CMPUT 261, Winter 2023
Assignment #3

Due: Thursday, March 23/2023
Total points: 99

For this assignment use the following consultation model:

1. you can discuss assignment questions and exchange ideas with other *current* CMPUT 261 students;
2. you must list all members of the discussion in your solution;
3. you may **not** share/exchange/discuss written material and/or code;
4. you must write up your solutions individually;
5. you must fully understand and be able to explain your solution in any amount of detail as requested by the instructor and/or the TAs.

Anything that you use in your work and that is not your own creation must be properly cited by listing the original source. Failing to cite others' work is plagiarism and will be dealt with as an academic offence.

First name: Dipesh

Last name: Patel

CCID: dipesh1@ualberta.ca

---

1. (**Neural Networks**; 54 points) The MNIST dataset is a set of images of handwritten digits labelled by their actual digit. We will operate on two versions of this dataset: one with the images shifted 2 pixels to the upper-left, and one with the images shifted 2 pixels to the bottom-right.

   This question requires the use of PyTorch. You may need to install PyTorch using the following command:

   ```
   pip3 install --user torch torchvision
   ```

   Torch idiosyncrasies. In class, we specified the units in a dense feedforward network as performing a linear combination followed by an activation function application (e.g., relu of linear combination). However, PyTorch treats these operations as separate layers. See the implementation of `MLP1` for an example: The hidden layer is specified as a `Linear` layer followed by a `ReLU` layer. Similarly, to specify a convolutional layer with ReLU activation, you first specify a `Conv2d` layer followed by a `ReLU` layer.

   (a) [**10 points**] Implement a fully-connected feed-forward neural network for classifying MNIST images according to the digit that they represent by editing the `MLP2` class in the provided `cnn.py` file.

   The network should have two hidden layers: one dense linear layer with 128 rectified linear ('relu') activation, and one with 64 rectified linear units. The output should be a fully-connected linear layer of 10 units with the softmax activation. The `cnn.py` file contains an example implementation of a network with a single hidden layer in the `MLP1` class that you may template from. The `main` function will test your program for you; you may add any tests that you like.

   We will run your code by importing the `cnn` module and training and evaluating the model implemented by `MLP2`, so it is important that your code follow these naming conventions.

   Submit all of your code for this question (including provided boilerplate files) in a single zip file.

   (b) [**30 points**] Implement a convolutional neural network for classifying MNIST images by editing the `CNN` class in the provided `cnn.py` file.

   The network should have the following architecture:

   - A layer of 32 convolutional units with a kernel size of 5×5 and a stride of $1, 1$, with relu activation
   - A max-pooling layer with a pool size of 2×2 and a stride of $2, 2$.
   - A layer of 64 convolutional units with a kernel size of 5×5 and the default stride, with relu activation.
   - A max-pooling layer with a pool size of 2×2 and the default stride.
   - A `Flatten` layer (to reshape the image from a 2D matrix into a single long vector)
   - A layer of 512 fully-connected linear units with relu activation
   - A layer of 10 fully-connected linear units with log-softmax activation (the output layer). This gives an output of log-probabilities.

   Submit all of your code for this question (including provided boilerplate files) in a single zip file.

(c) **[2 points]** What was the accuracy of your trained 2-hidden-layer feedforward network on the two test sets?

accuracies : tst1 = 0.9671

tst2 = 0.5266

(d) **[2 points]** What was the accuracy of your trained convolutional neural network on the two test sets?

accuracies : tst1 = 0.9913

tst2 = 0.7953

(e) **[10 points]** Did one of your implementations perform substantially better on one of the test sets than the other implementation did? If so, why? If not, why not?

CNN performed substantially better than two hidden layers feedforward network.

Because CNN also use two new additional operations which are pooling and convolutions.

