UNIVERSITY OF PENNSYLVANIA

ESE 546: PRINCIPLES OF DEEP LEARNING

FALL 2020

FINAL EXAM

12/15 9AM ET - 12/18 8.59AM ET

DURATION: 180 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.
- You will have 180 minutes from the time you start the exam on Gradescope to upload your solutions. We will be very strict as regards to late submissions. They will only receive 50% of the credit, so make sure you keep enough buffer to upload solutions into Gradescope.
- You should use pen and paper (do not use a pencil because it might not be readable when scanned) to write your solutions and click pictures and upload your solutions as a PDF. Please make sure to write in legible handwriting. You can also use a tablet to write your solutions. Please make sure your solutions are legible.
- Begin each problem on a fresh page. Solutions that are not correctly annotated on the Gradescope outline will not receive credit.
- Each question mentions how short/long we would like your responses to be. DO NOT write excessively verbose 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 you are trying to do clearly. We will give partial credit if you are on the right track.
- The questions are NOT arranged in order of difficulty, try to attempt every question.
- **This exam has 11 problems**.
- In case of emergency, send the Instructor an email at pratikac@seas.upenn.edu.

Problem 1 (2 points). What letter grade would you give yourself for this course? Give a one sentence justification.

Problem 2 (10 points). For $x, \mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}_+$, evaluate

$$\nabla_\mu \underset{x \sim N(\mu,\, \sigma^2)}{\mathbb{E}} \left[x^2\right],$$
$$\nabla_\sigma \underset{x \sim N(\mu,\, \sigma^2)}{\mathbb{E}} \left[x^2\right] \text{ and}$$
$$\nabla_\mu \underset{x \sim N(\mu,\, \sigma^2)}{\mathbb{E}} \left[x^3\right],$$

where $N(\mu, \sigma^2)$ denotes the Normal distribution with mean μ and variance σ^2.

Problem 3. (12 points)

1. **(3 points)** For $x \in \mathbb{R}$, consider a probability distribution $p(x)$ which is a mixture of two Gaussians
$$p(x) = \alpha N(x;\ \mu_1, \sigma_1^2) + (1 - \alpha) N(x;\ \mu_2, \sigma_2^2);$$
here $\alpha = 0.75$ is the mixture probability and μ_1, σ_1^2 and μ_2, σ_2^2 are the means and variances of the two Gaussians. If we solve a variational optimization problem
$$q^* = \underset{q \in \mathcal{Q}}{\text{argmin}}\ \text{KL}(q \mid\mid p)$$
where $\mathcal{Q} = \mathcal{P}(\mathbb{R})$ is the set of all probability distributions with support on the entire real line, then what is q^*? Explain your answer. Hint: This question does not require any calculation. Think logically.
2. **(3 points)** Answer True or False with a 1-2 sentence explanation. There exist first-order non-stochastic optimization algorithms that converge faster than Nesterov's accelerated gradient descent for all convex functions $\ell(w)$.
3. **(3 points)** Answer True or False with a 1-2 sentence explanation. After a Generative Adversarial Network (GAN) has converged and is generating images that look like Nature's images, the discriminator $d_u(g_v(z))$ has an accuracy close to 100% for all noise vectors $z \sim N(0, I)$.
4. **(3 points)** Answer True or False with a 1-2 sentence explanation. The kernel trick can be used to train a perceptron/SVM using SGD. It pre-computes summations like $\sum_{ij} \left\langle x^i, x^j \right\rangle$ in the update equation using a kernel matrix $K_{ij} := k(x^i, x^j)$. This kernel trick can also be used to train a deep neural network using SGD.

Problem 4. (10 points)

1. **(3 points)** Explain how inverted Dropout works in 1-2 sentences.

2. **(3 points)** We were training a neural network on the CIFAR-10 dataset with 10 output classes using the cross-entropy loss on the softmax predictions. We saw that the training loss after randomly initializing the weights of the network was 2.3. Can you explain why?
3. **(4 points)** Why is the convolutional operator used as a building block in CNNs? Why do typical CNNs have Average/Max-Pooling operators?

Problem 5 (10 points). Let $\mathcal{F}$ be a finite hypothesis class. We saw in PAC-learning that for any $\epsilon, \delta > 0$, if we draw a dataset D with n samples and if

$$n \geq \frac{1}{2\epsilon^2}\left(\log|\mathcal{F}| + \log\frac{2}{\delta}\right)$$

then with probability at least $1 - \delta$ for all hypotheses $f \in \mathcal{F}$, we have

$$|\hat{R}(f) - R(f)| \leq \epsilon.$$

The quantities $\hat{R}(f)$ and $R(f)$ are the empirical risk and the population risk of a hypothesis f respectively. In particular, we have

$$|R(f_{\text{ERM}}) - R(f^*)| \leq 2\epsilon$$

with probability at least $1 - 2\delta$ if uniform convergence holds. This says that the population risk of the hypothesis f_{ERM} obtained using the training set is close to the population risk of f^* which is the hypothesis that achieves the smallest population risk in $\mathcal{F}$.

