IE643 Deep Learning : Theory and Practice **July-Dec 2021**

Assignment 1: Due On 3^{rd} September 2021 (11:59 PM IST)

1 Instructions

We make the following assumptions for all questions: a vector $u \in \mathbb{R}^d$ for some $d \geq 1$, is represented as $\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_d \end{pmatrix}$; a vector $u \in \mathbb{R}^d$ with $u_i = 0 \; \forall i = 1, \ldots, d$ is called a zero vector and is represented by $\mathbf{0}$; the notation $|\alpha|$ denotes the absolute value of some $\alpha \in \mathbb{R}$.

Answer all questions. Write your answers clearly. You can score a maximum of 56 marks in this assignment.

Please make sure that all your answers are present in a single pdf document. If you use python notebook (.ipynb) files, make sure that your answers and plots are visible in .ipynb file. Upload on moodle, the python code, plots and pdf document as a single zip file named as "IE643_rollno_assignment1.zip". All your files within the zip file should follow similar naming convention. There will be no extensions to the submission deadline.

The link for all related files used in Assignment 1 is provided in Moodle/MS Teams.

Note: The questions which need to be answered as part of the assignment are provided in Section 2. There are some practice questions in Section 3 which need not be submitted as part of the assignment.

2 Assignment Questions (Solutions need to be submitted)

1. (a) **[5 marks]** Recall that a hyperplane $H = (w, b)$ for some $w \neq \mathbf{0} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ is defined as $H = \{x \in \mathbb{R}^d : \langle w, x \rangle = b\}$. Show that for every hyperplane H and for every $\beta > 0$, there exists another hyperplane $\tilde{H} = (\tilde{w}, \tilde{b})$ such that $\|\tilde{w}\|_2 = \beta$. Illustrate the relationship between (w, b) and $(\tilde{w}, \tilde{b})$. (Recall that for $u \in \mathbb{R}^d$, $\|u\|_2 = \sqrt{\langle u, u \rangle} = \sqrt{\sum_{i=1}^{d} |u_i|^2}$ is the ℓ_2 norm of u).

   (b) **[5 marks]** Consider a data set $D = \{(x^1, y^1), \ldots, (x^n, y^n)\}$, where $x^j \in \mathbb{R}^d$, $y^j \in \{+1, -1\}, \forall j = 1, 2, \ldots, n$. Let $\max_j \|x^j\|_2 \leq R$. Recall that D is linearly separable if there exist $w^* \in \mathbb{R}^d$ and $\gamma > 0$ such that $y^j \langle w^*, x^j \rangle \geq \gamma, \; \forall j = 1, \ldots, n$. Show that if D is linearly separable, the mistake bound proved in class

$$M \leq \frac{R^2 \|w^*\|_2^2}{\gamma^2}$$

   can be rewritten simply as $M \leq \frac{R^2}{\eta^2}$, where $\eta > 0$ (which might be same as γ or different from γ). (**Hint: Use the previous result about hyperplane.**)

2. Consider a data set $D = \{(x^1, y^1), \ldots, (x^n, y^n)\}$, where $x^j \in \mathbb{R}^d$, $y^j \in \{0, 1\}, \forall j = 1, 2, \ldots, n$. Suppose that you are not allowed to change the data set features and labels explicitly.

(a) **[2 marks]** Come up with a suitable condition to check if data set D is linearly separable or not.

(b) **[8 marks]** Modify the perceptron learning algorithm minimally so that you can train the perceptron with data set D. Note that using a simple conditional statement to convert the labels into $+1$ and -1 is not recommended. Implement your algorithm in **Python** (using the perceptron code template posted in drive) and highlight the changes you made with appropriate comments. Execute your code on a suitable linearly separable data set with maximum 200 data points in each class (classes 0 and 1) and plot the behavior of the separating hyperplane during the execution.

