

CPTS 434/534: Neural Network Design And Application

Instructor: Yan Yan

Spring, 2023

Class Hours: Tue/Thu 7:45AM - 9:00AM

Class Room: CLEV 30

Office Hours: Tue/Thu 9AM - 10AM (need sign-up)

Office: EME 123

TA: Azza Fadhel

TA office hour: **TBA** (need sign-up)

TA office: **TBA**

Contact:

Course materials: Instructor

Assignments/grading: TA

Preferred: Message on Canvas

Alternative: Email

Instructor: yan.yan1@wsu.edu

TA: azza.fadhel@wsu.edu

1 Course Description

This course provides an introduction to deep neural networks (DNNs), a key component in modern machine learning. It includes both *theoretical* foundations of basic machine learning, and hands-on *implementation* of deep neural networks for some applications.

This course starts with basic concepts and components in statistical machine learning to shed light on the answers to these questions: (i) *what is learning*, (ii) *what can be learned*, (iii) *how to learn it*, (iv) *how well we can learn it*, and (v) *what machine learning can do*.

Then we will cover the motivation of DNNs, as modern machine learning methods, over traditional machine learning algorithms in the viewpoint of *representation learning*, an important task of deep learning. As a key type of machine learning method, neural networks themselves have unique properties. We will discuss (i) *general frameworks and typical components* in DNNs, (ii) the *computational challenges*, and (iii) *key principles* to learn DNNs.

After that we will detail several common types of neural networks that are commonly used for different modalities of data or learning targets, such as (i) convolutional neural networks (CNNs) for image data, (ii) recurrent neural networks (RNNs) for sequence data, (iii) graph neural networks (GNNs) for graph data, (iv) generative adversarial networks (GANs), and (v) a family of neural architecture search (NAS) methods. Additionally, we will also go over some recent open research problems in deep learning such as diffusion model, contrastive learning, conformal prediction.

2 Course Sections/Topics

- Machine learning [ML]
 - Basic concepts: general ideas, terminologies, examples.
 - PAC learning: formal definition of what is learning, what can be learned, how well we can learn.
 - ERM and MLE: common ways to build/interpret a machine learning model.
- Neural networks [NN]
 - Basic concepts: general ideas, terminologies, examples.
 - Composition of functions: what is neural network structure
 - Units in NN: commonly used functions for building a neural network
 - BP: how to optimize the learning problem in NN.
 - MLP Softmax: a real example of learning a neural network Softmax classifier with BP.
- Convolutional neural networks [CNN]
 - Convolutional operation on image: how the convolutional operation works and what is the output.
 - CNN design: typical Convolutional layers and properties.
 - CNN architecture: the history of CNN architecture design and properties (advantages and disadvantages).
 - Practical tricks for building CNNs: normalization, batch normalization, data augmentation, dropout, weight decay, pre-trained model, stagewise training.
- Recurrent neural networks [RNN]
 - Sequence data: properties, motivation for recurrent networks.
 - Design RNNs: typical way to build RNNs, character-level language model.
 - Design LSTM: from conventional RNN cell to LSTM cell.
 - RNN architecture: more complicated RNN architectures: bi-directional RNN, Seq2seq RNN.
 - Attention mechanism: architecture design.
 - Transformers (a neural network architecture, not a film).
- Graph neural networks [GNN]
 - Graph data in ML: from i.i.d. data to connected (non-i.i.d.) data.
 - Graph representation: how to describe a graph.
 - GNN: how to aggregate neighbors in a neural networks.
 - Oversmoothing in GNN: a key challenge.

- Generative adversarial networks [GAN]
 - Adversarial ML: definition and properties.
 - Generative model: definition and properties.
 - GAN: how to generate synthetic data.
- Neural architecture search [NAS]

How to learn an architecture in a data-driven way.
- Open problems in deep learning [OP]
 - Contrastive learning.
 - Meta-learning.
 - Federate learning.
 - Conformal prediction.
 - Diffusion model.

