Oliver Tonnesen
V00885732

Assignment 3

CSC 486B
February 8, 2020

Abstract

This assignment implements a linear classifier and demonstrates its performance on the CIFAR 10 data set when equipped with different models and features. This report gives a derivation of the gradient for cross entropy loss used in the classifier and a summary of the results obtained, including best classification accuracy and figures generated during training.

Derivation of the gradient for cross entropy

We begin by rewriting L_i:

$$\begin{aligned} L_i &= -\log \frac{\sum_k t_{ik} e^{s_{ik}}}{\sum_j e^{s_{ij}}} \\ &= -\log \sum_k t_{ik} p_{ik} \\ &= -\sum_k \log t_{ik} p_{ik} \end{aligned}$$

We find $\frac{\partial L_i}{\partial s_{ik}}$:

$$\begin{aligned} \frac{\partial L_i}{\partial s_{ik}} &= \frac{\partial(-\sum_k \log t_{ik} p_{ik})}{\partial s_{ik}} \\ &= -\sum_k \frac{\partial \log t_{ik} p_{ik}}{\partial s_{ik}} \\ &= -\frac{\partial \log p_{iy_i}}{\partial s_{ik}} \\ &= -\frac{1}{p_{iy_i}} \cdot \frac{\partial p_{iy_i}}{\partial s_{ik}} \end{aligned}$$

To find $\frac{\partial p_{ij}}{\partial s_{ik}}$, note that it splits into two cases based on whether or not $j = k$. We first consider the case where $j = k$:

$$\begin{aligned} \frac{\partial p_{ij}}{\partial s_{ik}} &= \frac{\partial \frac{e^{s_{ij}}}{\sum_l e^{s_{il}}}}{\partial s_{ik}} \\ &= \frac{e^{s_{ij}} \sum_l e^{s_{ij}} - e^{s_{ij}} e^{s_{ik}}}{\left(\sum_l e^{s_{il}}\right)^2} \\ &= \frac{e^{s_{ij}}}{\sum_l e^{s_{il}}} \cdot \frac{\sum_l e^{s_{il}} - e^{s_{ik}}}{\sum_l e^{s_{il}}} \\ &= \frac{e^{s_{ij}}}{\sum_l e^{s_{il}}} \cdot \left(\frac{\sum_l e^{s_{il}}}{\sum_l e^{s_{il}}} - \frac{e^{s_{ik}}}{\sum_l e^{s_{il}}}\right) \\ &= p_{ij}\left(1 - p_{ik}\right) \end{aligned}$$

Next, we consider the case where $j \neq k$:

$$\begin{aligned}
\frac{\partial p_{ij}}{\partial s_{ik}} &= \frac{\partial \frac{e^{s_{ij}}}{\sum_l e^{s_{il}}}}{\partial s_{ik}} \\
&= \frac{0 - e^{s_{ij}} e^{s_{ik}}}{\left(\sum_l e^{s_{il}}\right)^2} \\
&= \frac{e^{s_{ij}}}{\sum_l e^{s_{il}}} \cdot \frac{e^{s_{ik}}}{\sum_l e^{s_{il}}} \\
&= -p_{ij} p_{ik}
\end{aligned}$$

So in the end, we get

$$\begin{aligned}
\frac{\partial L_i}{\partial s_{ik}} &= \begin{cases} -\frac{1}{p_{iy_i}} \cdot p_{iy_i}\left(1 - p_{ik}\right) & \text{if } y_i = k \\ -\frac{1}{p_{iy_i}} \cdot \left(-p_{iy_i} p_{ik}\right) & \text{if } y_i \neq k \end{cases} \\
&= \begin{cases} p_{ik} - 1 & \text{if } y_i = k \\ p_{ik} & \text{if } y_i \neq k \end{cases}
\end{aligned}$$

Training curve

The HOG feature was tested with both the SVM and logistic regression models. The results can be seen in Figs. 1, 2, 3, and 4.

