HW2. Back Propagation

Question 2.

b) $E = -\sum_i t_i \log(t_i)$; $X_i = \frac{e^{S_i}}{\sum_{c=1}^{m} e^{S_c}}$

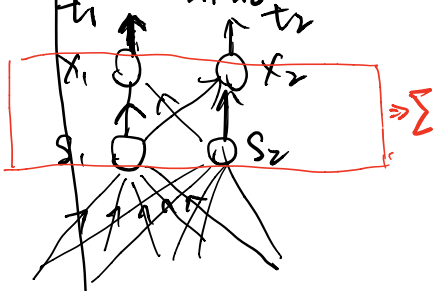
$X_1 = \frac{e^{S_1}}{e^{S_1}+e^{S_2}+e^{S_3}+e^{S_4}+e^{S_5}}$

$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial X_i} \cdot \frac{\partial X_i}{\partial S_i} \cdot \frac{\partial S_i}{\partial w_{ji}}$

$\frac{\partial E}{\partial X_L} = -\left(\frac{t_i}{X_c}\right)$

$\frac{\partial X_c}{\partial S_i} = \begin{cases} \frac{e^{S_c}\sum e^{S_c} - e^{S_i}\cdot e^{S_c}}{\left(\sum_{c=1}^{m} e^{S_c}\right)^2} = X_i(1-X_c) & : \text{if } i=c \\ \frac{-e^{S_i}\cdot e^{S_c}}{\left(\sum_{c=1}^{m} e^{S_c}\right)^2} & \text{if } i \neq c \end{cases}$

$= X_i \cdot X_c$



$\frac{\partial E}{\partial X_c} \cdot \frac{\partial X_c}{\partial S_i} \cdot \frac{\partial S_i}{\partial w_{ji}}$

① $\frac{\partial E}{\partial X_c} = -\frac{t_i}{X_c}$

② $\frac{\partial X_c}{\partial S_i} = \begin{cases} \text{if } c=i: & \frac{e^{S_c}\cdot\sum e^{S_d} - e^{S_c}\cdot e^{S_c}}{\left(\sum_d e^d\right)^2} = X_c(1-X_c) \\ \text{if } c\neq i: & \frac{-e^{S_c}\cdot e^{S_i}}{\left(\sum_d e^d\right)^2} = -X_i X_c \end{cases}$

$\frac{e^{S_c}\cdot\left(\sum e^{S_d} - e^{S_c}\right)}{\left(\sum_d e^d\right)^2}$

③ $\frac{\partial S_i}{\partial w_{ij}} = y_j$

$\Rightarrow \frac{\partial E}{\partial w_{ij}} = -\frac{t_i}{X_c}\left[X_c(1-X_c) + \sum_{i\neq c}(-X_i X_c)\right]\cdot y_j$

$= \left[-t_i(1-X_c) + \sum_{i\neq c} t_i X_c\right]\cdot y_j$

$= \left[-t_i + t_i X_c + \sum_{i\neq c} t_i X_c\right]\cdot y_j$

$= \left[\sum_i t_i X_c - t_i\right]\cdot y_j$

② class 7:

Logistic Regression

$$P(y_i|x_i) = f(x_i)^{y_i} \cdot (1-f(x_i))^{1-y_i}$$

$$\prod_{i=1}^{N} P(y_i|x_i) = \prod_{i=1}^{N} f(x_i)^{y_i} \cdot (1-f(x_i))^{1-y_i} \Rightarrow \text{LIKELIHOOD}$$

$$\log \prod_{i=1}^{N} P(y_i|x_i) = \sum_{i=1}^{N} \log \left[f(x_i)^{y_i} \cdot (1-f(x_i))^{(1-y_i)} \right]$$

$$L(\theta) = \sum_{i=1}^{N} y_i \log f(x_i) + (1-y_i) \cdot \log (1-f(x_i))$$

Maximum Likelihood: $f(x_i) = \text{sigmoid}(z_i) = \frac{1}{1+e^{-z_i}} = \frac{1}{1+e^{-\theta^T x}}$; $z = +\theta^T \cdot x$

