March 25, 2016

Abstract

1 Overview of Machine Learnnig

1. Logistic Classification
2. Stochastic Optimization
3. Data and Parameter Turning
4. Deep Networks
5. Regularization
6. Convolutional Networks
7. Embeddings
8. Recurrent Models

2 History of Neural Networks

1. Fukushimas Neocognitron - 1980's
2. Lecun's Net -1990
3. Krizhevsky's Alexnet
4. Speech Recognition -2009
5. Computer Vision -2012
6. Machine Translation - 2014

3 Classification

Given set of images and labels in training data. In test data completely new image comes. Classify image. After classification we can do

1. Regression
2. Ranking - In web page. Classify relevant or irrrelevant
3. Reinforcement Learning
4. Detection - Eg : Detect presence or absence of pedestrian

3.1 Logistic Classifier

$$WX + b = Y \tag{1}$$

X - Image Pixels
Y - Labels
W- Weight
b- Bias

3.1.1 Soft Max Function

Softmax function converts scores into probability

$$S(y_i) = \frac{e_i^y}{\sum e_j^y} \tag{2}$$

3.1.2 One Hot Encoding

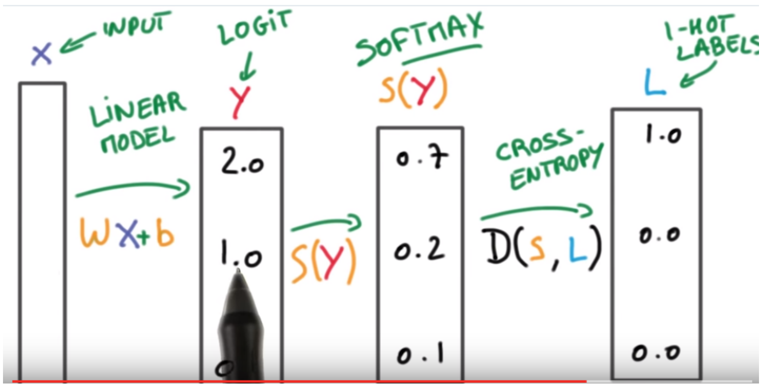
One for correct class and zero to other class labels.

3.1.3 Cross Entrophy

The way to measure distance between two probabilities is called cross entrophy.

$$D(S, L) = -\sum_i L_i log(S_i) \tag{3}$$

Cross Entropy is not symmetric. $D(S, L) \neq D(L, S)$



4 Training Loss

Loss = Average Cross entrophy. We do to minimize distance between similar labels and maximize distance between dissimilar labels.

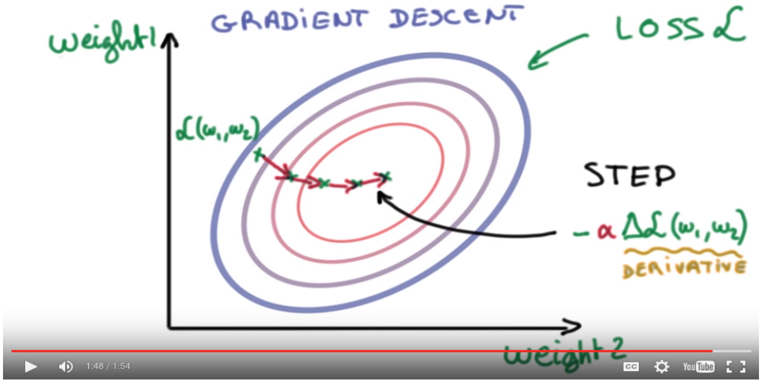
$$L = \frac{1}{N} \sum_i D(S(WX_i + b), L_i) \tag{4}$$

5 Gradient Descend

While taking average to calculate training loss we are taking average of distance between probabilities. Gradient descend for two weights is calculated as follows.

$$GradientDescentforweightw_1andw_2 = \Delta L(w_1, w_2) \tag{5}$$

But in real-time the weight is computed for all parameters.



