



COLORADO STATE UNIVERSITY — GLOBAL —

CSC407: ADVANCED NEURAL NETWORKS

Credit Hours: 3

Contact Hours: This is a 3-credit course, offered in accelerated format. This means that 16 weeks of material is covered in 8 weeks. The exact number of hours per week that you can expect to spend on each course will vary based upon the weekly coursework, as well as your study style and preferences. You should plan to spend 14-20 hours per week in each course reading material, interacting on the discussion boards, writing papers, completing projects, and doing research.

Faculty Information: Faculty contact information and office hours can be found on the faculty profile page.

COURSE DESCRIPTION AND OUTCOMES

Course Description:

This course introduces theories associated with neural networks and neural network models. Students will apply basic principles and methodologies associated with network processing, learning algorithms, and applications. Students will analyze propagation, feedforward networks, perceptrons, and self-organizing networks.

Course Overview:

This course will look at Deep Learning which is also referred to as Neural Networks (or Neural Nets)—a topic that has been around for a long time. In this course, you will examine the different approaches to deep learning including the basic neural network components and the different techniques that can be applied to process data. This course will reinforce the material you have learned in Artificial Intelligence and Machine Learning courses by identifying associations within a set of patterns and determining relationships in a neural network. This course will also look at the quantitative components that are used in deep learning problems and discuss the approaches for solving them.

Course Learning Outcomes:

- Identify basic neural network constructs.
- Apply techniques for the processing of data.
- Discuss numerical computation components used in deep learning problems.
- Implement components for deep learning networks.
- Identify associations within a set of patterns.
- Implement techniques for determining relationships in a neural network.

PARTICIPATION & ATTENDANCE

Prompt and consistent attendance in your online courses is essential for your success at CSU-Global Campus. Failure to verify your attendance within the first seven days of this course may result in your withdrawal. If for some reason you would like to drop a course, contact your advisor.

Online classes have deadlines, assignments, and participation requirements just like on-campus classes. Budget your time carefully and keep an open line of communication with your instructor. If you are having technical problems, problems with your assignments, or other problems that are impeding your progress, let your instructor know as soon as possible.

COURSE MATERIALS

Required:

- Chollet, F. (2017). *Deep learning with Python*. Shelter Island, NY: Manning Publications. ISBN: 9781617294433
- Ravichandiran, S. (2019). *Hands-on deep learning algorithms with Python*. Birmingham, UK: Packt Publishing. ISBN: 9781789344158
- Trask, A. (2019). *Grokking deep learning*. Shelter Island, NY: Manning Publications. ISBN: 9781617293702

Suggested:

None

NOTE: All non-textbook required readings and materials necessary to complete assignments, discussions, and/or supplemental or required exercises are provided within the course itself. Read through each course module carefully.

COURSE SCHEDULE

Due Dates

The Academic Week at CSU-Global begins on Monday and ends the following Sunday.

- **Discussion Boards:** The original post must be completed by Thursday at 11:59 p.m. MT and Peer Responses posted by Sunday at 11:59 p.m. MT. Late posts may not be awarded points.
- **Opening Exercises:** Take the opening exercise before reading each week's content to see which areas you will need to focus on. You may take these exercises as many times as you need. The opening exercises will not affect your final grade.
- **Mastery Exercises:** Students may access and retake mastery exercises through the last day of class until they achieve the scores they desire.
- **Critical Thinking:** Assignments are due Sunday at 11:59 p.m. MT.

WEEKLY READING AND ASSIGNMENT DETAILS

Module 1

Readings

- Chapter 1 in *Grokking Deep Learning*

- Chaphalkar, N.B., Iyer, K.C., & Patil, S. K. (2015). Prediction of outcome of construction dispute claims using multilayer perceptron neural network model. *International Journal of Project Management*, 33(8), 1827-35.
- Dawson, M. & Gupta, M. (2017). Probability matching in perceptrons: Effects of conditional dependence and linear nonseparability. *PLoS ONE*, 12(2), 1–13.
- Rusk, N. (2016). Deep learning. *Nature Methods*, 13(1), 35.
- Xiao, N. (2016). Spatial pattern and analysis. In *GIS algorithms* (pp. 177-210). 55 City Road, London: SAGE Publications Ltd.

Opening Exercise (0 points)

Discussion (25 points)

Critical Thinking: Title (60 points)

Option #1: K-Nearest Neighbor Classifier

Rework the first worked example in Module 1 on page 4 by modifying the Python script to generate classification boundaries based on Iris petal sizes.