2. (**Bayesian Learning**; 14 points)

Suppose that you have three models $(\theta_1, \theta_2, \theta_3)$ of changes in the price of a single stock. You know that one of these models is the true model. Each model gives the probability that tomorrow's price will be higher than today's price (y_{t+1}), based on whether today's price was higher than the price the day before (y_t). So you can make money on average by buying the stock when $p(y_{t+1} \mid y_t, \theta^*) > .5$ and selling when $p(y_{t+1} \mid y_t, \theta^*) < .5$, where θ^* is the true model.

Your prior belief is that θ_1 and θ_2 are equally likely to be true, and θ_3 is three times more likely than θ_1 to be true.

You have a dataset D of past observations, and you have computed that

$$p(D \mid \theta_1) = .0210$$
$$p(D \mid \theta_2) = .0168$$
$$p(D \mid \theta_3) = .0014.$$

(a) **[6 points]** What are the posterior probabilities of each model being the true model?

we have $\theta_1 = \theta_2$ and $\theta_3 = 3\theta_1$

$$P(\theta_1 \mid D) = \frac{P(D\mid\theta_1)\cdot \cancel{P(\theta_1)}}{\cancel{P(\theta_1)}\, P(D\mid\theta_1) + P(D\mid\theta_2) + 3P(D\mid\theta_3)} = \frac{0.0210}{0.0210 + 0.0168 + 3\times 0.0014} = 0.5$$

$$P(\theta_2 \mid D) = \frac{P(D\mid\theta_2)\cdot \cancel{P(\theta_2)}}{\cancel{P(\theta_2)}\, P(D\mid\theta_1) + P(D\mid\theta_2) + 3P(D\mid\theta_3)} = \frac{0.0168}{0.0210 + 0.0168 + 3\times 0.0014} = 0.4$$

$$P(\theta_3 \mid D) = \frac{3P(D\mid\theta_3)\cdot \cancel{P(\theta_3)}}{\cancel{P(\theta_3)}\, P(D\mid\theta_1) + P(D\mid\theta_2) + 3P(D\mid\theta_3)} = \frac{0.0014\times 3}{0.0210 + 0.0168 + 3\times 0.0014} = 0.1$$

(b) [**4 points**] Now suppose that you run each model, and they make the following predictions:

$$p(y_{t+1} \mid y_t, \theta_1) = .4$$
$$p(y_{t+1} \mid y_t, \theta_2) = .75$$
$$p(y_{t+1} \mid y_t, \theta_3) = .6.$$

What is the maximum a posterior estimate for $p(y_{t+1} \mid y_t)$? Based on the MAP estimate, would you be better off buying or selling?

From part (a) we got $P(\theta_1|D) = 0.5$ $P(\theta_2|D) = 0.4$, $P(\theta_3|D) = 0.1$

As $0.5 > 0.4 > 0.1$

Thus maximum posterior estimate for $P(y_{t+1}|y_t)$ is $P(y_{t+1}|y_t,\theta_1)$

As $P(y_{t+1}|y_t,\theta_2) = 0.4 < 0.5$. Based on MAP,

$\therefore$ selling stock is better

(c) [**4 points**]
What is the estimate according to the posterior predictive distribution for $p(y_{t+1} \mid y_t)$? (I.e., using model averaging.) Based on the PPD, would you be better off buying or selling?

$$P(Y|D) = \sum_{\theta} P(Y|\theta)\,P(\theta|D)$$

$$= P(y_{t+1}|y_t,\theta_1)\times 0.263 + P(y_{t+1}|y_t,\theta_2)\times 0.211 + P(y_{t+1}|y_t,\theta_3)\times 0.526$$

$$= (0.4)(0.5) + (0.75)(0.4) + (0.6)(0.1)$$

$$= 0.2 + 0.3 + 0.06$$

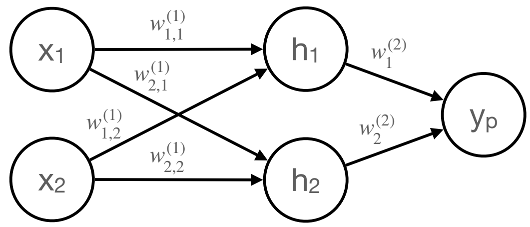
$$= 0.56$$

As $0.56 > 0.5$

$\therefore$ Buying is better

3. (**Automatic Differentiation**; 31 points)

