Gradient Descent

Johanni Brea

Einführung Machine Learning

GYMINF 2022

- For linear regression there is an analytical solution that minimizes the RMSE.

Optimization in Machine Learning

- For linear regression there is an analytical solution that minimizes the RMSE.
- For logistic regression (and most other methods) there is no analytical solution.

Optimization in Machine Learning

- For linear regression there is an analytical solution that minimizes the RMSE.
- For logistic regression (and most other methods) there is no analytical solution.
- For many models there are specialized optimizers.

Optimization in Machine Learning

- For linear regression there is an analytical solution that minimizes the RMSE.
- For logistic regression (and most other methods) there is no analytical solution.
- For many models there are specialized optimizers.
- There is a course at EPFL on Optimization for machine learning
  `https://edu.epfl.ch/coursebook/en/optimization-for-machine-learning-CS-439`

Optimization in Machine Learning

- For linear regression there is an analytical solution that minimizes the RMSE.
- For logistic regression (and most other methods) there is no analytical solution.
- For many models there are specialized optimizers.
- There is a course at EPFL on Optimization for machine learning
  `https://edu.epfl.ch/coursebook/en/optimization-for-machine-learning-CS-439`
- A simple optimizer that works usually well for parametric models is

  gradient descent.

Table of Contents

Gradient Descent

1. Input: loss function L, initial guess $\beta^{(0)} = \left(\beta_0^{(0)}, \ldots, \beta_p^{(0)}\right)$ learning rate η, maximal number of steps T.
2. For $t = 1, \ldots, T$
   - $\delta_i = \eta \dfrac{\partial L}{\partial \beta_i}\left(\beta^{(t-1)}\right)$
   - $\beta_i^{(t)} = \beta_i^{(t-1)} - \delta_i$
3. Return $\beta^{(T)}$

Gradient Descent

1. Input: loss function L, initial guess $\beta^{(0)} = \left(\beta_0^{(0)}, \ldots, \beta_p^{(0)}\right)$ learning rate η, maximal number of steps T.
2. For $t = 1, \ldots, T$
   - $\delta_i = \eta \dfrac{\partial L}{\partial \beta_i}\left(\beta^{(t-1)}\right)$
   - $\beta_i^{(t)} = \beta_i^{(t-1)} - \delta_i$
3. Return $\beta^{(T)}$

Automatic Differentiation software uses the chain rule and symbolic derivatives for primitive functions, to compute the derivative of almost any code we write.

Practical Considerations

- Choosing a good learning rate can be tricky.
- Scaling the loss function has an impact on gradient descent.
  It is e.g. advisable to have L independent of the size of the data set, e.g. replace $L = \sum_{i=1}^{n}(y_i - \beta x_i)^2$ by $L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta x_i)^2$.
- Additive constants in the loss function L that do not depend on the parameters have no impact on gradient descent; they are often removed from the loss function.
- Preprocessing the input and output may have a strong effect on gradient descent. There are domain-specific “best preprocessing practices” (e.g. for images or audio). Standardizing inputs (and outputs in the case of regression) is an option.

Table of Contents

Convex and Non-Convex Loss Functions

Globally Convex Loss Function

Loss function has a unique global minimum

Convex and Non-Convex Loss Functions

Globally Convex Loss Function

Loss function has a unique global minimum

From any initial condition there is a path towards the global minimum along which the loss is monotonically decreasing. The same solution is found by gradient descent independently of the initial condition.

Convex and Non-Convex Loss Functions

Globally Convex Loss Function

Loss function has a unique global minimum

From any initial condition there is a path towards the global minimum along which the loss is monotonically decreasing. The same solution is found by gradient descent independently of the initial condition.

The loss function of (multiple) (logistic) linear regression (with L1 or L2 regularization) is globally convex.

Convex and Non-Convex Loss Functions

Globally Convex Loss Function

Loss function has a unique global minimum

From any initial condition there is a path towards the global minimum along which the loss is monotonically decreasing. The same solution is found by gradient descent independently of the initial condition.

