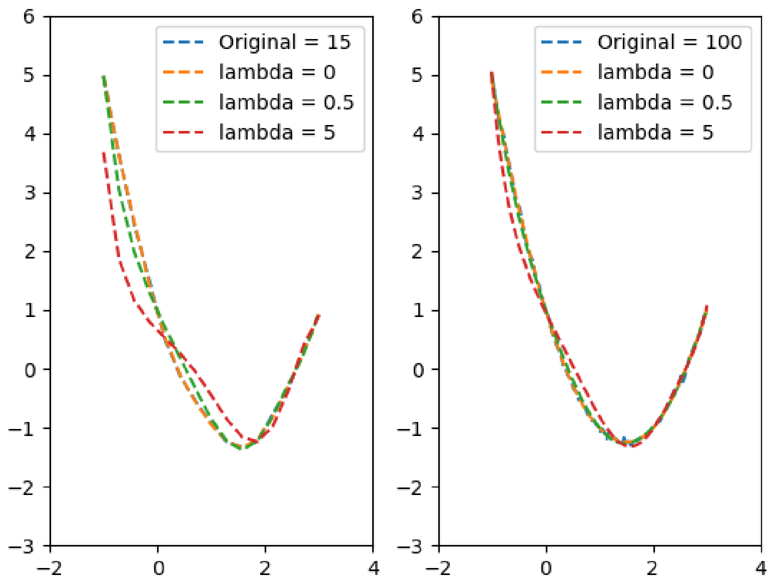
Deep Learning: Homework 2

Qi Luo
A02274095

2/1/2019

Problem 1



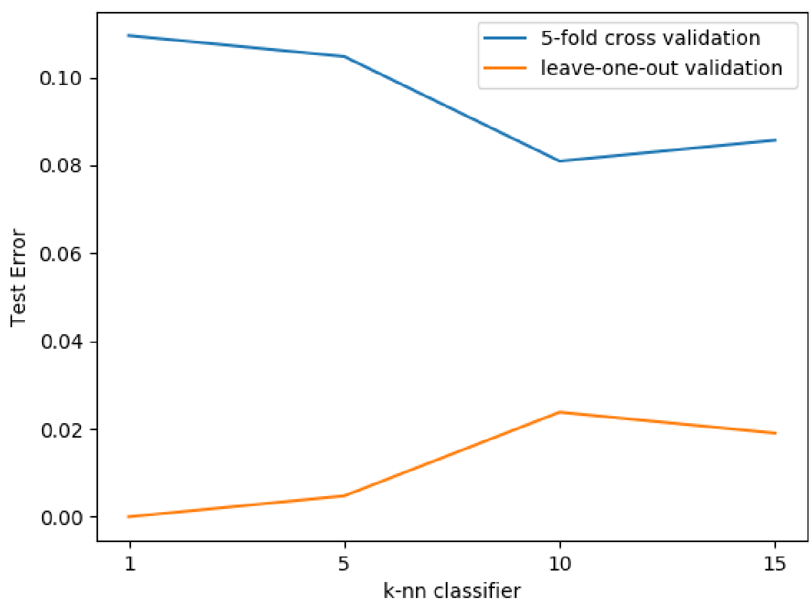

```
N= 15 lamb= 0
mse= 0.01609729498753939
weights: [[ 0.93678933 -3.10972877  1.59477047  0.32636481 -1.20925361  0.22479312
   0.59711141 -0.3975049   0.09398688 -0.0077787 ]]
N= 15 lamb= 0.5
mse= 0.8152249839200794
weights: [[ 9.81934702e-01 -1.98887570e+00  4.58166134e-01 -6.42780498e-01
   3.53482235e-01 -1.47821397e-01  2.45215342e-01 -1.32177503e-01
   2.34680367e-02 -8.74709458e-04]]
N= 15 lamb= 5
mse= 7.834369501941033
weights: [[ 0.65117494 -0.93030207  0.24525379 -0.60182884  0.24761391 -0.32154835
   0.42726454 -0.20831568  0.04492133 -0.00383345]]

N= 100 lamb= 0
mse= 0.17445189056012259
weights: [[ 1.01409962 -3.03955179  0.91056565  0.08847884  0.1601568  -0.12512667
  -0.08988528  0.11242474 -0.03762503  0.00419059]]
N= 100 lamb= 0.5
mse= 0.5668908623695427
weights: [[ 1.02825294e+00 -2.69443856e+00  7.02506290e-01 -3.78192656e-01
   3.93203128e-01  4.29523763e-02 -1.11621567e-01  1.54861398e-02
   7.49359076e-03 -1.62612206e-03]]
N= 100 lamb= 5
mse= 8.287441387252901
weights: [[ 0.96950757 -1.86965361  0.3995376  -0.67760295  0.39187305 -0.17798425
   0.3162178  -0.19626039  0.04546226 -0.00343721]]
```

According to the plots, when N = 15 and $\lambda = [0, 0.5, 5]$ matches to [overfitting, appropriate fit, underfitting]. But for N = 100, it matches to [appropriate fit, appropriate fit, underfitting], because it have enough data to fit in.

