Problem Set 4
ECE 685D Fall 2020

Instructor: Vahid Tarokh
ECE Department, Duke University

Due: 8:59:59 p.m. EST on Oct. 23, 2020

Problem 1: Classification with Convolutional Neural Networks

In this problem, we will train a convolutional neural network (CNN) to classify 10 objects from "The Quick, Draw!" Data set. The Quick Draw Dataset is a collection of 50 million drawings across 345 categories, contributed by players of the game Quick, Draw! (`https://quickdraw.withgoogle.com/data`). Here, we only focus on 10 objects out of 345 categories as given in the following list:

Labels={airplane, basket, basketball, bed, bus, calculator, cookie, eyeglasses, mushroom, pizza}

You need to download the data set, and make it ready for using in PyTorch. Since, the full data set is very large, you may just download the data from the above classes. To make the pre-processing steps easy, you can only download the simplified and numpy version of the data set. To do this, please go to the `https://github.com/googlecreativelab/quickdraw-dataset#the-raw-moderated-dataset`, and select Numpy bitmap files format. Then in the opened console use the name of the above objects to filter the other classes. Next download all the 10 classes data. After downloading, please use 6000 random samples for each class, and another random 1000 samples for the test data set. Please remember that your test set should not include any samples from the training set. To do splitting between the train and the test data sets, you may either write you own data loader and batch sampling function, or use a customized data loader in PyTorch. Once you prepared the data set, it is the time to design a model for the classification of the above objects.

In particular, you need to design a convolutioanl neural network (CNN) which can classify the above 10 objects. You can use the fallowing operations: Convolution, Non-linear functions (Relu, leaky Relu, sigmoid), (Max, Average) pooling, Batch-normalization, Dropout, and Fully connected layers. You can use any optimization scheme you have learned in the course. Please report the train and the test accuracy in each epoch by plotting these quantities versus epoch number.

Problem 2: Feature Extraction Using Autoencoder

In this problem you are asked to do a classification task based on the features extracted using various approaches such as autoencoder (AE), contractive autoencoder (CAE) and PCA (i.e., linear autoencoder). The data set is MNIST which you can download using PyTorch.

AE-based feature extraction

Use the following architecture for the encoder:

- Four fully connected layers with 128, 64, 12 and 3 output neurons in each layer;
- The first three layers use ReLU activation functions while the last layer uses linear activation function (i.e., no activation)

The last layer of the decoder uses $\tanh(\cdot)$ activation function. The loss function for DAE is the MSE (see the lecture notes). You should implement forward pass by yourself. For backward pass you can use autograd.

CAE-based feature extraction. Use the same architecture as the AE with contractive loss function (see lecture notes).

PCA-based feature extraction. Use three components. You can use the built-in function in sklearn.

Classification method. Use multiclass logistic regression with default parameters in sklearn. You can use the built-in function.

Report the train and the test classification accuracies for all three feature extraction methods.

Problem 3: Bayesian Networks

Student X has registered for a deep learning course and wants to know his/her odds of obtaining a letter of recommendation from the instructor. In order to do this, the student can use the Bayesian network shown in Fig. 1.

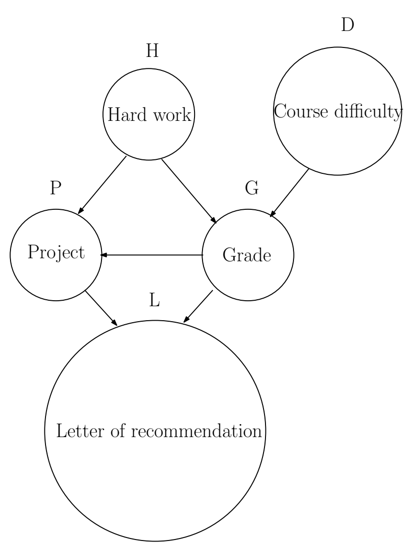

Figure 1: Bayesian network.

Indicate whether the following statements are true or false and explain why (H: Hard work, D: Course difficulty, P: Projects, G: Grade, L: Letter of recommendation).

- $(H \perp\!\!\!\perp L)|P$
- $(H \perp\!\!\!\perp L)|P, G$
- $(P \perp\!\!\!\perp D)|G$
- $(H \perp\!\!\!\perp D)|L$

Write down the factorized form of the joint distribution $\mathbb{P}(H, D, P, G, L)$.

Write down the expression for the probability of obtaining the letter of recommendation $\mathbb{P}(L = 1)$ using $G \in \{A, B\}$ and $P \in \{A, B\}$. Compute $\mathbb{P}(L = 1)$ using the following probability tables.

Table 1: Joint probability distribution of P and G

$\mathbb{P}(P = A, G = A) = 0.4$
$\mathbb{P}(P = A, G = B) = 0.3$
$\mathbb{P}(P = B, G = A) = 0.1$
$\mathbb{P}(P = B, G = B) = 0.2$

Table 2: Conditional probability distribution of L given P and G

$\mathbb{P}(L = 1 \vert P = A, G = A) = 0.8$
$\mathbb{P}(L = 1 \vert P = A, G = B) = 0.6$
$\mathbb{P}(L = 1 \vert P = B, G = A) = 0.3$
$\mathbb{P}(L = 1 \vert P = B, G = B) = 0.1$

Problem 4: Restricted Boltzmann Machines

Recall from the lecture notes in RBMs, the energy of the joint random variables $(\mathbf{x}, \mathbf{h})$ is given by

$$\begin{aligned} E(\mathbf{x}, \mathbf{h}) &= -\mathbf{h}^T \mathbf{W} \mathbf{x} - \mathbf{c}^T \mathbf{x} - \mathbf{b}^T \mathbf{h} \\ &= -\sum_j \sum_k W_{j,k} h_j x_k - \sum_k c_k x_k - \sum_j b_j h_j, \end{aligned}$$

where $\mathbf{x} \in \{0, 1\}^D$ denote the visible binary random variables and $\mathbf{h} \in \{0, 1\}^F$ represent the hidden random variables. Then, the joint probability distribution is given by

$$p(\mathbf{x}, \mathbf{h}) = \exp\left[-E(\mathbf{x}, \mathbf{h})\right] / Z,$$

where $Z = \sum_{\mathbf{x},\mathbf{h}} \exp\left[-E(\mathbf{x}, \mathbf{h})\right]$ is the partition function.
Show that the conditional distributions $p(x_k = 1|\mathbf{h})$ and $p(h_j = 1|\mathbf{x})$ are respectively given by

$$\begin{aligned} p(x_k = 1|\mathbf{h}) &= \sigma\left(c_k + \mathbf{h}^T \mathbf{W}_{.k}\right) \\ p(h_j = 1|\mathbf{x}) &= \sigma\left(b_j + \mathbf{W}_{j.} \mathbf{x}\right), \end{aligned}$$

where $\sigma(.)$ is the sigmoid function, $\mathbf{W}_{.k}$ and $\mathbf{W}_{j.}$ denote the kth column and the jth row of $\mathbf{W}$, respectively. Next, for a given observation $\mathbf{x}$, show that the derivative of the negative log-likelihood (NLL) is given by the following formula:

$$-\frac{\partial \log p(\mathbf{x})}{\partial \theta} = \mathbb{E}_{\mathbf{h}}\left[\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial \theta} | \mathbf{x}\right] - \mathbb{E}_{\mathbf{x},\mathbf{h}}\left[\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial \theta}\right].$$