1. **(2 points)** John was trying to fit a model after learning this result and found some gap in the training and validation error with $n = 100$ samples. If he wants to reduce the gap by half, how many samples should he use?
2. **(4 points)** John collected the extra samples and observed that the gap between training and validation error did not halve. Give **two reasons** as to why this might be the case. Assume that John was diligent in collecting the new data and made sure that there is no mismatch in the distributions of the training and test data.
3. **(4 points)** Unlike John, you feel that it is cumbersome to gather more data and would instead like to use your knowledge of machine learning to create a good model. One strategy to use is as follows: you start fitting simple models on the dataset and sequentially increase the model complexity. In the first iteration, you use a hypothesis class $\mathcal{F}^1$ to get a model f^1_{ERM}, in the second iteration, you use a larger hypothesis class $\mathcal{F}^2$ to get a model f^2_{ERM}, and so on always ensuring that

$$\mathcal{F}^1 \subset \mathcal{F}^2 \subset \cdots$$

Suppose that all these hypothesis classes are finite and you can correctly compute $\log|\mathcal{F}^k|$ at each step. Using your knowledge of the the generalization bound given above, **without using a validation set or cross-validation** (say, the training dataset is too small to even do cross-validation) answer which model out of

$$\{f^1_{\text{ERM}}, f^2_{\text{ERM}}, \dots, \}.$$

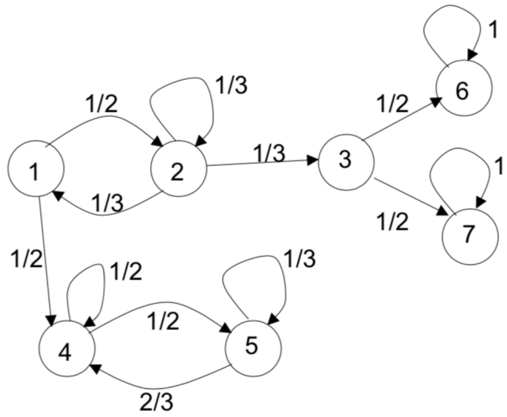
will you choose? Provide a 1-2 sentence explanation for your choice.

Problem 6. (10 points)

1. **(3 points)** For stochastic gradient descent with constant step-size η for an m-strongly convex and L-smooth function with global optimum w^*, the sub-optimality of SGD is

$$\mathop{\mathbb{E}}_{\omega_0,\dots,\omega_t} \left[\ell(w^{t+1}) - \ell(w^*)\right] \leq \frac{\eta L \sigma_0}{2m} + (1-\eta m)^t \left(\ell(w^0) - \ell(w^*) - \frac{\eta L \sigma_0}{2m}\right).$$

   The second term here converges very quickly with time while the first term is a constant with respect to time t. Does SGD converge with a constant step-size? How do we change the step-size so that SGD converges to the global minimum as $t \to \infty$ from any initial condition?
2. **(3 points)** We know that mini-batch SGD is not any cheaper computationally than SGD in theory; the reduction in the number of weight updates required to reach an ϵ-neighborhood of the global optimum (for convex objectives) is balanced out by the extra amount of computational work done during each weight update if the batch-size is larger than one. Why do we then use mini-batch size larger than 1 to train neural networks in practice?
3. **(4 points)** Consider the Markov chain with transition probabilities shown below.



How many steady-state distributions are there? Argue your answer in 1-2 sentences. You do not need any computations to solve this problem, think logically.

Problem 7 (8 points). Estimate the integral

$$\int_{x \in \mathbb{R}} e^{-N(2x^2+x^4)} \,\mathrm{d}x$$

using the Laplace approximation if N is large. You will find it useful to know that

$$\int_{x \in \mathbb{R}} e^{-\frac{x^2}{2\sigma^2}} \,\mathrm{d}x = \sqrt{2\pi}\,\sigma.$$

Problem 8 (6 points). We assumed while studying SGD for machine learning problems that there exist constants $\sigma_0 \geq 0$ and $\sigma \geq 0$ that give an upper bound on the norm of the stochastic gradients

$$\mathop{\mathbb{E}}_{\omega}\left[\|\nabla \ell_\omega(w)\|^2\right] \leq \sigma_0 + \sigma \|\nabla \ell(w)\|^2$$

for any w in the domain.

