Problem Set 3

Oke

CEE 616: Probabilistic Machine Learning *10.18.2025*

Due October 25, 2025 at 11:59PM. Submit on Moodle.

The standard problems are worth a total of **53 points**.

Problem 1 *Batch learning (5 pts)*

In training a neural network via batch learning, the weight updates are given by:

$$\boldsymbol{\theta}_{\ell,t+1} \quad = \quad \boldsymbol{\theta}_{\ell,t} - \rho \frac{1}{N} \sum_{n=1}^{N} \frac{\partial \mathcal{L}_n(\boldsymbol{\theta}_{\ell,t})}{\partial \boldsymbol{\theta}_{\ell,t}} \tag{1}$$

Now, imagine you are training an MLP with $N = 4000$ observations.

(a) If you use a batch learning approach (i.e. all observations are used to compute a single weight update), how many iterations/weight updates are obtained in one training epoch? *[1]*

(b) If you use a stochastic minibatch approach with a batch size $B = 32$, how many training iterations are obtained in one epoch? (Show your calculation.) *[2]*

(c) What is the maximum number of iterations possible in 50 epochs if your batch size must be no less than 10 samples (note that sampling for SGD approaches is typically done without replacement)? *[2]*

Problem 2 *Subdifferential of ReLU (7 pts)*

The **subdifferential** $\partial f(\boldsymbol{x})$ of a function f at a point $\boldsymbol{x}$ is the *set* of all the **subgradients** $\boldsymbol{g}$ at that point. Thus, for instance, if $f(x) = |x|$, then the subdifferential at $x = 0$ is given by

$$\partial f(0) = [-1, 1] \tag{2}$$

At $x = 1$, however, there is only one unique subgradient: $\partial f(1) = \{1\}$, which is the subdifferential (set of 1 element). Note that $f(x)$ is not differentiable at $x = 0$, *but* it is subdifferentiable.

(a) The rectified linear unit activation function (ReLU) is defined as: *[2]*

$$\text{ReLU}(a) = \max(a, 0) = a\mathbb{I}(a > 0) \tag{3}$$

Sketch/plot $\text{ReLU}(a)$ for $a \in [-5, 5]$.

(b) *Write* the piecewise subdifferential function $\partial\varphi(a)$, where $\varphi \equiv \text{ReLU}$. (Refer to **PMLI** 8.1.4.1 for an example of how to do this). *[3]*

(c) Sketch/plot the subdifferential function $\partial\varphi(a)$, where $\varphi \equiv \text{ReLU}$ (label both axes). *[2]*

Problem 3 *Optimizers (6 pts)*

PMLI Section 8.4.5 describes approaches to reduce the variance in SGD.

(a) Read the section

(b) List three of the SGD variants described in the section and write their weight update equations. *[6]*

Problem 4 *Feed forward neural network (15 pts)*

The neural network shown in Figure 1 is to be trained for a binary classification problem, $y_n \in \{0, 1\}$. Logistic sigmoid activations are used in the hidden layer neurons.

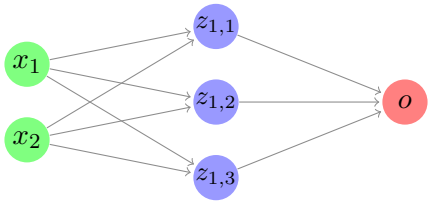

Figure 1

[1] **(a)** What is the dimensionality of the input vector $\boldsymbol{x}$ (i.e. what is D)?

[2] **(b)** How many parameters (weights/biases) are required to train this network? (Show your calculations explicitly.)

(c) We can write the hidden units (activations) at layer 1 as:

$$\boldsymbol{z}_1 = \begin{bmatrix} z_{1,1} \\ z_{1,2} \\ z_{1,3} \end{bmatrix} \tag{4}$$

Thus,

$$\boldsymbol{z}_1 = \boldsymbol{\sigma}(\boldsymbol{b}_1 + \boldsymbol{W}_1 \boldsymbol{x}) \tag{5}$$

where $\boldsymbol{x} = [x_1, x_2]^\top$.

[1] **i.** What is the length of b_1?

[1] **ii.** What are the dimensions of $\boldsymbol{W}_1$?

