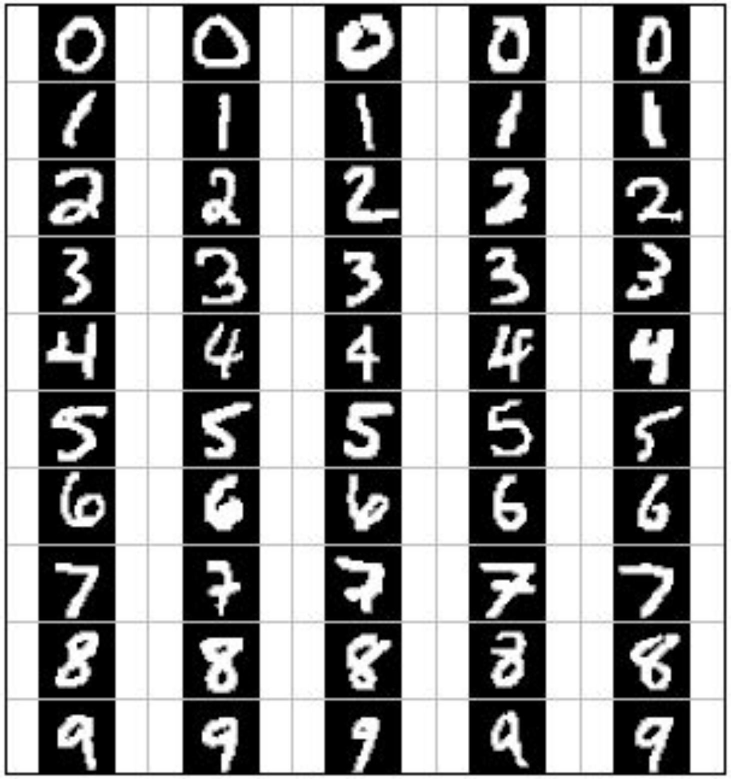
# Introduction to Artificial Intelligence
# Project 3 – Classification

Jianmin Li & Minlie Huang

Spring 2018

# Classification

- Training with MNIST



# Pre-requirements

- At least 400MB disk space and 800MB memory
- Python package that you should be familiar with
  - *numpy*
  - *sklearn*
- Python packages that should be installed:
  - *numpy*
  - *skimage*
  - *sklearn*
  - Anaconda is recommended

# Basic Tasks

- KNN (4 points)
  - Implement *KNNClassifier.classify* method in *classifiers.py*
  - *python dataClassifier.py –c knn –n 5*
- Softmax Regression (6 points)
  - Implement *PerceptronClassifier.train* in *classifiers.py*(5 points)
    - *python dataClassifier.py –c perceptron*
  - Visualize the weights (1 points)
    - Answer the question *answers.py* in the method *q3*, returning either *'a'* or *'b'*
- *sklearn* **MUST NOT BE USED** in the above two tasks, **OR you will not pass the autograder.**

# Basic Tasks

- Training SVM with *sklearn* (3 points)
  - Implement *SVMClassifier.train*, *SVMClassifier.classify* using package *sklearn*, in *classifiers.py*.
  - You should be familiar with some *sklearn* API
- Obtaining better classification results (2 points+ 1 point bonus)
  - Implement *BetterClassifier.train*, *BetterClassifier.classify* in *classifiers.py*.
  - Try to obtain good accuracy as much as you can.
  - 1 point extra credit for the leading classification accuracy

# Submission

- A 2-3 pages report (either Chinese or English)
  - Some analysis on the q3 in the report is useful for better grading
  - Some analysis on different algorithms/feature extractor techniques is useful for better grading
- Zip the files as the following structure
  - student_id.zip (e.g. 20090112xx.zip)
    - student_id.pdf
    - classifiers.py
    - answer.py

# Due

- 2018/5/22 23:59:59
- Correctness of algorithms (80%)
- Report (20%)