1. **(4 points)** Show that $\sigma \geq 1$.
2. **(2 points)** Can σ_0 be zero? Explain your answer.

Problem 9 (14 points). Consider optimizing a one-dimensional loss function

$$\ell(w) = \frac{1}{2n} \sum_{i=1}^{n} (a^i w - b^i)^2$$

where $w, a^i, b^i \in \mathbb{R}$ and $\left\{(a^i, b^i)\right\}_{i=1,\dots,n}$ is the dataset. We will use stochastic gradient descent (SGD) with a constant step-size $\eta > 0$ and constant batch-size of 1 in this problem. You can use the Markov chain model of SGD to answer the following questions; in this case, since the batch-size is 1, the updates of SGD are modeled as

$$W^{t+1} = W^t - \eta \nabla \ell(W^t) + \eta\, \xi^t$$

where $\xi^t \sim N(0, I)$ is standard Gaussian noise.

1. **(4 points)** Write down all critical points of $\ell(w)$, i.e., all locations where the gradient $\nabla \ell(w) = 0$.
2. **(5 points)** What is the asymptotic expected value

$$\lim_{t \to \infty} \mathop{\mathbb{E}}_{\omega_1, \dots, \omega_t} \left[w^{t+1}\right]$$

   of the iterates of SGD? Here ω_t denotes the sample of the datum $(a^{\omega_t}, b^{\omega_t})$ used to update the weights w^t at the tth iteration.
3. **(5 points)** If we have $a^1 = a^2 = \ldots = a^n = 1$, what is the asymptotic variance of the iterates of SGD?

$$\lim_{t \to \infty} \mathop{\mathrm{Var}}_{\omega_0, \dots, \omega_t} \left[w^{t+1}\right]$$

Problem 10 (10 points). In your homework problem, you developed a variational generative model to synthesize MNIST digits. At test time, you sampled latent factors $z \sim N(0, I)$ and ran the decoder $p_v(x|z)$ to get new images different from the ones in the training set. However, you did not get any control as to which digit was generated by the decoder.

1. Explain in 4-5 sentences how you will modify the variational auto-encoder (VAE) to synthesize images of a particular digit. At test time, your model should take in as input a one-hot encoding of the digit and synthesize images only of that particular digit; it should be able to accept any of the 10 digits as input.
2. Clearly explain the inputs, outputs of all parts of your model, the loss function you will use to train the model and the trainable parameters.
3. Explain what the synthesized image will look like if instead of providing a one-hot vector as input, you provide a two-hot vector as input at test time, e.g.,

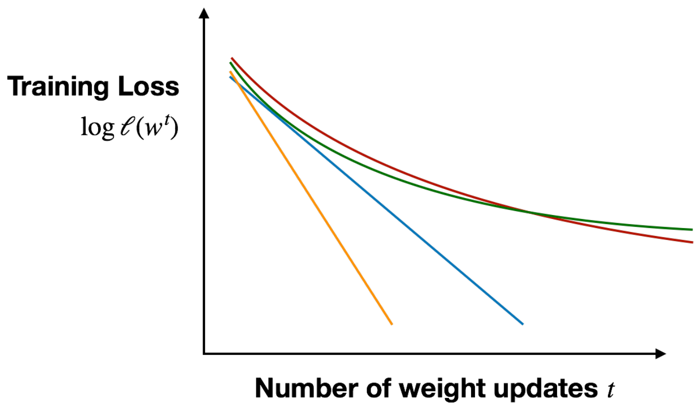
   $$[0, 0, 1, 0, 1, 0, \ldots] = \text{one-hot}(2) + \text{one-hot}(4).$$

   Hint: To answer this part, think of how the latent space $z \in \mathbb{R}^m$ is structured for a standard VAE.

Problem 11 (8 points). We trained a machine learning model which had a strongly convex objective $\ell(w)$ using four algorithms

(1) Gradient descent,
(2) Gradient descent with Nesterov's acceleration,
(3) Stochastic gradient descent with a small batch-size, and
(4) SGD with Nesterov's acceleration and a small batch-size.

We recorded the logarithm of the loss $\log \ell(w^t)$ where $\ell(w^t)$ refers to the loss on the full training data at iteration t. But we made a mistake in logging the data and cannot decide which plot belongs to which algorithm. Clearly state which color corresponds to which algorithm, explain your reasoning in 3-4 sentences.



END OF EXAM