The loss function of (multiple) (logistic) linear regression (with L1 or L2 regularization) is globally convex.

Non-Convex Loss Function

Loss function has multiple local minima

Convex and Non-Convex Loss Functions

Globally Convex Loss Function

Loss function has a unique global minimum

From any initial condition there is a path towards the global minimum along which the loss is monotonically decreasing. The same solution is found by gradient descent independently of the initial condition.

The loss function of (multiple) (logistic) linear regression (with L1 or L2 regularization) is globally convex.

Non-Convex Loss Function

Loss function has multiple local minima

The solution of gradient descent depends on the initial condition.

Table of Contents

Stochastic Gradient Descent (SGD)

Computing the loss over all samples $1, \ldots, n$ can be computationally costly.
A subset of the training data may be sufficient to estimate the gradient direction.

Stochastic Gradient Descent (SGD)

Computing the loss over all samples $1, \dots, n$ can be computationally costly.
A subset of the training data may be sufficient to estimate the gradient direction.

1. Input: loss function L, initial guess $\beta^{(0)} = \left(\beta_0^{(0)}, \dots, \beta_p^{(0)}\right)$ learning rate η, maximal number of steps T, batch size B.
2. For $t = 1, \dots, T$
   - Determine batch of training indices $\mathcal{I}$
   - $\delta_i = \eta \dfrac{\partial L}{\partial \beta_i}\left(\beta^{(t-1)}; \mathcal{I}\right)$
   - $\beta_i^{(t)} = \beta_i^{(t-1)} - \delta_i$
3. Return $\beta^{(T)}$

where $L(\beta; \mathcal{I})$ is the loss function evaluated on the training samples with indices in $\mathcal{I}$, e.g.

$$L\left(\beta; \mathcal{I}\right) = \frac{1}{B} \sum_{i \in \mathcal{I}} \left(y_i - x_i^T \beta\right)^2$$

Stochastic Gradient Descent (SGD)

Computing the loss over all samples $1, \ldots, n$ can be computationally costly.
A subset of the training data may be sufficient to estimate the gradient direction.

1. Input: loss function L, initial guess $\beta^{(0)} = \left(\beta_0^{(0)}, \ldots, \beta_p^{(0)}\right)$ learning rate η, maximal number of steps T, batch size B.
2. For $t = 1, \ldots, T$
   - Determine batch of training indices $\mathcal{I}$
   - $\delta_i = \eta \frac{\partial L}{\partial \beta_i}\left(\beta^{(t-1)}; \mathcal{I}\right)$
   - $\beta_i^{(t)} = \beta_i^{(t-1)} - \delta_i$
3. Return $\beta^{(T)}$

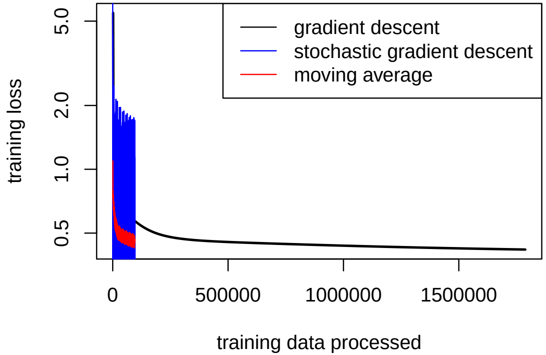
where $L(\beta; \mathcal{I})$ is the loss function evaluated on the training samples with indices in $\mathcal{I}$, e.g.

$$L(\beta; \mathcal{I}) = \frac{1}{B} \sum_{i \in \mathcal{I}} \left(y_i - x_i^T \beta\right)^2$$

Example $B = 5$

	batch 1	batch 2	batch 3	ba
$\mathcal{I}$	1 8 3 13 93	9 14 2 26 31	$\cdots$	

Example Learning Curve



The training loss on batches of size 32 is very variable. But if we look at the moving average over the training loss of 50 subsequent batches, we see that stochastic gradient descent drops to a fairly low loss after processing far less training data than gradient descent.