3 Course Objectives

- Understand machine learning basics: the pipeline of machine learning, what can be learned, how to learn, and statistical/optimization/computational challenges
- Understand the benefits of DNNs for learning representations from raw data over traditional machine learning algorithms, and the computational/statistical challenges of learning DNNs.
- Understand the key principles/ideas for efficiently learning parameters of DNNs from data.
- Understand the structures of different types of DNNs, e.g., feedforward neural networks, CNNs, RNNs and GNNs, and how they can handle target data types.
- Able to apply DNNs to solve supervised learning problems over images, text, and graphs.
- Able to read research papers in deep learning, understand the issues raised by them and catch their ideas of proposed solutions.

4 Textbooks

- Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. “Foundations of Machine Learning”. MIT Press, Second Edition, 2018. Online: <https://cs.nyu.edu/~mohri/mlbook/>.
- Ian Goodfellow, Yoshua Bengio, and Aron Courville. “Deep Learning”. MIT Press. Online: <https://www.deeplearningbook.org/>.
- Aston Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola. “Dive into Deep Learning”. Online book with code: <https://d2l.ai/>.

5 Course Materials

All course materials can be found on Canvas "Home" page or "Modules" page. Slides will be grouped into sections as listed in Section 2. Assignments for each sections will also be open in the corresponding modules.

6 Contact Instructor And TA Efficiently

There are many ways to reach instructor and TA. This section includes several efficient ways to do so. Generally,

- regarding assignments and grading result, contacting TA is preferred;
- regarding course materials, contacting instructor is preferred.

6.1 During Class

Discussion during class or right after class is always a direct way to communicate with instructor and TA. Technical questions about course materials are welcome.

If you have other concerns/requests (such as assignment extension requests, technical questions) about assignments and grading result, you can also talk with instructor directly right after class. In this case, a follow-up Message on Canvas is still necessary. See Section 7.6 for requesting extension of assignments, Section 7.7 for addressing your concerns about grading result.

6.2 Office Hour

Instructor and TA will create office hour appointments on Canvas Calendar, so it is convenient to make appointments to organize and participate in discussion during office hour. See Appendix A for how to sign up an appointment on Canvas Calendar.

A walk-in office hour is also possible, but not guaranteed, so signing-up a time slot on Canvas Calendar is highly recommended.

6.3 After Class And Office Hour

A preferred way to contact instructor and TA after class is to send a Message on Canvas to instructor and/or TA. Check Appendix A about how to send a Message on Canvas to a user in your course. Most Messages on Canvas will be replied within 3 workdays, so that is the recommended/preferred way to guarantee response on time. (Instructor and TA may sometimes miss your message during some busy period of a semester, so please send a second message if not getting response in 3 workdays).

An alternative way is to send an email to instructor and TA, but we still recommend using the preferred way for better organization of communication.

7 Grading Policy

7.1 Assignments (60% grade)

- HW1: ML, code + written, 10% of final grade
- HW2: NN, code + written, 10% of final grade
- HW3: CNN, code + written, 10% of final grade
- HW4: RNN, code, 10% of final grade
- HW5: GNN, code, 10% of final grade
- HW6: GAN, code, 10% of final grade

All coding assignments will be based on [Python](#), [Pytorch](#) and [Jupyter notebook](#). This course focuses on neural networks, and does not cover the particular topics about how to program in Python and how to use Pytorch and jupyter notebook. However, there are many good resources as guidelines for learning them. For example, [a tutorial for Python and numpy](#) provided by CS231n from Stanford University (or Google what you need, Appendix B).

When submitting code assignments:

For coding assignments, it is extremely important to save your running records and make sure they can be displayed in your submissions. Each time when submitting, double check you have uploaded (i) a .ipynb file for Jupyter notebook that can be re-produced by TA, and (ii) a .pdf or .html file converted from the .ipynb file that tracks all your running records to show that you have completely finished this assignment. Note that uploading these 2 files are sufficient, and zipping the 2 files is NOT needed.

7.2 Mid-Term Quiz (12% grade)

- Format: Canvas Quiz, online, 2-hour time limit. Open book: textbooks and slides can be used, but cannot search questions/answers on internet.
- Types of questions: multiple choice (choose one answer), multiple answers (choose at least one answer), ordering, True/False, etc.
- Number of questions: 24.
- Sections covered: ML, NN, CNN.
- Available time window: Feb 27, 8:00AM - Feb 28, 11:59PM, 2023.
- Grading method: Canvas auto grading.

