Machine Learning: Digits Recognition

Angel Chaico ACECOM
Facultad de ciencias, UNI* Diego ,Angelica

29 de diciembre de 2020

I. Introduction

II. Related work

1. Online learning

2. Cost Function

3. Gradient Descent

4. Learning Rate

5. Stochastic Gradient Descent

Nearly all of deep learning is powered by one very important algorithm: Stochastic gradient descent(SGD).Stochastic descent is an extension of the gradient gradient descent algorithm introduced in (**??**). Of the equation (**??**) :

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m} Cost(\theta, x^{(i)}, y^{(j)})$$

The SGD consist in:

1. Randomly shuffle dataset.
2. K repetitions of:

$$\begin{gathered} for \ : i = 1, 2, \cdot, m \\ \theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x^{(i)} \\ (for\ j = 0, 1, \cdot, n) \end{gathered} \quad (1)$$

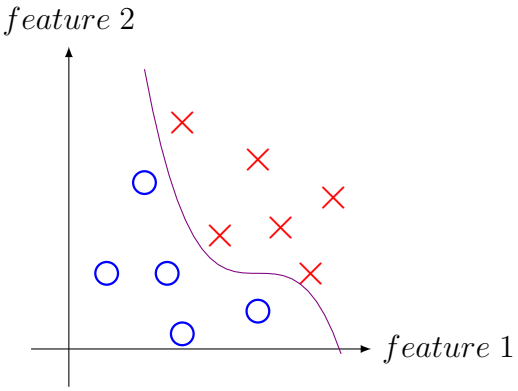
Generally move the parameters in the direccion of the global minimum but not always. It doesn't actually converge in the same sense as batch gradient descent us and what events are doing is wandering around continuously in some region close to the global minimum but it doesn't actually just get to the global minimum and stay there but the practice this isn't a problem because you know so long as the parameters end up in some region there maybe it is pretty close to the global minimum.So lost parameters ends up pretty close to the global minimum that will be a pretty good hypothesis.

K in the (1) you need to do only one if m is very large but in general taking anywhere from 1 through 10 pulses through your data set you know may be fairly common but they're really it depends on the size of your training set.

*Universidad Nacional de Ingenieria

6. Logistic Regression

Classification



$$\boxed{y \in \{0, 1\}}$$

0 : Negative Class(Benign tumor)

1 : Positive Class(Malign tumor)

Hypothesis Representation

Want: $0 \leq h_\theta(x) \leq 1$

I propose:

$$h_\theta(x) = g(\theta^T x); \quad g(z) = \frac{1}{e^{-z}}$$

$$\boxed{h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}} \tag{2}$$

Where the variables:

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1} \; ; \; \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

where defined: $x_0 = 1$.

Graphing the function $g_(z)$:

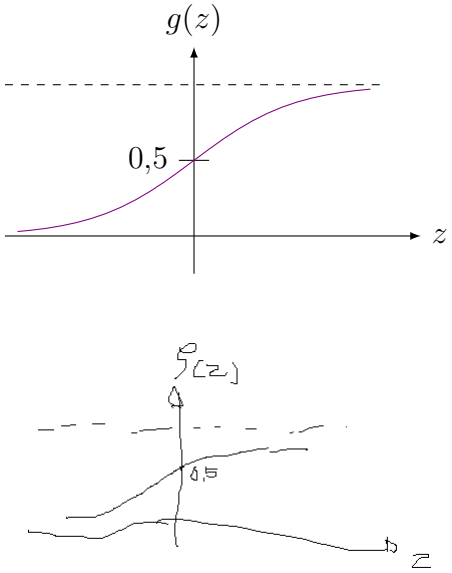


Figura 1: the graphics will make better

Interpretation of hypothesis output: $h_\theta(x)$:Estimated probability that $y = 1$ on input x.

$h_\theta(x)$:Probability that $y = 1$, given x, parametrized by θ

Notation:

$$\boxed{h_\theta(x) = P(y = 1|x, \theta)}$$

$$hence:$$

$$P(y = 1|x, \theta) + P(y = 0|x, \theta) = 1$$

Decision boundary

Of that function:

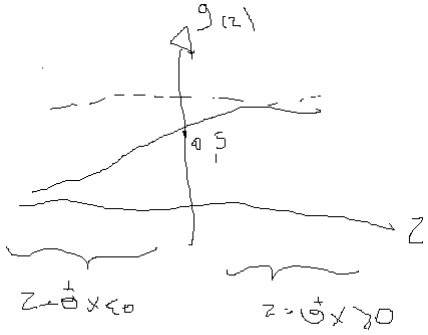


Figura 2: the graphics will make better

Predict $y = 1$ if $h_\theta(x) > 0{,}5$, hence:

$$\boxed{z = \theta^T x > 0}$$

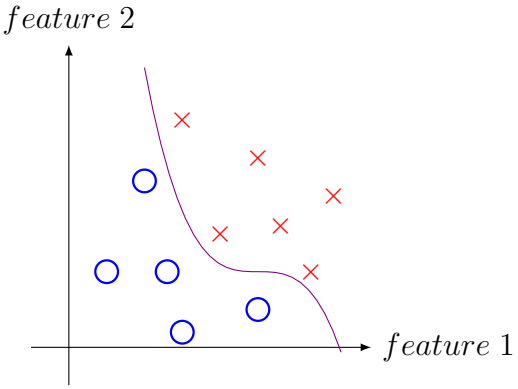
Predict $y = 0$ if $h_\theta(x) < 0{,}5$, hence :

$$\boxed{z = \theta^T x < 0}$$

If $h_\theta(x) = 0{,}5$ hence:

$$\boxed{z = \theta^T x = 0} \quad Decisition\ Boundary$$

Graphing:



