Program Outline

Program Format and Modules

Each week, you will receive new course materials. The learning content of the week will be delivered through videos and interactives followed by assignments. The assignments will enable you to apply your knowledge of the concepts learned.

Throughout the course, you will be able to exchange ideas with your fellow classmates through the discussion boards on the learning platform.

Table of Contents

Module 0: Program Orientation

Key Activities

- Course Introduction
- Learning Platform Overview
- Introduce Yourself
- Course Agreement
- Capstone Assignment Overview

Module 1: Introduction and Universal Approximation

Learning Outcomes:

1. Describe significant impacts and successful use cases of neural networks in contemporary society.
2. Describe the structure and function of a perceptron.

3. Explain how the perceptron uses real-valued inputs for construction of decision boundaries.
4. Explain why multilayer perceptrons are considered to be universal function approximators.
5. Identify the basic properties of a deep neural network.
6. Define a neural network as a function.
7. Describe neural network optimization in terms of function minimization.
8. Define a gradient as a vector of partial derivatives.
9. Explain gradient descent as an approach to minimizing a function.
10. Identify the purpose of a decaying learning rate.

Key Activities

- Videos 1.1–1.8
- Interactives 1.1–1.2
- Quizzes 1.1–1.2
- Capstone Assignment Checkpoint 1

Back to top

Module 2: Training Multilayer Perceptrons

Learning Outcomes:

1. Compare and contrast popular activation functions.
2. Define vector activations as neurons that produce multiple outputs.
3. Explain output representations for binary and multiclass classification problems.
4. Describe the purpose and significance of the divergence term in the network loss function.
5. Set up variables to compute the derivative of a single training input.
6. Order the steps to calculate the output of the network from input (forward pass).
7. Explain the chain rule of calculus in the context of neural networks.
8. Order the steps of backpropagation.
9. Order the steps of the forward pass using vector notation.
10. Explain the chain rule for vector functions in the context of neural networks.
11. Order the steps of backpropagation using vector notation.
12. Summarize the forward and backward pass algorithms for neural networks.
13. Implement the forward pass, backward pass, and stochastic gradient descent for a multilayer perceptron.

Key Activities

- Videos 2.1–2.9
- Interactives 2.1–2.4
- Quizzes 2.1–2.3
- Programming Assignment 2.1: Multilayer Perceptron (Part A)
- Capstone Assignment Checkpoint 2

Back to top

Module 3: Stochastic Gradient Descent and Optimizers

Learning Outcomes:

1. Describe the relationship of learning rate and the ability of gradient descent to converge to a global minimum of a function.
2. Describe incremental updates (i.e., stochastic gradient descent) to learn network parameters.
3. Explain the necessary conditions for stochastic gradient descent to converge.
4. Describe methods to mitigate variance problems with training using stochastic gradient descent (minibatch descent).
5. Identify the resulting effect of using a moving average on gradient trajectory (momentum).
6. Describe higher order methods: RMSprop.
7. Introduce ADAM and summarize improvements to gradient descent in terms of their effects on convergence.
8. Explain covariate shifts and batch normalization.
9. Explain backpropagation in batch normalization.
10. Explain the impact of data underspecification and the need for regularization.
11. Describe dropout as a means of regularization.
12. Identify additional heuristics and describe overall problem setup.
13. Implement a neural network to classify the phonemes of an audio file.

Key Activities

- Videos 3.1–3.11
- Interactives 3.1–3.2
- Quizzes 3.1–3.3
- Programming Assignment 3.1: Multilayer Perceptron (Part B)
- Capstone Assignment Checkpoint 3

Back to top

Module 4: Basics of Convolutional Neural Networks

Learning Outcomes:

1. Describe the need for shift-invariant pattern detection.
2. Describe the process of shift-invariant pattern detection using a scanning neural network.
3. Explain the training of neural networks with shared parameters.
4. Explain the reordering of the computation of a scanning neural network.
5. Explain the benefits of distributing pattern matching procedures over multiple layers in a scanning neural network.
6. Summarize findings of early research into animal vision processes.