Problem 2

1. Import data

2. According to the plots, when k=1 and 5 is overfitting.



3. For logistic regression, I got 9.05% error on traing data and 20% on testing data.
   For SVM, I got 7.38% error on traing data and 17.14% on testing data.
   SVM performed better than logistic regression. Both of them are a little bit overfit the data. Compare to k-nn classifier, when k = 10 has the best performance.

Problem 3

1. $E[\hat{\bar{x}}] - \bar{x} = E[\frac{1}{n}\sum_{i=1}^{n} X_i] - \mu = [\frac{1}{n}\sum_{i=1}^{n} E[X_i]] - \mu = \frac{1}{n} * n\mu - \mu = 0$

2. $Var[\hat{\bar{x}}] = Var[\frac{1}{n}\sum_{i=1}^{n} X_i] = \frac{1}{n^2}\sum_{i=1}^{n} Var[X_i] = \frac{1}{n^2} * n\sigma^2 = \frac{\sigma^2}{n}$

3. Since it can be shown that the MSE of an estimator is equal to the square of its bias plus the variance.$MSE(\hat{\bar{x}}) = Var[\hat{\bar{x}}] + E[\hat{\bar{x}}] - \bar{x}$.
   Therefore, $MSE(\hat{\bar{x}}) = \frac{\sigma^2}{n} + 0 = \frac{\sigma^2}{n}$.

4. $E[\hat{s}^2] - \sigma^2 = E[\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2] - \sigma^2$
   $= E[\frac{1}{n}\sum_{i=1}^{n}((X_i - \mu) - (\bar{X} - \mu))^2] - \sigma^2$
   $= E[\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^2 - 2(\bar{X} - \mu)(X_i - \mu) + (\bar{X} - \mu)^2] - \sigma^2$
   $= E[\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^2 - (\bar{X} - \mu)^2] - \sigma^2$
   $= -E[(\bar{X} - \mu)^2] = -\sigma^2 \neq 0$
   Therefore, this estimator is biased. If $\hat{s}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$ as a new estimator of σ^2, the estimator is unbiased.

Problem 4

1. For a single perceptron network,

$$\begin{cases} 0 & , weights * x + biases \leq 0 \\ 1 & , weights * x + biases > 0 \end{cases}$$

   multiply it by a positive constan, $c \geq 0$

$$\begin{cases} 0 & , c * (weights * x + biases) \leq 0 \\ 1 & , c * (weights * x + biases) > 0 \end{cases}$$

   Since $cisapositiveconstantnumber$ and the perceptron network is only consider the sign of the output. Therefore, multiply by a positive constant will not change the sign and output of the output for each perceptron. Hence, for multiple perceptron network, the output of the entire network is unchanged.

2. For a single sigmoid neuron network [Not consider w*x +b = 0], $sigmoid(z) = \frac{1}{1+e^{-z}}$, where $z = weights * x + biases$, multiply it by a positive constant c, $z = c * (weights * x + biases)$.
   When $c \to \infty$, if $z > 0$,

$$\lim_{z \to +\infty} e^{-z} = 0$$

   , then the output of this neuron is 1.
   if $z < 0$,

$$\lim_{z \to -\infty} e^{-z} = \infty$$

   , then the output of this neuron is 0.
   Same as perceptron.

3. Multilayer perceptron

```
prob4_3
 [0 0 0] [0.]
 [0 0 1] [1.]
 [0 1 0] [1.]
 [0 1 1] [1.]
 [1 0 0] [1.]
 [1 0 1] [0.]
 [1 1 0] [1.]
 [1 1 1] [1.]
```

4. Sigmoid neurons

```
prob4_4
[0 0 0] [0.569265]
[0 0 1] [0.58501229]
[0 1 0] [0.62245933]
[0 1 1] [0.63314399]
[1 0 0] [0.56986717]
[1 0 1] [0.57508402]
[1 1 0] [0.61732588]
[1 1 1] [0.62831133]
```

5. Two-bit Binary addition