Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Hyperparameter Learning, Linear Regression with Nonnegative Target, Logistic Regression

Krisztian Buza

Department of Artificial Intelligence
Eötvös Loránd University
Budapest, Hungary

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Announcements

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Announcements

- Tentative Grading Scheme

85 pts – 100 pts → 5 („very good“)
70 pts – 84 pts → 4 („good“)
55 pts – 69 pts → 3 („average“)
40 pts – 54 pts → 2 („sufficient“)
0 pts – 39 pts → 1 („not sufficient“)

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Announcements

- Quick Test Replacement Test (QTRT)
  - In the same session with the „big test“
  - 11th Dec: „big test“ (60 minutes) + ~~QTRT (30 minutes)~~
  - Due to multiple requests, QTRT will be organized between 16th Dec and 20th Dec
  - You may collect maximal 14 points which replace the result of the 7 worst quick tests

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Students' Presentations

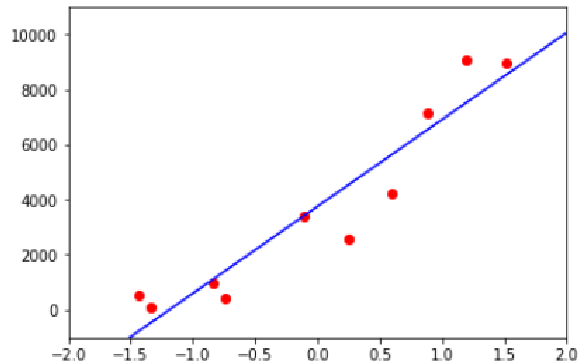
- Everyone in the team should contribute to the presentation
- Team members can distribute the work internally as they wish (e.g., it is not needed that everyone talks, but everyone should do something for the presentation)
- First slide: title of the presentation
  name and Neptun code of all team members
- Last slide: who did what
- Send your slides in PDF(!) format to buza@inf.elte.hu at least two days before the presentation

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Overfitting from the Perspective of RMSE

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Calculation of Root Mean Square Error



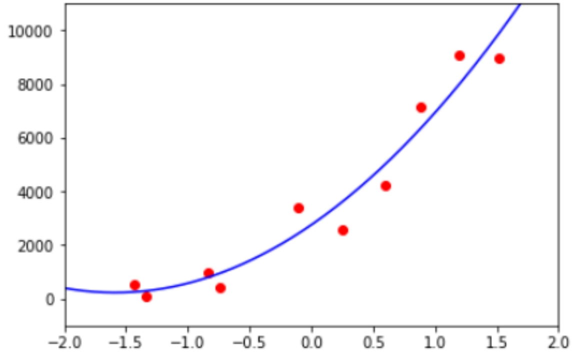
y	$\hat{y}$
549	-770
100	-470
976	1132
441	1433
3401	3435
2600	4537
4241	5638
7150	6539
9100	7541
9000	8542

n = 10

$$\sqrt{\frac{1}{10}\left((549-(-770))^2+(100-(-470))^2+(976-1132)^2+(441-1433)^2+(3401-3435)^2+(2600-4537)^2+(4241-5638)^2+(7150-6539)^2+(9100-7541)^2+(9000-8542)^2\right)}$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

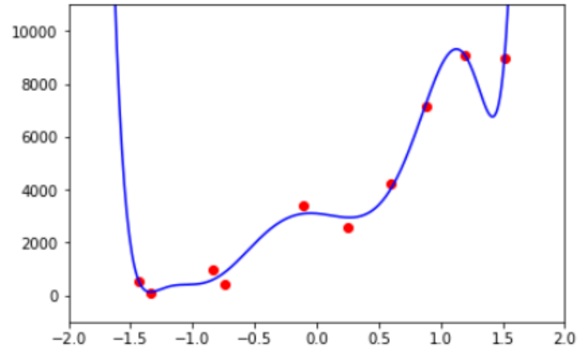
Calculation of Root Mean Square Error



y	$\hat{y}$
549	254
100	293
976	805
441	958
3401	2433
2600	3602
4241	5004
7150	6332
9100	7999
9000	9867