7. Describe the first computational model of position-invariant animal vision: the unsupervised neocognitron.
8. Explain how adding supervision to the neocognitron leads to a convolutional neural network.
9. Describe the process and meaning of convolution in the context of a convolutional neural network.
10. Describe the cube view for understanding convolutional maps at any location in a convolutional neural network.
11. Explain the significance of factors that determine convolution output size.
12. Describe the processes of upsampling and downsampling.
13. Describe the role of layers that perform downsampling by pooling.
14. Identify the parameters needed to perform image classification with a convolutional neural network.
15. Relate the basic architecture of a time-delay neural network to convolutional neural network processes.
16. Implement the forward and backward pass of a 1D convolutional layer.

Key Activities

- Videos 4.1–4.9
- Interactives 4.1–4.6
- Quizzes 4.1–4.3
- Programming Assignment 4.1: Convolutional Neural Networks (Part A)
- Capstone Assignment Checkpoint 4

Back to top

Module 5: CNN: Training and Variants

Learning Outcomes:

1. Summarize the backpropagation process through flat, convolutional, and pooling layers of a convolutional neural network.
2. Explain how to compute derivatives for the affine map z in convolutional layers of a convolutional neural network through backpropagation.
3. Explain the dependency paths between individual elements of an activation map and the loss.
4. Explain the computation of derivatives for the elements of the activation maps through backpropagation.
5. Explain how computing the derivatives for an activation map through backpropagation is actually a convolution.
6. Explain the formula for computing the derivatives of the filter weights for backpropagation.
7. Explain how computing the derivatives of filter weights through backpropagation is actually a convolution.

8. Describe changes in the convolution method when the convolutional stride is greater than one.
9. Explain the programmatic shortcut for computing derivatives through backpropagation.
10. Explain the derivatives at pooling layers through backpropagation.
11. Explain forward interference and backpropagation in upsampling layers.
12. Compare and contrast variations and extensions of the basic CNN model and additional information.
13. Implement a neural network to classify images of faces.

Key Activities

- Videos 5.1–5.8
- Interactives 5.1–5.4
- Quizzes 5.1–5.3
- Programming Assignment 5.1: Face Classification (Part B)
- Capstone Assignment Checkpoint 5

Back to top

Module 6: Basics of Recurrent Neural Networks

Learning Outcomes:

1. Describe the types of problems that require recurrence in neural networks.
2. Explaining the pictorial representation of recurrent neural networks used in this program.
3. Explain why recurrent connections are needed to refer to historical trends and patterns.
4. Summarize the basic algorithm used by a recurrent neural network.
5. Compare and contrast the architectures and processes of unidirectional and bidirectional recurrent neural networks.
6. Summarize the algorithm for backpropagation through time for a unidirectional recurrent neural network.
7. Summarize the algorithm for backpropagation through time for a bidirectional recurrent neural network.
8. Discuss the stability and memory limitations of a variety of activation functions in a recurrent neural network.
9. Explain the problem of vanishing gradients while training deep and recurrent neural networks.
10. Describe the long-term dependency problem and a possible solution.
11. Describe the architecture of a long short-term memory cell.
12. Explain the Gated Recurrent Unit as an alternative to the Long Short-Term memory unit.
13. Summarize the algorithm for backpropagation for a long short-term memory recurrent neural network.
14. Implement backpropagation through time for a bidirectional recurrent neural network.

Key Activities

- Videos 6.1–6.12
- Interactives 6.1–6.2
- Quizzes 6.1–6.3
- Programming Assignment 6.1: Recurrent Neural Networks (Part A)
- Capstone Assignment Checkpoint 6

Back to top

Module 7: Connectionist Temporal Classification and Sequence-to-Sequence Models

Learning Outcomes:

1. Describe the architecture and training process for a time-synchronous recurrent neural network.
2. Describe the greedy approach to decoding the output of an order-synchronous, but time-asynchronous recurrent network.
3. Identify the role of alignment in terms of computing divergence between input and output for an order-synchronous, but time-asynchronous recurrent network.
4. Describe and critique the iterative algorithm of repeatedly determining the alignment and updating model parameters to train an order-synchronous network.
5. Explain the alternative approach of training the model over all possible alignments of the input.
6. Define forward and backward a posteriori symbol probability.
7. Sequence the steps to calculate the forward symbol probabilities recursively.
8. Sequence the steps to calculate the symbol probabilities recursively.
9. Sequence the steps to calculate the expected divergence and its derivative.
10. Summarize the steps for training a sequence-to-sequence recurrent neural network.
11. Describe the purpose of blanks in training a sequence-to-sequence recurrent neural network.
12. Summarize the steps for training a sequence-to-sequence recurrent neural network that contains blanks.
13. Compare and contrast the outcome of greedy decoding and the actual objective of decoding.
14. Describe the beam-search method of using semi-trees to optimize decoding for connectionist temporal classification.
15. Implement a speech-to-text model that directly produces readable English.