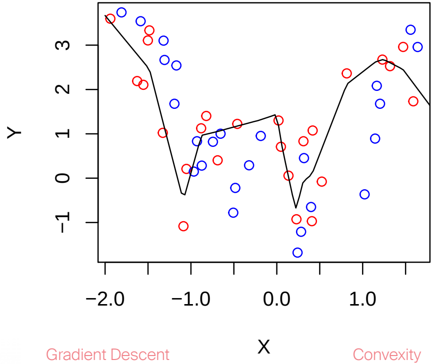
Table of Contents

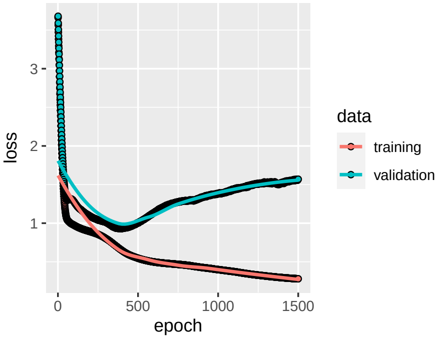
Early Stopping

Start with small weights and stop gradient descent when validation loss starts to increase.

training data validation data

black line: a flexible neural network trained with gradient descent





Quiz

- Bei konstanter Lernrate ist solange der Gradient nicht Null ist, nimmt der Trainingsfehler ab in jedem Gradientenabstiegsschritt.

Quiz

- Bei konstanter Lernrate ist solange der Gradient nicht Null ist, nimmt der Trainingsfehler ab in jedem Gradientenabstiegsschritt.
- In jedem Gradientenabstiegsschritt gibt es eine positive Lernrate, so dass der Trainingsfehler abnimmt, solange der Gradient nicht Null ist.

Quiz

- Bei konstanter Lernrate ist solange der Gradient nicht Null ist, nimmt der Trainingsfehler ab in jedem Gradientenabstiegsschritt.
- In jedem Gradientenabstiegsschritt gibt es eine positive Lernrate, so dass der Trainingsfehler abnimmt, solange der Gradient nicht Null ist.
- In jedem Schritt von <u>stochastischem</u> Gradientenabstieg gibt es eine positive Lernrate, so dass der Trainingsfehler abnimmt, solange der Gradient nicht Null ist.

Quiz

- Bei konstanter Lernrate ist solange der Gradient nicht Null ist, nimmt der Trainingsfehler ab in jedem Gradientenabstiegsschritt.
- In jedem Gradientenabstiegsschritt gibt es eine positive Lernrate, so dass der Trainingsfehler abnimmt, solange der Gradient nicht Null ist.
- In jedem Schritt von <u>stochastischem</u> Gradientenabstieg gibt es eine positive Lernrate, so dass der Trainingsfehler abnimmt, solange der Gradient nicht Null ist.
- Sei n die Anzahl Trainingsbeispiele. Die Berechnung eines Gradientenabstiegsschrittes braucht n/B so viel Zeit wie die Berechnung eines Schrittes mit stochastischem Gradientenabstieg mit Batchgrösse B.

Quiz

- Bei konstanter Lernrate ist solange der Gradient nicht Null ist, nimmt der Trainingsfehler ab in jedem Gradientenabstiegsschritt.
- In jedem Gradientenabstiegsschritt gibt es eine positive Lernrate, so dass der Trainingsfehler abnimmt, solange der Gradient nicht Null ist.
- In jedem Schritt von stochastischem Gradientenabstieg gibt es eine positive Lernrate, so dass der Trainingsfehler abnimmt, solange der Gradient nicht Null ist.
- Sei n die Anzahl Trainingsbeispiele. Die Berechnung eines Gradientenabstiegsschrittes braucht n/B so viel Zeit wie die Berechnung eines Schrittes mit stochastischem Gradientenabstieg mit Batchgrösse B.
- Early stopping im Gradientenabstieg führt zu Modellen mit kleinerem Bias aber grösserer Varianz, verglichen mit Gradientenabstieg ohne early stopping.