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

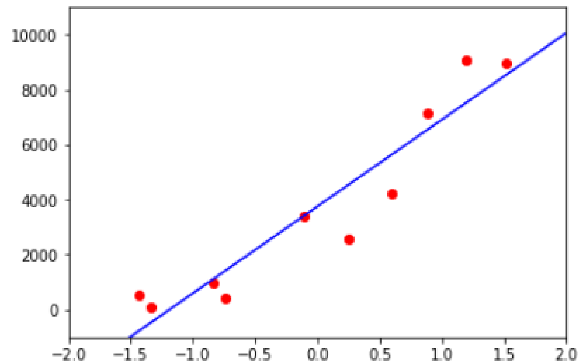
Calculation of Root Mean Square Error

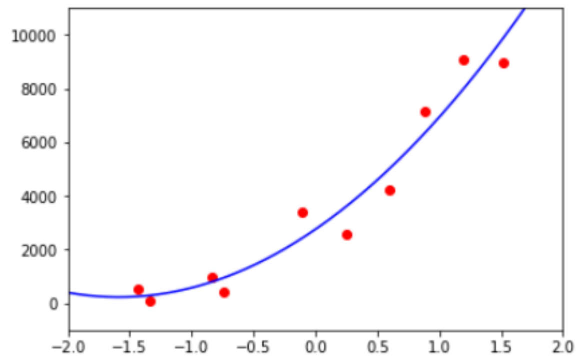


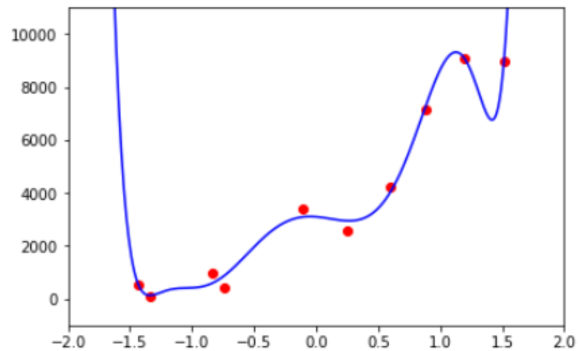
y	$\hat{y}$
549	534
100	131
976	608
441	894
3401	3100
2600	2951
4241	4011
7150	7247
9100	9081
9000	9002

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Root Mean Square Error of the Models



