Homework 12: Neural Networks

Kaveh Fathian, Email: fathian@ariarobotics.com

Handout: 2025-12-01

Due: 2025-12-8, 11:59pm, on Canvas

General Instructions:

- You should solve the homework and submit your report **individually**. Identical submissions will receive a grade of zero.
- Getting help from others or checking your answers with other students (not the TAs) is okay and encouraged.
- Ask any questions on **Ed Discussion** (instead of emailing).
- **Before** the homework due date, TAs are strictly prohibited from **pre-grading** your homework. Do not expect the TAs to help you verify if your answers are correct or give you the problem solution.
- **After** the homework due date, if you do not know how to solve a problem, reach out to the TAs. They will walk you through the solution and help you understand it. Note that homework solutions will **not** be posted because some problems will be used in next year's class.
- **Exams** may contain questions related to homework, so make sure you learn how to solve the homework problems correctly.
- The deliverables are outlined for each problem, and you should carefully **follow the instructions**. Failing to follow instructions will result in **points being subtracted**.
- You will submit a **single PDF** file to Canvas as your homework report. The PDF must contain your **answers** and any requested **outputs** (e.g., printouts, snapshots of code, or GUIs). If requested, follow the instructions specified by the problem to provide your **code** (e.g., in a compressed .zip or .tar file) in addition to the PDF file.
- **Grading:** Each homework in this class will contribute **5pts** to your final grade (there will be 12 homework assignments, each 5pts, leading to 60pts for all assignments). A detailed grading **rubric** will be posted on **Canvas** after the homework due date. Any bonus points will be added to your overall course bonus points, which will be added to your final grade.
- **Late submission:** Late or missed submission will not be accepted and will receive a grade a zero. Any excused absence must be documented and disclosed to the instructor (extensions will be granted on a case-by-case basis). Three or more missed homework lead to an INC grade.

Homework 12: Neural Networks

Kaveh Fathian, Email: fathian@ariarobotics.com

Handout: 2025-12-01

Due: 2025-12-8, 11:59pm, on Canvas

EXERCISE 1 (5pts) – In this assignment, you will train a neural network to classify handwritten digits in the MNIST dataset. MNIST contains 70,000 images of handwritten digits (0-9), split into 60,000 images for training, and 10,000 images for testing. Do *not* train on the test set! You may use the code in hw12_template.ipynb, provided in the course GitHub repository, to load the dataset. You must follow the instructions below to receive full credit:

Data Preparation:

- Each MNIST image is 28 × 28 pixels. Flatten each image into a 784-dimensional vector before feeding it to the network.
- Conver each digit label into a "one-hot" encoded label, and **explain** what the one-hot label is and why is it used. You may use tensorflow.keras.utils.to_categorical(y, 10) for one-hot label conversion.

Neural Network Architecture: You *must* implement the following architecture:

- Input layer: 784-dimensional flattened image.
- Hidden layer: Fully connected (dense) layer with **128 neurons** and **ReLU** activation.
- Output layer: Fully connected (dense) layer with **10 neurons** and **softmax** activation.

Training:

- Loss function: Use Categorical Cross-Entropy loss function $L(y, \hat{y}) = \sum_i y_i \log(\hat{y}_i)$, which is implemented in tensorflow.keras.losses.categorical_crossentropy(). See the example here.
- You must use the **Adam** optimizer.
- Train the network for a minimum of **10 epochs** using **batch size of 64** in the mini-batch gradient descent.

Required Deliverables:

- Printout of your entire code
- Explanation of "***one-hot***" encoded labels
- Plot of the ***loss function vs. training epochs***
- Compute & print out the prediction ***accuracy*** on the ***testing*** dataset
- Compute & print out the ***confusion matrix***