Hence: $L(\theta) = \sum_{i=1}^{N} y_i \log \frac{1}{1+e^{-z_i}} + (1-y_i) \cdot \log \left(\frac{e^{-z_i}}{1+e^{-z_i}} \right)$

$$\Rightarrow f(x_i) = \frac{1}{1+e^{-x_i}}$$

$$\frac{f(x_i)}{\partial \theta} = \frac{+e^{-z_i}}{(1+e^{-z_i})^2} \cdot \frac{\partial z}{\partial x} = f(x_i) \cdot (1-f(x_i)) \cdot (+x_i)$$

$$\Rightarrow \frac{\partial L(\theta)}{\partial \theta} = \sum_{i=1}^{N} y_i \frac{f(x_i) \cdot (1-f(x_i)) \cdot (+x_i)}{f(x_i)} + (1-y_i) \cdot \frac{-1}{1-f(x_i)} \cdot f(x_i) \cdot (1-f(x_i)) \cdot (x_i)$$

MAX-LIKIHOOD $= \sum_{i=1}^{N} y_i \cdot (1-f(x_i)) \cdot (+x_i) + (1-y_i) \cdot (-f(x_i)) \cdot (+x_i)$

$$= \sum_{i=1}^{N} \left[y_i - y_i f(x_i) + y_i f(x_i) - f(x_i) \right] (+x_i)$$

$$= \sum_{i=1}^{N} (y_i - f(x_i)) \cdot (x_i) = 0 \quad \times$$

$$f(x_i) = \frac{1}{1+e^{-\theta^T x}}$$

Gradient Descent

③ posterior $\Rightarrow P(\theta|x) = \frac{P(x|\theta) P(\theta)}{P(x)}$ ← Likelihood

ECE-GY 6143, Spring 2020

Midterm Exam

- Total duration: 90 minutes.
- You **can** use one page as a cheat sheet.
- You **cannot** consult your notes, textbook, your neighbor, or Google.
- Maximum points: 60.

---

1. **(5 points)** Please write down the time at the *start* and *end* of your exam. The difference should not exceed 90 minutes. Please also include your *signature* on the first page; by doing so, you are affirming the NYU Tandon School of Engineering student code of conduct.

2. **(10 points)** This is a slight variant of a homework problem. Let $\{x_1, x_2, \ldots, x_n\}$ be a set of points in d-dimensional space, and let $\{\beta_1, \beta_2, \ldots, \beta_n\}$ be positive numbers that represent weights. Suppose we wish to produce a single point estimate $\mu \in \mathbb{R}^d$ that minimizes the *weighted* squared-error:

$$\beta_1 \|x_1 - \mu\|_2^2 + \beta_2 \|x_2 - \mu\|_2^2 + \ldots + \beta_2 \|x_n - \mu\|_2^2$$

Find a closed form expression for μ and prove that your answer is correct.

2.

$$\beta_1 \|x_1 - u\|_2^2 + \beta_2 \|x_2 - u\|_2^2 + \cdots + \beta_n \|x_n - u\|_2^2$$

$$f(u) = \sum_{i=1}^{n} \beta_n \|x_n - u\|^2 = \sum_{i=1}^{n} \beta_n \cdot \|x_n^T \cdot x_n - 2x_n^T \cdot u + u^2\|$$

$$\frac{\partial f(u)}{\partial u} = \sum_{i=1}^{n} \beta_n \cdot \|-2x_n^T + 2u\| = 0$$

$$\sum_{i=1}^{n} \beta_n (+2x_n^T) = \sum_{i=1}^{n} \beta_n \cdot (2u)$$

$$u = \frac{\sum_{i=1}^{n} \beta_n \cdot x_n^T}{\sum_{i=1}^{n} \beta_n}$$

3. **(10 points)** Assuming a classification application with n data points in d dimensions where $n = 10d = 10 * 10^8$, *rank* the following algorithms in decreasing order of (i) training times; (ii) testing times. Include a few sentences justifying your reasoning.

   a. Perceptron.

   b. Nearest neighbors.

   c. Kernel perceptron with a polynomial kernel of order 4.

   d. Kernel perceptron with gaussian kernel.

   e. Linear support vector machines.