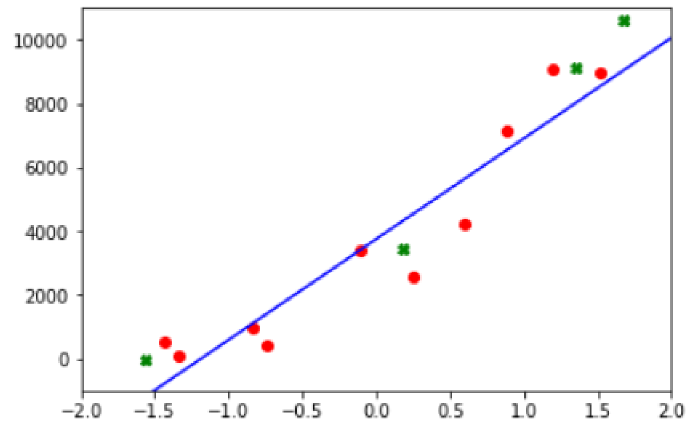




1085.88 745.50 248.66

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

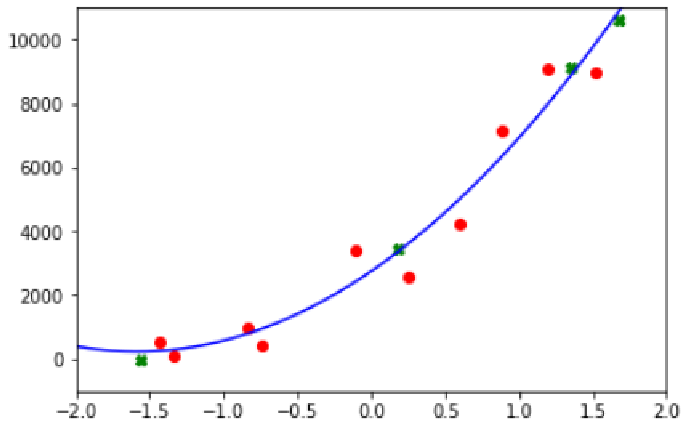
Root Mean Square Error on New Data



y	$\hat{y}$
9	-1171
3464	4337
9125	8041
10609	9043

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

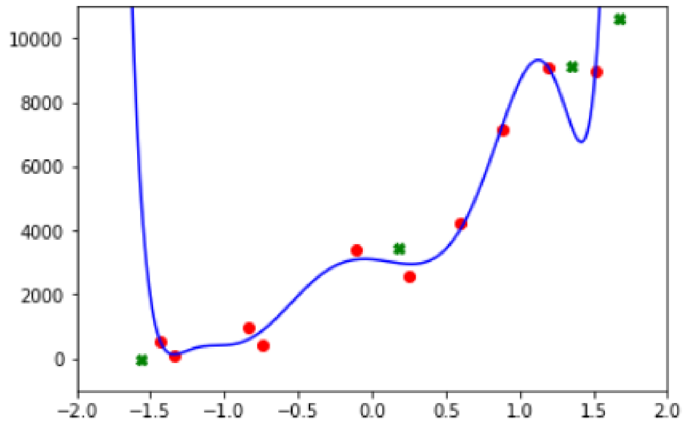
Root Mean Square Error on New Data



y	$\hat{y}$
9	4779
3464	2979
9125	7204
10609	37347

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

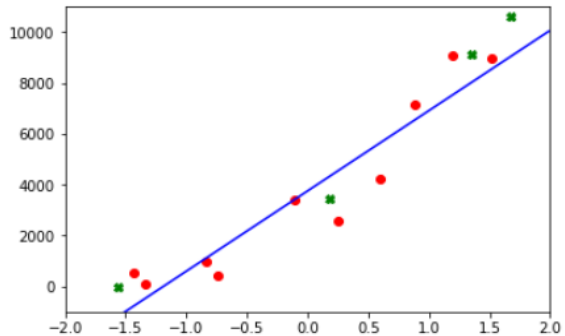
Root Mean Square Error on New Data

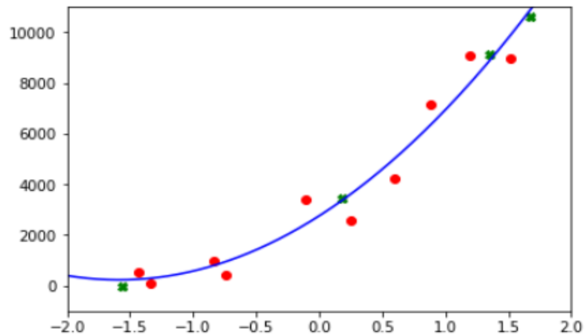


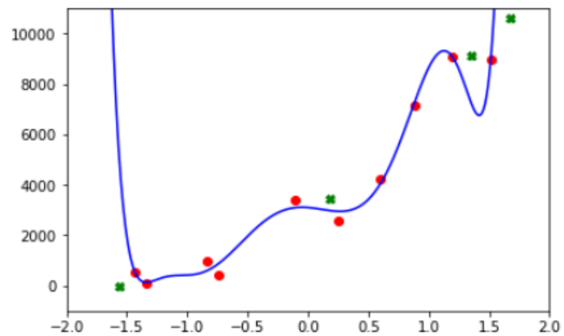
y	$\hat{y}$
9	4779
3464	2979
9125	7204
10609	37347

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Root Mean Square Error of the Models







on training data:

1085.88	745.50	248.66

on new data (test data):

1202.28	209.73	13616.00

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Hyperparameter Learning

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Parameters and Hyperparameters

- Polynomial regression: $\hat{y} = w^{(0)} + w^{(1)}x + w^{(2)}x^2 + ... + w^{(p)}x^p$

  Parameters: $w^{(0)}, w^{(1)}, ..., w^{(p)}$
- Hyperparameters: parameters of the learning algorithm (optimization algorithm), such as
  - Degree of the polynomial
  - Learning rate
  - „Importance“ of the Regularization Term...
- Parameters are detemined by the learning algorithm, whereas hyperparameter may be set by the „expert“

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

How to Find the Appropriate Values of Hyperparameters?