[2] **iii.** Write an equation for $z_{1,3}$ in terms of x_1, x_2, the relevant elements of $\boldsymbol{W}_1$ (e.g. $w_{3,1}$, etc.) and the relevant element(s) of $\boldsymbol{b}_1$. The equation should be written in scalar form and should include the exponential function explicitly.

(d) The weights in the final (output) layer are given by $\boldsymbol{\theta}_2 = (\boldsymbol{w}_2, b_2)$.

[1] **i.** What is the length of $\boldsymbol{w}_2$?

[1] **ii.** Given that this is a binary classification problem and only a single neuron is specified in the final (output) layer, what activation function should be used in the output layer?

[2] **iii.** Write an explicit equation for o in terms of $\boldsymbol{w}_2$, $\boldsymbol{z}_1$ and b_2.

[1] **iv.** What does o represent?

[3] **(e)** Write/derive the binary cross-entropy loss function $\mathcal{L}_n$ for an observation n in this problem in terms of o_n and y_n.

Problem 5 *Backpropagation for MLP (20 pts)*

(From PMLI Exercise 13.1) Consider the following classification MLP with one hidden layer:

$$\begin{aligned} \boldsymbol{x} &= \text{input vector} \in \mathbb{R}^D && (6)\\ \boldsymbol{z} &= \sigma(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}_1) \in \mathbb{R}^K && (7)\\ \boldsymbol{h} &= \text{ReLU}(\boldsymbol{z}) \in \mathbb{R}^K && (8)\\ \boldsymbol{a} &= \boldsymbol{V}\boldsymbol{h} + \boldsymbol{b}_2 \in \mathbb{R}^C && (9)\\ \mathcal{L} &= \text{CrossEntropy}(\boldsymbol{y}, \text{softmax}(\boldsymbol{a})) \in \mathbb{R} && (10) \end{aligned}$$

where $\boldsymbol{x} \in \mathbb{R}^D$, $\boldsymbol{b}_1 \in \mathbb{R}^K$, $\boldsymbol{W}_1 \in \mathbb{R}^{K\times D}$, $\boldsymbol{V} \in \mathbb{R}^{C\times K}$, where D is the input dimension, K is the number of hidden units, and C is the number of classes. Show that the gradients for the parameters and input are given by:

[4]

(a)

$$\nabla_{\boldsymbol{V}}\mathcal{L} = \left[\frac{\partial \mathcal{L}}{\partial \boldsymbol{V}}\right]_{1,:} = \boldsymbol{u}_2\boldsymbol{h}^\top \in \mathbb{R}^{C\times K} \tag{11}$$

[4]

(b)

$$\nabla_{\boldsymbol{b}_2}\mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}_2}\right)^\top = \boldsymbol{u}_2 \in \mathbb{R}^C \tag{12}$$

[4]

(c)

$$\nabla_{\boldsymbol{W}}\mathcal{L} = \left[\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}}\right]_{1,:} = \boldsymbol{u}_1\boldsymbol{x}^\top \in \mathbb{R}^{K\times D} \tag{13}$$

[4]

(d)

$$\nabla_{\boldsymbol{b}_1}\mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}_1}\right)^\top = \boldsymbol{u}_1 \in \mathbb{R}^K \tag{14}$$

[4]

(e)

$$\nabla_{\boldsymbol{x}}\mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}}\right)^\top = \boldsymbol{W}^\top\boldsymbol{u}_1 \in \mathbb{R}^D \tag{15}$$

where the gradients of the loss with respect to the logit layer and hidden layer are given by:

$$\begin{aligned} \boldsymbol{u}_2 &= \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}}\right)^\top = \text{softmax}(\boldsymbol{a}) - \boldsymbol{y} \in \mathbb{R}^C && (16)\\ \boldsymbol{u}_1 &= \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}}\right)^\top = \boldsymbol{V}^\top\boldsymbol{u}_2 \odot H(\boldsymbol{z}) \in \mathbb{R}^K && (17) \end{aligned}$$

and where $\odot$ denotes the element-wise (Hadamard) product and $H(\boldsymbol{z})$ is the Heaviside step function defined as:

$$H(z_i) = \begin{cases} 1 & z_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$