UNIVERSITY OF PENNSYLVANIA

ESE 546: PRINCIPLES OF DEEP LEARNING

FALL 2020

PRACTICE MID-TERM EXAM

DURATION: 100 MINUTES

Read the following instructions carefully before you begin

- This is a closed book exam. You are allowed to use one A4/Letter paper cheat sheet (front and back). You may not use laptops, phones or the Internet during this exam. You may not discuss with your peers.
- The exam is designed to be completed in 75 minutes. You will have 100 minutes from the time you download the PDF of the exam questions on Gradescope to upload your answers. **Late submissions will only receive 50% of the credit**. If you lose access to Gradescope during this window (and only then), you can email your solutions to the instructor via email pratikac@seas.upenn.edu.
- Begin each problem on a fresh page. Solutions that are not correctly annotated on the Gradescope outline will not receive credit.
- Some questions require you to write a short 1-2 sentence answer. DO NOT write verbose answers, we are just looking for the main idea. You can be as brief as you'd like while writing these answers.
- None of the questions require long derivations. If you find yourself going through lots of equations reconsider your approach or consider moving on to the next question.
- If you are stuck, explain your answers and what are trying to do clearly along with derivations. We will give partial credit for good explanations.
- The questions are NOT arranged in order of difficulty, try to attempt every question.

Problem 1. (16 points)

(1) **(2 points)** Suppose you have a 3-dimensional input $x = (x_1, x_2, x_3) = (2, 2, 1)$ and one-dimensional output y_k for different nonlinearities σ_k

$$a = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$
$$y_k = \sigma_k (a)$$

where b is the bias. If $w = (w_1, w_2, w_3) = (0.5, -0.2, 0)$ and $b = 0.1$ and we got $(y_1, y_2, y_3, y_4) = (0.67, 0.7, 1, 0.7)$ which could be the nonlinearities? No need for explanations.

(i) sigmoid, tanh, indicator function, linear;
(ii) linear, indicator function, sigmoid, ReLU;
(iii) sigmoid, linear, indicator function, leaky ReLU; or
(iv) ReLU, linear, indicator function, sigmoid.

> **Answer:** (iii) sigmoid, linear, indicator function, leaky ReLU

(2) **(4 points)** Explain Dropout in 1-2 sentences. Explain in 1-2 sentences how it performs approximate model averaging.

> **Answer:** Dropout works by efficiently creating large ensembles. It works during training by masking the activations with zeros with probability p each time and then scaling the activations by $1 - p$ during inference time. By scaling the weights during the test time, we are effectively averaging outputs of exponentially large number of models explicitly at test time; this argument is accurate for linear classifier.

(3) **(4 points)** Consider five functions x, x^2, x^3, x^4, x^5 where $x \in \mathbb{R}$. Which of these functions are convex on $\mathbb{R}$? Which are strictly convex on $\mathbb{R}$? Which are strongly convex on $\mathbb{R}$? Which are strongly convex on [0.45, 0.5]? (No explanations necessary)

> **Answer:**
> - Convex: x, x^2, x^4
> - Strictly convex: x^2, x^4
> - Strongly convex: x^2
> - Strongly convex on $[0.45, 0.5]$: x^2, x^3, x^4, x^5

(4) **(2 point)** If a convolutional layer has 3 input channels, a 5×5 kernel-size, 10 output channels and no biases what is the number of parameters of this layer?

Answer: Number of parameters = $3 \times 5 \times 5 \times 10 = 750$

(5) **(2 points)** Explain what "model.train()" and "model.eval()" in PyTorch does for the batch-normalization (BN) layer.

Answer: The calls "model.train()" and "model.eval()" affect the Batch-normalization layers as follows. The former sets the network in the training mode, i.e., batch-normalization uses the mean and standard deviation of the current mini-batch to normalize the data. Note that BN also maintains a running mean and running standard-deviation of these batch-wise statistics which are updated during forward propagation. In the evaluation mode, BN uses the stored running mean and standard deviation to normalize the data instead of compute the statistics of the validation mini-batch.

(6) **(2 points)** Define Bayes error of a classifier. How does the Bayes error of the dataset depend on the model class, e.g., the size of our neural network?

Answer: Bayes error is the population risk of the optimal model.

$$\text{Bayes error} = \mathop{\mathbb{E}}_{(x,y)\sim P}\left[(f^*(x) - y)^2\right].$$

Bayes error does not depend on the hypothesis class.

Problem 2. (14 points)

(1) **(2 points)** Say we want to build a classifier that can classify images of different cars on the road. Which of the following data-augmentation techniques should we employ. Mark all the ones you will use and the ones you will not use, give brief explanation.

(i) Flipping images left to right

(ii) Flipping images upside down

Answer: Use (i). Do not use (ii). Using (ii) changes the distribution of images. The test data is unlikely to have images that are flipped upside down and augmenting the training set using (ii) is therefore not beneficial.

(2) **(2 points)** For m-strongly-convex function with L-Lipschitz gradients, what is an upper bound on the eigenvalues of the Hessian? What is a lower bound on the eigenvalues of the Hessian?

Answer:

$$m \leq \lambda_{\text{min}}(\nabla^2 f(x))$$

$$\lambda_{\text{max}}(\nabla^2 f(x)) \leq L.$$

(3) **(4 points)** If $\hat{y} \in \mathbb{R}^m$ denotes the logits of an m-class classifier for an input x and if $y \in \{1, \dots, m\}$ is the true label, write down the cross-entropy loss function. Explain why label smoothing is used in a deep network classifier.