3. **[Use only Python]** Take the feed-forward network code file `FeedForwardNet(Assignment 1).ipynb` posted in drive. Check that the code uses simple squared error (SE) and cross entropy (CE) loss functions. Also note that the code currently has support for only the logistic sigmoid activation function. Answer the following:

   (a) The tanh activation function is given by $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$. Implement the python functions to compute $\tanh(z)$ and its gradient. [**2 marks**]

   (b) The linear activation function is given by $\text{linear}(z) = z$. Implement the python functions to compute $\text{linear}(z)$ and its gradient. [**2 marks**]

   (c) The ReLU activation function is given by $\text{ReLU}(z) = \max\{z, 0\}$. Implement the python functions to compute $\text{ReLU}(z)$ and its (sub-)gradient. [**2 marks**]

   (d) Consider an appropriate neural network architecture where each layer has only logistic sigmoidal activation functions. Illustrate the exploding gradient and vanishing gradient problems in this network. Justify the architecture you used, indicate how you checked the exploding gradient and vanishing gradient problems, and explain your observations. [**2 marks**]

   (e) Consider an appropriate neural network architecture where each layer has only tanh activation functions. Illustrate the exploding gradient and vanishing gradient problems in this network. Justify the architecture you used, indicate how you checked the exploding gradient and vanishing gradient problems, and explain your observations. [**2 marks**]

   (f) Consider an appropriate neural network architecture where each layer has only ReLU activation functions. Illustrate the exploding gradient and vanishing gradient problems in this network. Justify the architecture you used, indicate how you checked the exploding gradient and vanishing gradient problems, and explain your observations. [**2 marks**]

   (g) Among the three networks considered in questions 3d, 3e, 3f, describe which networks were more prone to the vanishing gradient issue and which networks were more prone to the exploding gradient issue. Use appropriate justifications for your observations involving the quantities used during backpropagation. [**3 marks**]

4. **[Use only Python]** Take the feed-forward network code file `FeedForwardNet(Assignment 1).ipynb` posted in drive. Consider the IMDB data set posted in drive. The data set features are available in `IMDB_feats.txt` and corresponding labels are in `IMDB_labels.txt`.

   Data format for features: Note that the features in file `IMDB_feats.txt` are stored in the format idx_1:val_1 idx_2:val_2 idx_3:$\text{val}_3 \ldots \text{idx}_k$:$\text{val}_k$ where idx_j denotes a feature index and val_j denotes the feature value corresponding to the feature index idx_j. For example, if m-th line of file `IMDB_feats.txt` is of the form 0:1 5:1 998:1 it indicates that m-th sample has zero-th feature with a value 1, fifth feature with value 1 and 998-th feature with value 1 and all other features have zero value. Note that feature indexing starts from zero.

   Data format for labels: Similar to the features, note that the labels in file `IMDB_labels.txt` are stored in the format idx_1:val_1 idx_2:val_2 idx_3:$\text{val}_3 \ldots \text{idx}_k$:$\text{val}_k$ where idx_j denotes a label index and

val_j denotes the label value corresponding to the label index idx_j. For example, if m-th line of file `IMDB_labels.txt` is of the form 0:1 3:1 27:1 it indicates that the m-th sample is associated with labels $0, 3$ and 27. Note here too that label indexing starts from zero. Also realize that the data corresponds to a multi-label setting.

(a) Write code to read this data into suitable `numpy` arrays. [**2 marks**]

(b) Write the required code to shuffle and split the data set into three sets S_1, S_2 and S_3 such that S_1 contains 70% of the data, S_2 contains 15% of the data and S_3 contains 15% of the data. [**2 marks**]

(c) Design a single feed forward neural network and a corresponding loss function to perform training on the IMDB data set. Justify the design choice of your neural network and loss function and implement the loss function in the code. [**3 marks**]

(d) Illustrate how you will carry out backpropagation for the new loss function in the last layer. Include its implementation in code. [**3 marks**]

(e) Design a suitable performance metric called `MLperf` (similar to **accuracy** discussed for binary and multi-class classification) which can measure how good the predictive power of the network is on a multi-labelled data set. Explain your performance metric and implement it in code. [**3 marks**]