   Consider the following feedforward neural network:



- Nodes x_1 and x_2 are input nodes.
- Nodes h_1, h_2, and y_p are rectified linear units; i.e., they use the activation $g(t) = \max\{0, t\}$.
- The inputs to h_1 are weighted by $w_{1,1}^{(1)}$ and $w_{1,2}^{(1)}$; h_2's inputs have weights $w_{2,1}^{(1)}$ and $w_{2,2}^{(1)}$
- The inputs to y_p are weighted by $w_1^{(2)}$ and $w_1^{(2)}$. (Note that the output node is called y_p to distinguish it from the "true" value of y in a training example)
- There are no offsets in this network; the weights are the only parameters

(a) [**5 points**] Let $y_p = f\left(x_1, x_2; w_{1,1}^{(1)}, w_{1,2}^{(1)}, w_{2,1}^{(1)}, w_{2,2}^{(1)}, w_1^{(2)}, w_1^{(2)}\right)$ be the prediction of the neural network on an input x_1, x_2. Give an expression for f.

$$y_p = g\left[w_1^{(2)}\cdot\left(g\left(w_{1,1}^{(1)}\cdot x_1 + w_{1,2}^{(1)}\cdot x_2\right)\right) + w_2^{(2)}\left(g\left(w_{2,1}^{(1)}\cdot x_1 + w_{2,2}^{(1)}\cdot x_2\right)\right)\right]$$

$$= \max\left[0, w_1^{(2)}\cdot\left(\max\left(0,(w_{1,1}^{(1)}x_1 + w_{1,2}^{(1)}x_2)\right)\right) + w_2^{(2)}\cdot\left(\max\left[0, (w_{2,1}^{(1)}x_1 + w_{2,2}^{(1)}x_2)\right)\right)\right]$$

(b) [**3 points**] Let $L\left(w_{1,1}^{(1)}, w_{1,2}^{(1)}, w_{2,1}^{(1)}, w_{2,2}^{(1)}, w_1^{(2)}, w_1^{(2)}\right)$ be the mean-squared error of f on a single example (x_1, x_2, y). Give an expression for L.

$$L = \frac{1}{n}\sum(\text{true} - \text{predicate})^2$$

$$= \frac{1}{1}\left[y_{true} - \max\left[0, w_1^{(2)}\cdot\left(\max\left(0,(w_{1,1}^{(1)}x_1 + w_{1,2}^{(1)}x_2)\right)\right) + w_2^{(2)}\cdot\left(\max\left[0, (w_{2,1}^{(1)}x_1 + w_{2,2}^{(1)}x_2)\right)\right)\right]\right]^2$$

(c) [**10 points**] Give a *finite numerical algorithm* for computing L.

$s_1 = x_1$
$s_2 = x_2$
$s_3 = w_{1,1}^{(1)}$
$s_4 = w_{1,2}^{(1)}$
$s_5 = w_{2,1}^{(1)}$
$s_6 = w_{2,2}^{(1)}$
$s_7 = w_1^{(2)}$
$s_8 = w_2^{(2)}$
$s_9 = y_{true}$

$s_{10} = s_3 \cdot s_1$
$s_{11} = s_4 \cdot s_2$
$s_{12} = s_{10} + s_{11}$
$s_{13} = \max(0, s_{12})$
$s_{14} = s_7 \cdot s_{13}$
$s_{15} = s_5 \cdot s_1$
$s_{16} = s_6 \cdot s_2$
$s_{17} = s_{15} + s_{16}$
$s_{18} = \max(0, s_{17})$
$s_{19} = s_8 \cdot s_{18}$