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Hyperparameter Learning

- Determining the best (appropriate) values of hyper-parameters is part of the training process.
- Therefore, we need **completely new** data („Test data 2“) in order to have an unbiased („fair“) estimate of the quality (RMSE...) of the model with „best“ hyperparameters.
- Terminology: Training data → Training data
  Test data 1 → Validation data
  Test data 2 → Test data

(sometimes „validation data“ refers to „test data 2“ and „test data“ is used to refer to „test data 1“)

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Batch Gradient Descent and Stochastic Gradient Descent

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Gradient Descent for Linear Regression

(Regularization is omitted for simplicity)

- We consider models of the form $\hat{y} = \sum_{j=1}^{m} w^{(j)} x^{(j)}$
- Procedure:
  - **Step 1** Set $w^{(1)}, w^{(2)}, \ldots, w^{(m)}$ to some random values
  - **Step 2** for j in $1 \ldots m$

The gradient of the objective function (a.k.a. cost function)

$$w^{(j)} \leftarrow w^{(j)} - \epsilon \boxed{\frac{1}{n} \sum_{i=1}^{n} 2x_i^{(j)} (\hat{y}_i - y_i)}$$

where $\hat{y}_i = \sum_{j=1}^{m} w^{(j)} x_i^{(j)}$

  - **Step 3** Repeat Step 2 as long as you can non-negligibly decrease the error

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Variants of Gradient Descent

- Gradient is calculated using all instances of the training data → **gradient descent**
- Gradient is calculated using a subset of the instances of the training data → **batch gradient descent**
  - Batch size: the number of instances in that subset that is used to calculate the gradient
  - Each time when step 2 is executed, a different subset should be selected
- Gradient is calcualted using a single instance → **stochastic gradient descent**

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Stochastic Gradient Descent for Lin. Reg.

(Regularization is omitted for simplicity)

- We consider models of the form $\hat{y} = \sum_{j=1}^{m} w^{(j)} x^{(j)}$
- Procedure:
  - **Step 1** Set $w^{(1)}, w^{(2)}, ..., w^{(m)}$ to some random values
  - **Step 2** $x_i \leftarrow$ **select a random instance of the training data**
    for j in $1...m$
    $$w^{(j)} \leftarrow w^{(j)} - \epsilon \frac{1}{n} \cancel{\sum_{i=1}^{n}} 2x_i^{(j)} (\hat{y}_i - y_i)$$
    where $\hat{y}_i = \sum_{j=1}^{m} w^{(j)} x_i^{(j)}$
  - **Step 3** Repeat Step 2 as long as you can non-negligibly decrease the error

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Batch Gradient Descent for Lin. Reg.