Key Activities

- Videos 7.1–7.13
- Interactive 7.1
- Quizzes 7.1–7.3
- Programming Assignment 7.1: Transcript Generation (Part B)
- Capstone Assignment Checkpoint 7

Back to top

Module 8: Attention and Translation

Learning Outcomes:

1. Explain how to compute embeddings of words from one-hot encodings, using language prediction with neural networks.
2. Describe the architecture of a recurrent neural network used for language prediction.
3. Describe the synchrony problem of sequence-to-sequence models.
4. Describe the "simple" encoder-decoder architecture using recurrent neural networks for sequence-to-sequence conversion.
5. Describe the training of a simple encoder-decoder system composed of recurrent neural networks.
6. Describe the impact of greedy drawing on the decoder outputs of an RNN-based encoder-decoder system.
7. Describe beam-search decoding to improve the optimality of the decoder output.
8. Describe the use of encoder-decoder models for machine translation and image captioning.
9. Explain how differential weighting of encoder outputs can improve sequence-to-sequence conversion output.
10. Introducing the attention model for sequence-to-sequence conversion.
11. Describing beam-decoding with attention models.
12. Explaining attention weight patterns in sequence-to-sequence conversion with attention networks.
13. Describe the backpropagation process for an attention model.
14. Describe important extensions of attention models and their use cases.

Key Activities

- Videos 8.1–8.13
- Interactive 8.1
- Quizzes 8.1–8.4
- Capstone Assignment Checkpoint 8

Back to top

Module 9: Transformers and Graph Networks

Learning Outcomes:

1. Describing the problem of overlapping classes.
2. Explain the relationship of the logistic regression model and a perceptron with a sigmoid activation function.
3. Describe how the maximum likelihood estimate is used to learn the parameters of a logistic regression model.

4. Generalize the logistic regression to classification with features derived from the neural network layers until the penultimate layer.
5. Describe the manifold hypothesis and its relationship to the linear separability of features derived by the lower layers of a multilayer perceptron.
6. Explain how perceptrons are basic pattern detectors and how their output can be used to reconstruct the input.
7. Describe the structure of the simplest linear autoencoder and how it performs PCA.
8. Describe how neural network output with non-linear activations performs non-linear PCA.
9. Describe how the decoder of a neural-network autoencoder composes a non-linear constructive dictionary for the data.
10. Explain the usefulness of neural dictionaries through a signal separation tasks.

Key Activities

- Videos 9.1–9.9
- Interactive 9.1
- Quizzes 9.1–9.3
- Capstone Assignment Checkpoint 9

Back to top

Module 10: Representation, Variational Autoencoders, and Generative Adversarial Networks

Learning Outcomes:

1. Identify the impact of naively removing recurrence in the encoder portion of an autoencoder.
2. Describe the architecture and purpose of a multi-head self attention block in the context of encoders.
3. Explain how positional vectors use word position to determine final embedding.
4. Summarize the architecture of a transformer and a couple of variants with respect to encoder and decoder structure.
5. Describe properties of data types that are able to be represented as a graph.
6. Describe properties of problems that are able to be addressed by a graph network.
7. Summarize the steps used in training a graph neural network.
8. Explain how statistical constraints placed upon an autoencoder relate to generated output quality.
9. Explain how statistical constraints placed upon an autoencoder alter the training approach for that architecture.
10. Explain how the encoder in an autoencoder must encode the distribution of the hidden representations of the data.
11. Summarize steps used to train the encoder and decoder of a variational autoencoder.
12. Describe the generation of new synthetic samples using a variational autoencoder.

13. Relate the purposes of the generator and discriminator components of a generative adversarial network to the overall operation of the network.
14. Summarize the behavior of an optimally-trained generative adversarial network.
15. Contrast properties of variational autoencoders with generative adversarial networks.
16. Summarize various state-of-art models and identify the components that we have learned about.

Key Activities

- Videos 10.1–10.16
- Quizzes 10.1–10.4
- Capstone Assignment Submission

Back to top