CS229a Project Final Report
James Whitbeck
jamesbw@stanford.edu
Fall 2011

Pedestrian Detection in Images

Abstract

Three machine learning models are applied to images taken from a car, the objective being to determine whether a pedestrian is present. A logistic regression is first applied with satisfactory results. A neural network is then trained to perform better, even though convergence is very slow. Finally an SVM classification is attempted, but performs less well.

Introduction

With electronics ever more present in the automobile industry, automated decision making is becoming a key feature in modern cars. In particular, the ability to detect pedestrians and provide quick reaction to their presence is an area of active research. Not only may it offer assistance to the driver, it may also, one day, enable safe autonomous self-driving cars.

Munder & Gavrilla [2006] collected a large dataset of images of pedestrians and non-pedestrians, in collaboration with automobile maker Daimler, and applied machine learning techniques to classify them. To encourage improvements to their approach, they released this dataset to the public. In more recent research [Enzweiler & Gavrilla 2009; Keller, Enzweiler & Gavrilla 2010; Enzweiler, Eigenstetter, Schiele & Gavrila 2011] the scope is expanded to include wider angles, stereo views and partially ocluded pedestrians.

This project will be using the initial dataset from 2006. It includes 3 training sets and 2 test sets, eaching containing 4800 images of pedestrians and 5000 images without pedestrians. Each image is grayscale with 36x18 pixels. The pedestrians images are cropped and centered around the pedestrian. Each pedestrian image is also mirrored and randomly shifted vertically or horizontally to produce more pedestrian data.

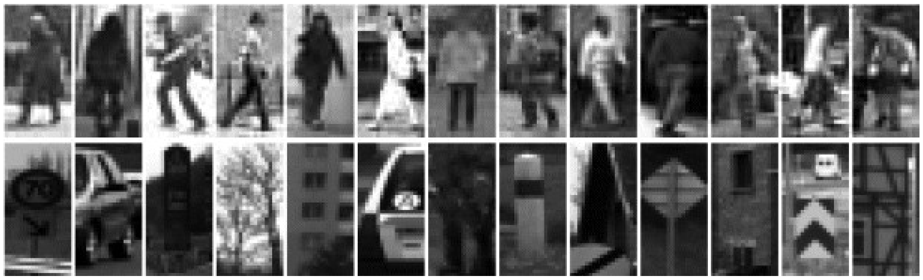

Figure 1: Examples of pedestrian and non-pedestrian images

In this project, we will apply both logistic regression and neural networks to classify the images into two classes: pedestrian and non-pedestrian. The logistic regression will give us a first view of the data and the neural network will be a more advanced approach. In both cases, we will use all three training sets for training, one test set for cross-validation and the second test set for actual testing. An SVM is also trained on one of the training sets.

We will analyze the impact of varying the number of training examples, the regularization constant and the number of iterations so as to avoid both over-fitting and under-fitting.

Logistic regression

Logistic regression was applied to the training sets and tested on the cross-validation set (Test1). In order to figure out whether we were over-fitting or under-fitting the data, we tried out different values of m (the number of training examples), λ (the regularization parameter) and Maxiter, the number of iterations for the cost-minimization algorithm.

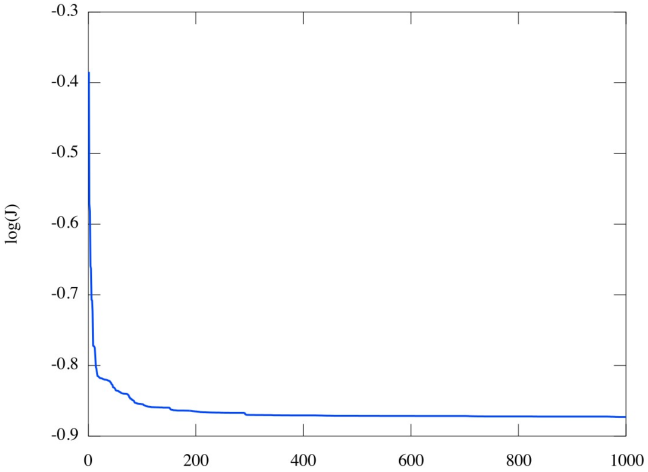

Figure 2: Cost (log J) as a function of the number of iterations

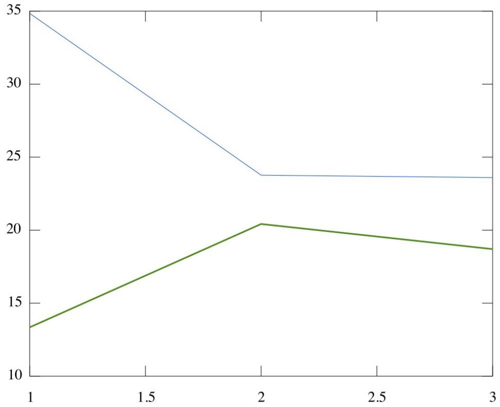

Figure 3: Cross validation (blue) and training (green) errors as a function of the number of training sets used (1,2 or 3)

Figure 2 shows the cost function leveling out after approximately 300 iterations. This is the value we will use from now on. We then trained the data on increase amounts of training data. First, we used just training set 1 (Training1) then we added Training2 and finally Training3 too. As a reminder, each training set have 4800 pedestrian examples and 5000 non-pedestrian examples. Figure 3 shows that with just one training example (m = 9800), the cross-validation error is much higher than the training error. With 2 and 3 training sets (m = 19,600 and m = 29,400), the gap narrows considerably. However the error for both training and CV is around 20%, which is quite high and suggests under-fitting (high bias).