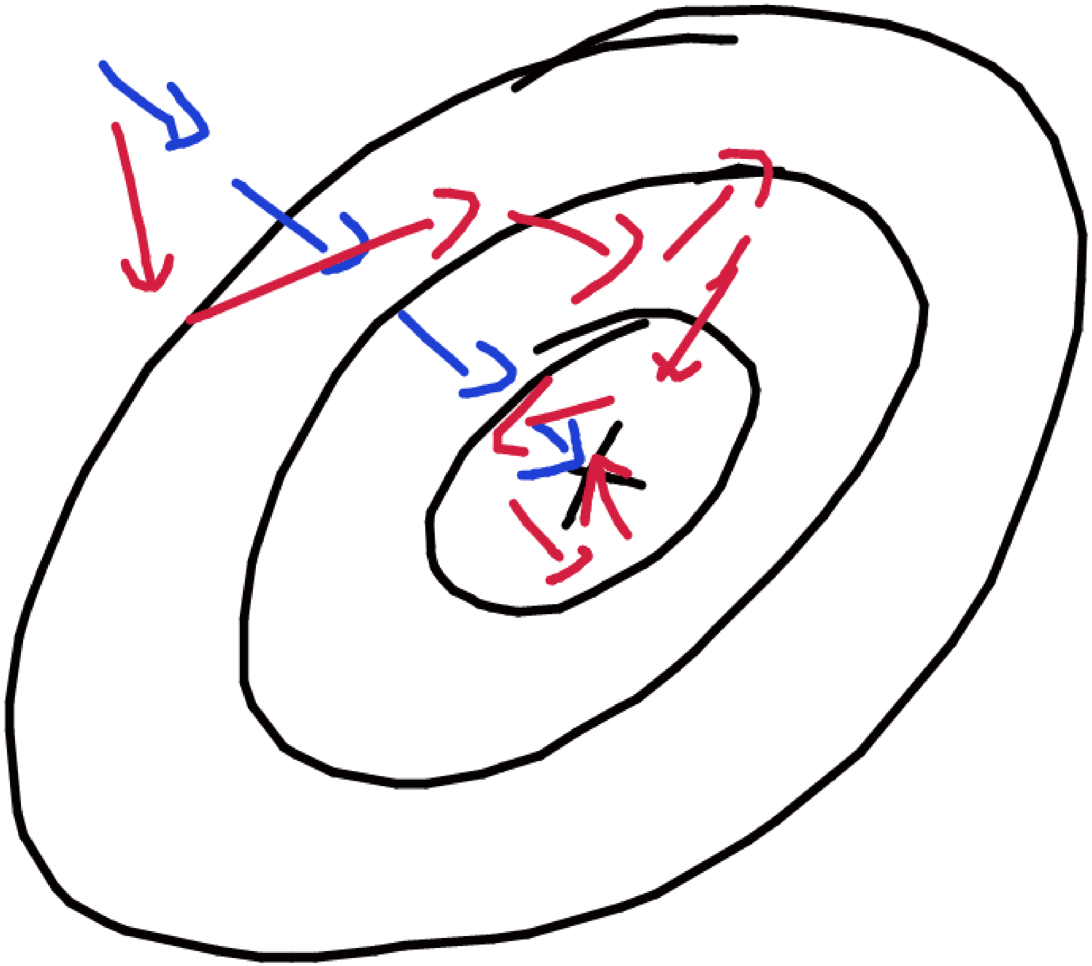
(Regularization is omitted for simplicity)

- We consider models of the form $\hat{y} = \sum_{j=1}^{m} w^{(j)} x^{(j)}$
- Procedure:
  - **Step 1** Set $w^{(1)}, w^{(2)}, ..., w^{(m)}$ to some random values
  - **Step 2** $D' \leftarrow$ **select a random subset of the training data**
    for j in $1...m$
    $$w^{(j)} \leftarrow w^{(j)} - \epsilon \frac{1}{n} \sum_{x_i \text{ in } D'} 2x_i^{(j)} (\hat{y}_i - y_i)$$
    where $\hat{y}_i = \sum_{j=1}^{m} w^{(j)} x_i^{(j)}$
  - **Step 3** Repeat Step 2 as long as you can non-negligibly decrease the error

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

How do Different Versions of Gradient Descent Find the Optimum?

Blue: gradient descent
Red: stochastic gradient descent



Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Linear Regression with Nonnegative Target

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Linear Regression with Nonnegative Target

