivatives
   c. Chain rule for backpropagation
3. Feed-forward networks
   a. Architectural components
   b. Forward propagation
   c. Loss and cost functions
   d. Backpropagation and gradient descent
   e. Learning rate and convergence/divergence
4. Optimizations
   a. Hyperparameter tuning
   b. Stochastic gradient descent and mini-batching
   c. Regularization and dropout
   d. Weight initialization
   e. Batch normalization
5. Recurrent neural networks
   a. Types of sequential data
   b. Simple recurrent architecture
   c. Vanishing/exploding gradients
   d. Long short-term memory networks
   e. Gated recurrent units
   f. Bidirectional RNNs
6. Convolutional neural networks (CNNs)
   a. Computer vision applications
   b. Convolutional filters
   c. Activation maps
   d. Pooling layers
7. Generative models
   a. Discriminative vs. generative tasks
   b. Generative adversarial networks
   c. Variational autoencoders
8. Additional topics for exploration
   a. Transfer learning
   b. Deep reinforcement learning
   c. Self-supervised learning
   d. End-to-end learning
9. Ethics and explainability

Lab Content

1. Environment setup
   a. Navigating Jupyter notebooks or chosen environment
   b. Running code cells, troubleshooting common errors
   c. Installing and importing DL frameworks (e.g., PyTorch or TensorFlow)

2. Implementing basic neural networks
    a. Building a single-layer perceptron and training on a small dataset
    b. Adding hidden layers to form an MLP
    c. Evaluating performance with basic metrics
3. Hyperparameter experiments
    a. Adjusting learning rate, batch size, and hidden units
    b. Tracking accuracy and loss over epochs
    c. Using a validation set for early stopping
4. Advanced training techniques
    a. Adding batch normalization and experimenting with different optimizers
    b. Trying different activation functions (ReLU, LeakyReLU)
    c. Comparing training stability and convergence speed
5. Working with CNNs
    a. Implementing a simple CNN for image classification
    b. Inspecting feature maps
6. Transfer learning
    a. Loading a pre-trained model
    b. Fine-tuning on a custom dataset
    c. Observing improvements in training time and accuracy
7. Sequence models
    a. Training a simple RNN or LSTM for time-series forecasting
    b. Adjusting sequence length and analyzing results
    c. Comparing RNN outputs to feed-forward network predictions
8. Exploring transformers
    a. Running inference with a pre-trained transformer model for classification
    b. Observing improvements over RNN-based models
9. Generative modeling
    a. Implementing a simple GAN
    b. Examining generated samples and adjusting training steps
    c. Noting stability issues in GAN training
10. Data pipeline management
    a. Demonstrating shuffling, batching, caching, and prefetching with a chosen framework
    b. Handling categorical or structured data (conceptually with code snippets)
11. Responsible artificial intelligence in practice
    a. Checking for bias in a dataset (e.g., class imbalance)
    b. Interpreting and explaining a model's behavior
    c. Discussing potential mitigations to bias and documenting findings
12. Project integration and review
    a. Combining steps: data preprocessing --> model building --> training --> evaluation
    b. Implementing a chosen project (e.g., image classification with a pre-trained model)
    c. Reporting results, plotting training curves, showing example predictions

Special Facilities and/or Equipment

1. The college will provide access to a computer laboratory with Python and an IDE installed, with sufficient privileges to allow students to install Python packages.
2. The college will provide a website or course management system with an assignment posting component (through which all lab assignments are to be submitted) and a forum component (where students can discuss course material and receive help from the instructor). This applies to all sections, including on-campus (i.e., face-to-face) offerings.
3. When taught online, the college will provide a fully functional and maintained course management system through which the instructor and students can interact.
4. When taught online, students must have currently existing email accounts and ongoing access to computers with internet capabilities.

Method(s) of Evaluation

Methods of Evaluation may include but are not limited to the following:

Tests and quizzes
Lab notebook
Written laboratory assignments which include source code, sample runs, and documentation
Reflective papers
Final examination or project

Method(s) of Instruction

Methods of Instruction may include but are not limited to the following:

Instructor-authored lectures which include mathematical foundations, theoretical motivation, and coding implementation of deep learning algorithms
Detailed review of assignments which includes model solutions and specific comments on the student submissions
Discussion which engages students and instructor in an ongoing dialog about deep learning
Instructor-authored labs that rigorously demonstrate a student's ability to implement deep learning models

Representative Text(s) and Other Materials

Chollet, François. Deep Learning with Python, 2nd ed.. 2021.

Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed.. 2022.

Raschka, Sebastian, Yuxi (Hayden) Liu, and Vahid Mirjalili. Machine Learning with PyTorch and Scikit-Learn. 2022.

Types and/or Examples of Required Reading, Writing, and Outside of Class Assignments

1. Reading
    a. Textbook assigned reading averaging 30 pages per week
    b. Reading the supplied handouts and modules averaging 10 pages per week
    c. Reading online resources as directed by instructor though links pertinent to programming
    d. Reading library and reference material directed by instructor through course handouts
2. Writing
    a. Writing technical prose documentation that supports and describes the programs that are submitted for grades

Discipline(s)

Computer Science