Linear Classification

1/4/19

Two major components, a score function and loss function:

**Score Function:** Assume there is a training set x ∈RD, each having a label yi. Let I be from 1,...,N and y­­­I which is from 1,…,K. That is we have N examples, each with dimension D, and K labels. We desire a function f:RD→RK that maps images (in pixel form) to class scores.

We start with the simplest function

In the equation we assume that the image xi has the pixels flattened to a single column vector (1xD). W is a matrix of size KxD and b is a vector of size Kx1. W is called the weights and b is called the bias vector.

Note:

* Out goal is to get W and b to give correct classifications for the entire training set, once we have these we discard the training set

Each row of w represents a template image of the category.

Important to preprocess image as this by centering each pixel at 0.

**Loss Function:** Helps to measure unhappiness with the outcome of the score function. Commonly used loss is called the Multiclass Support Vector Machine. The SMV wants the correct class for the image to have a score higher than the incorrect classes by at least Δ.

Precisely the score function will give a vector of class scores, call that s. The multiclass loss for the ith example image is

The 0 assures that we only care about differences at least Δ. Our goal is to minimize this function.

Look at data for Regularization

**Setting Δ:** Set to 1.0 in all cases.

**Softmax:**

Gradient and Visualizing the Loss Fxn

1/7/19

**Optimizing W**: We wish to find a W that generates the lowest value for the loss function (ie Nonlinear optimization). One method of doing this is generating a random W, then computing the gradient around W using the loss function. We then would like to choose a direction to change our W so that we have the lowest loss function by choosing the direction with the lowest value in the gradient (blind mountain descending analogy). Choosing step size is the most important part, as if it’s too small, we will get consistent good values, but it will take to long, too large, we may “skip” over the lowest values.

**Finding the Gradient**: Use either the analytical or computational method. In computational, use classic slope equation with a low h (commonly 1\*e(-5)) and compute the gradients with that. Tradeoff is that it is a approximation and is computationally expensive. Analytical approach is to take actual partial derivates, much less expensive but prone to error.

Backpropagation Intuitions

1/7/19

**Backpropagation:** Way of computing gradients with recursive application of chain rule.