7.3 Course Project (5% + 8% + 10% = 23% grade)

Course project is a team work. There can be *one, two, or three* (up to three) students in each team. There are 3 milestones for each course project, by which a report is required, i.e., project proposal, progress report and final report. Below is the key information regarding each of them.

- Project proposal (5% grade): (i) determine team members, (ii) decide what problem you are going to investigate, and (iii) submit a written proposal to describe your idea and plan.

Due date: Feb 19, 2023.

Length limit: 2 pages (title, abstract, main text) + unlimited reference page + unlimited appendix page.

Format: PDF (exported from Microsoft Word or compiled from LaTeX, e.g., [a LaTeX template](#)).

What should be included:

- Title: give a name to the project to highlight the significance, what contributions this project can have. The title should be short and relevant to your project.
 - Team members and student IDs. You still need to ensure that you create a group for your team and all team members have been included in one group. Otherwise, team members not included in the group will NOT receive credit. See Appendix [A](#) for creating group on Canvas.
 - Abstract: a paragraph that introduce key points, e.g., (i) why you choose this problem and this project, (ii) what unique advantage your project will have, (iii) a plan of the key steps for finishing this project, or anything else.
 - Introduction: a detailed explanation/elaboration of Abstract on the key points, e.g., one paragraph to elaborate one key point.
 - Literature review: any references that are important and relevant to the project, e.g., some techniques in the project that are inspired by other people's work including tutorials, Git repository, research publication, blogs, etc. Introduce more about what inspired the project.
 - Technical plan: a detailed elaboration of techniques used in this project (based on the techniques in Section "Literature review", if any). This plan can be described by a flowchart or any other figures.
 - Reference: if anything inspired your project or is relevant to your project, a citation is required.
- Progress report (8% grade): after deciding the project, each team starts their project. A progress report is required to elaborate how the project is going.

Due date: Mar 26, 2023.

Length limit: 4 pages (title, abstract, main text) + unlimited reference page + unlimited appendix page.

Format: PDF (exported from Microsoft Word or compiled from LaTeX, e.g., [a LaTeX template](#)).

What should be included:

- Title: give a name to the project to highlight the significance, what contributions this project can have. The title should be short and relevant to your project.

- Team members and student IDs. You still need to ensure that you create a group for your team and all team members have been included in one group. Otherwise, team members not included in the group will NOT receive credit. See Appendix A for creating group on Canvas.
- Abstract: a paragraph that introduce key points, e.g., (i) why you choose this problem and this project, (ii) what unique advantage your project will have, (iii) a plan of the key steps for finishing this project, or anything else.
- Introduction: a detailed explanation/elaboration of Abstract on the key points, e.g., one paragraph to elaborate one key point.
- Literature review: any references that are important and relevant to the project, e.g., some techniques in the project that are inspired by other people's work including tutorials, Git repository, research publication, blogs, etc. Introduce more about what inspired the project.
- Technical plan: a detailed elaboration of techniques used in this project (based on the techniques in Section "Literature review", if any). This plan can be described by a flowchart or any other figures.
- Intermediate results (NEW): following the plan, what has been done in the project? Elaborate what the team tried for finishing the plan, and what observation and knowledge the team acquired so far. Both theoretical analysis and experimental result can be regarded as the intermediate results.
- Future work (NEW): to ensure that the proposed plan can be done, what else work remaining for this project? Elaborate what the team plans to try. This section may modify the plan originally proposed in proposal, if any new understanding is established and some adjustment is necessary.
- Reference: if anything inspired your project or is relevant to your project, a citation is required.

Note that the progress report can re-use materials from the project proposal. New sections, i.e., intermediate results and future work must be included in progress report.

- Final report (10% grade): a report summarizing all the work a team has done, including theoretical or empirical results.

Due date: Apr 23, 2023.

Length limit: 6 to 8 pages (title, abstract, main text) + unlimited reference page + unlimited appendix page.

Format: PDF (exported from Microsoft Word or compiled from LaTeX, e.g., [a LaTeX template](#)).

