

07-280 Learning Objectives

Spring 2026 - Midterm 2

Course Level Learning Outcomes

1. Implement and analyze existing algorithms, including well-studied methods for heuristic-based search, constraint satisfaction problems, classification, regression, and feature learning
2. Integrate multiple facets of practical AI and ML in a single system: problem formulation, data preprocessing, learning, regularization, and algorithm/model selection
3. Describe the formal properties of algorithms and models in AI and ML, and explain the practical implications of those results
4. Describe the trade-offs between computationally cheap, less efficient/accurate methods and computationally expensive, but more accurate methods, especially as they apply to heuristic implementation and sampling techniques
5. Compare and contrast different paradigms for learning (supervised, unsupervised, etc.)
6. Design experiments to evaluate and compare different machine learning techniques on real-world problems
7. Employ probability, statistics, calculus, linear algebra, and optimization in order to develop new predictive models or learning methods
8. Given a description of an AI/ML technique, analyze it to identify (1) the expressive power of the formalism; (2) the size and complexity of the search space; (3) the computational properties of the algorithm; (4) any guarantees (or lack thereof) regarding termination, convergence, correctness, accuracy, or generalization power
9. Select and apply an appropriate supervised learning algorithm for classification and regression problems (e.g., decision trees, linear regression, logistic regression, neural networks)

ML Fundamentals

1. Optimization for ML
 - a. Identify the relationships between the concepts of loss, risk, empirical risk, and empirical risk minimization as they relate to finding a hypothesis function/model in machine learning
 - b. Given a loss function and hypothesis function structure (e.g., linear regression structure), formulate the machine learning optimization problem
 - c. Apply (batch) gradient descent, stochastic gradient descent, and mini-batch gradient descent to optimize an objective function
 - d. Explain the issues with optimizing non-convex functions
2. Calculus
 - a. Derive the gradient of a differentiable, multi-variable objective function by applying knowledge of multivariable calculus, including:
 - i. Partial derivatives involving scalars, vectors, and matrices

- ii. Numerator and denominator layout of partial derivatives
 - iii. Multivariable chain rule
 - b. Derive matrix calculus properties by expanding linear algebra into scalar form as needed
3. Neural Networks
- a. Explain the biological motivations for a neural network
 - b. Combine simpler models (e.g. linear regression, binary logistic regression, multi-class logistic regression) as components to build up feed-forward neural network architectures
 - c. Explain the reasons why a neural network can model nonlinear decision boundaries for classification
 - d. Compare and contrast feature engineering with learning features
 - e. Identify (some of) the options available when designing the architecture of a neural network
 - f. Implement and train a feed-forward neural network from scratch, including training with the backpropagation algorithm
 - g. Instantiate an optimization method (e.g., SGD) when the parameters of a model are comprised of several matrices corresponding to different layers of a neural network

AI Alignment

1. Describe the goal of AI alignment and distinguish between different types of alignment (e.g., user goals vs. broader human values).
2. Identify and explain key challenges in aligning AI systems (e.g., reward hacking, distribution shift, difficulty of specification and evaluation).
3. Compare general approaches to improving alignment (e.g., post-training with feedback, guardrails, interpretability, improved reward design), including their strengths and limitations.

Building AlexNet

1. Neural Network Applications
 - a. Combine various layers and loss functions to create, train, and deploy task-specific neural network architectures, such as convolutional neural networks and autoencoders
 - b. Identify (some of) the options available when designing the architecture of a neural network
 - c. Utilize pre-trained parameters of a neural network to fine-tune training a new network such that it can be applied to a different but related task
 - d. Leverage functionality within a deep learning toolkit, such as PyTorch, to train and deploy a neural network for a specific task
2. Computer Vision
 - a. Describe the potential benefits of using convolutional layers rather than fully connected layers
 - b. Describe the parameters used in convolutional neural networks
 - c. Explain how pooling and kernel stride length can reduce the number of values as the network gets deeper

- d. Understand the reasons for using additional techniques in deep learning, including residual connections, normalization, and dropout
 - e. Discuss the impact of AlexNet on AI
- 3. Parallel Processing
 - a. Describe why GPUs can outperform CPUs for neural network training and inference
 - b. Discuss how training data and neural network models can be split across multiple GPUs.

MLE and Markov Chains

- 1. MLE
 - a. Recall probability basics, including but not limited to: discrete and continuous random variables, probability mass functions, probability density functions, events vs. random variables, expectation and variance, joint probability distributions, marginal probabilities, conditional probabilities, independence, and conditional independence
 - b. Describe common probability distributions such as Bernoulli, Categorical, Gaussian, etc.
 - c. State the principle of maximum likelihood estimation and explain what it tries to accomplish
 - d. State the principle of maximum a posteriori estimation and explain why we use it
 - e. Derive the MLE parameters of a simple model in closed form
 - f. Define maximum conditional likelihood estimation (MCL) and describe how it differs from MLE
 - g. Provide a probabilistic interpretation of linear regression
 - h. For linear regression, show that the parameters that minimize squared error are equivalent to those that maximize conditional likelihood
 - i. Explain the practical reasons why we work with the log of the likelihood
- 2. Markov Chains and N-grams
 - a. Express an n-gram language model as a markov chain probability distribution
 - b. Connect the frequency counts of N-grams with the solution to maximizing the log likelihood of the corpus

Building GPT2

- 1. Language models
 - a. Build a language model using word embeddings by formulating the objective using token encoding and decoding linear layers, deriving the gradient update equations, and training the model in a self-supervised manner
 - b. Adjust basic language models (n-grams and word embeddings) to use longer context than just a single token and discuss the impact of increased context size on model size, training data, and model performance.
 - c. Describe byte pair encoding tokenization as it compares to character- and word-level tokenization
 - d. Implement positional encoding to help language models differentiate between different orders of tokens in the input context.

- e. Given an input token context, generate the next token by sampling from the probabilities generated by a trained language model using additional techniques, including top-k and temperature sampling.
- f. Compare different techniques for basic language models, specifically n-grams, word embeddings, and attention
- g. Describe the motivation for using attention mechanisms rather than a uniform combination of input content embeddings
- h. Describe the motivation for using multiple attention heads rather than a single attention head
- i. Give various examples of the role of linear operations in GPT language models
- j. Identify the size of and difference between parameters and computed values (activations) in various components within transformer models, especially as the context size changes.
- k. Implement and train a transformer language model with multiple attention heads and multiple transformer blocks.

References

Built from 10-301/601 Learning objectives