A comparison of three machine learning algorithms for predicting rainfall amount

Zachary DeStefano, Elmira Forouzmand, and Daniel Quang

March 18, 2015

1 Introduction

Regression is a common problem in the field of machine learning. It can be formulated as follows: Given some training data D, a set of n points of the form $D = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^p, y_i \in \mathbb{R}\}$, find a prediction function f that minimizes the error function. One additional constraint commonly encountered in regression problems is the requirement that all labels y_i are non-negative. Typically, the error function is the mean-squared error (MSE):

$$\frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}_i) - y_i)^2$$

We explore three different machine learning algorithms: regression trees, k-Nearest Neighbors, and neural networks, and compare their performance on predicting the amount of rainfall at a given location, using remote sensing (satellite) and numerical model data.

2 Materials and Methods

2.1 Data

Training and testing data were downloaded from the in-class Kaggle competition, **How's the weather?**. The training dataset includes 60000 samples and the testing dataset includes 40000 samples. Labels are only available for training data, corresponding to non-negative rainfall amounts at various locations. Features are split into two sets: **X1** and **X2**. **X1** is a set of 91 primary features, and **X2** is a set of 441 features derived from raw IR3 local image patches.

2.2 Implementation

The backpropagation algorithm for neural network training was implemented in Pylearn2, a machine learning Python library. Pylearn2 is built on top of Theano, another Python library that optimizes and stabilizes mathematical expressions and then compiles them to a GPU backend. The Python codes were tested on Ubuntu Linux 14.04 with GCC 4.8.2, NVCC 6.5.12, and Python 2.7.9. Computations were performed using a machine with 6 Intel Xeon cores, an NVIDIA Titan Z graphics processor, and 64 GB memory.

3 Results

3.1 Regression Trees

3.2 k-Nearest Neighbors

3.3 Neural Network

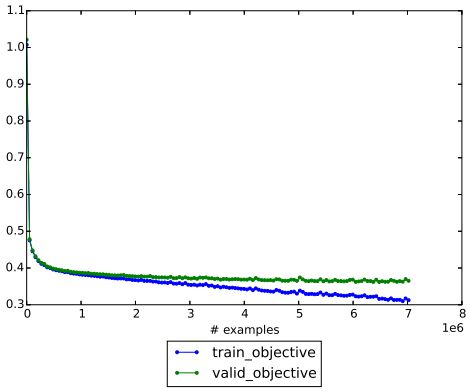
We next trained a neural network to predict rainfall amounts. Briefly, neural networks are statistical learning algorithms inspired by biological neural networks in which interconnected "neurons" can compute values based only on the values of neighboring neurons. Deep learning is a branch of machine learning that explores the performance of stacking many hidden layers of neurons, through which multiple levels of abstraction can be inferred. This is advantageous for problems with many features because deep neural networks can infer the optimal set of features to use without any manual feature selection. Neural networks have recently become more popular due to their state-of-the-art performances in machine learning problems such as image and speech recognition, the development of advanced deep learning techniques such as dropout and unsupervised pretraining, and the application of powerful highly parallelized graphical processing units.

To study how deeper networks perform at regression, we trained neural networks with between 1 and 5 hidden layers (data for model with 4 hidden layers not shown). Due to large number of parameters, neural networks have a tendency to overfit if trained for too many training steps or epochs. Therefore, we further split the training data into a training and validation set in a 9:1 ratio. Training is terminated if the validation MSE, which is evaluated at the end of each training epoch, fails to decrease after 10 training epochs. Each hidden layer contains 1000 hidden neurons and the learning rate is fixed at 0.001. Rectified linear units serve as the activation function for computing the values of the hidden neurons and the final output layer is a standard linear layer. All weights were initialized by randomly drawing from a Gaussian distribution with mean 0 and standard deviation 0.01.

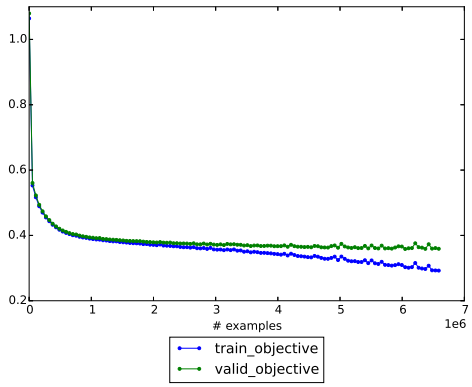
Learning curves for the neural network training clearly displays evidence of overfitting (**Figure 3**). Optimal performance on the validation was reached around 122 training epochs. Based on the performance on the validation set, we selected the neural network with 2 hidden layers as our "best" model. We retrained this neural network with the full training data for 122 epochs and evaluated. Because linear output layers are not strictly non-negative, we apply an additional preprocessing step of converting all negative predictions to 0 before evaluating the trained model on the testing set. Evaluating this model on the test data, we achieved an MSE of 0.364.

4 Discussion and conclusions

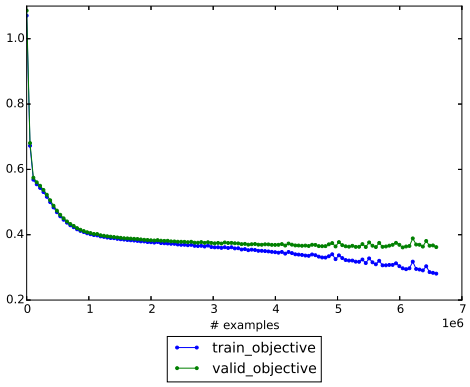
In this paper, we tuned and compared three different machine learning regression models and compared their performance on predicting rainfall amounts. Based on the performance on the test data, k-Nearest Neighbors performs the worst, while regression trees and neural networks achieve very



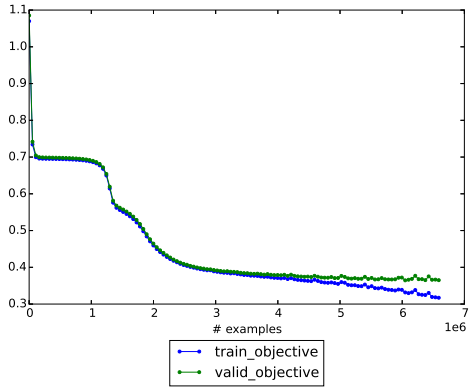
$N_{hidden} = 1$
Train MSE: 0.316
Validation MSE: 0.362



$N_{hidden} = 2$
Train MSE: 0.303
Validation MSE: 0.358



$N_{hidden} = 3$
Train MSE: 0.297
Validation MSE: 0.361



$N_{hidden} = 5$
Train MSE: 0.332
Validation MSE: 0.365

Figure 1: Learning curves for neural network training showing the MSE (y-axis) on training data (blue) and validation data (green) as a function of the number of training samples processed (x-axis). Below each plot is the number of hidden layers in the neural network, and the training and validation MSE of the model yielding the lowest validation MSE during training.

similar performance. Future work can focus on improving performance on the individual models. For k-Nearest Neighbors, we can try weighing the k nearest neighbors differently instead of uniformly like we did in this paper. Based on the overfitting observed in the neural network training, we can explore how dropout improves performance. Applying momentum training, which adjusts the learning rate throughout training, is another option. Different ensemble strategies for regression trees, such as AdaBoost and Extra Trees, may also improve performance. Another avenue of future interest is to combine results from multiple models, including the 3 models considered in this paper, into an ensemble model.

5 Authors' contributions

All authors wrote the manuscript. ZD implemented regression trees. EF implemented k-Nearest Neighbors regression. DQ implemented neural networks. All authors read and approved the final manuscript.