What should be included:

- Title: give a name to the project to highlight the significance, what contributions this project can have. The title should be short and relevant to your project.
- Team members and student IDs. You still need to ensure that you create a group for your team and all team members have been included in one group. Otherwise, team

members not included in the group will NOT receive credit. See Appendix A for creating group on Canvas.

- Abstract: a paragraph that introduce key points, e.g., (i) why you choose this problem and this project, (ii) what unique advantage your project will have, (iii) a plan of the key steps for finishing this project, or anything else.
- Introduction: a detailed explanation/elaboration of Abstract on the key points, e.g., one paragraph to elaborate one key point.
- Literature review: any references that are important and relevant to the project, e.g., some techniques in the project that are inspired by other people's work including tutorials, Git repository, research publication, blogs, etc. Introduce more about what inspired the project.
- Technical plan: a detailed elaboration of techniques used in this project (based on the techniques in Section "Literature review", if any). This plan can be described by a flowchart or any other figures.
- Complete results (**NEW**): following the plan (or the modified plan after the progress report), what has been done in the project? Elaborate what the team tried for finishing the plan, and what observation and knowledge the team acquired so far. Both theoretical analysis and experimental result can be regarded as the results.
- Future work after this course (**NEW**): given what the team tried and learned from the project, is there any idea in the future that can be explored further?
- Reference: if anything inspired your project or is relevant to your project, a citation is required.

Note that the final project report can re-use materials from the project proposal and progress report. New sections, i.e., complete results and future work after this course must be included in progress report.

What can be *optionally* included:

- Github repository (a URL link) that includes a complete copy of your implementation code (in this way, your code will be open source under a selected license).
- Jupyter-lab notebook that is used to demo how your model can work and display experimental results.

How to select a project?

- Example reports Previous reports from other courses
 - CS229, Stanford: [Fall 2017](#), [Fall 2018](#), [Fall 2019](#), [Spring 2020](#), [Spring 2021](#) (report + poster), [suggested ideas](#)
 - CS221, Stanford: [Spring 2017](#) (titles only)
 - CS231n, Stanford: [Spring 2017](#), [Spring 2022](#) (report + poster)
- Examples from open competition on [Kaggle competitions](#): deep learning applications

- X-ray image classification/ranking: [CheXpert competition](#) from Stanford ML group
- Landmark Retrieval: [Google landmark retrieval competition 2021](#)
- Find extraterrestrial signals in data from deep space: [SETI Breakthrough Listen - E.T. Signal Search](#)
- Identify the category of foliar diseases in apple trees: [Plant Pathology 2021 - FGVC8](#)
- Or any other ideas

7.4 Attendance (5% grade)

In-class quizzes (randomly), for attendance check purposes. For absence, please see the [university policy for absences](#). Contact instructor (Section 6) to address your absence under this policy.

7.5 Late Work Grading

Late submissions can still be graded under some circumstances.

7.5.1 Written Assignments And Course Project

For written assignments or written part of assignments in the format of Canvas Quiz, and course project (proposal + progress report + final report), there is NO late submission policy, and the submission will be closed after due date. In other words, the due date for written assignments is firm.

7.5.2 Coding Assignments

For coding assignments or coding part of assignments, TA grader will use the following late work policy.

- After the scheduled due date, you will have a 2-calendar-day non-penalty period, allowing any variations of everyone's situation.
- After the 2-calendar-day non-penalty period, the maximum possible grade is decreased by 20% per day.
- Maximum 5-calendar-day late submission (i.e., 2-calendar-day non-penalty and 3-calendar-day penalty) is allowed. After that, assignment submission will be closed.
- Example 1: the scheduled due date is Feb 13, 23:59, PT. I submit my assignment at Feb 13, 23:58, PT (1 minute before the scheduled due date). The maximum possible grade I can get is 100 (full points). No action is taken.
- Example 2: the scheduled due date is Feb 13, 23:59, PT. I submit my assignment at Feb 15, 23:58, PT (1 minute before the end of two-day non-penalty period). The maximum possible grade I can get is 100 (full points). No action is taken. However, the late submission may influence my schedule for following assignments, which may bring me time constraint.