Submit your Python script as a .py file and include a screen shot of your KNN classification map.

Option #2: Evaluating the Cosine Function Using Excel and Neural Network Emulation

Rework the second worked example in Module 1 on page 4 by modifying the Excel workbook to predict the output of the cosine function using the deep learning network.

Include a graphical comparison of the predicted output and the truth.

Submit your Excel workbook showing your calculations

Mastery Exercise (10 points)

Module 2

Readings

- Chapter 2 in *Grokking Deep Learning*
- Dabbura, I. (2018, March 31). Coding neural network—Forward propagation and backpropagation. Retrieved from <https://towardsdatascience.com/coding-neural-network-forward-propagation-and-backpropagation-ccf8cf369f76>
- Kamimura, R. (2016). Simplified information maximization for improving generalization performance in multilayered neural networks. *Mathematical Problems in Engineering*, 2016, 1-17.
- Kamimura, R. (2019). Neural self-compressor: Collective interpretation by compressing multi-layered neural networks into non-layered networks. *Neurocomputing*, 323, 12-36.
- Luhaniwal, V. (2019, May 7). Forward propagation in neural networks—Simplified math and code version. Retrieved from <https://towardsdatascience.com/forward-propagation-in-neural-networks-simplified-math-and-code-version-bbcfef6f9250>

Opening Exercise (0 points)

Discussion (25 points)

Critical Thinking: Title (65 points)

Option #1: Create a Neural Network

Create a neural network with only one hidden layer (of any number of units) that implements the following logic function:

$$(A \vee \sim B) \oplus (\sim C \vee \sim D)$$

Do not forget to assign weights to each of the units in the hidden layer and the output unit.

Make a truth table for each combination of inputs, the logical output of the equation, as well as the output of the neural network.

Submit your drawing and truth table in a Word document.

Note that the challenging part of this assignment will be devising a set of weights for the hidden layer units and output unit.

Option #2: Implement Python Code for a Neural Network

Implement a Python solution for the neural network model shown in Figure 16 on page four of Module 1. Do not worry about training the model.

Run the Python code for angle values in the following Python list: [0, 30, 45, 60, 90].

Submit the Python code (.py file) and a screenshot showing the iterative outputs in a zip file using the name LastName_FirstName_CSC407_CTA2_2.zip.

Mastery Exercise (10 points)

Module 3

Readings

- Chapters 3 & 4 in *Grokking Deep Learning*
- Hezaveh, Y., Levasseur, L., & Marshall, P. (2017). Fast automated analysis of strong gravitational lenses with convolutional neural networks. *Nature*, 548(7669), 555-57.
- Lusch, B., Weholt, J., Maia, P., and Kutz, N. (2018). Modeling cognitive deficits following neurodegenerative diseases and traumatic brain injuries with deep convolutional neural networks. *Brain and Cognition*, 123, 154-64.
- Mahanta, J. (2017, April). Keep it simple! How to understand gradient descent algorithm. Retrieved from <https://www.kdnuggets.com/2017/04/simple-understand-gradient-descent-algorithm.html>
- Pandey, P. (2019, March 18). Understanding the mathematics behind gradient descent. Retrieved from <https://towardsdatascience.com/understanding-the-mathematics-behind-gradient-descent-dde5dc9be06e>

- Ren, Y. (2019, May 30). A step-by-step implementation of Gradient Descent and backpropagation. Retrieved from <https://towardsdatascience.com/a-step-by-step-implementation-of-gradient-descent-and-backpropagation-d58bda486110>

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking: Title (60 points)

Option #1: Gradient Descent in One Neuron, Option 1

Using the gradient descent techniques of neural learning for a single neuron you learned in Module 3:

- Find the least number of steps required to make a prediction within 0.0000001 of the given goal prediction.
- Repeat the same calculation for the hot and cold algorithm.
- Print out the iterations from both programs and save the results from both programs to a single output text file.
- Once your processing is complete, summarize the comparison in your output text file.
- In addition, comment on what would happen if alpha were less than zero.
- Use the following data for the gradient descent:
weight, goal_pred, input, alpha = (0.72, 0.8, 2.5, 0.1)
- Use the following data for the hot-and-cold processing:
weight, goal_prediction, input, step_amount = (0.72, 0.8, 2.5, 0.001)
- Submit your two Python scripts and output text file as zip archive.