4. **(15 points)** The following represents python code for an algorithm that attempts to perform linear regression. (a) Identify the algorithm. (b) Explain why this algorithm may not converge as implemented below, and identify the line in the algorithm that makes this happen. (c) Suggest a way to fix this algorithm.

```
def optim_alg(init, steps, grad):
  xs = [init]
  for step in steps:
    xs.append(xs[-1] - step * grad(xs[-1]))
  return xs

def linear_reg_grad(X,y,w):
  row_id = numpy.random.randint(X.shape[1])
  x = X[row_id,:]
  return x.T.dot(x.dot(w)-y)

input_to_optim_alg = lambda w: linear_reg_grad(X,y,w)
learning_rates = [0.001]*300
ws = optim_alg(w0,learning_rates,input_to_optim_alg)
```

random choose from 1 to 100

a)

$\frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i))$ $\quad \frac{1}{2}(y_i - f(x_i))^2$ $\quad f(x_i) = \theta_1 x + \theta_0$

$= \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}(y_i - \theta_1 x - \theta_0)^2$

$\frac{\partial f}{\partial \theta_1} = \frac{1}{N}\sum_{i=1}^{N} (y_i - \theta_1 x_i - \theta_0)(-x_i) = 0$

$\frac{1}{N}\sum_{i=1}^{N} -y_i x_i + \theta_1 x_i^2 + \theta_0 x_i = 0$

$\frac{\partial f}{\partial \theta_0} = \frac{1}{N}\sum_{i=1}^{N} (y_i - \theta_1 x - \theta_0)\cdot(-1) = 0$

② $\quad \theta_0 = \frac{\sum_{i=1}^{N} (y_i - \theta_1 x_i)}{N}$ ①

$\sum_{i=1}^{N} \theta_1 x_i^2 = \sum_{i=1}^{N} +y_i x_i - \theta_0 x_i$

$\sum_{i=1}^{N} \theta_1 x_i^2 = \sum_{i=1}^{N} y_i x_i - \sum_{i=1}^{N} x_i \left[\frac{1}{N}\sum_{i=1}^{N} y_i - \frac{1}{N}\theta_1 \sum_{i=1}^{N} x_i\right]$

$\sum \theta_1 x_i^2 = \sum y_i x_i - \sum x_i \cdot \frac{1}{N}\sum y_i + \frac{1}{N}\theta_1 \sum x_i \sum x_i$

$\theta_1 = \frac{\sum y_i x_i - \frac{1}{N}\sum x_i \sum y_i}{\sum x_i^2 - \frac{1}{N}\sum x_i \sum x_i}$

(2- Dimension- case. $y_i = \theta x + \theta_0$

b)

1. Gradient Descent may not converge in 300 steps. It needs more steps to converge.
2. learning rate can be smaller to make sure it converge.

c)

See: tolerance = 0.0001

General case ⇒ Stochastic GD.

$J(\theta) = \frac{1}{2}\sum_{i=1}^{N} (h_\theta(x_i) - y_i)^2$

$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{2} \cdot 2(h_\theta(x_i) - y_i) \cdot \frac{\partial h_\theta(x_i)}{\partial \theta_j}$

$= \frac{1}{2} \cdot 2(h_\theta(x_i) - y_i) \cdot \frac{\partial \theta^T x}{\partial \theta_j}$

$= (h_\theta(x_i) - y_i) \cdot x_j$

5. **(10 points)** To combat the COVID-19 pandemic, an enterprising NYU Tandon graduate student decides to build a logistic regression model to predict the conditional likelihood of a person being one of three states – susceptible, infected, cured – based on forehead temperature measurements over the last 30 days. Fortunately, such measurements for a population of 10,000 persons is available.

   a. Identify the parameters of the problem (number of samples n, data dimension d, number of classes k.)

   b. If X and y denote the arrays that encode the training data points and labels, what are the sizes of X and y?

   c. Starting from the definition of conditional likelihood, derive the loss function used to train the model.

a). b)

k=3

d=30 . n=10000

$X \in R^{n\times 30}$, $Y \in R^{n\times 1}$, $y_i \in \{-1, 0, 1\}$

Bernoulli Case.