- Example 3: the scheduled due date is Feb 13, 23:59, PT. I submit my assignment at Feb 16, 00:03AM, PT (4 minutes after the end of three-day non-penalty period). The maximum possible grade I can get is 80 (= 100 - 20). If my submission originally gets 81 points out of 100, it will be capped at 80. If my submission originally gets 79 points out of 100, it will not be changed.
- Example 4: the scheduled due date is Feb 13, 23:59, PT. I submit my assignment at Feb 18, 23:59, PT (last minute of the six-day late submission). The maximum possible grade I can get is 40 (= 100 - 3*20). If my submission originally gets 81 points out of 100, it will be capped at 40.
- Example 5: the scheduled due date is Feb 13, 23:59, PT. I would like to submit my assignment at Feb 19, 00:01AM, PT. I found the submission has been closed, and my submission will not be graded.

7.6 Extension of Assignments

If more time for finishing an assignment is necessary, it is possible to get extension. For example, you have difficulty for finishing assignments on time due to health condition or other emergency condition. In this case, you can request extension with any supporting document such as doctor's appointment records, doctor's note, medical prescription, etc. Request without reasonable supporting document will not be considered.

How to request extension: (i) contact TA if your request is to extend less than 7 calendar days (including 7 calendar days), or (ii) contact both TA and instructor if your request is to extend more than 8 calendar days (including 8 calendar days).

See Section 6 for how to efficiently contact instructor and TA.

7.7 Grading Process

| time milestone | grading progress |
|------------------|--|
| due date | - |
| 1 week after due | graded, or technical issue: check TA's comment |
| 2 week after due | address technical issues in TA office hour |

Table 1: Grading timeline for an assignment.

TA is responsible to handle grading of assignments. The entire grading process should be within around 2 weeks after the due date. Most assignments will be graded within 1 week after the due date. See Table 1.

If the submission is still missing after the submission is closed (see Section 7.5 when submission will be closed), then this assignment will be graded as 0 credit.

If there is any issue for TA to grade your coding assignment during the 1st week after due date, TA will temporarily place 40% credit for your assignment and leave a comment to your submission on Canvas, so it is important to check your grade (at least double check your grade after 1 week of the due date) and respond to TA if anything is required for TA to finish grading. For example, if TA cannot re-produce your code (things can be complicated on different systems), an in-person demo is required, and TA will leave a comment on Canvas to your submission. This

| | |
|------------|----|
| 94-100.00% | A |
| 90-93.99% | A- |
| 86-89.99% | B+ |
| 82-85.99% | B |
| 78-81.99% | B- |
| 74-77.99% | C+ |
| 70-73.99% | C |
| 66-69.99% | C- |
| 62-65.99% | D+ |
| 58-61.99% | D |
| 0-57.99% | F |

Table 2: Grading scale to letter grade

in-person demo can then be done during a TA office hour (see Section 6.2 for how to sign up on Canvas Calendar).

If your submission has not been graded within 1 week after the due date, and there is no comment for the next step from TA, please contact TA grader (see how to in Section 6) immediately.

During the *2nd week* after due date, if you have concern about grading result, please leave a comment on Canvas submission, and meanwhile sign up a TA office hour (see Section 6.2) to address your concern. After the end of this week, grading result will be official.

7.8 Grading Scale to Letter Grade

Table 2 shows the grading scale from percentage to letter grade. Curving grade is possibly considered only if the final grade is very biased and skewed that cannot reflect the performance in this course.

8 Course Schedule

The more detailed course schedule can be found on Canvas "Course Schedule" page or "Home" page. The schedule is tentative and subject to change.