Option #2: Gradient Descent in One Neuron, Option 2

Using the gradient descent techniques of neural learning for a single neuron you learned in Module 3:

- Find the least number of steps required to make a prediction within 0.0000001 of the given goal prediction.
- Repeat the same calculation for the hot and cold algorithm.
- Print out the iterations from both programs and save the results from both programs to a single output text file.
- Once your processing is complete, summarize the comparison in your output text file.
- In addition, comment on what would happen if alpha were less than zero.
- Use the following data for the gradient descent:
weight, goal_pred, input, alpha = (0.72, 0.8, 2.0, 0.01)
- Use the following data for the hot-and-cold processing:
weight, goal_prediction, input, step_amount = (0.72, 0.8, 2.0, 0.01)
- Submit your two Python scripts and output text file as zip archive.

Module 4

Readings

- Chapter 5 in *Grokking Deep Learning*

- Andrychowicz, M., Denil, M., Colmenarejo, S., Hoffman, M., Pfau, D., Schaul, T. . . , & Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. Retrieved from <http://papers.nips.cc/paper/6461-learning-to-learn-by-gradient-descent-by-gradient-descent.pdf>
- Bottou, L. (1991). Stochastic gradient learning in neural networks. Retrieved from <https://leon.bottou.org/publications/pdf/nimes-1991.pdf>
- Dukor, S. (2018, November 12). Neural representation of AND, OR, ANOT, XOR and XNOR logic gates (perceptron algorithm). Retrieved from <https://medium.com/@stanleydukor/neural-representation-of-and-or-not-xor-and-xnor-logic-gates-perceptron-algorithm-b0275375fea1>
- Fernandes, J. (2018, May 4). Gradient descent in *Neural Networks and Convolutional Neural Networks Essential Training*. [Video file].
- Gradient Descent and Python Supplemental Videos
- Santoro, A., Bartunov, S., Botvinick, M. Wierstra, D., & Lillicrap, T. (2016). Meta-learning with memory-augmented neural networks. Retrieved from <http://proceedings.mlr.press/v48/santoro16.pdf>

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking: Title (60 points)

Option #1: Investigating the Hyperbolic Tangent Neural Network Activation Function

Revise the worked example from What You Need to Know for This Module's Critical Thinking Assignment where, instead of using the sigmoid activation function, use the hyperbolic tangent activation function.

Generate a plot of the cost function resulting from training of 5000 data points.

Place your plot and comment on the loss over the training iterations in a Word document.

Submit your Word document and Python script in the form of a zip file with the file name as LastName_FirstName_CSC407_CTA4_1.zip.

Option #2: Investigate the Leaky ReLU Neural Network Activation Function

Revise the worked example from What You Need to Know for This Module's Critical Thinking Assignment where, instead of using the sigmoid activation function, use the leaky ReLU activation function with the alpha value = 0.01.

Generate a plot of the cost function resulting from training of 5000 data points.

Place your plot and comment on the loss over the training iterations in a Word document.

Submit your Word document and Python script in the form of a zip file with the file name as LastName_FirstName_CSC407_CTA4_1.zip.

Portfolio Milestone (125 points)

