Homework 2

Fall 2019
(Due: Friday October 11, 2019)

Introduction

This assignment is on deep neural network *regularization*, which is covered in Chapter 7 of the recommended text.

Exercises

1. (Definition)
   a) Define *regularization* for machine learning algorithms.
   b) Why is regularization needed in machine learning?
   c) State one regularization method (that is not already listed in Chapter 7 of the text) used in machine learning.

2. (L^2 Parameter Regularization)
   Consider the weight decay penalty, which is an L^2 parameter norm regularizer used to push model weights closer to the origin. Here, we can consider a model with no bias terms in order to simplify the presentation. Therefore, the model parameters, $\boldsymbol{\theta}$, are just the weights, $\boldsymbol{w}$. The objective function for such a model is given as follows:

$$\tilde{J}(\boldsymbol{w};\boldsymbol{X},\boldsymbol{y}) = \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} + J(\boldsymbol{w};\boldsymbol{X},\boldsymbol{y}). \tag{1}$$

   In (1) above, $J(\cdot)$ is the standard objective function, $\tilde{J}(\cdot)$ is the regularized objective function, $\alpha \in [0,\infty)$ is a hyperparameter that weights the relative contribution of the norm penalty (if $\alpha = 0$, then no regularization is applied), and $\boldsymbol{X}$ and $\boldsymbol{y}$ are the model inputs and outputs, respectively. The gradient of (1) w.r.t. $\boldsymbol{w}$ can be represented as

$$\nabla_{\boldsymbol{w}}\tilde{J}(\boldsymbol{w};\boldsymbol{X},\boldsymbol{y}) = \alpha\boldsymbol{w} + \nabla_{\boldsymbol{w}}J(\boldsymbol{w};\boldsymbol{X},\boldsymbol{y}), \tag{2}$$

   where a single gradient step to update the weights - with learning rate ϵ - can be performed by:

$$\boldsymbol{w} \leftarrow (1-\epsilon\alpha)\boldsymbol{w} - \epsilon\nabla_{\boldsymbol{w}}J(\boldsymbol{w};\boldsymbol{X},\boldsymbol{y}). \tag{3}$$

   a) Through a quadratic approximation of the objective function, provide an interpretation - supported by a mathematical derivation - of how L^2 regularization affects the weight updates, over the course of training, in terms of the eigenvectors and eigenvalues of the Hessian, $\boldsymbol{H}$, of $J(\cdot)$ w.r.t. $\boldsymbol{w}$ evaluated at $\boldsymbol{w}^*$.
   b) Discuss the impact of the condition number of $\boldsymbol{H}$ on the performance of the L^2 regularizer.

3. (L^1 Regularization)
   Assume a quadratic approximation of the L^1 regularized objective function where no correlation exists between the input features (for example, through PCA pre-processing).

a) Give an interpretation of the L^1 regularizer - supported by a mathematical derivation - in terms of the eigenvalues of the Hessian of $J(\cdot)$.

b) Discuss the relationship between the L^1 regularizer and feature selection.

4. (Constrained Optimization)

a) Explain how norm penalties can produce local optima.

b) How do explicit constraints with reprojection help overcome this issue? More specifically, illustrate the advantage, in terms of optimization, when high learning rates are used.

c) A regularization strategy was suggested in 2012 where the norm of each column of the weight matrix, of a single neural network layer, is constrained. Why is this method useful and how does it help regularization? Explain your answer in terms of avoiding dead units in a neural network.

5. (Regularization and Under-Constrained Problems)

a) What is the role of regularization in stabilizing under-constrained, or underdetermined, problems? Comment on the the connection of this to Occam's razor.

6. (Dataset Augmentation and Noise Robustness)

a) Discuss the application of dataset augmentation for image object recognition. Specifically, state some operations that can be performed on images as regularization techniques.

b) For images of letters and numbers, are horizontal flips and 180° rotations appropriate data augmentation techniques? Why or why not?

c) Consider the least-squares cost associated with training a function $\hat{y}(\boldsymbol{x})$ that maps a set of features $\boldsymbol{x}$ to a scalar y. The cost between the model's predictions, $\hat{y}(x)$, and the true values, y is given by:

$$J = \mathbb{E}_{p(x,y)}[(\hat{y}(\boldsymbol{x}) - y)^2]. \tag{4}$$

Assume that we also include a random perturbation $\epsilon_{\boldsymbol{w}} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \eta \boldsymbol{I})$ of the network weights (i.e., we inject a small amount of noise into $\boldsymbol{w}$). Describe the effect of $\epsilon_{\boldsymbol{w}}$ on the structure of the optimal solution.

d) Describe the effect of *label smoothing* by explaining how it is achieved (mathematically). Further, explain how softmax label smoothing works.

7. (Generative Training)

a) Explain what is meant by both the purely generative training criterion and the purely discriminative training criterion.

b) Explain how each of the following can be used to strike a balance between the generative and discriminative training criteria:

[i] Semi-Supervised Learning

[ii] Multi-task Learning

[iii] Parameter Sharing

8. (Early Stopping)

a) What is the *validation set*? Explain how it is different from the *testing set*, and what it is used for.

b) Explain the term *early stopping* on a high level (not algorithmically) and why it is used. (**Hint**: Explain its usage in terms of the U-shaped validation loss over time.)

c) After achieving the purpose of *early stopping*, it is desirable to add the validation set to the training set. Discuss the challenges of determining the number of training rounds (epochs) once the validation set is combined with the training set (i.e., why is choosing the number of epochs a challenge despite that being the goal of *early stopping?*).

d) Similarly, discuss the challenges of continuing to train with the validation set (resume training instead of re-training).

9. (Early Stopping as an L^2 Regularizer)

   (a) Show - using a mathematical derivation - how early stopping is equivalent to L^2 regularization, with an automatic setting of the regularization parameter α, under a quadratic approximation of the cost. (**Hint:** Consider a simple setting where there are no bias terms ($\boldsymbol{\theta} = \boldsymbol{w}$) and begin by modeling the regularized cost function, $\tilde{J}(\cdot)$, in terms of $J(\cdot)$, the empierically optimal weight values, $\boldsymbol{w}^*$, and the Hessian of J with respect to w evaluated at $\boldsymbol{w}^*$.)

10. (Bootstrap Aggregating (Bagging) and Boosting)

    (a) Describe how *bagging* works. Can *bagging* be useful even if the entire training dataset is used for training each model? Why or why not?

    (b) When using *boosting*, how can an individual neural network be interpreted as an ensemble?

Bonus

11. (Dropout)

    (a) Describe an interpretation of Dropout as using a *bagged* ensemble of exponentially many neural networks.

    (b) Describe the differences between dropout training and bagging training. (**Hint:** Recall parameter sharing.)

12. (Dropout Continued)

    (a) Define and explain the *weight scaling inference rule*. State one experimental result in the literature that compares it to the Monte Carlo approximation (What metric was used to claim this experimental result?).

    (b) Explain each of the following:

        [i] Dropout boosting

        [ii] DropConnect

        [iii] Using non-sparse masks