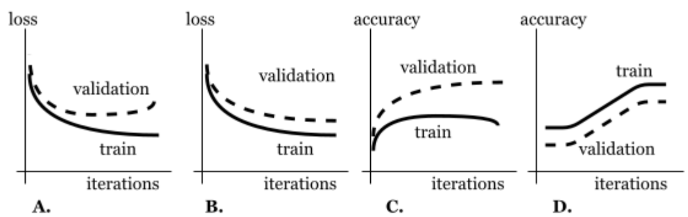
Practice Midterm

Instructor: Paris Perdikaris

Friday, October 23, 2020

1. Which one (or more than one curves) represent over-fitting? Is it possible to encounter a curve like **C** or not? Give a simple justification of you answers.)

   (a) A
   (b) B
   (c) C
   (d) A and C
   (e) A and B



2. Suppose that we wish to calculate $P(Y|X_1, X_2)$ and we have no conditional independence information.

   (a) Which of the following sets of numbers are sufficient for the calculation?
      i. $P(X_1, X_2)$, $P(Y)$, $P(X_1|Y)$, $P(X_2|Y)$
      ii. $P(X_1, X_2)$, $P(Y)$, $P(X_1, X_2|Y)$
      iii. $P(Y)$, $P(X_1|Y)$, $P(X_2|Y)$

   (b) Suppose that you know that $P(X_1|Y, X_2) = P(X_1|Y)$ for all values of Y, X_1, X_2. Now which of the above three sets are sufficient?

3. Briefly explain what is meant by over-fitting. Is it true that if you choose the hyper-parameters (e.g. number of hidden units in a neural network) well, then there will be no over-fitting? Why or why not? (Either YES or NO is acceptable, as long as you justify your answer.)

4. Which of the following is (or are) true about optimizers?

   (a) We can speed up training by employing an optimizer that uses a different learning rate for each weight.
   (b) Reducing the batch size when using Stochastic Gradient Descent always improves training.
   (c) It does not make really sense to use Stochastic Gradient Descent to train a linear regression model because linear regression is convex.
   (d) All of the above.

5. Recall that for logistic regression the gradient of the binary cross-entropy loss is given by

$$\frac{\partial}{\partial \theta_j}\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{N}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)x_{ij}.$$

Which gradient descent update rule below is correct for logistic regression with a learning rate of η? (Choose one or more)

(a) $\theta_j \leftarrow \theta_j - \eta \frac{1}{N}\sum_{i=1}^{N}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)x_{ij}$ (simultaneously update for all j)

(b) $\theta_j \leftarrow \theta_j - \eta \frac{1}{N}\sum_{i=1}^{N}(\frac{1}{1+\exp(-\boldsymbol{\theta}^T\boldsymbol{x}_i)} - y_i)x_{ij}$ (sim. update for all j)

(c) $\theta_j \leftarrow \theta_j - \eta \frac{1}{N}\sum_{i=1}^{N}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)\boldsymbol{x}_i$ (simultaneously update for all j)

(d) $\theta_j \leftarrow \theta_j - \eta \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{\theta}^T\boldsymbol{x} - y_i)\boldsymbol{x}_i$

(e) None of the above.

6. Your training set is $\mathcal{D} = \{(x^{(1)}, y^{(1)}), (x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$. Assume that your model for the data is

$$y^{(i)} \sim \text{Laplace}(\theta^T x^{(i)}, 1).$$

The probability density function of the Laplace distribution with mean μ and scale parameter b is given by:

$$p(x|\mu, b) = \frac{1}{2b}\exp(\frac{-|x-\mu|}{b}).$$

1) Derive the loss function you would use to train this model using maximum likelihood estimation (MLE) on the training data-set $\mathcal{D}$.
2) Considering a zero-mean Laplace prior for the model parameters, i.e. $p(\theta) = \lambda \exp(-\lambda|\theta|)$, derive the loss function you would use to perform maximum a-posteriori (MAP) estimation.
3) Derive the Gradient Descent update rule for the loss corresponding to the MAP estimate.