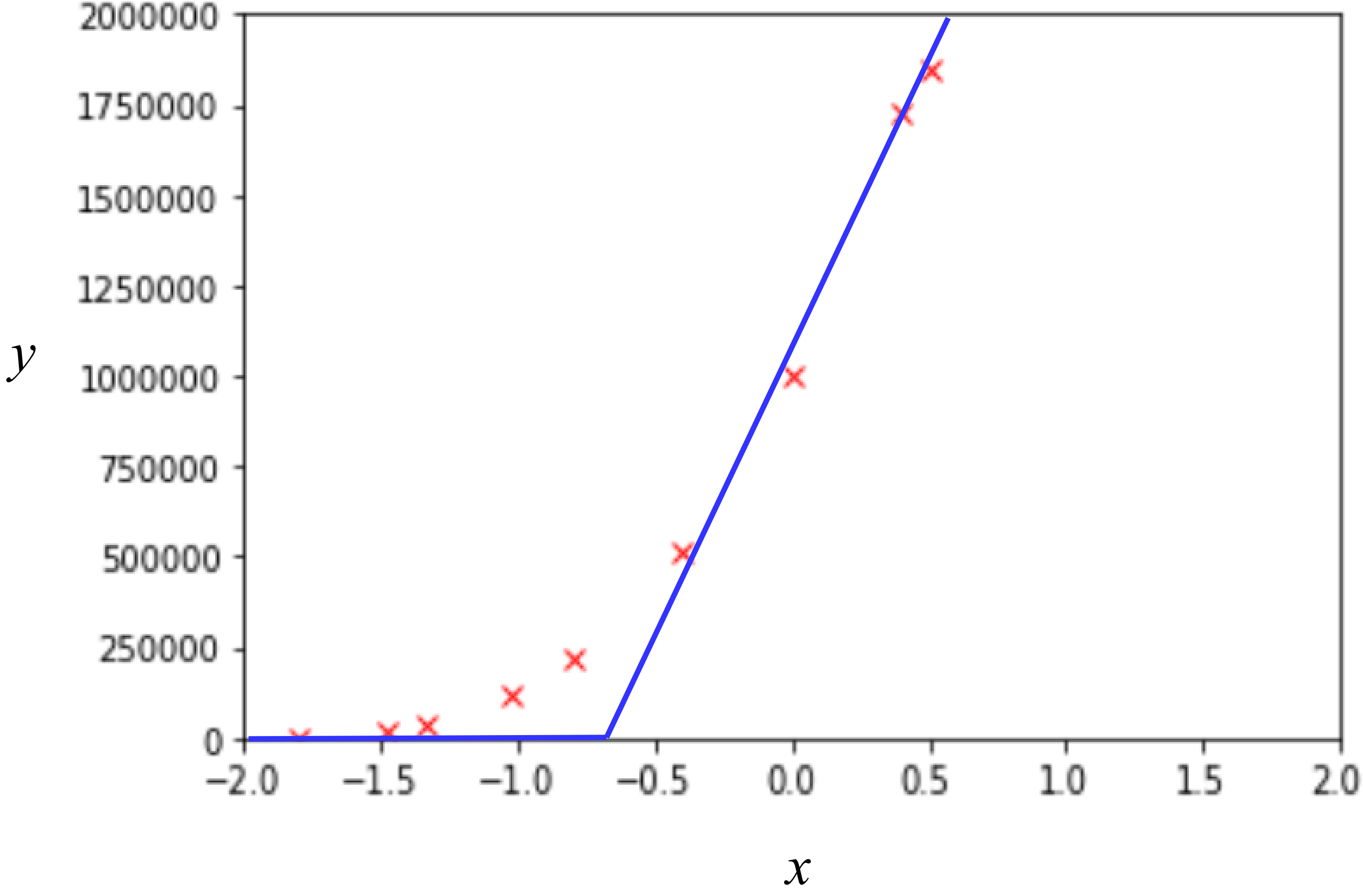
- We consider models of the form

$$z = \sum_{i=0}^{m} w^{(i)} x^{(i)}$$

$$\hat{y} = \begin{cases} z & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Linear Regression with Nonnegative Target



Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Cost Function and Its Partial Derivatives

- Cost function: mean of squared errors
(regularisation is omitted for simplicity)

$$E = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

$$\hat{y} = \begin{cases} z & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases} \qquad z = \sum_{i=0}^{m} w^{(i)} x^{(i)}$$

- Partial Derivative w.r.t. $w^{(j)}$

$$\frac{\partial E}{\partial w^{(j)}} = \begin{cases} \frac{1}{n} \sum_{i=1}^{n} 2 x_i^{(j)} (\hat{y}_i - y_i) & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Cost Function and Its Partial Derivatives

- Cost function: mean of squared errors
  (regularisation is omitted for simplicity)

$$E = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

$$\hat{y} = \begin{cases} z & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases} \qquad z = \sum_{i=0}^{m} w^{(i)} x^{(i)}$$