Answer: The cross-entropy loss is

$$-\log\left(z_y\right).$$

where $z \in \mathbb{R}^m$ denotes the softmax operator applied to the logits $\hat{y}$

$$z_k = \frac{e^{\hat{y}_k}}{\sum_{k'=1}^m e^{\hat{y}_{k'}}}.$$

The cross-entropy loss is minimized when the true logit $\hat{y}_y$ goes to infinity. Label smoothing is an operation that changes the one-hot ground truth label to have a small ϵ-probability on all the incorrect classes. This ensures that the value of $\hat{y}_y$ that minimizes the cross-entropy loss is a finite number.

(4) **(2 points)** Suppose you have a dataset with lots of samples, and after training a deep network, both the training and validation error are high, what changes will you make to improve performance?

> **Answer:** Both training and validation error being high indicates that the model being used is too small to fit the data. We should therefore pick a larger model, or reduce the amount of regularization used during training on our existing model. This increases the size of the hypothesis class.

(5) **(4 points)** Explain in 3-4 sentences the difference between the maximum likelihood estimation (MLE) estimate and the maximum a posteriori (MAP) estimate of the model weights. Explain which one you will use if you have a small training dataset?

> **Answer:** Given inputs x^i and targets y^i, MLE estimate computes the best weights that maximize the likelihood
>
> $$w^*_{\text{MLE}} = \operatorname*{argmax}_w \frac{1}{n} \sum_{i=1}^{n} \log p_w(y^i | x^i)$$
>
> where $p_w(y^i|x^i)$ is a parametrized probability distribution, say a Gaussian. MAP estimation uses a prior distribution on the weights $p(w)$ that incorporates our understanding of which weights are likely to work better (without seeing the training dataset)
>
> $$w^*_{\text{MAP}} = \operatorname*{argmax}_w \frac{1}{n} \sum_{i=1}^{n} \log p_w(y^i | x^i) + \log p(w)$$
>
> A small dataset will have a high variance in the bias-variance tradeoff when we fit the MLE. The MAP estimate will generalize better because the prior effectively reduces the set of hypotheses that are allowed in the above maximization.

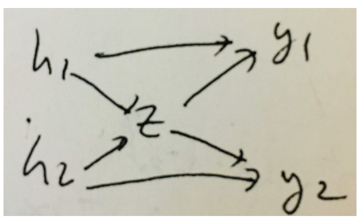
Problem 3. (10 points) The softmax function takes in a vector $(h_1, \ldots, h_d)$ and outputs a vector $y = \text{softmax}(h)$ whose i^{th} entry is

$$y_i = \frac{e^{h_i}}{\sum_{j=1}^d e^{h_j}} = \frac{e^{h_i}}{z}.$$

where $z = \sum_{j=1}^d e^{h_j}$ is the normalization constant. We would like to compute the back-propagation equations for the computation graph of softmax for $d = 2$. The input variables of this graph are h_1, h_2, output variables are y_1, y_2 and z is a temporary variable.

(1) **(2 points)** Draw the computation graph for the 5 variables, i.e., for each variable, say y_2 you should draw a directed edge from each variable that is necessary to compute the value of y_2.

Answer:



(2) **(8 points)** Given the values of $\overline{y_i}$ for $i = 1, 2$, write down the expression for $\overline{h_i}$ and $\overline{z}$.

Answer:

$$\overline{z} = -\overline{y}_1 \frac{e^{h_1}}{z^2} - \overline{y}_2 \frac{e^{h_2}}{z^2}$$

$$\overline{h}_i = \overline{y_i} \frac{e^{h_i}}{z} + \overline{z}\, e^{h_i}.$$

Partial credit will be given for such problems if you show your computations clearly.

Problem 4. (8 points) Consider a dataset $\{(x_i, y_i)\}_{i=1,\dots,n}$ where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ with $d \leq n$. If we arrange all the data in a matrix $X \in \mathbb{R}^{n \times d}$ and outputs as a vector $Y \in \mathbb{R}^n$ we can perform linear regression by solving

$$\min_{w \in \mathbb{R}^d} \ \ell(w)$$

where $\ell(w) = \|Y - Xw\|_2^2$.

(1) **(2 points)** Why is ℓ_2 regularization, i.e., using the loss function

$$\ell(w) + \frac{\lambda}{2}\|w\|_2^2$$

instead of the original loss function $f(x)$, called weight decay?

Answer: If we use ℓ_2 regularization, instead of the original updates

$$w^{(t+1)} = w^{(t)} - \eta \ \nabla \ell(w^{(t)}; x, y)$$

our gradient updates become

$$w^{(t+1)} = (1 - \eta\lambda) w^{(t)} - \eta \ \nabla \ell(w^{(t)}; x, y).$$

In which we can observe that with each updates, we are shrinking, i.e., decaying, our weights by a factor of $(1 - \eta\lambda)$ before subtracting the gradient.

(2) **(6 points)** Derive the solution of the regularized least squares problem

$$\min_{w \in \mathbb{R}^d} \left\{ \ell(w) + \frac{\lambda}{2}\|w\|_2^2 \right\}.$$

Answer: We are solving for w^*. Set the derivative to zero to get

$$\begin{aligned} -2X^\top(Y - Xw) + \lambda w = 0 \\ \Rightarrow X^\top Y = \left(X^\top X + \frac{\lambda}{2} I \right) w \\ \Rightarrow w^* = \left(X^\top X + \frac{\lambda}{2} I \right)^{-1} X^\top Y. \end{aligned}$$

END OF EXAM