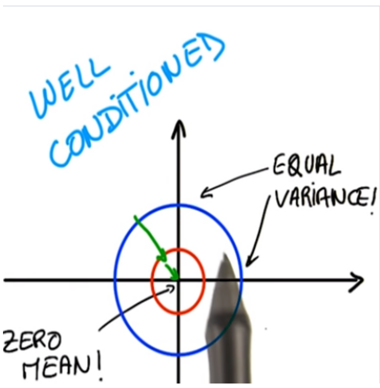
5.1 Big Loss Function because of Numerical Unstability

To overcome Big Loss function keep mean zero and equal variance. Mean

$$X_i = 0$$

Variance

$$\sigma(X_i) = \sigma(X_j)$$



5.2 Normalize Input for Gradient Descend

To normalize pixel input normalize as follows.

$$\frac{R-128}{128} \qquad \frac{G-128}{128} \qquad \frac{B-128}{128}$$

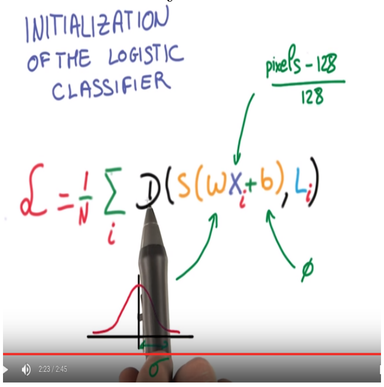
5.3 Initialize Weight for Gradient Descend

Initialize w_0, b_0

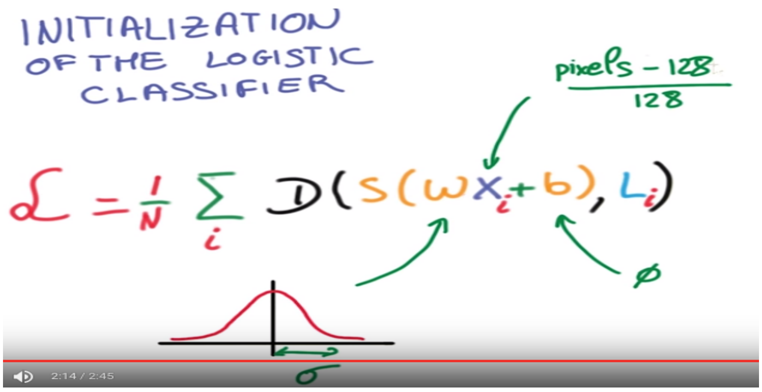
Large value of σ - Distribution has large peaks

Small value of σ - Distribution is very uncertain.

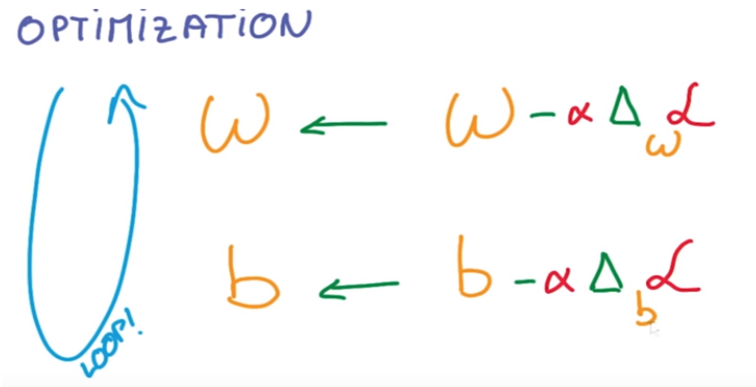
Start with very small value of σ



5.4 Steps to Train Logistic Classifier



1. X_i - Training data is normalized to zero mean and equal variance
2. w - Initialized with random weights
3. Do softmax
4. Do cross entrophy loss
5. calculate average for entire training data
6. Optimization - Compute derivative loss function w.r.to weight


OPTIMIZATION
$w \leftarrow w - \alpha \Delta_w \mathcal{L}$
$b \leftarrow b - \alpha \Delta_b \mathcal{L}$
LOOP!