$s_{20} = s_{14} + s_{19}$
$s_{21} = \max(0, s_{20})$
$s_{22} = s_9 - s_{21}$
$s_{23} = s_{22} \cdot s_{22}$

(d) **[13 points]** Give a *finite numerical algorithm* for computing the gradient of L with respect to $\left(w_{1,1}^{(1)}, w_{1,2}^{(1)}, w_{2,1}^{(1)}, w_{2,2}^{(1)}, w_1^{(2)}, w_1^{(2)}\right)$. You need not re-define the intermediate steps for the forward pass defined in (3c).

$$\bar{s}_{23} = \frac{\partial s_{23}}{\partial s_{23}} = 1$$

$$\bar{s}_{22} = \frac{\partial s_{23}}{\partial s_{22}} = \bar{s}_{23} \cdot 2 s_{22}$$

$$\bar{s}_{21} = \frac{\partial s_{23}}{\partial_{21}} = \frac{\partial s_{23}}{\partial s_{22}} \cdot \frac{\partial s_{22}}{\partial s_{21}} = \bar{s}_{22} \cdot (-1)$$

$$\bar{s}_{20} = \frac{\partial s_{23}}{\partial s_{20}} = \begin{cases} \bar{s}_{21} & \text{if } s_{20} > 0 \\ 0 & \text{if } s_{20} \leq 0 \end{cases}$$

$$\bar{s}_{19} = \frac{\partial s_{23}}{\partial s_{19}} = \frac{\partial s_{23}}{\partial s_{20}} \cdot \frac{\partial s_{20}}{\partial s_{19}} = \bar{s}_{20} \cdot 1$$

$$\bar{s}_{18} = \frac{\partial s_{23}}{\partial s_{18}} = \frac{\partial s_{23}}{\partial s_{19}} \cdot \frac{\partial s_{19}}{\partial s_{18}} = \bar{s}_{19} \cdot s_8$$

$$\bar{s}_{17} = \frac{\partial s_{23}}{\partial s_{20}} = \begin{cases} \bar{s}_{18} & \text{if } s_{17} > 0 \\ 0 & \text{if } s_{17} \leq 0 \end{cases}$$

$$\bar{s}_{16} = \frac{\partial s_{23}}{\partial s_{16}} = \frac{\partial s_{23}}{\partial s_{17}} \cdot \frac{\partial s_{17}}{\partial s_{16}} = \bar{s}_{17} \cdot 1$$

$$\bar{s}_{15} = \frac{\partial s_{23}}{\partial s_{15}} = \frac{\partial s_{23}}{\partial s_{17}} \cdot \frac{\partial s_{17}}{\partial s_{15}} = \bar{s}_{17} \cdot 1$$

$$\bar{s}_{14} = \frac{\partial s_{23}}{\partial s_{14}} = \frac{\partial s_{23}}{\partial s_{20}} \cdot \frac{\partial s_{20}}{\partial s_{14}} = \bar{s}_{20} \cdot 1$$

$$\bar{s}_{13} = \frac{\partial s_{23}}{\partial s_{13}} = \frac{\partial s_{23}}{\partial s_{14}} \cdot \frac{\partial s_{14}}{\partial 13} = \bar{s}_{14} \cdot s_7$$

$$\bar{s}_{12} = \frac{\partial s_{23}}{\partial s_{12}} = \begin{cases} \bar{s}_{13} & \text{if } s_{12} > 0 \\ 0 & \text{if } s_{12} \leq 0 \end{cases}$$

$$\bar{s}_{11} = \frac{\partial s_{23}}{\partial s_{11}} = \frac{\partial s_{23}}{\partial_{12}} \cdot \frac{\partial s_{12}}{\partial s_{11}} = \bar{s}_{12} \cdot 1$$