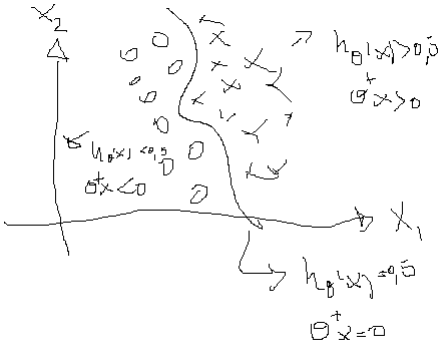


Figura 3: the graphics will make better

Cost Function

$$Cost(h_\theta(x); y) = \begin{cases} -log(h_\theta(x); y)\ ; if\ y = 1 \\ -log(1 - h_\theta(x))\ ; if\ y = 0 \end{cases}$$

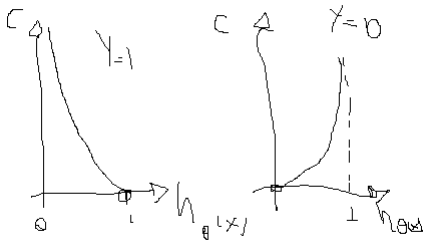


Figura 4: the graphics will make better

Simplied the cost:

$$Cost(h_\theta(x); y) = \\ -ylog(h_\theta(x)) - (1 - y)log(1 - h_\theta(x))$$

Therefore the cost function is:

$$\boxed{J_\theta(x) = -\frac{1}{m}\sum_{1}^{m} Cost(h_\theta(x^{(i)}); y^{(i)})} \quad (3)$$

To fit parameter:θ

$$min_\theta J(\theta)$$

to make a prediction given new x:

$$h_\theta(x)$$

Applying the gradient descent:

$$\boxed{\theta_j := \theta_j - \alpha\frac{\partial J(\theta)}{\partial \theta_j}}$$

$$where:$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

7. Neural Networks

8. Testing and Validating

we must split our data into two sets:the *thaining set* and the *test set*.O course the training set is for train your model. The error rate on new cases is called the *generalization*

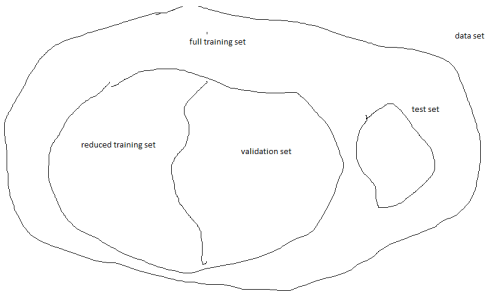
error(or out-of-sample error), and by evaluating your model on the test set, you get an estimate of this error.

If the training error is low(your model makes few mistake on the training set) but the generaization error is high, it means that your model is overfitting the training data.

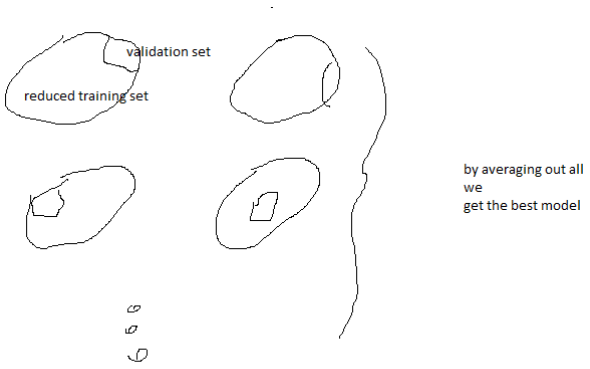
Hyperparameter Tuning and Model Selection

How we selected the better between two model?.One option is to train both and compare how well they generalize using the test set.Suppose you find the best *hyperparameter* values that produces a model with the lowest generalization error, but there is one problem.There problem is that you *measured the generalization error muliple times on the test set, and you adapted the model and hyperparameters to produce the best model* ***for that model***.This model unlikely perform as well on new data.
Solution for this problem:

Holdout validation: Simply hold out part of the training set to evaluate several candidate model and select the best one.The new held-out set is called the *validation set*(development set).More specifically , you train multiple model with various hyperparameters on the reduced training set(full trainig set minus the validation set) and you select the model that performs best on the validation set .After you train the best model on the full training set , and give you the final model.Lastly, you evaluate this final model on the test set to get an estimate of the generalization error.This model work well but if the validation set is small or too large this methods will be imprecise.



Cross validation: In this case we take many small validation sets. Each model is evaluated once per validation set after it is trained on the rest of the data. By averaging out all the evaluations of a model, you get a much more accurate measure of its performance.



There is a drawback:the training time is multiplied by the number of validation sets.

III. Preliminaries

IV. Method

V. Experiments

VI. Conclusion

Acknowledgment

References