Option #1: Testing a Prediction Using the Radial Basis Function

Modify the following code in What You Need to Know about This Module's Portfolio to include a user prompt to allow the user to enter the coordinates for a point to be classified.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
x = np.array([[1,3],[1,2],[1,1.5],[1.5,2],[2,3],[2.5,1.5],
              [2,1],[3,1],[3,2],[3.5,1],[3.5,3]])
y = [0]*6 + [1]*5
svc = svm.SVC(kernel='rbf', C=1, gamma=3).fit(x, y)
X,Y = np.mgrid[0:4:200j,0:4:200j]
Z = svc.decision_function(np.c_[X.ravel(),Y.ravel()])
Z = Z.reshape(X.shape)
plt.contourf(X,Y,Z > 0,alpha=0.4)
plt.contour(X,Y,Z,colors=['k', 'k', 'k'], linestyles=['--','-', '--'],levels=[-1,0,1])
plt.scatter(svc.support_vectors_[:,0],svc.support_vectors_[:,1],s=120,facecolors='none')
plt.scatter(x[:,0],x[:,1],c=y, s=50,alpha=0.9)
```

- Plot the user point and recalculate the radial basis function.
- Allow the user to enter up to ten different data points.
- Show a graph of the decisional area for each additional point the user enters.
- Comment on the accuracy of the classification.
- Submit your plots and comments in a Word- formatted document.
- Submit your Word document and Python script in the form of a zip file with the file name as LstName_FirstName_CSC407_Portfolio_MOD4_Opt1.zip.

Option #2: Testing a Prediction Using a Polynomial Kernel

Modify the following code in What You Need to Know about This Module's Portfolio to include a user prompt to allow the user to enter the coordinates for a point to be classified. This is the example of applying the SVC algorithm with a polynomial kernel.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
x = np.array([[1,3],[1,2],[1,1.5],[1.5,2],[2,3],[2.5,1.5],
              [2,1],[3,1],[3,2],[3.5,1],[3.5,3]])
y = [0]*6 + [1]*5
svc = svm.SVC(kernel='poly', C=1, degree=3).fit(x, y)
X,Y = np.mgrid[0:4:200j,0:4:200j]
Z = svc.decision_function(np.c_[X.ravel(),Y.ravel()])
```

```

Z = Z.reshape(X.shape)
plt.contourf(X,Y,Z > 0,alpha=0.4)
plt.contour(X,Y,Z,colors=['k', 'k', 'k'], linestyles=['--','-', '--'],levels=[-1,0,1])
plt.scatter(svc.support_vectors_[:,0],svc.support_vectors_[:,1],s=120,facecolors='none')
plt.scatter(x[:,0],x[:,1],c=y, s=50,alpha=0.9)

```

- Plot the user point and recalculate the polynomial kernel.
- Allow the user to enter up to ten different data points.
- Show a graph of the decisional area for each additional point the user enters.
- Comment on the accuracy of the classification.
- Submit your plots and comments in a Word- formatted document.
- Submit your Word document and Python script in the form of a zip file with the file name as LstName_FirstName_CSC407_Portfolio_MOD4_Opt2.zip.

Module 5

Readings

- Chapter 2 “Getting to Know TensorFlow” in *Hands-On Deep Learning Algorithms with Python*
- AI Blog (2019, June 23). Deep learning with TF 2.0: 04.00- numerical computation. Retrieved from <https://www.adhiraiyan.org/deeplearning/04.00-Numerical-Computation>
- GeeksforGeeks (n.d.) *Introduction to TensorFlow*. Retrieved from <https://www.geeksforgeeks.org/introduction-to-tensorflow/>
- GitHub (2019). *Math operations in TensorFlow*. Retrieved from <http://bit.ly/2YSYbYu>
- GitHub (2019). *Neural network from scratch*. Retrieved from <https://github.com/sar-gupta/neural-network-from-scratch>
- TensorFlow (n.d.) *Basic classification: Classify images of clothing*. Retrieved from <https://www.tensorflow.org/tutorials/keras/classification>
- TensorFlow (n.d.) *TensorFlow core r2.0*. Retrieved from https://www.tensorflow.org/api_docs/python/tf

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking: Title (65 points)

Option #1: Using Keras to Classify Clothing Images

In this assignment, you will use Keras to classify images of clothing using the *Fashion MNIST dataset*, which contains 70,000 grayscale images in ten categories.

- Use the instructions in your assignment in Canvas to complete this assignment.
- When finished, submit a screenshot of your predictions along with your Python source code in a zip archive file entitled: LastName_FirstName_CSC407_CTA5_1.zip

Option #2: Using Keras to Train Neural Networks

In this assignment, you will use a neural network to predict future revenues from the sales of a new video game.

- Download the following files for this assignment:
 - **sales_data_training.csv**
 - **sales_data_test.csv**
 - **proposed_new_product.csv**
- Use the instructions in your assignment in Canvas to complete this assignment.
- When finished, submit a screenshot of your predictions along with your Python source code in a zip archive file entitled: LastName_FirstName_CSC407_CTA5_1.zip

Module 6

Readings

- Chapter 3a “Demystifying Gradient Descent” in *Hands-On Deep Learning Algorithms with Python*
- Cecen, A., Dai, H., Yabunsu, Y., Kalidindi, S. & Song, L. (2018, March). Material structure-property linkages using three-dimensional convolutional neural networks, *Acta Materialia*, 146, pp. 76-84.
- Simonyan, K. & Zisserman, A. (2015, April 10). Very deep convolutional networks for large-scale image recognition.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going deeper with convolutions. *Computer Vision and Pattern Recognition*, 1- 9. doi: 10.1109/CVPR.2015.7298594
- Wu, J., Zhong, S., Ma, Z., Heinen, S., & Jiang, J. (2018). Foveated convolutional neural networks for video summarization. *Multimedia Tools and Applications*, 77(22), 29245–67.