8.1 Class Meeting

Week 01, 01/09 - 01/13: Course overview, [ML]

- Syllabus
- ML basic concepts

Week 02, 01/16 - 01/20: [ML]

- PAC learning
- ERM, MLE

Week 03, 01/23 - 01/27: [NN]

- NN basics
- composition of functions

Week 04, 01/30 - 02/03: [NN]

- Units in NN, BP
- MLP softmax classifier

Week 05, 02/06 - 02/10: [CNN]

- Convolutional layer

Week 06, 02/13 - 02/17: [CNN]

- design CNN model
- CNN architectures, practical training for CNN

Week 07, 02/20 - 02/24: [RNN] + mid-term quiz

- Sequence data
- RNN
- LSTM

Week 08, 02/27 - 03/03: [RNN]

- RNN architecture
- Attention
- Transformer

Week 09, 03/06 - 03/10: [GNN]]

- Graph data in ML
- Graph representation

Week 10, 03/13 - 03/17: **Spring vacation****Week 11, 03/20 - 03/24:** [GNN]]

- GNN
- Over-smoothing

Week 12, 03/27 - 03/31: [GAN]

- Adversarial ML
- Generative model

Week 13, 04/03 - 04/07: [GAN]

- GAN
- Train GAN

Week 14, 04/10 - 04/14: [NAS]

- NAS

Week 15, 04/17 - 04/21: [OP]

- Contrastive learning
- Meta-learning

Week 16, 04/24 - 04/28: [OP]

- Federated learning
- Conformal prediction, diffusion model

8.2 Reading

ML – Book "Foundations of machine learning":

- * Chapter 1 Introduction
- * Chapter 2 The PAC learning framework: (Definitions are required to read. Theoretical results, examples and proofs are optional to read)
- * Chapter 3 Rademacher complexity and VC-dimension is optional to read
- * Chapter 4 Model selection: Read Section 4.1 and 4.2. Proof is optional to read

– Book "Deep learning": Section 5.5 Maximum Likelihood Estimation

NN – Book "Deep learning": Section 6 Deep forward networks

CNN – Book "Deep learning":

- * Section 9 Convolutional networks
- * 7.4 data augmentation
- * 7.1 parameter norm penalties
- * 7.2 norm penalties as constrained optimization
- * 7.3 regularization and under-constrained problem
- * 7.12 dropout

- Book "Dive into deep learning":
 - * 13.1 data augmentation
 - * 4.5 weight decay
 - * 4.6 dropout
- Reference papers mentioned by slides (note that only some sections of each paper are required to read. See below):
 - * [Alexnet] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems 25 (2012): 1097-1105. Conference proceeding version at [NeurIPS 2012 Abstract page](#) or [NeurIPS 2012 paper page](#) (Section 3.5 required, other sections are optional)
 - * [pyramid] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Spatial pyramid pooling in deep convolutional networks for visual recognition." IEEE transactions on pattern analysis and machine intelligence 37, no. 9 (2015): 1904-1916. [ArXiv page](#) (Section 2.2 required, other sections are optional)
 - * [NIN] Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." arXiv preprint arXiv:1312.4400(2013). [ArXiv page](#) (Section 3.2 required, other sections are optional)
 - * [Inception] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. 2015. [ArXiv page](#) (Section 4 and 5 required. Other sections optional)
 - * [ResNet] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016. [ArXiv page](#) (Section 3.1, 3.2 and 3.3 required. Other sections optional)
- RNN
 - * Book "Deep learning"
 - Chapter 10: Section 10.1 - 10.5, 10.7, 10.10
 - * optional external reading material:
 - [Blog: understanding LSTM](#)
 - [Self-attention lecture slides from Shusen Wang](#)
 - [CS231n lecture slides for RNN](#)

A Canvas Tutorials

- Send/reply Canvas Message: [this tutorial](#).
- Canvas appointment sign-up: [this tutorial](#).
- Create a group as a student on Canvas: [this tutorial](#).

B Tips for Coding Assignments

Python, Pytorch, Jupyter notebook are very commonly used tools for building deep neural networks. This course covers principles/rationales of deep neural networks, and does not cover how to use these tools to implement neural networks. 6 assignments will provide this chance to practice hands-on skills using these tools. If you have any specific issues regarding coding, [Google](#) and check out some online tutorials showcasing how these tools can work, such as [Python Numpy Tutorial \(with Jupyter and Colab\)](#) from Stanford CS231n.

C Academic Integrity Policy

We must ensure that academic honesty is upheld. Learn more about Plagiarism, Unauthorized Assistance, Fabrication by visiting [WSU's Academic Integrity Policy](#).

D Access Center

Let instructor know if you need special accommodations as early as possible. You may also visit [WSU's Access Center](#).