HSD5230 High Performance Computing Homework 15

Miao Cai[*]

2019-04-27

1. Depth versus width of a neural network
(a) Define the depth and width of a neural network in lay terms.
(b) How might you choose depth/width of a network?

- The number of layers (functions) is defined as the **depth** of a neural network. The complexity of the hidden layer (functions) is defined as the **width** of a neural network.
- We should start with a rough guess of the depth and width based on prior knowledge. Then try variations of the neural network and pick the best one based on the prediction performance (test error).

2. Forward propagation
(a) What is the final output from forward propagation?
(b) What is the input?
(c) Define forward propagation in lay terms.

- What is the final output from forward propagation?
- What is the input?
- Define forward propagation in lay terms.

3. Back propagation
(a) What is the goal of backpropagation? How does it differ from forward propagation?
(b) Define backpropagation in lay terms.

4. Activation functions
(a) Describe 3 different activation functions potentially used by a hidden layer. How do they differ?

5. Activation functions for output units versus hidden layers
(a) Why is the choice of activation function for the output unit (output layer) more constrained compared to the activation functions for hidden layers? What considerations are there when choosing this activation function?

6. Universal approximation function theorem
(a) Does the universal approximation function theorem mean we should prefer shallow, wide networks? Why or why not?

[*] Department of Epidemiology and Biostatistics, College for Public Health and Social Justice, Saint Louis University. Email miao.cai@slu.edu

7. When implementing parameter norm penalties, we could fit a different penalty strength for every neuron or layer in the network. Why might we decide against that and fit the same penalty across all layers?

It can be expensive to search for the correct value of multiple hyperparameters, so it is still reasonable to use the same weight decay at all layer just to reduce thte size of search space.

8. When implementing a parameter norm penalty (like L1 or L2), do we penalize the bias terms for each layer, or just the weights? Why?

We do not induce too much variance by leaving the biases unregularized, while regularizing the bias parameters can induce a significant amount of underfitting. Therefore, we only penalize the weights of the affine transformation at each layer and leave the biases unregularized.

9. What are the different effects of L1 versus L2 regularization in deep learning?

Compared to L^2 regularization, L^1 regularization results in a solution that is more sparse. The sparsity here refers to the fact that some parameters have an optimal value of zero. This sparsity property of L^1 regularization has been extensively used as a feature selection mechanism.

10. Explain why early stopping can be considered a form of regularization. How do we know when to stop?

We can think of early stopping as a very efficient hyperparameter selection algorithm, with the number of training steps as another hyperparameter. We stop when no parameters have improved over the best recorded validation error for some pre-specified number of iterations.

11. Pick one regularization technique covered in the chapter (other than parameter norm penalties like L1/L2 or early stopping). Briefly explain what the technique is and why it works to regularize the model.

Parameter sharing is forcing sets of parameters to be equal. In this technique, we are interpreting various models ir model components as sharing a unique set of parameter. The major advantage is that only a subset of the parameters (the unique set) needs to be stored in memory.