(f) For the chosen loss function, choose the learning rates from the set $\{0.1, 0.01, 0.001, 10^{-4}, 10^{-5}\}$ and mini-batch sizes from $\{10, 20, 50\}$. For each (learning rate, mini-batch size) pair, run the mini-batch stochastic gradient descent algorithm on S_1, with 300 epochs. For every 5 epochs, record the loss and `MLperf` achieved on the sets S_1 and S_2. Now plot the loss for every 5 epochs for each (learning rate, mini-batch size) pair on S_2 (use a single plot and different colors for different pairs). Similarly plot the `MLperf` for every 5 epochs for each (learning rate, mini-batch size) pair on S_2 (use a single plot and different colors for different pairs). Can you come up with a suitable selection procedure for the best (learning rate, mini-batch size) pair using the experiments conducted? Explain your selection procedure and justify. [**4 marks**]

(g) Using the best (learning rate, mini-batch size) pair identified above, conduct training using mini-batch SGD on the set $S_1 \bigcup S_2$ with max epochs set to 300. For every 5 epochs, record the loss and `MLPerf` achieved on the sets $S_1 \bigcup S_2$ and S_3. Include a stopping condition such that you can stop the training when the loss on the set $S_1 \bigcup S_2$ does not decrease significantly for p epochs with a suitable choice for p. Plot the loss on $S_1 \bigcup S_2$ and S_3 in a single plot and comment on the observations. Similarly plot `MLPerf` on $S_1 \bigcup S_2$ and S_3 in a single plot and comment on the observations. [**4 marks**]

3 Practice Questions (Not for submission)

- The inner product (or dot product or scalar product) between two vectors $u, v \in \mathbb{R}^d$ is defined as $\langle u, v\rangle = \sum_{i=1}^{d} u_i v_i$. The following properties are related to the inner product:
  1. Prove $\langle u, v\rangle = \langle v, u\rangle, \forall u, v \in \mathbb{R}^d$.
  2. Prove $\langle u, v + w\rangle = \langle u, v\rangle + \langle u, w\rangle, \forall u, v, w \in \mathbb{R}^d$.
  3. Prove $\langle \alpha u, v\rangle = \alpha\langle u, v\rangle, \forall u, v \in \mathbb{R}^d, \forall \alpha \in \mathbb{R}$.
- A vector norm defined on $\mathbb{R}^d$ is a function $\|\cdot\| : \mathbb{R}^d \longrightarrow [0, +\infty]$, which satisfies the following properties:
  - **[Non-negativity]** $\|u\| \geq 0, \forall u \in \mathbb{R}^d$ and $\|u\| = 0$ if and only if $u = \mathbf{0}$ is the zero vector.

- **[Absolute scaling]** $\|\alpha u\| = |\alpha|\|u\|, \forall u \in \mathbb{R}^d, \forall \alpha \in \mathbb{R}$.
- **[Triangle inequality]** $\|u + v\| \leq \|u\| + \|v\|, \forall u, v \in \mathbb{R}^d$.

The following questions are related to vector norms.

1. Consider $\|u\|_2 = \sqrt{\langle u, u \rangle} = \sqrt{\sum_{i=1}^{d} u_i^2}$. Show that $\|\cdot\|_2$ is a vector norm. You must verify all the three properties described above. (Note that this is the popular Euclidean norm and is induced by the inner product definition given in the previous question.)
2. Consider $\|u\|_1 = \sum_{i=1}^{d} |u_i|$. Show that $\|\cdot\|_1$ is a vector norm.
3. Consider $\|u\|_p = \left[\sum_{i=1}^{d} (u_i)^p\right]^{\frac{1}{p}}$, where $p > 2$. Show that $\|\cdot\|_p$ is a vector norm.
4. Consider $\|u\|_\infty = \max\{|u_1|, |u_2|, \ldots, |u_d|\}$. Show that $\|\cdot\|_\infty$ is a vector norm.

- Prove the Cauchy-Schwarz inequality $|\langle u, v \rangle| \leq \|u\|_2 \|v\|_2, \ \forall u, v \in \mathbb{R}^d$.