- Partial Derivative w.r.t. $w^{(j)}$ calculated on a single instance

$$\frac{\partial E}{\partial w^{(j)}} = \begin{cases} \frac{1}{n} \cancel{\sum_{i=1}^{n}} 2x_i^{(j)} (\hat{y}_i - y_i) & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Stochastic Gradient Descent for Linear Regression with Nonnegative Target

(Regularization is omitted for simplicity)

- Procedure:
  - **Step 1** Set $w^{(1)}, w^{(2)}, \ldots, w^{(m)}$ to some random values
  - **Step 2** $x_i \leftarrow$ **select a random instance of the training data**
    for j in $1 \ldots m$
    $$w^{(j)} \leftarrow w^{(j)} - \epsilon \frac{\partial E}{\partial w^{(j)}}$$
  - **Step 3** Repeat Step 2 as long as you can non-negligibly decrease the error

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Linear Regression with Nonnegative Target

- The gradient may be zero for some instances with substantial error.
- Therefore, the previous algorithm may find a model that fits some of the instances well whereas some other instances are ignored.
- What about using several models „together“?

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Regression vs. Classification

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Regression vs. Classification

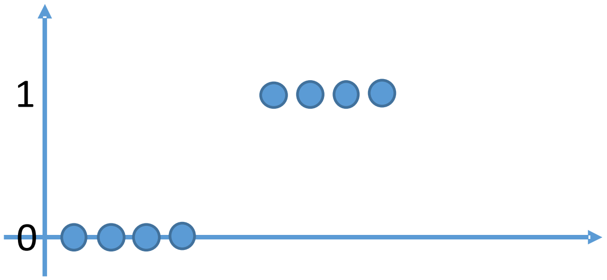
- **Regression**: target is continuous
  - price of an appartment, product, etc.
  - UPDRS score
- **Classification**: target is discrete (it takes the value from a finite set)
  - Recognition of handwritten digits
  - Is the e-mail spam or not?
    (class 0: „spam“, class 1: „not spam)
- Regressor: the model used to solve regression tasks
- Classifier: the model used to solve classification tasks

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Logistic Regression
(it is a classifier, not a regressor!)

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

„Classification via Regression“



Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

„Classification via Regression“

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Hypothesis

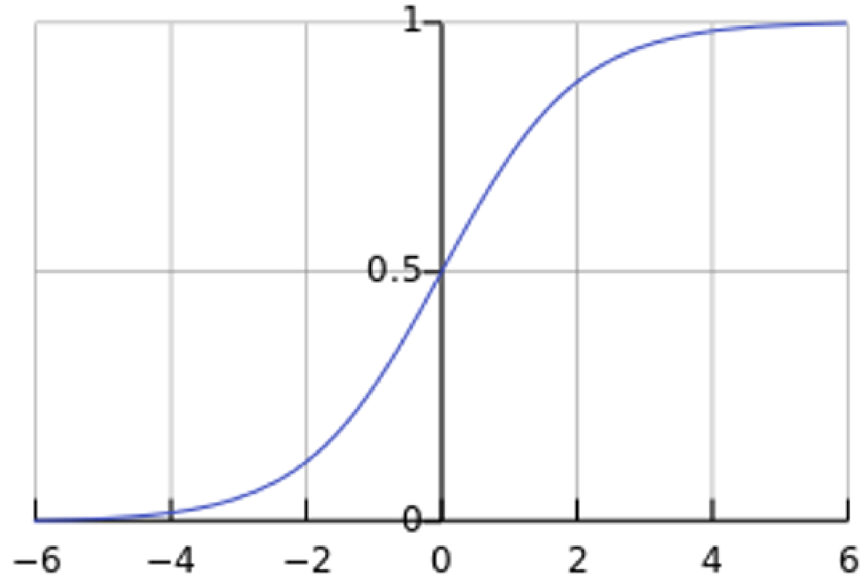
$$\mathbf{x} = (x^{(0)}, x^{(1)}, ..., x^{(m)}) \qquad x^{(0)} = 1 \qquad x^{(i)} \in \mathbf{R}$$

$$\mathbf{w} = (w^{(0)}, w^{(1)}, ..., w^{(m)}) \qquad w^{(i)} \in \mathbf{R}$$

- Linear Regression: $\hat{y} = \mathbf{wx}$
- Logistic Regression: $\hat{y} = \sigma(\mathbf{wx})$

Sigmoid function
(a.k.a. logistic function)

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



- Probabilistic interpretation

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Objective Function

- Linear Regression (L_2 regularization):

$$E = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 + \lambda \sum_{i=1}^{p}(w^{(i)})^2$$

- Can we use the „same“ function with $\hat{y} = \sigma(\mathbf{wx})$?