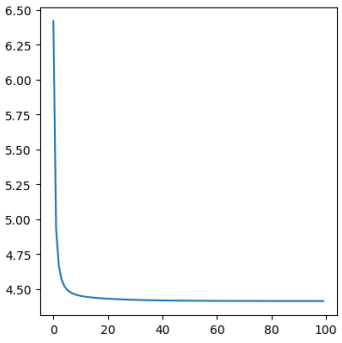

Figure 1: HOG SVM Loss

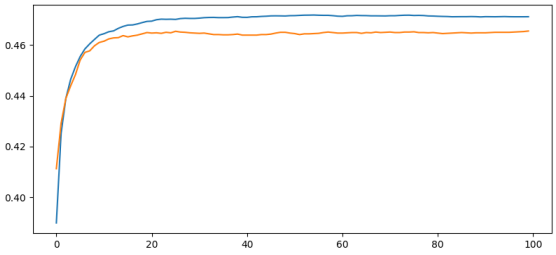

Figure 2: HOG SVM Accuracy

The model yielding the most highest validation accuracy in conjunction with the HOG feature was logistic regression, with an average validation accuracy of 48.92%.

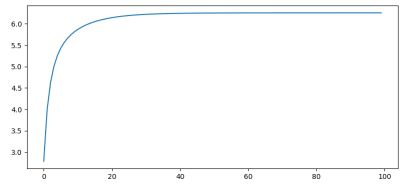

Figure 3: HOG Logistic Regression Loss

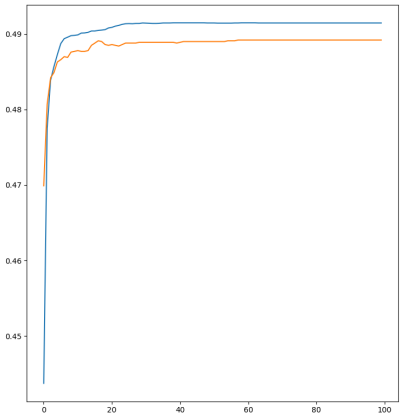

Figure 4: HOG Logistic Regression Accuracy

Cross validation results

We tested the HOG feature with the Logistic Regression model using cross validation with two sets of hyper-parameters: learning rate 0.0001 and reg_lambda 0.1 resulted in 49.29% accuracy, and learning rate 0.00012 and reg_lambda 0.05 resulted in 49.37% accuracy, the highest achieved. Below we give the accuracy and loss graphs for these sets of hyper-parameters:

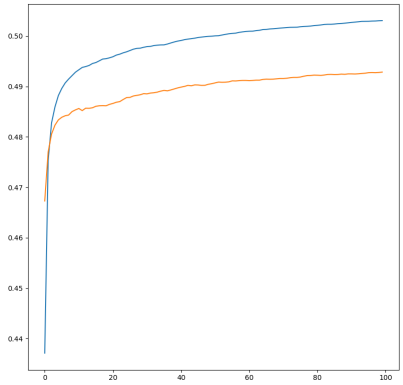

Figure 5: Reg lambda: 0.1, learning rate: 0.0001

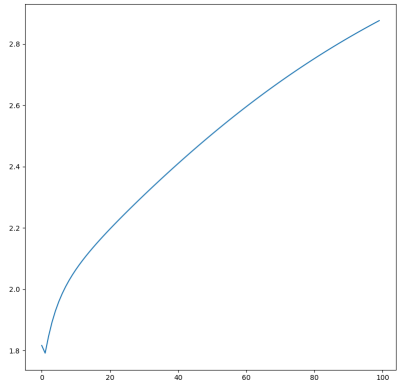

Figure 6: Reg lambda: 0.1, learning rate: 0.0001

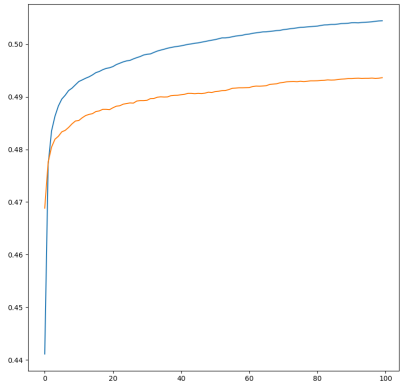

Figure 7: Reg lambda: 0.05, learning rate: 0.00012

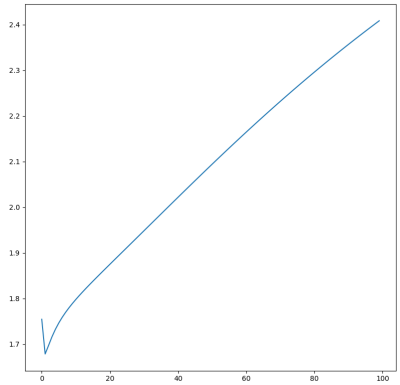

Figure 8: Reg lambda: 0.05, learning rate: 0.00012