$$\bar{s}_{10} = \frac{\partial s_{23}}{\partial s_{10}} = \frac{\partial s_{23}}{\partial s_{12}} \cdot \frac{\partial s_{12}}{\partial s_{10}} = \bar{s}_{12} \cdot 1$$

$$\bar{S}_9 = \frac{\partial S_{23}}{\partial S_9} = \frac{\partial S_{23}}{\partial S_{22}} \cdot \frac{\partial S_{22}}{\partial S_9} = \bar{S}_{22} \cdot 1$$

$$\bar{S}_8 = \frac{\partial S_{23}}{\partial S_8} = \frac{\partial S_{23}}{\partial S_{19}} \cdot \frac{\partial S_{19}}{\partial S_8} = \bar{S}_{19} \cdot S_{18}$$

$$\bar{S}_7 = \frac{\partial S_{23}}{\partial S_7} = \frac{\partial S_{23}}{\partial S_{14}} \cdot \frac{\partial S_{14}}{\partial 7} = \bar{S}_{14} \cdot S_{13}$$

$$\bar{S}_6 = \frac{\partial S_{23}}{\partial S_6} = \frac{\partial S_{23}}{\partial S_{16}} \cdot \frac{\partial S_{16}}{\partial S_6} = \bar{S}_{16} \cdot S_2$$

$$\bar{S}_5 = \frac{\partial S_{23}}{\partial S_5} = \frac{\partial S_{23}}{\partial S_{15}} \cdot \frac{\partial S_{15}}{\partial S_5} = \bar{S}_{15} \cdot S_1$$

$$\bar{S}_4 = \frac{\partial S_{23}}{\partial S_4} = \frac{\partial S_{23}}{\partial S_{11}} \cdot \frac{\partial S_{11}}{\partial S_4} = \bar{S}_{11} \cdot S_2$$

$$\bar{S}_3 = \frac{\partial S_{23}}{\partial S_3} = \frac{\partial S_{23}}{\partial S_{10}} \cdot \frac{\partial S_{10}}{\partial S_3} = \bar{S}_{10} \cdot S_1$$

$$\bar{S}_2 = \frac{\partial S_{23}}{\partial S_2} = \frac{\partial 23}{\partial S_{11}} \cdot \frac{\partial S_{11}}{\partial S_2} + \frac{\partial 23}{\partial S_{16}} \cdot \frac{\partial S_{16}}{\partial S_2} = \bar{S}_{11} \cdot S_4 + \bar{S}_{16} \cdot S_6$$

$$\bar{S}_1 = \frac{\partial S_{23}}{\partial S_1} = \frac{\partial S_{23}}{\partial S_{10}} \cdot \frac{\partial S_{10}}{\partial S_1} + \frac{\partial 23}{\partial S_{15}} \cdot \frac{\partial S_{15}}{\partial S_1} = \bar{S}_{10} \cdot S_3 + \bar{S}_{15} \cdot S_5$$

Submission

The assignment you downloaded from eClass is a single ZIP archive which includes this document as a PDF *and* its LaTeX source as well as a Python file needed for Question 1. You are to unzip the archive into an empty directory, work on the problems and then zip the directory into a new single ZIP archive for submission.

Each assignment is to be submitted electronically via eClass by the due date. **Your submission must be a single ZIP file containing**:

1. a single PDF file with your answers;
2. file(s) with your Python code.

To generate the PDF file with your answers you can do any of the following:

- insert your answers into the provided LaTeX source file between `\begin{answer}` and `\end{answer}`. Then run the source through LaTeX to produce a PDF file;
- print out the provided PDF file and legibly write your answers in the blank spaces under each question. Make sure you write as legibly as possible for we cannot give you any points if we cannot read your hand-writing. Then scan the pages and include the scan in your ZIP submission to be uploaded on eClass;
- use your favourite text processor and type up your answers there. Make sure you number your answers in the same way as the questions are numbered in this assignment.