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking: Title (60 points)

Option #1: Neural Network Image Classifiers

For this project, you will need to have TensorFlow and Keras available for the Python-base project.

- Access the Keras home page: <https://keras.io/> and click on "Getting Started" and read the "Guide to the Sequential Model."
- Look over the CIFAR-10 dataset. You will work with neural network image classifiers for this dataset. The URL is <https://www.cs.toronto.edu/~kriz/cifar.html>
- Download the file `cifar10_cnn.py` from the example folder: <https://github.com/keras-team/keras/tree/master/examples>
- Complete the following tasks by modifying the code in `cifar10_cnn.py`.
 - Compare the accuracy of the convolutional neural network in the file `cifar10_cnn.py` on the cifar10 dataset to the accuracy of simple dense neural networks with 0, 1, 2, 3 and 4 hidden layers of 512 rectified linear units each.
 - Modify the code in **`cifar10_cnn.py`** to obtain simple dense neural networks with 0, 1, 2, 3 and 4 hidden layers of 512 rectified linear units (with a dropout rate of 0.5).

- Produce a graph that contains six curves (one for the convolutional neural net and one for each dense neural net of 0-4 hidden layers). The y-axis is the test (validation) accuracy and the x-axis is the number of epochs (# of passes through the training set).
- Produce curves for the first 10 epochs. Although 10 epochs are not sufficient to reach convergence, they are sufficient to see the trend.
- Explain the results (i.e., why some models perform better or worse than other models).
- Submit your Python code, screenshots of your graphs, and analysis in a zip archive file named LastName_FirstName_CTA6_1.zip.

Option #2: Value Iteration vs. Model-Free Learning

For this project, you will need to have TensorFlow and Keras available for the Python-base project.

- Access the Keras home page: <https://keras.io/> and click on "Getting Started" and read the "Guide to the Sequential Model."
- Look over the CIFAR-10 dataset. You will work with neural network image classifiers for this dataset. The URL is <https://www.cs.toronto.edu/~kriz/cifar.html>
- Download the file `cifar10_cnn.py` from the example folder:
- <https://github.com/keras-team/keras/tree/master/examples>
 - Answer the following questions by modifying the code in `cifar10_cnn.py`. Compare the accuracy achieved by rectified linear units and sigmoid units in the convolutional neural network in `cifar10_cnn.py`.
 - Modify the code in `cifar10_cnn.py` to use sigmoid units.
 - Produce a graph that contains two curves (one for rectified linear units and another one for sigmoid units). The y-axis is the test (validation) accuracy and the x-axis is the number of epochs (# of passes through the training set).
 - Produce curves for the first 10 epochs. Although 10 epochs is not sufficient to reach convergence, it is sufficient to see the trend.
 - Explain the results (i.e., why did one model perform better than the other model).
- Submit your Python code, screenshots of your graphs, and analysis in zip archive file named LastName_FirstName_CTA6_2.zip.

Module 7

Readings

- Chapter 4a "Introducing RNNs" in *Hands-On Deep Learning Algorithms with Python*
- GitHub (2019). *Handwriting generation by RNN with TensorFlow*. Retrieved from <https://github.com/snowkylin/rnn-handwriting-generation>
- Quax, S. & Van Gerven, M. (2018). Emergent mechanisms of evidence integration in recurrent neural networks. *PLoS ONE*, 13(10).
- Sherstinsky, A. (2018, November 4). *Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network*.
- Yu, A. (2019, February 26). Getting a machine to do my English homework for me: Using recurrent neural networks for text generation. Retrieved from <https://towardsdatascience.com/getting-a-machine-to-do-my-english-homework-for-me-5d339470fe42>

