Tuning Neural Networks - Introduction

[(https://github.com/learn-co-curriculum/dsc-tuning-neural-networks-intro-v2-4)](https://github.com/learn-co-curriculum/dsc-tuning-neural-networks-intro-v2-4) [(https://github.com/learn-co-curriculum/dsc-tuning-neural-networks-intro-v2-4/issues/new)](https://github.com/learn-co-curriculum/dsc-tuning-neural-networks-intro-v2-4/issues/new)

Introduction

Now that you have a general sense of the architecture of neural networks and some of their underlying concepts, its time to further investigate how to properly tune a model for optimal performance.

Regularization

You've seen regularization before in many other models including linear regression. For example, recall the L1 and L2 penalties which modify ordinary linear regression. These updated loss functions can help tune models so they do not overfit to the training data. For neural networks, you'll use a surprisingly similar process in order to achieve well trained models that are neither overfit nor underfit.

Normalization

Another modeling problem occurs when one gets trapped into a local minimum when searching for an optimal solution using an iterative approach such as gradient descent. One technique for counteracting this scenario is normalizing features. Normalization in deep learning models can drastically decrease computation time, mitigate common issues such as vanishing or exploding gradients, and increase model performance.

Optimization

You'll also look at alternative optimization algorithms. These are of primary interest when one encounters local minimum. Knowing when one has hit such a pitfall can be challenging and typically requires experimenting with different optimization approaches and learning rates.

Convolutional Neural Networks

There are several issues when using densely connected neural networks on image data. Firstly, dense layers learn global patterns rather than local patterns, and densely connected networks can really grow very big if we have high resolution images. In this section, you'll see why Convolutional Neural Networks are often preferred over densely connected networks for image processing. Additionally, you'll learn what a convolution operation is, the different building blocks of convolutional neural networks (including filters, padding schemes, strided convolutions, etc.), and the types of network layers that are part of your convolutional neural networks.

Building a CNN from Scratch

Once you understand how CNNs work, you'll practice building one from scratch. You'll learn how to preprocess your image data so your model can be trained using Keras. Just like with densely connected networks, Keras provides an extremely user-friendly tool to build CNNs.

Visualizing Intermediate Activations

As with densely connected networks, CNNs are complicated networks that are considered a "black box" tool with little insight in what's happening in the network layers. However, when using CNNs, you're essentially changing your image through filters in every layer. You'll learn to get some insight in your black box models by visualizing the intermediate layers in your CNNs!

Summary

In this section, you'll extend your deep learning knowledge by learning about regularization, normalization, and convolutional neural networks.