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

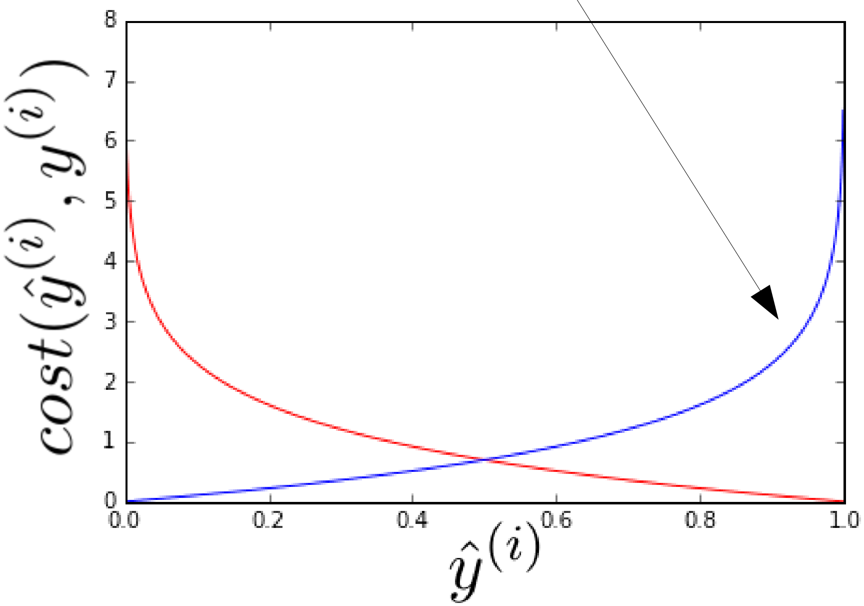
Objective Function – Logistic Regression

- Cost for the i-th instance:

$$cost(\hat{y}^{(i)}, y^{(i)}) = \begin{cases} -log(\hat{y}^{(i)}) & \text{if } y^{(i)} = 1 \\ -log(1 - \hat{y}^{(i)}) & \text{if } y^{(i)} = 0 \end{cases}$$

- Total cost
(for a set of instances):

$$\frac{1}{n} \sum_{i=1}^{n} cost(\hat{y}^{(i)}, y^{(i)})$$



Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Objective Function – Logistic Regression

- Cost for the i-th instance:

$$cost(\hat{y}^{(i)}, y^{(i)}) = \begin{cases} -log(\hat{y}^{(i)}) & \text{if } y^{(i)} = 1 \\ -log(1 - \hat{y}^{(i)}) & \text{if } y^{(i)} = 0 \end{cases}$$

$$= -y^{(i)} log(\hat{y}^{(i)}) - (1 - y^{(i)}) log(1 - \hat{y}^{(i)})$$

- Total cost
(for a set of instances):

$$-\frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} log(\hat{y}^{(i)}) + (1 - y^{(i)}) log(1 - \hat{y}^{(i)}) \right)$$

- This cost function is also known as *cross-entropy.*

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Cross-entropy vs. Mean Squared Error

See **Figure 5** in

X Glorot, Y Bengio (2010):
Understanding the difficulty of training deep feedforward neural networks

http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf?hc_location=ufi

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Summary

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Summary

- Selection of appropriate hyperparameters is part of the training process
- Batch gradient descent, stochastic gradient descent
- Linear Regression with Nonnegative Target
- Classification
- Logistic Regression

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Essential Concepts

- Parameter – Hyperparameter
- Stochastic Gradient Descent – Batch Gradient Descent
- Batch Size
- Training data – Validation Data – Test data
- Regression – Classification
- Regressor – Classifier
- Class
- Cross-entropy