$y_i = \theta^T x$

$f(y_i) = \text{sigmoid}(y_i) = \frac{1}{1+e^{-\theta^T x}}$

$L(\theta) = \prod_{i=1}^{N} f(y_i)^{y_i} \cdot f(1-y_i)^{(1-y_i)}$

$\log L(\theta) = \sum_{i=1}^{N} \log f(y_i)^{y_i} f(1-y_i)^{(1-y_i)}$

$= \sum_{i=1}^{N} y_i \log f(y_i) + (1-y_i) \cdot \log f(1-y_i)$

$\frac{\partial f(y_i)}{\partial \theta} = \frac{\partial f(y_i)}{\partial z_i} \cdot \frac{\partial z_i}{\partial x_i} = \frac{\partial \frac{1}{1+e^{-z_i}}}{\partial z_i} \frac{\partial z_i}{\partial \theta}$

$= \frac{+e^{-z}}{(1+e^{-z})^2} \cdot x_i$

$= f(y_i) \cdot (1-f(y_i)) \cdot x_i$

$\frac{\partial \log L(\theta)}{\partial \theta} = \sum_{i=1}^{N} \left[y_i \frac{f(y_i)\cdot(1-f(y_i))}{f(y_i)} - (1-y_i) \cdot \frac{f(y_i)\cdot(1-f(y_i))}{f(1-y_i)} \right] \cdot x_i$

$= \sum_{i=1}^{N} [y_i \cdot (1-f(y_i)) - (1-y_i) \cdot f(y_i)] \cdot x_i$

$= \sum_{i=1}^{N} [y_i - y_i f(y_i) - f(y_i) + y_i f(y_i)] \cdot x_i$

$= \sum_{i=1}^{N} [y_i - f(y_i)] \cdot x_i$

c) Method 1. One-vs-rest

Binary Case. (Bernoulli Case):

Step1: Problem1 $X \in R^{n\times 30}$, $Y \in R^{n\times 1}$, $y_i \in \{-1, 0, 1\}$

Transform label 0 into label 1 Temperary for step1.

$y_i \in \{-1, 1\}$

Step2: Problem2: $X \in R^{(1000)\times 30}$, $Y \in R^{1000\times 1}$, $y_i \in \{-1, 1\}$

Transform label 0 into label -1 Temperary.

Method2.

Multinomial Logistic Regression.

Softmax: $h_\theta(x) = \frac{e^{\theta_j^T x}}{\sum_{j=1}^{3} e^{\theta_j^T x}}$

$L(\theta) = \prod_{i=1}^{n} \left(\frac{e^{\theta_j^T x}}{\sum_{j=1}^{3} e^{\theta_j^T x}} \right)^{1(y^{(i)}=1)} = \prod_{i=1}^{n} \left(\frac{e^{\theta_1^T x}}{e^{\theta_1^T x}+e^{\theta_2^T x}+e^{\theta_3^T x}} \right)^{1(y^{(i)}=1)}$

$\left(\frac{e^{\theta_2^T x}}{e^{\theta_1^T x}+e^{\theta_2^T x}+e^{\theta_3^T x}} \right)^{1(y^{(i)}=2)} \left(\frac{e^{\theta_3^T x}}{e^{\theta_1^T x}+e^{\theta_2^T x}+e^{\theta_3^T x}} \right)^{1(y^{(i)}=3)}$

6. **(10 points)** Recall that the kernel trick is basically used to compute inner products between data points that are implicitly transformed into some higher dimensional space via a mapping ϕ:

$$K(x, z) = \langle \phi(x), \phi(z) \rangle.$$

Let's assume that data points are 2-dimensional. So if $K(x, y)$ = (1 + xˆT z)ˆ2$ is the quadratic kernel, write down the explicit form of the mapping ϕ (i.e., write down what $\phi(x)$ and $\phi(z)$ are for this kernel).

$$K(x,y) = (1 + x^T \cdot z)^2$$

$$= (1 + x^T \cdot z)^T \cdot (1 + x^T \cdot z)$$

$$= (1 + z^T \cdot x) \cdot (1 + x^T \cdot z)$$

$$= \phi(x) \cdot \phi(z)$$

$$\phi(x) = (1 + z^T \cdot x)$$