This analysis is confirmed by plotting the training and CV errors as a function of the regularization parameter λ. Figure 4 shows that there is no improvement in the CV error when increasing λ. When λ become very large, both errors increase sharply and get closer to each other, indicating that θ is being reduced to a constant.

It appears that with a training set of almost 30,000 examples and 648 features (36x18 pixels), logistic regression is under-fitting our data. With all the training sets, lambda = 0 and Maxiter = 300, we get the results in table 1. In the next section, we will train a neural network to perform better. The neural network allows adding many new features easily by increasing the number of hidden layers and their sizes.

Training Error	CV error	Test error
18.71%	23.54%	17.43%
Table 1: λ = 0, Maxiter = 300, m = 29,400		

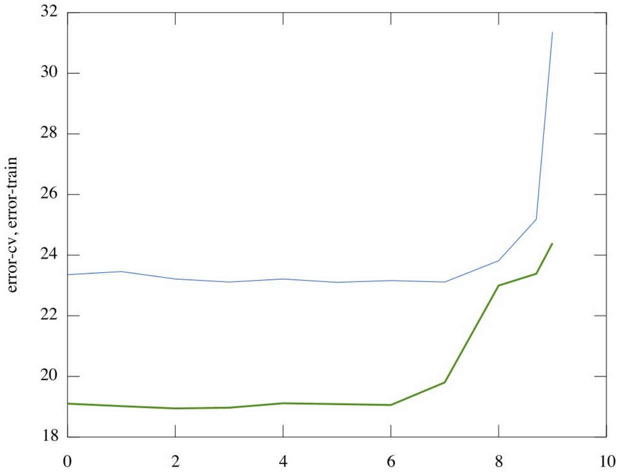

Figure 4: Cross validation (blue) and training (green) errors as a function of log10(λ)

Neural network

Several regularized neural networks were trained against the three training sets, with both 1 and 2 hidden layers, and with 10 to 300 hidden units. The input layer has 36 x18 +1 = 649 units, whereas the output layer has only one unit, which represents the classification of the image in pedestrian or non-pedestrian classes.

Training on all three training sets (29,400 images) helped a lot in reducing high variance. We further reduced the bias and variance by tuning the regularization parameter λ, the number of hidden layers (1 or 2) and the number of hidden units per layer to minimize cross-validation error.

We then ran the model on the test set for λ = 0.1 and 2 hidden layers of sizes 150 and 25. The results obtained, shown in table 2, are a little better than for the logistic regression. However, they can probably be made even better by running the optimization algorithm longer. Indeed, Figure 5 shows that the cost function J is still decreasing steadily after 1000 iterations. In fact, it was still decreasing after 4000 iterations (for some other value of λ). Our limitation here is time and computational power: with so many training samples and parameters, it took over 3 hours to run 1000 iterations.

Training Error	CV error	Test error
14.14%	21.02%	14.53%

Table 2: λ = 0.1, Maxiter = 1000, m = 29400, hidden_layer_1 = 150, hidden_layer_2 = 25

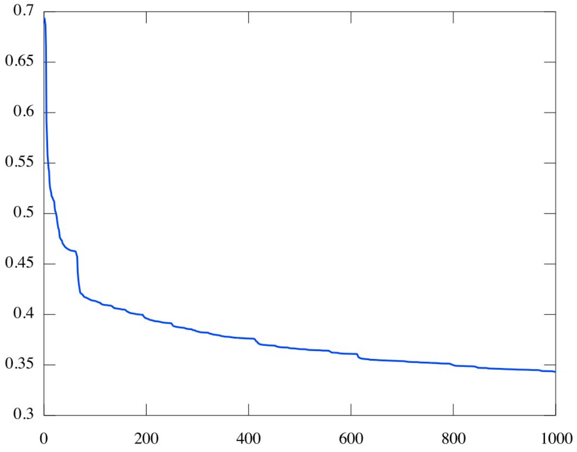

Figure 5: Cost (log J) as a function of the number of iterations

SVM

Because the neural networks took so long to train, we attempted to train an SVM classifier. While the algorithms ran a lot faster, the results were not as good. For starters, the CPU limitation from neural networks was replaced with a memory limitation. Indeed, the SVM algorithm computes X*X' for the kernel and with almost 30,000 rows in X, Octave could not allocate enough memory for the result. So we trained on only one of the training sets. This meant that over-fitting became an issue: with a gaussian kernel and C > 1, we would have 0% training error and 30% cross-validation error. We increased σ and decreased C in order to get the training error closer to the CV-error but they both just got closer to 30%. Further decreasing C caused the SVM to always predict pedestrian (leading to a 52% error for both training and CV).

In the end, we could not find good values for σ and C that led to low training and CV errors.

Conclusion

We presented three basic approaches toward this important subject of computer vision research. Of the three, the neural networks appear to be the most promising, provided enough computational power and time can be devoted to it. The articles referenced below provide additional insights into how to improve these first efforts. In particular, extracting additional features seems to be a good path to pursue.

References

S. Munder and D. M. Gavrila. “An Experimental Study on Pedestrian Classification”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp.1863-1868, Nov. 2006.

M. Enzweiler and D. M. Gavrila. “Monocular Pedestrian Detection: Survey and Experiments”. *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol.31, no.12, pp.2179-2195, 2009.

C. Keller, M. Enzweiler, and D. M. Gavrila, “A New Benchmark for Stereo-based Pedestrian Detection," *Proc. of the IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, 2011.

M. Enzweiler, A. Eigenstetter, B. Schiele and D. M. Gavrila,
Multi-Cue Pedestrian Classification with Partial Occlusion Handling,
IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

http://www.gavrila.net/Research/Pedestrian_Detection/Daimler_Pedestrian_Benchmark_D/daimler_pedestrian_benchmark_d.html