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Module 8

Readings

- Chapter 7 in *Fundamentals of Deep Learning*
- Cho, K., Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., . . . Bengio, Y. (2014). *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. Cornell University, pp. 1-15.
- Chollet, F. (2017, September 29). *A ten-minute introduction to sequence-to-sequence learning in Keras*. Retrieved from <https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html>
- Ma'amari, M. (2018, November 8). NLP | Sequence to sequence networks | Part 1 | Processing text data. Retrieved from <https://towardsdatascience.com/nlp-sequence-to-sequence-networks-part-1-processing-text-data-d141a5643b72>
- Ma'amari, M. (2018, November 15). NLP | Sequence to sequence networks | Part 2 | Seq2seq model (EncoderDecoder model). Retrieved from <https://towardsdatascience.com/nlp-sequence-to-sequence-networks-part-2-seq2seq-model-encoderdecoder-model-6c22e29fd7e1>
- Sutskever, I., Vinyals, O., & Quoc, V., (2014). *Sequence to Sequence Learning with Neural Networks*. Cornell University, pp. 1-9.
- Takezawa, A. (2018, March). *How to implement seq2seq LSTM model in Keras #shortcutNLP*. Retrieved from <https://towardsdatascience.com/how-to-implement-seq2seq-lstm-model-in-keras-shortcutnlp-6f355f3e5639>

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Portfolio Project (225 points)

Option #1: Implement a Date Converter Using Sequence-to-Sequence Learning

- Review the instructions for your final project in your assignment in Canvas.
- Submit the following items:
 - your Python source file (only one is needed),
 - a logic diagram of your model, and
 - a screenshot of your model's predictions.
- Name your project: LastName_FirstName_CSC407_FinalPortfolio_Opt1.zip.

Option #2: Implement an LSTM Encoder-Decoder Using Sequence-to-Sequence Learning

- Review the instructions for your final project in your assignment in Canvas.
- Submit the following items:
 - a comprehensive flowchart depicting the processing performed by your neural network model,
 - the Python source file that is thoroughly documented, and
 - an output text file showing the run-time predictions of your model.
- Name your project: LastName_FirstName_CSC407_FinalPortfolio_Opt2.zip

COURSE POLICIES

Grading Scale	
A	95.0 – 100
A-	90.0 – 94.9
B+	86.7 – 89.9
B	83.3 – 86.6
B-	80.0 – 83.2
C+	75.0 – 79.9
C	70.0 – 74.9
D	60.0 – 69.9
F	59.9 or below

Course Grading

20% Discussion Participation
0% Opening Exercises
0% Live Classroom
8% Mastery Exercises
37% Critical Thinking Assignments
35% Final Portfolio Project

IN-CLASSROOM POLICIES

For information on late work and incomplete grade policies, refer to our [In-Classroom Student Policies and Guidelines](#) or the Academic Catalog for comprehensive documentation of CSU-Global institutional policies.

Academic Integrity

Students must assume responsibility for maintaining honesty in all work submitted for credit and in any other work designated by the instructor of the course. Academic dishonesty includes cheating, fabrication, facilitating academic dishonesty, plagiarism, reusing /re-purposing your own work (see *CSU-Global Guide to Writing and APA Requirements* for percentage of repurposed work that can be used in an assignment), unauthorized possession of academic materials, and unauthorized collaboration. The CSU-Global Library provides information on how students can avoid plagiarism by understanding what it is and how to use the library and internet resources.

Citing Sources with APA Style

All students are expected to follow the *CSU-Global Guide to Writing and APA Requirements* when citing in APA (based on the APA Style Manual, 6th edition) for all assignments. For details on CSU-Global APA style, review the APA resources within the CSU-Global Library under the “APA Guide & Resources” link. A link to this document should also be provided within most assignment descriptions in your course.

Disability Services Statement

CSU-Global is committed to providing reasonable accommodations for all persons with disabilities. Any student with a documented disability requesting academic accommodations should contact the Disability Resource Coordinator at 720-279-0650 and/or email ada@CSUGlobal.edu for additional information to coordinate reasonable accommodations for students with documented disabilities.

Netiquette

Respect the diversity of opinions among the instructor and classmates and engage with them in a courteous, respectful, and professional manner. All posts and classroom communication must be conducted in accordance with the student code of conduct. Think before you push the Send button. Did you say just what you meant? How will the person on the other end read the words?

Maintain an environment free of harassment, stalking, threats, abuse, insults or humiliation toward the instructor and classmates. This includes, but is not limited to, demeaning written or oral comments of an ethnic, religious, age, disability, sexist (or sexual orientation), or racist nature; and the unwanted sexual advances or intimidations by email, or on discussion boards and other postings within or connected to the online classroom. If you have concerns about something that has been said, let your instructor know.